探勘封閉高效益移動序列型樣

顏秀珍
銘傳大學資工系
sjyen@mail.mcu.edu.tw

李御璽
銘傳大學資工系
leeys@mail.mcu.edu.tw

洪偉庭
銘傳大學資工系
dog0329@gmail.com

摘要

在資料探勘的領域中，序列型樣探勘是一個相當重要的課題。所謂的序列型樣，便是一連串有時間順序性且具有相當代表性的事件。而高效益移動序列型樣探勘則是想探勘出顧客在有序的地點與時間上的購買行為，且能為使用者帶來足夠的獲利。由於現存的方法會探勘出過多的型樣，可能會導致使用者難以做出決策，所以本論文將提供一個探勘封閉高效益移動序列型樣(Closed High Utility Mobile Sequential Pattern Mining)的方法，從封閉高效益移動序列型樣即可推導出所有高效益移動序列型樣，只探勘封閉高效益移動序列型樣，不但可以減少多餘型樣的產生也可以節省探勘的時間和所耗費的記憶體空間。

關鍵詞：高效益序列型樣、高效益移動序列型樣、封閉高效益移動序列型樣。

1.前言

高效益序列型樣探勘 [1, 2]是近來較新興的研究課題，例如可探勘出顧客購買了A商品後，會再購買B商品，且購買的商品組合可為商家帶來足夠的獲利。隨著電子商務的發展，許多商務網站的建立以及網際網路的流行，網路探勘(web mining) [7, 8, 9, 10]的需求也逐漸被重視並提出，但遠端通訊技術的快速發展，商家的需求增加，例如在移動交易序列資料庫中，探勘出多數的顧客在 A 商店購買 i1 商品後，接著經過 C 商店，最後在 E 商店購買了 i3 商品，且能為商家帶來足夠的獲利。因此衍伸出高效益移動序列型樣探勘 [5]的研究。高效益移動序列型樣可應用於規劃線上購物網站的網站結構或設計促銷活動廣告，甚至利用 GPS 服務，在移動的環境(餐廳、購物中心、旅遊景點等)下設計路徑規劃服務，藉由預測路徑來推薦使用者購買商品。然而探勘高效益移動序列型樣所產生的型樣中，有些型樣的購買行為其實是被其他型樣所包含的。因此本篇論文將提出一個封閉高效益移動序列型樣探勘的研究來過濾掉不必要的型樣。

2.問題定義

表一. 移動交易序列資料庫

SID	Mobile transaction sequence
S1	<(A;{[i1,2]}), (B;null), (C;{[i2,1]}), (D;{[i4,1]}), (E;null), (F;{[i5,2]}) >
S2	<(A;{[i1,3]}), (B; null), (C;{[i2,2], [i3, 5]}), (K; null), (E;{[i6,10]}), (F;{[i5,4]}), (G;{[i8,2]}), (L; null), (H;{[i7,2]}) >
S3	<(A;{[i1,3]}), (B; null), (C;{[i2,1], [i3, 5]}), (D; {[i4,2]}), (E; null), (F;{[i5,1], [i6,2]}), (G; null), (H;{[i7,1]}) >
S4	< (A;{[i1,1]}), (W; null), (C;{[i3,10]}), (E; null), (F;{[i5,1]}), (G;{[i8,2]}), (L; null), (H;{[i7,1]}), (E;{[i9,1]})>
S5	<(A;{[i1,4]}), (B; null), (C;{[i3, 10]}), (D;{[i4, 1]}), (E; null), (F;{[i5,1]}), (G; null), (H;{[i7,2]}) >
S6	<(C;{[i2,2]}), (D; null), (E; {[i9,1]}), (F;{[i5,1]}) >

表二. 項目效益表

Item	i1	i2	i3	i4	i5	i6	i7	i8	i9
Profit	1	5	3	11	18	2	5	1	3

$L = \{l_1, l_2, \ldots, l_p\}$為一個在移動交易環境下的位置集合，$I = \{i_1, i_2, \ldots, i_g\}$為一個在位置上賣出的項目集合，在移動交易序列資料庫D中，一筆移動交易序列$S = <T_1, T_2, \ldots, T_n>$為一個顧客多筆按照時間排序的交易集合，而交易中$T_j = \{ l_j; [i_{j1}, q_{j1}], [i_{j2}, q_{j2}], \cdots, [i_{jh}, q_{jh}]\}$表示在$l_j$購買了$q_{jp}$個$i_{jp}$項目。

