

串流大數據循序行為預測式分析與動態矯正

張維凱 吳秀陽

Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, Taiwan, ROC
weikaichang1018@gmail.com showyang@gms.ndhu.edu.tw

Abstract

隨著物聯網及雲端運算科技的快速發展，串流大數據分析成為熱門的研究，其中循序行為預測、行為異常偵測與異常行為矯正均為重要的議題。本論文提出一個動態矯正機制，其中包含循序行為分析預測與偏異判定方法，並假設以致動器(actuators)矯正偏異行為，提出三種規劃致動器的演算法，我們稱之為行動規劃演算法。分析預測與偏異判定使用循序行為的回顧資料以及前瞻資料來評估其對於常態行為模式的符合度與偏異可能性。當評估的偏異度超過可容許範圍時，透過行動規劃演算法規劃一系列致動器，引導目標物回到正軌。在行動規劃演算法上，透過不同矯正構想，分別提出尾隨矯正法、跳島式矯正法以及平行逼近矯正法來規劃出一系列矯正偏異循序行為的致動器。我們以加爾各答天氣資料以及模擬資料進行實驗，測試所提方法的效果、效能以及與 A*演算法的比較。實驗結果顯示，本論文所提的預測式偏異分析可以有效提早偵測偏異，超前啟動矯正機制。偏異行為在動態偏異矯正機制之下可以成功矯正回常態行為，並持續追蹤目標。最後三種動態偏異矯正的行為規劃演算法中，規劃一系列矯正致動器的時間上也有不錯的表現。

關鍵字: 串流大數據分析、循序行為分析、預測式偏異評估、偏異矯正、行動規劃演算法、致動器組合

I. 研究目的與方法

本篇論文主要研究循序行為預測式偏異偵測與動態矯正。首先，為了能因應更加廣泛的問題，我們需要延伸本實驗室學長對於空間時間序列行為模式辨識與追蹤之研究，將擁有高維度特徵的資料納入我們的討論範圍。在運用趨勢預測發展出的異常偵測技術下，提出一種動態調適(dynamic adjustment)機制。動態調適機制藉由良好的偵測技術可以識別觀察標的其行為模式是否正常或是發生偏異。偵測正常則繼續追蹤，若偵測偏異發生，則制定一系列的致動器(actuators)來矯正(correction)偏異標的，直到目標回到常態或是可接受的行為。

其中，發展異常偵測技術後，本論文定義行動規劃問題以及行動規劃問題中所有可接受的解。在模擬出複數個合理的致動器後，依照上方的定義，按照不同思路提出三種行動規劃演算法解決問題。

II. 相關研究

近年來，時間序列分析應用在許多方面如醫學、工業、智慧物聯網以及資料探勘任務。為了從大量時間序列中找到有價值的資訊，而發展出許多研究領域包括分群(Clustering)，分類(Classification)、異常偵測(Anomaly detection)等等[1]，其中相似度測量(Similarity measure)扮演一項很重要的角色。在點對點的相似度測量中，很容易因為異常值或是比較不同的點序而導致計算結果不盡理想。在2002年 M.Vlachos 和 G.Kollios 提出使用改良後的最長公共子序列(Longest Common Subsequence, LCSS)的方法來計算軌跡的相似度，克服因異常值或不同長度所造成的影響[2]。在 M.Vlachos 和 G.Kollios 提出的論文中，演算法利用兩個閾值(threshold)分別約束兩序列中配對序列點的距離 ϵ 以及可以被配對的差距 δ ，如果配對序列點的差距小於或等於 ϵ ，則將相似度分數加上一個單位。如果大於 ϵ ，則不給予分數。當所有序列點都配對完後就可以加總出一個相似度分數。該演算法可以根據使用者自訂閾值 ϵ 使配對序列點更加彈性並避開異常值。透過使序列延伸並允許某些序列點不匹配，使能夠有效率地計算近似值。改良後的 LCSS 解決了異常值造成的誤差，但是要得到理想的結果之前，還需要依靠一組適合的參數，而在不同的應用中需要使用不同的參數，參數的設定大大的影響實驗結果的好壞。為了調整參數閾值這種斷崖式分數給予的影響，2020年 G. Soleimany, M. Abessiy 提出 Developed Longest common Subsequence(DLCSS)方法，使用兩個相似性閾值 ϵ_1 和 ϵ_2 ，第一個相似性閾值 ϵ_1 用於識別兩個序列點的決定相似性(Definite Similarity)，第二個相似性閾值 ϵ_2 值用於檢測兩個序列點的條件相似性(Conditional Similarity)。如果兩個時間序列的兩個序列點之間的差的絕對值小於或等於決定相似性閾值 ϵ_1 ，則認為它們肯定是相似的，相似度分數加上一個單位值。如果這兩個序列點之間的差的絕對值大於且小於或等於條件相似性閾值 ϵ_2 ，則這兩個序列點可能相似，演算法會對分數做調整，並之前計算的相似度分數相加。如果這兩個序列點差的絕對值大於 ϵ_2 ，則認定它們為不相似。利用這個方法能夠使計算結果有更高的準確性並降低誤差[3]。

