國立臺北大學資訊工程學系專題報告

如何找出洗錢帳戶？

專題組員：甘岱融、盧建勳、邱彥程、許元碩

專題編號：PRJ-NTPUCSIE-106-04

執行期間：106 年 9 月至 107 年 5 月

1.摘要

當今的社會，人心不古，有心人士透過不正當的行為來滿足貪婪的慾望。最廣為人知的有詐騙、洗錢、搶劫等，而本專題就是針對「洗錢」這一行為分析。因為洗錢行為與各項重大犯罪常具有共生結構，使其對社會經濟危害程度，遠大於傳統型態的犯罪。

2.簡介

在科技發達的時代，人們的生活品質隨之提升。舉例來說：以前人們出門便需要將錢隨時帶在身上，或是將自身的財產置於家中。而隨著時代的變革，現在人們可以憑藉一張卡便能遊走各地，財產亦能以非實體的形式保存。

然而這也孕育層出不窮的犯罪事件。有心人士為了增加自己的財產，利用法律以及系統的漏洞，透過將不當管道獲取的金錢以交易的形式來轉換成自身的財產，偽裝成普通收入。

眼看世界要捲入混沌的漩渦之中，我們與其他的小夥伴們決定躋身對抗惡勢力。於是我們希望透過本次的專題，分析洗錢的方式或其所需具備的條件，進而以洗錢者的角度去思考他們是如何犯罪，加以防制。

3.實作提要

3.1 confusion table

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive Power	False positive Type I error
	Predicted condition negative	False negative Type II error	True negative

圖(一)、confusion table 格式

- Accuracy：預測符合實際狀況/所有狀況。
- Precision:在預測為正確的情況下，實際確實為正確的比例。
- Recall：在實際為正確的情況下，預測正確的比例。
- F1 score：Precision 與 Recall 有時候會出現矛盾的情況，為了要綜合考慮兩者，最常見的方法就是 F-Measure(亦 稱 為 F-Score) 。 F-Measure 是 Precision 和 Recall 加權調和平均 ： $\frac{(\alpha^2+1)PR}{\alpha^2P+R}$。當$\alpha$=1 時，就是最常見的 F1-score。綜合前述兩者結果得出的數值，數值越高越有效。

3.2 前置作業

對於洗錢帳戶的分析，我們用了兩種不同的想法及方法實作，以下會分開詳述兩種方法的實作過程。我們利用的資料為各帳戶的匯款及受款基本資訊，根據資料的 attribute，我們篩選出以下四項 attribute 作為本次實作的參考項目：匯出匯入、帳戶、交易日期、交易金額。而

以下的兩種實作方法都將以此資料作為基礎去發想。

永豐銀行有給我們疑似洗錢的帳戶，是他們根據經驗法則篩選出來的。我們將此經驗法則的結果視為 Ground truth。另外他們也有提供一個簡易判斷是否為可疑帳戶的規則(詳細內容見後)，以下將此規則稱之為 rule base，視為 Predicted。而用 rule base(Predicted)預測經驗法則(Ground truth)的結果如下:

		Ground truth				
		Positive	Negative	Sum	accuracy	0.5378
Predicted	Positive	349	222	571	recall	0.0866
	Negative	3679	4190	7869	precision	0.6112
	Sum	4028	4412	8440	f1_score	0.1517

表(一)、規則法偵測可疑帳戶的準確度

如表(一)，Ground truth 的可疑帳戶約占總帳戶的 48%，Predicted 的佔 7%。發現兩者對於彼此預測的命中率並不高。兩者都視為可疑帳戶的有 349 筆，占 Predicted 可疑帳戶的 61%，占 Ground truth 可疑帳戶的僅 8.7%。

