國 立 政 治 大 學 統 計 學 研 究 所

碩 士 學 位 論 文

General Adaptive Penalized Least Squares 模型選取方法之模擬與其他方法之比較

研究生 : 陳柏錞 撰

指導教授: 黃子銘 博士

中華民國 一佰零參 年 七 月

謝誌

穿著碩士服走在前往畢業典禮會場的路上，周圍的氣氛才讓我感覺終於到了這一刻，兩年的時間說長不長，日子在過的速度卻也快得有點驚人，想當初還只是大學生的我，懵懵懂懂的選擇了攻讀碩士班這條路，準備研究所的過程中，我一定要感謝幾位共患難的夥伴，分別是小班、懿慧、阿饅、尚均和佳逸，沒有你們也不會有今天從將政大畢業的我。

進到碩士班後，我認識了一群有時認真上進卻也總愛胡鬧的同學，大家有著相同的興趣，平時研究室中大伙認真地改考卷或是念書寫論文等等，禮拜三卻又會齊聚著一群人討論著如何參加一些特殊活動，這都讓研究生的生活熱鬧得很。在此需特別感謝某些同學，首先是宜萱，沒有妳熱心地給予同學們課業上的指導，我想大家壓力應該都會大得多吧。再來要感謝建佑，雖然你一直被嗆，但大家都知道你絕對是一位好好先生。再來還要感謝有著相同興趣的一群人，分別為維屏、華哥、嘉煒、東穎、軒哥、憲哥、PC 和范范，沒有你們想必研究室會少了很多來自討論或手機音效且能讓研究室熱鬧起來的聲音吧。再來則是要感謝我的指導教授黃子銘老師，因為我理解的速度比較慢，所以很容易讓人不耐煩，但老師總是能一步一步地帶領我思考的方向，而且非常照顧我們，儘管老師在國外，也會找時間聯絡我們看進度是否有在軌道上，且會直接使用網路電話討論，這都讓我覺得老師真的很用心，希望我的表現也能讓老師感到驕傲，謝謝老師。

再來我一定要感謝我家人，若沒有他們的支持，我無法認真準備研究所考試，也無法專心的攻讀碩士學位，而且在我感到身心疲憊時，只要回到家，總有進避風港的感覺，什麼都不會擔心，現在家中有小朋友的存在了，每次回家都會跟小

朋友玩得不亦樂乎，讓身心放鬆感到愉快，然後充完電後就可以在出發邁向自己的未來。

念碩士的過程是一時的，學的東西也要看自己將來會怎麼使用，但過程中學到的態度卻是能深深影響我一輩子的，所以最後我想感謝每一位曾經影響過我人生中這一階段的人，謝謝你們。

摘要

在迴歸分析中，若變數間具有非線性 (nonlinear) 的關係時，B-Spline 線性迴歸是以無母數的方式建立模型。B-Spline 函數為具有節點(knots)的分段多項式，選取合適節點的位置對 B-Spline 函數的估計有重要的影響，在希望得到 B-Spline 較好的估計量的同時，我們也想要只用少數的節點就達成想要的成效，於是 Huang (2013) 提出了一種選擇節點的方式 APLS (Adaptive penalized least squares)，在本文中，我們以此方法進行一些更一般化的設定，並在不同的設定之下，判斷是否有較好的估計效果，且已修正後的方法與基於 BIC (Bayesian information criterion)的節點估計方式進行比較，在本文中我們將一般化設定的 APLS 法稱為 GAPLS，並且經由模擬結果我們發現此兩種以 B-Spline 進行迴歸函數近似的方法其近似效果都很不錯，只是節點的個數略有不同，所以若是對節點選取的個數有嚴格要求要取較少的節點的話，我們建議使用基於 BIC 的節點估計方式，除此之外 GAPLS 法也是不錯的選擇。

【關鍵字】 B-Spline、GAPLS (General adaptive penalized least squares)、BIC、無母數方法、分段多項式、節點選取。

Abstract

In regression analysis, if the relationship between the response variable and the explanatory variables is nonlinear, B-splines can be used to model the nonlinear relationship. Knot selection is crucial in B-spline regression. Huang (2013) propose a method for adaptive estimation, where knots are selected based on penalized least squares. This method is abbreviated as APLS (adaptive penalized least squares) in this thesis. In this thesis, a more general version of APLS is proposed, which is abbreviated as GAPLS (generalized APLS). Simulation studies are carried out to compare the estimation performance between GAPLS and a knot selection method based on BIC (Bayesian information criterion). The simulation results show that both methods perform well and fewer knots are selected using the BIC approach than using GAPLS.

keywords: B-spline, generalized adaptive penalized least squares,BIC, nonparametric method, piecewise polynomial, knot selection

目錄

圖目錄

表目錄

第一章　緒論

在迴歸分析當中，我們考慮的模型如下:

$$y_i = f(x_i) + \varepsilon_i \ , \ i = 1,2,\cdots,n \ . \qquad (1\text{-}1)$$

其中 $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ 是我們的資料，ε_i 是誤差項，當 f 為非線性函數時，我們可以使用無母數的方法估計 f ，而此時資料來自於迴歸函數式子(1-1)，且資料本身假設 x_i 的值落在區間 (0,1) 之間，但若資料 x_i 的值沒落在區間 (0,1) 之間的話，則可以對資料作簡單的轉換，使其落在區間 (0,1) 內，等做完選取後在將比例復原即可。

B-Spline 迴歸是一種無母數的估計方法，在使用 B-Spline 函數近似 f 時 ，需要設定函數最高次數與節點的個數與位置，其中節點個數與位置的設定對 B-Spline 函數估計的影響很大，若是沒有任何處罰項的限制，很有可能會造成過度配適的問題，也就是節點選取過多，另一方面來看，節點若選取過少，則會造成估計誤差變大，所以節點選取的方法是很重要的。

B-Spline 函數估計的節點選取方法已有許多相關文獻所提出，例如: 基於 BIC 的節點估計方式與 Huang (2013) 所提出的一種選擇節點的方式 APLS (Adaptive Penalized Least Squares)，這兩種方法在 B-Spline 函數近似時，都使用了最小平方法和懲罰項的加總來取得誤差與節點數量間的平衡，目的都是希望能夠以較少的節點數量，就達到我們希望的近似效果。

在本文中，我們根據 APLS 為主，進行了一些更一般化的修改，在此我們將修改後的 APLS 法稱為 GAPLS 法(General Adaptive Penalized Least Squares) ，所以我們想要了解 GAPLS 法與其他不同的節點選取方法之間，找出的節點對於配適模型成效上有什麼相異與優劣之處。

第二章　文獻回顧

在本章節我們討論了兩種節點選取方法，分別為對基於 BIC 的節點估計方式與 APLS 法，兩種方法的目的都是希望能夠透過適當的懲罰項來制衡節點數量過度膨脹的問題，且兩方法的資料都來自於迴歸模型 (1-1)，其中 x_i 落在 (0,1) 之間。

2.1　基於 BIC 的節點估計方式

BIC 是由 Schwarz (1978) 所提出，其公式為 ：

$$\text{BIC} = -2 \cdot \ln\hat{L} + k \cdot \big(\ln(n) + \ln(2\pi)\big)，$$

其中 $\hat{L}$ 為概似函數，k 為 參數個數 ，n 為樣本個數，在 n 很大的時候，上式可近似為下列式子:

$$\text{BIC} = -2 \cdot \ln\hat{L} + k \cdot \ln(n)， \tag{2-1}$$

在B-Spline迴歸模型之下，若假設誤差 ε_i 服從 $N(0,\sigma^2)$，則可以經由一些推導，得到下列式子:

$$\text{BIC}(k) = n\ln\left(\frac{\text{RSS}(\hat{s}_k)}{n}\right) + k\ln(n)， \tag{2-2}$$

