東海大學資訊工程與科學研究所

碩士論文

指導教授：許玟斌博士

應用分層隨機抽樣和動差保留法採掘重要關聯規則之方法

Mining Interesting Association Rules by Stratified Sampling and Moment-Preserving Thresholding

研 究 生：陸 坤 義

中華民國九十二年六月二十六日

摘要

資料探勘(Data Mining)是現今非常熱門的研究領域，其中，有關如何快速產生關聯規則(Association Rules)的議題，更是被廣泛的討論與研究。產生關聯規則可分為二個階段；第一階段，自交易資料庫中找出高於使用者設定門檻值的高頻項目集(Frequent Itemsets)。第二階段，利用高頻項目集產生信賴度高的關聯規則。由於資料庫的資料量相當龐大，若經由頻繁的存取來產生高頻項目集相當的花費時間，因此，須以有效率且有效果之演算法來進行以節省成本。鑑於以上所述，本研究以分層隨機抽樣(Stratified Sampling)和動差保留法(Moment Preserving Thresholding)為演算基礎，期望能減少自資料庫中採掘高頻項目集和關聯規則所需的時間。在計算項目集支持度時，許多演算法[5][11][12][16]均未考慮購買數量，造成支持度之誤差，進而影響高頻項目集的參考價值。有鑑於此，本研究嘗試在採掘高頻項目集時，將購買數量做為計算支持度之加權值，以避免產生誤差並提昇支援決策的效果。本研究之演算法可分為五個步驟：步驟一、利用模擬程式產生交易資料庫。步驟二、利用動差保留法將每筆交易依利潤分類。步驟三、利用分層隨機抽樣法自前一步驟產生的分類結果中抽取足夠的樣本。步驟四、使用 Apriori 演算法掃描樣本以採掘高頻項目集。步驟五、利用高頻項目集產生關聯規則。經模擬証明，本研究提出之演算法除能夠有效且快速的自資料庫採掘高頻項目集之外，產生的關聯規則對決策者更具有參考價值。

關鍵字：分層隨機抽樣、動差保留法、高頻項目集、關聯規則、Apriori 演算法。

Abstract

Data mining is a very important issue; the association rule mining is the mostly studied one due to the wide applications among the proposed mining methods. The association mining problem should be proceeding by efficient algorithm; it could be divided into two phases. Phase 1: mining frequent itemsets from database. Phase 2: using frequent itemsets to generate the association rules. The proposed algorithm is based on stratified sampling and moment-preserving thresholding approach. Because of the reducing size of dataset, the proposed algorithm is efficient for the association rule mining problem. Moreover, we considered the buying quantities of items in support counting phase to increase the persuasion of frequent itemsets and association rules. The proposed algorithm has five steps. Step 1: generating transaction database by our simulator. Step 2: using moment-preserving thresholding approach to classify transaction by profit. Step 3: using stratified sampling to draw sample database. Step 4: mining frequent itemsets in sample database by Apriori algorithm. Step 5: generating association rules by frequent itemsets. By way of simulation results, the proposed algorithm is efficient in mining frequent itemsets. Besides, the association rules generated by proposed quantitative support counting method are more valuable.

Keywords: stratified sampling, moment-preserving thresholding approach, frequent itemsets, association rules, Apriori algorithm

目錄

圖目錄

表目錄

第 1 章 緒論

資料探勘能夠幫助企業取得有意義的資訊並藉以創造競爭優勢，故引起廣大的重視，也成為一個成長非常快速的研究領域。此外，如何有效的利用資料探勘的技術，分析及找出原先隱藏在龐雜的資料中有用且具有價值的資訊，對於處在瞬息萬變競爭中的企業是非常重要的，以下我們將針對資料探勘的定義和技術做一詳細的探討。

1.1 資料探勘

資料探勘是現今一個非常熱門的研究領域。其主要目的便是期望利用統計模式或人工智慧的方式，從龐大的資料中採掘出有價值的隱藏資訊或知識，並依據不同的問題建立其所屬的模型，以做為決策的依據。資料探勘興起的原因，大致可分為三點：

1. 資料大量產生：由於電腦的使用率日漸普及，因此各個行業都普遍使用電腦來收集資料，然而在資料庫的設計上，收集的欄位可能達上百個，資料筆數更是無法計算，再加不斷新增的資料，所以龐大資料庫的形成是可想而知的。

2. 資料倉儲形成：若將資料庫中的資料，按資料庫設計者設計的型態分別的存放於資料庫中，而逐漸形成一個大型的資料庫，如此一來，便可從這些資料當中找尋出可被利用的資訊，而這個經過分門別類所設計出來的資料庫，就成了資料倉儲(Data Warehouse)。資料倉儲就是一種將資料聚集成資訊來源的場所。

3. 資料探勘演算法：由於資料量的日趨龐大，因此，想在如此龐大的資料庫中進行資料探勘，若無一有效率的演算法，將會相當的花費時間和資源。因此，許多的研究均在探討如何產生有效率的演算法。

進行資料探勘時的步驟，如圖 1-1 所示。首先，由於經過長時間的資料收集，故形成一資料量龐大的資料庫。在建立資料探勘模型之前，必須先瞭解行業特性和對欲解決之問題做一清楚的定義。經過此一步驟後，才能獲得在建立資料探勘模型時所需的資訊，以避免建立模型時產生誤判。經過瞭解本業和需求定義之後，便能夠利用收集到的資訊和演算法建立資料探勘模型；所建立的資料探勘模型是否具有代表性，乃是視前一步驟所收集的資訊是否足夠且正確而定，因此，收集資訊乃是資料探勘中非常重要的一環。在建立資料探勘模型之後，便可利用所建立的資料探勘模型來進行資料探勘，並產生探勘結果。最後，由決策者決定探勘結果是否對於決策有幫助，若決策者認為此一結果對於決策有幫助，則保留探勘結果，反之，則再次進行需求定義，並重新建立探勘模型，直到獲得所需的探勘結果。

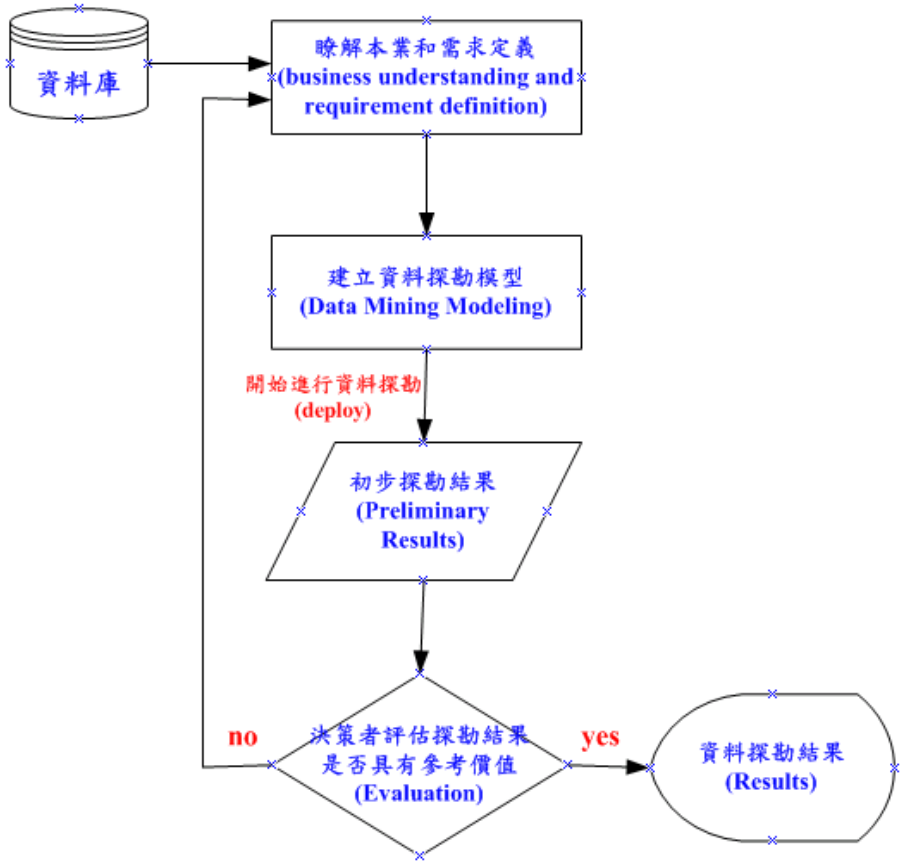

圖 1-1. 資料探勘的流程

1.2 資料探勘相關技術

在這個資訊爆炸的時代，企業必須面對的是競爭對手的挑戰和龐大資料量所帶來的衝擊。雖然資料量的不斷增加，使得企業必須花費更多的資源去維持這些資料，但由於對於企業具有價值之資訊均隱藏其中，因此，若企業能夠運用有效率的方法，從這些龐大的資料中獲得對企業競爭力有助益的資訊，如此一來，花費大量的資源來維持這些資料，對企業而言，便是重要且必須的工作。

資料探勘或知識探勘(Knowledge Mining)是藉由歸類、分析和整理的方式，並運用一些有效的演算法，從龐大的資料中獲得有意義資訊的一種分析方法。在過去，要在資料庫中尋找資料的特性或分析結果，乃是藉由資料庫軟體的協助來達成，如使用 SQL 語法查詢或藉由運用線上分析處理(On-Line Analytical Processing, OLAP)與資料倉儲(Data Warehousing)來找出對於資料的分析結果，而資料探勘對於資料分析與模型建立更是一種有效的工具；以技術面來看，資料探勘泛指各式各樣從大量資料中歸納出資訊與知識的方法。

資料探勘的主要核心技術經歷了數十年的發展；這些成熟的技術，加上高性能的資料庫以及廣泛的資料收集，讓資料探勘技術成為目前非常受到重視的研究領域。資料探勘的技術可大致分為購物籃分析(Market Basket Analysis)、分類技術(Classification)和群集技術(Clustering)等。以下針對不同的探勘技術做說明：

