

國立臺灣大學電機資訊學院暨中央研究院

資料科學學位學程

碩士論文

Data Science Degree Program

College of Electrical Engineering and Computer Science

National Taiwan University and Academia Sinica

Master's Thesis

基於分割法的無母數迴歸

Non-parametric Regression Using Partitioning Methods

郭晉良

Chin-Liang Kuo

指導教授：張明中、楊鈞浩 博士

Advisor: Ming-Chung Chang, Chun-Hao Yang Ph.D.

中華民國 113 年 7 月

July, 2024





Acknowledgements

從化學、企管、再到資料科學領域，我的學術生涯跨了不小的幅度，從討厭數學，到喜歡統計，再到以機器學習作為吃飯的傢伙，人的興趣總是多變而難以捉摸，誰也無法確定終點在哪，唯一能做的事就是把握當下的求知慾，將這股力量化為真實的知識資本。

在此，我要衷心感謝我的指導教授張明中老師，老師在我初入此領域時最懵懂無知的時候，願意收我為指導學生，並在我學術生涯的各個階段提供了無數的指導和支持，老師總是站在學生的角度上為我們著想，在許多層面上都給予非常積極的幫助，老師的耐心教導和無私奉獻對我的學業以至於未來的成就都有相當深遠的影響，我永遠感激您的啟發和指導。

感謝在這段艱難而充實的學術旅程中與我共同奮鬥的戰友們，每個專案都做出超乎分配的工作量，從未搭便車，你們始終展現出極大的熱心和責任感，未來定會有非凡的成就，未來若有機會，我很樂意再次與你們攜手。

最後，我要特別感謝我的家人，你們在我最欠缺資源的時候，給予我最大的支持及陪伴，成為我最堅強的後盾，感謝你們永遠支持我的決定，你們的愛和鼓勵讓我在面對困難時充滿勇氣和信心，並堅定地朝著我的目標前進。

在這段學術旅程的尾聲，我深感自己是幸運的，擁有如此多的良師益友和親人的支持與陪伴，我將帶著你們的期望與祝福，繼續前行！



摘要

本文研究了一種基於分割法的無母數迴歸技術，旨在提升迴歸問題的預測精度。本文首先介紹了使用機器學習處理分類及迴歸問題的基本概念，再針對處理迴歸問題的常見方法進行探討，最後聚焦在本文所使用的無母數迴歸方法。在本文的核心研究中，提出了一種新的演算法 PE-Kmeans，該演算法在非監督式學習中的 K-means 演算法的基礎上進行改進，形成二階段的分群方法。第一階段在輸出空間進行 K-means 分群，第二階段則在每個母群中進行輸入空間的再次 K-means 分群，這種方法充分考慮了輸出變量的信息，使得它成為一個監督式學習模型，可以用來處理迴歸問題。本文以 Supervised Compression 及著名的 Regression Tree 作為比較模型，前項方法通過選擇性以輸入空間或輸出空間作為分割中心點，逐步將輸入空間分割成不規則的 Voronoi region 子區域，後項方法則是透過二元分類將輸入空間分割為長方形。通過對模擬資料和真實世界資料的實驗，本文驗證了前述三種方法在不同情境下的性能，實驗結果表明，PE-Kmeans 在處理相對不平滑的函數及真實世界資料時，能夠更有效地進行預測。

關鍵字：機器學習、無母數迴歸、監督式學習、分割法、資料壓縮、迴歸樹、群集分析



Abstract

This study investigates a non-parametric regression technique based on segmentation, aimed at enhancing the prediction accuracy of regression problems. The paper first introduces the basic concepts of using machine learning to handle classification and regression problems. It then discusses common methods for addressing regression issues, with a focus on the non-parametric regression method employed in this study. At the core of this research, a new algorithm called PE-Kmeans is proposed. This algorithm improves upon the K-means algorithm used in unsupervised learning, forming a two-stage clustering method. In the first stage, K-means clustering is performed in the output space. In the second stage, K-means clustering is again performed in the input space within each parent cluster. This method fully considers the information of the output variables, making it a supervised learning model suitable for handling regression problems. The study compares the proposed method with Supervised Compression and the well-known Regression Tree. The former method selectively uses either the input space or the output space as the segmentation center, gradually dividing the input space into irregular Voronoi region sub-regions. The latter method segments the input space into rectangles through binary classification. Through experiments on simulated and real-world data, the study validates the performance of the three aforementioned methods under different scenarios. The experimental results demonstrate that PE-Kmeans can more effectively make predictions when dealing with relatively non-smooth functions and real-world data.

Keywords: Machine Learning, Non-parametric Regression, Supervised Learning, Segmentation Method, Data Compression, Regression Tree, Cluster Analysis



Contents

	Page
Verification Letter from the Oral Examination Committee	i
Acknowledgements	ii
摘要	iii
Abstract	iv
Contents	v
List of Figures	viii
List of Tables	x
Denotation	xi
Chapter 1 緒論	1
1.1 監督式學習	1
1.2 處理迴歸問題的常見方法	2
1.2.1 線性模型	2
1.2.2 集成學習	4
1.2.3 類神經網路	4
1.3 無母數迴歸	5
1.4 研究目的	6
1.5 論文架構	6
Chapter 2 文獻回顧	7
2.1 Regression Tree	7
2.2 Supervised Compression	8

2.3	Partitioning Estimate	9
Chapter 3	研究方法	10
3.1	模型	10
3.1.1	以 Supervised Compression 做 Partitioning Estimate	10
3.1.2	以 PE-Kmeans 做 Partitioning Estimate	11
3.1.3	模型超參數設定	12
3.2	比較方法之指標	13
3.2.1	預測誤差	13
3.2.2	組間、組內之殘差平方和分析	13
3.2.3	運行時間	14
3.2.4	可視化	14
Chapter 4	模擬資料分析	15
4.1	資料集設定	15
4.2	模擬	15
4.2.1	Two Dimensional Michalewicz 函數	15
4.2.1.1	可視化	21
4.2.2	Dropwave 函數	26
4.2.2.1	可視化	31
4.2.3	OTL Circuit 函數	36
4.2.4	Piston 函數	41
4.2.5	Borehole 函數	44
4.2.6	函數模擬實驗總結	46
Chapter 5	真實資料分析	48
5.1	資料集設定	49
5.2	預測誤差	49
5.3	組內組間變異分析	50
5.3.1	組內變異	50



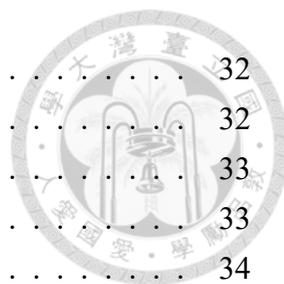
5.3.2	組間變異	50
5.4	運行時間	51
Chapter 6	結論與未來展望	52
References		54





List of Figures

3.1	Supervised Compression	11
4.1	Two Dimensional Michalewicz 之 RMSE (SNR = 500)	16
4.2	Two Dimensional Michalewicz 之組內變異 (SNR = 500)	17
4.3	Two Dimensional Michalewicz 之組間變異 (SNR = 500)	17
4.4	Two Dimensional Michalewicz 之 RMSE (SNR = 50)	18
4.5	Two Dimensional Michalewicz 之組內變異 (SNR = 50)	18
4.6	Two Dimensional Michalewicz 之組間變異 (SNR = 50)	19
4.7	Two Dimensional Michalewicz 之 RMSE (SNR = 5)	19
4.8	Two Dimensional Michalewicz 之組內變異 (SNR = 5)	20
4.9	Two Dimensional Michalewicz 之組間變異 (SNR = 5)	20
4.10	Two Dimensional Michalewicz 之運行時間 (秒)	21
4.11	原始函數之等高線圖	22
4.12	Regression Tree 模型之等高線圖	22
4.13	Supervised Compression 模型之等高線圖	23
4.14	PE-Kmeans 模型之等高線圖	23
4.15	Regression Tree 模型之分區圖	24
4.16	Supervised Compression 模型之分區圖	24
4.17	PE-Kmeans 模型之分區圖	25
4.18	PE-Kmeans 模型之分區圖 (二)	25
4.19	Dropwave 之 RMSE (SNR = 500)	26
4.20	Dropwave 之組內變異 (SNR = 500)	27
4.21	Dropwave 之組間變異 (SNR = 500)	27
4.22	Dropwave 之 RMSE (SNR = 50)	28
4.23	Dropwave 之組內變異 (SNR = 50)	28
4.24	Dropwave 之組間變異 (SNR = 50)	29
4.25	Dropwave 之 RMSE (SNR = 5)	29
4.26	Dropwave 之組內變異 (SNR = 5)	30
4.27	Dropwave 之組間變異 (SNR = 5)	30
4.28	Dropwave 之運行時間 (秒)	31



4.29	原始函數之等高線圖	32
4.30	Regression Tree 模型之等高線圖	32
4.31	Supervised Compression 模型之等高線圖	33
4.32	PE-Kmeans 模型之等高線圖	33
4.33	Regression Tree 模型之分區圖	34
4.34	Supervised Compression 模型之分區圖	34
4.35	PE-Kmeans 模型之分區圖	35
4.36	PE-Kmeans 模型之分區圖 (二)	35
4.37	OTL Circuit 之 RMSE (SNR = 500)	36
4.38	OTL Circuit 之組內變異 (SNR = 500)	37
4.39	OTL Circuit 之組間變異 (SNR = 500)	37
4.40	OTL Circuit 之 RMSE (SNR = 50)	38
4.41	OTL Circuit 之組內變異 (SNR = 50)	38
4.42	OTL Circuit 之組間變異 (SNR = 50)	39
4.43	OTL Circuit 之 RMSE (SNR = 5)	39
4.44	OTL Circuit 之組內變異 (SNR = 5)	40
4.45	OTL Circuit 之組內變異 (SNR = 5)	40
4.46	OTL Circuit 之運行時間 (秒)	41
4.47	Piston 之 RMSE (SNR = 500)	42
4.48	Piston 之組內變異 (SNR = 500)	42
4.49	Piston 之組間變異 (SNR = 500)	43
4.50	Piston 之運行時間 (秒)	44
4.51	Borehole 之 RMSE (SNR = 500)	44
4.52	Borehole 之組內變異 (SNR = 500)	45
4.53	Borehole 之組間變異 (SNR = 500)	45
4.54	Borehole 之運行時間 (秒)	46
5.1	RMSE 比較	49
5.2	組內變異比較	50
5.3	組間變異比較	51
5.4	運行時間比較	51



List of Tables

5.1	Parameter Ranges	48
-----	----------------------------	----



Denotation

Y 輸出變量／應變量／被解釋變量／相依變量

X 輸入變量／自變量／解釋變量／獨立變量

SNR 訊號雜訊比

RMSE 均方根誤差



Chapter 1 緒論

1.1 監督式學習

監督式學習是機器學習的一種方法，通過觀察訓練資料中的輸入和對應的預期輸出來建立模型，這些訓練資料由事先標記的範例組成，其中包括輸入物件和預期輸出，該方法的目標是通過訓練資料中的模式來預測新的輸入的輸出，這需要將已知資料一般化，以便對未知情況進行推斷，監督式學習可以生成全域模型或區域模型，前者將輸入映射到預期輸出，後者實現了對應關係的一種局部形式。解決監督式學習問題的步驟包括決定訓練資料的形式、搜集資料、決定輸入特徵的表示方法、選擇適合的學習函數和演算法，以及設計並調整模型參數，監督式學習使用各種分類器，如 Neural Network、Support Vector Machine、k-Nearest Neighbors、Gaussian Mixture Model、Naive Bayes Classifier、Decision Tree 等，來解決不同的問題。總言之，監督式學習是一種通過觀察和標記的訓練資料來建立模型的方法，以便預測新的輸入的輸出。

分類問題是指當目標是將資料分類到預定的幾個類別中時所遇到的問題，在這種情況下，模型的任務是根據訓練資料中提供的已知類別標籤，將新的資料分配給這些類別之一。舉例來說，假設你擁有一個電子商務網站，想要預測用戶是否會流失。在這個案例中，你可能會搜集用戶的性別、年齡、購買頻率等資料，並且將用戶分為「會流失」和「不會流失」這兩個類別。監督式學習模型的目標是學習這些資料的模式，以便將新用戶分類為這兩個類別中的一個，分類問題也適用於許多其他情況，例如垃圾郵件偵測、文章分類、語種偵測等。在這些情況下，模型需要根據資料的特徵將其歸類到預定的類別中，這些類別可能是「垃圾郵件」和「非垃圾郵件」、不同的文章主題類別、或是不同的語言等。



迴歸問題是監督式學習中的一種，用於預測連續值的輸出變量，以電子商務中的顧客流失率問題為例，企業可以利用迴歸分析來建立一個模型，該模型能夠預測顧客流失率，以幫助制定相應的客戶保留策略。舉例來說，假設一家電子商務公司收集到了每位顧客的相關資料，如購買頻率、最近一次購買時間、平均購買金額等，以及對應的顧客流失情況，透過建立一個迴歸模型，公司可以找到這些顧客行為特徵與顧客流失率之間的關係。一旦建立了模型，公司就可以將新顧客的資料輸入到模型中，進而預測顧客流失率，基於這些預測，企業可以制定相應的客戶保留策略，如提供折扣、推出促銷活動、改善客戶服務等，以降低顧客流失率。總之，迴歸問題旨在通過建立特徵和連續輸出變量之間的關係來預測連續值的輸出變量，在電子商務中的顧客流失率問題中，這意味著建立一個模型，該模型能夠根據顧客的行為特徵來預測顧客流失率的值。

本研究提出 PE-Kmeans 演算法是一種在 K -means 演算法的基礎上改變而生，原始的 K -means 演算法屬於非監督式演算法，用於無標籤資料的分類問題 [1]，而 PE-Kmeans 演算法由兩層 K -means 演算法所構成，第一層為輸出空間的聚類，第二層則為輸入空間的聚類，因此適用於有標籤的資料，也能夠處理迴歸問題，故屬於監督式學習。

1.2 處理迴歸問題的常見方法

1.2.1 線性模型

線性模型用於描述輸入變量和輸出變量之間的線性關係，因此核心精神就是認為輸出變量可以通過輸入變量的線性組合來進行預測或解釋，簡單線性模型包括簡單線性迴歸和多元線性迴歸，在簡單線性迴歸中，只有一個輸入變量，而在多元線性迴歸中，有多個輸入變量。

線性模型的一般形式可以表示為：

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon$$

其中， Y 是輸出變量， X_1, X_2, \dots, X_n 是輸入變量， $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ 是模型的係數， ε 是誤差項，線性模型的目標是找到最佳的係數來最好地擬合數據，使得輸出變量的預測值與實際觀察值之間的殘差（誤差）最小化。這其中，誤差項 ε 是假設獨立且具有常態分佈的，這意味著誤差項的平均值應為零，且其變異數在各個輸入變量水平上應相等。至於廣義線性模型（Generalized Linear Models, GLMs）是對線性模型的擴展，允許輸出變量不必遵從常態分佈，並且可以透過鏈接函數將預測的線性組合轉換為實際的預測值，儘管在名稱上包含「線性」一詞，但廣義線性模型並不一定是嚴格的線性模型。

廣義線性模型通常應用於以下情況：

1. 二元迴歸模型（Binary Regression）：用於解釋二元輸出變量（例如，是/否、成功/失敗等）和多個輸入變量之間的關係。在二元迴歸模型中，使用的鏈接函數通常是 logit 函數，稱為 Logistic 迴歸模型。
2. 多項式迴歸模型（Polynomial Regression）：當輸入變量和輸出變量之間的關係不是線性的時候，可以使用多項式迴歸模型，它通過添加輸入變量的高次方項來擴展線性模型，從而擬合非線性關係。
3. 嶺迴歸模型（Ridge Regression）、Lasso 迴歸模型（Lasso Regression）和彈性網迴歸模型（Elastic Net Regression）：這些模型都用於處理多重共線性的問題，即輸入變量之間存在較高的相關性。嶺迴歸模型在線性迴歸模型的損失函數中添加了一個正則化項，以限制係數的大小，從而減少過度擬合。Lasso 迴歸使用的正則化項是係數的絕對值，使其具有將係數推進零的傾向，從而實現特徵選擇的效果。彈性網迴歸模型結合了嶺迴歸模型和 Lasso 迴歸模型，同時應用了 L1 和 L2 正則化項，以平衡兩者之間的優勢。
4. 加權線性迴歸模型（Weighted Linear Regression Models）：將不同觀測值賦予不同的權重，以反映它們對模型擬合的重要性。

這些模型在統計學、機器學習和其他領域中都有廣泛的應用，可根據不同的數據特點和任務需求來選擇適當的模型。



1.2.2 集成學習

集成學習 [2] 是一種統計學和機器學習方法，旨在通過組合多個學習算法的預測來提高預測性能。單個學習算法可能難以找到一個良好的假設來準確預測特定問題，因此，集成學習利用多個不同的學習算法，希望組合它們的預測，從而獲得比單個算法更好的性能。基本思想是將多個基本的學習算法組合在一起，形成一個更強大的整體模型。這些基本的學習算法可以是相同的，也可以是不同的。常見的集成方法包括 Bagging、Boosting、Bayesian Model Averaging、Stacking 等。

在集成學習中，模型之間的差異往往是關鍵的。當模型之間存在顯著差異時，集成通常會產生更好的結果。因此，許多集成方法試圖促進模型之間的多樣性。這種多樣性可以通過使用不同的學習算法、不同的特徵子集或不同的訓練數據等方式實現。集成學習方法通常需要比單個模型更多的計算資源，因為需要訓練和組合多個模型。然而，在實際應用中，集成學習通常能夠提高預測性能，特別是當基本模型之間存在差異時。以隨機森林為例，它是一種集成學習方法，通過組合多棵存在差異的決策樹的預測來實現更準確的預測。

1.2.3 類神經網路

類神經網路 (Artificial Neural Networks, ANNs) [3] 模仿生物神經網路的結構和功能，這一概念最早於 20 世紀 40 年代提出，但直到 1980 年代，隨著計算能力的提升和新的學習算法的出現，ANNs 才開始受到廣泛關注，其基本組成要素包括結構、神經元、激活函數、學習規則和學習過程。

1. 結構 (Architecture)：ANNs 通常包含輸入層、隱藏層和輸出層。輸入層接收外部資料，隱藏層用於處理中間表示，而輸出層生成最終結果。
2. 神經元 (Neurons)：神經元是 ANNs 的基本單元，接收前一層的輸入並使用權重和激活函數計算輸出。
3. 激活函數 (Activation Function)：將神經元的加權輸入轉換為輸出。常見的激活函數包括 linear、sigmoid、ReLU、softmax 和 tanh 等等。

- 
4. 學習規則 (Learning Rule)：指導權重如何根據訓練資料調整的規則。常見的算法包括反向傳播和梯度下降。
 5. 學習過程 (Training Process)：透過訓練資料，ANNs 反覆迭代學習，調整權重以最小化損失函數。通常將資料集分為訓練集、驗證集和測試集。

類神經網路廣泛應用於機器視覺、語音辨識、自然語言處理等領域，能有效處理複雜非線性問題(也能處理線性問題)。儘管如此，它們仍然面臨著許多挑戰，包括大量訓練資料的需求、超參數的調整和高計算成本。

1.3 無母數迴歸

無母數迴歸 [4] 用於在不假設預先確定的函數形式的情況下估計輸出變量 Y 與一個或多個輸入變量 X 之間的關係。在這種情況下， $f_0(x) = E(Y|X = x)$ 代表迴歸函數，表示在 $X = x$ 的情況下 Y 的期望值。無母數迴歸的主要目標是使用獨立且同分佈的樣本 $(x_1, y_1), \dots, (x_n, y_n)$ 從與 (X, Y) 相同的聯合分佈中抽取，構建對 f_0 的估計 \hat{f} 。

