探勘有代表性的多維度序列型樣

顏秀珍 李御璽 楊順安
銘傳大學 銘傳大學 銘傳大學
資訊工程學系 資訊工程學系 資訊工程學系
sjyen@mail.mcu.edu.tw leeys@mail.mcu.edu.tw coolancool@hotmail.com

摘要

序列型樣探勘(Sequential Patterns Mining)是要從*交易資料庫*(Transaction Database)中找出大部分顧客依序購買商品的行為。因為不同特徵的顧客可能會有不同的購買行為，所以我們必須考慮何種特徵的顧客，會有何種依序購買商品的行為，這樣的資訊稱為*多維度序列型樣*(Multidimensional Sequential Patterns)。然而，從交易資料與顧客特徵資料中所探勘出來的多維度序列型樣，數量可能會很多，太多的資訊反而會造成決策者的困擾。而*封閉多維度序列型樣*(Closed Multidimensional Sequential Patterns)是具有代表性且沒有多餘資訊的多維度序列型樣，其可以推導出資料庫中所有的多維度序列型樣。本論文提出一個有效率的演算法，可以同時從客戶資料與交易資料直接得到封閉多維度序列型樣，不需要花費時間找尋非封閉多維度序列型樣，再予以刪除。實驗結果也顯示，我們的演算法確實比之前的方法更有效率。

關鍵詞：多維度序列型樣、封閉多維度序列型樣、 封閉序列型樣。

1. 前言

隨著科技的進步，電腦所能儲存及處理的資料也越來越龐大，許多企業試著想從現有的大量資料中，找出潛在的、有用的資訊，做為決策時的參考。而這樣的需求就必須仰賴*資料探勘*(Data Mining)的技術，也使得這樣的技術成為近年來熱門的研究議題。其中多維度序列型樣探勘就是資料探勘中一種重要的技術，它的主要目的是解決傳統序列型樣探勘的缺點。傳統序列型樣探勘是從交易資料庫中，找出大多數的顧客都會依照何種順序來購買哪些商品的行為，有了這些顧客消費行為的資訊後，在商品的行銷上就可以預測顧客買了某些商品之後，接著會再購買的商品，以推銷其有需求的商品，增加企業獲利，但是探勘序列型樣並沒有考慮到這樣的消費行為是以何種特徵的顧客會有的消費行為，無法正確預測顧客未來所需求的商品，而多維度序列型樣探勘正是改善此一缺點的技術。

多維度序列型樣探勘是從顧客資料和交易資料中，找出何種的特徵顧客會有何種序列型樣的消費行為。有了多維度序列型樣的資訊後，在商品的行銷上就可以預測有某種特徵的顧客，購買了某些商品之後，接著會再購買什麼商品，使得要行銷的目標更明確，不但能增加企業的獲利也降低了企業行銷成本。

以下我們定義多維度序列型樣探勘的相關名詞。一個*多維度序列資料庫 MSD*(Multidimensional sequence database)通常包含顧客編號、顧客特徵紀錄與顧客交易序列，如表 1 所示，其中 X_1, X_2 和 X_3 表示顧客的三種特徵。對於資料庫中的一筆*紀錄*(record) D=(CID, X_1, X_2, …,X_r, S)，CID 是顧客編號，X_1, X_2, …,X_r 是*多維度資訊*(Multidimensional Information)，也就是顧客依序購買商品的行為。而(X_1, X_2, …,X_r, S)稱為顧客多維度序列。令 I={i_1, i_2, …, i_m}，其中 i_k($1 \le k \le m$)為多維度序列資料庫中所出現的*項目*(Item)，每個項目代表一個商品。一個序列是由項目依照購買時間的先後順序排列而成，表示為<s_1, s_2, …, s_n>，其中 s_k($1 \le k \le n$)為一項目，一個項目可以在一個序列中出現多次，因為客戶可能會在這次交易中購買某樣商品，而在下次的交易中，又購買同一商品。一個序列的長度為在此序列中所包含的項目個數。一個長度為 K 的序列，我們稱為 *K-序列*(K-sequence)。對於兩個序列 A=<a_1, a_2, ...,a_p>與 B=< b_1, b_2, ...,b_q>，若存在整數 $1 \le j_1 < j_2 < \ldots < j_p \le q$，使得 $a_1 = b_{j1}$，$a_2 = b_{j2}$，…，$a_p = b_{jq}$，則序列 B 包含序列 A，表示為 $A \subseteq B$。一序列 S 的*支持數*(Support Count)為 MSD 中包含 S 的顧客交易序列數，而 S 的*支持度*(Support)為 S 的支持數與總顧客數的比值。若 S 的支持度不小於最小支持度，則 S 為序列型樣。若所有包含 S 的序列($\neq$S)，其支持度都小於 S，則 S 為一封閉序列。

一組特徵資料是由m個特徵值所形成的集合($V_1, V_2, ..., V_s$)，其中 V_i ($1 \le i \le S$)可為"＊"，表示此特徵為任意值，可省略不記，例如(1, 5, ＊, 9)可簡記為(1, 5, 9)。一個特徵資料 V 的長度為 V 中不為任意值"＊"的特徵值個數，例如(1, 5, ＊, 9)的長度為 3，長度為 1 的特徵資料為一個特徵值。對於兩個特徵資料 V=($v_1, v_2, ..., v_r$)和 U= ($u_1, u_2, ..., u_s$)，若存在整數 $1 \le j_1 < j_2 < \ldots < j_r \le S$ 使得 $v_1 = u_{j1}$，$v_2 = u_{j2}$，…，$v_r = u_{jr}$，則特徵資料 U 包含特徵資料 V，表示為 $V \subseteq U$。一個特徵資料 V 的支持數為 MSD 中包含 V 的顧客記錄筆數，而 V 的支持度為 V 的支持數與總顧客數的比值。若 V 的支持度不小於最小支持度，則 V 為一個頻繁特徵資料。若所有包含 V 的特徵資料($\neq$V)，其支持度都小於 V，則 V 為一封閉特徵資料。