1.2.1 購物籃分析

購物籃(Market Basket)是指在一獨立發生的交易中，顧客所購買的商品集合。購物籃分析主要被用來尋找資料庫中項目或屬性之間共同發生的關係。當決策者取得這些資訊後，就可以考慮針對不同項目擬定不同的銷售策略，以增加更多的利潤。進行購物籃分析的主要目的是期望能夠自交易資料庫中，採掘出最常被一起購買的商品集合，稱為高頻項目集。再利用由高頻項目集產生之關聯規則

分析消費者的購物習慣。關聯規則是指在龐大的資料庫中某些項目間彼此的關聯性，其最早的研究乃是針對交易資料庫，分析市場購物籃資料的交易紀錄，並找出相關商品集彼此的關聯性，幫助企業決定賣場或目錄中商品的擺設方式和吸引顧客的注意，以獲得更高的利潤。

在產生關聯規則之前，必須先從資料庫中找出經常被一起購買項目集(Itemset)，稱為高頻項目集(Frequent Itemset or Large Itemset)。項目(Item)通常可視為一項商品，因此，若有 n 項商品，便有 n 個項目，可表示為$\{I_1, I_2 \ldots I_n\}$；而“i-項目集”(i-Itemset)通常是指包含有 i 個項目的集合，換言之，亦指含有 i 個商品的商品集；而包含項目集的交易佔整體資料庫之百分比稱為該項目集的支持度(Support)，如(式 1-1)，通常表示為：支持度(項目集)。

$$\text{支持度(項目集)} = \frac{\text{含項目集的交易筆數}}{\text{資料庫所含的交易筆數}} \cdots\cdots\cdots\cdots \text{(式 1-1)}$$

若一項目集的支持度高於一使用者定義的最小支持度門檻值(Minimum Support Threshold)，則此一項目集為高頻項目集。此外，有可能成為高頻項目集，但仍需加以驗証其支持度是否足夠的項目集，稱為候選項目集(Candidate Itemset)。舉例來說，若表 1-1 為一含有五筆交易記錄之交易資料庫，且此一交易資料庫含有五個欄位：交易索引值、顧客索引值、交易內容、商品價格和交易日期。假定最小支持度門檻值為百分之五十；由表 1-1 可以發現項目集{牛奶，麵包}在五筆交易中其中三筆出現，因此項目集{牛奶，麵包}的支持度為百分之六十，高於門檻值，因此為高頻項目集。

由於資料庫所含的資料量非常龐大，且項目的數量非常多，因此許多演算法均期望利用有效率的方式來採掘高頻項目集，以減少企業在進行採掘高頻項目集時所需的時間和成本。藉由採掘演算法自資料庫採掘出所有的高頻項目集之後，便可以利用這些高頻項目集來產生關聯規則。舉例來說，X 和 Y 為二個獨立的高頻項目集，若我們想瞭解消費者在購買 X 項目集後會去購買 Y 項目集的機率，

可以透過關聯規則 $X \rightarrow Y$ 來表示。關聯規則通常可利用一信賴度(Confidence)來衡量此一規則的可靠程度。以規則 $X \rightarrow Y$ 為例，其信賴度為：

$$\text{信賴度}(X \rightarrow Y) = \frac{\text{支持度}(X \cup Y)}{\text{支持度}(X)} \quad \cdots\cdots\cdots\cdots \text{(式 1-2)}$$

如同採掘高頻項目集，在產生關聯規則時，同樣利用一使用者定義的最小信賴度門檻值(Minimum Confidence Threshold)來決定關聯規則的保留與否；若一關聯規則之信賴度高於門檻值，則表示該規則具有足夠的信賴度並加以保留，反之，則不保留該規則。因此，在表 1-1 中，假定信賴度門檻值為百分之七十，由於項目集{牛奶}之支持度為百分之八十，且項目集{牛奶，麵包}之支持度為百分之六十，因此，{牛奶}→{麵包}此一關聯規則之信賴度為：$0.6/0.8 = 0.75 = 75\%$；由於其信賴度高於門檻值，代表此一關聯規則具有足夠的信賴度，可保留之。

在購物籃分析中有二項重要的特性值得注意。在購物籃分析中最具有代表性的 Apriori 演算法便是利用其特性來進行削減候選項目集的步驟。

1. 若某一項目集為非高頻項目集(Infrequent Itemset)，則包含該項目集的所有母項目集(Super Set)均為非高頻項目集。

2. 若某一項目集為高頻項目集，則該項目集的所有子項目集(Sub Set)均為高頻項目集。

表 1-1. 含有五筆交易記錄之交易資料庫

交易索引值	顧客索引值	交易內容	商品價格	交易日期
101	201	{牛奶，麵包，果醬}	{100，30，50}	2003/3/30
102	202	{牛奶，飲料，泡麵}	{100，20，25}	2003/4/1
103	202	{牛奶，麵包，泡麵}	{100，30，25}	2003/4/3
104	203	{泡麵，雜誌}	{25，120}	2003/4/5
105	204	{牛奶，麵包}	{100，30}	2003/4/10

由以上敘述可以發現，購物籃分析可以分為二個階段：第一階段、自資料庫中採掘高頻項目集。第二階段、利用高頻項目集產生關聯規則。由於只要採掘出高頻項目集後，關聯規則便能夠快速且直覺的產生；因此，第一階段乃是購物籃分析是否有效率的關鍵。鑑於以上所述，本研究將提出一有效率的應用分層隨機抽樣和動差保留法之演算法，經模擬實驗證明，本研究提出的演算法能夠快速且準確的自大量的資料庫採掘出高頻項目。

1.2.2 分類技術

分類在統計學領域是一個行之有年的技術[10][11][13]。分類的目的在於利用訓練資料(Tranning Data)中的特徵或屬性來建立分類器(Classifier)。分類採掘通常必須將每一個類別的特徵清楚定義，再使用與訓練資料特徵和屬性相同，但內容不同且已給定分類的測試資料(Test Data)來判定建立的分類器是否有足夠的準確度。若分類器具有足夠的準確度，則可應用於未來新資料集的分類。

分類技術之相關研究已提出許多不同的分類模型。如類神經網路(Neural Network)、基因演算法(Genetic Algorithm)、貝氏法則(Bayesian Theory)、決策表(Decision Table)、統計方法(Statistical Methods)和樹狀結構模型(Tree Structures Model)等。其中樹狀結構又稱為分類樹(Classification Tree)或決策樹(Decision Tree)，為一種最常被應用的分類演算法。決策樹具有下列優點，因此，常被應用做為資料分類演算法：

- 能夠將分類結果以直覺式的樹狀結構表示，故較易於瞭解。
- 建構決策樹時不需要輸入任何參數。
- 能夠非常快速且準確的將大量資料做分類。

決策樹在進行分類時可能由於資料過於龐大，而建造出記憶體無法容納的決策樹。若遇到這種情形時，可針對決策樹進行削減(Prune)。決策樹從根節點到葉

節點形成一規則，若樹長得太龐大且分支太多，便會產生過多的規則而造成判斷上的困擾。因此，在建構決策樹時應適度的進行削減，將深度過長的分析剪短，或不需分得太徹底。

圖 1-2 為賣場應用決策樹分類法對其顧客購買力分類的一個簡單範例。由圖可知年齡小於 30 和年齡大於 30，沒有房子，薪資小於 5 萬元的客戶其購買力較弱，因此，賣場中較高價的商品對於這個族群的客戶較無吸引力，相反的，年齡大於 30，有房子和年齡大於 30，沒有房子但薪資高於 5 萬元的客戶其購買力較強。若賣場獲得以上的這些資訊，則可藉用分析賣場中那種類型的客戶較多來決定賣場中的商品進貨類型和商品陳列方式，或針對不同購買力的客戶進行不同的促銷方式以增加利潤。

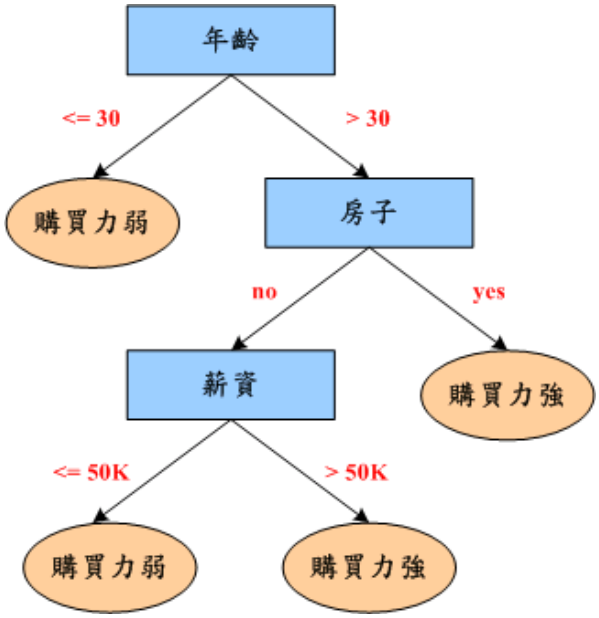

圖 1-2. 賣場客戶購買力之決策樹範例

1.2.3 群集技術

群集技術乃是指使用適合的群集化演算法，有效的將所有資料依照其同質程度區隔出同質性較高的群集或是子群集，並使得各個群集的特徵(Centroid of Cluster)能夠突顯。將資料群集化可應用許多技術來達成，如統計方法(Statistical

Methods)、機器學習(Machine Learning)和生物學相關技術(Biology)。和分類技術不同的是，群集技術並沒有依靠事先明確定義的類別來進行分類；此外，群集的意義須由事後的闡釋才能得知。圖 1-3 中有三個群集Ⅰ，Ⅱ和Ⅲ。各個群集中的圖形均相同代表群集內特性相同，而由於群集間距離較遠代表群集之間特性有所不同。