以下兩種實作方法，我們都會以此表做為比較參考，皆以 Ground truth 為標準答案，比較兩種方法與此的成效差異。

4.第一套方法（多層結構演算）

4.1 想法

在我們初步研究有關於洗錢的各項特徵時，有發現洗錢的其中一個特徵是「多層化(layer)」:洗錢者會利用多層複雜的交易使該筆金錢的來源難以追溯。因此若假設洗錢者利用銀行轉帳進行此步驟，只要我們將每筆交易的連結建立起來，或許可以發現某些規律。

4.2 目標

利用資料裡各筆的匯出匯入資料，嘗試建立出各筆交易中，款項匯出方及匯入方的連結。再以 Predicted 的可疑帳戶當標準，嘗試將可疑帳戶之間的連結串起來，找出完整的疑似洗錢路徑。期望能從中找出洗錢是否有固定的路徑模式，以及從 Predicted 中找出更可疑的帳戶。

4.3 資料前處理

4.3.1 找出帳戶間的交易 link

將匯出和匯入的資料分開，匯入的資料共有 339869 筆;匯出的資料共有 399273 筆。再比對兩份資料，若有一筆匯出的資料與另一筆匯入資料的金額及日期相等，則將兩者合併，建立成一個 link(視為該筆交易的匯出方及匯入方)。若有重複的交易資料符合上述條件，則會重複計算。而在 link 上的資料含有金額、時間、匯出及匯入方等資訊。

在此步驟，因為資料表中沒有更細節的時間資料，因此只能用這樣的方式建立 link。所建立的 link 不一定都是完全真實存在的，小金額的重複 link 會較多。但我們之後會篩選出較可疑的 link，可以將大部分小金額的重複 link 雜訊刪除，因此實作仍具可信度。最後共建立出 8145 條 link。

4.3.2 找出可疑帳戶

根據一個帳戶的匯入及匯出來判定此帳戶是否為可疑帳戶，採用的標準為：「該帳戶匯出的總金額 / 匯入此帳戶的總金額」的比例不小於 0.9，即判定其為可疑帳戶(此即 rule base 的標準)。因為在洗錢帳戶的判定上，作為中間媒介的人頭帳戶，其轉入的金額多半會轉出至其他帳戶。

4.4 實作

4.4.1 找出可疑帳戶間的 link

根據資料前處理的結果，我們將匯出方及匯入方皆為 rule base 預測之可疑帳

戶的 link 挑出來(共 707 條)以下將其稱為 s_link。

4.4.2 將可疑 link 串接成 path

s_link 建立完成之後，再將其串接成 path。以下將對於 path 做簡單的定義：長度是依照該條 path 上的帳戶(node)數量來定義，意即為長度 = node 數量。

演算法介紹：

我們生成長度為 n 的 path 方法是將長度為 n-1 的 path 加上 s_link 結合而成。舉例來說，長度為 3 的 path 就用兩條 s_link 合併；長度為 4 的 path 就用前面生成長度為 3 的 path 與 s_link 合成。因此長度為 n 的 path 其實就是由 n-1 條不同的 s_link 組合而成。而為了避免 cycle 發生，我們設定在同一條 path 中，node 不得重複。

合成條件：

考慮一長度為 n 的 path 及一 s_link。若要將兩者合成，則必須符合下列兩條件：

1. path 尾的帳號 = s_link 的匯出帳號
2. path 尾的交易時間不晚於 s_link 的交易時間。

若同時符合上述兩項條件則將該 s_link 的匯入方接到 path 的尾端，組合成一個長度為 n+1 的 path。我們依序生成了長度分別從 3 至 7 的 path（長度 2 的 path 即是 s_link；長度高於 7 的 path 不存在，數量為 0(如圖(二))。下圖針對各長度的path統計出他們的數量，發現path長度愈長，path 數量愈少。

path長度	總數
2(link)	707
3	384
4	318
5	154
6	31
7	2

圖(二)、Link 長度數量統計表

4.4.3 找出洗錢路徑的源頭及目的地

我們假定源頭與目的地為洗錢這一行為的實際受益者，因此為了規避法律責任，其應與人頭帳戶相異較為合理，所以在篩選過程中將不考慮 Predicted 之可疑帳戶(Predicted positive)。