表 1 多維度序列資料庫(MSD)

CID	X1	X2	X3	Sequences
1	1	5	8	CAABC
2	2	6	7	ABCB
3	1	6	8	CABC
4	3	4	9	ABBCA
5	2	6	8	BACC

一個*多維度序列* (Multidimensional Sequence)，為一組特徵資料與一序列的組合 <($v_1, v_2, \cdots, v_m$) $i_1 \cdots i_k$>。對於兩個多維度序列 α 和 β，若 β 中的特徵資料包含 α 中的特徵資料或 β 中的序列包含 α 中的序列，則 β 包含 α。若 $\alpha \neq \beta$，則我們稱 α 是 β 的*子多維度序列*(Sub - Multidimensional Sequence)，而 β 是 α 的*超多維度序列* (Super-Multidimensional Sequence)。例如：多維度序列<(1, *, 5) badc>、<(1, 3, 5) bad>和<(1, *, 5) bad>都是<(1, 3, 5) badc>的子多維度序列。一個多維度序列 MS 的支持數為 MSD 中包含 MS 的顧客多維度序列個數，而 MS 的支持度為 MS 的支持數與總顧客數的比值。若 MS 的支持度不小於最小支持度，我們稱 MS 為多維度序列型樣(Multidimensional Sequential Patterns)。例如表 1 的多維度序列資料庫中，假設最小支持度為 40%，有兩筆顧客序列包含<(2, 6, *) BC>，分別是顧客編號 2 和顧客編號 5，則此多維度序列的支持數為 2，支持度為 2/5=40%，不小於最小支持度，因此多維度序列<(2, 6, *) BC>為一多維度序列型樣。

關於多維度序列型樣探勘，已有多篇論文提出不同的探勘方法，它們都是從給定的多維度序列資料庫中，找出多維度序列型樣。在許多實際的應用中，如 3C 資訊賣場及民生用品大賣場等，行銷決策者很難將探勘產生的多維度序列型樣，應用於實際行銷上，原因在於探勘出來的多維度序列型樣數量太多，需要花費更多的時間、更多的儲存空間去探勘多維度序列型樣，而且資訊太多也造成決策者的困擾。因此如何有效率地從多維度序列資料庫中，探勘出具有代表性的多維度序列型樣，便成為一項具有現實應用需求的研究議題，這方面的研究稱為*封閉多維度序列型樣探勘*(Closed Multidimensional Sequential Patterns Mining)[9,10]。若一個多維度序列 α 的支持度不小於最小支持度，則 α 為多維度序列型樣。若多維度序列 α 大於它的所有超多維度序列的支持度，則 α 為封閉多維度序列型樣。探勘封閉多維度序列型樣不但可以得到精簡且具有代表性的資訊，也可以推導出所有多維度序列型樣的資訊。

封閉多維度序列型樣這個觀念最早是 Songram 等學者[10]在 2006 年提出，並在同篇論文提出一個 CCMD 演算法探勘封閉多維度序列型樣，但是 CCMD 演算法需要大量記憶體去儲存*候選封閉多維度序列型樣*(Closed Multidimensional Sequential Pattern Candidates)，和花費過多時間刪除不是封閉多維度序列型樣的候選封閉多維度序列，因此隨後 Songram 等學者[9]在 2007 年提出 CIScombine 演算法，減少產生候選封閉多維度序列。然而，CIScombine 演算法還是需要大量記憶體去儲存候選封閉多維度序列，並花費過多時間去拆解交易序列和刪除不是封閉多維度序列型樣的候選封閉多維度序列。

在這篇論文中，我們提出一個有效率探勘封閉多維度序列型樣的演算法 CMSP (mining Closed Multidimensional Sequential Patterns)。CMSP 在探勘過程中，利用一些機制，可以直接判斷特徵資料與序列的組合是否為封閉多維度序列型樣，而且可以減少很多不必要的組合，這樣的好處是不需產生並儲存非封閉多維度序列型樣，也省去了判斷並刪除非封閉多維度序列型樣所花費的時間。

2.相關工作

在本節中我們分別介紹封閉項目集探勘演

算法、封閉序列型樣探勘演算法，多維度序列型樣探勘演算法和封閉多維度序列型樣探勘演算法的相關研究。

Pasquier 等學者 [5, 6] 在 1999 年提出封閉項目集觀念，並且在同篇論文中提出 A-Close 演算法從資料庫中探勘封閉項目集，隨後陸續有學者提出了許多封閉項目集演算法[2, 3, 4, 7, 13, 15]。以下我們將介紹 A-Close 演算法。