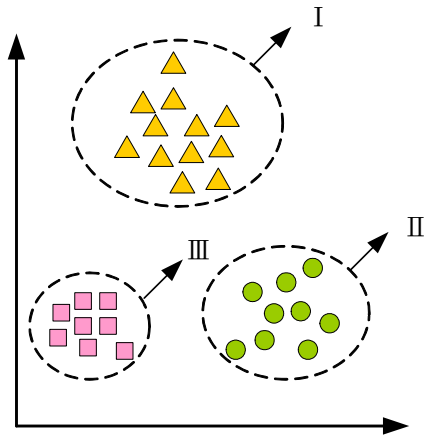

圖 1-3. 群集技術

1.3 研究動機與目的

本研究是針對資料探勘中的購物籃分析為研究方向。購物籃分析中最重要且對於決策支援具有參考價值的資訊，稱為“關聯規則”。以交易資料庫為例，資料庫中除了記錄著所有的交易之外，其中還隱含了所有顧客的消費習慣。因此，若能從其中採掘最常被購買的商品集合或分析出顧客的購物習慣將對於企業行銷有很大的幫助。在資料探勘中，最早被提出的觀念便是找尋資料之間的關聯規則，所謂採掘關聯規則，乃是期望從資料庫中發掘出資料彼此間的關係，亦可以一事件為中心，找出和其有關的資料模式。舉例來說，在經過分析之後，零售業者可能發現，有 90%的顧客在購買尿布和奶瓶之後，會去購買奶粉；股票分析師可能發現，在過去一個月中，有 90%的時間，當 A 股票和 B 股票上漲時，C 股票會上漲；地震學家可能發現，在過去的五年內，有 90%的時間，當 A 地和 B

地發生地震後，C 地會發生地震。由以上所述，可以發現關聯規則能夠描述資料或事件之間的關聯性，因此具有相當重要的參考價值。雖然關聯規則相當的重要，但由於在這個資訊爆炸的時代裡，資料庫所含的資料量相當龐大，若藉由不斷的存取資料庫，從資料庫中採掘出關聯規則將需花費相當多的時間，因此，必須以有效率的方式來進行以節省成本和時間。在產生關聯規則之前，必須先從資料庫中採掘出現頻率最高的項目集。採掘出所有高頻項目集後，便可以利用高頻項目集來產生關聯規則。由於進行購物籃分析是否具有效率，其關鍵便是在採掘高頻項目集的步驟，因此，若期望能夠快速的產生關聯規則，必須減少採掘高頻項目集所需的時間。鑑於以上所述，本研究將嘗試以分層隨機抽樣和動差保留法為演算基礎，期望本研究提出的演算法能夠自資料量龐大的資料庫中快速且有效的進行採掘高頻項目集的動作。

進行購物籃分析需花費大量的時間和成本，其根本的原因，便是由於資料庫的資料量相當龐大，因此，若能夠以特定的方式將資料量減少並保留資料的特徵，便能夠快速的採掘高頻項目集。Hannu Toivonen[8]在 1996 年提出利用抽樣方式來有效的減少資料庫的資料量，以快速的採掘高頻項目集。Hannu Toivonen 提出之抽樣演算法乃是利用單純隨機抽樣演算法來做為抽樣方法，雖然單純隨機抽樣演算法擁有簡單且快速的特性，但其缺點為其抽樣時容易產生抽樣誤差，若偏差情形嚴重將造成樣本資料集不具代表性，故自樣本中採掘的高頻項目集將和資料庫中實際之高頻項目集差異相當大；如此一來，藉由高頻項目集所產生之關聯規則，對於決策支援的參考價值便會相對較低。鑑於以上所述，本研究將以分層隨機抽樣演算法取代單純隨機抽樣演算法，並將獨立交易中的利潤，藉由動差保留法將整體資料庫分類，期望能夠控制抽樣方法所產生之誤差，並以有效率的方法自龐大的交易資料中採掘出高頻項目集。此外，傳統購物籃分析在進行篩選高頻項目集時，並未將交易中的商品購買數量納入考慮，僅以商品集合出現之交易數做為評估項目集是否為高頻項目集之依據，因此，若某一商品集合被購買的

次數少，但每次均為大量購買，依照傳統購物籃分析的計算方法，此種商品集合將無法成為高頻項目集，但其對於賣方來說，卻是相當重要且能夠創造大量利潤的商品集合。因此，本研究期望在進行篩選高頻項目集時，能夠將商品集合的購買數量納入考慮，以避免上述情形。

1.4 論文架構

在本研究中，我們將問題分成以下二個部份，以達到能夠快速自資料庫中採掘高頻項目集和賦與探勘結果更高的決策價值的目的。

1. 由於資料庫的資料量相當龐大，為節省成本與時間，必須能夠快速且有效的採掘高頻項目集，並利用產生之關聯規則做為決策參考。本研究提出以分層隨機抽樣和動差保留法為基礎之演算法，期望能夠有效且正確的減少資料量，以增加採掘高頻項目集和關聯規則時的效率。分層隨機抽樣在進行之前，必須先將資料分為若干層以進行抽樣；此外，由於本研究的目的為期望能夠增加購物籃分析的效率，因此，若分類方法無法快速的進行，將會浪費相當多的時間而使本研究失去意義。動差保留法為一能夠快速且有效的資料分類法，因此，本研究選擇動差保留法為資料分類演算法，以利分層隨機抽樣法之進行。在第三章中將詳細分層隨機抽樣和動差保留法之演算方式。而第四章將利用模擬實驗分析本研究所提出之演算法的效能，並將就執行效率上和 Apriori 演算法做比較。

2. 傳統購物籃分析中，對於高頻項目集之篩選是以項目集出現之交易數做為依據。其對於項目集之支持度計算並未考慮該項目集之被購買數量，故進行項目集支持度計算時將造成誤差，採掘出之高頻項目集亦較不具有代表性；因此，本研究提出一數量化支持度計算法，期望在計算項目集支持度時，能夠將項目集之購買數量做為加權值，以增加高頻項目集之代表性。利用數量化支持度計算法來計算項目集支持度，除能夠增加高頻項目集之代表性外，藉

由高頻項目集產生之關聯規則對於決策支援之效果將更加顯著。在第三章中將對於數量化支持度計算法做一詳細說明。

本論文的架構如下，第二章為文獻探討，第三章為理論架構，第四章為模擬實驗，將對本研究所提出之演算法做分析與討論，第五章為結論與未來研究。

第 2 章 文獻探討

本章將介紹若干已被提出且應用於自龐大資料庫中採掘高頻項目集的演算法，如 Apriori 演算法[14]、DHP 演算法[12]、Pincer-Search 演算法[4]和 Sampling 演算法[8]等。我們將介紹上述演算法之概略做法，並簡單的探討其優缺點。

2.1 Apriori 演算法

Apriori 演算法(Apriori Algorithm)[14]是購物籃分析中最具代表性的演算法。Apriori 演算法是一種典型的由下而上(Bottom-Up)的演算法，換言之，Apriori 演算法乃是自長度短的項目集開始，並逐漸朝長度較長的項目集進行分析。Apriori 演算法在處理每一層不同長度的項目集時均可分為二大階段：第一階段、產生候選項目集。第二階段、計算項目集之支持度。

Apriori 演算法的第一階段—產生候選項目集又可分為結合(Join Step)和削減(Prune Step)二個步驟。除了第一次的候選項目集(以 C_1 表示)是直接利用掃描資料庫一次來獲得之外，其餘的候選項目集(C_k，$k > 1$)均可利用結合和削減來產生。結合和削減的詳細說明如下：

- 結合步驟：在處理 k-項目集時，Apriori 演算法會利用擁有(k-1)個相同項目的高頻項目集來產生長度為(k+1)的候選項目集。舉例來說，{A,B,C}和{A,B,D}為二個 3-高頻項目集。由於此二項目集擁有(3-1) = 2 個相同項目，因此，經過結合之後，會產生{A,B,C,D}此一候選“4-項目集”。利用結合的方式，Apriori 演算法能夠快速的產生候選項目集。

- 削減步驟：Apriori 演算法利用在 1.2.1 節中介紹的項目集特性來進行削減候選項目集的動作。由於若一項目集為非高頻項目集，其所有的母項目集均為非高頻項目集。因此，Apriori 演算法在進行削減候選“k-項目集”時，會檢定所有經由結合步驟產生的候選項目集之子項目集，若一

候選項目集之任一子項目集並非為高頻項目集，則該候選項目集會被刪除。舉例來說，{A,B,C}、{A,B,D}、{A,C,D}、{A,C,E}和{B,C,D}為二個高頻“3-項目集”，經過結合之後會產生{A,B,C,D}和{A,C,D,E}二個候選“4-項目集”。但由於其中{A,C,D,E}的其中一個子項目集{C,D,E}並非高頻項目集，因此，{A,C,D,E}將會被刪除。

Apriori 演算法的第二階段—計算項目集之支持度。此階段需經過掃描資料庫來計算候選項目集之支持度，並視其是否高於最小支持度門檻值來決定候選項目集是否為高頻項目集。第一次利用掃描資料庫一次所獲得之候選項目集均會擁有一支持度，將該項目集支持度和最小支持度門檻值比較，若高於最小支持度門檻值則可成為含有一個項目的高頻項目集稱為 L_1，接著將 L_1 藉由上述的結合和削減步驟可產生長度為 2，也就是含有二個項目的候選項目集 C_2，接著再藉由掃描資料庫來刪除 C_2 中支持度不足的項目集，便可產生 L_2，以此類推直到無法再找出高頻項目集為止。由以上所述，我們可以歸納 Apriori 演算法之演算方式，除 L_1 之外，由 C_k 可獲得 L_k，而 C_k 是利用 L_{k-1}，藉由結合和削減等二個步驟來產生，C_k 和 L_k 均表示項目集中含有 k 個項目，此外，每次計算 C_k 的支持度均需掃描資料庫 一次。圖 2-1 為 Apriori 之演算法。

```
1)  L1 = {large 1-itemsets};
2)  for ( k = 2; Lk−1 ≠ ∅; k++ ) do begin
3)     Ck = apriori-gen(Lk−1); // New candidates
4)     forall transactions t ∈ D do begin
5)        Ct = subset(Ck, t); // Candidates contained in t
6)        forall candidates c ∈ Ct do
7)           c.count++;
8)     end
9)     Lk = {c ∈ Ck | c.count ≥ minsup}
10) end
11) Answer = ⋃k Lk;
```