其中 $\hat{s}_k$ 為節點個數固定為 k 的那些節點向量當中，可使估計誤差 $\text{RSS}(\hat{s}_k)$ 的值達到最小的一組節點。接著在以此公式作為本方法評估模型選取的標準，之後將以 BIC 法當作此節點選取方法的代稱。

我們給定正整數k時，令 S_k 為 (0,1) 區間上長度為 k 的節點向量 t_k 所成的集合，其中節點為 $\xi_1, \ldots, \xi_k$，且不包含邊際點 0 和 1，則

$$\hat{s}_k = argmin_{t_k \in S_k} \sum_{i=1}^{n} \left(y_i - \hat{f}(x_i; t_k)\right)^2, \quad (2\text{-}3)$$

$$\mathrm{RSS}(\hat{s}_k) = \sum_{i=1}^{n} \left(y_i - \hat{f}(x_i; \hat{s}_k)\right)^2, \quad (2\text{-}4)$$

$$\mathrm{BIC}(k) = n\ln\left(\frac{\mathrm{RSS}(\hat{s}_k)}{n}\right) + k\ln(n), \quad (2\text{-}5)$$

我們使用統計軟體 R 的 optim 函數，找出最佳的節點個數與位置，此函數需要設定我們想要最小化的目標函數及一組起始節點向量供其計算，此時目標函數如下:

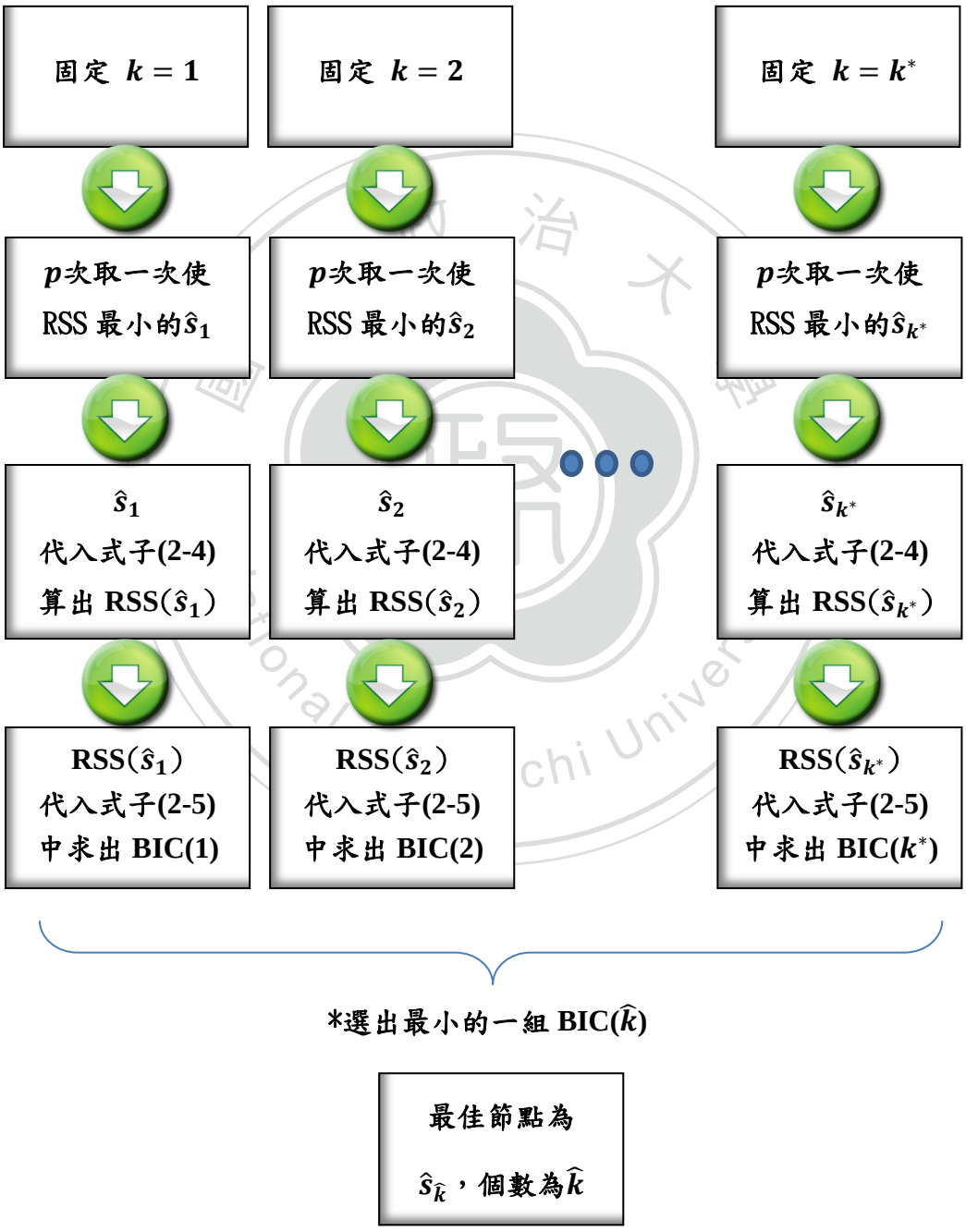
$$\sum_{i=1}^{n} \left(y_i - \hat{f}(x_i; t_k)\right)^2,$$

而起始節點向量的決定方法說明如下。給定節點向量長度 k 時，每次隨機在區間 (0,1) 上抽取 k 個節點，並自行決定共抽取 p 次，每次均計算其 RSS，在 p 次中選取 RSS 最小的那組節點 ，則此組節點即為 $\hat{s}_k$ 。

接著對 $k = 1, 2, 3, \ldots, k^*$ 皆做相同的步驟，計算出固定每個 k 之下最佳的節點向量 $\hat{s}_i, \mathrm{i} = 1,2, \ldots, k^*$ ，其中 k^* 為我們預計取到的最大節點數，在將這些都代入式子(2-4)中算出 RSS 值後，在將 k^* 個 RSS 值代入式子(2-5)中求出 BIC，進而在之間挑出使 BIC 值最小的那組節點向量當作最適解 $\hat{s}_{\hat{k}}$，其最佳節點數為 $\hat{k}$ ，因為以模擬的結果來看，節點數通常不會選超過十個，所以本文第四章節模

擬時，都只令 k^* 最多到 10 ， $p=3$ 的情況去進行模擬。

方法流程圖



2.2 APLS 法

本節為 Huang (2013) 所提出的一種選擇節點的方式，此方法和 BIC 法雖然都是有帶入懲罰項的觀念，但此方法還有將最適性(Adaptive)納入考慮，證明不用太多的樣本數就可以達成一樣的收斂速度，而此方法的最終目的來說一樣是在對處罰項給定限制之下，找出最合適的節點數量與位置。

