應用以約定值為基礎之演算法於關聯規則探勘

葉進儀
嘉義大學資訊管理學系

林彣珊
嘉義大學資訊管理學系

郭文熙
嘉義大學資訊管理學系

摘要

現存於大型資料庫的關聯規則探勘方式，大都利用支持度修剪策略來降低搜尋關聯規則的時間，但此策略於低支持度門檻時，無法有效的找出潛在有價值的樣式，而且因為支持度太低，導致額外的資源（例如記憶體）需求也過大；在高支持度門檻時，則會遺失具有低支持度，但卻有高信賴度與高相關性的樣式。本研究先證明約定值具有跨支持度特性，然後再利用此特性修剪及刪除沒有價值的項目集，加快演算法的執行速度與節省系統的資源，而且如果一個項目集其約定值大於最小約定值門檻，則這一個項目集的支持度會大於某一個程度的底限，由此項目集所延伸出來的關聯規則，其信賴度也會大於某一程度的底限，因此利用約定值所探勘出來的關聯規則是有價值的。本研究最後將此演算機制應用於真實之交易資料上，實驗結果顯示利用約定值跨支持度特性的修剪策略可以減少尋找大型項目集的時間，且所探勘出的大型項目集，其項目間也具有高度的相關性。

關鍵字：資料探勘，關聯規則，約定值，跨支持度

Applying Bond-based Algorithm for Mining Association Rules

Jinn-Yi Yeh
Department of Information Management, National Chiayi University

Wen-San Lin
Department of Information Management, National Chiayi University

Wen-Hsi Kuo
Department of Information Management, National Chiayi University

Abstract

Most current methods of mining association rules for large database use support pruning strategy to reduce searching space of finding out association rules. However, the strategy is not efficient to mine valuable patterns because it consumes lots of resources when the support threshold is low. Meanwhile when the support threshold is high, it will lose valuable itemsets which have lower support, higher confidence, and higher correlation. This paper applies the concept of bond-based threshold to mine association rules for large databases. We first prove that the bond has a cross-support property and then use this property to prune invaluable itemsets. This can improve the efficiency of the algorithm and reserve system resources. If the bond of itemset is greater than the bond-based threshold, the support of this itemset would be greater than some limit. The confidence of the association rules produced by the itemset would also be greater than some limit. The itemset would have high correlation between individual items. Therefore, when we use both bond and support pruning strategy, the association rules will be valuable. Our experiments were performed on real data sets. The experimental results show that this approach can reduce search space and find the valuable patterns, and the valuable patterns have high correlation between individual items.

Keywords: data mining, association rules, bond, cross-support

壹、緒論

資料探勘被定義為「在一個大量的資料集裡，搜尋出有用或者是有價值的潛在資訊」，且資料探勘並非一次性的活動，經過資料探勘分析後的樣式，仍然必須要經過專家判定是否具有價值，因為也許該樣式結果雖為真實的，但是卻並非具有價值，這樣的樣式必須被刪除(Hand et al., 2000；曾憲雄等人，2004)。資料探勘是由許多專業學科所匯合而成，例如包含有統計、資料探索分析、機器學習、資料庫等，而且區分為多種技術，可分別應用在不同的領域裡，例如有類神經網路、決策樹、基因演算法、模糊理論等技術，應用在分類及分群等這些領域，其目的是為了找出有價值的資訊，而找出的資訊可以將之稱為樣式(pattern)，樣式被定義為「在資料集裡的一個獨特的關聯」。

而關聯規則探勘(association rules mining)為資料探勘的一種應用，時常被應用在找尋產品之間的關聯(彭文正，2001)。大部分的關聯規則探勘演算法是以支持度為基礎的修剪策略(support-based pruning strategy)，以減少搜尋空間，達到降低搜尋有價值項目集(itemset)的時間之目標，然而利用此策略所探勘出的項目集，有某一部分其項目間的相關性並不高，且當面對具有歪斜支持度分配的資料集時，有可能探勘效率會降低，原因有二：(1)如果最小支持度門檻值設定過低，有可能會挖掘出很多沒有用的樣式(pattern)，而這些樣式含有的項目(item)，其個別支持度差距是很大，這些樣式被稱為「微弱相關性的跨支持度(cross-support)樣式」(Xiong et al., 2006)，這些樣式之所以沒有用，是因為其項目和項目之間的相關性是很低，會被挖掘出來只是因為其中一個項目的支持度很高，導致另一個支持度低的項目和此項目常常共同發生，此外降低支持度的門檻也會增加計算及記憶體的使用成本；(2)如果最小支持度門檻值設定過高，則有可能會遺失掉具有低支持度的有價值樣式，通常出現在被購買頻率不高，卻有高收益的產品上。

雖然有學者提出了可以解決在低支持度門檻時，會耗費很多資源與時間的關聯規則探勘演算法，例如Liu et al. (1999) 提出可以讓使用者針對不同的項目使用不同的支持度門檻值，來解決低支持度門檻值時耗費資源的問題，但是這些演算法並沒有解決跨支持度樣式以及項目集的項目間相關性過低的問題；Omiecinski (2003) 則提出了以約定值(bond)為基礎的修剪策略，雖然約定值可以探勘出高度相關性的項目集，然而卻忽略了探勘出的項目集佔資料庫總交易筆數的比例，致使探勘出的項目集雖然具有高度相關性，但可能會因為該項目集被購買的頻率太低而沒有商業上的價值。另外過去不具有跨支持度特性的關聯規則探勘演算法，使用了沒效率的方式來刪除跨支持度樣式，第一步先將最小支持度門檻調低，等探勘出所有樣式後，第二步再將跨支持度樣式刪除，然而這樣的方式會因為最小支持度門檻值低，所探勘出來的樣式數量也會急遽升高，導致計算所需花費的資源可能會過高，執行時間也會急劇增加，且探勘出的項目集，項目間的相關性仍然很低。

為了解決上述問題，本研究提出了一個應用約定值概念的演算法，先證明約定值(bond)(Omiecinski, 2003)具有跨支持度的特性，可以有效的幫助刪除不具價值的樣式，而這些被刪除的樣式，其部份項目和項目間的支持度差異很大，接著更證明利用約定值

探勘出的項目集，其項目和項目間的相關性會大於最小約定值門檻，因此所探勘出的項目集比只單用支持度門檻所探勘出的項目集更具價值。並且藉由利用約定值的跨支持度特性刪除這些不具價值的樣式，可以節省系統所需花費的資源與縮短系統執行所需的時間。

本研究的架構如下：第一章節緒論，說明研究動機及研究目的；第二章節相關文獻探討，說明何謂資料探勘以及對於關聯規則探勘的一些基本定義與名詞解說，接著並探討了關聯規則探勘的幾個重要特性與方法；第三章節研究方法，介紹了本研究所採用的關聯規則探勘方法與特性，並詳細說明系統的內容、流程和步驟，接著會詳細探討本研究所使用的演算法，並說明其特點；第四章節實驗分析與效率評估，呈現實驗的成果；最後在第五章節做最後的總結與未來研究方向的探討。

貳、文獻探討

本章節介紹過去到現在關聯規則探勘的一些特性，接著介紹本研究演算法使用到的兩個重要特性，分別是「候選項目集的產生方式」與「反單調性(anti-monotone)」。

一、關聯規則探勘

關聯規則探勘又稱為購物籃分析(market-basket analysis)，可以在大量的交易資料中，挖掘出產品之間的關聯性，挖掘這些有價值的關聯規則，可以將之用於行銷方面產品的交叉銷售、即時輔助購買決策、DM名單、貨品的架位安排等，以增加企業的銷售業績。Blischok (1995) 認為現在是顧客導向的商業，較好的顧客服務來自於較多的顧客知識，我們必須透過銷售交易資料來了解顧客的購買行為，購物籃分析就是其中一種了解顧客購買行為的方法。

要找出關聯規則有許多方式，例如藉由建構決策樹找出關聯規則；Agrawal et al. (1993)另外提出了頻繁項目集(frequent itemsets)的概念，他們先找出所有的頻繁項目集，再利用這些頻繁項目集建立關聯規則。自從他們提出頻繁項目集的概念後，這個概念開始在資料探勘和知識發現(knowledge discovery)領域受到注意，且廣泛的被使用，例如棉被、床單、枕頭，可能就會是一個頻繁項目集，因為這些商品常常會同時一起被購買。本研究的目的，就是要找出這些有價值的頻繁項目集。以下將介紹關聯規則的背景知識、項目集的搜尋方式、資料表達方式、一般關聯規則探勘步驟，以及其他關聯規則的相關研究。

（一）背景知識

市場購物籃的資料是由許多的交易(transaction)所組成，每一筆交易，則是由多個項目所組成，這裡所指的項目即為產品。一般定義$I=\{i_1,i_2,\mathrm{K},i_m\}$是所有項目的集合，表資料庫中全部共有$m$個項目，$D$表示一個交易資料庫(transaction database)，為交易資料的集合，其中每一筆交易T皆是I的一個子集合，而每一筆交易皆會給予一個唯一的交易識別

碼稱之為TID(transaction identifier)。交易資料庫主要紀錄商業交易上的交易資料，大部分的交易資料乃採用關聯式資料庫的架構來記錄資料，因此交易資料庫可以說是關聯式資料庫的一個特例。