定義1. 假設i為一個項目，i在移動交易序列資料S_j中$lloc$的效益$u(<lloc; i>, S_j)$定義如下：

$u(<lloc; ik>, S_j) = p(i) \times q_i$ ，$p(i)$為i在表二上所表示的效益。

以表一及表二為例，$u(<A; i1>, S1) = 1 \times 2$。

定義2. 假設Y為一個*loc-itemset*，Y在移動交易序列資料庫中的效益$u(Y)$定義如下：

$$u(Y) = \sum_{S_j \wedge (Y \subseteq S_j) \wedge (S_j \in D)} \sum_{k=1}^{g} u(<l_{loc}; i_k>, S_j)$$

以表一及表二為例，令$Y = <C; \{i2,i3\}>$，$u(<C; \{i2,i3\}>) = u(<C; \{i2,i3\}>, S2) + u(<C; \{i2,i3\}>, S3) = (5 \times 2 + 3 \times 5) + (5 \times 1 + 3 \times 5) = 25 + 20 = 45$。

定義3. 假設X為一個由多個*loc-itemset*所組成的*loc-pattern*，表示成$<l_1; \{i_{11}, i_{21}, \ldots, i_{g1}\}><l_2; \{i_{12}, i_{22}, \ldots, i_{g2}\}>\ldots<l_m; \{i_{1m}, i_{2m}, \ldots, i_{gm}>$，$X$在移動交易序列資料$S_J$中的效益$u(X, S_j)$定義如下：

$$u(X, S_j) = \sum_{\forall Y \in X} u(Y, S_j)$$

以表一及表二為例，令$X = <A; i1><C; \{i2, i3\}>$，$u(X, S2) = u(<A; i1>, S2) + u(<C; \{i2, i3\}>, S2) = 28$

定義4. 假設X為*loc-pattern*，X在移動交易序列資料庫中的效益$u(X)$定義如下：

$$u(X) = \sum_{S_j \wedge (X \subseteq S_j) \wedge (S_j \in D)} u(X, S_j)$$

以表一及表二為例，令X = <A; i1><C; {i2, i3}>，u(X) = u(X,S2)+u(X,S3) = 28+23 = 51。

定義5. 假設P為一個*loc-pattern*和一條path所組成的 *moving pattern*，表示成$P = <\{<l_1; \{i_{11}, i_{21}, \ldots, i_{g1}\}><l_2; \{i_{12}, i_{22}, \ldots, i_{g2}\}>\ldots<l_m; \{i_{1m}, i_{2m}, \ldots, i_{gm}>; l_1 l_2 \ldots l_m>$，其$u(P)$定義為在資料庫中包含著此路徑的*loc-pattern*效益，而*sup*(P)定義為在資料庫中包含著此路徑的移動交易序列筆數。

以表一及表二為例，令P = <{<*A; i1 ><C; {i2, i3}*>}; *ABC*>，$u(P) = u(<A; i1><C; \{i2, i3\}>, S2) + u(<A; i1><C; \{i2, i3\}>, S3) = 28+23 = 51$, $sup(P) = 2$。

定義6. 給定一個最小支持度門檻δ和一個最小效益門檻ε，若一*moving pattern* P，$sup(P) \geq \delta$且$u(P) \geq \varepsilon$，則稱P為高效益移動序列型樣，其長度為*pattern*中*loc-itemset*的數量。

以表一及表二為例，令$P = <\{<A; i1 ><C; \{i2, i3\}>\}; ABC>$，若$\delta = 2$，$\varepsilon = 50$，$sup(P) \geq 2$，$u(P) \geq 50$，則稱$P$為一長度為2的high utility mobile sequential pattern，簡稱為2-UMSP。

定義 7. 對於任兩個 *loc-pattern* $l_1 = <a_1; s_1 ><a_2; s_2 >\ldots<a_m; s_m >$和$l_2 = <b_1; t_1\}><b_2; t_2 >\ldots<b_n; t_n>$，若存在一有序整數 $1 \leq i_1 < i_2 < \ldots < i_m \leq n$，使得 $a1 = b\, i_1$ ，…， $am = b\, i_m$且 $s1 \subseteq t\ i_1$ ，…，$sm \subseteq ti_m$ ，則稱l_1為l_2的 *sub loc-pattern* ， l_2為l_1的 *super loc-pattern*，記為$l_1 \subseteq l_2$ 。