III. 循序行為預測式分析與動態矯正

A. 系統架構

我們將介紹整個動態調適的過程，包含預測式分析、規劃矯正致動器、執行已規劃致動器、判定是否回到常態、繼續追蹤等等，流程圖如下。整體機制開始前，有兩個部分會先進行方法以及參數的選擇。第一部分，預測式偏異偵測中對於回顧資料與前瞻資料權重計算方式、回顧時間長度 α 與前瞻時間長度 β 、對於回顧資料欲遞減倍數、對於前瞻資料欲遞增倍數，此處對應流程圖的上方。第二部分，在三種行動規劃演算法中選擇一個演算法來進行一系列致動器的規劃，此處對應流程圖的右方。動態矯正為一個不會停止的迴圈。首先，系統會接收觀察標的新的序列點資料，接著進入預測式偏異偵測機制，利用回顧資料來預測前瞻資料後，使用這些序列點來評估當下序列的偏異度。如果判定為沒有偏異則回到持續接收觀察標的新的序列點資料；如果判定發生偏異則進入行動規劃矯正的流程裡面。接下來，已選擇的行動規劃演算法將會制定一系列的致動器來進行矯正。執行完致動器後，使用改良後的相似度計算方法來判斷是否回到常軌，沒有則繼續下一次的規劃與矯正；如果判定回到常軌，則回到持續接收觀察標的新的序列點資料，繼續追蹤，直到判斷下次偏異的發生。

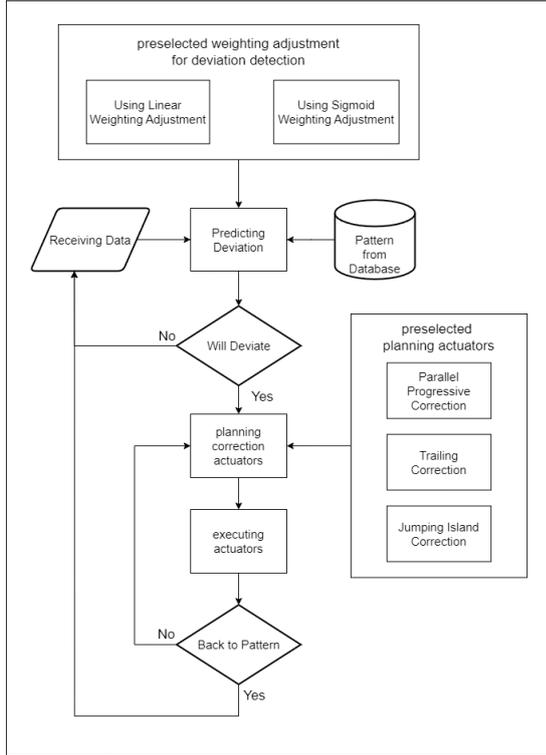


圖 1 動態矯正流程圖

B. 相似度計算方式

(Spatial and Temporal Longest Common SubSequence, STLCS)的定義[4]。假設經過資料探勘後已知 k 組有順序的序列點樣式(簡稱序列樣式) S_1, S_2, \dots, S_k ，和目前未

知觀測行為序列 O 如下。其中第 i 個序列樣式的第 j 個序列點 $S_{ij} = (N, t)$ 由多個向量組成，分別為序列點狀態 $N = (L, F)$ 以及時間戳記 t 。在每個序列點狀態中，有地理位置向量以及序列點特徵向量，假設地理位置向量為一個二維特徵，序列點特徵向量有 g 個特徵，則 $N \cdot L = (x, y)$ 、 $N \cdot F = (f_1, f_2, \dots, f_g)$ ，以及序列點發生時間 t 。我們需要在 S_1, S_2, \dots, S_k 中找到與 O 最近似的序列作為其可能繼續依照行動之序列樣式。

$$S_1 = \{((L_{11}, F_{11}), t_{11}), ((L_{12}, F_{12}), t_{12}), \dots, ((L_{1n}, F_{1n}), t_{1n})\}$$

$$S_2 = \{((L_{21}, F_{21}), t_{21}), ((L_{22}, F_{22}), t_{22}), \dots, ((L_{2m}, F_{2m}), t_{2m})\}$$

$$S_k = \{((L_{k1}, F_{k1}), t_{k1}), ((L_{k2}, F_{k2}), t_{k2}), \dots, ((L_{kr}, F_{kr}), t_{kr})\}$$

和

$$O = \{((L_{O1}, F_{O1}), t_{O1}), ((L_{O2}, F_{O2}), t_{O2}), \dots, ((L_{Oc}, F_{Oc}), t_{Oc})\}$$