項目集為一些項目的集合，其中每一個項目皆為I的其中一個元素，若該項目集包含有k個項目，則我們以k-項目集(k-itemset)來表示。關聯規則的表達方式$X \rightarrow Y$，表示X關聯到Y，其中$X,Y \subseteq I$，且$X \cap Y = 0$，而X又稱作條件句(antecedent)，Y又稱作結論句(consequent)，一個關聯規則$X \rightarrow Y$的支持度定義為：在D中，包含有X和Y的所有交易佔D的全部交易百分比，又稱為覆蓋率(coverage)，是統計重要度的測量。一個關聯規則$X \rightarrow Y$的信賴度定義為：在D中，包含有X和Y的所有交易佔包含有X的所有交易的百分比，又稱為準確性(accuracy)，是效度的測量。支持度與信賴度的公式如下(Han and Kamber, 2001)：

$$support(X \rightarrow Y) = \frac{|XY|}{|D|} = P(X \cup Y) \tag{1}$$

$$confidence(X \rightarrow Y) = \frac{|XY|}{|X|} = P(Y \mid X) \tag{2}$$

其中｜XY｜代表「在交易資料庫D中，包含有X和Y的交易筆數」，｜X｜代表「在交易資料庫D中，包含有X的交易筆數」，｜D｜代表「交易資料庫D的總交易筆數」，$P(X \cup Y)$代表「在交易資料庫D中，X和Y共同出現的機率」，$P(Y \mid X)$代表「在交易資料庫D中，當樣本空間為X時，Y出現之機率」。給定一個交易資料集合D，在資料探勘中，關聯規則探勘的工作就是要找出所有滿足使用者自行定義之最小支持度(min_support)和最小信賴度(min_confidence)門檻的有價值關聯規則。

（二）項目集的搜尋方式

丁一賢與陳牧言(2005)提到關聯規則探勘演算法項目集的搜尋方式，主要可以分成廣度優先搜尋(breadth-first search; BFS)和深度優先搜尋(depth-first search; DFS)，在關聯規則演算法中，大部分都是採用廣度優先搜尋的方法，此種搜尋方式在產生大型k-項目集之前，所有的大型(k-1)-項目集就已經先被產生了，如此一層一層向下執行，直到沒有大型k-項目集的產生為止。最著名的Apriori演算法(Agrawal et al., 1994)就是使用廣度優先搜尋的演算法之一；深度優先搜尋方法乃是以遞迴的方式隨著由項目集矩陣而來的樹狀結構，由上到下的搜尋並計算項目集的支持度，著名的FP-Growth演算法(Han et al., 2004)就是利用此種搜尋方法。本研究採用廣度優先搜尋法來找出具有價值的項目集。

（三）資料的表達方式

關聯規則探勘的交易資料庫資料表達方式有兩種，分別為水平式與垂直式，主要是針對計算項目集支持度的方法來區分。表1為交易資料水平式表達方式，在水平式的表達方式中，每一筆交易具有一群項目，而資料庫包含了多筆的交易，此種資料表達方式，當要計算某一個項目集的支持度時，必須要去掃描整個交易資料庫，當某一筆交易包含有該項目集時，就將累計值加一，直到資料庫所有交易資料皆掃描完為止，即可得知該

項目集的支持度，常見的Apriori演算法(Agrawal et al., 1994)與FP-Growth演算法(Han et al., 2004)皆使用此種資料表達方式。

表2為垂直式表達方式，在垂直的表達方式中，每一個項目具有一群交易識別碼(又稱為tidset或tidlist)，表含有該項目的所有交易，通常項目中的交易識別碼列表都是以升冪的方式儲存，以提昇整體的計算效率，著名的Partition演算法(Savasere et al., 1995)與Eclat演算法(Ogihara and Li, 1997)皆使用此種資料表達方式。

表1：交易資料水平式表達方式

TID	項目
1	4, 5
2	1, 2
3	2
4	1, 2 , 3
5	1, 2, 3, 4
6	1, 2
7	1, 3
8	1, 3, 6
9	1, 2, 3, 4, 5 ,6
10	1, 2, 3

表2：交易資料式垂直表達方式

項目	TID集合
1	2, 4, 5, 6, 7, 8, 9, 10
2	2, 3, 4, 5, 6, 9, 10
3	4, 5, 7, 8, 9, 10
4	1, 5, 9
5	1, 9
6	8, 9

（四）關聯規則探勘步驟

關聯規則探勘的步驟，一般主要有兩個，目的是為了找出高於最小支持度和最小信賴度門檻的有價值關聯規則：(1)找出所有滿足(大於或等於)最小支持度門檻的項目集。一個項目集的支持度代表著有多少交易包含了此項目集，滿足此最小支持度的項目集，稱之為大型項目集(large itemsets)或頻繁項目集，而其他未滿足的則稱為小型項目集(small itemsets)；(2)利用上一階段得到的大型項目集，產生關聯規則。假設有一大型項目集X，且任一X的子集合為Y，若要產生規則$(X-Y)\rightarrow(Y)$，則必須滿足$support(X)$除以$support(X-Y)$大於或等於最小信賴度門檻，同時考量X的所有子集合來產生符合門檻的各種規則。過去到現在大部分的研究，皆集中在比較困難且重要的第一步驟，找出大型項目集或頻繁項目集，本研究即利用約定值與支持度兩個測量方式來產生高度相關性大型項目集。

（五）關聯規則的相關研究

其它的關聯規則相關研究，包含有開發新的關聯規則測量方式、數量化的關聯規則探勘、多層次關聯規則探勘、改善關聯規則探勘的效能，或是結合其他方法或特性產生具有特定目的之關聯規則等，例如Park et al. (1995) 提出了DHP(direct hashing and pruning)演算法，此演算法從兩方面著手加快大型項目集的產生，首先以雜湊技術減少初期候選項目集的產生個數，接著隨著演算法的執行，逐漸刪減交易資料庫，刪減的方式有二：(1)刪除資料庫中在下一階段無用的交易資料；(2)刪除單一筆交易在下一階段無用的產品項目；Brin et al. (1997) 提出了DIC(dynamic itemset counting)演算法，此演算法每一次只讀取交易資料庫的M筆資料，而不一次讀取整個交易資料庫，因此若想計算某一項目集的支持度，則任何時候都可以開始計算，不需等到下一輪資料庫重新讀取，藉此可以減少讀取整個交易資料庫的次數，達成縮短探勘時間之目的。

Lawrence et al. (2001) 則利用顧客之前的購買行為開發出了個人化的產品推薦系統，其功能為自動推薦新產品給顧客，顧客可以使用PDA來接受（推薦）或傳送（交易）資訊，他利用將顧客過去的購買花費標準化為向量，再將顧客作分群，然後從每一個分群中找出最常被購買的產品組成熱門產品目錄，再配合依顧客歷史交易記錄探勘出來的產品類別或子類別關聯規則作結合，算出顧客和某類產品的配對分數，依此分數，就可以產生個人化的推薦目錄，可以針對目標顧客作推薦的動作；Brijs et al. (2004) 整合了「大型項目集的發掘」和「個體經濟學模組」成為「PROFSET模組」，此模組考量到了關聯規則的商業價值。

另外，在某些情況下，如果將關聯規則探勘區域化，可能會比利用全域資料探勘的關聯規則更具價值，例如先將交易資料做分群(clustering)，之後再針對個別群組做關聯規則探勘的動作，如此可以產生區域關聯規則(localized association rules)，如Aggarwal et al. (2002) 利用一個相當彈性的分群演算法CLASD(clustering for association discovery)，先將資料做分群，然後在已分群的資料裡找出區域關聯規則，如此可以找到利用未分群資料做分析時，所無法找到的區域關聯規則，這樣的一個區域關聯規則，又可以被稱做為個人化的關聯規則，特別適合用在目標市場上。

二、候選項目集的產生方式

Agrawal and Srikant (1994) 提出的Apriori和AprioriTid演算法，其產生大型項目集的方式，必須經過好幾個階段，符號L_k用來表示所有大型k-項目集(large k-itemset)所形成的集合，一開始此演算法先掃描整個交易資料庫，計算每一個單一項目的支持度來決定哪些項目的支持度大於或等於最小支持度門檻，並將這些大型1-項目集儲存在L_1，之後的每一階段，皆利用前一階段所產生的大型項目集產生新的潛在大型項目集，又稱之為候選項目集(candidate itemsets)，並計算這些候選項目集的支持度，刪除小於門檻值的項目集，而留下來的大型項目集就再成為下一階段產生候選項目集的輸入。這樣一個流程，利用L_1產生L_2，L_2產生L_3，一直持續到沒有大型項目集的產生為止。

假設所有「大型$(k-1)$-項目集」儲存在L_k-1、「候選k-項目集」儲存在C_k，且所有

的項目集中的項目以項目編號升冪方式排序完成，Apriori和AprioriTid演算法利用一個apriori_gen的函式帶入參數L_{k-1}來產生候選k-項目集並儲存在C_k，產生的方式為結合兩筆大型$(k-1)$-項目集，假設有兩筆大型$(k-1)$-項目集分別為p、q，結合的方式如下：(1)首先必須比對大型$(k-1)$-項目集p、q的前$(k-2)$個項目編號是否皆相等，若成立，則再進一步比對p的第$(k-1)$個項目之編號是否小於q的第$(k-1)$個項目之編號，若亦成立，則結合p的所有項目與q的最後一個項目成為候選k-項目集，並將此候選k-項目集加入到C_k中；(2)接著在候選k-項目集合C_k中，找出所有候選k-項目集其任一$(k-1)$項目子集合不包含在L_{k-1}，並將這些候選項目集刪除，刪除後的C_k即為此階段產生的候選k-項目集。

Omiecinski (2003) 提出了另外兩種新的關聯規則測量方式，分別為all-confidence和約定值，此兩種測量方式皆可以找出利用支持度所無法找到的關聯規則，他利用交易串列(tidlist)的交集(intersection)與聯集(union)來計算all-confidence和約定值，交易串列是由一群交易識別碼所組成，而這些交易識別碼所代表的交易皆必須包含某個相同的項目集，其產生候選項目集的方式和Apriori演算法相同；Song et al. (2006) 則提出了TM(transaction mapping)演算法，在這個演算法中，利用交易樹(transaction tree)使每一個項目集的交易識別碼皆被對應、歸納到某一個連續的交易區間，而每一個項目集的支持度就是利用這些區間做交集運算得出。TM演算法改變了支持度的計算方式，不需要去掃瞄整個資料庫的所有交易，並利用前序式詞彙樹尋訪法(lexicographic prefix tree)資料結構來產生候選項目集與使用深度先尋找出大型項目集。本研究採用了Aprori演算法的apriori_gen函式來產生候選項目集，以避免候選項目集的重複產生。