根據以下敘述的篩選標準找出洗錢的源頭以及目的地。

源頭：

1. 匯出至 Predicted 可疑帳戶的個數：2
2. 匯出至 Predicted 可疑帳戶的總金額超過 200 萬

目的地：

1. Predicted 可疑帳戶匯入此帳戶的個數：5
2. Predicted 可疑帳戶匯入此帳戶總金額超過 500 萬

將 4.4.2 與 4.4.3 整合成完整路徑

以源頭為例，只要該源頭有匯出到任一條 path 的第一個帳戶，且時間不晚於 path 的第一筆時間，就將其連接。目的地連接同理。我們發現主要的 path 集中在長度為 3、4 中，而 path 多於 4 的數量就急速下降了(如圖(三))。

path長度	數量
3	160
4	129
5	45
6	24
7	7

圖(三)、Path 長度數量統計表

4.5 觀察結果並討論

根據圖(三)，我們可以假設是否大多數的洗錢帳戶都是透過共一至二個人頭帳戶去從事非法洗錢手段，進而縮小搜查範圍。

而我們也嘗試將 path 中，出現頻率較高的帳戶篩選出來，與 Ground truth 比對。將連結完成的所有 path(不含源頭、目的地，即 4.4.2 的結果)，計算每個 node 出現次數。定義出現超過 20 次的帳戶預

測為可疑帳戶。將其與 Ground truth 比較。

		Ground truth				
		Positive	Negative	Sum	accuracy	0.4273
Predicted	Positive	28	6	34	recall	0.0802
	Negative	321	216	537	precision	0.8235
	Sum	349	222	571	f1_score	0.1462

表(二) 、多層結構演算法偵測可疑帳戶

在表(二)的 confusion table 中，因為最後預測出的可疑帳戶數量少，所以 recall 值非常低是可預期的，其也連帶影響了 accuracy 及 f1 score。但值得注意的是 precision 值高達八成，這代表我們根據此基於此演算法所得出的可疑帳戶，其預測的準確度是非常高的！

4.6 本演算法過程之數學式

1. 人頭帳戶

$$\exists V_x \ if \ \frac{O_x}{I_x} \geq \alpha$$

2. 可疑交易

$$\exists E_{xyt} \ if \ \exists V_x \ and \ \exists V_y \ and \ \left(\frac{W_{xyt}}{O_x} \geq \beta_1 \ or \ \frac{W_{xyt}}{l_y} \geq \beta_2 \right)$$

註：$\frac{W_{xyt}}{O_x}$ 意即 x 與 y 的此筆交易金額佔 x 帳戶總匯出金額的比例

3. 可疑交易序列

$$Base\ case\ \exists P_{xz(t,t)} \ if \ \exists E_{xzt} \text{ 須 } t_{11} \leq t_{12},\ t_{21} \leq t_{22}$$

註：t_{11}：第一個 path 的第一筆時間
t_{12}：第一個 path 的第二筆時間

$$Recursive\ case\ \exists P_{xz(t_{11},t_{12})} \ if \ \exists P_{yz(t_{21},t_{12})} \ and \ t_{12} \leq t_{21}$$

4-1. $Source\ \exists V_s \ if \ \nexists V_s$ ($\ncong$ 可疑帳戶)

$$Let\ M_{(s)} \{x \mid \exists W_{sxt} \ and \ V_s\} \ and \sum_{x \in M_{(s)}} W_{sxt} \geq T_1$$

$$\mid M_{(s)} \mid \geq n_1$$

4-2. $Destination\ \exists V_d \ if \ \nexists V_d$

$$Let\ M_{(d)} = \{z \mid \exists W_{zdt} \ and \ V_z\} \ and \sum_{z \in M_{(d)}} W_{zdt} \geq T_2$$

$$\mid M_{(d)} \mid \geq n_2$$

5-1. $\forall V_{(s)}, x \in M_{(s)}, W_{sxt}$, 則 $\exists E_{sxt}$

5-2. $\forall V_{(d)}, x \in M_{(d)}, W_{zdt}$, 則 $\exists E_{zdt}$

6-1 $\exists P_{xz(t_s,t_z)} \ if \ \exists E_{sxt}, P_{xz(t_{11},t_{22})} \ and \ t_s \leq t_1$