舉例，令l_1 = <A; {i1}><C; {i2}><H; {i7}>，l_2 = <A; {i1}><C; {i2}><F; {i5}><G; {i6}><H; {i7}>， l_1的所有*loc-itemset*，皆包含在l_2裡，且順序性相同，則稱l_1為l_2的*sub loc-pattern* ， l_2為l_1的*super loc-pattern*。

定義 8.對於任兩條 *path* $p_1 = a_1 a_2 \ldots a_n$ ，另一 *path* $p_2 = b_1 b_2 \ldots b_m$ ，
若存在一有序整數$1 \le i_1 < i_2 < \ldots < i_m \le n$，使得 $a_1 = b\ i_1, \ldots, a_m = b\ i_{m1}$，則稱$p_1$為$p_2$的連續子路徑(*subpath)*，$p_2$為$p_1$的連續超路徑(*superpath*)，記為$p_1 \subseteq p_2$。
舉例，令p_1 = ACF， p_2 = ACFGH，p_1的所有location皆包含於p_2，且順序性相同，則稱p_1為p_2的連續子路徑，p_2為p_1的連續超路徑。

定義9.對於一個movuing pattern M(LP; P)若不存在任何movuing pattern M'(LQ; Q)，使得LP⊆LQ, P⊆Q且sup(M)=sup (M')則M為closed。

定義 10.若一 *closed moving pattern CP*，其支持度和效益皆達到使用者自訂門檻值，則稱 *CP* 為 *Closed High Utility Mobile Sequential Pattern*。

定義 11. 移動交易序列資料*Sj*的序列效益*SU*(*Sj*)定義為*Sj*裡所有項目的效益總和。
以表一及表二為例，*SU*(*S6*) = *u*(<*C*; *i2*>,*S6*)+*u*(<*E*; *i9*>,*S6*)+*u*(<*F*; *i5*>,*S6*) = 10+ 3+18 = 31。

定義12. *loc-itemset, loc-pattern*或*moving pattern*的*sequence weighted utilization*(*SWU*)，定義為移動交易序列資料庫中包含在其中的*SU*總和。
以表一及表二為例，*SWU*(<*D*; *i4*>) = *SU*(*S1*)+*SU*(*S3*)+*SU*(*S5*) = 54+72+73 = 199；
SWU(<*A*;*i1*><*C*;{*i2*,*i3*}>) = *SU*(*S2*)+*SU*(*S3*) = 132+72 = 204；*SWU*(<{<*A*;*i1*><*C*;{*i2*,*i3*}>}; *ABC*>) = *SWU*(<*A*; *i1*><*C*;{*i2*, *i3*}>, *S2*)+*SWU*(<*A*; *i1*><*C*;{*i2*, *i3*}>, *S3*) = 132+72 = 204。

定義 13.
若一 *loc-itemset* 的 *sup*(*Y*)≥ δ且 *SWU*(*Y*) ≥ε ，簡稱為 *WULI*(*high sequence weighted utilization loc-itemset*)。
若一 *loc-pattern* 的 *sup*(*Y*)≥ δ且 *SWU*(*Y*) ≥ε ，簡稱為 *WULP*(*high sequence weighted utilization loc-pattern*)。
若一 *moving pattern* 的 *sup*(*Y*)≥ δ且 *SWU*(*Y*) ≥ε，簡稱為 *WUMSP*(*high sequence weighted utilization mobile sequential pattern*)。
若一 *moving pattern* 的 *sup*(*Y*)≥ δ且 *SWU*(*Y*) ≥ε，且為 Closed 簡稱為 C*WUMSP*(Closed *high sequence weighted utilization mobile sequential pattern*)。
若一 *loc-itemset* 的 *sup*(*Y*)≥ δ且 *SWU*(*Y*) <ε ，則稱為 *Y* 的 loc 為一 *frequent location*。