三、具反單調性的關聯規則探勘

「反單調性(anti-monotone; downward closure)」定義為：假設有某一個項目集並沒有滿足所規定的門檻，則任何一個該項目集組成的超集[1](superset)也不會滿足所規定的門檻，相反的，若某一個項目集滿足了所規定的門檻，則任意該大型項目集的子集合必定會滿足所規定的門檻。由於Apriori演算法的候選項目集是由上一階段的大型項目集延伸而來，例如候選k-項目集是由兩個大型$(k-1)$-項目集做結合，然後再刪除其$(k-1)$-項目集子集合非為大型項目集的候選項目集，接著利用支持度來判斷是否為大型k-項目集，因此具有反單調性，也就是說若一個項目集其為大型項目集，則任意其子集合也必定為大型項目集，反之若一個項目集不為大型項目集則其任意超集也必定不是大型項目集。

接著本研究說明利用支持度來進行測量的方式具有反單調性，絕對值符號代表筆數：假設X為一大型項目集，則X的支持度為：$support(X) = |X|/|D|$，另外Y為X的一個子集合，則Y的支持度為：$support(Y) = |Y|/|D|$。由於Y為X的子集合，所以Y的項目數小於X的項目數，因此包含有Y的交易筆數會大於等於包含有X的交易筆數，即可證得：$support(Y) \geq support(X) \geq min\_support$。

[1] 超集為某一集合的延伸，假設有兩集合X、Y，若$Y \supseteq X$，則Y為X的超集，反之X為Y的子集。

由此可知，若Y為X的一個子集合，當X為大型項目集時，Y也必定會為大型項目集，因此利用支持度來進行測量的方式具有反單調性。Omiecinski (2003) 提出的all-confidence和約定值測量方式也都具有反單調性；all-confidence定義為在交易資料庫D中，某一項目集X出現的交易次數除以X的所有子集合出現的交易次數之最大，即all-confidence = $|X|/Max|Y_i|$，因此X的all-confidence等於X所產生的所有關聯規則之信賴度的最小，所以若一個項目集X是大型項目集，則表示X所產生的所有關聯規則之信賴度皆大於最小all-confidence門檻，則X的任意子集合也會是大型項目集；Xiong et al. (2003a; 2003b; 2006) 提出了h-confidence的測量方式，h-confidence為改良了all-confidence而來，因此h-confidence也具有反單調性，雖然二者在數理上是相同的，但是它們的定義卻是不同的，all-confidence檢查了給定之項目集的所有關聯規則，但是h-confidence只檢查給定之項目集的所有「條件句」只有單一個項目的關聯規則，假設若有一筆交易P，則h-confidence只檢查了所有$\{i\} \rightarrow P-\{i\}$，其中左手邊的$i$表示交易$P$的單一個項目。利用反單調性可以有效的刪除不具價值的項目集，進而減少候選項目集的產生個數，加快演算法的執行效能，而本研究所採用的約定值也具有此一特性。

參、約定值為基礎之演算法

本研究利用約定值(bond)的概念，加入到探勘大型項目集的演算流程，藉此達成節省整體資源，降低執行所需的時間，以及探勘出更具價值的項目集之目的。本章節將對約定值的定義與特性詳細的介紹，再說明研究流程與內部運作，最後會探討詳細的演算過程。

一、約定值之定義與特性

假設$L=\{l_1,l_2,\mathrm{K},l_n\}$為一些項目的集合，即項目集，且 ；$T(L)$表示在$D$中包含有$L$全部項目的所有交易，若括號中為單一個項目，則表示在D中包含有該項目的所有交易；$|\cdots|$絕對值代表交易筆數。約定值原始的定義為：在資料庫D中包含有L全部項目的所有交易，除以包含有任意L子集合的所有交易取聯集。本研究將約定值的公式分母部份做了一些修改，以利更快速的計算出結果，修改後之公式如定義一所示。

定義一：一個項目集L的約定值計算公式定義為

$$bond(L)=\frac{|T(L)|}{|T(l_1)\cup T(l_2)\cup\cdots\cup T(l_n)|}\quad ,\ 1\le n\le m \tag{3}$$

其中n為項目集L的項目個數，m為資料庫D的總項目個數。若一個項目集的約定值越趨近於1，代表著該項目集中的所有項目有很大的機率會共同發生，反之若一個項目集的約定值越遠離1，則代表著該項目集中的所有項目不常共同發生。獲得約定值之後，可以由定義二了解到一個項目集該如何判斷它是否是有趣或是有價值。

定義二：給定一個最小約定值門檻*min_bond*，若一個項目集*L*是有價值的，則必須符合*bond(L)* ≥ *min_bond*，反之項目集*L*則為無價值的。

門檻值*min_bond*可以自行設定，端看自身的需求決定。在下述輔助特性一中說明約定值的測量方式具有反單調性。

輔助特性一：約定值的測量方式具有反單調性。假設有一項目集L，且$bond(L) \geq min_bond$，則可以知道任意L的子集合$L'=\{l_i,l_j,\mathrm{K},l_k\}$的約定值也必定大於或等於約定值門檻*min_bond*，更明確的表達，即如果

$$bond(L)=\frac{|T(L)|}{|T(l_1)\cup T(l_2)\cup\cdots\cup T(l_n)|}\geq min\_bond \tag{4}$$

則 $\forall L'\subset L$

$$bond(L')=\frac{|T(L')|}{|T(l_i)\cup T(l_j)\cup\cdots\cup T(l_k)|}\geq min\_bond \tag{5}$$

證明：由於$L'\subset L$，可以知道公式(4)、(5)分子部分，在D中包含有L'的交易筆數必定大於L，因此｜*T(L')*｜ ≥ ｜*T(L)*｜，而其分母部分，因為L'的項目數小於L的項目數，所以$|T(l_i)\cup T(l_j)\cup\cdots\cup T(l_k)|\leq|T(l_1)\cup T(l_2)\cup\cdots\cup T(l_n)|$。由此可以證得$bond(L') \geq bond(L) \geq min_bond$ 輔助特性一成立。

相反的，可以利用此特性了解到，若一個項目集L其任意子集合，只要有其中一個不為大型項目集，則此項目集L可以從候選項目集中被刪除。利用約定值探勘出的項目集，其項目與項目之間具有高度的相關性，以下將證明約定值的此一特性。給定一個項目集$L=\{l_1,l_2\}$，L具有兩個項目分別為l_1與l_2，本研究與Xiong et al. (2006) 同樣利用*cosine*相似性測量來計算項目與項目之間的相關性，公式(6)為*cosine*相似性測量之計算方式：

$$cosine(L)=\frac{support(\{l_1,l_2\})}{\sqrt{support(\{l_1\})support(\{l_2\})}} \tag{6}$$

於下述輔助特性二中，將證明當項目數為二，此特性成立。

輔助特性二：假設有一個項目集$L=\{l_1,l_2\}$，具有兩個項目，則當此項目集的約定值大於最小約定值門檻*min_bond*，則我們可以知道此項目集的相似性$cosine(L) \geq min_bond$，必定也會成立。

證明：由定義一可以知道$bond(L)=|T(L)|T(l_1)\cup T(l_2)|$，假設$support(\{l_1\})\geq support(\{l_2\})$，若$L$為一大型項目集，則由定義二得知$bond(L)\geq min_bond$，又因為$support(\{l_1\})\geq support(\{l_2\})$，因此其兩項目之間的相似性為

$$cosine(L)=\frac{support(\{l_1,l_2\})}{\sqrt{support(\{l_1\})support(\{l_2\})}}\geq\frac{support(\{l_1,l_2\})}{support(\{l_1\})}=\frac{|T(L)|}{|T(l_1)|}\geq bond(L)$$
$$=\frac{|T(L)|}{|T(l_1)\cup T(l_2)|}\geq min\_bond \tag{7}$$

輔助特性二成立。

輔助特性二說明：若約定值門檻夠高，則當某一項目數為2的項目集，其約定值大於約定值門檻時，則該項目集的兩個項目之間的相似性也會大於約定值門檻，表示兩項目間的相關性很高。接著下述特性一將證明當項目集的項目數大於2時，若該項目集的約定值大於約定值門檻(*min_bond*)，則該項目集兩項目之間的*cosine*相似性亦會大於約定值門檻(*min_bond*)，表示兩項目之間具有某一程度的相關性。

特性一：給定一個項目集$L=\{l_1,l_2,K\ ,l_n\}$，且n大於2，若項目集L的約定值大於或等於約定值門檻(*min_bond*)，則可以知道L中任意兩兩項目$P=\{l_1, l_j\}$，$P \subset L$，其*cosine*相似性之值*cosine(P)*必定也會大於等於約定值門檻(*min_bond*)。

證明：由輔助特性一的反單調性得知，若項目集L的約定值大於或等於約定值門檻(*min_bond*)，則其任意子集合的約定值也必定大於或等於約定值門檻(*min_bond*)，又因為$P \subset L$，因此所有$P=\{l_1, l_j\}$之約定值也必定大於或等於約定值門檻(*min_bond*)，接著由輔助特性二的証明可以知道，項目集L的任意兩兩項目P，其*cosine*相似性之值*cosine(P)*必定也會大於等於約定值門檻(*min_bond*)。

特性一說明了利用約定值探勘出的項目集，其兩兩項目間具有某一程度的相關性，而只利用支持度探勘出的項目集，項目間的相關性則會比較小，因此當本研究將約定值的概念加入演算法後，所探勘出的項目集會更具價值。Omiecinski (2003) 另外提到一個約定值的重要特性，如果一個項目集L的約定值滿足了最小約定值門檻，則任意由此項目集所產生的關聯規則之信賴度必定大於或等於最小約定值門檻，因此可以知道由約定值所探勘出來的關聯規則是具有價值的，而先前本研究提到，若將支持度門檻提高時，會遺失掉那些雖然具有低支持度，但卻有著高信賴度的樣式，現在將約定值的概念加入演算流程中，則當將支持度門檻降低，可以藉著約定值的特性，使探勘出的關聯規則，皆具有高信賴度。

二、跨支持度特性(cross-support property)

若一個樣式具有跨支持度特性，即意味著該樣式所包含的項目中，有某兩個項目的支持度差距過大，導致其不具價值，Xiong et al. (2006) 對跨支持度特性作了明確的定義，詳細說明如定義三所示。