圖 2-1. Apriori 演算法

根據以上所述，我們以一簡單的例子再加以詳細說明。若資料庫中含有四筆交易資料，此外，假定使用者定義之最小支持度門檻值為 50%，則 Apriori 演算法之過程，如圖 2-2[12]所示。由於資料庫中含有{A}、{B}、{C}、{D}和{E}等五個候選“1-項目集”；經過第一次掃描資料庫後，可以獲得此五個項目各自的支持度，和支持度門檻值比較後可獲得{A}、{B}、{C}、{E}等四個高頻“1-項目集”。接著利用結合和削減來產生候選“2-項目集”，也就是 C_2，再經過掃描資料庫便獲得 C_2 中所有項目集之支持度，再和支持度門檻值比較，便產生{A,C}、{B,C}、{B,E}、{C,E}等四個高頻“2-項目集”，反覆進行上述步驟便可產生所有的高頻項目集。

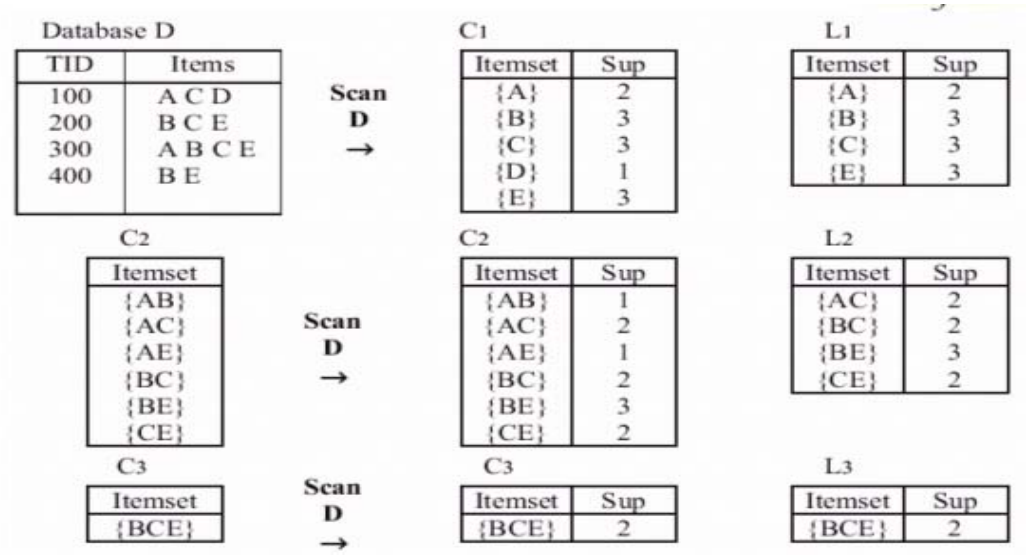

圖 2-2. Apriori 演算法產生候選項目集和高頻項目集之範例[12]

由以上所述，可以發現 Apriori 演算法的優點為其步驟簡單且系統容易實作，但其缺點為產生相當大量的候選項目集和需要相當頻繁的掃描資料庫，如此將造成在採掘高頻項目集時相當的浪費時間，因而使得採掘效率不彰。

2.2 DHP 演算法

DHP(Direct Hashing and Pruning)演算法[12]乃是以 Apriori 演算法為基礎，再應用雜湊的技術來增加採掘高頻項目集的效率。由於 Apriori 演算法執行效率緩

慢的關鍵，在於會產生過多的候選項目集，尤其以在產生候選“2-項目集”時的情況最嚴重。因此，J. S. Park[12]等人於 1995 年提出 DHP 演算法，其主要目的在減少候選“2-項目集”的數量。DHP 演算法在系統第一次掃描資料庫時，除了計算“1-項目集”的出現次數之外，順便利用雜湊函數(Hash Function)和雜湊表(Hash Table)的資料結構來估計候選“2-項目集”的出現次數。利用這些雜湊技術的目的在於減化計算，由於雜湊表會發生碰撞(Collision)的情形，因此所計算的出現次數可能會高估，若被高估的出現次數亦低於最小支持度門檻值，則該項目集真正的出現次數必同樣地會低於門檻值，因此在利用高頻“1-項目集”結合產生候選“2-項目集”時，便可忽略而不產生。

為了產生候選“k-項目集”，DHP 演算法利用雜湊函數建立雜湊表 H_k，雜湊表中記錄了雜湊位址值(Hash Address)和位址計數值(Bucket Count)等二種資料。若一項目集經由雜湊函數計算出屬於某一雜湊位址，則將其所在位址的計數值加 1。若我們從 H_k 中獲知其位址計數值小於支持度門檻值時，便可以知道對應的位址中所含的項目集不可能為候選項目集，因此在產生候選項目集時便可不予考慮，藉此達到減少候選項目集數量的目的。

圖 2-3 為 DHP 演算法應用雜湊技術來產生候選項目集 C_2 之範例。資料庫 D 中含有四筆交易，其 TID 分別為 100、200、300 和 400；此外，將最小支持度門檻值設定 50%；因此，在此例中，項目集在四筆交易中至少出現在二筆交易中，才會被視為高頻項目集。首先，DHP 演算法會先經過掃描資料庫一次之後產生高頻“1-項目集”，L_1 = {A}、{B}、{C}、{E}；接著，利用雜湊函數(Hash Function)建立一可計算資料庫中所有“2-項目集”出現頻率的雜湊表 H_2；在圖 2-3 中所使用的雜湊函數如(式 2-1)所示：

$$h(\{X,Y\}) = ((\text{項目X的次序}) * 10 + (\text{項目Y的次序})) \bmod 7 \quad \cdots \text{(式 2-1)}$$

在(式 2-1)中，項目的次序(Order)乃是視該項目的編號而定。舉例來說，圖

2-3 中所使用的資料庫 D 中共有 5 種項目，分別為{A}、{B}、{C}、{D}和{E}；若我們分別將它們的編號訂為 1 到 5，則可利用(式 2-1)計算出所有“2-項目集”對應的雜湊位址。舉例而言，項目集{A,B}所對應的雜湊位址為 h({A,B}) = ((1*10)+(2)) mod 7 = 5；利用上述的方法，DHP 演算法便可計算出資料庫中所有“2-項目集”所對應的雜湊位址，再利用位址計數值來估計項目集的出現次數。建立雜湊表 H_2 之後，便可利用 L_1 和雜湊表中所估計的項目集出現次數來產生候選“2-項目集” C_2。

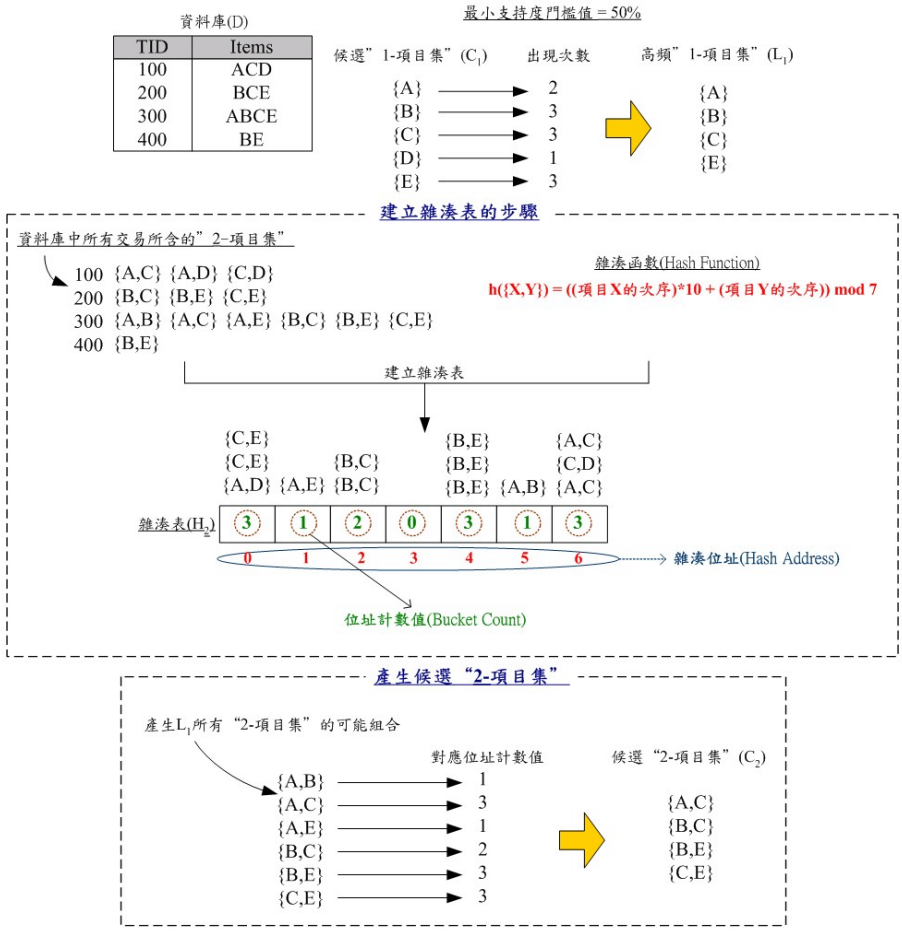

圖 2-3. DHP 演算法利用雜湊技術產生候選項目集 C_2

除使用雜湊技術外，DHP 演算法亦運用有效率的刪減技術來減少交易資料庫的大小，以大幅地減少搜尋交易資料庫的時間並提昇資料探勘的效率。如圖 2-4 所示，交易 100 只包含一個候選“2-項目集”{A,C}，因{A,D}與{C,D}只出