無母數迴歸中的兩種常見設置如下：隨機輸入設置：該設置假設 X_1, \dots, X_n 為隨機輸入。每個 y_i 則被生成為 $y_i = f_0(X_i) + \epsilon_i$ ，其中 $\epsilon_1, \dots, \epsilon_n$ 是均值為零的獨立同分佈的隨機誤差。此外，還假設每個誤差項 ϵ_i 獨立於 X_i 。固定輸入設置：在這種情況下， X_1, \dots, X_n 為固定輸入，並且模型保持不變： $y_i = f_0(X_i) + \epsilon_i$ ，其中 $\epsilon_1, \dots, \epsilon_n$ 是均值為零的獨立同分佈的隨機誤差。雖然無母數迴歸估計通常不會針對隨機或固定設置進行明確定義，但某些方法可能會假設特定條件。例如，某些估計方法如小波可能會假設均勻間隔的固定輸入。理論性陳述在隨機和固定輸入設置之間可能會有所不同，假設固定輸入點，尤其是均勻間隔的點時，某些結果可能更為精確。無母數迴歸涵蓋了各種技術，如 Regression Tree、Kernel Smoothing、Local Polynomial、Regression Spline、Smoothing Spline、Reproducing Kernel Hilbert Space 和 Wavelet。這些方法既可以應用於單變量，也可以應用於多變量情況，儘管討論通常圍繞著單變量情況 ($d = 1$)，因為在更高維度中存在計算和統計效率方面的問題。討論中討論的一種流行的無母數迴歸技術是

k -nearest neighbor，它通過對於給定點 X 的 k 個最近鄰居的響應進行平均來估計迴歸函數。儘管簡單且廣泛使用， k -nearest neighbor 可能會產生崎嶇的估計，特別是對於較小的 k 值。此外，它受到維度災難（Curse of dimensionality）的影響，隨著維度數量的增加，估計變得顯著更加困難。



1.4 研究目的

根據預測模型分割區塊方式的不同，在不同的函數模擬或真實資料之下，可能具有不同的預測效能，例如 Regression Tree 可能在輸出空間上較方正的函數模擬上表現優良，但在輸出空間上較圓弧或不平滑的函數模擬時就有可能表現得不够理想，主要原因就是 Regression Tree 會在輸入空間上分割出較方正、數量較有限的區塊，在擬合函數的時候就會產生較大的誤差。故本研究欲提出一種演算法，該演算法須具備能夠擬合較圓弧或不平滑函數的特性，以補足類似 Regression Tree 這種模型設計上的不足。

1.5 論文架構

本章旨在引領讀者深入研究核心主題，提供對監督式學習方法及迴歸問題處理方法的進入知識。第二章深入回顧相關文獻，從 Regression Tree 到 Supervised Data Compression，再到 Partitioning Estimate，在不同領域間的各種方法進行綜合分析，提供讀者深入理解現有相關研究的重要參考，並引出研究主軸。第三章開始進入主軸：以原先應用於不同領域的模型進行 Partitioning Estimate，本章涵蓋了所採用模型的詳細解說，以及模型超參數的設定，同時介紹了模型的評估方法。第四章深入探討了對模擬資料的分析，包括資料集的設定和對模型在各種模擬函數下的性能評估，旨在全面評估模型的性能和適用性。第五章進一步將分析擴展到真實世界的資料，著重探討將模型應用於真實數據集的情況。最後，第六章通過總結研究結果並提出未來研究的潛在方向和改進建議，以推動相關領域的進一步發展。



Chapter 2 文獻回顧

2.1 Regression Tree

廣義的迴歸樹 (Regression Tree) 是一種機器學習演算法的類別，通過在特徵空間中進行分割，將數據集切分為多個子集，每個子集中的數據具有相似的輸出變量值，每個分割點根據特徵和閾值選擇，以最大程度地減少子集中輸出變量的差異性，最終，每個葉子節點對應著一個常數預測值，使得迴歸樹能夠捕捉到輸入特徵和輸出變量之間的非線性關係。依照其節點分裂準則的不同，可以細分成許多不同的演算法，最早的演算法是 1963 年 Morgan JN 等人所設計的 Automatic Interaction Detection(AID)[5]，之後發展出 ID3、C4.5[6] 等以 Information Gain 作為分裂準則的演算法，以上演算法一次可以產生多個節點，而本研究所用的迴歸樹是 1984 年 Breiman 所提出的 Classification and Regression tree (CART)[7]，即 CART Regression Tree，其特色是使用 Gini index 作為分裂準則，並且一次只產生兩個節點，是目前最廣為人知的迴歸樹演算法。此外，上述演算法的最佳化方法皆是使用貪婪演算法 (greedy algorithm)，也就是在迭代時只顧及本代的最優選擇，因此可能會使節點過度生長而導致過擬合 (overfitting)，因此我們需要另外在建構出初步模型之後進行後剪枝 (Post-pruning)，後剪枝的方法有許多種，如 Reduced Error Pruning、Minimum-Error Pruning 等 [8]，本研究採用的是 Error-Complexity Pruning，它會先產生一系列被不同數量修剪的樹，然後透過檢查錯誤分類的數量來選擇其中一棵樹。



2.2 Supervised Compression

本研究使用的監督是資料壓縮方法名為 SuperCompress[9]，它是一種用於子抽樣 (Subsampling) 的監督式機器學習演算法，可以取得有代表性的子數據集，可以在巨量資料分析中有效節省計算機效能，有別於對輸入變量空間聚類的無監督方法，SuperCompress 同時考慮對輸出變量空間的聚類，由於考慮了輸出變量的關鍵訊息，可以更有效率地進行子抽樣。

1. 以一維數據為例，首先對 X 空間進行 k -means ($k = 2$) 聚類，獲得 X 空間上的中心 x_1, x_2 ，同時將 X 空間劃分為兩個區塊。
2. 以 x_1, x_2 的對應輸出變量 y_1, y_2 作為預測值，各自計算兩個區塊 (I_i) 內的所有「輸出變量 (Y_j) 與預測值 (y_i) 的誤差平方和」：

$$\sum_{j \in I_i} (Y_j - y_i)^2$$

3. 挑選誤差最大的區塊，並進行以下兩組計算（假設誤差最大的是 x_2 ）。
 - 將此區塊內的原中心放棄，再進行一次 X 空間上 k -means ($k = 2$) 聚類，獲得 x'_2, x_3 。
 - 將此區塊內的原中心放棄，進行一次 Y 空間上 k -means ($k = 2$) 聚類，獲得 y'_2, y_3 ，及與之對應的輸入變量預測值 x'_2, x_3 。
4. 選擇可使上式最小的方式取得 x'_2, x_3 ，至此便有 x_1, x'_2, x_3 三個中心。
5. 重複以上步驟獲得 x'_3, x_4 ，以此類推。

至於總共要選出幾個中心，作者建議觀察其 RMSE、及 adj-R square 到達平穩期時停止演算法。為求穩健性，可調整原本只考慮 Y 空間的損失函數，改為 Y 空間與 X 空間的加權，以上即本研究 SuperCompress 採用的方法，詳見第三章之模型超參數設定。

2.3 Partitioning Estimate



在電腦科學領域中，將一個大問題分解為可以單獨解決的較小子問題，然後組合它們的解決方案來解決原始問題，稱之為 Partitioning Algorithms。而這邊的 Partitioning Estimate[10] 也有異曲同工之妙，它指的是在資料的輸入變量空間進行區域劃分，區域間將獨立進行估計，故在一區域內產出相同的預測值，這種方法在非參數統計方法中，在計算上相對高效，尤其是以二元決策樹類型的方法計算時（迴歸樹即屬於這類）。另一個好處是，對於輸出變量的估計（不管其為連續變量或類別變量），我們僅須關注輸入變量空間中資料集中的分區，這項優勢用來壓縮資料時非常有效，因為它無需花費儲存資源給沒有資料的區域，一個簡單的例子是對二維或三維輸入變量做 Cubic partition，它將整個輸入變量空間等分切割成正方形或立方形，相對於儲存所有資料，我們僅需儲存含有資料的分區，這個分區的數量遠少於資料數，但仍對輸出變量有相當好的估計能力。



Chapter 3 研究方法

3.1 模型

3.1.1 以 Supervised Compression 做 Partitioning Estimate

Supervised Compression 將資料集替換為多個具有代表性的區域中心點，在輸入變量空間中，資料點將被分配到歐式距離 (Euclidean Distance) 最近的區域中心點，這形成稱為 Voronoi region 的區塊，在這個區塊邊界內的原始資料點及未來可能加入的新資料點，相對於其他區域中心點，一定距離區塊內的區域中心點歐式距離最短，意即 Supervised Compression 將整個輸入變量空間切成一個個不規則形狀的 Voronoi region 子集合。值得注意的是，在 Supervised Compression 的演算法中，對於輸出空間變化較大的局部輸入空間，會有密集的區域中心點及子集合，對於輸出空間變化較小的局部輸入空間，會有稀疏的區域中心點及子集合（如圖3.1），這將有助於將儲存資源分配在輸出變量變化較高的局部輸入空間，以在有限儲存資源下提升模型對輸出變量預測能力。做 Partitioning Estimate 時，本演算法搜索距離測試資料點最近的訓練資料點，回傳該點所在之群組的區域中心點的輸出變量作為預測值。

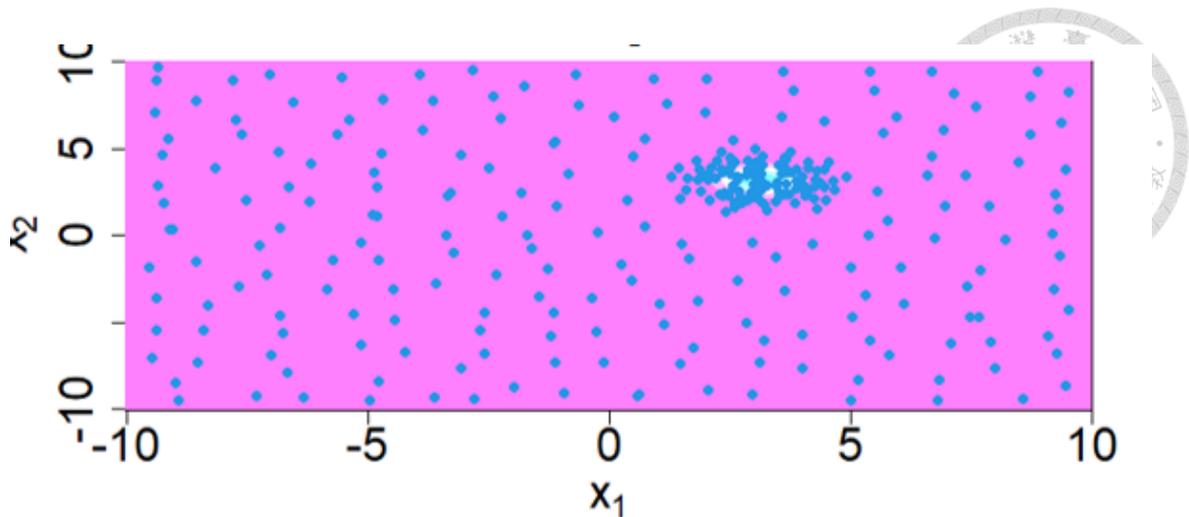


圖 3.1: Supervised Compression

3.1.2 以 PE-Kmeans 做 Partitioning Estimate

PE-Kmeans 為二階段 K-means 分群 [11]，第一階段先將輸出變量做 K-means 分群（稱為母群），達到指定的分群條件後，再分別將每群中的資料以輸入變量做第二次分群（稱為子群），每個母群中的子群數目可不相同，因為了讓輸出變量在分割的區塊中較為集中且連續，而非分散在輸入空間上，第一階段先做輸出變量的分群，且分群條件設置較第二階段輸入變量分群時更嚴格，故此演算法更倚重輸出變量空間的分群，演算法如下：

Algorithm 1 PE-Kmeans

```

Perform Algorithm2 to determine the number of parent clusters
// First-stage clustering
First-stage clustering result  $\leftarrow$  Perform K-means on the output space (Number of parent clusters)
// Second-stage clustering
for Each First-stage cluster do
    Perform Algorithm2 to determine the number of sub-clusters
    Second-stage clustering result  $\leftarrow$  Perform K-means on the input space (Number of sub-clusters)
end for
return Second-stage clustering result

```

做 Partitioning Estimate 時，本演算法以 1nn 搜索距離測試資料點最近的訓練資料點，回傳該點所在群組的輸出變量平均值作為預測值。

Algorithm 2 Dynamic Adjustment of Clusters Algorithm

Number of cluster ($Number$) = 2
while $Number <$ Maximum number of cluster **do**
 Perform k-means clustering on the y_{train} data with $number$ clusters
 Calculate the ratio of between-cluster sum of squares to total within-cluster sum of squares ($Ratio$)
 if $Ratio \geq$ Ratio condition **then**
 break {Increase the number of clusters starting from the $number$ size until the Ratio condition are satisfied}
 end if
 if $Ratio <$ Ratio requirement **then**
 Increment $Number$ by 1
 end if
end while
return Number of cluster

3.1.3 模型超參數設定

1. **Regression Tree:** 採用 R 套件 ‘tree’，其中 tree 函數有兩個影響樹結構的重要超參數：第一個為分裂閾值 mindev，定義是節點內偏差高於 mindev 值時採取繼續分裂，設置為 0.001；另一個是 minsize，定義為節點內可接受的最小資料數，這裡設置為 2（即最小值），這兩項數值設定都寬鬆於預設值，是為了讓樹在後剪枝之前自由發展，雖增加許多計算資源，但能保留了樹的所有可能性，我們不必擔心過擬合問題，因為緊接著就是進行後剪枝，即先使用 cv.tree 以 K 折驗證（K 預設值為 10）的方式取得最佳的節點數（node），接著使用 prune.tree 函數取得剪枝後的模型，這兩項函數的超參數皆採取系統預設值。
2. **Supervised Compression:** 採用 R 套件 ‘supercompress’，其中 supercompress 函數之超參數 lam 為穩健性（Robustness）參數，取值在 0（完全監督）和 1（完全無監督）之間，這邊設置為原文獻中的穩健形式，即 $1/(1+p)$ ，其中 p 為資料之輸入空間的維度，其餘超參數採取系統預設。此外 supercompress 需要給定一個區域中心數 n ，即想要產生多少區域中心，參考原文獻的理論，本研究自行設計一個演算法實現該理論，該理論認為合理的區域中心數量應該在模型的 adj-R square 隨區域中心數的上升，呈現不再陡峭上漲的平緩區域，因此本研究使用移動平均數 MA5（領先指標）及 MA15（落後指

標)，如果在迭代增加區域中心數的過程中 $MA5 - MA15 < 0.005$ ，則使用該數作為區域中心數。



3. **PE-Kmeans:** 本研究自建演算法，共兩個主要函數，分別為輸出變量分群及輸入變量分群，皆呼叫 R 套件 'stats' 之 kmeans 函數執行 k-means 分群，該函數之超參數 iter.max 為最大迭代數，設置為 100（輸出變量分群及輸入變量分群皆相同）；nstart 即要生成多少不同版本的隨機初始中心集，設置為 1000（輸出變量分群及輸入變量分群皆相同）；centers 視資料結構不同有兩個定義，當資料結構為數量時（即本演算法採用），代表中心的數量，在輸出變量分群部分由 2 開始迭代至最多 50，在”組間變異/組內變異”達到”100/1”時提前終止，在輸入變量分群部分由 2 開始迭代至最多 50，在”組間變異/組內變異”達到 95/5 時提前終止。

3.2 比較方法之指標

3.2.1 預測誤差

三組演算法皆採用測試集上的預測均方根誤差（Root Mean Square Error，RMSE）作為評估指標。採用 R 套件 ggplot 以風琴圖、箱型圖疊圖做成圖表。

3.2.2 組間、組內之殘差平方和分析

組內之殘差平方和為被分為同群的資料其所有輸出變量與該組預測輸出變量之間的誤差平方之和；組間之殘差平方和為所有群組的預測輸出變量與整體輸出變量平均值之間的誤差平方之和，呈現方式同上。



3.2.3 運行時間

運行時間僅做為參考，因為除了模型本身的運行時間之外，各演算法的運行時間包含最佳化過程（即模型超參數的搜索），在 PE-Kmeans 中，包含了二階段分群（輸出變量及輸入變量分群）時的區域中心數量的決定，在 Supervised Compression 中亦須決定區域中心數量，但決定方法兩者略有不同，在前一節模型超參數設定中已詳細敘述，在 Regression Tree 中則包含了使用交叉驗證、剪枝來決定分支數量，本演算法採用的 R 套件在超參數搜索上面的程序優化程度可能相對較高，故有運行時間上的優勢，呈現方式同上，單位為秒。

3.2.4 可視化

此節僅用於二維輸入變量的函數，指以輸入變量的兩個維度分別作為橫坐標及縱座標，使我們能夠探索不同演算法對於輸入變量空間中分群結構的影響；此外，通過以等高線圖的形式呈現函數的輸出變量，們可以直觀地比較不同演算法所建立的模型在幾何特徵和分佈上的異同。



Chapter 4 模擬資料分析

4.1 資料集設定

1. **訓練集:** 採用 20000 筆資料，加入信噪比 (SNR) = 5, 50, 500 (視實驗而定)，三種不同程度的噪音，公式如下：

$$\text{函數輸出值} = F(\text{輸入變量})$$

$$\text{噪音值} = \sqrt{\frac{\text{var}(\text{函數輸出值})}{\text{SNR}}} \times \text{標準常態隨機變數}$$

$$\text{輸出變量} = \text{函數輸出值} + \text{噪音值}$$

2. **測試集:** 採用 4000 筆資料，測試集未加噪音，訓練集與測試集在同代中的隨機數設定不同，故產生不同兩組不同的資料。
3. **重複試驗:** 本試驗重複 20 次，每次重複時更換訓練集及測試集的隨機數設定。

4.2 模擬

4.2.1 Two Dimensional Michalewicz 函數



預測誤差 (SNR = 500)

由圖4.1可得，PE-Kmeans 的中位數最小且與其他演算法差距明顯；Supervised Compression 與 Regression Tree 的中位數相當接近。分佈上以 PE-Kmeans 最集中；Supervised Compression 最分散；Regression Tree 在兩者之間。

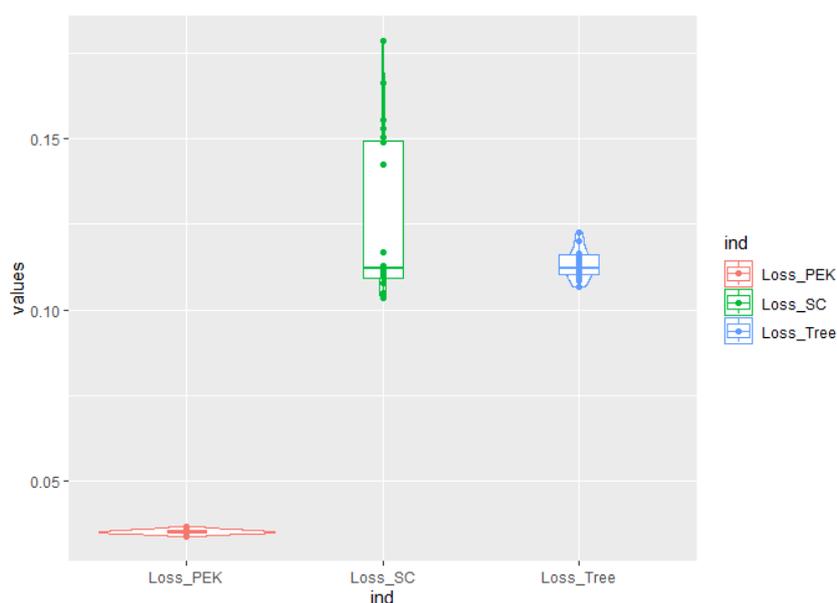


圖 4.1: Two Dimensional Michalewicz 之 RMSE (SNR = 500)

組內及組間變異分析 (SNR = 500)

組內變異：由圖4.2可得，PE-Kmeans 的中位數最小且與其他演算法差距明顯；Supervised Compression 與 Regression Tree 的中位數相當接近。分佈上以 PE-Kmeans 最集中；Supervised Compression 最分散且有相對極端的極端值；Regression Tree 在兩者之間。與預測誤差的型態大致相同。

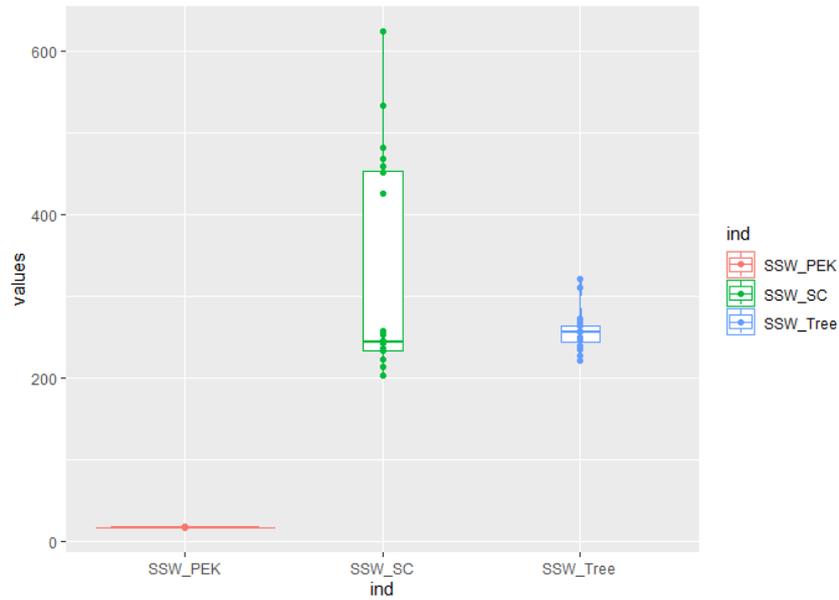


圖 4.2: Two Dimensional Michalewicz 之組內變異 (SNR = 500)