除了兼顧多維度的特徵，ST-LCSS還具備可調整式時空誤差容許範圍之設定。假設我們有兩個序列樣式 S_p 與 S_q 分別長度為 n 和 m 如下，Definition 1 則是延伸後的 STLCS 分數計算公式。

$$S_p = \{((L_{p1}, F_{p1}), t_{p1}), ((L_{p2}, F_{p2}), t_{p2}), \dots, ((L_{pn}, F_{pn}), t_{pn})\}$$

$$S_q = \{((L_{q1}, F_{q1}), t_{q1}), ((L_{q2}, F_{q2}), t_{q2}), \dots, ((L_{qm}, F_{qm}), t_{qm})\}$$

$$S_{pi} = ((L_{pi}, F_{pi}), t_{pi}) \in S_p, S_{qj} = ((L_{qj}, F_{qj}), t_{qj}) \in S_q,$$

$$1 \leq i \leq n, 1 \leq j \leq m$$

Definition 1 給定 t 、 l 和 δ 對分別為匹配時間、地理位置和點序等差異容忍度，給定和兩個向量匹配距離閾值，以及來決定點匹配分數與點發生時間的權重，我們定義了兩序列的匹配分數 $STLCSS_{t,l,\delta,\epsilon_1,\epsilon_2,w_1,w_2}(S_p, S_q)$ ，表示成 $M(S_p, S_q)$ ：

$$M(S_p, S_q) = \begin{cases} 0, & \text{if } S_p = \emptyset \text{ or } S_q = \emptyset \\ \max \left\{ \begin{array}{l} local + M(S_p \setminus S_{pn}, S_q \setminus S_{qm}) \\ M(S_p, S_q \setminus S_{qm}) \\ M(S_p \setminus S_{pn}, S_q) \end{array} \right\}, & \text{if } d(S_{pn}, N, S_{qm}, N) \leq \epsilon_2, |S_{pn}, t - S_{qm}, t| \leq t, \\ & \|S_{pn}, L - S_{qm}, L\| \leq l, |n - m| \leq \delta \\ \max \left\{ M(S_p, S_q \setminus S_{qm}) \right\}, & \text{if } \epsilon_2 < d(S_{pn}, N, S_{qm}, N) \end{cases}$$

其中，

$$local = w_1 \cdot a + w_2 \cdot \left(1 - \frac{|S_{pn}, t - S_{qm}, t|}{T_{max}}\right),$$

$$a = \begin{cases} 1, & \text{if } d(S_{pn}, N, S_{qm}, N) \leq \epsilon_1 \\ \frac{\epsilon_2 - d(S_{pn}, N, S_{qm}, N)}{\epsilon_2 - \epsilon_1}, & \text{if } \epsilon_1 < d(S_{pn}, N, S_{qm}, N) \leq \epsilon_2 \end{cases},$$

$w_1 + w_2 = 1$ ， $0 \leq w_1, w_2 \leq 1$ ， $\epsilon_1 < \epsilon_2$ ， $S_i \setminus S_{in}$ 為整個 S_i 去除第 S_{in} 項， T_{max} 為時間差距最大量， $d(S_{pn}, N, S_{qm}, N)$ 為特徵向量距離。

常數 δ 控制了兩個時間序列匹配的延伸度，而 t 和 l 分別對時間序的時間、地理位置的比對進行約束。當比對兩序列點之距離小於 ϵ_1 (絕對相似度閾值) 並且時間和空間都在限制內，會給予其一個單位的分數並與兩序列點發生之時間做分數調整。而兩序列點之距離大於 ϵ_1 但小於 ϵ_2 (條件相似度閾值)，我們則依照相似的比例計算相似度得分，然後也會依照兩序列點發生之時間做分數調整。如果兩序列點之距離大於 ϵ_2 ，我們則不給予得分。其中權重 w_1, w_2 能會對兩序列點發生之時間

做分數調整，如果使用者對於兩序列點發生之時間較不注重，則可以升高 w_1 降低 w_2 ，反之亦然。

Definition 2 當給定 $t, l, \delta, \varepsilon_1, \varepsilon_2, w_1, w_2$ 以及 α_1, α_2 。 α_1 和 α_2 為決定對兩序列總匹配分數與兩序列整體行為時間長度的權重，同時對於總匹配分數我們也考慮到兩序列的長度差距，為此序列和之間的相似度的計算如下：

$$Sim_{t,l,\delta,\varepsilon_1,\varepsilon_2,w_1,w_2,\alpha_1,\alpha_2}(S_p, S_q) = \alpha_1 \cdot \frac{M(S_p, S_q)}{\min(n, m)} \cdot \left(1 - \frac{|n-m|}{\max(n, m)}\right) + \alpha_2 \cdot \frac{|(S_{pn} \cdot t - S_{p1} \cdot t) - (S_{qm} \cdot t - S_{q1} \cdot t)|}{T_{max}}$$

其中， $\alpha_1 + \alpha_2 = 1$ ， $t, l, \delta, \varepsilon_1, \varepsilon_2, w_1, w_2$ 如 Definition 1，使用者給定 α_1, α_2 ， $0 \leq \alpha_1, \alpha_2 \leq 1$ ，而使用者可以依照兩序列整體行為時間長度的注重程度，對權重做調整。