6-2 $\exists P_{sd(t_s,t_d)} \ if \ \exists P_{sz(t_s,t_z)}, E_{zdt_d} \ and \ t_2 \leq t_d$

5.第二套方法(SVM 交叉驗證)

5.1 想法

我們想利用機器學習中的監督式學習方法，去檢測出可疑帳戶是否有一些金錢交易情況是我們仍未發現的。

5.2 目標

以帳戶為單位，將各帳戶的交易資料轉成數個 feature，並希望透過 SVM 將重要的 feature 找出來。而在挑選的 feature 中，有一部分是針對位數作分析的，此靈感來源出自於班佛定律。

5.3 資料前處理

5.3.1 篩選 SVM 所需的 feature

所有 feature 如下 ：

・基本統計量(金額)：

1.匯入平均、2.匯出平均、3.總平均

4.匯入中位數、5.匯出中位數、 6.總中位數、7.總交易筆數、8.標準差

・位數分析：

9.交易金額平均位元和、10.交易金額平均位數、11.交易金額平均出現幾個 0、12.交易金額出現的 0 平均佔有所有位元的比例

・該帳號較高金額的交易數據：

13.前 10%高的交易金額,0 出現平均次數、14.前 10%高的交易金額，0 占所有位元的平均比例、15.前 10%高的交易金額平均、16.前 5%高的交易金額,0 平均出現次數、17.前 5%高的交易金額，0 平均占所有位元的比例、18.前 10%高的交易金額平均 / 交易金額中位數

以上合計共 18 個 feature

5.3.3 切資料及參數設定

我們將所有帳戶，按照可疑與非可疑等比例切成五份做交叉驗證(帳戶為隨機排序)。每份資料的可疑帳戶與非可疑帳戶比例都是相同的。

5.4 實作

5.4.1 基本介紹

Classification 方式：

SVM(Support Vector Machine)，是一種藉由將資料點投射到高維空間，並且在不同類別中找出辨識分割線的機器學習方法。

工具：

利用 R 語言中的 libsvm 套件實作，為一個易於使用和快速有效的 SVM 模式識別與迴歸的套件。

Cross validation（交叉驗證）：

將資料切五份，做五輪 SVM。每份 data 都會被當 4 次 training data 跟 1 次 testing data。

5.4.2 設計之演算法概念

先分析所有 feature，依序刪除不重要的 feature。先篩選 recall 高的組合，再篩選 precision 高的組合。我們在此較注重 recall 的理由是：我們希望得出的衡量分類標準對於正確的識別能力較高。簡單來說即為寧可錯殺一百也不可放過一人。

5.4.3 實作演算法介紹

假設現在有 n 個 feature，做 n 次 cross validation。每次都挑出 n-1 個 feature，因此每個 feature 都會被捨棄一次，看哪一次的組合 recall 值最高再把這 n-1 個 feature 挑出來。所以每輪會刪掉一個 feature,再依這 n-1 個 feature 繼續做下一輪，直到無法出現更高的 recall 值時停止;precision 之過程同理。

5.5 觀察結果並討論

5.5.1 影響顯著的 Feature

將套入上述演算法中跑出的 feature 再經簡略整理後，得出的顯著 feature 如下：

(1)匯出金額平均

若只有正常交易的話，一個帳戶的匯出平均金額理論上不會太大，而洗錢人士為了將金錢轉移，會將收到的錢大部分移出，因此會有大量的金額轉出。

(2)交易次數

人頭帳戶可分為長期使用或是一次性帳戶，而帳戶交易次數過於頻繁或稀少皆異於常人的行為。

(3)前 10%高交易金額平均

洗錢帳戶在高金額金錢流動上，金額會比一般帳戶高。因為雖為規避法律責任，通常不會有到數百萬的交易。但幾十萬為單位的交易也不可小覷，這金額仍已超出了一般正常人的交易金額。