現一次而無法成為候選項目集，且由於“1-項目集”{A}和{C}在交易 100 中出現的次數均為 1 次，其小於最小支持度門檻值，所以將此交易刪除，不再列入下一次被搜尋的交易資料庫。同理，交易 300 包含了四個候選“2-項目集”{A,C}、{B,C}、{B,E}和{C,E}，該交易的其中二個子項目集{A,B}與{A,E}由於只出現一次，故無法成為候選“2-項目集”而將其刪除；再計算{A}、{B}、{C}、{E}在交易 300 的出數次數，由於{A}的次數只有 1 而必須刪除，因此，僅保留 B、C 和 E 而刪除項目 A。經過上述步驟後，下一次被搜尋的交易資料庫，僅保留了{ <200 B,C,E> , <300 B,C,E> }。

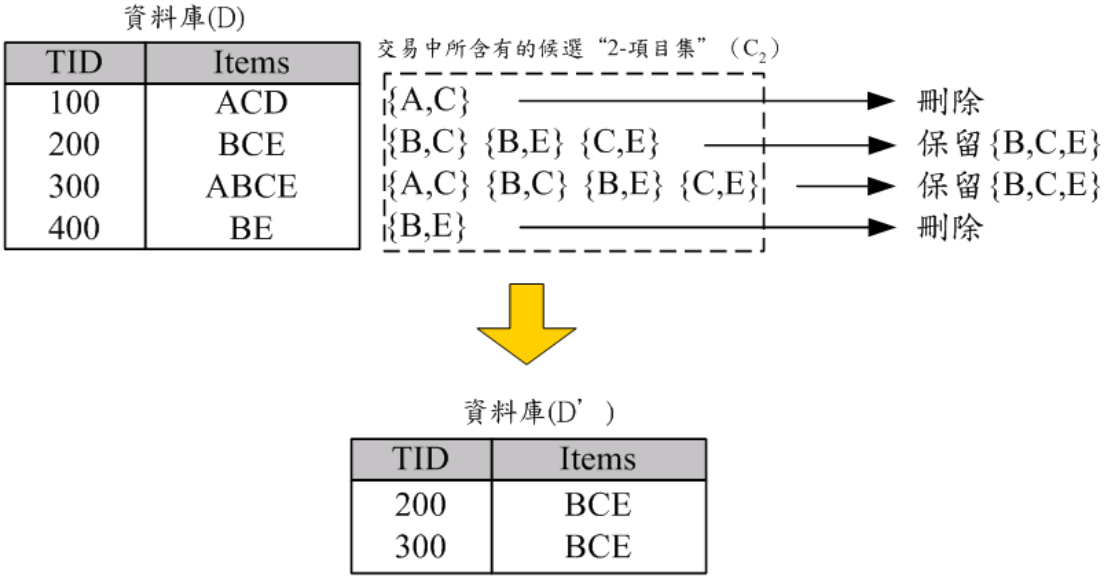

圖 2-4. DHP 演算法刪減交易資料庫之範例

由以上所述可以發現， DHP 演算法雖然需花費額外的時間建立雜湊表，但其對於其後之候選項目集的產生卻有相當大的改善，故 DHP 演算法相較於 Apriori 演算法能夠較快速的採掘出高頻項目集而達到提昇效率的目的。DHP 演算法主要的缺點便是仍然會產生大量的候選項目集且由於利用雜湊技術僅是對於出現次數的估計，因此其掃描資料庫的次數仍和 Apriori 演算法一樣多。DHP 的演算法，如圖 2-5 所示。

```
//Step 1 of the DHP algorithm
Function build_hash_table()
1.  Initialize all hash buckets in the hash table H2 to zero;
2.  for all transactions  t ∈ database  D do
3.      Insert and count the supports of all 1-itemsets in a hash tree ;
4.      for all 2-item subsets x of t do
5.          H2[ h2(x) ]++; // h2 is a hash function
6.  L1 = {c | c. count  ≥  minimum support, c is in the hash tree} ;
End_of_build_hash_table

//Step 2 of the DHP algorithm
Function gen_candidate(L1, H2, C2)
1.  C2 = L1  ×  L1 = { X ∪ Y | X, Y ∈ L1};
2.  for all 2-itemsets c ∈ C2 do
3.      if H2[ h2(c) ] < minimum support then remove c from C2 ;
4.  Scan D to obtain the supports of all 2-itemsets in C2 ;
5.  L2 = {c | c.count  ≥  minimum support, c is in C2} ;
End_of_gen_candidate
```

圖 2-5. DHP 演算法

2.3 Pincer-Search 演算法

許多演算法自所含項目之平均長度較短的項目集中採掘高頻項目集時效能較佳，若項目集中所含項目之平均長度較長，其採掘效能較差。由於項目集的平均長度乃是隨項目的數量成指數成長，若項目的數量多，其組合而成的項目集數量將相當龐大且項目集之平均長度亦將相當可觀，故在採掘高頻項目集時將耗費許多時間；此外，若項目集的平均長度較長，則需掃描更多次資料庫，造成增加採掘高頻項目集所需的 I/O 時間。因此，若採用傳統 Apriori 演算法由下而上的階層式(Level-Wise)演算法自所含項目之平均長度較長的項目集中採掘高頻項目集，其花費的時間將隨項目集長度增加而成長。

有鑑於以上所述，D. I. Lin 等人於 2002 年提出 Pincer-Search 演算法[4]，期望能夠自所含項目之平均長度較長的項目集中快速的採掘出高頻項目集。

Pincer-Search 演算法藉由快速的採掘出資料庫中所有的最大高頻項目集(Maximum Frequent Itemset)以解決上述問題。最大高頻項目集是指所有高頻項目集中，其所有子項目集皆為高頻項目集且不為其他高頻項目集之子項目集的特定高頻項目集。在此特性下，若能夠採掘自資料庫中採掘出所有的最大高頻目集，便如同採掘出資料庫中所有的高頻項目集。

Pincer-Search 演算法除採用傳統由下而上的階層式掃描方式之外，還加入了由上而下的掃描機制。一般而言，單純採用由下而上之掃描方式的演算法，對於所含項目集之平均長度較短的交易擁有較佳的採掘效率，反之，由上而下的掃描方式對於所含項目平均長度較長的項目集則有較佳的採掘效率。Pincer-Search 演算法由於同時採用二種掃描機制，因此其效率較採用單一掃描方式之演算法來的好。圖 2-6 為 Pincer-Search 演算法之流程示意圖。Pincer-Search 演算法乃交叉運用在 1.2.1 節中介紹之項目集的特性以進行掃描。藉由項目集的特性以利用由上而下和由下而上二個方向朝中心點搜尋，以快速的採掘最大高頻項目集。由於由上而下的掃描方式對於所含項目平均長度較長的項目集有較佳的採掘效率，因此，能夠快速的採掘出所有的最大候選項目集，再利用由下而上的掃描方式所獲得的資訊，便可快速的篩選出所有的最大高頻項目集。由以上所述可以發現，Pincer-Search 演算法利用二種掃描機制來減少掃描次數，故能夠有效的提昇採掘效率。雖然 Pincer-Search 演算法能夠有效的提昇採掘效率，但其仍有缺點；Pincer-Search 演算法的主要缺點在於其仍然需多次的掃描交易資料庫，此外，當資料庫中最大高頻項目集的平均長度較長時，應用 Pincer-Search 演算法來進行資料探勘可獲得很好的效率，但當資料庫中最大高頻項目集的平均長度較短時，則 Pincer-Search 演算法無法獲得良好的採掘效率。

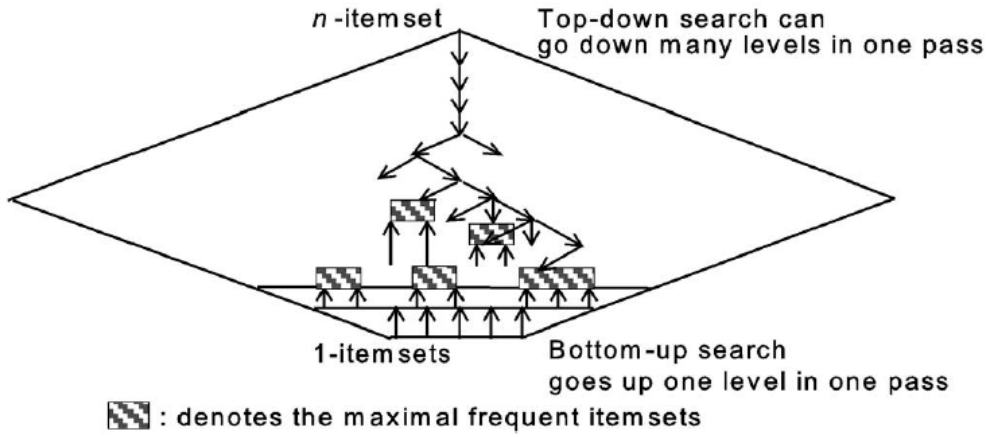

圖 2-6. Pincer-Search 演算法之流程示意圖

2.4 Sampling 演算法

採掘高頻項目集需花費大量時間的主要原因，在於資料庫中的資料量過於龐大；因此，若能夠有效的減少需處理的資料量便能夠增加資料探勘的效率。H. Toivonen[8]期望利用抽樣方法自原始資料庫中抽取樣本，使其能夠儲存於記憶體中，再針對樣本資料庫採掘高頻項目集以減少進行資料探勘所需的時間。由於樣本體積較原始資料小了許多，故能夠快速的採掘出所有的高頻項目集。H. Toivonen 所提出之演算法其樣本大小決定公式，如(式 2-2)所示。

$$|s| \geq \frac{1}{2\varepsilon^2} ln \frac{2}{\delta} \quad \cdots\cdots\cdots\cdots \text{(式 2-2)}$$