例如若最小支持度門檻δ = 2，最小效益門檻ε = 100；
P_1 = <*D*; {*i4*}>，*sup*(P_1) = 3，*SWU*(P_1) = 199；
簡稱P_1 為一 *WULI*。
P_2 = <*A*;{*i1*}><*C*;{*i2*,*i3*}>，*sup*(P_2) = 2，*SWU*(P_2) = 204；
簡稱P_2 為一 *WULP*。
P_3= <{<*A*;{*i1*}><*C*;{*i2*,*i3*}>}; *ABC*>，*sup*(P_3) = 2，*SWU*(P_3) = 204。
簡稱P_3 為一 *WUMSP*。
P_4 = <*B*; null>，*sup*(P_4) = 4，*SWU*(P_4) = 0；
簡稱 *B* 為一 *frequent location* 。

問題描述：

給定一個移動交易序列資料庫、項目效益表、最小效益門檻和最小支持度門檻，目標為在資料庫探勘出效益和支持度皆超過門檻值的封閉高效益移動序列型樣。

3.相關文獻

2001年為找出頻繁序列型樣的問題，由Jian Pei等人提出了PrefixSpan演算法[3]，而其所採取的策略為pattern-growth的方式：先掃描一次原始資料庫，找出所有頻繁項目，並以這些頻繁項目作為長度為1的序列型樣；接著重新掃描一次資料庫並以各頻繁項目為前綴（prefix）產生對應各前綴的投影資料庫（projected database）；接著對各投影資料庫重複相同的步驟，掃描一次投影資料庫找出所有的頻繁項目，再令這些頻繁項目與原本長度k的前綴組合產生長度為(k+1)的序列型樣，並以這些長度為 (k+1)的序列型樣為新的前綴來產生新的投影資料庫，直到找不到任何頻繁項目為止。此方法利用前綴與投影資料庫來找出所有前綴超序列的序列型樣，既不需產生候選序列也不需進行複雜的組合或檢查，且整個過程僅需掃描2次原始資料庫，使執行效率得到大幅度的提升；但是在執行過程中會產生大量的投影資料庫，且這些資料庫會持續占用記憶體空間直到被計算並產生下一層的投影資料庫才會被釋放，故記憶體使用量的需求較高。

2003年為了探勘封閉序列型樣(closed sequential pattern)，由X. Yan等人提出CloSpan演算法[6]。以PrefixSpan的架構為基礎，一樣僅需掃描兩次資料庫，藉由各前綴的投影資料庫，避免計算不必要的候選序列，但在掃瞄投影資料庫前，CloSpan必須判斷所增加的投影資料庫是否相同的判斷，利用向後子序列和向後超序列檢查長度k的子序列和超序列投影資料庫是否相同，若投影資料庫包含的項目個數相同，表示投影資料庫相同，則刪除子序列資料庫。如此反覆檢查並刪除，最後候選封閉序列型樣的集合便完成，最後在掃一次資料庫找出真正的封閉序列型樣。

2011年為了探勘封閉網路瀏覽型樣由Shao-An Liao等人提出MCWTP演算法[11]。以table的形式只儲存探勘網路瀏覽型樣所必須用到的資訊，僅需掃描兩次資料庫，先掃描一次原始資料庫，找出頻繁項目並作為長度為1的序列型樣，接著紀錄項目在第幾筆序列、前一個項目、序列中的位子、後一個項目，共四個資訊，紀錄完成後就可以藉由各項目的table去組合並檢查是否封閉，此演算法省去了prefix不斷建立投影資料庫的缺點，並保有樹狀結構利用遞迴方式組合項目集的便利性。

2005年為探勘高效益項目集的問題，Y. Liu等人提出了Two-Phase演算法[4]，不同於過去對探勘頻繁項目集之研究，傳統的研究中只有考慮商品的出售頻率而不考慮其獲利情況，可能使那些可以為商家帶來較高利潤的商品無法被找到。與支持度不同，一個項目集或序列的效益並不存在「若一個項目集或序列的效益超過最小效益門檻值，則其所有子集或子序列的效益也都會超過最小效益門檻值」這樣的向下封閉性；但過去研究的方法幾乎都是以向下封閉的特性為基礎，來減少需要搜尋的空間，這樣的技巧在沒有向下封閉的情況自然也無法加以利用。故本篇的作者便針對項目集的部分提出了交易交易權重效益（transaction -weighted utility, or twu）這樣的估計值，以所有包含此項目集的交易的整體效益值之總和來當作此項目集的效益，並藉此保持向下封閉的特性，即「若一個項目集的twu超過最小效益門檻值，則其所有子集的twu也都會超過最小效益門檻值」，且因為「若一個項目集的效益超過最小門檻值，則其twu亦會超過最小效益門檻值」的特性可以確保找到所有高效益項目集，使過去的方法僅需修改一小部分即可套用在新的問題上。此可以分成兩個階段：第一個階段為找出所有htwu項目集（high twu itemset, or htwu itemset），也就是找出所有twu較最小效益門檻值高的項目集；第二階段則是重新掃描原始資料庫，計算每個htwu項目集的實際效益，並找出所有重新藉由最小效益門檻