(4)標準差

標準差是看資料的離散程度，一個正常帳戶的交易金額應該大部分都在某個值區間內。但人頭帳戶的判定方法有一種是其某幾筆交易金額遠超過其職業會接觸到的金額。因此若有出現這樣的情況，該帳戶的交易金額標準差會變很大。

(5)交易金額平均 0 出現的次數、比例

人頭帳戶交易的金額通常會很乾淨，如：15,000、20,000,000 等等。但一般帳戶的交易金額應該是比較隨機、雜亂的。

5.5.2 篩選後 feature 及未篩選比較

我們將參考所有 feature 的 confusion table 以及參考顯著 feature 的 confustion table 拿來比較：

		Ground truth				
		Positive	Negative	Sum	accuracy	0.571
Predicted	Positive	997	588	1585	recall	0.2475
	Negative	3031	3824	6855	precision	0.629
	Sum	4028	4412	8440	f1_score	0.3552

表(三)、SVM 參考所有 feature 偵測可疑帳戶

		Ground truth				
		Positive	Negative	Sum	accuracy	0.58
Predicted	Positive	1063	575	1638	recall	0.263
	Negative	2965	3837	6802	precision	0.649
	Sum	4028	4412	8440	f1_score	0.374

表(四)、SVM 篩選 feature 後偵測可疑帳戶

參考表(三)及表(四)，發現兩者的各項數值都非常相近，但這不代表篩選 feature 是沒用的，而是意味著篩掉的 feature 都是對於資料沒有正面分類影響的，因此刪除這些 feature 仍是必要的。

5.5.3 SVM 與 rule base 比較

觀察表(四)，雖然沒有數值是特別高的，但若將其與表(一)比較，其各項預測標準的數值皆比表(一)還要高。尤其是 recall 值的表現，將近是表(一) recall 值的三倍，意即 SVM 能找出實際可疑帳戶的數量是 rule base 的三倍。。代表此方法對於所有帳戶是否可疑的預測方法，是完全勝過表(一)的 rule base 法的。

6. 將 SVM 法用多層結構演算優化

6.1 想法

5.中所述的 SVM cross validation 其與 rule base 都是將所有帳戶做預測，篩選出可疑帳戶。而 4.中所述的多層結構演算是將篩選出的可疑帳戶做進階篩選、優化。因此我們打算將 SVM 用多層結構演算做優化，並將其與 4.的結果比較。

6.2 目標

將 SVM cross validation 篩選出的可疑帳戶再用多層結構演算做優化，並將其與 rule base 經多層結構演算優化的結果比較，期望可以更精確。

6.3 資料前處理、實作

過程皆與 4.多層結構演算相同。

6.4 觀察結果並討論

為方便比較，將 SVM cross validation 及 rule base 經過多層結構演算優化的 confusion table，都拉到預測所有帳戶的層級下。而那些沒有被多層結構演算參考的非可疑帳戶就全部預測為非可疑帳戶（如表(五)、表(六)）。

		Ground truth				
		Positive	Negative	Sum	accuracy	0.5254
Predicted	Positive	28	6	34	recall	0.007
	Negative	4000	4406	8406	precision	0.8235
	Sum	4028	4412	8440	f1_score	0.0139

表(五)、rule base 優化(考慮所有帳戶)

		Ground truth				
		Positive	Negative	Sum	accuracy	0.5273
Predicted	Positive	48	10	58	recall	0.0119
	Negative	3980	4402	8382	precision	0.8276
	Sum	4028	4412	8440	f1_score	0.0235

表(六)、SVM 優化(考慮所有帳戶)

多層結構演算篩選的標準很嚴格，因此其篩選出的可疑帳戶會非常少。而現在又考慮到所有帳戶，因此絕大部分的帳戶都會被預測成非可疑帳戶，也在預料之中。綜上所述，兩者的 recall 值非常低也是可預期的事。