組間變異：由圖4.3可得，PE-Kmeans 的中位數最大且與其他演算法差距明顯；Supervised Compression 與 Regression Tree 的中位數相當接近。分佈上 PE-Kmeans 與 Regression Tree 較集中；Supervised Compression 最分散且有相對極端的極端值。

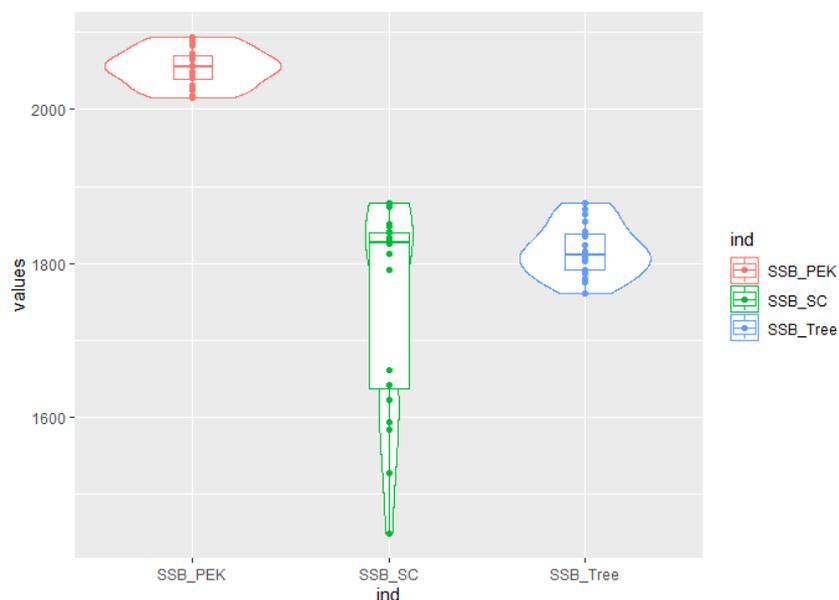


圖 4.3: Two Dimensional Michalewicz 之組間變異 (SNR = 500)



預測誤差 (SNR = 50)

由圖4.4可得，結果與 SNR = 500 時大致相同，但整體的誤差縮小。

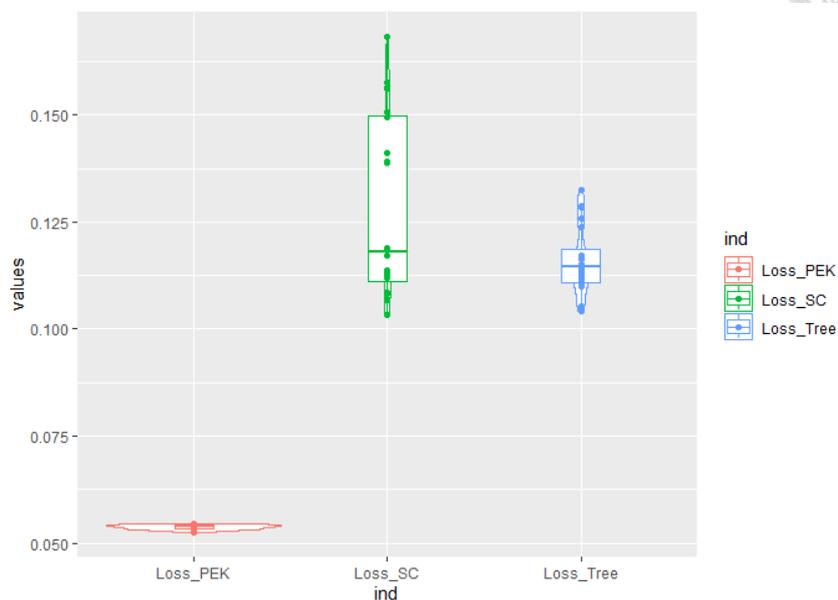


圖 4.4: Two Dimensional Michalewicz 之 RMSE (SNR = 50)

組內及組間變異分析 (SNR = 50)

組內變異：由圖4.5可得，結果與 SNR = 500 時大致相同，但整體變異稍微增加。

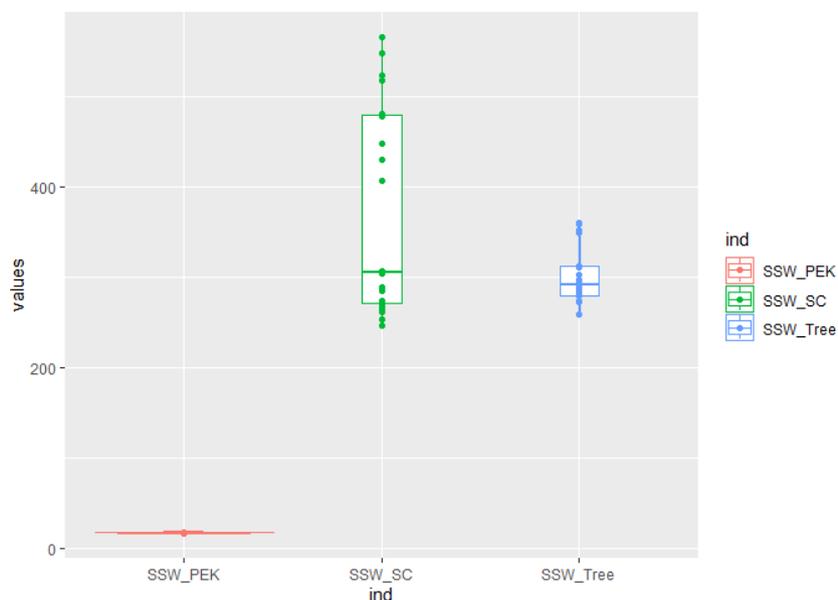


圖 4.5: Two Dimensional Michalewicz 之組內變異 (SNR = 50)

組間變異：由圖4.6可得，結果與 SNR = 500 時大致相同。

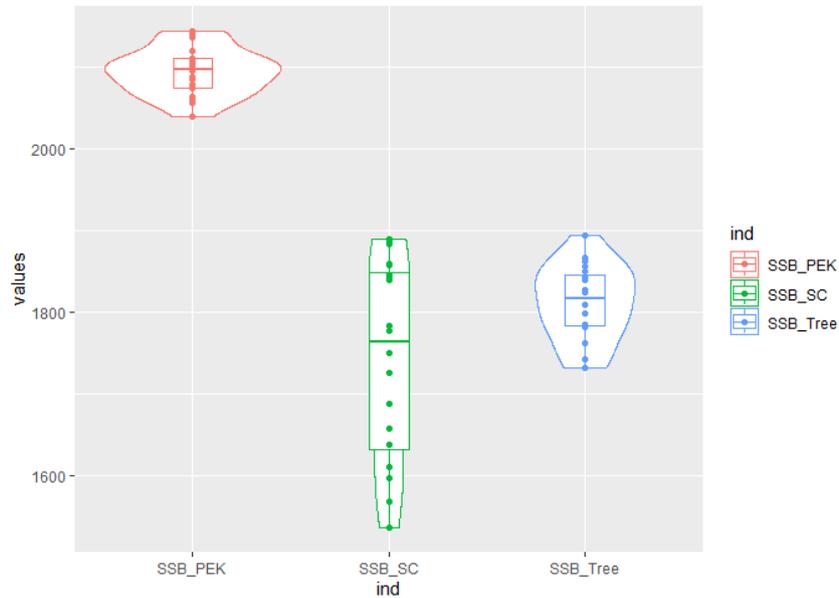


圖 4.6: Two Dimensional Michalewicz 之組間變異 (SNR = 50)

預測誤差 (SNR = 5)

由圖4.7可得，整體的誤差又再縮小，排名在此處發生逆轉，PE-Kmeans 的中位數最大，Supervised Compression 與 Regression Tree 的中位數相當接近。分布上，Supervised Compression 的長尾在此 SNR 底下更加明顯，Regression Tree 發生分散的現象。

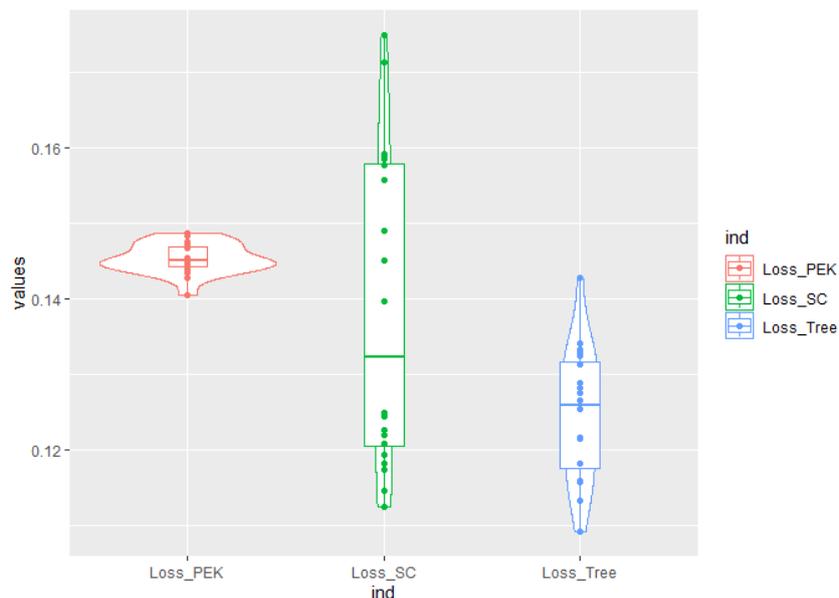


圖 4.7: Two Dimensional Michalewicz 之 RMSE (SNR = 5)



組內及組間變異分析 (SNR = 5)

組內變異：由圖4.8可得，結果順序與 SNR = 50 時相同，但除了 PE-Kmeans 外，其餘兩種演算法皆大幅增加組內變異。

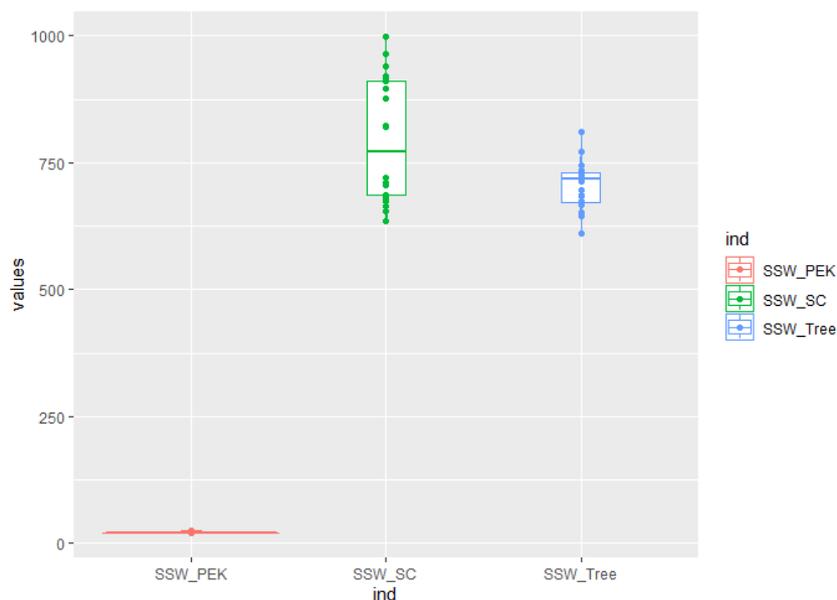


圖 4.8: Two Dimensional Michalewicz 之組內變異 (SNR = 5)

組間變異：由圖4.9可得，PE-Kmeans 大幅增加組間變異，其餘兩種演算法之結果與 SNR = 50 時大致相同。

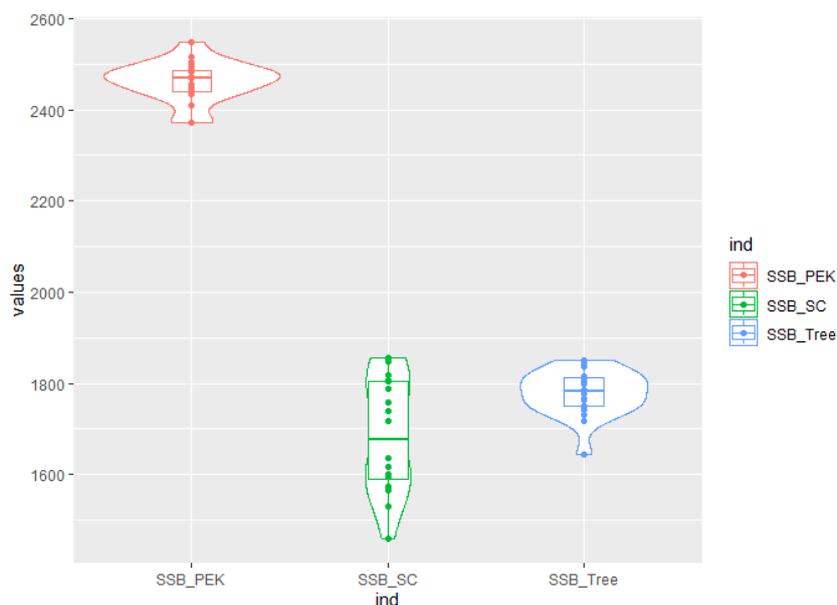


圖 4.9: Two Dimensional Michalewicz 之組間變異 (SNR = 5)



Two Dimensional Michalewicz 函數之總結

在預測誤差的分析中，SNR = 50 及 500 的情況下，PE-Kmeans 皆優於其餘兩種演算法，SNR = 5 的情況下，PE-Kmeans 劣於其餘兩種演算法；在組內及組間變異的分析中，PE-Kmeans 在組內皆為最低變異，組間皆為最高變異，其餘兩種演算法在組內及組間的變異程度相似，與預測誤差不同的是，這裡並未發生次序上的改變。

運行時間

由圖4.10可得，PE-Kmeans 花費在 35 秒左右；Supervised Compression 多在 18 秒左右但分佈相對分散，Regression Tree 多在 1 秒左右完成。

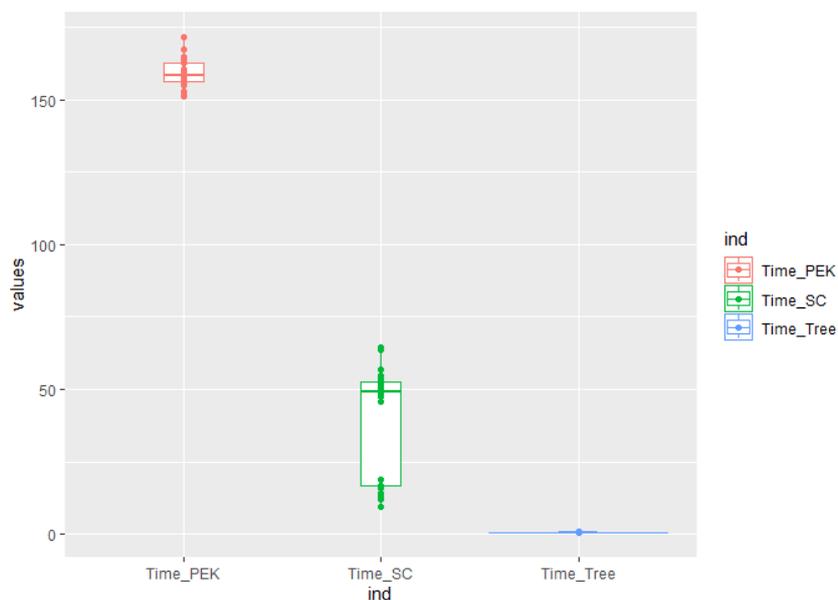


圖 4.10: Two Dimensional Michalewicz 之運行時間 (秒)