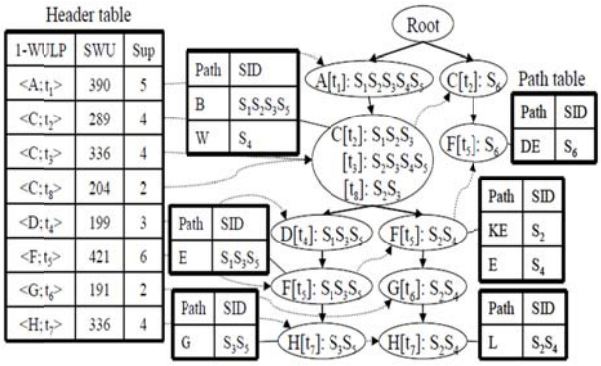
值來篩選出高效益項目集。這樣的方法優點是僅需進行少部修改即可使用過去的方法，但第二階段的計算卻較為耗費時間；且由於第一階段所拿來計算的twu是一種把一個項目集的效益高估的估計值，容易產生許多雖然twu超過最小效益門檻值但實際效益卻不足的項目集，使接下來的第二階段進行許多不必要的計算。

2011年為了探勘出高效益移動序列型樣由Bai-En Shie等人提出UMSP DFG演算法[5]。採用以FP-Growth為基礎的方法，第一步掃描資料庫計算並儲存各序列的SU，再計算每個loc-itemset並產生長度為1的1-WULI。以表一表二為例，假設門檻值min_uitility=100和min_sup=2，sup(A;i1)=5, SWU=(A;i1)=54+132+72+59+73=390。若sup和SWU都超過門檻值則稱為1- WULI，找出後轉換成相對應的表格，如表三。

表三Mapping表

1-WULI	Mapping	1-WULI	Mapping
A;i1	A;t1	F;i5	F;t5
C;i2	C;t2	G;i8	G;t6
C;i3	C;t3	H,i7	H;t7
D;i4	D;t4	C;{i2,i3}	C;t8

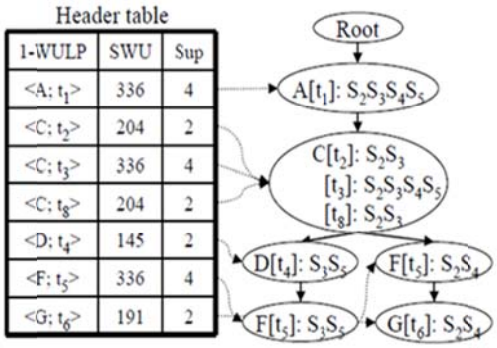
第二步建立MTS-Tree，建立原則按照移動交易序列依序建樹，當讀取到loc-itemset過程中，有對應到Mapping表則轉換並建立節點，若此loc-itemset為null或不在Mapping表裡，則視為所經過的路徑，存入路徑表。以表一為S1序列為例，< (A;{[i1,2]}), (B; null), (C;{[i2,1]}), (D;{[i4,1]}), (E; null), (F;{[i5,2]}) >，讀取A; i1時有在Mapping表裡，所以轉換成A;t1，在Root底下建立節點並儲存SID，並在Header table裡累加上此序列效益和出現次，而讀取到B; null則當作路徑暫存起來，接下來讀取到C;i2時有在Mapping表裡，所以轉換成C;t2，並把暫存路徑接到節點上的路徑表上，後面以此類推，完成後如圖一所示。