A-Close 演算法可分成兩個階段，第一個階段先找出資料庫中可用來產生封閉項目集的頻繁項目集。第二個階段利用頻繁項目集來產生封閉項目集。第一個階段的步驟是類似 Apriori 演算法[1]，同樣都是掃描資料庫計數長度為 k 的*候選項目集*(Candidate Itemset)的支持數以找出長度為 (k+1) 的頻繁項目集。項目集組合的方式也與 Apriori 相同。但與 Apriori 不同的地方在於當 A-Close 計數完每個長度為 k 的候選項目集後，需要針對每個候選項目集 X 搜尋長度為 (k-1) 的頻繁項目集 Y，如果 X⊃Y 且 X 與 Y 的支持數相同，則刪除頻繁項目集 X。此步驟可以減少一些候選項目集的產生，但是必須多花費一些時間搜尋相同支持數的*子項目集*(Sub Itemset)。若頻繁項目集已經無法再組合出下一個長度的候選項目集，則進行第二個階段。

A-Close 的第二個階段是利用被保留下來的頻繁項目集產生封閉項目集，在這個階段需要再掃描一次資料庫。從被保留的頻繁項目集產生封閉項目集的過程敘述如下，假設用來產生封閉項目集的項目集為 X，則挑出所有包含 X 的交易，並將這些交易進行交集運算，交集的結果是一個包含 X，且支持數與 X 相同的封閉項目集。最後，去除重複的封閉項目集，留下來的項目集即為封閉項目集。

Yan 等學者 [14] 在 2003 年提出封閉序列型樣這個觀念，並且在同篇論文中提出 Clospan 演算法從資料庫中探勘封閉序列型樣，隨後陸續有學者提出了許多封閉序列型樣演算法[11, 12]。以下我們將介紹 BIDE 演算法。

在 BIDE 演算法提出之前的封閉序列型樣研究，都必須在記憶體內維持已經被探勘出來的*候選封閉序列* (Closed Sequence Candidates)，這樣的做法需要大量記憶體空間去儲存候選封閉序列，並花費許多時間在刪除不是封閉序列型樣的候選封閉序列，因此 BIDE 提出一個新的檢查封閉序列的方法，稱為 *BI-Directional Extensional checking*。根據頻繁封閉序列的定義，如果一個長度為 n 的 sequence (n-sequence) $S=e_1\ e_2 \ldots e_n$ 不是封閉序列的話，那至少會存在一個項目(event) e’，可以使得 S 去擴展得到一個新的序列 S’，而且 S’ 的支持度和 S 的支持度一樣。BIDE 演算法主要分成兩個階段，第一個階段是 subpattern checking，在這個階段要檢查最新找出來的型樣是否可被已找出的*候選封閉型樣*(Closed Pattern Candidates) 包含。第二個階段是 superpattern checking，在這個階段要檢查最新找出來的型樣是否包含已找出的*候選封閉型樣*(Closed Pattern Candidates)。

多維度序列型樣這個觀念最早是 Pinto 等學者[8]在 2001 年提出，並在同篇論文中提出 Seq-Dim 演算法從多維度序列資料庫中探勘出多維度序列型樣。

多維度序列型樣演算法在探勘密集型資料庫的過程中產生過多的多維度序列，導致執行時間較緩慢與記憶體使用量不足的情況。面對探勘產生大量的多維度序列型樣，行銷者很難將這些探勘出來的大量資訊應用於行銷決策上，因此有學者提出封閉多維度序列型樣這個觀念，想從多維度交易資料庫中找出具有代表性的多維度序列，並且探勘出來的多維度序列個數較少，利於行銷者應用於行銷決策上。

目前探勘封閉多維度序列型樣的演算法有 CCMD 和 CIScombine 演算法，首先，介紹 CCMD 演算法。CCMD 演算法分成兩個階段，第一個階段先從資料庫客戶維度部份，利用所封閉項目集演算法，探勘出封閉項目集集合，並將此集合稱為 CI。接著從資料庫顧客交易序列部份，利用封閉序列型樣探勘演算法，探勘出封閉序列型樣集合，並將此集合稱為 CS。第二個階段，將第一階段 CI 集合內的每一個封閉項目集和 CS 集合內的每一個封閉序列型樣做組合的動作，產生候選封閉多維度序列，接著掃瞄一次資料庫，計數每個候選封閉多維度序列在資料庫出現的次數，刪除小於最小支持數的候選封閉多維度序列。最後在刪除不是封閉多維度序列型樣的候選封閉多維度序列。

CCMD 演算法產生許多不是封閉多維度序列型樣的候選多維度序列，因此需要花費許多時間刪除多餘的候選多維度序列，CIScombine 演算法改善了這個缺點，以下將介紹 CIScombine 演算法

CIScombine 演算法分成兩個階段。第一個階段先從資料庫客戶維度部份，利用第一節所

提到的封閉項目集演算法，探勘出封閉項目集。第二個階段從第一個階段找出的封閉項目集裡，挑出長度最短的封閉項目集，如此時有個長度最短的封閉項目集 X，接著從資料庫裡找出和 X 有相關的顧客交易序列，利用封閉序列型樣探勘演算法，探勘出封閉序列型樣，如找出的封閉序列型樣為 AA、AC 且支持度分別為 3、4。接著從第一個階段找出的封閉項目集裡，找出 X 的超項目集(Super-itemset)，如此時 X 的超項目集是 Y。接著我們要找出和 Y 所相關的封閉序列型樣，則將 X 在資料庫出現的顧客編號集合去減掉 Y 在資料庫出現的顧客編號集合，如此時減掉後的結果為顧客編號 2，我們就將顧客編號 2 的顧客交易序列做拆解的動作，看哪些序列是 X 的封閉序列型樣，如拆解顧客編號 2 的交易序列找出 AA 是 X 的封閉序列型樣，則將 X 的封閉序列型樣 AA 的支持數減掉一次，就可得到和 Y 相關的封閉序列型樣 AA 且支持數為 2。接下來以此類推，找出封閉項目集和它相關的封閉序列型樣。最後刪除不是封閉多維度序列型樣的候選封閉多維度序列。