但其實仔細比較，便可發現，將 SVM 優化的成果，其預測的可疑帳戶數(58)比 rule base 的優化結果(34)多上一倍，而且兩者的 precision 值幾乎相同。這意味著在相同的命中率下，SVM 優化仍能比 rule base 的優化成果找出更多的實際可疑帳戶。

7. 主要成果與評估

在第一套成果中，4.4 的結果展現，其對於 Ground trurh 的可疑帳戶 precision 非常高(約八成)。比表(一)的 precision 高出許多(僅五成)，因此第一套方法對於從 Predicted 的可疑帳戶中，篩選出符合 Ground trurh 可疑帳戶具有高度的可靠度。

而在第二套成果中，我們發現利用 SVM 的交叉驗證結果完勝於原始表(一)的方法。而其 recall 值比起表(一)，成長了快要三倍。代表比起銀行簡易判斷洗錢帳戶的方法，SVM 法能抓出更多的實際可疑帳戶。

再將 SVM 篩選出的可疑帳戶再用多層次結構演算優化後，在相同的 precision 下，其所篩選出的實際可疑帳戶也比用多層次結構演算優化的 rule base 方法增加了快一倍。雖然篩選出的可疑帳戶總數不多，但將抓出的實際可疑帳戶數量提高了。

上述方法有各自不同的適用目標、情況。銀行實際的交易資料不是僅有這些資訊，但此次專題我們是想針對最直觀的交易資訊做初步的分析。若之後有機會取得更多資訊，也可以做更詳細的資料分析，或將本專題成果做為延伸去發想、實作。

8. 結語與展望

資料形式如何轉換、挑選 feature、工具的選擇、數據結果如何解釋……等等問題。透過團隊間的合作，上述難題逐漸明朗。因為有此次專題，讓我們學習解決處理問題的方式。從確認專題方向至成果呈現與產生，所花費的心力都將成為進步的動力，過程值得發人省思、慢慢咀嚼。期許未來繼續深入鑽研，能有更多的認知與體悟。

而本次專題許多步驟多屬實驗性質

，演算法的部分有許多可以優化之處，且洗錢方式也不只我們所提到的這幾種。篩選可疑帳戶的演算法、方式也有非常多種是我們沒想到或是沒提出的。像是初步篩選可疑帳戶不一定只能用機器學習，進階一點甚至可以利用類神經網路實作，精確度應該可以優化許多。而類似多層結構演算的優化法，我們是考慮其中一種洗錢的特徵去延伸的而已。若多考慮其他種洗錢的特徵，也可以做出許許多多的優化演算法。

9.銘謝

感謝戴指導老師在專題方面的耐心指導，在大家深陷徬徨的深淵時，伸出援手，將組員們從黯淡無邊的困境救贖。而在專題之外，為了避免組員們壓力過大而導致精神崩潰，老師亦細心地與各個組員進行心靈溝通，使溫暖流淌在我們的心房內，老師的諄諄教誨在我們的心湖中盪起了一波又一波的漣漪，彷若沒有這些動魄人心的激勵，又怎麼會有此時此刻在這裡的我們呢 ？ 對此，謙虛為懷的老師卻不曾居功，總說著這一切是我們自己的努力，但老師的付出，學生都看在眼裡，點滴在心頭。老師為了教育所做出的貢獻，早已在學生們的成長之路上點綴出無數的光彩。這樣一位將生活無私奉獻給教育的老師，又怎麼能不獲得大眾的喝采呢！

10.資料來源

永豐銀行

11.參考文獻

[1]法務部調查局洗錢防制處-疑似洗錢交易文宣
[2]維基百科條目：洗錢
[3]條文：洗錢防制法
[4]法規：中華民國銀行公會「銀行防制洗錢及打擊資恐注意事項範本」
[5]班佛定律 Archives-PanSci 泛科學