定義三：給定一個門檻值t，如果一個樣式P具有跨支持度特性，則必須滿足在P中，有兩個項目x和y，其中項目y的支持度大於項目x的支持度，且使得$support(x)/support(y)<t, 0<t<1$。

定義三利用$support(x)/support(y)<t$來判斷樣式P是否有跨支持度特性的原因在於，如果$support(x)/support(y)$小於所規定的門檻值t，且t大於0小於1，則表示兩支持度的比值小，所以兩支持度的差距大，因此具有跨支持度特性，假設有$\{a, b, f, g, x, y\}$等6個項目，圖1呈現出項目經過排序後的支持度折線圖，水平軸表示依支持度升冪排序的項

目，垂直軸表示支持度，若將門檻*t*設定為0.6，則樣式*{x, y, a}*為一個跨支持度樣式，因為該樣式包含有兩個項目*x*和*y*，其$support(x)/support(y)=0.3/0.6=0.5<t=0.6$，且還可看到項目*x*與項目*y*之間的支持度折線明顯的較為的陡峭，也可由此判斷可能具有跨支持度特性，因為$support(x)/support(y)$越小，斜率越大。接著可以利用項目*x*與*y*作為項目的分水嶺，將項目分為項目集*{b, g, x}*和項目集*{y, a, f}*，往後若有任何的樣式，同時具有這兩個項目集的項目，則可以判斷它具有跨支持度特性，直接予以刪除。

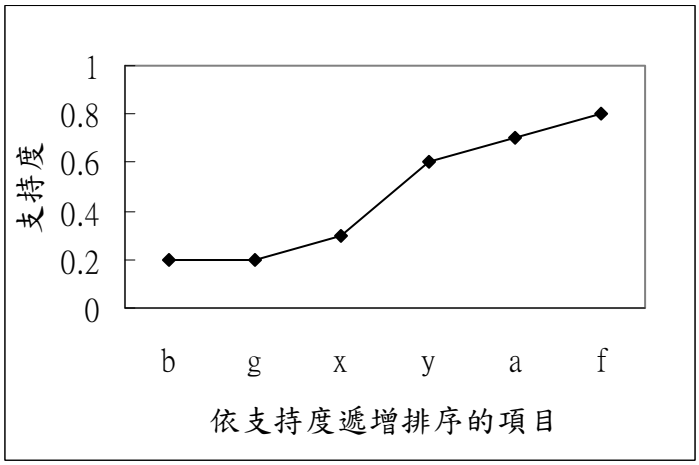

圖1：支持度折線圖

接下來證明約定值具有跨支持度的特性。意即在相同的門檻值*t*之下，若樣式*P*具有跨支持度特性，則利用約定值來測量時，樣式*P*也必定是無價值的樣式，於特性二作說明。

特性二：給定一個門檻*t*，如果一個樣式*P*，具有跨支持度的特性，則此樣式*P*也必定符合$bond(P)<t$。

證明：假設樣式*P={Λ ,x,Λ ,y,Λ}*，因為樣式*P*具有跨支持度特性，因此由定義三知道*P*包含兩個項目*x*和*y*，而且使得$support(x)/support(y)<t, 0<t<1$。

而

$$support(x)/support(y)=\frac{|T(x)|}{|D|}/\frac{|T(y)|}{|D|}=\frac{|T(x)|}{|T(y)|}<t \tag{8}$$

所以

$$\begin{aligned} bond(P)&=\frac{|T(P)|}{|\cdots\cup T(x)\cup\cdots\cup T(y)\cup\cdots|}\le\frac{|T(x)|}{|\cdots\cup T(x)\cup\cdots\cup T(y)\cup\cdots|}\\ &\le\frac{|T(x)|}{|T(y)|}=\frac{support(x)}{support(y)}<t \end{aligned} \tag{9}$$

特性二成立。

由特性二可以知道，若一個樣式P，其$support(x)/support(y)<t$，則$bond(P)<t$必定也會成立，因此只要先計算$support(x)/support(y)$是否小於t，即判斷$support(x)<t\times support(y)$是否成立，若成立，則可直接將此樣式刪除，而不需去計算較為繁複的$bond(P)$，以達成降低系統執行所需時間之目的，因為計算$bond(P)$需要用到數理上的聯集與交集。另外Cohen et al. (2001) 則提出了Jaccard相似度測量，雖然Jaccard概念上與約定值相似，因此亦具有跨支持度的特性，但Jaccard只能找出大型2-項目集，而約定值可以找出所有的大型k-項目集，因此比Jaccard更具價值。

參、系統流程與運作結構

圖2為本研究的系統流程圖，假設Li、Ck分別儲存「大型i-項目集」與「候選k-項目集」。此流程劃分為六個步驟，以下分別詳述：

（1）掃描整個資料庫的交易資料，並計算出每一個個別項目的支持度，如果該項目的支持度大於最小支持度門檻$min_support$，則將該項目加到候選集合C_1中，因為所有C_1裡的1-項目集，其約定值必為1，因此所有C_1裡的候選1-項目集，皆會為大型1-項目集。接著再將所有C_1裡的大型1-項目集儲存到L_1中，然後將L_1的大型1-項目集依支持度大小遞減作排序，再依項目編號遞減作排序。

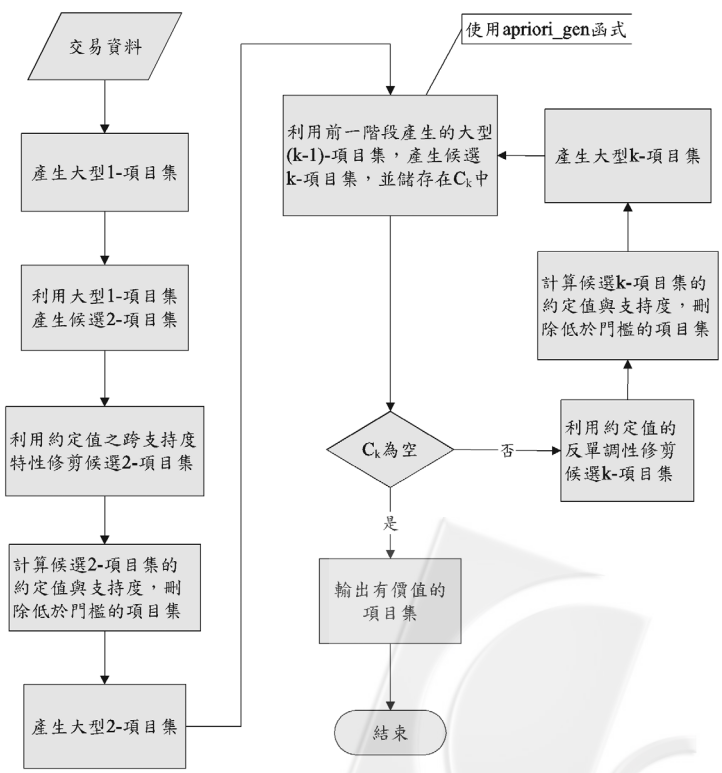

圖2：系統流程圖

（2）利用上一階段產生的大型*1*-項目集L_1*L_1，產生候選*2*-項目集C_2，接著執行約定值之跨支持度特性修剪候選*2*-項目集。由特性二可知，若給定一個約定值門檻*min_bond*，且某一候選*2*-項目集包含有項目*y*，則可以知道，如果該候選*2*-項目集的另一個項目*x*的支持度小於$support(y)\times min_bond$，則其必定為具有跨支持度特性的項目集，且其約定值也必定會小於約定值門檻*min_bond*，因此這樣的一個候選*2*-項目集可以直接被刪除。圖3為產生大型項目集之演算法運作情形。

（3）將上一階段所產生的大型*(k-1)*-項目集$L_{(k-1)}$代入apriori_gen函式，產生候選*k*-項目集，並儲存在C_k中，此函式產生候選*k*-項目集的方式則是利用$L_{(k-1)}$結合$L_{(k-1)}$以產生候選*k*-項目集C_k，如果C_k為空，則跳到步驟(6)，否則執行步驟(4)。

（4）利用約定值的反單調性修剪候選*k*-項目集；如果C_k中的候選*k*-項目集，其所有*(k-1)*-項目集子集合的任何一個不在$L_{(k-1)}$中，則刪除該候選k-項目集。

（5）計算C_k中每一個候選*k*-項目集的支持度和約定值，刪除支持度小於支持度門檻或是約定值小於最小約定值門檻的候選*k*-項目集。所有留下來的*k*-項目集為大型*k*-項目集，並將這些大型*k*-項目集儲存於L_k，做為下一階段的輸入。接著跳回到步驟(3)。

（6）輸出所有有價值的樣式。

舉例來說，表1為輸入的交易資料，表3輸入資料經過排序後的項目支持度，假設最小支持度門檻值為0.1，最小約定值門檻值為0.6。在第一步驟中，由於所有1-項目集的約定值必為1，且所有的支持度皆大於0.1，所以所有的1-項目集，必為大型1-項目集。接著利用將大型1-項目集L_1*L_1，產生候選2-項目集C_2，開始執行第二步驟約定值之跨支持度特性修剪候選2-項目集，以下分為3小項分別詳細描述：(1)利用約定值的跨支持度特性修剪候選2-項目集：由L_1中找到大型1-項目集{3}和{4}，滿足了$\text{support}(4) = 0.3 < 0.6 \times \text{support}(3) = 0.36$。因此可以將項目分為兩個項目集，分別為{1, 2, 3}和{4, 5, 6}，若有任何一個項目集同時具有這兩個項目集的項目，則其具有跨支持度的特性，可以直接將該項目集刪除。此步驟刪除了C_2的{1, 4}{1, 5}{1, 6}{2, 4}{2, 5}{2, 6}{3, 4}{3, 5}{3, 6}等項目集。若無跨支持度特性，則必須精確的計算出這些項目集的約定值再予以刪除，如此會花費大量的資源與時間；(2)利用約定值與支持度的最小門檻修剪項目集：剩下的候選2-項目集，其支持度皆大於或等於最小門檻值0.1，但是{2, 3}{4, 6}{5, 6}這三個候選2-項目集的約定值分別為0.444、0.333、0.333，皆小於最小約定值門檻0.6，因此被刪除；(3)剩餘的{1, 2}{1, 3}{4, 5}三個候選2-項目集，則被歸為大型2-項目集，作為下一階段的輸入。