CIScombine 演算法雖然比 CMDS 演算法產生較少候選封閉多維度序列，但需要大量記憶體空間去儲存候選封閉多維度序列，並花費許多時間刪除不是封閉多維度序列型樣的候選封閉多維度序列，因此我們提出一個有效率的演算法，在探勘的過程中，就可判斷此一組合是否為封閉多維度序列型樣，且不需要大量記憶體空間去儲存候選封閉多維度序列。

3. 我們的演算法

目前探勘封閉序列型樣最好的演算法是 BIDE 演算法，其在探勘過程中，不需要產生候選封閉序列型樣，因此節省了許多搜尋時間與儲存空間。因此我們利用 BIDE 演算法的概念，發展出探勘封閉多維度序列型樣的演算法 CMSP。以下我們先介紹 CMSP 演算法的相關定義。

定義 1:序列 Sp 在序列 S 中的第一前序序列為，從序列 S 的第一個項目開始到序列 S 中第一次出現序列 Sp 所形成的子序列。例如：序列 S 是<CAABCB>，前序序列 Sp 是<AB>，則序列<AB>在序列<CAABCB>的第一前序序列就是序列<CAAB>。

定義 2: 序列 Sp 在序列 S 中的後序序列為，從序列 S 中移除 Sp 在序列 S 中的第一前序序列，所留下的序列。例如：序列 S 是<CAABCB>，前序序列是<AB>，則<AB>在<CAABCB>的後序序列是<CB>。

定義 3 :序列 Sp 在一多維度交易序列資料庫 MSD 中的投影資料庫為，Sp 在每一個顧客交易序列中的後序序列集合。例如：MSD 有四筆交易序列紀錄，分別是<ABC>、<ABCB>、<CABC>和<ABBCA>，前序序列<AB>在 MSD 中的後序序列集合是{<C>, <CB>, <C>,<BCA>}。

定義 4 :序列 Sp 在序列 S 中的最後前序序列為，從序列 S 的第一個項目開始到序列 S 中最後一次出現序列 Sp 所形成的子序列。例如：序列 S 是<CAABCB>，前序序列 Sp 是<AB>，則<AB>在<CAABCB>中的最後前序序列是<CAABCB>。

根據封閉多維度序列型樣的定義，如果一個多維度序列 MS=<$(D_1D_2…D_k)s_1\ s_2…s_n$>不是封閉多維度序列的話，那麼至少會存在一個特徵值或項目 e'(適稱為 event)，使得 MS 可以擴展成一個新的多維度序列 MS'，而且 MS'的支持度和 MS 的支持度一樣。擴展 MS 的方法有下列六種: (1) MS'=<$(D_1D_2…D_k)s_1\ s_2…s_n$e'>; (2) MS'=<$(D_1D_2…D_k$e')$s_1\ s_2…s_n$ >;(3)至少存在一個 i ($1\le i < n$)，使得 MS'=<$(D_1D_2…D_k)s_1\ s_2…s_i$e'$s_{i+1}…s_n$>; (4)至少存在一個 i ($1\le i < k$)，使得 MS'=<$(D_1D_2…$e' $D_k)\ s_1\ s_2…s_n$>; (5) MS'=<$(D_1D_2…D_k)$ e'$s_1\ s_2…s_n$>。在第(1) 和第(2)中的 MS'，event e'出現在 D_k 或 s_n 後面，所以稱 e'是一個 Forward-extension event 而且 MS'是 MS 的 Forward-extension Multidimensional sequence。在第(3)、(4)和第(5)的 MS'中，event e'出現在 D_k 或 s_n 前面，所以稱 e'是一個 Backward-extension event 而且 MS'是 MS 的 Backward-extension Multidimensional sequence。

定理 1 若一多維度序列 MS = <$(D_1D_2…D_k)s_1\ s_2…s_n$ >是封閉多維度序列，則不存在任何一個 Forward-extension event 和 Backward-extension event 使得 MS 可以擴展得到一個新的多維度序列 MS'，且 MS 的支持度和 MS'的支持度相

同。

理由：若 MS 有任一 Forward-extension event 或 Backward-extension event，可以讓 MS 擴展得到 MS'，則 MS' 是 MS 的 super-multidimensional sequence，若 MS 和 MS' 支持度相同，根據封閉多維度序列型樣的定義，MS 不是封閉多維度序列。

定理 2 若一多維度序列 MS = $<(D_1D_2\ldots D_k)s_1 s_2\ldots s_n>$是封閉多維度序列，則$(D_1D_2\ldots D_k)$必定是封閉特徵資料，且$<s_1\ s_2\ldots s_n>$必定是封閉序列。

理由：根據定理 1，若 MS 是封閉多維度序列，則不存在任一 Forward-extension event 或 Backward-extension event，使得 MS 可以擴展出與其支持度相同的超多維度序列。因為 M=$(D_1D_2\ldots D_k)$無法加入任何特徵值，使其支持度與 M 相同，也就是所有包含 M 的特徵資料，其支持度都小於 M，所以 M 為一封閉特徵資料。因為 S=$<s_1\ s_2\ldots s_n>$也無法加入任何項目，使其支持度與 S 相同，也就是所有包含 S 的序列，其支持度都小於 S，故 S 為一封閉序列。