C. 預測式偏異偵測機制

基於上一章相似度的定義，我們發展出一套判斷是否發生偏異的機制。假設目標物為動態行進，我們會將觀察標的從進入偵測範圍到當下為止的觀測行為序列與所有常態行為之前綴(prefix)比較，找出目前最接近且相似度差異在容許範圍內的常模，作為該目標物件下的行為判定基礎。因為目標物件不一定依照常模運作，所以我們會依目前走勢用回歸分析預測其未來動向。

目前觀測行為為 S_q ，選擇比較的常模為 S_p ，我們以當下($T=t_{p(c_p)}$)，對應 S_q 的時間戳記為 $t_{q(c_q)}$ 為基準點，給定回顧時間長度為 α 的回顧行為作為穩定因子(stabilizer)，來穩定我們的偵測機制使其不會過度敏感，以及前瞻時間長度為 β 的前瞻行為來判定目標物件行為是否會發生偏異，概念如下。

由於對距離現在越久遠的資料較不注重，所以會給予遞減的權重。預測資料的權重上，考慮到距離現在越往後的偏離可能越難導，所以給予遞增的權重。藉由兩序列點的向量距離給定分數並與其權重做調整後，將所有分數加總並標準化，就可以評估目前觀測行為為 S_q 與常模 S_p 的差異度，藉由此差異度來判斷是否發生偏異。

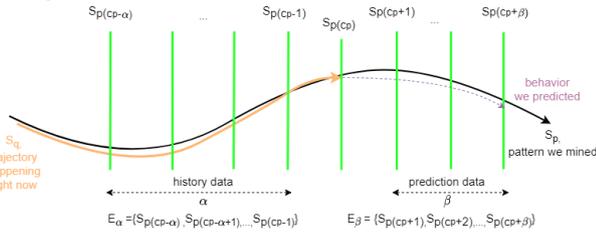


圖 2 偵測機制示意圖

D. 回顧資料與前瞻資料的權重計算

權重給定的想法如上方敘述，對於距離現在已發生且越久遠的資料較不注重，所以給予較低的權重。對於預測，由於越往後的偏離有可能越難矯正，所以對於越遠的權重會較高。除了使用線性的方式(Definition 5)給予權重外，由於靠近當下時間點的序列點有著穩定因子的功能，所以我們認為對於靠近當下時間點的權重可

以先有小幅度的變化，之後在隨著時間再有大規模的變化(Definition 6)。

Definition 3 線性調整權重指數

定義序列 S_p 在時間點 $t_{p(c_p)}$ 和正在觀察標的行為 S_q 在時間點 $t_{q(c_q)}$ 的偏異度評估。假設有興趣的回顧行為長度為 α ，如果超過已發生行為的長度則將設為已發生行為的長度。有興趣的前瞻行為長度為 β ，如果前瞻行為長度大於欲比較的剩下常模長度，則使用欲比較的剩下常模長度為，目前假設對於回顧資料最大降幅為 b^{-1} 、對於前瞻資料最大增幅為 f ，偏異度評估公式如下：

$$Score = \frac{\sum_{i=1}^{\alpha} u_i \cdot S_{p(c_p-i)} + 1 \cdot S_{p(c_p)} + \sum_{i=1}^{\beta} v_i \cdot S_{p(c_p+i)}}{\sum_{i=1}^{\alpha} u_i + 1 + \sum_{i=1}^{\beta} v_i}$$

其中，

$$S_{p(c_p+i)} = \begin{cases} 1, & \text{if } d(S_{p(c_p+i)}, N, S_{q(c_q+i)}, N) \leq \varepsilon_1 \\ \frac{\varepsilon_2 - d(S_{p(c_p+i)}, N, S_{q(c_q+i)}, N)}{\varepsilon_2 - \varepsilon_1}, & \text{if } \varepsilon_1 < d(S_{p(c_p+i)}, N, S_{q(c_q+i)}, N) \leq \varepsilon_2 \\ 0, & \text{if } \varepsilon_2 < d(S_{p(c_p+i)}, N, S_{q(c_q+i)}, N) \end{cases}$$

$j \in \{-\alpha, -\alpha + 1, \dots, \beta\} \in Z$ ， $u_i = b^{-\frac{i}{\alpha}}$ ， $v_i = f^{\frac{i}{\beta}}$ 。

Definition 4 S 型函數(sigmoid function)作為權重指數：

我們使用 S 型函數的值來當作權重的指數，S 型函數的極值趨近於 0 或 1，當其指數為正負 6 時，可以涵蓋到大約 99.5% 的值域，我們認為這個涵蓋範圍是足夠的。對於回顧行為來說，我們需要將線性插值到 [0, 6] 之間。此時回顧資料權重的值域為，然而我們想要使減幅介於，所以還會對 S 型函數做縮放平移處理，使縮放平移到。對於前瞻資料來說，我們則是將線性插值到 [-6, 0] 之間。此時前瞻時間權重的值域為 (0, 0.5]，然而我們想要使減幅介於，所以還會對 S 型函數做縮放處理，使 (0, 0.5] 縮放到 (0, 1]，最後權重如下：

$$u_i = b^{2 * \left(\frac{1}{1 + e^{-\left(\frac{\alpha-i}{6 * \frac{\alpha-1}{\alpha-1}}\right)}} - 1\right)}, \quad v_i = f^{2 * \left(\frac{1}{1 + e^{\left(\frac{\beta-i}{6 * \frac{\beta-1}{\beta-1}}\right)}}\right)}$$