此方法主要與 BIC 法不同的地方則是，S_k 雖一樣為 (0,1) 區間上長度為 k 的節點向量 t_k 所成的集合，但節點向量中的節點 $\xi_1,\dots,\xi_k$ 會從 $\left\{\frac{1}{2^J},\dots,\frac{(2^J-1)}{2^J}\right\}$ 之中取出，目的在於將可能選取的節點位置分開來，可避免同一個位置配適太多的節點數， 且節點之中不包含邊際節點 0 和 1，其中 J 為切割可能節點數量的正整數，所以為不同 J 之下可能的節點集合長度是不同的，於是為了分辨不同的 J 之下的 S_k，之後將以 S_j 來表示。

將 S_j 之中的節點向量去做 B-Spline 函數估計，皆可以得到 B-Spline 函數的基底，且其基底係數的絕對值均須小於某正整數 b，且 q 為一可自行設定的參數，功能為設定 B-Spline 函數為幾次多項式的函數，例如:通常會令 $q=3$，且多項式的次數為 $q-1$ ，所以可使函數圖形至少存在曲線關係。將上列得到的參數與節點蒐集起來後，我們可以得到一組 index j =$(b,q,\xi_1,\dots,\xi_k)$。

令

$$\widetilde{\Delta}_{2,j}=max_{1\le i\le k+1}(\xi_i-\xi_{i-1})，\tag{2-6}$$

$$\widetilde{\Delta}_{1,j}=min_{1\le i\le k+1}(\xi_i-\xi_{i-1})，\tag{2-7}$$

$$B'_j = \sqrt{2\pi e}\left(0.5 + \frac{q\sqrt{q}(2q+1)9^{q-1}\sqrt{\tilde{\Delta}_{2,j}}}{\sqrt{\tilde{\Delta}_{1,j}}}\right), \tag{2-8}$$

$$r_j = 1\vee\frac{1}{q\sqrt{\tilde{\Delta}_{2,j}(k+q)}}, \quad (\vee\text{: 兩者取最大}) \tag{2-9}$$

B'_j 當中主要有兩個參數，其為式子(2-6)和式子(2-7) ，此兩參數控制的是節點之間的距離，且因為 B'_j 是被放在懲罰項之中，所以此目的也是為了懲罰選取的節點之間太過接近。

且 $J_j = min\left\{J \geq 1 : \xi_1, \ldots, \xi_k \text{ are in } \left\{\frac{1}{2^J}, \ldots, \frac{(2^J-1)}{2^J}\right\}\right\}$，在 $\xi_0 = 0$ 且 $\xi_{k+1} = 1$ 之下。因為上述的 index j 有很多可能，所以在此令 Λ 為一個包含所有 index $j = (b, q, \xi_1, \ldots, \xi_k)$ 的集合，其中 b 為控制 B-Spline 基底的參數和 q 為控制 B-Spline 函數多項式的次數，此兩參數皆為正整數，其中 $2^{J_j} + q + b \leq n$ ，$r_j(2b)^2 \leq \delta_n$，且 $\{\delta_n\}$ 須滿足 ：

$$\lim_{n\to\infty} \delta_n = \infty \quad ,$$

和

$$\lim_{n\to\infty} \frac{\delta_n \log(n)}{n^\alpha} = 0 \quad , \text{ for all } \alpha > 0 \ ,$$

則當 $\{a_n\}$ 為一正數組成之數列，可自行設定，只須滿足$\lim_{n\to\infty} a_n = \infty$，用於

$$\hat{s} = argmin_{j\in\Lambda, u\in S_j}\left(\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i; t_k)\right)^2 + \eta_j\right), \quad (2\text{-}10)$$

$$\eta_j = \frac{a_n r_j (2b)^2}{n}\left((k+q)\log\left(B'_j\right) + (\log 2)2^{J_j} + q + b\right), \quad (2\text{-}11)$$

之下，最終選出的 $\hat{s}$ 即為本文所求的最適節點位置。

第三章 研究方法

本章節主要是為了比較 BIC 法與 一般化的 APLS 兩種方法對於節點選取的效果好壞，藉由各自取出的節點去估計 B-Spline 基底，進而求出估計值後，再計算估計誤差來進行比較，在此我們先介紹一下何謂一般化的 APLS，之後我們皆將一般化的 APLS 簡稱為 GAPLS。

3.1 GAPLS 法

此方法與 APLS 法的差異在於懲罰項的設定更一般化，也就是參數可以根據不同情況讓使用者自行設定，在 GAPLS 中修改了 APLS 法 當中的 η_j，以及 a_n, δ_n 的選取方式:

其中主要多加入的參數之中c_1, c_2 主要是控制懲罰項η_j前後兩段的權重，c_1 可控制 J, q 和 b 的大小不要太大，而 c_2 可控制 a_n 的大小，使 η_j 後面那段即使偏大或偏小，η_j 都可以 藉由 a_n 來控制。其中:

1. η_j 改成

$$\eta_j = \frac{a_n(2b)^2}{n}\Big((k+q)\log\big(B'_j\big) + c_1 K_0[(\log 2)2^{J_j} + q + b]\Big) \quad ,$$

其中參數 $c_1 > 0$ 為可調整的常數，而

$$K_0 = \sqrt{2\pi e}\big(3\sqrt{3}(7)9^2\big)$$

為本文模擬時自行設定的參數值。

2. δ_n 選取方式: 資料 $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ 產生後，先配適以下迴歸模型:

$$y_i = \sum_{j=1}^{m} \theta_j B_j(x_i) + \varepsilon_i \ , i = 1, \ldots, n \ ,$$

其中 $B_1, \ldots, B_m$ 為 B-spline 基底，order=3，且取用的節點為在 (0,1) 之間等距排放，個數為

$$\min\left(20, \lfloor n^{1/3} \rfloor\right) \ ,$$

$\lfloor \cdot \rfloor$ 為 floor function，意思就是不超過其值的最大整數，在此因為 n 必為正整數，所以只需將 $n^{1/3}$ 小的小數部分去掉即可，然後令 $\hat{\theta}_1, \ldots, \hat{\theta}_m$ 為最小平方法所估計出來的迴歸係數，而

$$\hat{b} = \left\lfloor \left(\max_{1 \le j \le m} \left| \hat{\theta}_j \right| \right) \right\rfloor + 2 \ ,$$

最後取 $\delta_n = 4\hat{b}^2 \log(n)$ 。

3. a_n 選取方式:首先須計算:

$$\hat{\sigma} = \text{median}_{1 \le i \le n/2} \frac{|y_{2i} - y_{2i-1}|}{0.6745\sqrt{2}} \ ,$$

則最後選擇

$$a_n = \frac{c_2 \hat{\sigma} \log(n)}{4\hat{b}^2 K_0\left(2(log2) + 4 + \hat{b}\right)} \ ,$$

其中 $c_2 > 0$ 為可調整的常數。

(計算 B-spline 基底時，我們運用了 R 的 splines 套件當中的 bs 指令，其中有個參數 q (order) 是自行設定的，通常我們都以 3 代入，原因在 page 6 有解釋。)

3.2 實際模擬

本文研究目的是在不同的情況之下，針對 GAPLS 進行模擬，進而比較與 BIC 法之間的優劣，本文中模擬資料的產生，皆根據式子 1-1 所產生，而我們一共考慮了八種真實函數 (詳見附錄一)，且每一種設定下產生五十筆的模擬資料 $(x_1, y_1), (x_2, y_2), \cdots, (x_{50}, y_{50})$。在此主要使用統計軟體 R，其中用到三個 R 的套件: Matrix、splines 以及 Quadprog。

在模擬的過程當中我們發現理論上的想法，並無法完全應用在實際操作上面，例如:

1. 參數的部分

GAPLS 中的懲罰項中有許多參數，其中式子 2-9 後面的部分

$$\frac{1}{\mathrm{q}\sqrt{(\tilde{\Delta}_{2,j}\ (\mathrm{k+q}))}} \leq 1 \quad ,$$

因為必定小於等於一，所以 r_j 一定為常數 1，因此可將此參數簡化為 1 就好。

2. 實際模擬

在模擬時會遭遇的困難在於當我們決定 δ_n 後，因為在 $2^{J_j} + q + b \leq n$ 與 $r_j(2b)^2 \leq \delta_n$ 的限制之下，所有可能的 q、b 組合非常多，因此資料量會過於膨脹，例如:新方法中的 J，只要大於等於四，所有的點就為 2^J 個且 $J \geq 4$ ，然後滿足所有限制之下，J_i 又會取 k 個，所以總共有 $C_k^{2^J}$ 種組合，$k = 1,2,\ldots,2^J$ ，全部都要求出且算出最後的

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{f}(x_i;t_k)\right)^2+\eta_j$$

進而找出最佳的一組節點向量 t_k ，可想而知如果 $J=4$ ， $k=8$ 這特例，$C_k^{2^J}$ 就有 12870 種，每個都要算出 $\hat{s}$ ，對電腦來說是很大的負擔且沒效率。

所以針對這項問題我們在模擬時選了另外的方式進行節點選取，因為模擬的結果發現，選取的節點數量大多都不會超過十個，所以我們先用 optim 這指令找出沒任何限制之下的最佳解 $\hat{t}_{\textbf{unrestricted}}=\xi_1,\ldots,\xi_k$ ，然後先把 $J\geq m$ ，$\left\{\frac{1}{2^J},\ldots,\frac{(2^J-1)}{2^J}\right\}$ 所有可選取的節點皆列出且挑出所有長度為 m 的向量(在此先以 $\tilde{t}=t_1,\ldots,t_m$表示)，運用最小平方法的概念，求出平方和後，使平方和最小的那組向量$\widetilde{t_l}$，即為最佳的節點向量$\hat{t}$，如下式:

$$\hat{t}=\underset{\widetilde{t_l}\in S_m}{\operatorname{argmin}}\left(\sum_{m}^{10}\sum_{1}^{m}(\widetilde{t_l}-\hat{t}_{\textbf{unrestricted}})^2\right).$$

第四章　模擬與比較

在本章節我們主要會先為了 GAPLS 法裡的參數 c_1, c_2 進行模擬，希望能找到在不同情況之下，應該選取什麼 c_1, c_2 的值來進行節點選取，再來則會與 BIC 法進行模擬比較，看看在不同類型的真實函數之下，誰的估計效果較佳，但因為不同的真實函數其產生的資料本身變異的程度就不同，所以我們使用控制訊噪比（Signal to noise ratio）的方式，來讓我們更容易比較不同類型的真實函數所產生的資料，其中:

$$\text{Signal to noise ratio} = \frac{\text{sd}(f(x_i))}{\text{sd}(\varepsilon_i)} \quad , i = 1, \dots, n$$

而在本文中，我們皆先將 $\text{sd}(y_i)$ 調成 1，再將 $\text{sd}(\varepsilon_i)$ 調成 1/7，即可將產生資料的訊噪比調整為 7，此處設訊噪比為 7 只是任意假設的值，目的希望亂數的變動可以比較小，之後也會將訊噪比更改為 4 進行模擬。

4.1 模擬 c_1, c_2

首先我們只生成第二組真實函數的亂數進行 c_1, c_2 的比較，且之後模擬時所使用的 x_i 為在 0 到 1 之間等距排放的五十個點，之後我們先模擬 c_1, c_2 在各種不同組合之下，分別做十次模擬，在將十次模擬後得到的估計誤差加以計算平均數與標準差，而 c_1, c_2 我選擇了兩種水準，主要為介於 0 到 1 之間的低水準，以及介於 1 到 100 的高水準來做分配組合，而模擬的結果如下列各表:

表(一) c_1 高水準 c_2 高水準

c_2 \ c_1	1	10	50	100
1	0.2853 (0.3996)	0.2853 (0.3996)	0.2853 (0.3996)	0.2853 (0.3996)
10	0.2852 (0.3981)	0.2852 (0.3981)	0.2852 (0.3981)	0.2852 (0.3981)
50	0.2747 (0.3914)	0.2747 (0.3914)	0.2747 (0.3914)	0.2747 (0.3914)
100	0.2822 (0.4012)	0.2822 (0.4012)	0.2745 (0.3934)	0.2745 (0.3934)

表(二) c_1 高水準 c_2 低水準

c_2 \ c_1	1	10	50	100
0.01	0.2873 (0.4023)	0.2873 (0.4023)	0.2841 (0.4003)	0.2841 (0.4003)
0.3	0.2856 (0.3993)	0.2847 (0.3999)	0.2847 (0.3999)	0.2847 (0.3999)
0.7	0.2852 (0.3996)	0.2852 (0.3996)	0.2852 (0.3996)	0.2852 (0.3996)
1	0.2853 (0.3996)	0.2853 (0.3996)	0.2853 (0.3996)	0.2853 (0.3996)

表(三) c_1 低水準 c_2 高水準

c_2 \ c_1	0.01	0.3	0.7	1
1	0.2853 (0.3996)	0.2853 (0.3996)	0.2853 (0.3996)	0.2853 (0.3996)
10	0.2852 (0.3981)	0.2852 (0.3981)	0.2852 (0.3981)	0.2852 (0.3981)
50	0.2747 (0.3914)	0.2747 (0.3914)	0.2747 (0.3914)	0.2747 (0.3914)
100	0.2822 (0.4012)	0.2822 (0.4012)	0.2745 (0.3934)	0.2745 (0.3934)

表(四) c_1 低水準 c_2 低水準

c_2 \ c_1	0.01	0.3	0.7	1
0.01	0.2873 (0.4023)	0.2873 0.4023)	0.2873 (0.4023)	0.2873 (0.4023)
0.3	0.2879 (0.4019)	0.2879 (0.4019)	0.2879 (0.4019)	0.2856 (0.3993)
0.7	0.2879 (0.4020)	0.2879 (0.4020)	0.2856 (0.3994)	0.2852 (0.3996)
1	0.2885 (0.4016)	0.2862 (0.3990)	0.2853 (0.3996)	0.2853 (0.3996)

由表(一)到表(四)的結果來看，我們可以看出在理論上 c_1, c_2 的設定值雖需視情況所定，但模擬的結果顯示，不同的 c_1, c_2 組合下，估計的效果並不會有太大的差異，大多都要到小數點第三位之後才會產生差異，此外以 GAPLS 法選

取的節點進行 B-spline 的函數估計，其估計誤差的標準差也都為 0.4 左右，於是結果顯示以 GAPLS 法來選取節點時，因為不同的 c_1, c_2 組合下，估計的效果都不會有太大的差異，對於 B-spline 的函數估計穩定性很高，所以我們建議取 $c_1 = 1, c_2 = 1$ 即可，且此時 c_1, c_2 控制權重的部分會與 APLS 法相同。