定理 3 若一特徵值或項目都出現在前序序列 MS 的第一前序序列中，則我們可以停止前序序列 MS 繼續和其它特徵值或項目做組合，產生更長的型樣。

3.1 CMSP 方法描述

給予一個多維度序列資料庫 MSD，CMSP 首先掃瞄 MSD 的顧客特徵紀錄，找出頻繁特徵值，並建立 CID list。對於每一個頻繁特徵值 x，CID list 儲存了包含 x 的顧客特徵記錄所對應的 CIDs。紀錄 CID list 的好處是在探勘過程中可以加速找尋特徵資料有出現在哪幾筆顧客特徵記錄中，節省搜尋的時間。CMSP 從第一個特徵維度開始，對於每一個頻繁特徵值，找出與此特徵值相關的所有封閉多維度序列型樣。對於每一個特徵值 x，CMSP 從 CID list 取得 x 在 MSD 出現的顧客記錄，建立 x 的子多維度序列資料庫 MSD_x，並找出頻繁特徵值與頻繁項目，若有特徵值 $y_1, y_2,\ldots$和 y_m 其支持度與 x 相同，則結合成一頻繁封閉特徵資料$(x, y_1, y_2,\ldots, y_m)$。否則 x 為一頻繁封閉特徵資料。將目前已找到的頻繁封閉特徵資料儲存於 CI 表格中。

CMSP 將每一個頻繁特徵資料 X 與每一個頻繁項目 f 組合成多維度序列<Xf>，再從 MSD_x 中找出有出現序列 f 的紀錄形成 $MSD_{<Xf>}$，並計算 f 在這些記錄中顧客交易序列的第一前序序列中每一項目的支持度，以及這些記錄中每一特徵值的支持度。若有項目或特徵值的支持度與 f 相同，表示<Xf>有超多維度序列的支持度與<Xf>相同，則<Xf>和其它項目或特徵值的組合，會和它的超多維度序列組合相同，並<Xf>不是一個封閉多維度序列，否則要繼續判斷<Xf>是否為封閉多維度序列。CMSP 計算 f 在 $MSD_{<Xf>}$ 中每一顧客交易序列的最後前序序列中每一項目的支持度，若有項目的支持度與 f 相同，則<Xf>不是封閉多維度序列。CMSP 從 $MSD_{<Xf>}$ 中產生 f 的投射資料庫 $D_{<xf>}$，並計算 $D_{<xf>}$ 中每一項目的支持度，找出 $D_{<xf>}$ 中的頻繁項目。若有項目的支持度與 f 相同，則<Xf>不是封閉多維度序列，否則<Xf>是封閉多維度序列。CMSP 繼續將<Xf>與 $D_{<xf>}$ 中的每一頻繁項目 g 結合成<Xfg>，依照上述方式判斷<Xfg>是否為封閉多維度序列。　　在下一節，我們將舉一個實例，更詳細的描述整個探勘的過程。

3.2 CMSP 演算法實例描述

在這一節中，我們以表 1 的多維度序列資料庫 MSD 舉例描述 CMSP 演算法的探勘過程。假設最小支持度為 40%也就是最小支持數是 2。CMSP 首先掃瞄資料庫中的顧客特徵紀錄一次，找出頻繁特徵值，結果如表 2 所示，並建立 CID list，如表 3 所示。

表 2 頻繁特徵值與支持數

頻繁特徵值	支持數
1	2
2	2
6	3
8	3

對於特徵值 1，由 CID list 得知其出現在 MSD 的第 1 和第 3 筆紀錄，故取出第 1 和第 3 筆紀錄如表 4 所示。因為目前沒有任何封閉特徵資料，所以繼續找尋表 4 中的頻繁特徵值與頻繁項目，如表 5 所示。

表 3 CID list

頻繁特徵值	CID list
1	1 3
2	2 5
6	2 3 5
8	1 3 5

表 4 第一和第三筆記錄

CID	X1	X2	X3	Sequences
1	1	5	8	CAABC
3	1	6	8	CABC

表 5 頻繁特徵值或項目與其支持數

頻繁特徵值或項目	支持數
8	2
A	2
B	2
C	2

因為特徵值 8 的支持數和特徵值 1 的支持數相同，因此，特徵值 1 不是一個頻繁封閉特徵資料。CMSP 將頻繁特徵值 1 和特徵值 8 組合成頻繁封閉特徵資料(1 8)，並將(1 8)和其支持數記錄到表格 CI 內，如表 6 所示。

表 6 表格 CI

頻繁封閉特徵資料	支持數
(1 8)	2

將(1 8)和表 5 中的頻繁項目 A 組合成多維度序列<(1 8) A>，並從表 4 中找出<A>的第一前序序列並計算每一特徵值或項目的支持數，如表 7 所示。從表 7 可發現 C 的支持數和<(1 8) A>相同，因此<(1 8) A>不是封閉多維度序列，根據定理 3，<(1 8) A>不用繼續組合產生更長的型樣，因為<(1 8) A>和其它項目的組合會和<(1 8) CA>相同。(1 8) 和表 5 中的頻繁項目 B 組合成多維度序列<(1 8) B>，從表 4 中找出<B>的第一前序序列並計算每一特徵值或項目的支持數，如表 8 所示。