第三步驟，將上一階段產生的大型2-項目集代入apriori_gen函式，會產生{1, 2, 3}這個候選3-項目集，並儲存於C_3，因為C_3不為空；接著執行步驟四，利用約定值的反單調性修剪項目集，由於{1, 2, 3}這個3-項目集的子集合{2, 3}並非為大型2-項目集，可以利用反單調性給予刪除，刪除後C_3為空，因此之後利用沒有任何元素的L_3產生之C_4亦會為空，停止程序的執行，並輸出結果。在此範例中共產生了三個大型項目集，分別是{1,2}{1,3}{4,5}。

表3：個別項目支持度

項目	支持度
1	0.8
2	0.7
3	0.6
4	0.3
5	0.2
6	0.2

產生大型項目集之程序

L_i = 所有長度為 i 的大型項目集之集合
C_k = 所有長度為 k 的候選項目集之集合
c(m) = 某一候選項目集的第 m 個項目
T[j] = 包含有項目 j 的交易串列
Totaltran = 總交易數
讀取交易資料庫
計算出所有單一項目的『交易串列』，以及所有單一項目的『支持度』
1) L_1={利用支持度計算出大型 1-項目集}
2) C_2={利用 L_1 計算出候選 2-項目集}
3) **for**(k=2 ; $C_k \neq 0$; k++) **do begin**
4) **if**(k!=2) **then**
5) //利用 ainti-monotone 特性修剪候選項目集
6) **for all** 候選項目集 $c \in C_k$ **do begin**
7) **if**(c 的任意子集合『(k-1)-項目集』不為大型項目集) **then**
8) 將 c 從 C_k 中刪除
9) **end if**
10) **end**
11) **else then**
12) //利用跨支持度特性修剪候選項目集
13) **for all** 候選項目集 $c \in C_2$ **do begin**
14) **if**(c 的第一個項目為 y，第二個項目為 x，使得
support(x) < min_bond × support(y)) **then**
15) 將 c 從 C_2 中刪除
16) **end if**
17) **end**
18) **end if**
19) //計算出剩餘 C_k 元素的精確約定值、支持度作修剪
20) **for all** 候選項目集 $c \in C_k$ **do begin**
21) c.約定值 = (T[c(1)]∩T[c(2)]∩…∩T[c(k)]) /
(T[c(1)]∪T[c(2)]∪…∪T[c(k)])
22) c.support = (T[c(1)]∩T[c(2)]∩…∩T[c(k)]) / Totaltran
23) **if**(c.約定值<min_bond || c.support<min_support) **then**
24) 將 c 從 C_k 中刪除
25) **end if**
26) **end**
27) $L_k = C_k$
28) $C_{(k+1)}$ = apriori_gen(L_k)
29) **end**
30) **return** $\cup_k L_k$

圖3：產生大型項目集之演算法

由於產生大型*1*-項目集時，大型*1*-項目集會先依支持度大小遞減作排序，再依項目編號遞減作排序，因此由大型*1*-項目集延伸而產生的所有候選*k*-項目集或是大型k-項目集，其個別項目集的*k*個項目元素，也會依照支持度大小遞減作排序，再依項目編號遞減作排序，因此當執行步驟(2)的利用約定值之跨支持度特性修剪候選*2*-項目集時，即第12行到第17行，該演算法只要去找出每一個候選*2*-項目集的第一個項目當作*y*，第二個項目當作*x*，然後去比較 $support(x) < min\_bond \times support(y)$ 是否成立，若成立則可將該候選*2*-項目集刪除。這是因為經過大型*1*-項目集的排序後，每一個延伸的候選*2*-項目集，其第一個項目的支持度會是該項目集的兩個項目中的支持度之最大，而第二個項目的支持度則會是該項目集的兩個項目中的支持度之最小，因此可以直接將第一個項目設為*y*，而第二個項目則直接設為*x*，再執行判斷。如此可以加快程式的執行，而不需要再去判斷哪一個項目的支持度較大，哪一個項目的支持度較小。

執行約定值之跨支持度特性修剪候選項目集步驟，只需在候選*k*-項目集之*k*值為2時才須執行，亦即當項目數為2的時候，是因為候選*2*-項目集在經過跨支持度特性修剪後，所有具有跨支持度特性的候選*2*-項目集皆會被刪除掉，因此之後延伸的所有*k*-項目集，已經不會再具有跨支持度特性了，所以此一特性之修剪，只須要在當*k*為2時執行。

另外在此演算法中，只需讀取一次交易資料庫，演算法就會記錄該交易資料庫每一個個別項目的「交易串列」和「支持度」，往後計算每一個k-項目集的支持度和約定值，只需利用交易串列的交集和聯集運算即可得出，如第19行到第26行，而不需再去重複讀取整個交易資料庫，因此可以加快程式的執行、節省系統資源。在實際運作執行中，大於1個項目以上的項目集，其交易串列並不會被記錄下來，以利系統資源的節省。因此對於某一項目集出現在幾筆交易資料中，本研究轉而利用將該項目集個別項目的交易串列取交集從而得出。

接著說明交易串列要如何運作，才能產生約定值與支持度。由上述範例中產生了{1, 2}、{1, 3}、{4, 5}三個大型2-項目集，再由表2得知大型2-項目集{1, 2}的交易串列為{2, 4, 5, 6, 9, 10}，而大型2-項目集{1, 3}的交易串列為{4, 5, 7, 8, 9, 10}，接著經由apriori_gen函式產生了候選3-項目集{1, 2, 3}，若要計算項目集{1, 2, 3}的交易串列，則必須將大型2-項目集{1, 2}與{1, 3}的交易串列作交集的動作，{2, 4, 5, 6, 9, 10}∩{4, 5, 7, 8, 9, 10}，因此可以得出交易串列{4, 5, 9, 10}，表示項目集{1, 2, 3}，出現在{4, 5, 9, 10}等4筆交易中，此即為約定值分子部分，接著若要計算約定值之分母，則必須將項目集{1, 2, 3}的個別項目之交易串列取聯集，由表2可知項目{1}的交易串列為{2, 4, 5, 6, 7, 8, 9, 10}，項目{2}的交易串列為{2, 3, 4, 5, 6, 9, 10}，項目3的交易串列為{4, 5, 7, 8, 9, 10}，經過聯集之後{2, 4, 5, 6, 7, 8, 9, 10}∪{2, 3, 4, 5, 6, 9, 10}∪{4, 5, 7, 8, 9, 10}，可以得到交易串列{2, 3, 4, 5, 6, 7, 8, 9, 10}，表示這9筆交易資料，至少會包含項目集{1, 2, 3}的任一個項目。接著再將交易串列交集後的交易筆數除以交易串列聯集後的交易筆數，即可得出約定值為4/9=0.444。

在實際運作執行中，大於1個項目以上的項目集，其交易串列並不會被記錄下來，以利系統資源的節省。因此對於某一項目集出現在幾筆交易資料中，本研究轉而利用將該項目集個別項目的交易串列取交集從而得出。例如在上述的例子中，項目集{1, 2, 3}，將利用項目{1}、{2}、{3}的交易串列取交集，{2, 4, 5, 6, 7, 8, 9, 10}∩{2, 3, 4, 5, 6, 9, 10}∩

{4, 5, 7, 8, 9, 10}，得出交易串列{4, 5, 9, 10}，表示項目集{1, 2, 3}，出現在{4, 5, 9, 10}等4筆交易中，亦可知道約定值之分子，且節省了系統資源。相同地，如果要計算一個項目的支持度，則只要將約定值之分子除以總交易數即可算出。

演算法第28行，使用了與Apriori演算法相同的的apriori_gen函式產生候選項目集$C_{(k+1)}$，雖然本演算法項目集裡的項目並不如同Apriori演算法以項目編號作排序，而是先以支持度大小遞減作排序，然後再以項目編號遞減作排序，但是仍然可以達成避免產生相同項目的候選項目集之主要目的，因此apriori_gen函式仍適用於本研究之演算法。

肆、實驗分析與效率評估

一、系統執行環境與實驗資料

本研究執行於個人電腦，系統架構於微軟Windows Server 2003作業系統，硬體資源方面，搭配有3.0GHz的CPU，和1GBytes的記憶體。本研究將演算法實驗於真實資料上，來源有二：(1)醫療藥品廠商的交易資料，資料特性為具有359項產品，和10,058筆交易，圖4為醫療藥品資料支持度分配圖，可以看出小部分產品出現頻率很高，但大部分則是很低，符合資料具有歪斜的特性；(2)比利時零售商的超級市場交易資料，資料特性為具有16,470項產品，和88,163筆交易，且平均每一筆交易具有13個產品項目(Brijs et al. 1999)，圖5則為超級市場資料支持度分配圖，亦符合資料具有歪斜的特性，此外該筆資料的產品項目之支持度大於0.1的個數只有70個，因此不在圖中繪出；在醫療藥品的交易資料方面，交易資料庫中具有「交易資料」、「產品編號」、「產品名稱」等三個資料屬性，至於超級市場的交易資料，則只具有「交易資料」、「產品編號」等兩種資料屬性，因此醫療藥品資料的實驗結果可以對應到真實的產品名稱，而超級市場的交易資料之實驗結果則無法對應到真實產品名稱。其中「交易資料」是以「水平的表達方式」儲存於交易資料庫中，因此本研究於演算法一開始，會將其轉為「垂直的表達方式」，以利於演算法的執行。