(式 2-2)中 s 代表其樣本大小；ε 為其誤差邊界(Error Bound)，代表其誤差的容忍範圍；δ 為其發生誤差的機率。透過控制 ε 和 δ，H.Toivonen 提出之演算法能夠快速的決定所需之樣本大小。不同的 ε 和 δ 所需之樣本大小如表 2-1 所示。由表中可以發現，當 ε 和 δ 愈小，其所需抽取的樣本數愈多；換言之，若期望樣本資料和原始資料之間的誤差愈小，則其所需的樣本數則愈多。

表 2-1. Sampling 演算法中不同的ε和δ所需之樣本大小

ε	δ	$\lvert s\rvert$
0.01	0.01	27,000
0.01	0.001	38,000
0.01	0.0001	50,000
0.001	0.01	2,700,000
0.001	0.001	3,800,000
0.001	0.0001	5,000,000

採用抽樣方法產生之樣本資料庫必定含有抽樣誤差，因此，H. Toivonen 提出之演算法乃利用降低最小支持度門檻值以避免遺漏高頻項目集。(式 2-3)為其降低最小支持度門檻值的公式。

$$low\_fr < min\_fr - \sqrt{\frac{1}{2|s|} ln \frac{1}{\delta}} \quad \cdots\cdots\cdots\cdots \text{(式 2-3)}$$

在(式 2-3)中，*low_fr* 代表降低的最小支持度門檻值；*min_fr* 代表原始的最小支持度門檻值；*s* 代表其樣本大小；δ 為其發生誤差的機率。藉由(式 2-3)便可以有效的降低最小支持度門檻值以避免遺漏高頻項目集。表 2-2 為當δ=0.001 對應不同之樣本大小 *s* 和最小支持度門檻值經過降低公式計算後，其最小支持度門檻值對照表。

表 2-2. 不同數本數量和最小支持度門檻值經過降低程序後之對應值

	Sample size			
min_fr (%)	20,000	40,000	60,000	80,000
0.25	0.13	0.17	0.18	0.19
0.50	0.34	0.38	0.40	0.41
0.75	0.55	0.61	0.63	0.65
1.00	0.77	0.83	0.86	0.88
1.50	1.22	1.30	1.33	1.35
2.00	1.67	1.77	1.81	1.84

H.Toivonen 提出之演算法乃使用抽樣方法自原始資料庫中抽取足夠的樣本，以產生樣本資料庫；由於樣本資料庫所含的資料量較少，故能夠藉此針對樣

本資料進行資料探勘以減少原本所需花費的大量 I/O 時間並增加效率；Sampling 演算法乃採用單純隨機抽樣法來進行抽樣動作；單純隨機抽樣法具有簡單且快速的特性，但由於單純隨機抽樣法在進行抽樣時，並未對抽樣方法做任何的限制，因此較容易產生資料扭曲(Data Skew)；若資料扭曲的情形嚴重時，將造成樣本資料集和原始資料之較的關聯程度較不明顯而不具代表性，故利用樣本資料探勘出的高頻項目集將和由原始資料庫採掘出的項目集不相符；如此一來，利用這些從樣本資料庫中採掘而得的高頻項目集來擬定銷售策略，對於企業而言，是相當危險的。此外，Sampling 演算法所提出之降低最小支持度門檻值的方式，雖然能夠避免遺漏高頻項目集，但此方式將造成不應被視為高頻項目集之候選項目集，卻被視為高頻項目集的情況。故產生之高頻項目集的數量將相當龐大，連帶造成決策者在進行決策選擇時的困難。

第 3 章 理論架構

本節將介紹本研究提出擷取關聯規則之演算法的處理流程和各種理論方法。首先，在研究步驟中將詳細介紹本研究提出之演算法的各個步驟；此外，在理論方法中將介紹本研究提出之演算法中的各種基礎方法，包括抽樣理論、動差保留法和本研究提出之數量化支持度計算法以及數量化信賴度計算法。本研究提出擷取關聯規則之演算法是以分層隨機抽樣和動差保留法為基礎，期望能夠藉由有效的減少資料庫中龐大的資料量，以增加採掘高頻項目集時的效率。此外，傳統購物籃分析在計算項目集的支持度時，並未考慮商品購買數量，因此將造成項目集支持度的失真。本研究將嘗試將商品購買數量在計算支持度時納入考慮，以解決上述問題。

3.1 研究步驟

本研究除期望藉由將資料庫所含龐大的資料量有效的減少，以提高採掘高頻項目集的效率之外，更希望能夠應用本研究提出之數量化支持度計算法，將購買數量做為支持項目集支持度時的加權值，使採掘的高頻項目集和關聯規則對於決策支援的效果能夠更顯著。圖 3-1 為本研究提出之應用分層隨機抽樣和動差保留採掘關聯規則演算法(Stratified sampling and Moment-preserving thresholding Association rule Generation Algorithm, SMAG Algorithm)之區塊流程圖，其詳細介紹如下：

1. 設定模擬程式中產生資料時各項機率參數和交易筆數，並利用模擬程式產生交易資料庫。

2. 本研究提出之 SMAG 演算法乃應用分層隨機抽樣法自資料庫抽取樣本，並針對抽取出之隨機樣本進行採掘高頻項目集和關聯規則的動作。在進行分層隨機抽樣法之前，須先將資料分為若干層以進行抽樣；又由於資料庫所含的

資料量龐大，過於複雜之分類法將花費大量的成本和時間，因此須以有效率且精準的演算法進行資料分類。鑑於以上所述，本研究選擇能夠快速且有效率的進行資料分類之動差保留法做為資料分類演算法。在此步驟中，須先決定分類數量，並以各筆交易的總利潤做為分類依據，再利用動差保留法計算分類門檻值，並依照計算出之利潤門檻值將各筆交易分配到所屬的類別。由於高利潤項目集能夠帶給企業較多的獲利，若採掘結果遺漏了該類項目集將造成企業的損失。鑑於以上所述，本研究採用交易利潤為分類依據，期望透過控制各層樣本數量的分配，使得發生抽樣誤差時，能夠減少遺漏高利潤項目集的機率。非比例配置之分層隨機抽樣法乃是視各層的資料變異情形來決定其樣本分配方式，變異程度較大的層需抽取較多的樣本，反之較少。藉由動差保留法並以交易利潤為依據將所有資料分為若干層後，由於較高利潤的分層其內的資料變異程度較大，故非比例配置之分層隨機抽樣法相較於其它抽樣方式，將在較高利潤的分層中抽取較多的樣本。由於在較高利潤的分層中抽取較多的樣本，因此可減少遺漏高利潤項目集之機率。

3. 利用分層隨機抽樣法自各類別中抽樣足夠的樣本。由於分層隨機抽樣法在進行抽樣時可依比例配置抽樣或非比例配置抽樣，又此二種方式均擁有其優點，因此，本研究將一併採用二種配置方法，並在模擬實驗中探討二種配置法之適用時機。值得注意的是，SMAG 演算法決定樣本數的方法，乃是利用分層隨機抽樣法之樣本決定公式(式 3-4)，其中，由於我們期望樣本資料庫和原始資料庫其平均利潤相同，因此，樣本決定公式中的誤差邊界(Error Bound)可以利用原始資料庫的平均利潤和我們的期望誤差百分比來決定。舉例來說，若原始資料庫中所有交易的平均利潤為 100 元，若期望誤差百分比為 3%，則誤差邊界為 $100 \times 3\% = 3$。

4. 利用 Apriori 演算法掃描利用前一步驟所抽取之隨機樣本，並自樣本中採掘出所有的高頻項目集。經過上一步驟後，便能夠有效的將原本龐大的資料量

減少。由於資料量的大量減少，因此，能夠大量的節省原本在進行採掘高頻目集時所需花費的時間。在此一步驟中，除了可以應用 Apriori 演算法來採掘高頻項目集之外，亦可應用其它採掘高頻項目集演算法[4][8][12]來進行。

5. 此一步驟中將應用本研究提出之數量化支持度計算法來產生高頻項目集。由於傳統購物籃分析在篩選高頻項目集並未考慮購買數量，造成項目集支持度之誤差，故產生之高頻項目集較不具代表性。本研究提出之數量化支持度計算法能夠呈現出項目集之間被購買時之關聯性，因此，利用數量化支持度計算法所產生之高頻項目集對於決策之支持度效果將更顯著。本研究提出之數量化支持度計算法的詳細做法將在其後之 3.2.3 節中說明。

6. 利用前一步驟所產生之高頻項目集來產生關聯規則。此一步驟將應用本研究在提出之數量化信賴度計算法來產生關聯規則。為避免產生之關聯規則不具有代表性而造成決策錯誤，本研究提出之數量化信賴度計算法乃將項目集之購買數量納入考慮以真實表達消費者的購物習慣。由於將購買數量做為加權值，因此能夠產生更具有代表性的關聯規則。本研究提出之數量化信賴度計算法將在 3.2.4 節中做詳細說明。

經由以上的六個步驟，便可產生對於決策者有幫助的關聯規則。雖然本研究提出之 SMAG 演算法在進行動差保留法時，須掃描一次資料庫以獲得各筆交易之利潤以進行分類，此舉將花費一些時間，但由於利分層隨機抽樣法所抽樣之隨機樣本，其所含的交易數量相較於整體資料庫而言少了許多，故能夠有效的減少採掘高頻項目集所需的時間。此外，由於 SMAG 演算法能夠有效率且精確的減少資料量，因此，除可應用 Apriori 演算法進行採掘分析之外，亦可應用其它較 Apriori 演算法更具有效率之相關演算法以增加效率。經模擬實驗証明，本研究所提出之 SMAG 演算法能夠藉由精準且快速的減少資料量，以增加採掘高頻目集和關聯規則的效率，此外，由於將購買數量納入考慮，產生的高頻項目集和關聯規則均較具有代表性，故其決策支援效果將更顯著。