4.2.1.1 可視化

等高線圖

由圖4.11可得，Two Dimensional Michalewicz 的函數看起來方正，相對平滑。

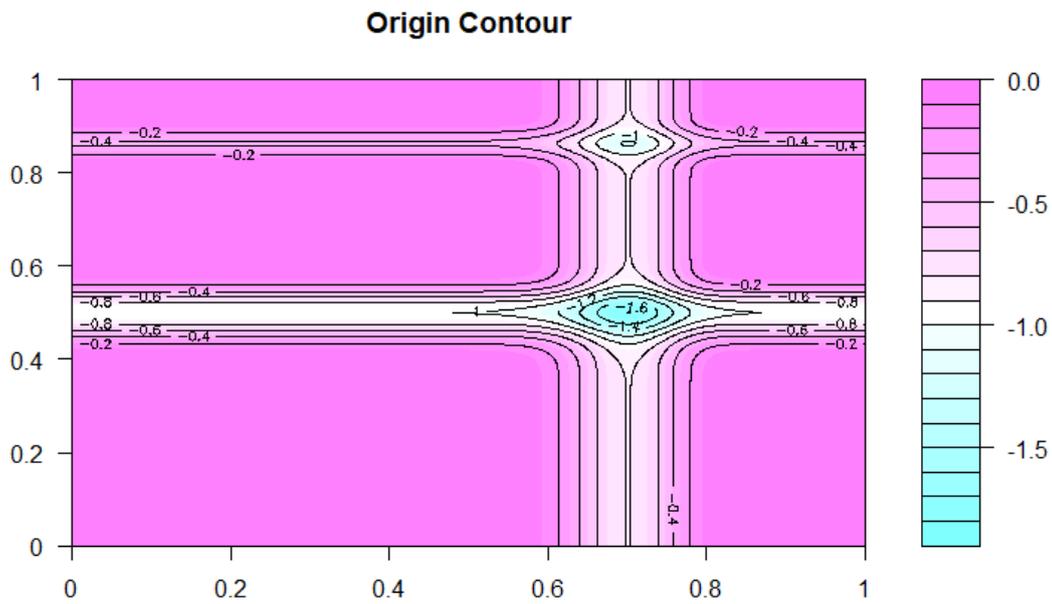


圖 4.11: 原始函數之等高線圖

由圖4.12可得，在 Regression Tree 模型下的等高線圖，在縱座標 0.8 到 1 之間的微低谷中心被省略了，整個水平線也未被模型捕捉。

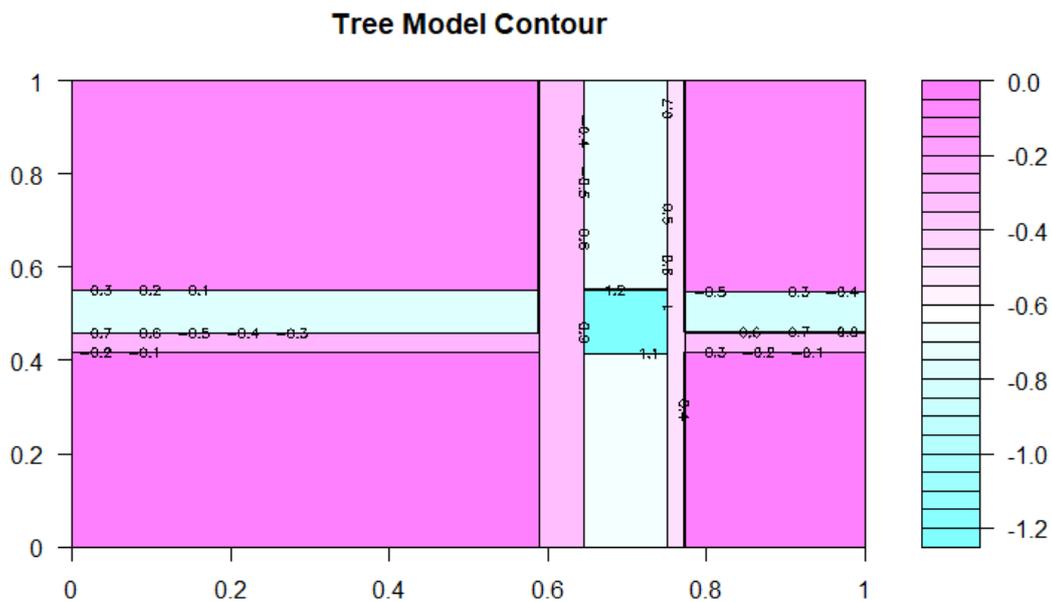


圖 4.12: Regression Tree 模型之等高線圖



由圖4.13可得，在 Supervised Compression 模型下的等高線圖，前項未被捕捉到的低谷區域大致有在這裡被捕捉到，也大致有兩條水平線。

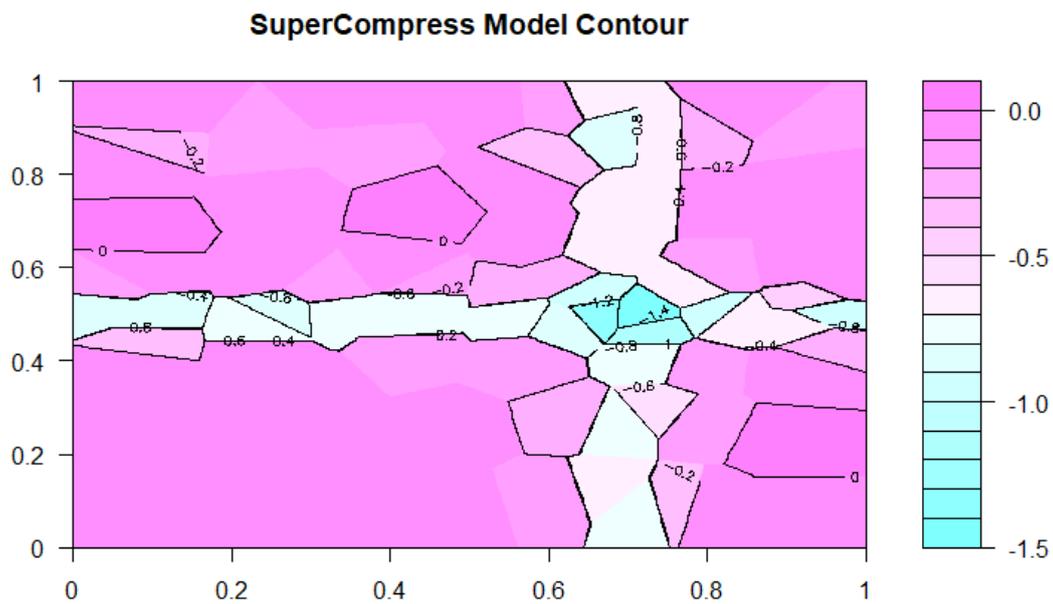


圖 4.13: Supervised Compression 模型之等高線圖

由圖4.14可得，在 PE-Kmeans 模型之下，可以看到已經非常接近原始函數的等高線圖。

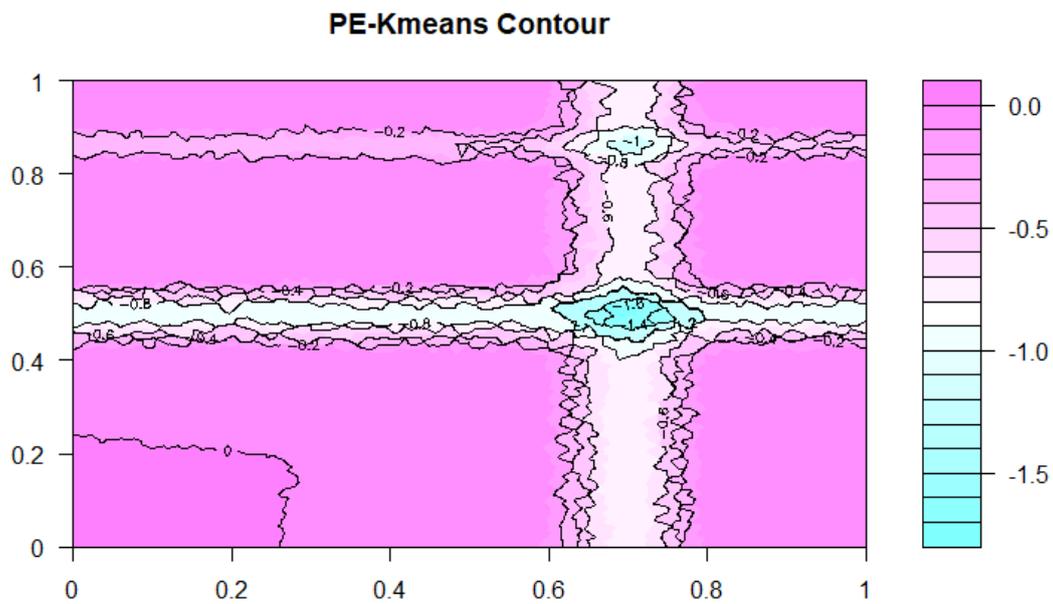


圖 4.14: PE-Kmeans 模型之等高線圖

分區圖



由圖4.15可得，Regression Tree 剪枝後共切了 13 塊長方形。

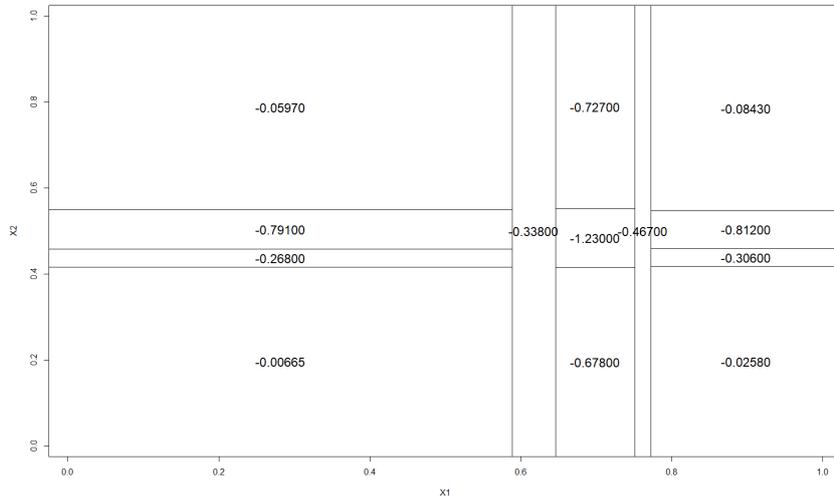


圖 4.15: Regression Tree 模型之分區圖

由圖4.16可得，Supervised Compression 切了 Voronoi region，圖上的點為區塊內的中心點，用來對應輸出值。

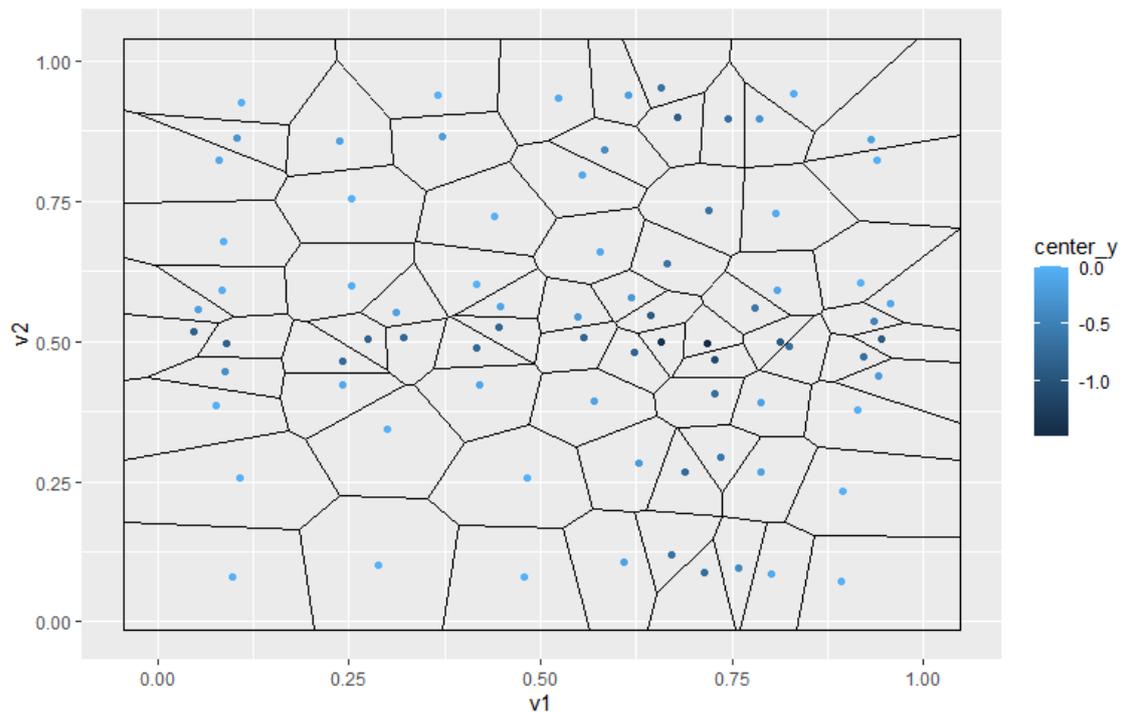


圖 4.16: Supervised Compression 模型之分區圖

由圖4.17可得，PE-Kmeans 以整個 Training dataset 作為分區中心點，故切割的非常密集。

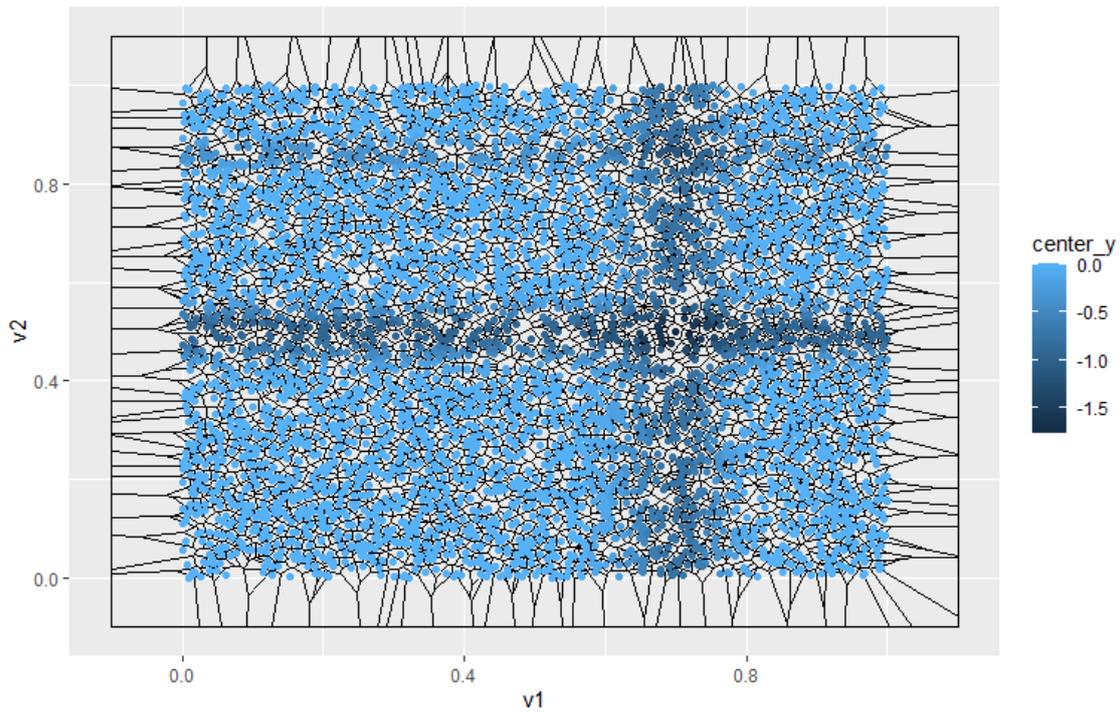


圖 4.17: PE-Kmeans 模型之分區圖

從上圖中無法看出實際的分組狀況，該分割僅代表所有落入本區的待預測資料點距離該分區中心點皆為最近的 Voronoi region，下圖4.18則呈現了 PE Kmeans 的分組邏輯，顏色代表著第一階段的分組，深淺則代表第二階段的分組。

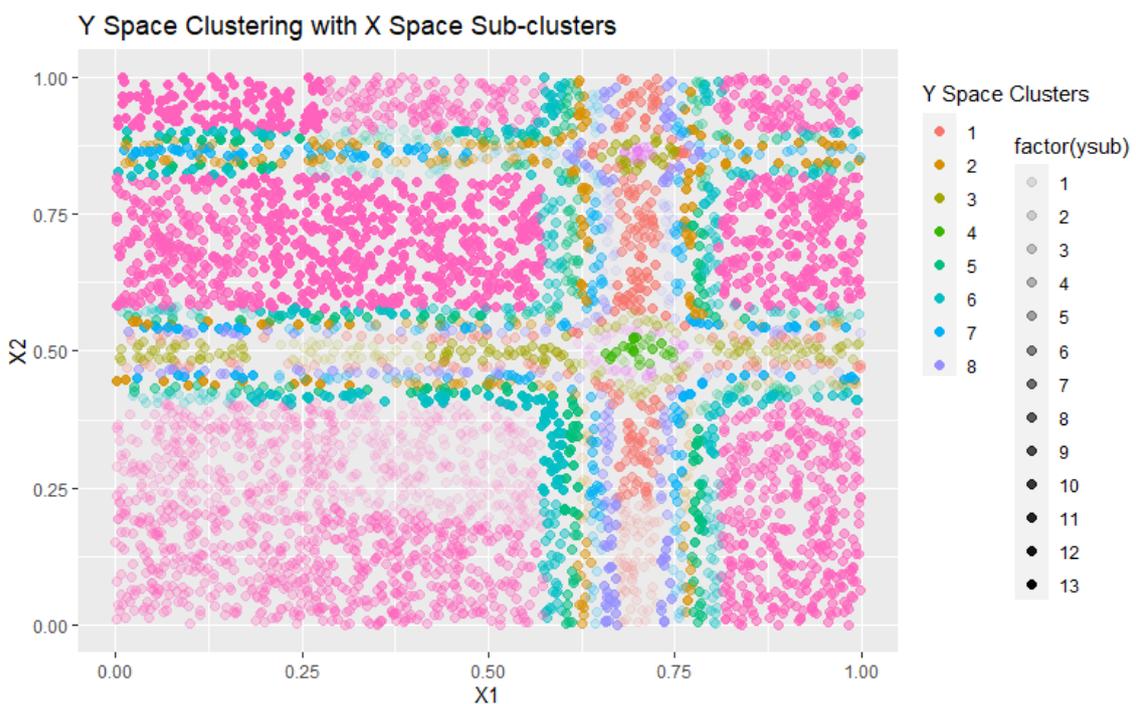


圖 4.18: PE-Kmeans 模型之分區圖 (二)



4.2.2 Dropwave 函數

預測誤差 (SNR = 500)

由圖4.19可得，PE-Kmeans 的中位數最小且與其他演算法差距明顯；Supervised Compression 與 Regression Tree 的中位數相當接近。三者分佈型態類似。

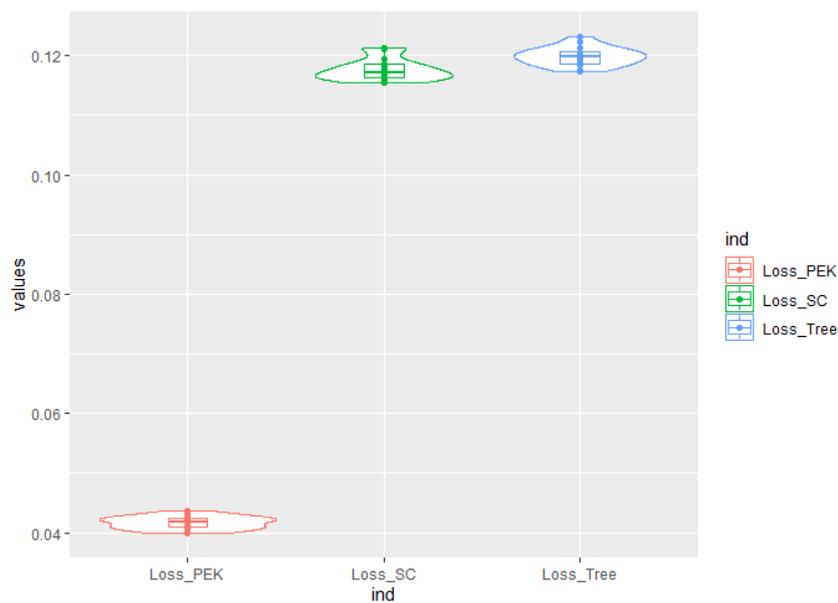


圖 4.19: Dropwave 之 RMSE (SNR = 500)

組內及組間變異分析 (SNR = 500)

組內變異：由圖4.20可得，PE-Kmeans 的中位數最小且非常接近 0，與其他演算法差距明顯；Supervised Compression 在兩者之間；Regression Tree 最大。分佈上以 PE-Kmeans 最集中；Supervised Compression 與 Regression Tree 型態類似。

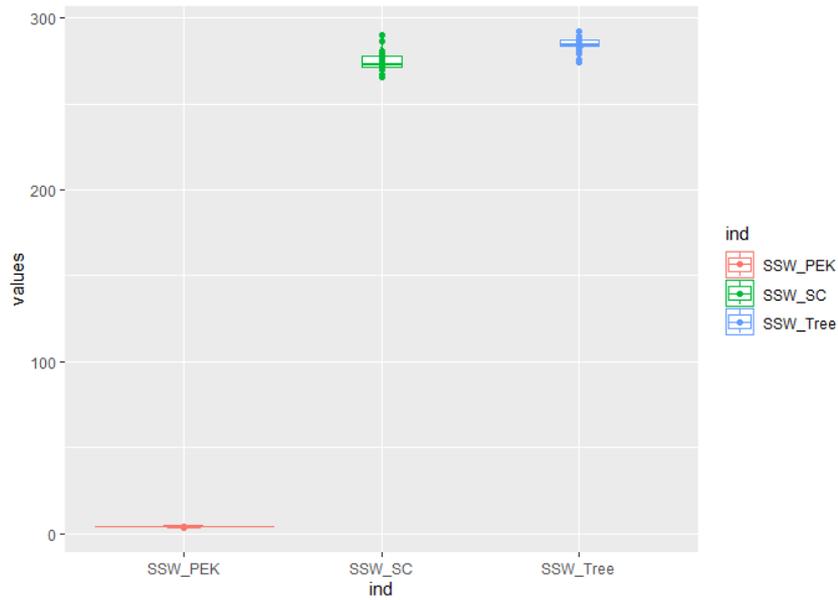


圖 4.20: Dropwave 之組內變異 (SNR = 500)

組間變異：由圖4.21可得，組間變異大小順序與組內變異之結果相反，三者分布差別不大。

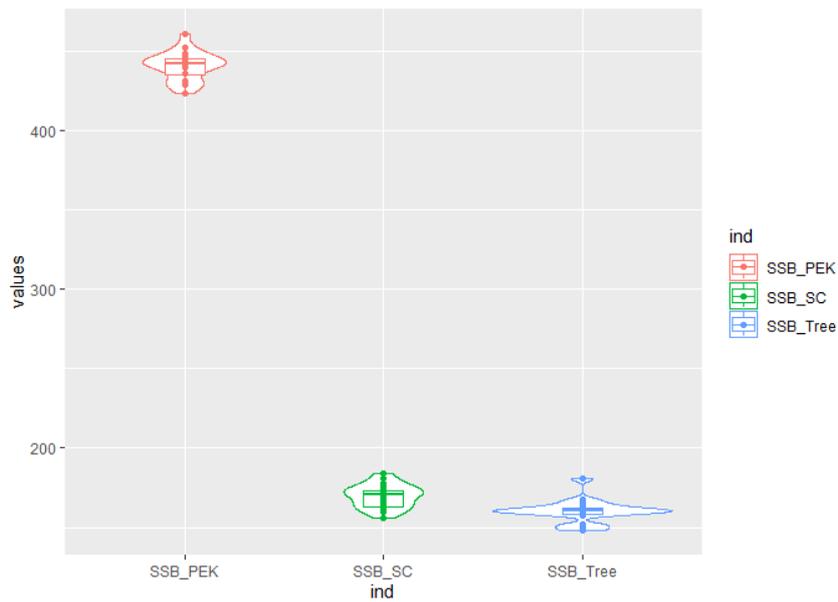


圖 4.21: Dropwave 之組間變異 (SNR = 500)

預測誤差 (SNR = 50)