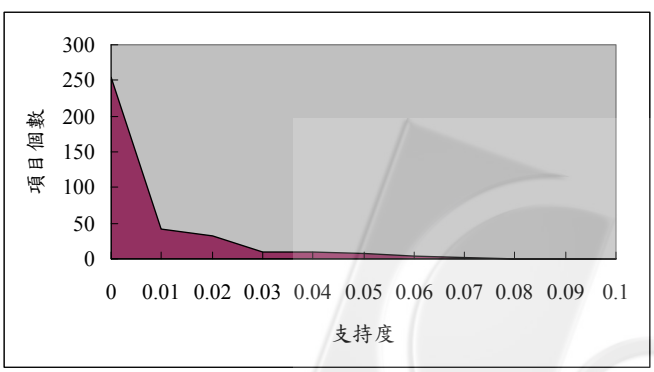

圖4：醫療藥品資料支持度分配圖

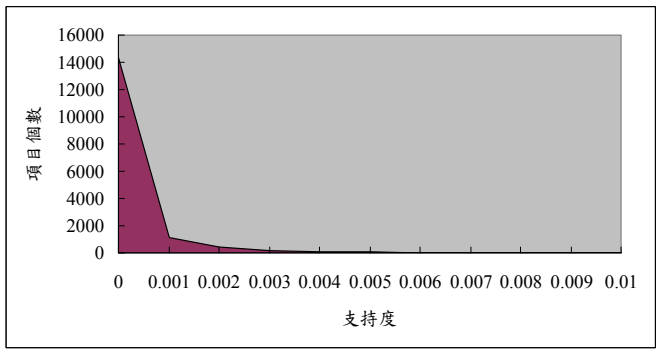

圖5：超級市場資料支持度分配圖

二、實驗評估

本研究的目的是要證明加入約定值此一評判項目集的標準，可以比以往只使用支持度門檻的演算法獲得更好的結果，以及更好的執行績效，因為約定值具有跨支持度特性，可以刪除具有此特性的無價值項目集，且利用約定值探勘出的項目集，其項目間的相關性會大於等於最小約定值門檻，而Omiecinski (2003) 也指出了利用約定值所探勘出來的關聯規則是具有價值的。當將約定值之門檻值*min_bond*設為0，則本研究演算法的執行結果將會和Apriori演算法之執行結果相同，因為當約定值之門檻值設為0，則表示跨支持度之門檻值*t*也會設為0，跨支持度特性之修剪策略也會無效，且所有的項目集的約定值皆會大於等於約定值門檻，如此就只剩支持度門檻值*min_support*有作用，與Apriori演算法只利用支持度門檻值作修剪的概念相同，因此本研究利用將約定值門檻設為0，將演算機制與Apriori演算法做探勘結果之相互比較。

每個實驗的「執行時間」，皆是由開始載入資料庫起始計算，到找出所有大型項目集結束計算。圖6呈現了醫療藥品資料在不同支持度門檻與約定值門檻探勘出的樣式個數，本研究探勘出的樣式為具有兩個項目以上之項目集稱之，由圖可以看出當固定支持度門檻的時候，加入約定值門檻可以有效的刪除不具價值的樣式，並且隨著約定值門檻值越大，刪除的項目集會越多，所探勘出的樣式則越來越少。例如當支持度門檻為0.001時，若約定值門檻值為0，會探勘出1,153個樣式，而當約定值門檻值為0.03時，探勘出的樣式下降到剩261個，約定值門檻值為0.04時，則只探勘出96個樣式。這是因為當約定值門檻為0時，演算法只剩支持度門檻，因此會如同Apriori演算法將所有的樣式都探勘出來，其中包含具有跨支持度特性的無價值樣式。由圖6也可得知當約定值為0時，隨著支持度門檻值下降，探勘出的樣式會急劇成長，但是當加入約定值門檻值時，探勘出的樣式個數，卻不會因為支持度門檻值下降而快速成長，可以有效的探勘出更具價值的項目集。

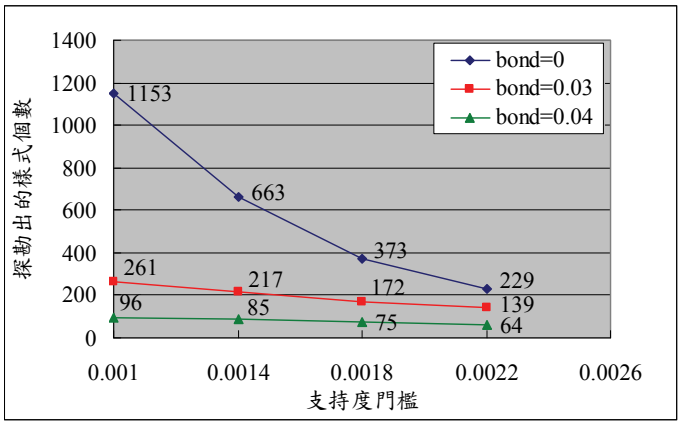

圖6：醫療藥品資料在不同門檻值探勘出的樣式個數

圖7則為醫療藥品資料在不同門檻值之演算法執行時間，可以看出當約定值門檻值為0時，所花費的時間很明顯的比約定值門檻值為0.03和0.04時來得高，且隨著支持度門檻的降低，執行時間會快速增長，尤其當支持度門檻值為0.001時，約定值門檻值為0所執行的時間是約定值門檻值為0.03執行時間的4倍，表示加入約定值門檻值不但可以探勘出較有價值的項目集，而且所花費的時間也比較少，然而配合圖6又可得知，隨著支持度門檻逐漸的提高，探勘出的大型項目集個數差距會逐漸縮小，造成不同約定值門檻的執行時間之差距也會逐漸縮小，尤其當支持度門檻為0.0022時，加入約定值反而沒有得到較佳的執行時間，原因是當將支持度門檻提高，探勘出的大型項目集個數差距會逐漸縮小，因此產生的候選項目集個數之差距也會縮小，而加入約定值門檻的演算法，在篩選過程中必須利用「聯集」計算出約定值之分母，反觀約定值為0時，並沒有執行「聯集」計算，才會造成此一現象。接著比較不同約定值門檻之執行時間，由圖7可知，當約定值門檻由0.03提昇到0.04，演算法所需的執行時間會下降，除了表示當約定值門檻提高，會有較多的候選項目集被刪除外，亦代表了跨支持度特性的修剪策略會刪除較多候選2-項目集，造成執行時間的下降，促使執行績效提昇。

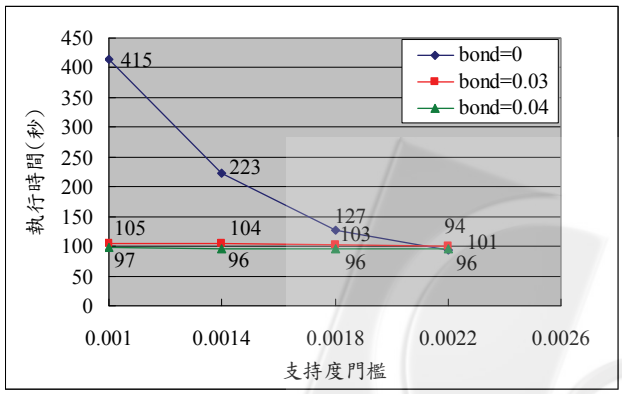

圖7：醫療藥品資料在不同門檻值之演算法執行時間

圖8則呈現了醫療藥品資料在不同約定值門檻下的項目間平均*cosine*相關性，由圖中可以看出當加入了約定值門檻後，所探勘出的樣式，其項目間平均*cosine*相關性比沒加入約定值門檻的樣式之平均*cosine*相關性還來得高，代表著加入約定值門檻的確可以過濾掉項目間具有低相關性的樣式，且隨著約定值門檻的提高，項目間平均*cosine*相關性也會提高。由圖8還可得知探勘出的樣式之項目間相關性平均會大於所設定之約定值門檻，證明利用將約定值門檻加入到探勘大型項目集的演算流程，的確有助探勘出較有價值的項目集。

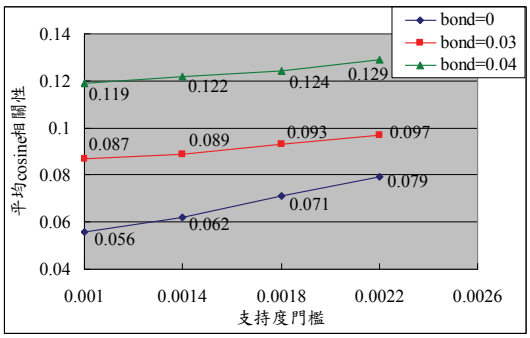

圖8：醫療藥品資料在不同約定值門檻下的項目間平均cosine相關性

接下來將探討利用跨支持度特性修剪候選2-項目集的效度，圖9顯示出將支持度門檻值固定在0.001所執行的結果，其中一條折線代表只使用反單調性修剪項目集的執行時間，另外一條則表示使用反單調與跨支持度兩特性修剪項目集所得到的執行時間，由圖可知，當將演算法加入跨支持度特性修剪策略，所得到的執行成效較佳，且隨著約定值門檻值的提高，獲得的成效越好，這是因為約定值門檻值等於跨支持度修剪策略的門檻值，因此當約定值門檻值提高，跨支持度修剪策略的門檻值也會跟著提高，利用此策略修剪的候選2-項目集也會因而增加所致。圖10表示利用跨支持度特性刪除醫療藥品資料的候選2-項目集個數，由圖中可以看出，隨著約定值門檻的提高，利用跨支持度特性刪除的候選2-項目集也跟著增加，當約定值門檻為0.08時，此修剪策略可以刪除638個候選2-項目集，若將約定值門檻提高到0.2，更可以刪除2737個候選2-項目集，與圖9相配合可得知，利用約定值之跨支持度特性修剪候選2-項目集，可以有效增進整體演算法之效能。

