應用機器學習演算法建立高維度非線性剖面資料之監控

組別/編號：C1_1　　學生　：汪志宸、林胤霆、呂彥澐　　指導老師：鄭春生

研究動機與目的

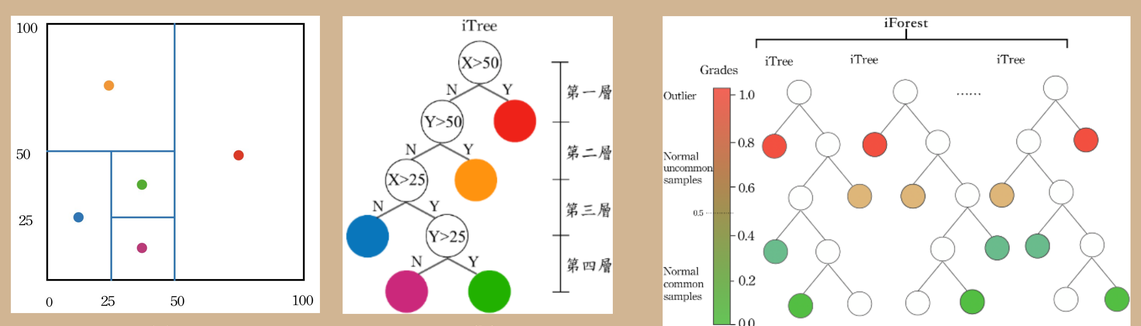
剖面監控中分為兩個階段步驟，階段 I (phase I) 的目的是選擇合適的模型以及提供受控模型參數的估計，又稱為離群值檢測 (outlier detection)。本研究著重於階段 I，目的在於使用機器學習來進行剖面資料監測。使用機器學習演算法—隔離森林(isolation forest, iForest) 及區域性離群因子 (local outlier factor, LOF) 找出樣本中離群值，並比較出最佳績效，進而討論最適合的方法。

研究方法

本研究共進行兩組實驗，利用 B-spline 擬合剖面資料以降低雜訊，另外針對 LOF 演算法使用主成分分析(principal component analysis, PCA) 降低資料維度並建立 iForest 及 LOF 兩種異常檢測模型，並評估其績效。

• 隔離森林

Fig1 (a) 為樣本數據和 Fig1 (b) 為一棵 iTree 之示意圖。iTree 的組成是利用選擇多次特徵值劃分樣本數據進而形成如 Fig1 (b)，特徵值可能是多個。舉例來說第一條件為 $X > 50$，在此條件下有一離群值 (Fig1 的紅點) 被隔離出，此時 iTree 的深度為第一層 (Fig1(b) 紅點位置)，接著再次隨機選擇第二個條件 $Y > 50$ 進行劃分，以此類推直到劃分結果為每個子空間都只有一個點，即為劃分結束。以上為一棵 iTree 之離群值劃分，在一數據集當中會生成許多的 iTree，多棵 iTree 的組成可形成隔離森林，也稱為 iForest (Fig 2)。



(a)　(b)

Fig1. iTree 原理示意圖

Fig2. iForest 示意圖

• 區域性離群

LOF 演算法不需要要求資料的分布，並且還能量化每個資料點的異常程度。在 LOF 中需要計算 k-distance，k-distance 內的資料點稱為 k-nearest-neighbor，被記為 $N_k(p)$，接著計算可達距離 (reachablity distance)，Fig 3 表示P1 資料點及 P2 資料點的可達距離皆相同。

區域性可達密度 (local reachablity density) 定義為資料點 p 的區域性可達密度為鄰近的資料點的平均可達距離之倒數。由此公式計算出的值表示密度，密度越高資料點越密集，越低則代表越有可能為離群值。

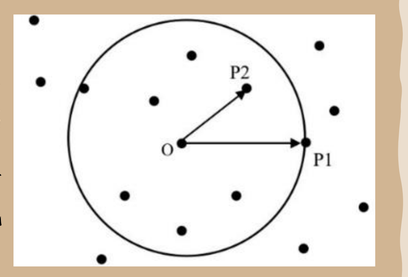


Fig3. 可達距離示意圖

$$lrdk_k(p) = 1/(\frac{\sum_{o \in Nk(p)} reach - dist_k(p,o)}{|N_k(p)|})$$

• B-spline

B-spline 是樣條 (spline) 曲線一種特殊的表示形式。樣條是一種特殊的函數，由多項式分段定義。一條曲折多且長的 B-spline 曲線，會配有大量的控制點數目，以控制曲線扭曲的狀況。