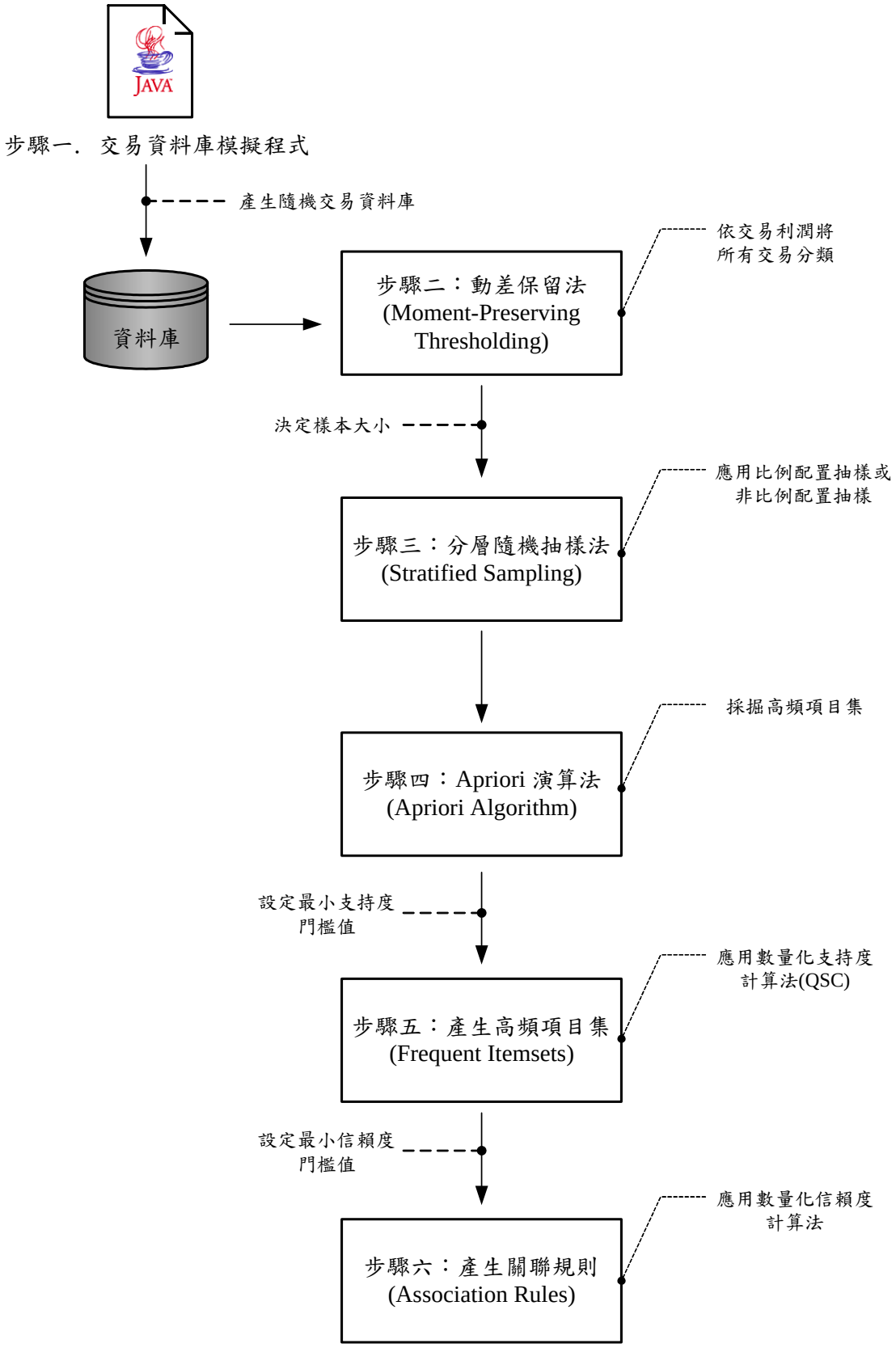

圖 3-1. SMAG 演算法之區塊流程圖

3.2 理論方法

本節除介紹本研究應用之相關技術之外，亦將介紹我們提出之能夠進行數量化購物籃分析之相關演算法。在相關技術方面，由於本研究乃應用抽樣方法以達到減少資料量的目的，因此，本節將介紹數種較重要之抽樣方法其適用時機和其特性。此外，本節將介紹可以快速且有效地將資料分類的演算法—動差保留法。在數量化購物籃分析方面，將介紹本研究提出之數量化支持度計算法和數量化信賴度計算法。

3.2.1 抽樣理論

由母體資料中抽取一部份個體做為樣本，並藉由觀察樣本來對母體做推論的方法，稱為抽樣。由於樣本僅為母體資料中的一部份，因此，若抽樣方法的設計不良，樣本的代表性便不足，故用以推論母體會造成誤差。主要的抽樣方法有：

1. **單純隨機抽樣法(Simple Random Sampling)**：單純隨機抽樣法是指在大小為 N 的母體資料中，抽取大小為 n 的樣本數，而每個可能樣本被選取的機會均相同的抽樣法。由於單純隨機抽樣法在進行抽樣時，對於母體未有任何的限制，因此，容易造成樣本資料不均勻而產生抽樣誤差的情形。單純隨機抽樣法之樣本大小計算公式[20]，如(式 3-1)所示；其乃是透過控制誤差邊界來決定所需之數量。(式 3-1)中 $Z^2_{\alpha/2}$ 代表其信賴區間(Confidence Interval)；S 代表資料集之標準差；B 則代表誤差邊界。

$$n \geq \frac{Z^2_{\alpha/2} S^2}{B^2} \quad \cdots\cdots\cdots\cdots \text{(式 3-1)}$$

2. **分層隨機抽樣法(Stratified Random Sampling)**：分層隨機抽樣法乃是在抽樣之前先將母體依其不同的特徵值，如位置、種類、性質或大小等，劃分為若干層。再分別自每一層中抽取一個單純隨機樣本。分層隨機抽樣法的重要原

則為：同層內，同質程度愈高愈好，層和層之間的同質程度愈低愈好。分層隨機抽樣法分配各層樣本個數的方法有二：

i. 比例配置(Proportional Allocation)：各層之樣本數係依照各層大小對應母體的比例來決定。此種配置法簡單且容易進行，尤其當各層變異數不大時，效果相當良好。一般來說，比例配置對於個數多的層抽取較多的個體，個數少的層則抽取較少的個體。當母體為 N 且分為 k 層，各層的個數為 $N_1,N_2,\cdots N_k$，若欲抽取 n 個個體做為樣本，則各層所需抽取的個數為：

$$n_i = n\left(\frac{N_i}{N}\right) \qquad i = 1,2,...,k \quad \cdots\cdots\cdots\cdots \text{(式 3-2)}$$

ii. 非比例配置(Disproportional Allocation)：利用比例配置，雖然簡單，但未考慮各層變異情形不同，而未能在各層中抽取適當的樣本數。一般來說，非比例配置是對同質程度高的層抽取的樣本比例較小，對同質程度低的層抽取的樣本比例較大。當母體為 N 且分為 k 層，各層的個數為 $N_1,N_2,\cdots N_k$，此外，各層的標準差為 δ_i，若樣本大小為 n，則各層所需抽取的個數為：

$$n_i = n\left(\frac{N_i\delta_i}{\sum N_i\delta_i}\right) \qquad i = 1,2,...,k \quad \cdots\cdots\cdots\cdots \text{(式 3-3)}$$

分層隨機抽樣之樣本大小決定公式[20]，如(式 3-4)；其中，w_i 為各層數量佔全體資料之比例；B 為誤差邊界。分層隨機抽樣法可以藉由控制誤差邊界來決定樣本大小；此外，雖然比例配置和非比例配置分層隨機抽樣法使用相同之樣本數量決定公式，但由於其分配方式不同，因此在不同配置方式下，即使樣本數量相同，各層被分配所需抽取之數量不會相同。

$$n=\frac{\sum_{i=1}^{L}\frac{N_i^2\delta_i^2}{w_i}}{N^2D+\sum_{i=1}^{L}N_i\delta_i^2}\ ,\ \left\{\begin{array}{l}\text{估計平均數}(\mu)\text{時，}D=\frac{B^2}{4}\\ \text{估計總數}(\tau)\text{時，}D=\frac{B^2}{4N^2}\end{array}\right\}\ \cdots\text{(式 3-4)}$$

3. **群集抽樣法(Cluster Sampling)**：群集抽樣法在抽樣前會先將母體依特定標準分為數個小群集；抽樣時則以群集為單位，自所有群集中抽樣數個群集做為樣本。群集抽樣的風險較高，但可節少時間、人力和經費；此外，和分層隨機抽樣法不同的，本法適用於群集間的差異數不大，而群集內的變異數較大時。

4. **系統抽樣法(Systematic Sampling)**：系統抽樣法乃是將母體所有的個體依次排列，並分為許多間隔，每隔若干個個體抽取數個個體做為樣本；系統抽樣法是一種等距的抽樣法，故又可稱為等距抽樣法。

3.2.2 動差保留法

動差保留法是[19]一種能夠簡單且快速的將影像像素灰階值分類的技術。影像處理在實際的應用上，如果我們能將要處理的主體和不需要存在的背景分離，則後續的處理動作將變得比較簡單。因此如何能夠選擇適當、正確的門檻值(Threshold Value)便是一項重要的工作。本研究將介紹一自動門檻值擷取技術—動差保留法(Moment Preserving Thresholding Approach)作為交易資料庫分類門檻值決定的依據。

動差保留法能夠不經過煩雜的運算和搜尋，而將影像快速的分成二階(Bi-Level Thresholding)或二階以上(Multi-Level Thresholding)。大部份的分類演算法對於影像主體和背景影像灰階值對比很強時，擁有較佳的效果，反之，則無法精確的找出門檻值。和相關的分類演算法比較，動差保留法能夠更快速且精準的找出且有代表性的門檻值。動差保留法對於二階分類的處理如下：