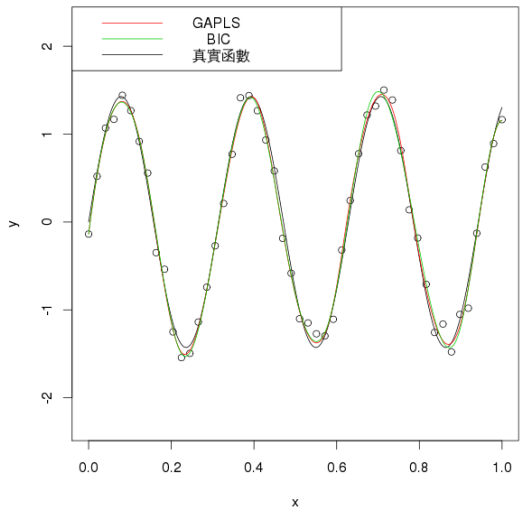
4.2 GAPLS 法 與 BIC 法的比較

本節目的在於比較 GAPLS 法與 BIC 法之間，在節點選取數量與估計效果優劣做比較，首先我們總共有八種真實函數，每種函數生成十組固定種子之下的隨機樣本資料，且兩方法皆針對這些資料進行模擬，進而比較兩法之間對於資料的估計誤差大小，以及估計的穩定度，和是否容易造成過度配適，而取過多節點數。

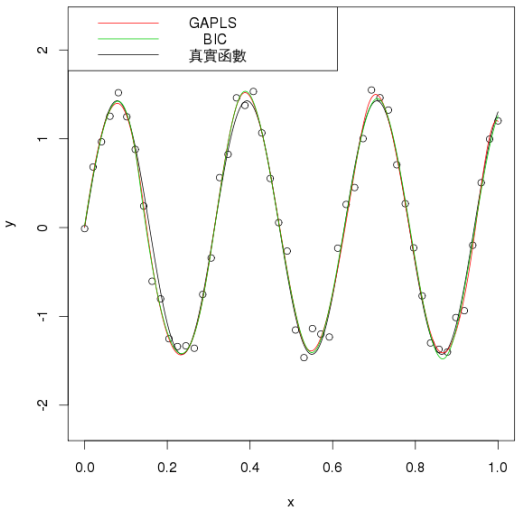
首先下列估計函數圖共分成兩種，第一種為 GAPLS 法模擬十次之下，此法估計誤差最小的那組樣本，然後將 GAPLS 法與 BIC 法根據此組資料所配適出來的函數圖放在一起比較，第二種則為 BIC 法模擬十次之下，此法估計誤差最小的那組樣本，而畫圖的方法則如第一種，且函數估計圖下方有顯示此估計效果來自哪組樣本。

於是八種函數 (附錄一) 我們共可以得到十六張圖，並且以根據下列估計函數圖發現 GAPLS 法與 BIC 法的估計效果都很不錯，配適出來的 B-Spline 迴歸函數與真實函數的差異都很小，不論是平滑型的函數圖型 (前四組函數)，或是有間斷型的函數圖型 (後四組函數)。