• 主成分分析

PCA可以廣泛運用於分析資料、降低數據維度以及去關聯的線性降維方法。PCA 主要之目的是將有相關的變數簡化成少數幾個沒有相關的主成分，也就是一種變數縮減的方式。

研究績效評估

機器學習中，常使用混淆矩陣 (confusion matrix) 來判斷該模型的表現，其中二元分類 (binary classification) 又是最為常見的，其目的是分析出預測結果與實際結果是否有相對應，分成四類，預測為真、假，實際為真/假，分類組合後如 Tab.1 所示。

Tab1. 混淆矩陣

	預測為真	預測為假
實際為真	True Positive (TP)	False Negative (FN)
實際為假	False Positive (FP)	True Negative (TN)

本研究透過混淆矩陣來進行績效評估，以各個模型之分類結果的數量來進一步計算其 F_2 分數 (F_2 score)。 F_2 分數越大代表分類模型越佳，藉此比較各個模型的績效。

$$F_2\ score = \frac{5TP}{5TP + FP + FN}$$

研究績效與結果

實驗資料來源皆引用過去學者文獻所提出之剖面方程式來產生模擬資料集。

實驗一引用 Nie et al. (2020) 使用懲罰迴歸方法在一般剖面檢測異常值中所驗算出的方程式。

$$f_a(x) = 10 - \frac{20ae^{-ax_j} sin(\sqrt{4-a^2}x_j)}{\sqrt{4-a^2}} + 10e^{-ax_j} \times cos\left(\sqrt{4-a^2}x_j\right) + \varepsilon$$

其中 $x \in \{0.08, 8\}$，總共劃分為 100 個共變異數，以 0.08 為一間隔均勻分布，正常剖面 $a = 0.5$、$\sigma = 1$。ε 為剖面的雜訊區段且符合常態分配。

實驗二引用 John *et al.* (2019) 文獻中提出的剖面方程式

$$\bar{x} = 46.962 + 7.472t + 0.521t^2 + 0.013t^3 - 0.013h_1(t) - 0.018h_2(t)$$

其中

$$h_1(t) = \begin{cases} 0, & if\ t \le -14 \\ (t-(-14))^3, & if\ t > -14 \end{cases}$$

$$h_2(t) = \begin{cases} 0, & if\ t \le 14 \\ (t-14)^3, & if\ t > 14 \end{cases}$$

$t \in \{-21, 21\}$ 劃分為 43 個變數，設定偏移量 0.8 為異常剖面，因為在偏移量 0.8 時演算法找出異常資料最為困難，所以為實驗對象為 0.8。

iForest 演算法採用 python 中的 scikit-learn 套件所建立，在實驗一時考慮三種汙染率 (contamination)，以及五種 α。在實驗過程中發現 iTree 為 500 能達到最低的型 I 及型 II 錯誤率， Tab2 為 iTree 在 500 時的實驗一之 F_2 分數。

在實驗二一樣考慮三種汙染率，但只計算 0.8 偏移量，並且發現 iTree 為 500 時能達到最低的型 I 及型 II 錯誤率，Tab3 為 iTree 在 500 時的實驗二之 F_2 分數。

Tab2. 實驗一之 F_2 分數 (iForest)

	α				
汙染率	0.7	0.9	1.1	1.3	1.5
0.1	0.7255	0.904	0.9435	0.967	0.9805
0.2	0.6275	0.7775	0.824	0.8485	0.8273
0.3	0.5678	0.6625	0.6972	0.7057	0.7108

Tab3. 實驗二之 F_2 分數 (iForest)

	偏移量
汙染率	0.8
0.1	0.7730
0.2	0.6318
0.3	0.5605

本研究 LOF 演算法亦是採用python 的 scikit-learn 套件，並搜尋參數 k (n_neighbors) 之最佳設定，但因為 LOF 並無演算法隨機性之影響，因此每組資料只進行 1 次試驗。 Tab4 和 Tab5 為結果分數。

Tab2. 實驗一之 F_2 分數 (LOF)

	α				
汙染率	0.7	0.9	1.1	1.3	1.5
0.1	0.99	1	1	1	1
0.2	0.9835	1	1	1	1
0.3	0.9776	1	1	1	1

Tab3. 實驗二之 F_2 分數 (LOF)

	偏移量
汙染率	0.8
0.1	0.9650
0.2	0.9730
0.3	0.9620

結論

研究結果顯示，利用 B-spline 擬合剖面資料降低雜訊，及針對 LOF 演算法使用 PCA 以降低資料維度，建立異常檢測模型，可以得到相當高之績效表現。透過 F_2 分數比較 iForest 與 LOF 兩種演算法，不論是實驗一或二，LOF 之績效表現明顯高於 iForest。因此，本研究所提出之 LOF 方法可以作為階段 I 異常剖面之檢測演算法。