由圖4.22可得，結果與 SNR = 500 時大致相同，除了 PK-Kmeans 稍有增加誤差，其餘幾乎沒有變化。

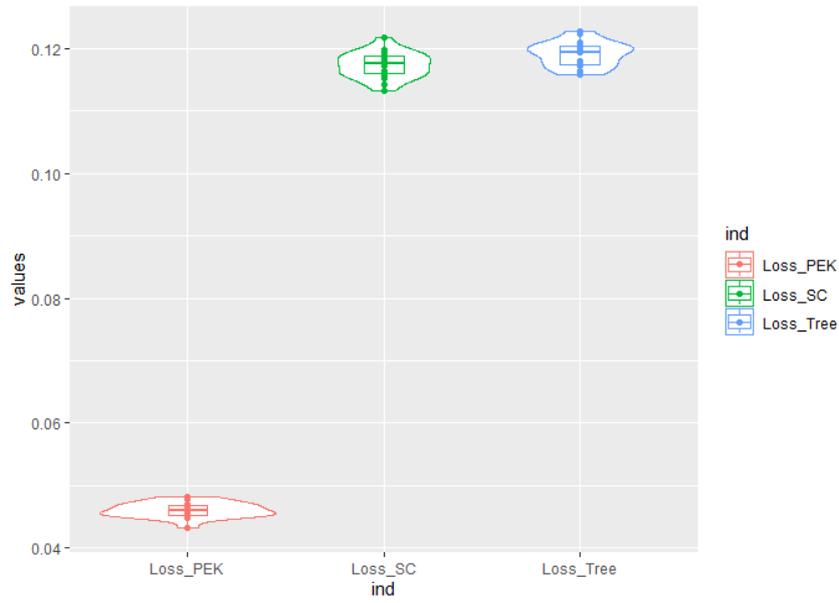


圖 4.22: Dropwave 之 RMSE (SNR = 50)

組內及組間變異分析 (SNR = 50)

組內變異：由圖4.23可得，結果與 SNR = 500 時大致相同。

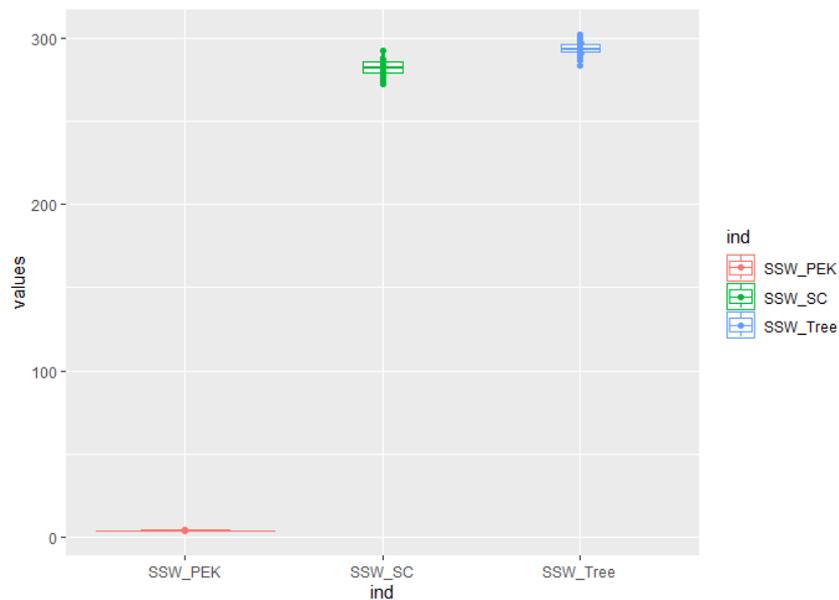


圖 4.23: Dropwave 之組內變異 (SNR = 50)

組間變異：由圖4.24可得，結果與 SNR = 500 時大致相同。

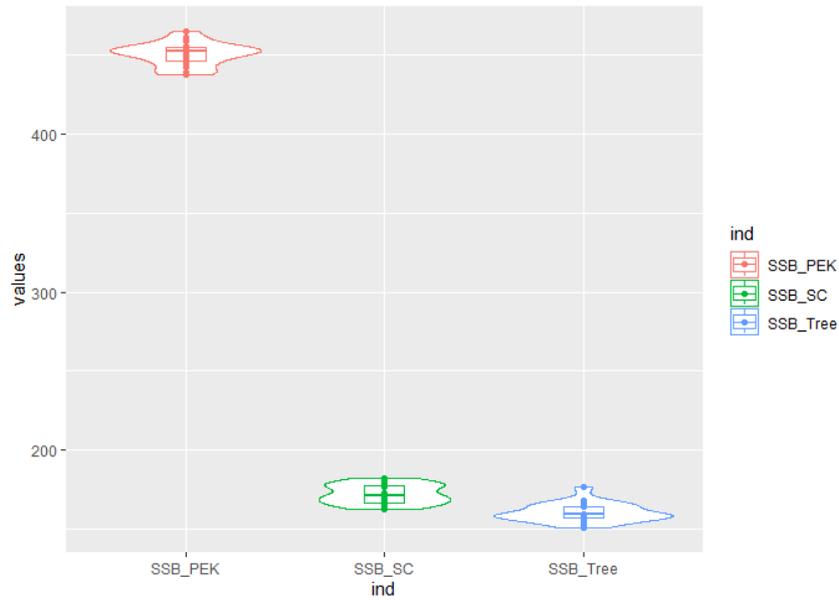


圖 4.24: Dropwave 之組間變異 (SNR = 50)

預測誤差 (SNR = 5)

由圖4.25可得，PK-Kmeans 誤差來到 0.07 至 0.08 之間，誤差增加較大，其餘兩種演算法幾乎沒有變化。

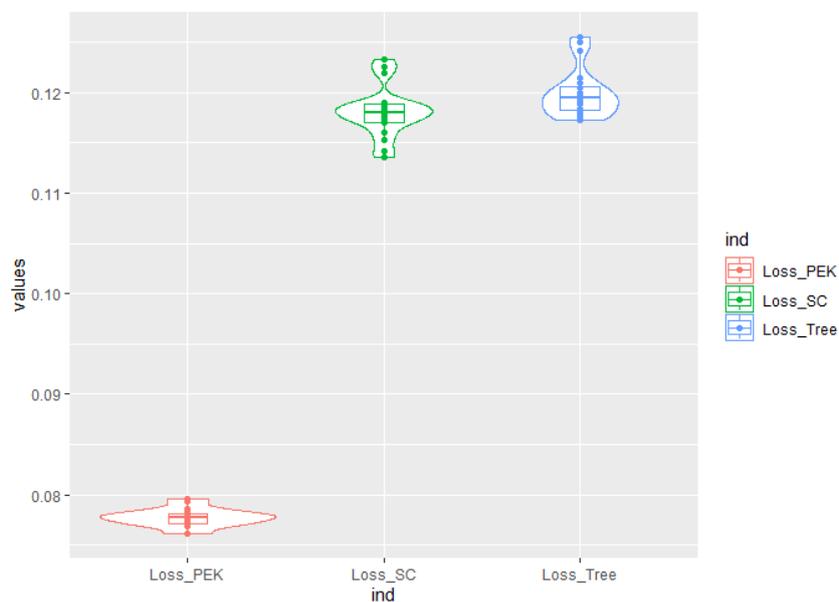


圖 4.25: Dropwave 之 RMSE (SNR = 5)



組內及組間變異分析 (SNR = 5)

組內變異由圖4.26可得，與 SNR = 50 時相比，除了 PE-Kmeans 之外，其餘兩種演算法有提高。

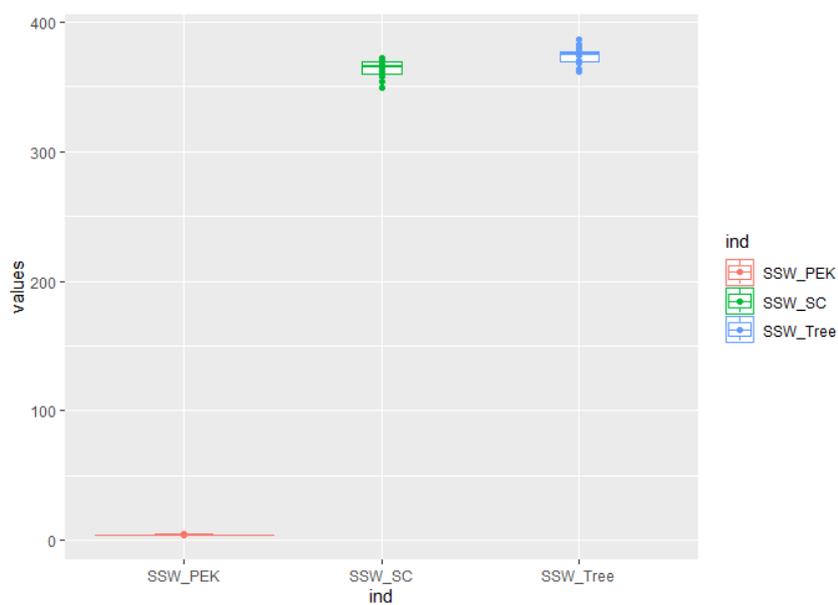


圖 4.26: Dropwave 之組內變異 (SNR = 5)

組間變異由圖4.27可得，與 SNR = 50 時大致相同，但 PE-Kmeans 有提高。

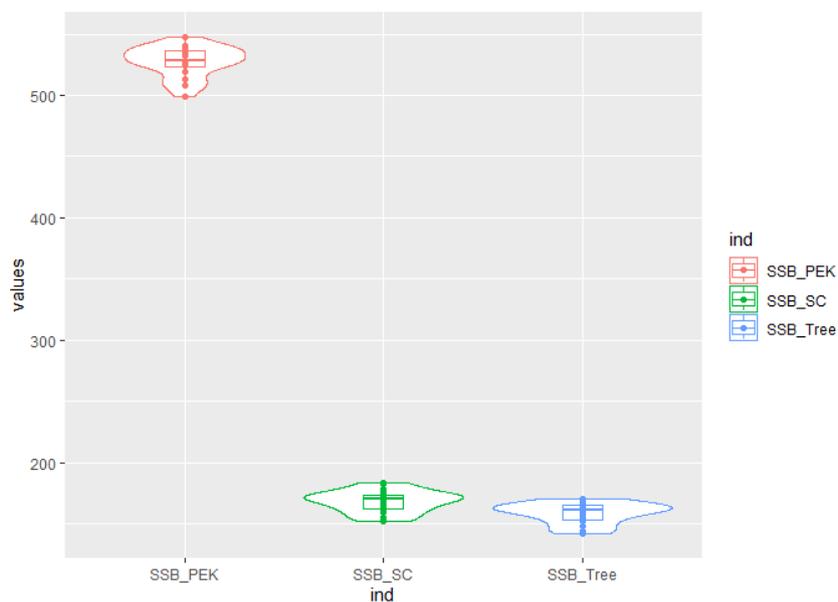


圖 4.27: Dropwave 之組間變異 (SNR = 5)



Dropwave 函數之總結

在預測誤差的分析中，任何 SNR 值的情況下，PE-Kmeans 皆優於其餘兩種演算法；在組內及組間變異的分析中，PE-Kmeans 在組內皆為最低變異，組間皆為最高變異，其餘兩種演算法在組內及組間的變異程度相似。

運行時間

由圖4.28可得，PE-Kmeans 花費在 300 秒左右，相對分散；Supervised Compression 在 20 秒左右，Regression Tree 多在 1 秒左右完成。

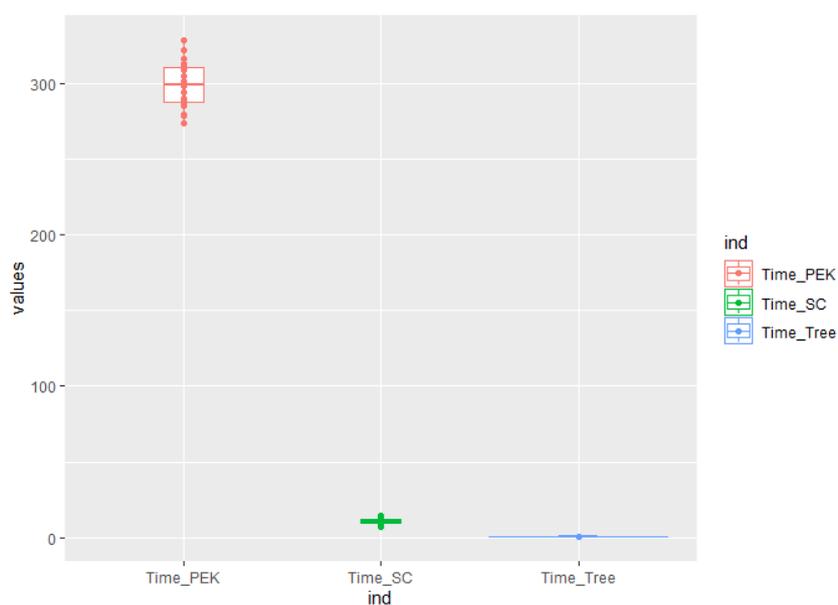


圖 4.28: Dropwave 之運行時間 (秒)