IV. 動態偏異矯正之行動規劃演算法

A. 偏異矯正與行動規畫定義

確認觀察標的行為已發生偏異，下步便是設定欲矯正至目標序列 T 以及制定一動作序列的調適計畫 π ，我們有 d 個致動器(actuators)表示為集合 $A = \{a_1, a_2, \dots, a_d\}$ 。每個致動器會有四個組成，分別為執行後對標的行為特徵之改變(假設一共有 g 個特徵)、對環境特徵改變(假設一共有 h 個特徵)、執行此致動器之資源成本以及執行此動作之時間，所以表示成 $a_i = (\overline{\Delta f_S}; \overline{\Delta f_R}; cost; time)$ 。我們假設觀察標的行為序列為 S_q ，其應延續常模為 S_p ，確認發生偏移的當下狀態點為 $S_{q(c_q)}$ ，對應到應延續常模的狀態點為 $S_{p(c_p)}$ ，我們取之後所有點(假設一共有 w 個)同時包含 $S_{p(c_p)}$ 作為欲矯正至的目標序列 $T = [S_{p(c_p)}, S_{p(c_p+1)}, \dots, S_{p(c_p+w)}]$ 。

Definition 5 偵測到標的行為已發生偏移當時的狀態點為 $S_{q(c_q)}$ 、欲矯正至目標序列 T ，定義一個規劃問題

(planning problem) P ，問題的目標是找到一長度為 k 的致動器序列 $\pi = [a_1, a_2, \dots, a_k]$ ，使標的行為經過一序列的動作後的狀態 $S_\pi = [S_{q(c_q)}, S_{q(c_q+1)}, \dots, S_{q(c_q+k)}]$ ，接近目標序列一定程度的任意子序列 $G = [S_{p(c_p+a)}, S_{p(c_p+b)}, \dots, S_{p(c_p+w)}]$ ：

$$P = (\Sigma, S_{q(c_q)}, T)$$

- $\Sigma = (S, R, A, \gamma)$ 為一個變遷系統
- $S = \{s_1, s_2, \dots\}$: 有限或可遞迴枚舉的行為集合
- $R = \{r_1, r_2, \dots\}$: 有限或可遞迴枚舉的環境狀態集合
- $A = \{a_1, a_2, \dots\}$: 有限或可遞迴枚舉的致動器集合
- $\gamma = (S \times R \times A_{\text{applicable}}) \rightarrow (S, R)$: 狀態遷移函數
- $A_{\text{applicable}} = \{a \mid a \in A \text{ and } a \models \gamma.S \text{ and } a \models \gamma.R\}$: 符合觀察標的行為、環境狀態，可採用之致動器集合
- \models : 符合(satisfies)

Definition 6 當有一個規劃問題發生，動作序列 π 的長度為 k 、標的行為當下狀態點為 $S_{q(c_q)}$ ， $S_{q(c_q)}$ 經過某一動作序列 π 後將會產生一序列的狀態 S_π ，假設一個可接受的調整後標的行為序列與目標調整序列距離閾值 ϵ_{corr} ，任何動作序列 π 可以形成的 S_π 其距離與欲矯正至目標序列 T 的任意子序列 G 只要小於 ϵ_{corr} 均可以視為可接受的解，並且定義 ζ 為所有可接受解的集合：

$$\zeta = \{\pi \mid \forall G \subseteq T, \exists \pi \text{ s.t. } dist(S_\pi, G) < \epsilon_{corr}\}$$

$$\pi = [a_1, a_2, \dots, a_k]$$

$$S_\pi = [S_{q(c_q)}, S_{q(c_q+1)}, \dots, S_{q(c_q+k)}]$$

B. 尾隨矯正法

對於觀察標的發生偏異行為，若想要選擇一序列的致動器做矯正，最直觀也是最簡單的选择策略，就是每一次都選擇可以矯正最多的致動器，示意圖如下。

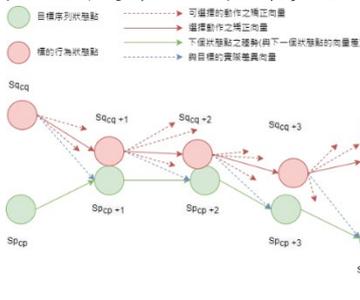


圖 3 尾隨矯正法示意圖

C. 跳島式矯正法

跳島式矯正法 (Jumping Island Correction, JIC) 構想來源是來自尾隨矯正法的問題。使用尾隨矯正法時，有些欲矯正至目標序列有著大幅度的變化，導致矯正時發生矯正效果追不上目標序列，結果發生越修正差距越大的情狀，或是花費大量沒有效率的致動器在追逐與欲矯正至目標序列點上的差異。所以跳島式矯正法會先找到欲矯正至目標序列的數值趨勢或是軌道趨勢，找尋是否有反曲點的發生，並將矯正目標設定在反曲點的下一點，不但可以避免追不上目標的情況發生，也不會花費大量沒有效率的致動器在追趕目標，示意圖如下。