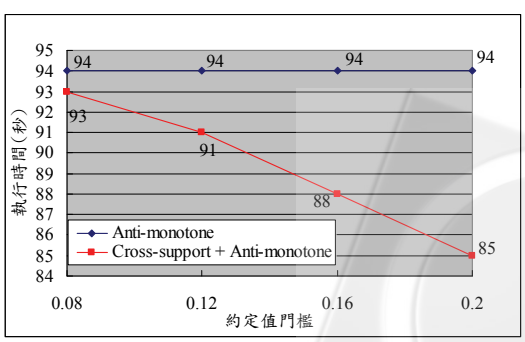

圖9：利用跨支持度特性修剪醫療藥品資料項目集之效用

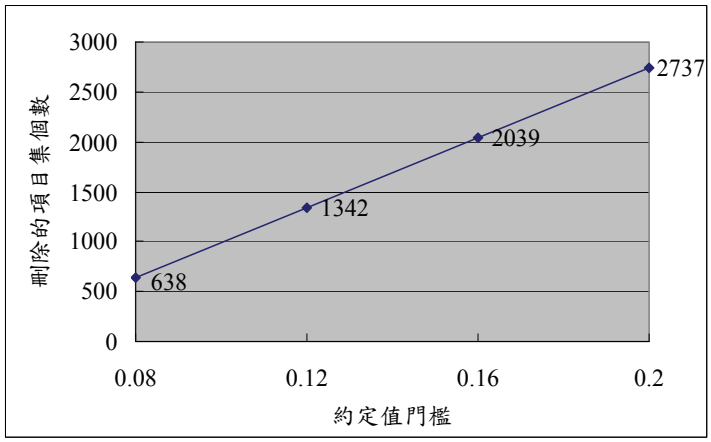

圖10：利用跨支持度特性刪除醫療藥品資料的候選2-項目集個數

表3為利用醫療藥品資料探勘出之部分樣式，設定的約定值門檻為0.06，支持度門檻為0.001，在此門檻之下，醫療藥品資料只探勘出項目數為2的項目集，由表中的資料可知被探勘出來的樣式，其*support(x)/support(y)*的結果皆大於約定值門檻0.06，表示項目間的支持度差距成功的被限制在自訂的範圍內，接著看到cosine相似度的計算結果亦皆大於約定值門檻0.06，由此可推斷，被探勘出的項目集其項目間具有某一程度的相關性，表示利用將約定值概念加入探勘項目集的演算法，可以有效的探勘出較有價值的項目集，例如由表3可以明顯看出樣式{ Nifepin-20 Cap. 保心律120'sBP、Caproine-50 Tab. 佳血平120'sBP }與{ Safe F.C.T. 壽福 50's、Eaci Tab. 一喜 30's }，這兩個樣式裡的產品項目之特性都很相近，第一個樣式中的兩個產品項目之臨床用途皆為「心臟血管疾病」(資料來源：行政院衛生署藥物資訊網)，第二個樣式的第一個產品「壽福」之臨床用途為「綜合維生素補給」，第二個產品之臨床用途則為「維生素C、E補給」(資料來源：行政院衛生署藥物資訊網)，兩產品的臨床用途皆是維生素補給，雖然這兩個樣式的支持度不高，但是樣式裡產品之間的屬性卻非常相近，顯示本研究成功的探勘出項目間具有高度相關性的項目集，如果演算法沒加入約定值的概念，則可能會因為支持度門檻設定較高而遺失這些有價值的項目集，接著可以看到表3中有部份樣式，其產品名稱很相近，例如樣式{ Forstrong Lyo-Inj 復力素 1ml、Forstrong Lyo-Inj 復力素 5ml }，這些樣式可能會被認為是一般的常識，因為樣式中的兩個產品差別只在於容量不同，然而資料庫中有更多其他名稱相同但是容量不同的產品，卻沒被挖掘出來，代表著也許這些被挖掘出來的樣式，具有某種特殊的價值，不過仍須經由專家的判斷才能確切得知其真實價值。

表3：探勘醫療藥品資料獲得的樣式

探勘出的樣式	支持度	約定值	*cosine* 相似度	$\frac{support(x)}{support(y)}$
Gludona Cap. 骨維康 60's、Beautyskin Cream 佳膚 150gm	0.002	0.07	0.146	0.50
Super-Ca Tab. 速補鈣 60's、Eaci Tab. 一喜 30's	0.002	0.07	0.133	0.59
Nifepin-20 Cap. 保心律120'sBP、Caproine-50 Tab. 佳血平120'sBP	0.004	0.07	0.042	0.57
Safe F.C.T. 壽福 50's、Eaci Tab. 一喜 30's	0.002	0.08	0.147	0.89
Gauze 保復膚 10cmx10、An-Fu Gauze 安膚10cmx10	0.014	0.24	0.404	0.55
Repacin Lyo-Inj. 利腫消 1mlx10、Solvent for Repacin 5mlx10Amp	0.003	0.25	0.476	0.28
Forstrong Lyo-Inj 復力素 1ml、Forstrong Lyo-Inj 復力素 5ml	0.011	0.89	0.943	0.89

表4為表3藥品資料樣式轉為關聯規則後計算出的信賴度，由表4可以看出，所有關聯規則的信賴度皆大於約定值門檻0.06，符合Omiecinski (2003) 的證明，表示利用將約定值概念加入探勘項目集的演算法，即使在低支持度門檻時，所探勘出的關聯規則，亦是非常有價值的。

表4：表3藥品資料樣式轉為關聯規則之信賴度

關聯規則	信賴度
Gludona Cap. 骨維康 60's → Beautyskin Cream 佳膚 150gm	0.103
Beautyskin Cream 佳膚 150gm → Gludona Cap. 骨維康 60's	0.207
Super-Ca Tab. 速補鈣 60's → Eaci Tab. 一喜 30's	0.102
Eaci Tab. 一喜 30's → Super-Ca Tab. 速補鈣 60's	0.172
Nifepin-20 Cap. 保心律120'sBP → Caproine-50 Tab. 佳血平120'sBP	0.1
Caproine-50 Tab. 佳血平120'sBP → Nifepin-20 Cap. 保心律120'sBP	0.173
Safe F.C.T. 壽福 50's → Eaci Tab. 一喜 30's	0.138
Eaci Tab. 一喜 30's → Safe F.C.T. 壽福 50's	0.155
Gauze 保復膚 10cmx10 → An-Fu Gauze 安膚10cmx10	0.300
An-Fu Gauze 安膚10cmx10 Gauze → 保復膚 10cmx10	0.546
Repacin Lyo-Inj. 利腫消 1mlx10 → Solvent for Repacin 5mlx10Amp	0.254
Solvent for Repacin 5mlx10Amp → Repacin Lyo-Inj. 利腫消 1mlx10	0.895
Forstrong Lyo-Inj 復力素 1ml → Forstrong Lyo-Inj 復力素 5ml	0.889
Forstrong Lyo-Inj 復力素 5ml → Forstrong Lyo-Inj 復力素 1ml	1.0

接著將此演算架構實驗在比利時超級市場的交易資料上，實驗結果也顯示支持本研究的証明，圖11呈現了超級市場資料在不同門檻值探勘出的樣式個數，可以明顯看出加入約定值門檻可以有效的刪除不具價值的項目集，例如當支持度門檻為0.005時，約定值門檻為0可探勘出359個樣式，但當約定值門檻增加為0.05，則只探勘出27個樣式；圖12則呈現出超級市場資料在不同門檻值之演算法執行時間，圖中顯示加入約定值門檻可以

有效降低演算法執行時間，尤其當約定值門檻為0時，所需的執行時間會增加相當的多，因此我們只針對當支持度為0.004與0.005的情況下作實驗，因為當把支持度門檻降更低時，所花費的時間勢必增加更多。

圖13為超級市場資料在不同約定值門檻下的項目間平均cosine相關性，圖中顯示加入約定值門檻確實可以促使探勘出的項目集，其項目間平均cosine相關性明顯的提高，與醫療藥品資料的實驗結果一致，由此可知加入約定值門檻不但可以降低演算法執行時間，更可增加探勘出的項目集之價值。

接著探討利用跨支持度特性修剪超級市場資料的候選2-項目集之效度，圖14顯示將支持度門檻值固定在0.002所執行的結果，由圖可知當演算法加入跨支持度特性修剪策略，所得到的執行成效較佳，亦與醫療藥品資料得到的結果一致。

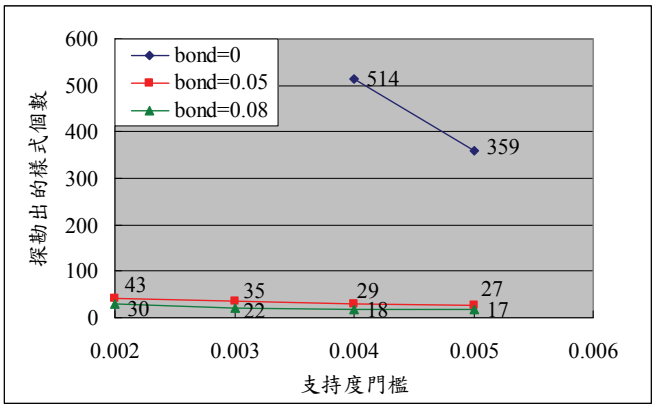

圖11：超級市場資料在不同門檻值探勘出的樣式個數

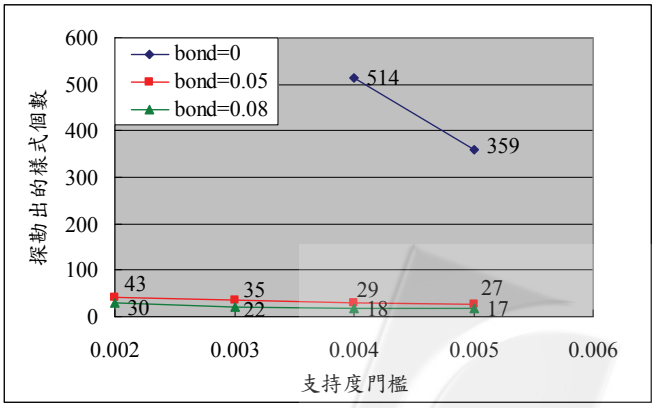

圖12：超級市場資料在不同門檻值之演算法執行時間

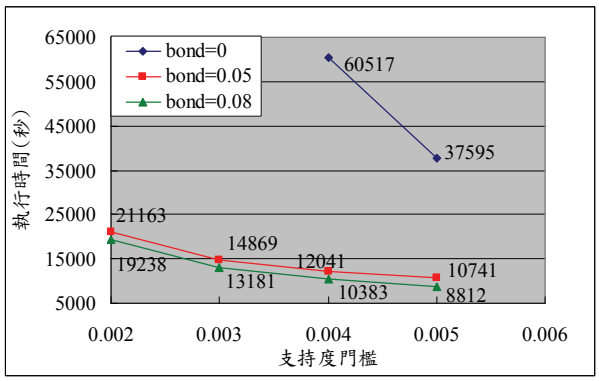

圖13：超級市場資料在不同約定值門檻下的項目間平均cosine相關性

圖15則為利用跨支持度特性刪除超級市場資料的候選2-項目集個數，由圖中可以看出，隨著約定值門檻的提高，利用跨支持度特性刪除的候選2-項目集也會跟著增加，因此圖14利用跨支持度特性之曲線的執行時間才會隨著約定值門檻提高而逐漸降低。