4.2.2.1 可視化

等高線圖

由圖4.29可得，此函數像水滴落到水池中產生的波紋。

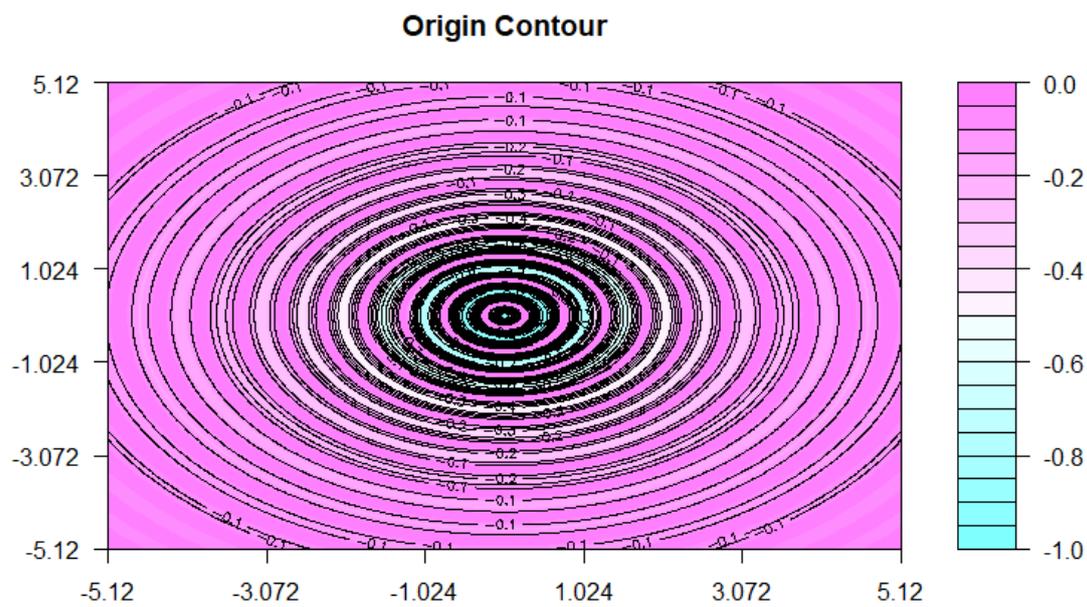


圖 4.29: 原始函數之等高線圖

由圖4.30可得，Regression Tree 大致抓到形狀，但方正的切割難以還原波紋。

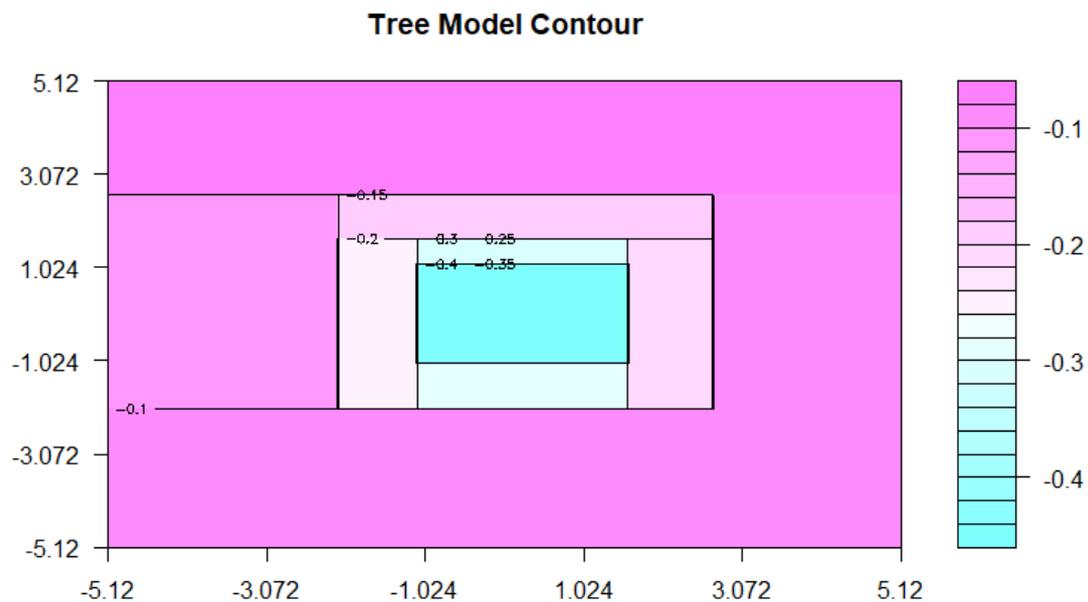


圖 4.30: Regression Tree 模型之等高線圖

由圖4.31可得，相對 Regression Tree 較有捕捉圓形及層次性的效果。

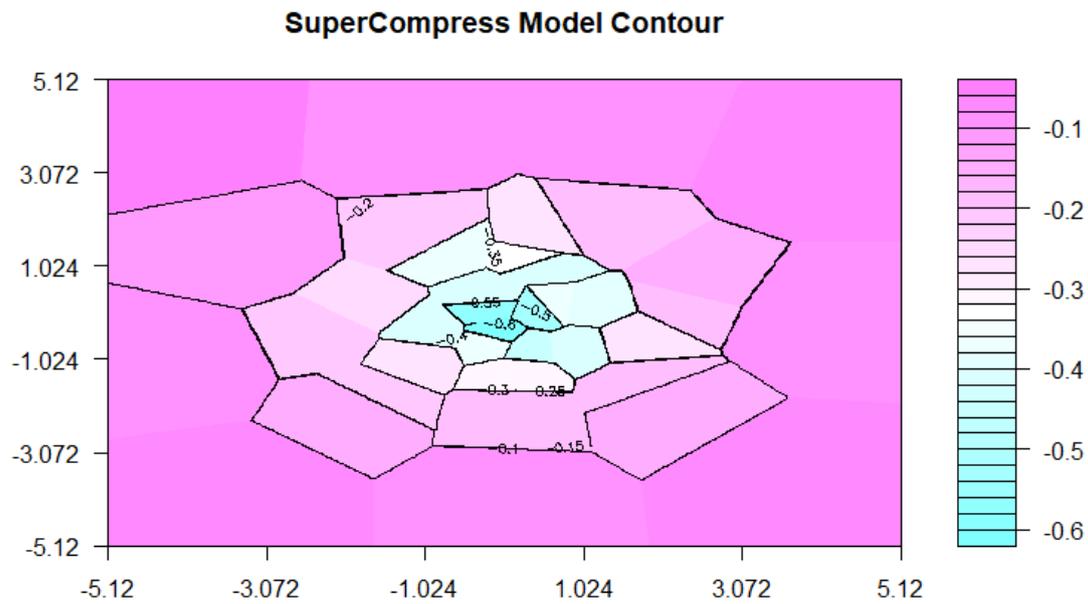


圖 4.31: Supervised Compression 模型之等高線圖

由圖4.32可得，PE-Kmeans 幾乎完美還原了波型。

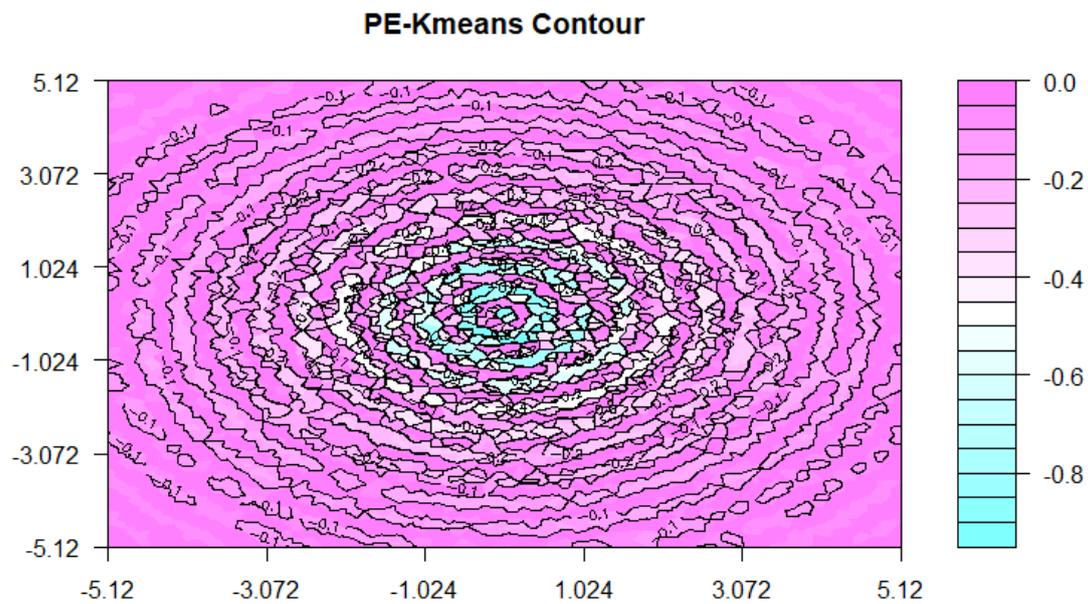


圖 4.32: PE-Kmeans 模型之等高線圖

分區圖



由圖4.33可得，Regression Tree 分了十個長方形區塊。

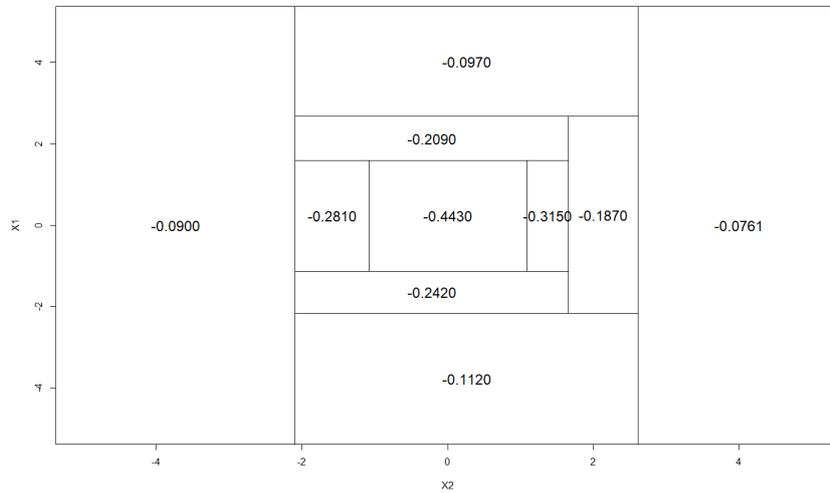


圖 4.33: Regression Tree 模型之分區圖

由圖4.34可得，Supervised Compression 的分區相對 Regression Tree 較多。

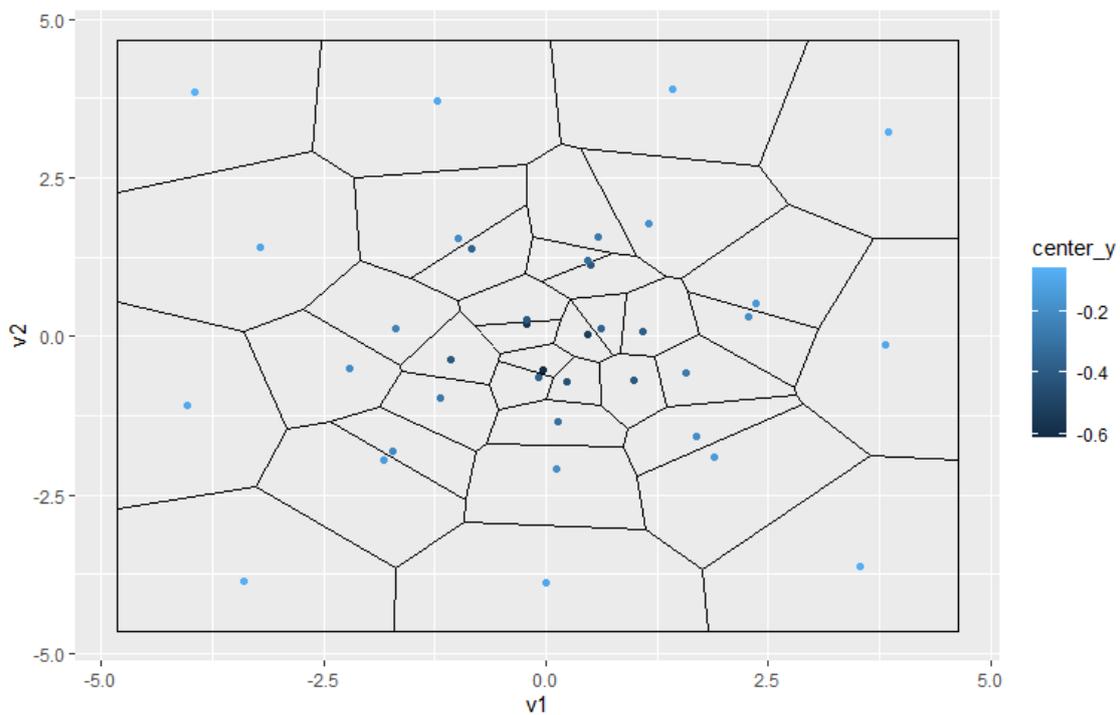


圖 4.34: Supervised Compression 模型之分區圖

由圖4.35可得，以整個訓練集資料點做切割，還原了波型的凹凸變化。

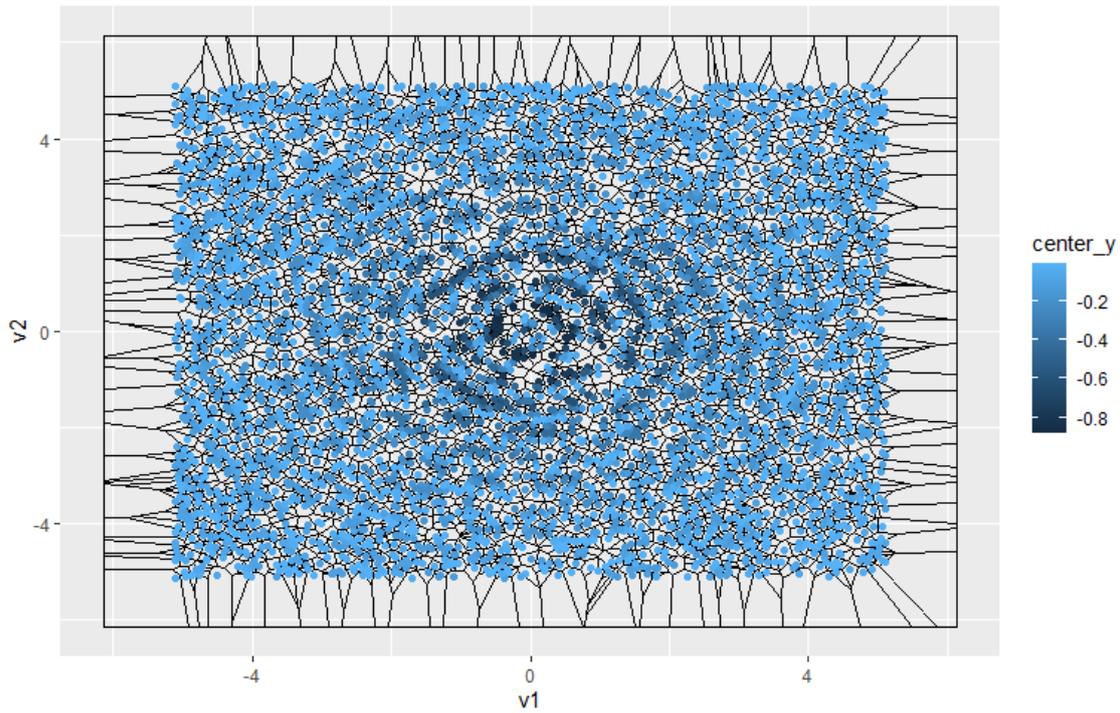


圖 4.35: PE-Kmeans 模型之分區圖

上圖分割僅代表所有落入本區的待預測資料點距離該分區中心點皆為最近的 Voronoi region，圖4.36則呈現了分組邏輯，顏色代表著第一階段的分組，深淺代表第二階段的分組；由顏色可看到一圈圈往外延伸的同心圓，分群的效果很好。

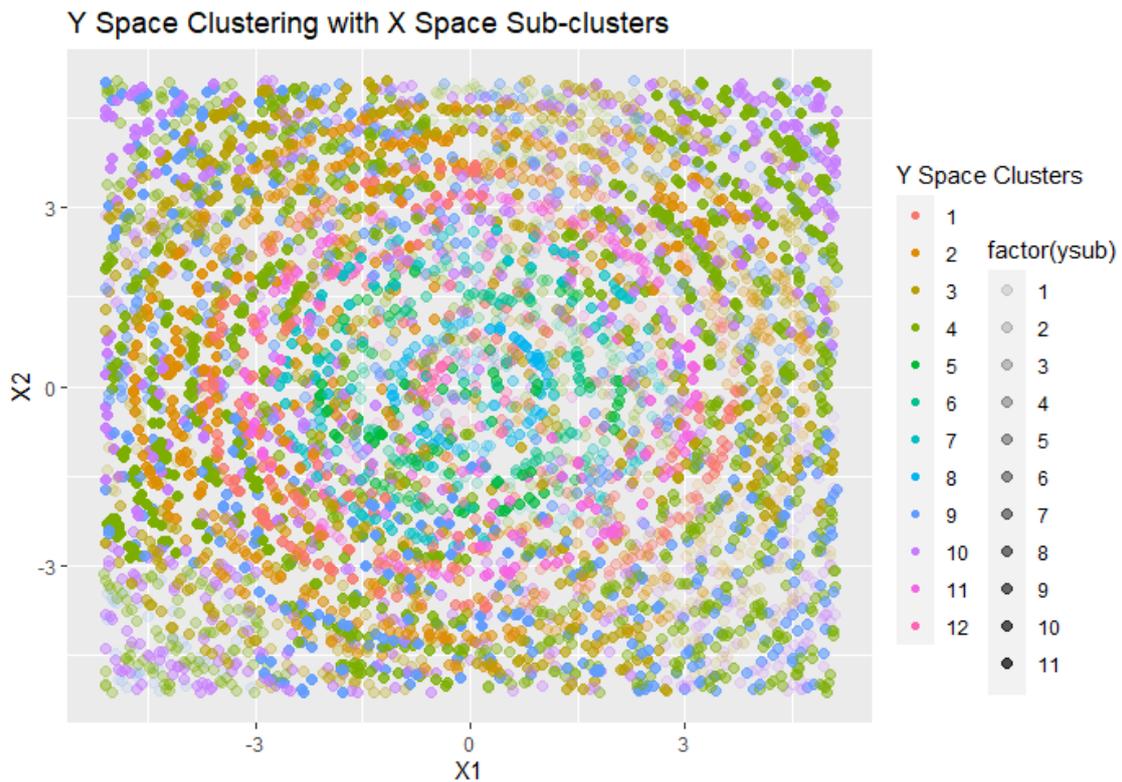


圖 4.36: PE-Kmeans 模型之分區圖 (二)



4.2.3 OTL Circuit 函數

預測誤差 (SNR = 500)

由圖4.37可得，PE-Kmeans 的中位數最小，些微贏過 Regression Tree；Supervised Compression 表現最差。以分布來說，Supervised Compression 變異程度相對大很多。

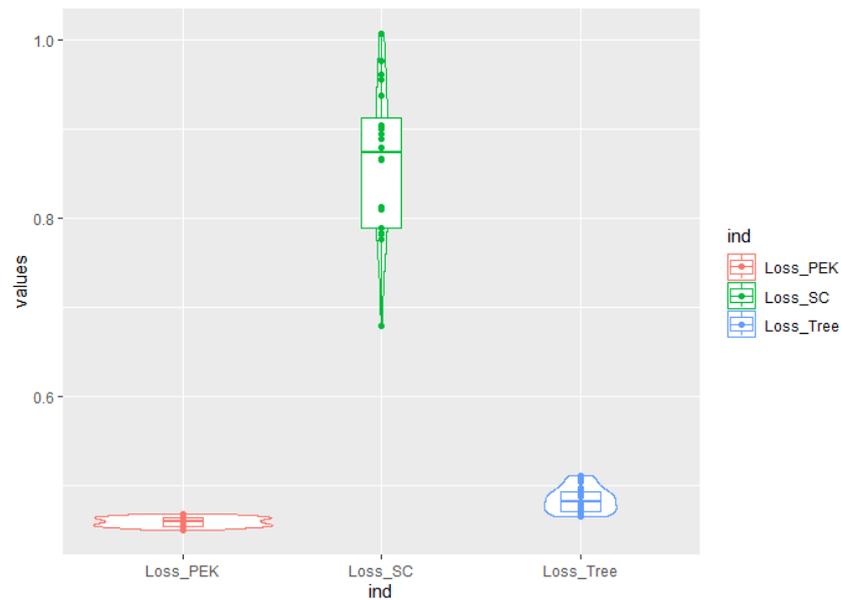


圖 4.37: OTL Circuit 之 RMSE (SNR = 500)

組內及組間變異分析 (SNR = 500)

組內變異：由圖4.38可得，PE-Kmeans 的中位數最小且非常接近 0，與其他演算法差距明顯；Supervised Compression 與 Regression Tree 相當接近。分佈上以 PE-Kmeans 最集中；Supervised Compression 與 Regression Tree 型態類似。

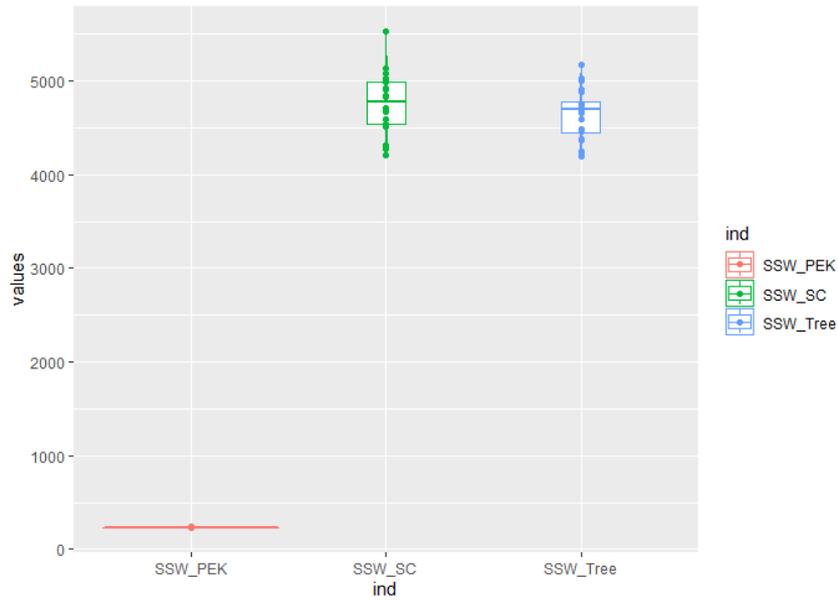


圖 4.38: OTL Circuit 之組內變異 (SNR = 500)

組間變異：由圖4.39可得，組間變異大小順序與組內變異之結果相反，三者分布差別不大。

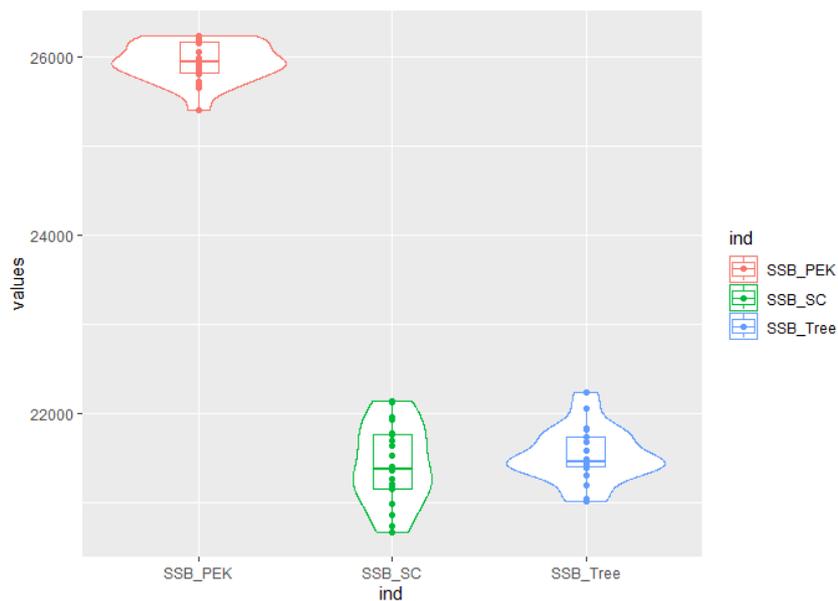


圖 4.39: OTL Circuit 之組間變異 (SNR = 500)

預測誤差 (SNR = 50)

由圖4.40可得，結果與 SNR = 500 時區別不大。

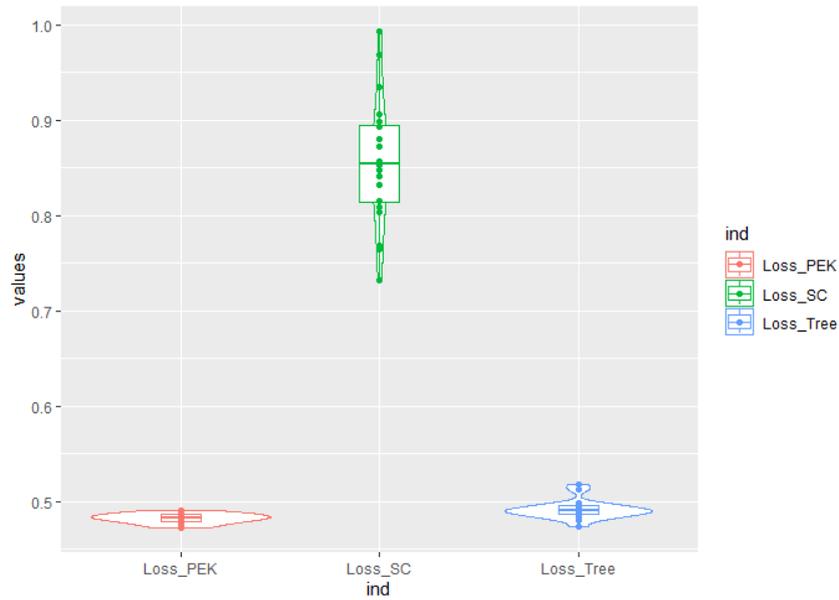


圖 4.40: OTL Circuit 之 RMSE (SNR = 50)

組內及組間變異分析 (SNR = 50)

組內變異：由圖4.41可得，除了 PE-Kmeans，其餘兩種演算法的組內變異提高。

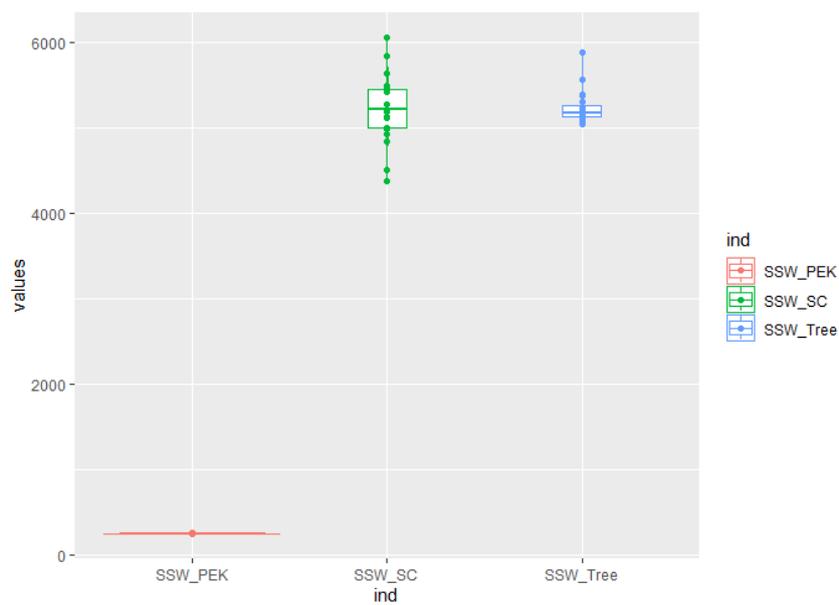


圖 4.41: OTL Circuit 之組內變異 (SNR = 50)