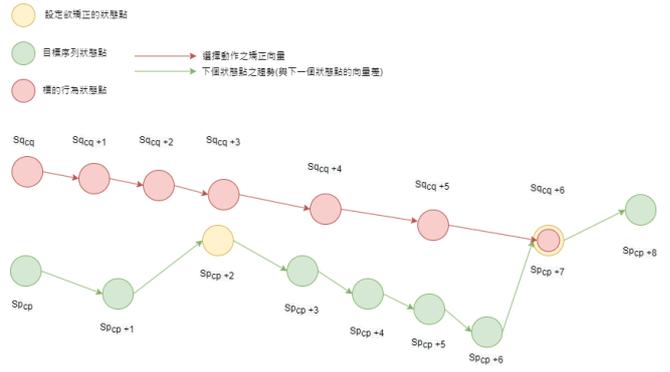


圖 4 跳島式矯正法示意圖

D. 平行逼近矯正法

平行逼近矯正法 (Parallel Progressive Correction, PPC) 是先找到單一差距最大的特徵作為欲矯正的估計特徵，然後演算法會從可以執行的致動器中，找到最能矯正此特徵的致動器來做矯正次數的估計，依照矯正次數規劃出每次矯正的目標矯正角度，再從致動器集合中選擇符合序列狀態、環境狀態、執行時間以及目標修正角度的致動器，從而逐漸逼近目標序列，示意圖如下

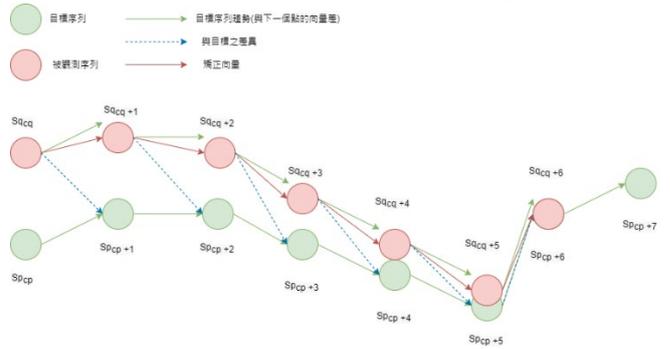


圖 5 平行逼近矯正法示意圖

V. 系統實作與效能評估

在實作方面我們選擇 Python 作為實作語言，由於資料結構使用 python 的 dictionary，所以使用版本 3.9。

表 1 系統環境設置

Software			
Python 3.9.13			
Desktop			
CPU	cores	Memory	GPU
AMD R5 3600	6	96 GB	NVIDIA RTX2070

A. 實驗資料與模擬方法

本研究使用真實與模擬資料進行實驗，實驗大致分為效果實驗、效能實驗與比較性實驗。效果實驗使用真實資料，其他實驗使用模擬資料。真實資料為加爾各答氣象資料集，以此來模擬串流資料，資料紀錄時間從 2015 年 1 月到 2020 年 12 月，該資料集包括時間、氣溫、濕度的資料樣本，資料集總共包含 52609 筆資料 [5]。模擬資料中，由於沒有資料集可以彈性調整特徵數，所以隨機產生特徵以及值為隨機 1-100 的方式來進行測試。

在預處理時會先找出每個特徵最大最小值以及單位時間的最大增減幅，每次偏異的值以及偏異的增減幅都會依照過去資料的最大最小值以及最大增減幅而有所限制。致動器上，會先假設已經有一組可以分別對每個特徵調整正負1的致動器。其餘產生致動器會隨機選擇對每個特徵有正負一個標準差的影響值、使用成本為隨機在1到10中選擇、執行時間為0.1到2小時的一個常態分佈。

B. 演算法矯正效果

在演算法矯正效果的實驗中，我們要測試本論文中提出的三種演算法是否能成功將被偵測到偏異的觀察標的矯正回常模。藍線為加爾各答的氣象資料的一個常模；橘線為觀察標的行為序列；紅線為矯正結果。我們的偵測機制在07-01 04:00判定行為發生偏異，所以啟動行動規劃演算法來規劃出一系列的致動器來矯正偏異行為。下面分別為三種演算法執行各自所制定的致動器後的矯正結果，左邊為溫度特徵行為、右邊為濕度特徵行為。可以看到三種演算法均成功將偏異觀察標的矯正回目標序列一定程度之任意子序列。