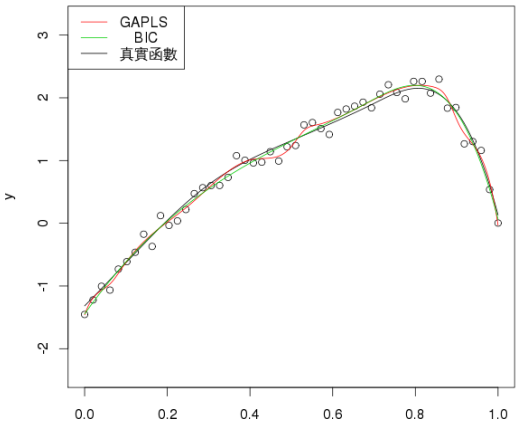
圖(一) 函數一



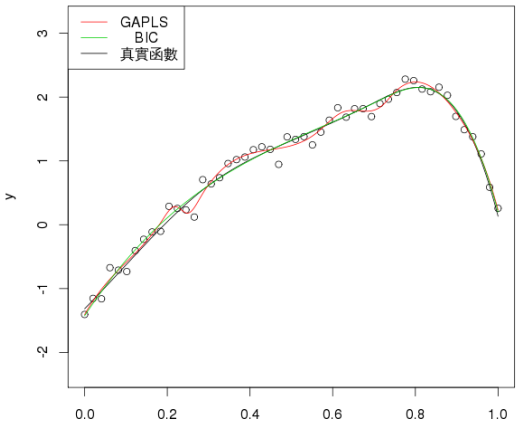
圖(二) 函數一



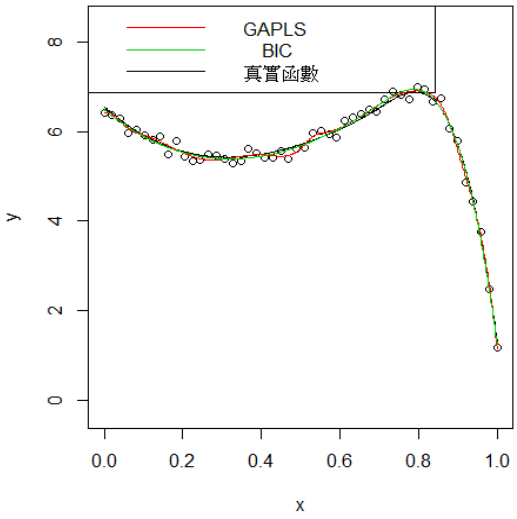
圖(三) 函數二



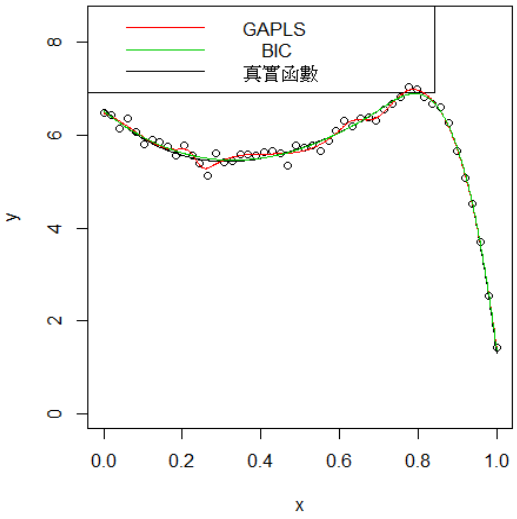
圖(四) 函數二



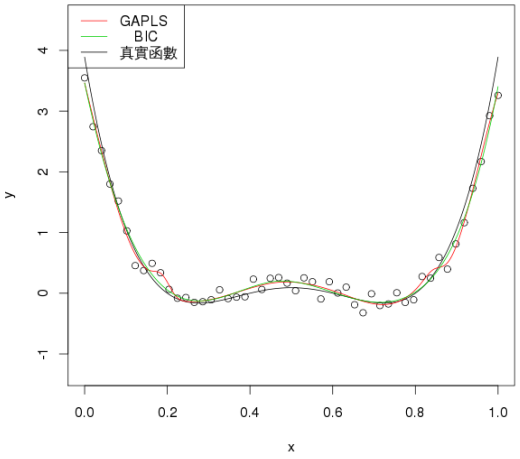
圖(五) 函數三



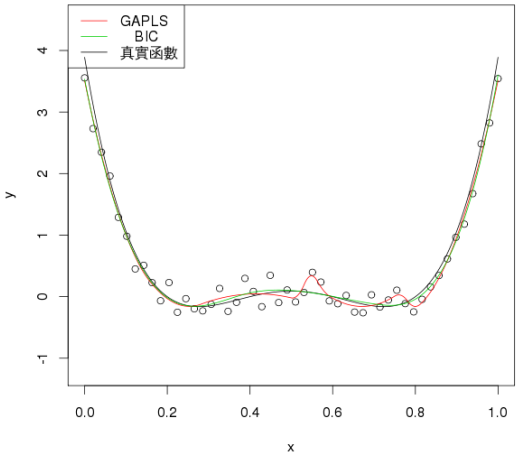
圖(六) 函數三



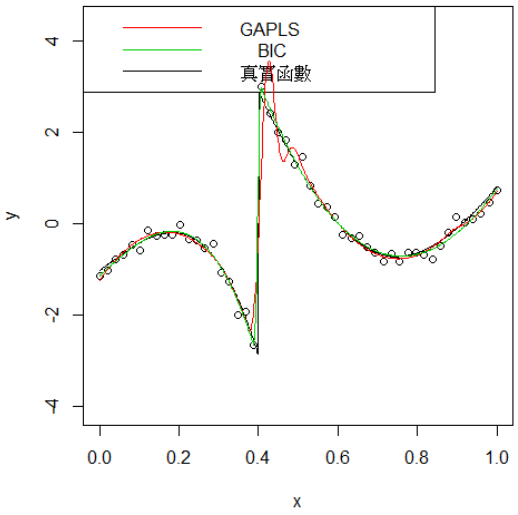
圖(七) 函數四



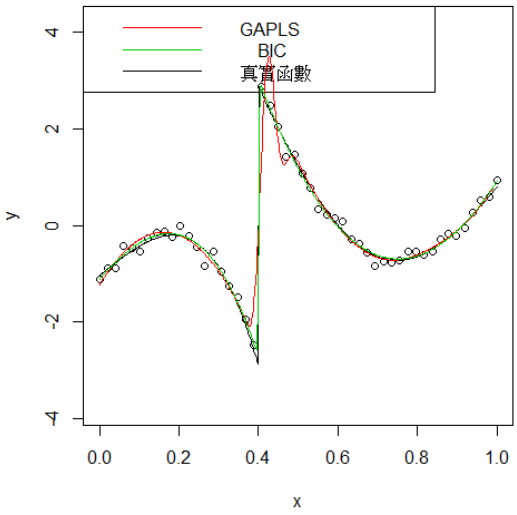
圖(八) 函數四



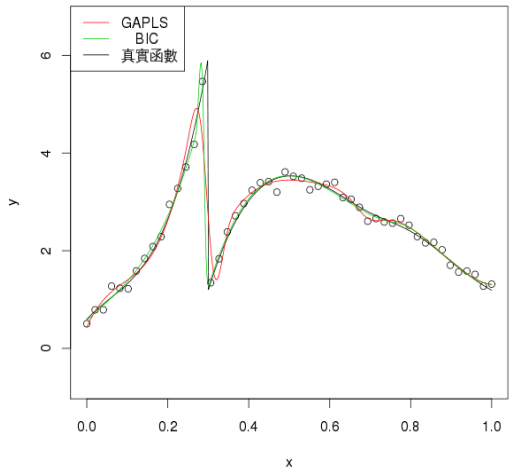
圖(九) 函數五



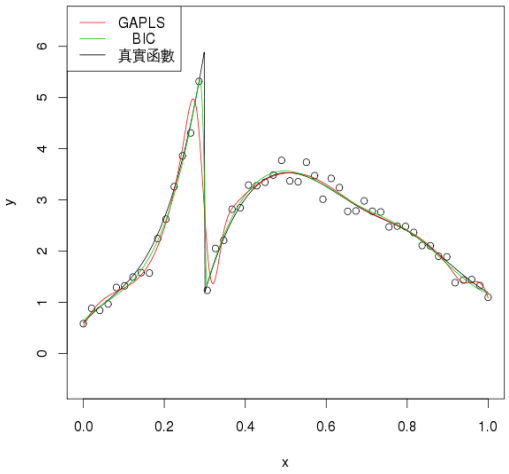
圖(十) 函數五



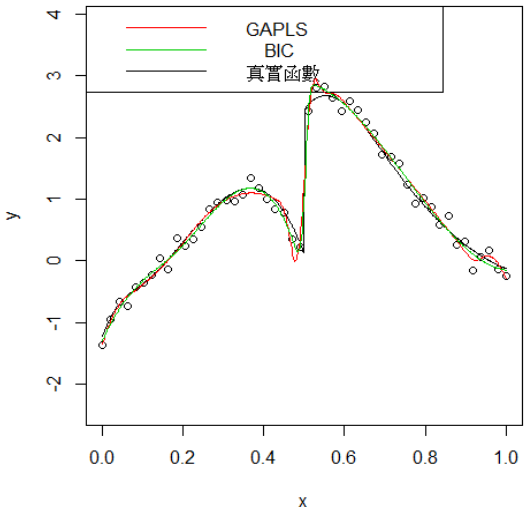
圖(十一) 函數六



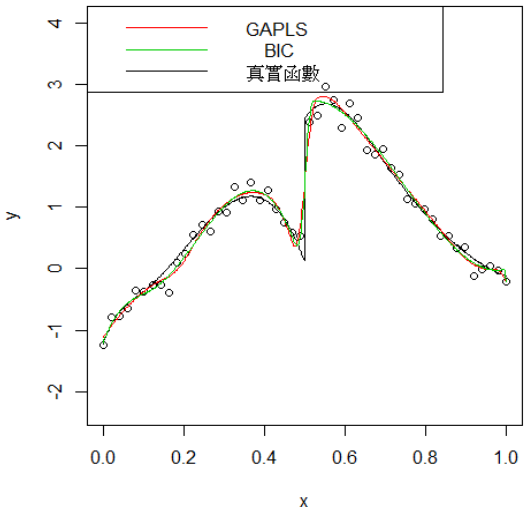
圖(十二) 函數六



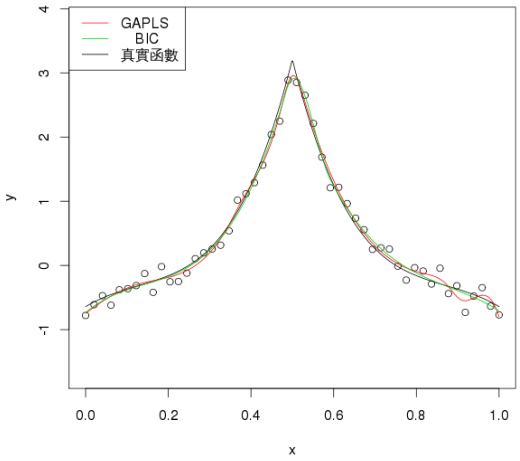
圖(十三) 函數七



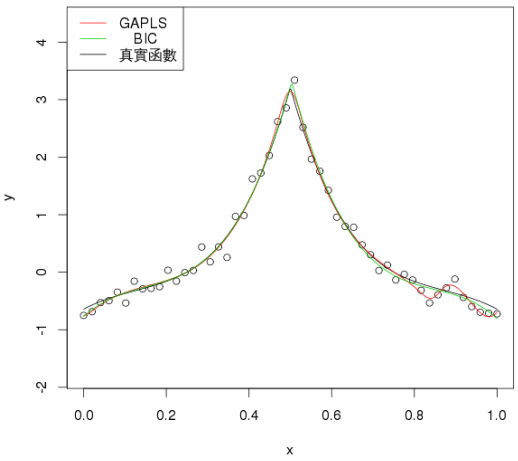
圖(十四) 函數七



圖(十五) 函數八



圖(十六) 函數八



因為圖型看起來估計效果都不錯，只看圖型看不出有什麼明顯的差異，那麼我們現在就使用 GAPLS 法與 BIC 法的估計誤差值來做比較，結果如下表:

表(五) 估計誤差的平均和標準差

函數 / 方法$(\bar{X},s)$	函數一	函數二	函數三	函數四
GAPLS	0.2783 (0.4178)	0.2899 (0.4203)	0.2881 (0.4158)	1.3286 (1.4673)
BIC	0.0081 (0.0031)	0.0042 (0.0032)	0.0040 (0.0037)	0.0178 (0.0050)

表(六) 估計誤差的平均和標準差

函數 / 方法$(\bar{X},s)$	函數五	函數六	函數七	函數八
GAPLS	0.3958 (0.4238)	0.3505 (0.6653)	0.2853 (0.3996)	0.2903 (0.4338)
BIC	0.1093 (0.0942)	0.0912 (0.0459)	0.02435 (0.0124)	0.00632 (0.0035)

首先由表(五)與表(六)可看出，不論函數的種類為何，BIC 法的估計誤差的平均數皆小於 GAPLS 法的估計誤差的平均數，但雖然 GAPLS 法在估計誤差的比較上是弱於 BIC 法的，可是其實兩種方法的平均估計誤差都不至於太大，藉由看圖(一)到圖(十六)也可以得知估計效果其實都還滿不錯，在來看估計誤差的標準差，BIC 法的估計誤差的標準差非常小，但 GAPLS 法的估計誤差的標準差就比較大了，可見兩種方法的估計穩定度不太一樣，在模擬中我發現此原因為 GAPLS 的估計誤差，十次中會有 1~2 次的值特別大，所以才會使得 GAPLS 法的估計誤差的平均數和標準差變大，雖然 BIC 法的估計效果在上述設定的情況之下比 GAPLS 法來得優秀，但兩方法對於迴歸函數近似來說，都還算是不錯的方法。

在來細看各函數模擬十次之下，GAPLS 法的估計誤差比 BIC 法的估計誤差來得小的個數，進而更仔細的比較兩方法之間優劣，模擬結果如下:

表(七) 十次中，GAPLS 法比 BIC 法好的個數

函數 / 估計誤差	函數一	函數二	函數三	函數四
# of GAPLS<BIC	0	0	0	0

表(八) 十次中，GAPLS 法比 BIC 法好的個數

函數 / 估計誤差	函數五	函數六	函數七	函數八
# of GAPLS<BIC	3	4	0	0

根據上表可以看出，雖然 BIC 法的估計效果比 GAPLS 法好得多，但是我們發現在不同的函數設定之下，兩方法之間對於估計效果的優劣關係還是會有差異的。

在來比較兩方法之下，對於不同的函數類型，其最後選取節點的平均個數，模擬結果如下表:

表(九) 平均選取節點數

函數 / 平均#knot	函數一	函數二	函數三	函數四
GAPLS	9.6	9.5	9.7	9.6
BIC	8.1	2	2	2.7
真實函數	*	3	2	*

(*因為為多項式函數，所以真實函數並無節點設定)

表(十) 平均選取節點數

函數 / 平均#knot	函數五	函數六	函數七	函數八
GAPLS	9.3	8.9	8.7	9.7
BIC	5	6.8	5.9	5
真實函數	5	6	6	6

結果顯示 GAPLS 法不論資料為什麼型態的函數關係，其選取的節點數量全都高於真實函數的節點數量，而且個數幾乎都落在 8 到 10 個之間，很接近我們設定的最高數量,相對的 BIC 法則選取的節點數量與真實函數相比則大都偏少，但比較接近真實函數的節點數量，這很有可能是因為 BIC 法中的懲罰項只單純懲罰節點配適過多的問題。

因為 BIC 法的近似效果在上述假設中大致看起來都比 GAPLS 法來的有效，所以我們考慮了一些基本假設的更動，首先第一部分，我選擇將建立隨機樣本時所使用的種子換成在接下來的十組，也就是換了一批隨機資料，然後照著前面相同的模擬方法進行模擬比較，模擬結果如下:

表(十一) 下十組資料

函數 / 估計誤差	函數一	函數二	函數三	函數四
# of GAPLS<BIC	1	1	1	0

表(十二) 下十組資料

函數 / 估計誤差	函數五	函數六	函數七	函數八
# of GAPLS<BIC	1	3	2	1

我們將這裡產出的表(十一)與表(十二)和表(七)和表(八)相比，可以看出已兩種設定之下已有明顯的不同，發現 GAPLS 法在十次的模擬當中，其估計誤差比 BIC 法的估計誤差來得小的個數已經明顯增加，所以我們可以得知不同的資料對於 GAPLS 法與 BIC 法的估計效果來說，影響也是滿明顯的。

再來第二部分的更動則是訊噪比，在此我不僅將資料換為接下來的十組資料，且也將資料的訊噪比調整為 4，讓資料的變動看起來更大，模擬結果如下:

表(十三) 下十組資料且訊噪比為四

函數 / 估計誤差	函數一	函數二	函數三	函數四
# of GAPLS<BIC	1	1	1	2

表(十四) 下十組資料且訊噪比為四

函數 / 估計誤差	函數五	函數六	函數七	函數八
# of GAPLS<BIC	1	3	2	1

看起來與表(十一)和表(十二)差異並不大，但是函數四的情況下，GAPLS 法比 BIC 法好的個數多了一次，所以可以得知其實訊噪比對於這兩方法的估計也是有一些影響的。

最後是第三部份的更動，這是是除了前兩部分的更動以外，還將樣本數調整為一百筆，而模擬結果如下:

表(十五) 下十組資料且訊噪比為四且樣本數為一百筆

函數 / 估計誤差	函數一	函數二	函數三	函數四
# of GAPLS<BIC	2	2	2	0

表(十六) 下十組資料且訊噪比為四且樣本數為一百筆

函數 / 估計誤差	函數五	函數六	函數七	函數八
# of GAPLS<BIC	5	6	3	2

在此我們可以看到，GAPLS 法的估計效果比 BIC 法的估計效果來得好的個數已經有明顯的增加，所以可以得知當樣本數變大時，很有可能 GAPLS 法會比 BIC 法來得表現優異。

最後看看兩方法在更改設定之後，其平均選取的節點個數是否有較接近真實函數的節點個數，但因為 GAPLS 法所選取的節點數量都落於 8 到 10 之間，並不會因為更改設定而有什麼明顯的影響，所以這裡我將 BIC 法模擬的結果進行整理，結果如下表：

表(十七) 節點差異

函數 / 平均節點差	函數一	函數二	函數三	函數四
下十組資料	*	1.3	2.2	*
訊噪比調整為四	*	1.1	2.2	*
樣本增加至一百	*	-0.4	0	*

表(十八) 節點差異

函數 / 平均節點差	函數五	函數六	函數七	函數八
下十組資料	1.3	1.2	0.7	-0.4
訊噪比調整為四	-0.7	1.5	0.2	-1.5
樣本增加至一百	0	-0.4	-0.2	-2.6

這裡可以看出，不同的設定之中，對於節點估計的準確度幾乎沒有什麼明顯的趨勢，其中較為明顯的因素則是樣本數，可以看出在本文的研究範圍內，樣本增加到一百筆以後，對於節點數量的準確度則會提升很多。