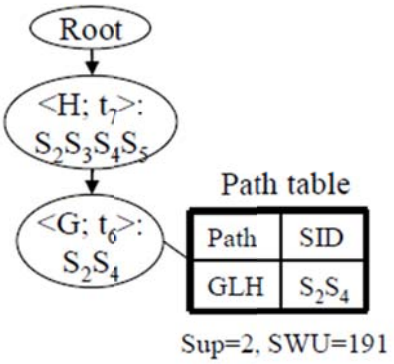
圖一 MTS-Tree

第三步為探勘MTS-Tree並產生WUMSPs，首先建立一棵WUMSP-Tree儲存答案，再來藉由Header table的連結從最底部的WULI開始建立條件樹，並組合出WULP若有達到效益門檻值，則儲存進WUMSP-Tree，並回到MTS-Tree裡搜尋路徑，路徑的支持度也達到門檻值，則繼續遞迴建立條件樹，直至無法產生長度更長的pattern。

以圖一為例，最底層的WULI為H;t7，藉由Header table上的連結來從樹上追蹤Conditional pattern base，也就是跟H;t7有關的路徑，總共會追蹤出HFDCA和HGFCA兩條路徑，並藉由節點上記錄的SID，可取交集再第一次掃描資料庫時儲存的各序列SU，可計算出跟H;t7組合出的WULP的SWU和sup，再建立條件樹，如圖二所示，接著同時在WUMSP-Tree插入H;t7節點，接著從條件樹的Header table上將最底層WULI- G;t6插入WUMSP-Tree 裡H;t7節點底下，接著回到MTS-Tree搜尋H;t7到G;t6所經過的路徑，路徑支持度有達到門檻，則記錄在WUMSP-Tree上，如圖三所示，接著遞迴到下一層，建立 < H;t7, G;t6>的條件樹，產生長度更長的pattern，直至無法產生為止。



圖二 <H;t7>的條件樹



圖三WUMSP-Tree

當整棵MTS-Tree探勘完後，只要走訪一次WUMSP-Tree，即可得到所有WUMSPs，最後再重掃一次原始資料庫計算出moving pattern的真正效益，即可找出所有的高效益移動序列型樣。

4.研究方法

本節將依序描述儲存結構，和如何利用這些儲存結構來探勘出封閉高效益移動序列型樣。

4-1.儲存結構

為了能有效的探勘出封閉高效益移動序列型樣，我們將建立WULI table和Path table這兩種儲存結構，WULI table是儲存長度為1的*1-WULI*的相關資訊，WULI table所儲存的資訊有SID、前一個loc-itemset(pre)、序列中的位置、下一個loc-itemset(next)，以表一例，若A;i1是一個*1-WULI*，之後我們去建立A;i1的WULI table，儲存 A;i1出現在S1，沒有前一個loc-itemset，則以∅做記號，當前位置則為1，下一個loc-itemset為(B; null)，整個表格建完則如表四所示。

表四 A;i1的WULI table

SID	pre	loc	next
1	∅	1	(B; null)
2	∅	1	(B; null)
3	∅	1	(B; null)
4	∅	1	(W;null)
5	∅	1	(B; null)

而Path table則紀錄frequent location的相關資訊，包含SID、序列中的位置、下一個loc-itemset(next)，以表一為例，若(B; null)為一frequent location則儲存它出現在S1，出現在第二個位置，下一個loc-itemset為C;i2，整個表格建完則如表五所示。

表五 (B; null)的path table

SID	loc	next
1	2	C;i2
2	2	C;i2
2	2	C;i3
2	2	C;{i2,i3}
3	2	C;i2
3	2	C;i3
3	2	C;{i2,i3}
5	2	C;i3

4-2. 我們的方法

在此節我們將詳細如何利用WULI table和Path table來找出所有的封閉高效益移動序列型樣。整個方法共有四個步驟，第一步，掃描第一次資料庫並把每個序列的SU儲存起來，方便之後計算每個loc-itemset的sup值和SWU，並產生長度為1的1-WULI和frequent location，之

後做簡化表示形式的動作，以表一為例，計算出的1-WULI和frequent location結果如表六和表七。

表六Mapping表

1-WULI	Mapping	1-WULI	Mapping
A;i1	A1	D;i4	D1
C;i2	C1	F;i5	F1
C;i3	C2	G;i8	G1
C;{i2,i3}	C3	H,i7	H1

表七 Frequent location

Frequent location	B;null	E;null	G;null
Mapping	B’	E’	G’

第二步，掃描第二次資料庫，建立相對應的WULI table 和Path table，如表四與表五。第三步，為利用WULI table 和Path table來組出封閉高效益移動序列型樣，判斷封閉的方法有兩個，第一個是欄位中pre或next若找無一個loc-itemset支持度相同，則為closed，第二個是若pre找無都為一相同loc-itemset，則可擴充為長度更長的封閉高效益序列型樣。