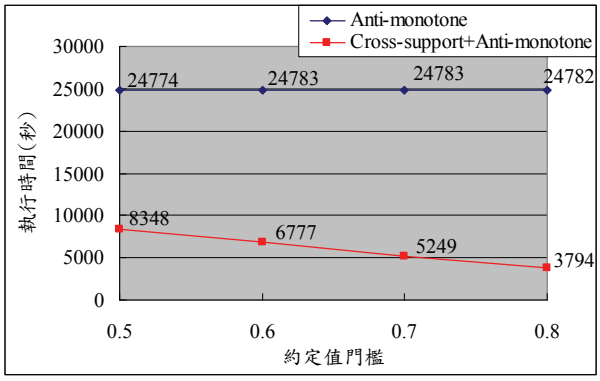

圖14：利用跨支持度特性修剪超級市場資料項目集之效用

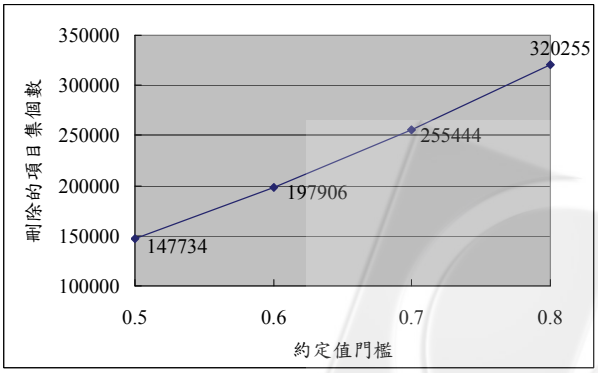

圖15：利用跨支持度特性刪除超級市場資料的候選2-項目集個數

實驗結果說明了當演算法只利用支持度門檻刪減項目集時，隨著支持度門檻越來越低，探勘出的樣式會急劇成長，且會包含許多具有跨支持度特性的微弱相關性項目集，但是當本研究加入約定值門檻時，探勘出的樣式個數，卻不會因為支持度變小而快速成長，且可以更有效的探勘出更具價值的項目集。接著實驗結果又說明了，加入約定值門檻可以有效縮短演算法執行的時間，反觀只利用支持度門檻修剪候選項目集的情況，可以發現隨著支持度門檻降低，則演算法執行的時間會急劇上昇。另外本研究還證明了利用約定值之跨支持度特性修剪候選2-項目集，確實可以有效的降低演算法的執行時間，且當約定值門檻越高，效果會越明顯。由改變項目個數的實驗中發現到，項目個數會明顯的影響到演算法探勘出的樣式個數，而由改變交易筆數的實驗中則發現到，交易筆數會明顯的影響到演算法的執行時間。最後實驗總結，經由各種不同資料的實驗結果可以得知，本研究的演算法可以有效的在低支持度門檻時，探勘出高度相關性的大型項目集。

伍、結論

過去利用支持度修剪策略的演算法，例如Apriori演算法，當面臨歪斜的資料分配時，產生的候選項目集可能有很大部份為跨支持度的微弱相關性樣式，而這些樣式是不具價值的(Xiong et al. 2006)。本研究利用將約定值門檻加入演算法，可以有效的刪除這些跨支持度的微弱相關性項目集，不但降低了演算法的執行時間，更進而在低支持度時，探勘出高度相關性的大型項目集，且所探勘出來的大型項目集，在支持度和信賴度上都具有某一程度的底限。另外利用支持度的修剪策略，再加上了約定值的修剪策略，優點在於可以先利用支持度門檻確保探勘出的項目集佔資料庫總交易筆數的比例，接著再利用約定值門檻確保探勘出的項目集，其項目間的相關性強度。接著我們設計了一個演算架構，在該架構中利用了約定值的跨支持度和反單調性修剪項目集，使演算法可以在低支持度門檻時有效探勘出高度相關性項目集。

未來可以朝以下幾個方向繼續研究，(1)本研究證明出利用約定值可以探勘出高度相關性項目集，而Xiong et al. (2006) 則是證明了all-confidence亦具有此特性，因此未來可以繼續找尋具有此特性的測量方式；(2)本研究所使用的演算法仍須產生候選項目集，未來可以使用其它不需產生候選項目集之演算法與約定值相配合產生高度相關性項目集，加快演算法執行速度，例如FP-Growth(Han et al., 2004)或Opportunistic Projection(Liu et al., 2002)；(3)將本研究的演算法配合資料分群技術找出高度相關性的區域性關聯規則，進而針對目標客戶群作客製化行銷的動作；(4)目前只實驗於二元的資料上，未來可以進一步實驗於連續數值或是量化的資料。

致謝

國科會補助計畫，NSC 95-2221-E-415-011。

參考文獻

1. 丁一賢與陳牧言，2005，資料探勘，台中：滄海書局。
2. 彭文正譯，Michael J. A. Berry and Gordon S. Linoff著，2001，資料採礦 顧客關係管理暨電子行銷之應用，台北：數博網資訊股份有限公司。
3. 曾憲雄、蔡秀滿、蘇東興、曾秋蓉與王慶堯，2004，資料探勘，台北：旗標出版股份有限公司。
4. Aggarwal, C. C., Procopiuc, C., and Yu, P. S. "Finding Localized Associations in Market Basket Data," *IEEE Transaction on Knowledge and Data Engineering* (14:1), 2002, pp. 51–62.
5. Agrawal, R., Imielinski, T., and Swami, A. "Mining Association Rules between Sets of Items in Large Databases," in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data,* Washington, D.C., 1993, pp. 207–216.
6. Agrawal, R., and Srikant, R. "Fast Algorithms for Mining Association Rules in Large Databases," *Proceedings of the 20th International Conference on Very Large Data Bases,* Santiago, Chile, 1994, pp. 487–499.
7. Blischok, T. "Every Transaction Tells a Story," *Chain Store Age Executive with Shopping Center Age* (71:3), 1995, pp. 50–57.
8. Brijs, T., Swinnen, G., Vanhoof, K., and Wets, G. "Using Association Rules for Product Assortment Decisions: a Case Study," *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* San Diego, California, 1999, pp. 254–260.
9. Brijs, T., Swinnen, G., Vanhoof, K., and Wets, G. "Building an Association Rules Framework to Improve Product Assortment Decisions," *Data Mining and Knowledge Discovery* (8:1), 2004, pp. 7–23.
10. Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. "Dynamic Itemset Counting and Implication Rules for Market Basket Data," *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, 1997, pp. 255–264.
11. Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J.D., and Yang, C. "Finding Interesting Associations without Support Pruning," *IEEE Transaction on Knowledge and Data Engineering* (13:1), 2001, pp. 64–78.
12. Han, J., and Kamber, M. *Data Mining: Concepts and Techniques*, London: Morgan Kaufmann, 2001.

13. Han, J., Pei, J., Yin, Y., and Mao, R. "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach," *Data Mining and Knowledge Discovery* (8:1), 2004, pp. 53–87.
14. Hand, D. J., Blunt, G., Kelly, M. G., and Adams, N. M. "Data Mining for Fun and Profit," *Statistical Science* (15:2), 2000, pp. 111–131.
15. Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M. S., and Duri, S. S. "Personalization of Supermarket Product Recommendations," *Data Mining and Knowledge Discovery* (5:1-2), 2001, pp. 11–32.
16. Liu, B., Hsu, W., and Ma, Y. "Mining Association Rules with Multiple Minimum Supports," *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, California, 1999, pp. 15–18.
17. Liu, J., Pan, Y., Wang, K., and Han, J. "Mining Frequent Item Sets by Opportunistic Projection," *Proceedings of 2002 International Conference on Knowledge Discovery in Databases (KDD'02)*, Edmonton, Canada, 2002, pp. 229–238.
18. Ogihara, M., and Li, W. "New Algorithms for Fast Discovery of Association Rules," *Technical Report*, University of Rochester, 1997.
19. Omiecinski, E. R. "Alternative Interest Measures for Mining Associations in Databases," *IEEE Transaction on Knowledge and Data Engineering* (15:1), 2003, pp. 57–69.
20. Park, J. S., Chen, M-S., and Yu, P. S. "An Effective Hash-Based Algorithm for Mining Association Rules," *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, San Jose, California, 1995, pp. 175–186.
21. Savasere, A., Omiecinski, E. R., and Navathe, S. B. "An Efficient Algorithm for Mining Association Rules in Large Databases," *Proceedings of the 21th International Conference on Very Large Data Bases*, Zrith, Switzerland, 1995, pp. 432–444.
22. Song, M., and Rajasekaran, S. "A Transaction Mapping Algorithm for Frequent Itemsets Mining," *IEEE Transaction on Knowledge and Data Engineering* (18:4), 2006, pp. 472–481.
23. Xiong, H., Tan, P.N., and Kumar, V. "Mining Hyperclique Patterns with Confidence Pruning," *Technical Report*, Department of Computer Science, University of Minnesota, 2003a.
24. Xiong, H., Tan, P.N., and Kumar, V. "Mining Strong Affinity Association Patterns in Data Sets with Skewed Support Distribution," *Proceedings of the Third IEEE International Conference on Data Mining*, 2003b, pp. 387–394.
25. Xiong, H., Tan, P.N., and Kumar, V. "Hyperclique Pattern Discovery," *Data Mining and Knowledge Discovery* (13:2), 2006, pp. 219–242.