表 7 特徵值或項目與其支持數

特徵值或項目	支持數
5	1
6	1
C	2

表 8 特徵值或項目與其支持數

特徵值或項目	支持數
5	1
6	1
A	2
C	2

從表 8 可發現項目 C 和 A 與<(1 8) B>的支持數相同，因此<(1 8) B>不是封閉多維度序列，根據定理 3，<(1 8) B>不用繼續組合產生更長的型樣，因為<(1 8) B>和其它項目的組合會和<(1 8) CAB>相同。(1 8) 和表 5 中的頻繁項目 C 組合成多維度序列<(1 8) C>，從表 4 中找出<(1 8) C>的第一前序序列並計算每一特徵值或項目的支持數，如表 9 所示。從表 9 中得知沒有特徵值或項目的支持數與<(1 8) C>相同。CMSP 從表 4 中找出<(1 8) C>的最後前序序列，並計算每一項目的支持數，如表 10 所示。

表 9 特徵值或項目支持數

特徵值或項目	支持數
5	1
6	1

表 10 項目與其支持數

項目	支持數
A	2
B	2
C	2

從表 10 可得知項目 C、B 和 A 的支持數和<(1 8) C>相同，所以<(1 8)C>不是封閉多維

度序列。從表 4 中將項目 C 和項目 A、項目 B 和項目 C，分別組合成<CA>、<CB>、<CC>，所以(1 8)可和<CA>、<CB>和<CC>組合成<(1 8) CA>、<(1 8) CB>、<(1 8) CC>。從表 4 中找出<(1 8) CA>的第一前序序列並計算每一特徵值或項目的支持數，如表 11 所示。

表 11 特徵值或項目與其支持數

特徵值或項目	支持數
5	1
6	1

從表 11 中得知沒有特徵值或項目的支持數與<(1 8) CA>相同。CMSP 從表 4 中找出<(1 8) CA>的最後前序序列，並計算每一項目的支持數，如表 12 所示。從表 12 中得知沒有項目的支持數與<(1 8) CA>相同。從表 4 中，建立<CA>的投射資料庫並計算每一項目的支持數，得到項目 B 和 C 的支持數為 2，故<(1 8) CA>不是為一封閉多維度序列，因為<(1 8) CA>與它的超多維度序列<(1 8) CAB>、<(1 8) CAC>支持數相同。CMSP 將<(1 8) CA>和項目 B 組合成<(1 8) CAB>，<(1 8) CA>和項目 C 組合成<(1 8) CAC>。從表 4 中找出<(1 8) CAB>的第一前序序列並計算每一特徵值或項目的支持數，如表 4 所示。

表 12 項目與其支持數

項目	支持數
A	1

從表 13 中得知沒有特徵值或項目的支持數與<(1 8) CAB>相同。CMSP 從表 4 中找出<(1 8) CAB>的最後前序序列，並計算每一項目的支持數，如表 14 所示。表 14 中得知沒有項目的支持數與<(1 8) CAB>相同。從表 4 中，建立<CAB>的投射資料庫並計算每一項目的支持數，得到項目 C 的支持數為 2，故<(1 8) CAB>不是為一封閉多維度序列，因為<(1 8) CAB>與它的超多維度序列<(1 8) CABC>支持數相同。CMSP 將<(1 8) CAB>和項目 C 組合成<(1 8) CABC>。從表 4 中找出<(1 8) CABC>的第一前序序列並計算每一特徵值或項目的支持數，如表 15 所示。從表 15 中得知沒有特徵值或項目的支持數與<(1 8) CABC>相同。CMSP 從表 4 中找出<(1 8) CABC>的最後前序序列，並計算每一項目的支持數，如表 16 所示。表 16 中得知沒有項目的支持數與<(1 8) CABC>相同。從表 4 中，建立<CABC>的投射資料庫並計算每一項目的支持數。沒有一個項目可和<(1 8) CABC>組合成更長的型樣，因此<(1 8) CABC>為一封閉多維度序列。

表 13 特徵值或項目與其支持數

特徵值或項目	支持數
5	1
6	1
A	1

表 14 項目與支持數

項目	支持數
A	1

表 15 特徵值或項目支持數

特徵值或項目	支持數
5	1
6	1
A	1

表 16 項目與其支持數

項目	支持數
A	1

表 17 第 2 和第 5 筆記錄

CID	X1	X2	X3	Sequences
2	2	6	7	ABCB
5	2	6	8	BACC

從頻繁特徵值 1 的組合中，我們找到<(1 8)CABC>為一封閉多維度序列，其它的組合皆不是封閉多維度序列。從表 2 的下一個頻繁特徵值 2 做組合的動作。從表格 CI 內找尋是否有頻繁封閉特徵資料包含頻繁特徵值 2，且支持數相同，此時表格 CI 內只記錄了(1 8)和它的支持度數 2，所以(1 8)沒有包含頻繁特徵值 2，繼續執行下一步驟。掃瞄 CID list 找到頻繁特徵值 2 出現在 MSD 的第 2 和第 5 筆記錄，故

取出第 2 和第 5 筆紀錄，如表 17 所示。從表 17 中計算每一特徵值與項目的支持數，如表 18 所示。

表 18 特徵值或項目與其支持數

特徵值或項目	支持數
6	2
7	1
8	1
A	2
B	2
C	2

從表 18 可得知特徵值 6 的支持數和頻繁特徵值 2 的支持數相同，因此頻繁特徵值 2 不為一頻繁封閉特徵資料。將頻繁特徵值 2 與特徵值 6 組合成頻繁封閉特徵資料(2 6)，並將(2 6)和它的支持數記錄到表格 CI 內，如表 19 所示。