最後可以得到一些小結論，不管函數類型、參數值、不同的隨機資料、訊噪比的大小以及樣本數的多寡，GAPLS 法所選取的節點數量幾乎都落在 8 到 10 個之間，而當樣本數增加時，GAPLS 法所選取的節點數量會稍微接近真實函數的節點數量，此外在不同的設定之下最有差異的地方，在於可以明顯的看出 GAPLS 法的估計效果比 BIC 法的估計效果來得好的次數有明顯增加，所以當訊噪比降低和樣本數增加時，GAPLS 法的估計效果可以有效地提升。而 BIC 法對於節點選取的個數，在樣本數提升的情況之下，BIC 法選取的節點就很靠近真實節點數量，整體看起來 BIC 法的估計效果是很不錯的。

第五章 結論與建議

5.1 結論

根據模擬的結果，我們可以得到以下幾個結論:

1. 不論資料服從什麼型態的函數關係，GAPLS 法與 BIC 法的函數近似效果都很不錯，而且穩定性很高。
2. GAPLS 法不論資料的型態為何，選取的節點數量皆偏多，而 BIC 法在平滑函數時選取的節點數量偏少，在間斷型函數時則偏多，但在模擬的情況當中，數量皆比 GAPLS 法所選取的節點數量來得少。
3. GAPLS 法中的c_1, c_2, r_j 只需令其為 1 即可。
4. 不同的隨機樣本以及樣本數，皆對 GAPLS 法與 BIC 法的估計效果有很大的影響，而在我們考慮的樣本數範圍內，樣本數越多，則 GAPLS 法的估計效果比 BIC 法的估計效果來得好的機率就越高

5.2 建議

根據以上的結論，如果想要使用 B-Spline 進行迴歸函數近似，但不要求控制節點數量的話，GAPLS 法與 BIC 法都是很不錯的近似方法，但若有要求節點數量較少的話，因為 BIC 法只單純懲罰節點數過高的問題，所以 BIC 法所選出的節點數量都偏少，於是我們推薦使用 BIC 法進行迴歸函數近似。

5.3 延伸題目

未來可以再加入更多不同的方法與真實函數進行模擬比較，以及探討收斂速

度的問題，例如當給定估計誤差須達到多少以下，則須使用多少樣本來讓估計更準確，則可進一步討論哪些方法更節省樣本數，還有訊噪比的差異以及大樣本之下(一千筆以上)可能會有什麼不同的情況產生。

附錄一

第四章節中，使用八種不同的函數比較兩種節點選取的方法，而資料為真實函數加上一隨機誤差項所生成:

$$y_i = f(x_i) + \varepsilon_i\,, i = 1\,, 2\,, \dots\,, n$$

且模擬時需自行將訊噪比調整為 7，此值為自行設定，且調整方式如第四章引言所述。

八種真實迴歸函數如下:

函數一:

真實迴歸函數為 $f(x_i) = \sin(20 \times x_i)$，誤差項 $\varepsilon_i \sim N(0, 0.23^2)$

函數二:

真實迴歸函數 B-Spline 函數，節點為 $t = (0.3,0.5,0.7)$，係數為 $\beta = (-10, -5,5,10,15,20,1)$，誤差項 $\varepsilon_i \sim N(0, 2.49^2)$

函數三:

真實迴歸函數 B-Spline 函數，節點為 $t = (0.5,0.7)$，係數為 $\beta = (10,8,8,10,12,2)$，誤差項 $\varepsilon_i \sim N(0, 0.48^2)$

函數四:

真實迴歸函數為一四次多項式 $f(x_i) = x_i^4 - 2 \times x_i^3 + 1.4 \times x_i^2 - 0.4 \times x_i + 0.0384$，誤差項 $\varepsilon_i \sim N(0, 0.003^2)$

函數五：

真實迴歸函數 B-Spline 函數，節點為 $t = (0.4,0.4,0.4,0.4,0.7)$，係數為

$\beta = (-1.78,-0.33,1.62,-5,5,1.85,-3,-0.35,1.39)$，誤差項 $\varepsilon_i \sim N(0,0.57^2)$

函數六：

真實迴歸函數 B-Spline 函數，節點為 $t = (0.3,0.3,0.3,0.3,0.7,0.8)$，係數為

$\beta = (2,5,4,20,4,16,10,8,5,4)$，誤差項 $\varepsilon_i \sim N(0,1.1^2)$

函數七：

真實迴歸函數 B-Spline 函數，節點為 $t = (0.1,0.5,0.5,0.5,0.5,0.7)$，係數為

$\beta = (-10,-5,-2,20,1,20,25,10,1,-1)$，誤差項 $\varepsilon_i \sim N(0,2.7^2)$

函數八：

真實迴歸函數 B-Spline 函數，節點為 $t = (0.2,0.5,0.5,0.5,0.5,0.8)$，係數為

$\beta = (-5,-3,-2,5,25,25,5,-2,-3,-5)$，誤差項 $\varepsilon_i \sim N(0,2.59^2)$

附錄二

下列各表為更改不同設定之下，GAPLS 法與 BIC 法所選出的平均節點數目:

表(十九) 下十組資料

函數 / 平均#knot	函數一	函數二	函數三	函數四
GAPLS	9.7	9.4	9.8	9.8
BIC	6.5	4.3	4.2	4.2
真實函數	*	3	2	*

表(二十) 下十組資料

函數 / 平均#knot	函數五	函數六	函數七	函數八
GAPLS	9.4	9.5	9.8	9.5
BIC	6.3	7.2	6.7	5.6
真實函數	5	6	6	6

表(二十一) 下十組資料且訊噪比為 4

函數 / 平均#knot	函數一	函數二	函數三	函數四
GAPLS	9.6	9.5	9.4	9.6
BIC	6.2	4.1	4.2	6.4
真實函數	*	3	2	*

表(二十二) 下十組資料且訊噪比為 4

函數 / 平均#knot	函數五	函數六	函數七	函數八
GAPLS	9.6	9.7	9.7	9.5
BIC	4.3	7.5	6.2	4.5
真實函數	5	6	6	6

表(二十三) 下十組資料且訊噪比為 4 且樣本數為 100

函數 / 平均#knot	函數一	函數二	函數三	函數四
GAPLS	9.6	9.1	9.2	9.4
BIC	5.6	2.6	2	1.8
真實函數	*	3	2	*

表(二十四) 下十組資料且訊噪比為 4 且樣本數為 100

函數 / 平均#knot	函數五	函數六	函數七	函數八
GAPLS	8.1	9.3	9	9.3
BIC	5	5.6	5.8	3.4
真實函數	5	6	6	6

參考文獻

[1] Tzee-Ming Huang . An adaptive knot selection method for regression splines via penalized minimum contrast estimation. National ChengChi University. Department. of Statistics. 2013.

[2] Huang, Tzee-Ming. "Convergence rates for posterior distributions and adaptive estimation." The Annals of Statistics 32.4 (2004): 1556-1593.

[3] Hardle, Wolfgang. Applied nonparametric regression. Vol. 27. Cambridge: Cambridge university press, 1990.

[4] Eubank, Randall L. Nonparametric regression and spline smoothing. CRC press, 1999.

[5] 何昕燁，一種基於 BIC 的 B-Spline 節點估計方式. 2012.

[6] T.A. Springer ，〈線性代數群〉 張瑞吉譯，1987.

若需要 GPALS 法的程式碼，請以 email 跟作者聯絡，以下是作者的信箱:

101354028@nccu.edu.tw