在我們方法組合過程中，假設有兩table，<X>-table、<y>-table，組合出<x1x2…xny>-table，對於<y>-table中任一筆紀錄Ty，若<X>-table中可以找到與Ty的 SID相同的紀錄Tx且Tx的loc值比Ty的值小1，則將Tx的pre值和Ty的SID、loc和next值組合成一筆紀錄Txy並加入<x1x2…xny>-table中。以A1來擴充組合出封閉高效益移動序列型樣舉例，一開始藉由A1的WULI table找出pre或next都找無一個loc-itemset支持度和A1相同，且B’的支持度有達到門檻值，所以擴充成A1 B’，並結合出A1 B’的table，計算sup值和SWU值，其結果如表八所示。

表八 A1 B’的WULI table

SID	pre	loc	next
1	∅	2	C1
2	∅	2	C1
3	∅	2	C1
5	∅	2	C2

接著重複檢查A1 B’的pre和next是否為closed 和是否能繼續擴充直至無法擴充為止。當每個WULI依序擴充組合過，所有的封閉高效益移動序列型樣則可被探勘出。第四步為還原和計算真正的效益值，因為 SWU 為一高估值，所以必須在掃一次資料庫去計算真正的效益，並還原出真正的型樣形式，例如A1B’C1D1E’F1代表的是{< A;{i_1}>< C;{i_2}>< D;{i_4}>< F;{i_5}>;ABCDEF}。

5.未來工作

探勘封閉高效益移動序列型樣的初步方法的優點為跟基於建樹的方法相比，可直接組合作計算，不須交集和遞迴組合；缺點為找相對應的SID和next需要建立許多link來加快執行效率，所以會多耗費不少記憶體。未來我們將針對以下三點去做改進：

1.改進基於table儲存結構所花費的空間。
2.降低SWU所高估Pattern的效益值。
3.設計出更容易還原原始型樣的簡化規則。

參考文獻

1. C.F. Ahmed, S.K. Tanbeer, and B.S. Jeong, “A Novel Approach for Mining High-Utility Sequential Patterns in Sequence Databases,” ***ETRI Journal***, Vol. 32:5, pp.676-686, 2010.

2. J. Yin, Z. Zheng and L. Cao, “USpan: An Efficient Algorithm for Mining High Utility Sequential Patterns,” ***Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining***, pp. 660-668, 2012.

3. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Daya and M.C. Hsu, “PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth,” ***International Conference on Data Engineering***, pp.215-224, 2001.

4. Y. Liu, W. Liao, and A. Choudhary, “A Fast High UtilityItemsets Mining Algorithm,” ***Proceedings of the ACM International. Conference on Utility-Based Data Mining Workshop*** (UBDM), pp 90 – 99, 2005.

5. Bai-En Shie, Hui-Fang Hsiao, Vincent S. Tseng, and Philip S. Yu “Mining High Utility Mobile Sequential Patterns in Mobile Commerce Environments,” ***Database Systems for Advanced Applications***, Lecture Notes in Computer Science Volume 6587, pp 224-238, 2011.

6. X. Yan , J. Han and R. Afshar“CloSpan: Mining: Closed Sequential Patterns in Large Datasets,” ***SIAM International Conference on Data Mining***, pp166-177, 2003.

7. J. Chen, T. Cook, “Mining Contiguous Sequential Patterns from Web Logs,” ***16th International World Wide Web Conference***, 2007.

8. R. Cooley, B. Mobasher, J. Srivastava, “Web Mining: Information and Pattern Discovery on the World Wide Web,” ***Proceedings of the 9th IEEE International Conference on Tool with Artificial Intelligence***, 1997.

9. J. Pei, J. Han, Behzad Mortazavi-sal, Hua Zhu, “Mining Access Patterns Efficiently from Web Logs,” ***Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining***, pages 396-407, 2000.

10. S.J. Yen and Y.S. Lee, “網路交易型樣探勘技術之研究,” ***電子商務學報***, 第十卷第四期, pp.989-1008, 2008.

11. S.J. Yen, Y.S. Lee and S.A. Liao (2011). "The Studies for Mining Closed Web Traversal Patterns," ***Proceedings of National Computer Symposium (NCS'2011)***, pp. 172-181, December 2011.