表 19 表格 CI

頻繁封閉特徵資料	支持數
(1 8)	2
(2 6)	2

表 20 特徵值或項目與其支持數

特徵值或項目	支持數
7	1
8	1
B	1

表 21 項目與其支持數

特徵值或項目	支持數
B	1

將(2 6)和表 18 中的項目 A 組合成多維度序列<(2 6) A>，並從表 17 中找出<A>的第一前序序列並計算每一特徵值或項目的支持數，如表 20 所示。從表 20 中得知沒有特徵值或項目的支持數與<(2 6) A>相同。CMSP 從表 17 中找出<(2 6) A>的最後前序序列，並計算每一項目的支持數，如表 21 所示。

表 21 中得知沒有項目的支持數與<(2 6) A>相同。從表 17 中，建立<A>的投射資料庫並計算每一項目的支持數，如表 22 所示。從表 22 中得知項目 C 的支持數為 2 與<(2 6) A>相同，故<(2 6) A>不為一封閉多維度序列。CMSP 將<(2 6) A>與項目 C 組合成<(2 6) AC>，並從表 17 中找出<AC>的第一前序序列並計算每一特徵值或項目的支持數。沒有項目或特徵值的支持數與<(2 6) AC>相同。CMSP 從表 17 中找出<(2 6) AC>的最後前序序列，並計算每一項目的支持數。沒有項目或特徵值的支持數與<(2 6) AC>相同。從表 17 中，建立<AC>的投射資料庫並計算每一項目的支持數。沒有項目的支持數與<(2 6) AC>相同，故<(2 6) AC>為一封閉多維度序列。CMSP 將(2 6)和表 18 中的項目 B 組合成多維度序列<(2 6) B>，並從表 17 中找出<B>的第一前序序列並計算每一特徵值或項目的支持數，沒有項目或特徵值的支持數與<(2 6)B>相同。CMSP 從表 17 中找出<(2 6) B>的最後前序序列，並計算每一項目的支持數。沒有項目的支持數與<(2 6) B>相同。從表 17 中，建立<B>的投射資料庫並計算每一項目的支持數，得到項目 C 的支持數與<(2 6) B>相同，故<(2 6) B>不為一封閉多維度序列。CMSP 將<(2 6) B>和項目 C 組合成多維度序列<(2 6) BC>，並從表 17 中找出<BC>的第一前序序列並計算每一特徵值或項目的支持數，沒有項目或特徵值的支持數與<(2 6)BC>相同。CMSP 從表 17 中找出<(2 6) BC>的最後前序序列，並計算每一項目的支持數。沒有項目的支持數與<(2 6) BC>相同。從表 17 中，建立<BC>的投射資料庫並計算每一項目的支持數，沒有項目的支持數與<(2 6) BC>相同，故<(2 6) BC>為一封閉多維度序列。

表 22 項目與其支持數

項目	支持數
B	1
C	2

從頻繁特徵值 2 的組合中，我們找到<(2 6)AC>與<(2 6)BC>為一封閉多維度序列。從表 3-3 的下一個頻繁特徵值 6 做組合的動作。從表格 CI 內找尋是否有頻繁封閉特徵資料包含頻繁特徵值 6，且支持數相同，此時表格 CI 內記錄了(1 8)和(2 6)，雖然(2 6)有包含頻繁特徵

值 6，但他們的支持數不相同，因此繼續執行下一步驟。掃瞄 CID list 找到頻繁特徵值 6 出現在 MSD 的第 2、第 3 和第 5 筆記錄，故取出第 2、第 3 和第 5 筆紀錄，如表 23 所示。從表 23 中計算每一特徵值與項目的支持數，如表 24 所示。從表 24 中得知，沒有特徵值支持數與頻繁特徵值 6 的支持數相同，故頻繁特徵值 6 為一頻繁封閉特徵資料，並將頻繁封閉特徵資料 6 與支持數儲存到表格 CI。

表 23　第 2、第 3 和第 5 筆記錄

CID	X1	X2	X3	Sequences
2	2	6	7	ABCB
3	1	6	8	CABC
5	2	6	8	BACC

表 24　特徵值或項目

特徵值或項目	支持數
8	2
A	3
B	3
C	3

CMSP 將(6)和表 24 的頻繁項目 A 組合成多維度序列<(6) A>，並從表 23 中找出<A>的第一前序序列並計算每一特徵值或項目的支持數，如表 25 所示。

表 25　特徵值或項目

特徵值或項目	支持數
1	1
2	2
7	1
8	2
B	1
C	1

沒有項目或特徵值的支持數與<(6)A>相同。CMSP 從表 23 中找出<(6)A>的最後前序序列，並計算每一項目的支持數，如表 26 所示。沒有項目的支持數與<(6) A>相同。從表 23 中，建立<A>的投射資料庫並計算每一項目的支持數，如表 27 所示。

表 26　項目與支持數

項目	支持數
B	1
C	1

表 27　項目與其支持數

項目	支持數
B	2
C	3

表 28　項目與其支持數

項目	支持數
B	1
C	2

表 29 封閉多維度序列

封閉多維度序列	支持數
<(1 8) CABC>	2
<(2 6) AC>	2
<(2 6) BC>	2
<(6) ABC>	2
<(6) AC>	3
<(6) BC>	3
<(6) CB>	2
<(6 8) AC>	2
<(6 8) BC>	2
<(6 8) CC>	2
<(8) AC>	3
<(8) BC>	3
<(8) CC>	3

從表 27 可得知項目 C 的支持數與<(6) A>相同，故<(6) A>不為一封閉多維度序列。CMSP 將<(6) A>與項目 B 組合成<(6) AB>，並從並從表 23 中找出<(6) AB>的第一前序序列並計算每一特徵值或項目的支持數。沒有項目或特徵