組間變異：由圖4.42可得，PE-Kmeans 的組間變異提高較明顯。

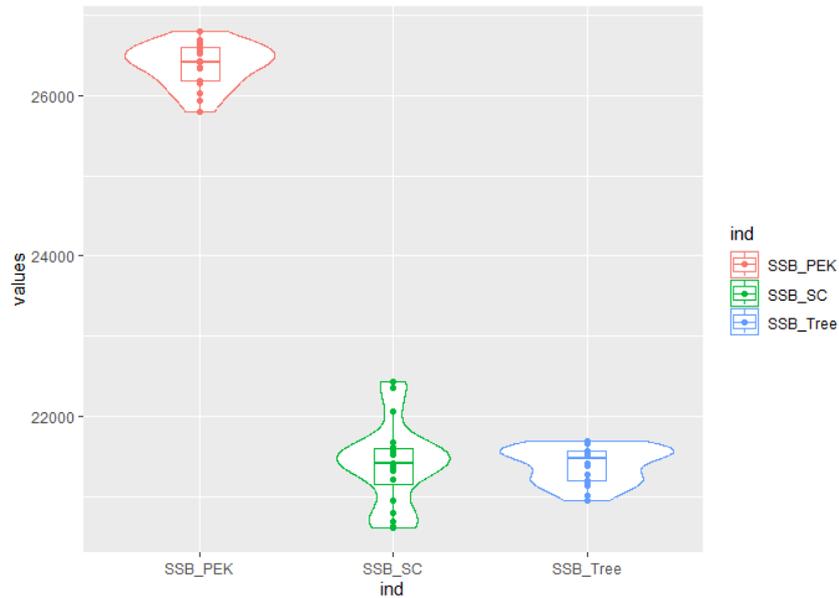


圖 4.42: OTL Circuit 之組間變異 (SNR = 50)

預測誤差 (SNR = 5)

由圖4.43可得，PE-Kmeans 誤差明顯提高，以至於輸給 Regression Tree，其餘兩種演算法變化不大。

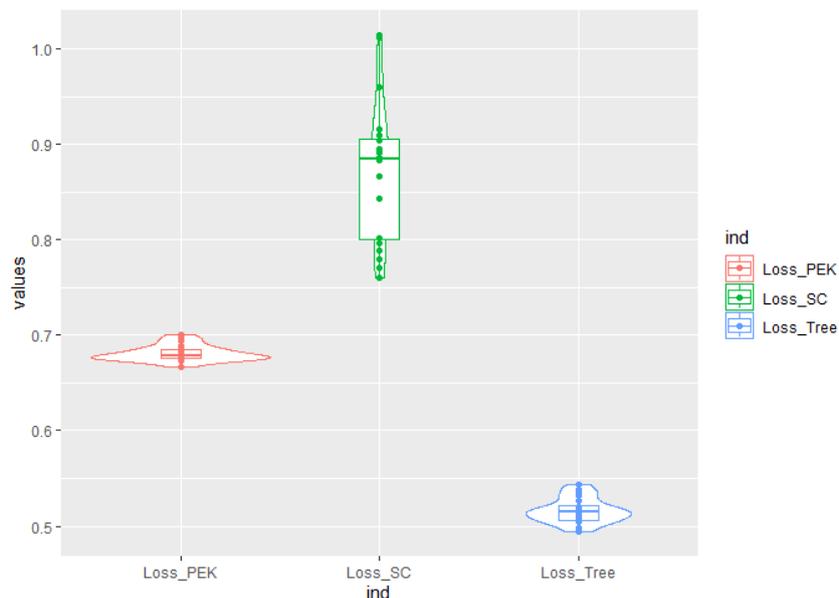


圖 4.43: OTL Circuit 之 RMSE (SNR = 5)



組內及組間變異分析 (SNR = 5)

組內變異由圖4.44可得，除了 PE-Kmeans，其餘兩種演算法的組內變異有大幅度的提升。

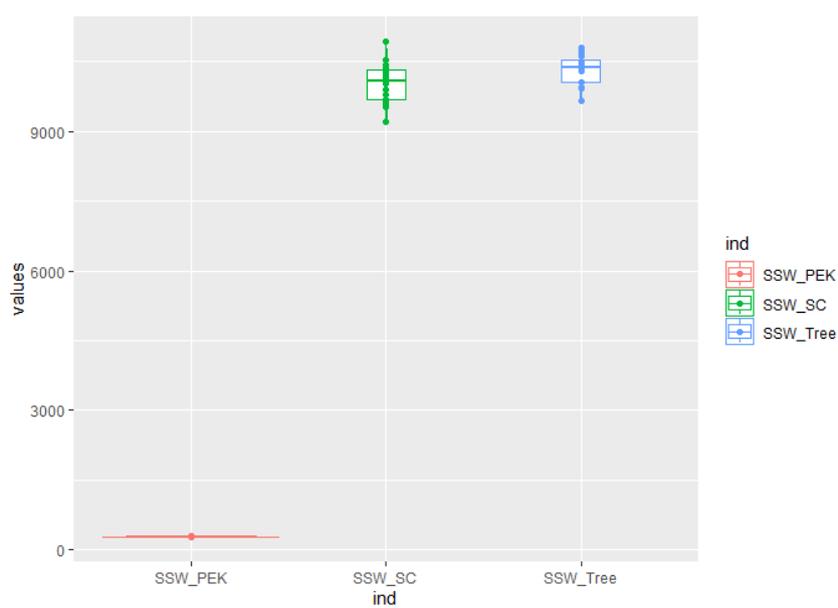


圖 4.44: OTL Circuit 之組內變異 (SNR = 5)

組間變異由圖4.45可得，PE-Kmeans 的組間變異大幅度提升。

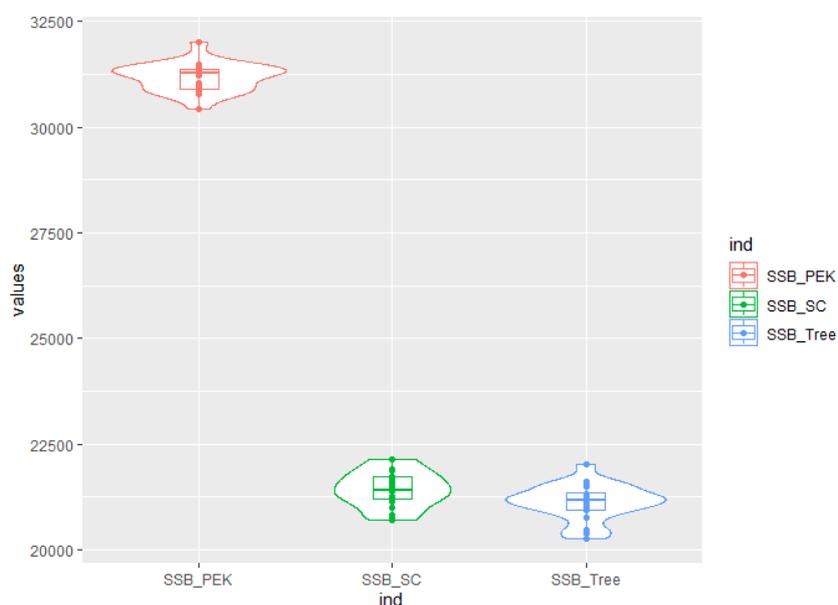


圖 4.45: OTL Circuit 之組內變異 (SNR = 5)



OTL Circuit 函數之總結

在預測誤差的分析中，在 SNR=500 及 50 的情況下，PE-Kmeans 優於 Regression Tree，但在 SNR=5 的情況下 Regression Tree 較優，三組噪音下 Supervised Compression 表現皆為最差；在組內及組間變異的分析中，PE-Kmeans 在組內皆為最低變異，組間皆為最高變異，其餘兩種演算法在組內及組間的變異程度相似。

運行時間

由圖4.46可得，PE-Kmeans 在運行時間上明顯高於其餘兩種演算法。

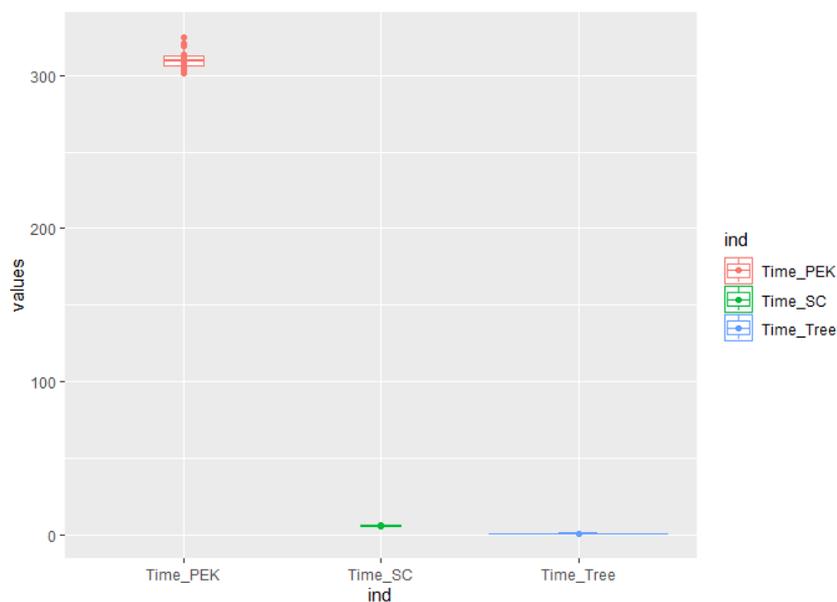


圖 4.46: OTL Circuit 之運行時間 (秒)

4.2.4 Piston 函數

預測誤差

由圖4.47可得，PE-Kmeans 表現最差；Supervised Compression 次之 Regression Tree 表現大幅領先。以分佈來說，Supervised Compression 變異程度相對較大。

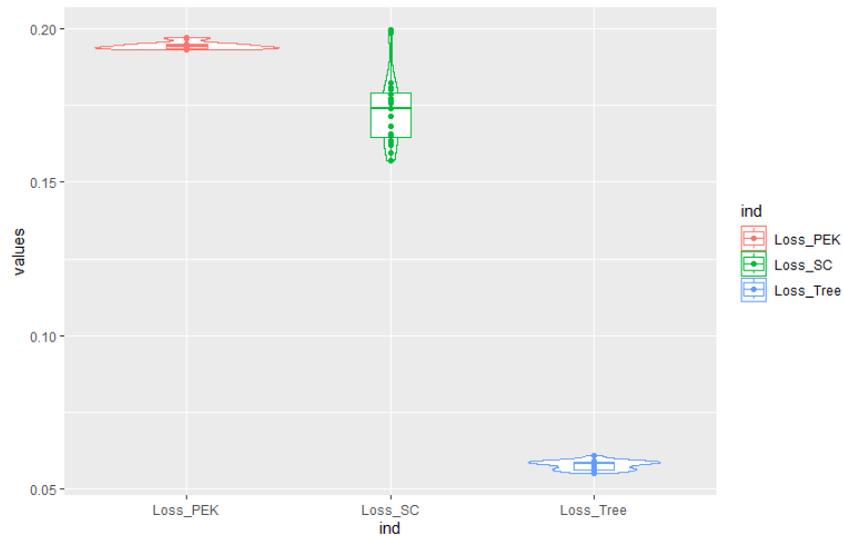


圖 4.47: Piston 之 RMSE (SNR = 500)

組內及組間變異分析

組內變異：由圖4.48可得，PE-Kmeans 的中位數最小且非常接近 0，與其他演算法差距明顯；Supervised Compression 最大；Regression Tree 在兩者之間；分佈上以 PE-Kmeans 最集中；Supervised Compression 與 Regression Tree 類似。

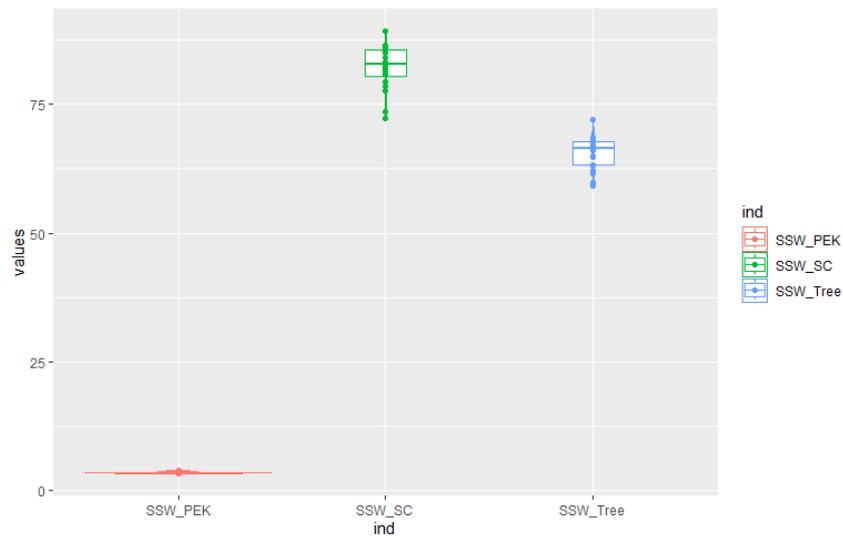


圖 4.48: Piston 之組內變異 (SNR = 500)

組間變異：由圖4.49可得，PE-Kmeans 的中位數最大，與其他演算法差距明顯；Supervised Compression 最小；Regression Tree 在兩者之間；分佈上三者型態大致相同。

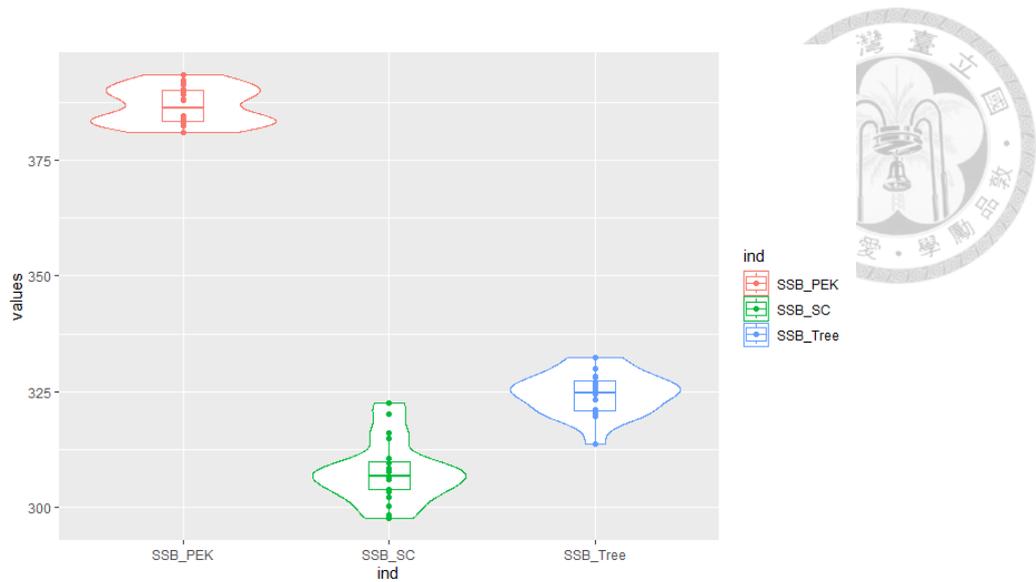


圖 4.49: Piston 之組間變異 (SNR = 500)

Piston 函數之總結

此函數相對複雜，故僅使用較少噪音 (SNR = 500) 的資料集，即使如此 PE-Kmeans 仍占不到優勢，預測誤差的結果顯示 Regression Tree 大幅優於其他兩種演算法；但在組內及組間變異的分析中，PE-Kmeans 仍在組內為最低變異，組間為最高變異。

運行時間

由圖4.50可得，PE-Kmeans、Supervised Compression 的運行時間離散程度很大，Tree 則很小且快速。

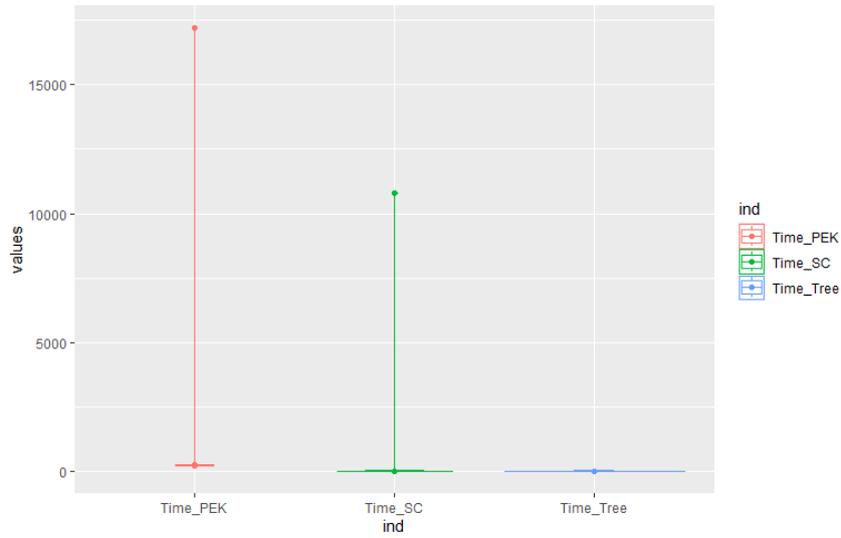


圖 4.50: Piston 之運行時間 (秒)

4.2.5 Borehole 函數

預測誤差

由圖4.51可得，PE-Kmeans 的中位數最大；Supervised Compression 次之，Regression Tree 最小且大幅領先。分佈上 Supervised Compression 變異程度相對較大。

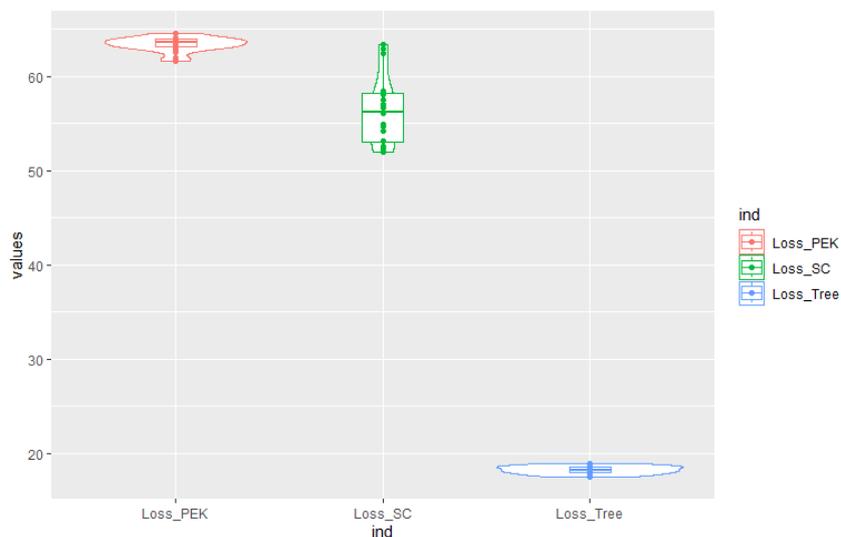


圖 4.51: Borehole 之 RMSE (SNR = 500)



組內及組間變異分析

組內變異：由圖4.52可得，PE-Kmeans 的中位數最小且非常接近 0，與其他演算法差距明顯；Supervised Compression 最大；Regression Tree 在兩者之間。分佈上以 PE-Kmeans 最集中；Supervised Compression 與 Regression Tree 類似。

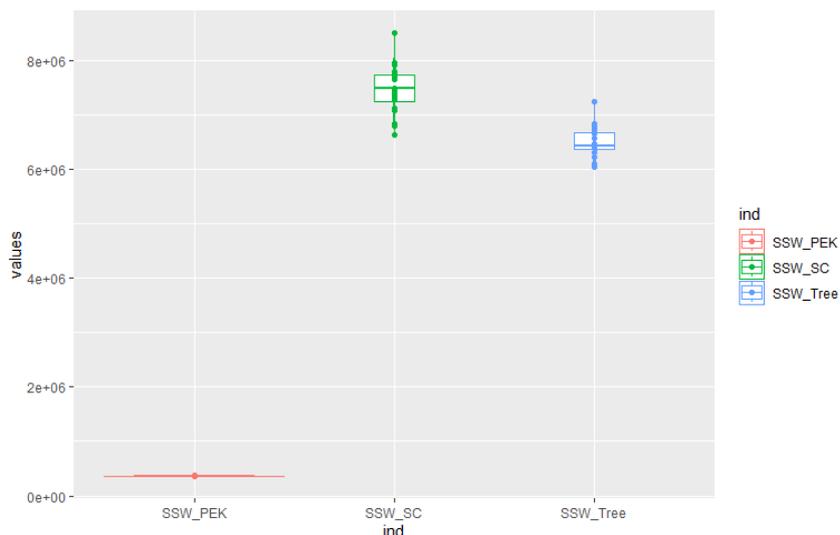


圖 4.52: Borehole 之組內變異 (SNR = 500)