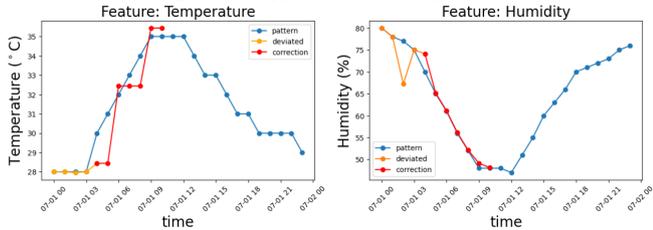


圖 6 尾隨矯正法在各維度的矯正效果

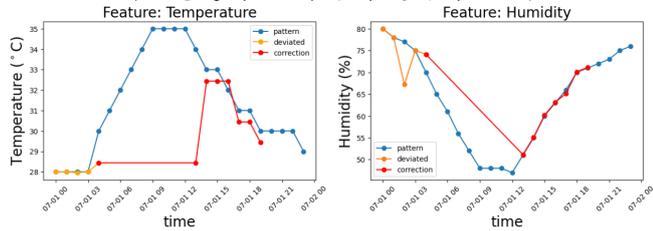


圖 7 跳島式矯正法在各維度的矯正效果

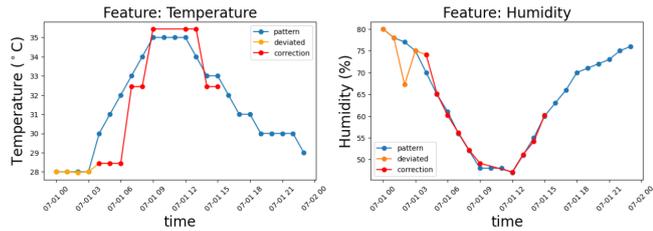


圖 8 平行逼近矯正法在各維度的矯正效果

C. 預測式偏異偵測使用效果

下方實驗，藍色線條為觀察標的被辨識的常模、橘色線條為觀察標的行為軌跡。偵測機制在01-11 03時判定會發生偏異，紅色線條為使用偵測機制的矯正行為，對比於沒有使用預測式偵測，不使用預測式偏異偵測的行為會到01-11 08時被發現偏異的發生，紫色線為不使用偵測機制的矯正行為。

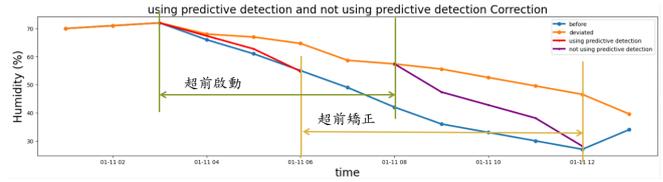


圖 9 使用預測式偏異偵測的提前矯正效果

圖 10 展示出，使用預測式偏異機制後可以超前啟動的時間。為了合理的比較差異，所以使用“標準化時間”，就是所有可以選擇致動器的執行時間的平均時間，來當比較基準。可以看到尾隨矯正法可以有15倍的標準化時間；跳島式矯正法可以有13倍的標準化時間；平行逼近矯正法可以有14倍的標準化時間。

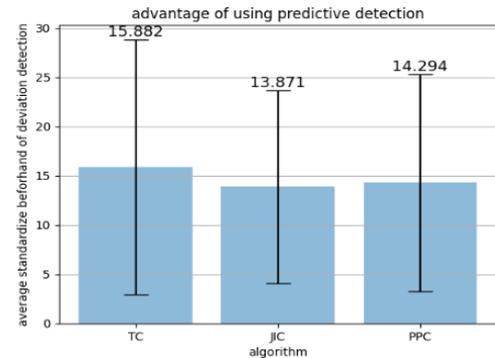


圖 10 使用預測式偏異偵測後，可以超前啟動的時間

F. 演算法在不同特徵維度的比較

在這個實驗中我們想要測試本論文提出的三種演算法在不同維度數量下的表現，實驗前會設定總特徵維度個數，接下來隨機產生這輪實驗的地理位置維度數以及特徵維度數，而每個地理位置以及每個特徵將為隨機1-100的值。實驗結果顯示當維度超過60時，跳島式矯正法的規劃時間將會快速上升，而其他兩個演算法則是逐漸緩慢的上升。而矯正長度也是如此。

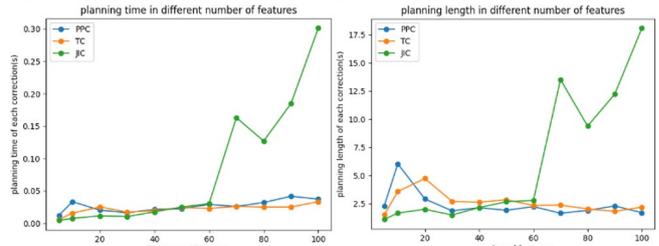


圖 11 三種演算法在不同維度上規劃時間與矯正長度

G. 演算法在不同使用者數量的比較

在這個實驗中，我們想透過實驗測試出三種行動規劃的演算法在多個使用者都需要同時計算的情況下是否會有太大的影響，我們認為在規劃每次矯正的時間上是隨使用者的數量呈現線性成長而不是指數成長。