值的支持數與<(6) AB>相同。CMSP 從表 23 中找出<(6) AB>的最後前序序列，並計算每一項目的支持數。沒有項目的支持數與<(6) AB>相同。從表 23 中，建立<AB>的投射資料庫並計算每一項目的支持數，如表 28 所示。從表 28 中得知項目 C 的支持數與<(6) AB>，故<(6) AB>不為一封閉多維度序列。MSP 將<(6) A>與項目 B 組合成<(6) ABC>，以上述方法判斷<(6) ABC>是否為一封閉多維度序列。

從頻繁特徵值 6 的組合中，我們找到<(6)ABC>、<(6)AC>、<(6)BC>、<(6)CB>、<(6 8)AC>、<(6 8)BC>和<(6 8)CC>為一封閉多維度序列。頻繁特徵值 8 的探組合方法也如上述的步驟一樣，最後找到<(8) AC>、<(8) CC>、<(8) BC>。從 MSD 找出的封閉多維度序列如表 29 所示。

4. 結論與未來研究工作

在本篇論文中，我們提出一個有效率探勘封閉多維度序列型樣的演算法 CMSP，探勘過程中不會產生非封閉多維度序列，因此沒有花費許多時間在刪除非封閉多維度序列的問題，並且可以快速找出封閉多維度序列，節省許多執行時間。從 CCMD 演算法可以得知，刪除非封閉多維度序列會花費許多的時間，而 CIScombine 演算法也會產生非封閉多維度序列。實驗顯示我們的 CMSP 演算法的效能皆優於 CCMD 和 CIScombine 演算法，即使將資料量增加，CMSP 仍然適用。

雖然從投射資料庫中可以快速的判斷多維度序列是否為封閉多維度序列，但建立投射資料庫也會花一些時間和佔用記憶體空間。因此，若能找到更好的方法，不去建立投射資料庫，而是從一些資訊來做快速的判斷，將更能夠縮短探勘的時間和空間。

參考文獻

[1] Agrawal R. and Srikant R., "Fast Algorithm for Mining Association Rules", ***Proc. of International Conference on Very Large Data Bases***, September, pp. 487–499, 1994.

[2] Han J., Wang J., Lu Y. and Tzvetkov P., "Mining Top-k Frequent Closed Patterns without Minimum Support", ***Proc. 2002 Int. Conf. on Data Mining (ICDM'02)***, pp211-218, 2002.

[3] Lucchese C., Orlando S. and Perego R., "DCI-CLOSED: A Fast and Memory Efficient Algorithm to Mine Frequent Closed Itemsets", ***Proc. Of IEEE ICDM workshop on Frequent Itemset Mining Implemention (FIMI'04)*** , Vol. 126, 2004.

[4] Li L., Zhai D. and Jin F., "GRG: An Efficient Method for Association Rules Mining on Frequent Closed Itemsets", ***Proc. of the 2003 IEEE Int. Sym. On Intelligent Control (ISIC'03)*** , pp.854-859, 2003.

[5] Pasquier Nicolas, Bastide Yves, Taouil Rafik and Lakhal Lotfi, "Efficient mining of association rules using closed itemset lattices", ***Proc. of the Information Systems***, pp. 25-46, 1999.

[6] Pasquier N., Bastide Y., Taouil R. and Lakhal L., "Discovering Frequent Closed Itemsets for Association Rules", ***Proc. of the 7th International Conference on Database Theory***, pp. 398–416, 1999.

[7] Pei J., Han J. and Mao R., "CLOSET: an Efficient Algorithm for Mining Frequent Closed Itemsets", ***Proc. 2000 ACM-SIGMOD Int. Workshop Data Mining and Knowledge Discovery (DMKD'00)***, pp.21-30, 2000.

[8] Pinto H., Han J., Pei J., Wang K., Chen Q. and Dayal U., "Multi-dimensional Sequential Pattern Mining", ***M. Sc. Thesis, Simon Fraser University***, Canada, 2001.

[9] Songram P. and Boonjing V., "Efficient Algorithms for Mining Closed Multidimensional Sequential Patterns", ***Proc. Of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)***, pp 749-753, 2007.

[10] Songram P., Boonjing V. and Intakosum S., "Closed Multidimensional Sequential Pattern Mining", ***Proc. Of the 3rd IEEE Int. Conf. on Information Technology: New Generations (ITNG'06)***, Las Vegas, Nevada, USA, pp.512-517, 2006.

[11] Tzvetkov P., Yan X. and Han J., "TSP: Mining Top-k Closed Sequential Patterns", ***Proc. of the 3rd IEEE Int. Conf. on Data Mining (ICDM'03)***, pp.347-354, 2003.

[12] Wang J. and Han J., "BIDE: Efficient Mining of Frequent Closed Sequences", ***Proc. of the 20th IEEE Int. Conf. on Data Engineering (ICDE'04)***, pp.79-90, 2004.

[13] Li L., Zhai D. and Jin F., "GRG: An Efficient Method for Association Rules Mining on Frequent Closed Itemsets", ***Proc. of the 2003 IEEE Int. Sym. On Intelligent***

Control (ISIC'03) , pp.854-859, 2003.

[14] Yan X., Han J. and Afshar R., "Clospan: Mining Closed Sequential Patterns in Large Database", ***Proc. of the SIAM Int. Conf. Data Mining***, pp.166-177, 2003.

[15] Zaki M.J. and Hsiao C.J., "CHARM: An Efficient Algorithm for Closed Itemsets Mining", ***Proc. 2002 SAIM Int. Conf. Data Mining (SDM'02)***, pp.457-473, 2002.