組間變異：由圖4.53可得，PE-Kmeans 的中位數最大，與其他演算法差距明顯；Supervised Compression 最小；Regression Tree 在兩者之間。分佈上三者型態大致相同。

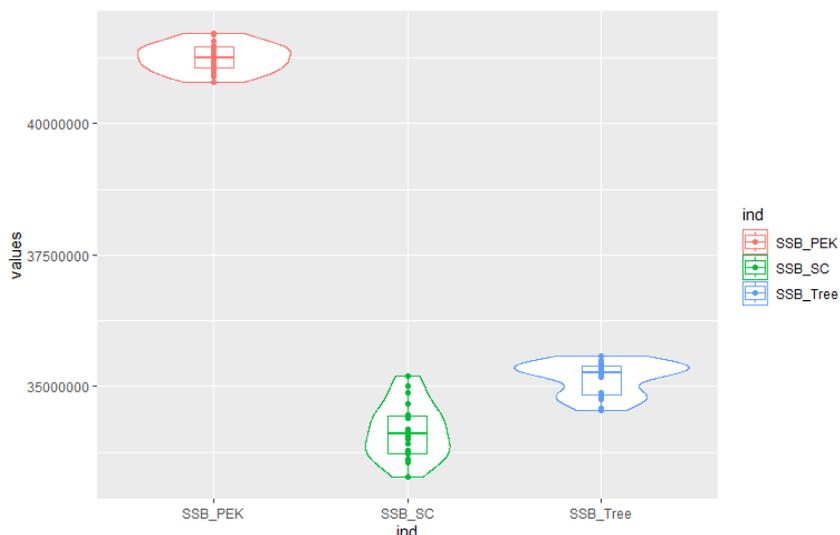


圖 4.53: Borehole 之組間變異 (SNR = 500)



Borehole 函數之總結

此函數相對複雜，故僅使用較少噪音 (SNR = 500) 的資料集，即使如此 PE-Kmeans 仍占不到優勢，預測誤差的結果顯示 Regression Tree 大幅優於其他兩種演算法；但在組內及組間變異的分析中，PE-Kmeans 仍在組內為最低變異，組間為最高變異，以上各種結果與 Piston 函數相近。

運行時間

由圖4.54可得，PE-Kmeans 的運行時間明顯高於其餘兩種演算法。

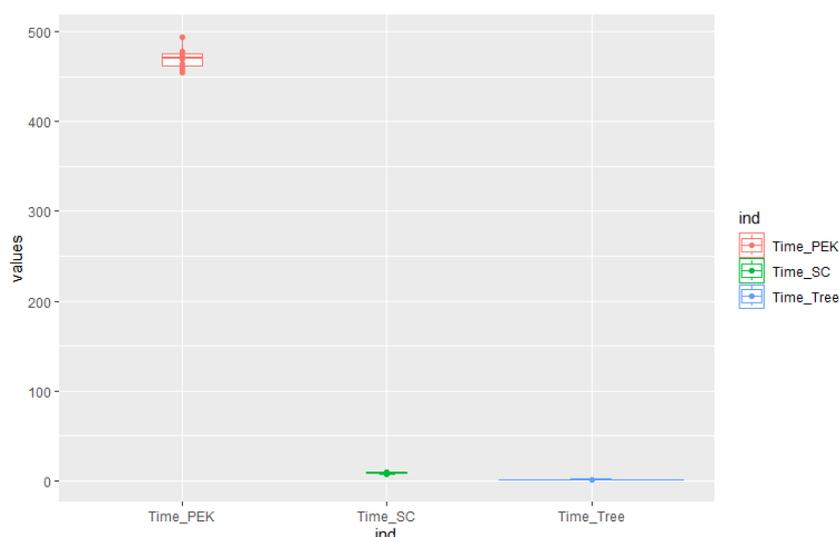


圖 4.54: Borehole 之運行時間 (秒)

4.2.6 函數模擬實驗總結

經本研究發現，PE-Kmeans 演算法在一些情況下具有優勢，從觀察 Two Dimensional Michalewicz 與 Dropwave 函數兩者的模擬，可以發現 PE-Kmeans 在 Dropwave 函數上的表現相對較優，既使是在噪音高的情況下的表現亦大幅優於其他兩種演算法，但在 Two Dimensional Michalewicz 函數上且噪音高的情況下則不敵 Regression Tree；而 Two Dimensional Michalewicz 與 Dropwave 函數最大的區別就是，後者的輸出變量在輸入空間上是呈同心圓狀的劇烈浮動，前者則相對更具有方正的大幅平坦區域，可以說在複雜度（或者說平滑程度）上，兩函數雖同為二維，後者卻相對複雜（不平滑）很多，意即 PE-Kmeans 在維度相同的函數之

下，對於複雜的函數上相對其他兩種演算法表現優異。

當使用六維函數 OTL Circuit 做模擬時，研究顯示在資料噪音較少時，PE-Kmeans 仍勝過其他兩種演算法，但當使用七維函數 Piston 及八維函數 Borehole 時，PE-Kmeans 均未能勝過其餘兩種演算法，且嘗試在常規實驗外額外再大幅減少噪音也無法改善，推測需要大幅增加資料量以適應函數維度的增加。





Chapter 5 真實資料分析

本研究利用 MARTHE 資料集，該資料集為法國地質調查局開發的 MARTHE 程式模擬的真實實現，用於模擬俄羅斯莫斯科 Kurchatov 研究所上部含水層中 鋇-90 的運輸情況，有助於深入了解地下水系統中 鋇-90 運輸的特性及其影響因素，這個資料集可用於測試誤差傳播和全局敏感度分析的方法。

表 5.1: Parameter Ranges

Parameter	Distribution	Range	Description
<i>per1</i>	Uniform	[1, 15]	Hydraulic conductivity layer 1
<i>per2</i>	Uniform	[5, 20]	Hydraulic conductivity layer 2
<i>per3</i>	Uniform	[1, 15]	Hydraulic conductivity layer 3
<i>perz1</i>	Uniform	[1, 15]	Hydraulic conductivity zone 1
<i>perz2</i>	Uniform	[1, 15]	Hydraulic conductivity zone 2
<i>perz3</i>	Uniform	[1, 15]	Hydraulic conductivity zone 3
<i>perz4</i>	Uniform	[1, 15]	Hydraulic conductivity zone 4
<i>d1</i>	Uniform	[0.05, 2]	Longitudinal dispersivity layer 1
<i>d2</i>	Uniform	[0.05, 2]	Longitudinal dispersivity layer 2
<i>d3</i>	Uniform	[0.05, 2]	Longitudinal dispersivity layer 3
<i>dt1</i>	Uniform	$[0.01 * d1, 0.1 * d1]$	Transversal dispersivity layer 1
<i>dt2</i>	Uniform	$[0.01 * d2, 0.1 * d2]$	Transversal dispersivity layer 2
<i>dt3</i>	Uniform	$[0.01 * d3, 0.1 * d3]$	Transversal dispersivity layer 3
<i>kd1</i>	Weibull	$(\alpha = 1.1597, \beta = 19.9875)$	Volumetric distribution coefficient 1.1
<i>kd2</i>	Weibull	$(\alpha = 0.891597, \beta = 24.4455)$	Volumetric distribution coefficient 1.2
<i>kd3</i>	Weibull	$(\alpha = 1.27363, \beta = 22.4986)$	Volumetric distribution coefficient 1.3
<i>poros</i>	Uniform	[0.3, 0.37]	Porosity
<i>i1</i>	Uniform	[0, 0.0001]	Infiltration type 1
<i>i2</i>	Uniform	[1, 0.01]	Infiltration type 2
<i>i3</i>	Uniform	[2, 0.1]	Infiltration type 3



5.1 資料集設定

該資料集包含了 300 個觀測數據，其中包括 20 個輸入變量，各輸入參數的分佈提供在表 5.1，意義為水力導電率、縱向和橫向分散度以及體積分佈係數等，該資料集有 10 個輸出變量，本研究採用 20 個輸入變量及第 1 個輸出變量。

1. 訓練集: 採用全部資料之 7 成作為訓練集。
2. 測試集: 全部資料扣除訓練集即為測試集。
3. 重複試驗: 本試驗重複 300 次，每次重複時之訓練集及測試集皆會不同。

5.2 預測誤差

由圖 5.1 可得，PE-Kmeans 的中位數最小且與其他演算法差距明顯；Supervised Compression 最大；Regression Tree 在兩者之間。在分布的型態上，PE-Kmeans 分布較集中，但在小範圍中稍微呈兩極化；Supervised Compression 分布最稀疏，呈現鐘形；Regression Tree 分布介於兩者之間，但有一些極端值。

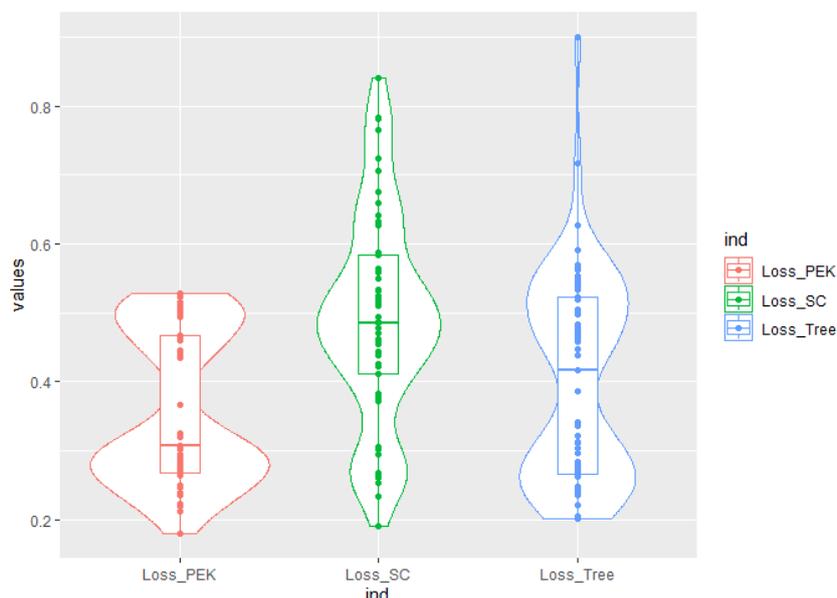


圖 5.1: RMSE 比較



5.3 組內組間變異分析

5.3.1 組內變異

由圖5.2可得，PE-Kmeans 的中位數最小，非常接近 0；Supervised Compression 最大；Regression Tree 在兩者之間。在分布的型態上，PE-Kmeans 分布非常集中，擁有很小的變異程度；Supervised Compression 次之；Regression Tree 分布相對非常廣泛，有較嚴重的極端值。

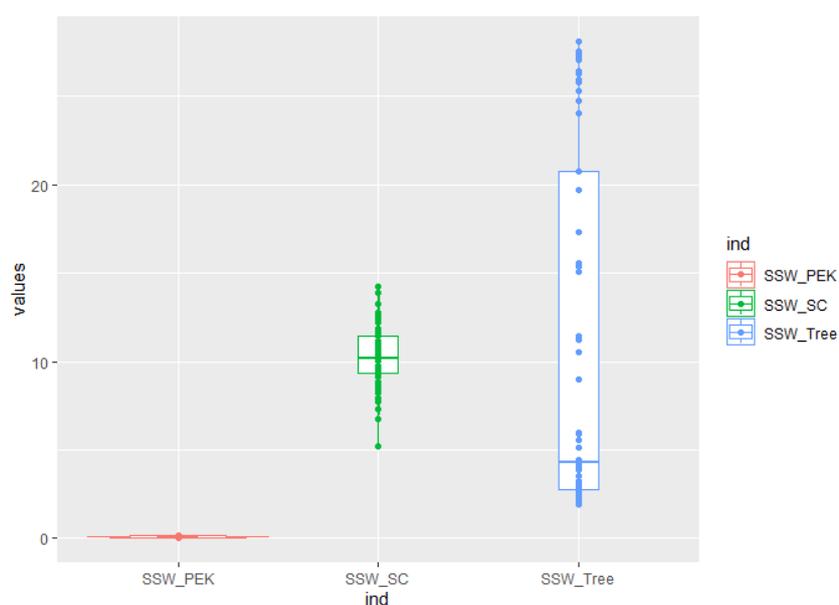


圖 5.2: 組內變異比較

5.3.2 組間變異

由圖5.3可得，PE-Kmeans 的中位數最大；Supervised Compression 與 Regression Tree 相差不遠。在分布的型態上，三種演算法差別不大。

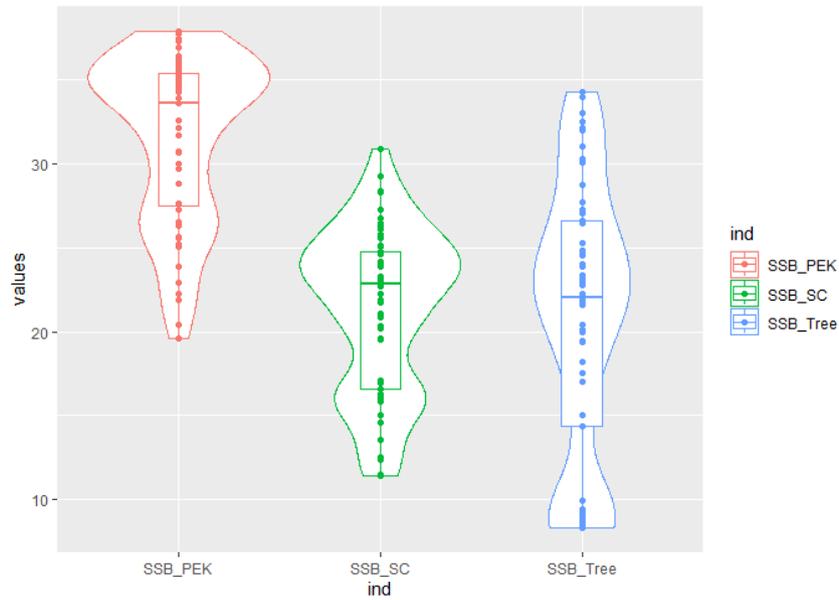


圖 5.3: 組間變異比較

5.4 運行時間

由圖5.4可得，PE-Kmeans 最耗時；Supervised Compression 次之；Regression Tree 非常接近 0。在分布的型態上，Regression Tree 非常集中，其餘兩種演算法差別不大。

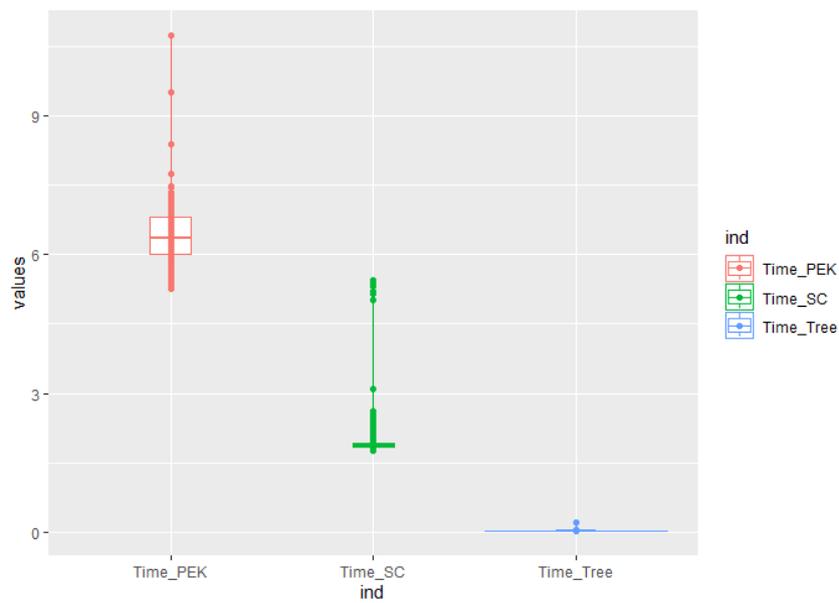


圖 5.4: 運行時間比較



Chapter 6 結論與未來展望

Regression Tree 為著名的無母數迴歸分析工具，它以二元分類逐步將輸入空間切割成多個區塊，在二維的輸入空間上切割成一個個長方形，每個區塊對應一個輸出值；Supervised Compression 為一種較新穎的資料壓縮演算法，原本資料壓縮通常是採用無監督式演算法，但此算法在輸入空間與輸出空間中選擇性以 K-means 聚類，至此成為監督式演算法，賦予了用於迴歸分析的可能性，它將輸入空間聚類成少數群心點，相當於切割成一個個 Voronoi region，每個區塊對應一個輸出值；基於上述概念，本研究提出 PE-Kmeans 演算法，則是先對輸出空間做 K-means 聚類，形成子空間，再對每個子空間的輸入空間做聚類，以整個訓練集資料做為與輸出值的對應，將原本無監督式的 K-means 聚類演算法賦予了用於迴歸分析的可能。

在本研究的函數模擬及真實資料集的實驗之下，我們得到與研究目的相呼應的結果，結果顯示本研究提出的 PE-Kmeans 在擬合較圓弧或不平滑函數以及真實資料時，相較 Regression Tree 及 Supervised Compression 能有較低的預測誤差，在無母數迴歸方法上具有獨特的功能性。

綜合本研究發現，研究顯示在固定資料數量為 2 萬筆之下：

1. 在低維度資料集上表現相對高維度資料集較好。
2. 相同維度之下，較複雜的函數表現相對較簡單的函數較好。
3. 受 SNR 的影響大，較為無雜音（高 SNR）的資料能有效縮小 PE-Kmeans 的預測誤差。

另外還研究觀察到 PE-Kmeans 的幾個特性如下：

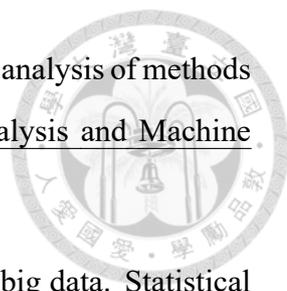
- 
1. PE-Kmeans 亦會隨著資料量的增加，而相對其他兩種演算法能更有效縮小預測誤差。
 2. 不管在何種函數模擬或是真實資料集上都有最小的組內誤差平方和及最大的組間誤差平方和。

從運算時間來看，PE-Kmeans 演算法相對耗時許多，前面提到在高維度空間時要減少 PE-Kmeans 的預測誤差可能需要透過增加資料量的方式，但因為增加資料量會導致運算時間的極大幅增加，導致變得難以實驗，PE-Kmeans 的耗時原因大部分是在進行搜尋兩次最佳分群數的時候產生，如果能優化這個搜索過程，能有效降低許多運算時間，將能探討 PE-Kmeans 在高維度資料上是否有能發揮的空間。



References

- [1] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. An efficient k-means clustering algorithm: analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):881–892, 2002.
- [2] Omer Sagi and Lior Rokach. Ensemble learning: A survey. WIREs Data Mining and Knowledge Discovery, 8(4):e1249, 2018.
- [3] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11):e00938, 2018.
- [4] Wolfgang Härdle. Applied nonparametric regression. Number 19. Cambridge university press, 1990.
- [5] James N. Morgan and John A. Sonquist. Problems in the analysis of survey data, and a proposal. Journal of the American Statistical Association, 58(302):415–434, 1963.
- [6] Badr HSSINA, Abdelkarim MERBOUHA, Hanane EZZIKOURI, and Mohammed ERRITALI. A comparative study of decision tree id3 and c4.5. International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Advances in Vehicular Ad Hoc Networking and Applications 2014, 4(2), 2014.
- [7] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. Classification and Regression Trees. Taylor & Francis, 1984.

- 
- [8] F. Esposito, D. Malerba, G. Semeraro, and J. Kay. A comparative analysis of methods for pruning decision trees. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(5):476–491, 1997.
- [9] V. Roshan Joseph and Simon Mak. Supervised compression of big data. Statistical Analysis and Data Mining: The ASA Data Science Journal, 14(3):217–229, 2021.
- [10] Partitioning Estimates, In: A Distribution-Free Theory of Nonparametric Regression, pages 52–69. Springer New York, New York, NY, 2002.
- [11] J. MacQueen. Some methods for classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1:281–297, 1967.