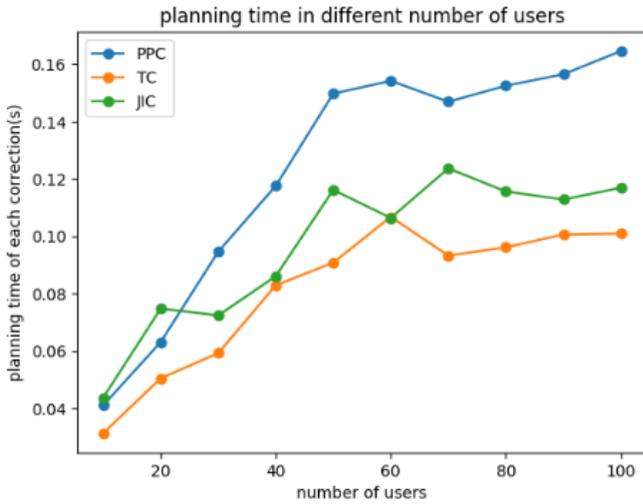


圖 12 三種演算法在不同數量使用者的規劃時間

H. 演算法與 A* 演算法的比較

在這個實驗中，我們想要測試本論文提出三種演算法的規劃時間以及矯正長度，是否與 A* 演算法有差異。實驗方式均使用同一組可選擇的致動器組來選擇，在模擬行為後，我們會偵測出偏異行為的發生，依照不同演算法選擇的矯正點來當作 A* 演算法的終點，當 A* 演算法找出的經致動器矯正後的序列與欲矯正的目標序列足夠接近時停止找尋。我們的演算法均可以有較快速的規劃時間，但是在矯正長度上能沒有 A* 演算法的精簡有效。

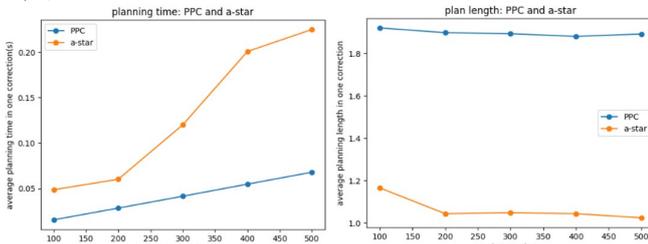


圖 13 平行逼近矯正演算法與 A* 演算法比較

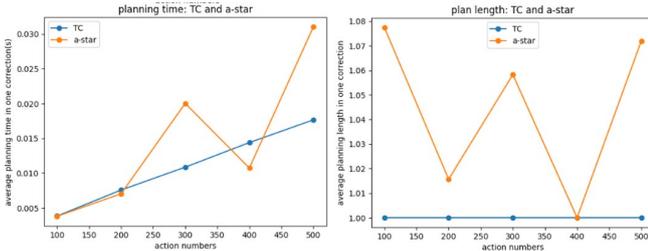


圖 14 尾隨矯正演算法與 A* 演算法比較

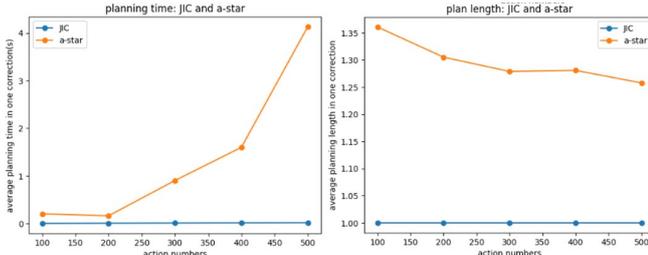


圖 15 跳島式矯正演算法與 A* 演算法比較

VI. 結論與未來工作

在傳統循序行為模式分析與異常偵測中，當異常行為發生後才會進行通知與矯正，導致偏異早已發生需要大量的致動器來矯正發生偏異的觀察標的。在本篇論文中提出的預測性偏異偵測與三種矯正演算法可以有效避免上述的缺失。最後我們可以結論如下：

- (1) 預測式偏異偵測具有可行性並有不錯的成果。
- (2) 動態偏異矯正以及行動規劃演算法可以在循序序列上有不錯的效果。
- (3) 對於行動規劃演算法來說，儘管規劃出的致動器沒有 A* 演算法所規劃的有效率，但是對有著更加快速的優勢。

本研究能有許多改善空間以及相關研究議題：

- (1) 使用人工智慧或是統計方法修改預測式的偏異分析
- (2) 三種演算法在尋找致動器上可以更有效率。
- (3) 循序行為的特徵可以有不一樣的權重以及使用其他方式測量兩序列點特徵的距離

REFERENCES

- [1] Z. Feng and Y. Zhu, "A Survey on Trajectory Data Mining: Techniques and Applications," in *IEEE Access*, vol. 4, pp. 2056-2067, 2016.
- [2] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," *Proceedings 18th International Conference on Data Engineering*, pp. 673-684, 2002.
- [3] Soleimani, Gholamreza & Abessi, Masoud, "DLCSS: A new similarity measure for time series data mining." *Engineering Applications of Artificial Intelligence*. 92. vol 14. 2020.
- [4] 陳昱衡(2020)，以整合式形狀與符號聚合近似法為基礎之時間序列資料分析與行為辨識。國立東華大學資訊工程學系碩士論文，花蓮市。
- [5] Suman Dey, Kaggle Temperature and Humidity of Kolkata from 2015-2020 <https://www.kaggle.com/datasets/sumandey/temperature-and-humidity-of-kolkata-from-20152020> (參照日期 2022/09/20)