假設 f(x,y)為影像上座標為(x,y)點之灰階值。T 為門檻值。若欲將影像分成二個群集(主體與背景)，則選定適當的門檻值 T

f(x,y) ＞ T 則 f(x,y) 設為主體。

f(x,y) ≦ T 則 f (x,y) 設為背景。

由於每張影像的特性都會有些不同，因此適用於每張影像的門檻值可能都會不一樣，因此如何找出每張影像的最佳門檻值是非常重要的課題。動差保留法便能夠快速且精準的找出每張影像最具代表性的門檻值，其原理如下：

若一擁有 n 個像素值(Pixel)的影像 f，其在影像位置(x,y)的灰階值以 f(x,y)表示。若想找一最佳門檻值， 將影像 f 分為二群。則定義 m_i 為影像 f 的第 i 階動差(Moment)：

$$m_i = (1/n)\sum_x \sum_y f^i(x,y), \qquad i = 1,2,3,... \quad \cdots\cdots\cdots\cdots \text{(式 3-5)}$$

動差亦可由影像 f 的灰階分佈圖來計算，其表示式為：

$$m_i = (1/n)\sum_j n_j (z_j)^i$$
$$= \sum_j p_j (z_j)^i \qquad \cdots\cdots\cdots\cdots \text{(式 3-6)}$$

其中 n_j 為影像 f 中具有灰階值 z_j 之總數，而 $p_j = n_j/n$。

在動差保留法中，若欲將影像 f 分為 N 個群集，則需要 N-1 個門檻值和 2N-1 個動差，因此，若欲將影像 f 分成二個群集，需要 1 個門檻值和 3 個動差。此外，影像 f 可視為是由一張只包含 2 個灰階值 z_0 和 z_1 的影像經模糊化後的結果，其中 $z_0 > z_1$。動差保留法會將影像 f 中高於該門檻值的灰階值以 z_1 取代，反之，則以 z_0 取代，因此，可得到二階影像(Bi-Level Image)g。影像 g 可被視為未模糊化前之影像 f。相對的，影像 g 的前三個動差可用下列式子表示：

$$m_i^{'} = \sum_{j=0}^{1} p_j (z_j)^i, \qquad i = 1,2,3 \cdots\cdots\cdots\cdots \text{(式 3-7)}$$

影像 g 的前三個動差守恆定理為：

$$m_i^{'} = m_i, \qquad i = 1,2,3 \cdots\cdots\cdots\cdots \text{(式 3-8)}$$

此外，由於：$p_0 + p_1 = 1$，因此，可得：

$$\begin{aligned} p_0 z_0^0 + p_1 z_1^0 &= m_0 \\ p_0 z_0^1 + p_1 z_1^1 &= m_1 \\ p_0 z_0^2 + p_1 z_1^2 &= m_2 \\ p_0 z_0^3 + p_1 z_1^3 &= m_3 \end{aligned} \quad \cdots\cdots\cdots\cdots \text{(式 3-9)}$$

根據 Szego[7]和 Tabatabai[1]二人指出，可以由下列三個步驟可解得(式 3-9)中的 p_0、p_1、z_0、z_1：

(Ⅰ) 解出下列輔助方程式中的 c_0、c_1：

$$\begin{aligned} c_0 m_0 + c_1 m_1 &= -m_2 \\ c_0 m_1 + c_1 m_2 &= -m_3 \end{aligned} \quad \cdots\cdots\cdots\cdots \text{(式 3-10)}$$

(Ⅱ) 解出下列式子的 z 值：

$$z^2 + c_1 z + c_0 = 0 \cdots\cdots\cdots\cdots \text{(式 3-11)}$$

(Ⅲ) 將(式 3-11)所解出的 z 值代入(式 3-9)後可解得 p_0、p_1。

其最後結果如(式 3-12)。

$$
\begin{cases}
c_d = \begin{vmatrix} m_0 & m_1 \\ m_1 & m_2 \end{vmatrix} \\
c_0 = (1/c_d)\begin{vmatrix} -m_2 & m_1 \\ -m_3 & m_2 \end{vmatrix}; c_1 = (1/c_d)\begin{vmatrix} m_0 & -m_2 \\ m_1 & -m_3 \end{vmatrix} \\
z_0 = (1/2)\left[-c_1 - (c_1^2 - 4c_0)^{1/2}\right] \\
z_1 = (1/2)\left[-c_1 + (c_1^2 - 4c_0)^{1/2}\right] \\
p_d = \begin{vmatrix} 1 & 1 \\ z_0 & z_1 \end{vmatrix}; p_0 = (1/p_d)\begin{vmatrix} 1 & 1 \\ m_1 & z_1 \end{vmatrix}; p_1 = 1 - p_0
\end{cases}
\quad \cdots\cdots\cdots\cdots \text{(式 3-12)}
$$

因此，可由 $p_0 = (1/n)\sum_{z_j \le t} n_j$ 求得最佳門檻值 t。由於動差保留法能夠精準且快速的將資料分類，因此，本研究選用動差保留法做為資料分類演算法。

3.2.3 數量化支持度計算法

傳統購物籃分析在計算項目集之支持度時，均以項目集出現的交易筆數做為考量，而不考慮商品購買數量。因此，在計算項目集之支持度時將造成失真，故產生的高頻項目集亦較不具代表性。傳統購物籃分析的項目集支持度計算公式如(式 1-1)。由(式 1-1)可以發現傳統購物籃分析是以含有項目集的交易筆數做為支持度計算之依據，故會造成項目集支持度失真的情形。舉例來說：某一交易 T={A, A, A, B, B}。傳統演算法會將該交易視為{A, B}，故計算項目集{A}、{B}、{A,B}的支持度時，本交易的加權值均為 1。利用此種方式計算項目支持度會造成出現次數少，但均被大量購買的商品被忽略。因此，本研究期望將商品購買數量在計算商品支持度時納入考慮，以避免上述情況。本研究提出之一數量化支持度計算法(Quantitative Support Counting, QSC)，其公式如下：

$$
\text{支持度(項目集)} = \frac{\text{項目集共同出現的次數}}{\text{Avg(所有長度為1的子項目集出現的次數加總)}} \quad \cdots\text{(式 3-13)}
$$

項目集共同出現的次數是指此一商品在資料庫中共同被購買的次數。以前一

例來說：該筆交易中項目集{A}出現了 3 次，而{B}出現了 2 次，因此，在該筆交易中，項目集{A}和{B}的出現次數，分別為 3 和 2。此外，計算多項目之項目集的出現次數時，我們則採用最小購買數量來做為加權值。換言之，在計算項目集{A, B}之出現次數時，由於{A}的出現次數為 3，而{B}的出現次數為 2，由於採用最小購買數量，因此，在此筆交易中項目集{A,B}共同出現的次數為 2 次。由於項目集{A,B}共同出現的次數為 2，而此項目集包含二個“1-項目集”{A}和{B}，又項目集{A}和{B}的出現次數分別為 3 和 2，故以此筆交易為例，項目集{A,B}的支持度為：$2/Avg(3+2)=0.8$。由於本研究所提出的支持度計算公式(式 3-11)中，分母為項目集所有長度為 1 的子項目集出現的次數總和，因此，若運算 QSC 法來計算“1-項目集”支持度時，其值均為 1。

採用 QSC 法，可避免項目集雖然出現的交易筆數低於最小支持度門檻值，但每次都被大量購買的情形。此外，由於項目集的支持度愈高，代表其項目之間的購買關聯愈高。因此，產生之高頻項目集較具有代表性，對於決策者在擬定銷售策略時的幫助更大。舉例來說，若一“3-項目集”{A,B,C}，其支持度為 1，代表所有的交易中，消費者只要買了 N 個 A，便會買 N 個 B 和 N 個 C。若企業能夠獲得此種資訊，在擬定策略時，或許便可以將 A、B、C 三種商品放置在同一個商品架上，以創造更多的購買機會。由以上所述可以發現，利用此一考慮購買數量之支持度計算法所產生的高頻項目集將較傳統購物籃分析所採用的項目集支持度計算法更具有意義。

3.2.4 數量化信賴度計算法

若 X 和 Y 為二個獨立的高頻項目集，則此二項目集之間的關聯規則可表示為：$X \rightarrow Y$。以傳統購物籃的計算方式而言，此一規則之信賴度為：

$$\text{信賴度}(X \rightarrow Y)=\frac{\text{含有項目集}\{X \cup Y\}\text{的交易筆數}}{\text{含有項目集}\{X\}\text{的交易筆數}}=\frac{\text{支持度}(X \cup Y)}{\text{支持度}(X)} \quad \cdots \text{(式 3-14)}$$

由(式 3-14)可以發現，如同計算項目集支持度時，傳統購物籃分析在計算關聯規則信賴度時，是以項目集出現的交易筆數為其依據，而未考慮購買數量。故傳統購物籃分析所產生之關聯規則較不具代表性；此外，其對於決策支援之效果亦較差。和傳統購物籃分析不同，本研究提出之關聯規則信賴度計算公式為：

$$\text{信賴度}(X \rightarrow Y) = \frac{\text{項目集}\{X \cup Y\}\text{共同出現的次數}}{\text{項目集}\{X\}\text{共同出現的次數}} \quad \cdots \text{(式 3-15)}$$

由於在採用數量化支持度計算法時，便已計算過所有項目集的共同出現次數，因此，在產生關聯規則時便不需重新計算，而可以快速的計算出所有關聯規則之信賴度。由(式 3-14)和(式 3-15)可以發現，傳統信賴度計算法對於項目集在一筆交易中，不論被共同購買多少次，其對應整體項目集支持度加權值均為 1，造成支持度失真的情形；由以上所述可以發現，傳統信賴度計算法無法完整的呈現項目集的交易情形，因此，產生之關聯規則亦較不具說服力。本研究提出之數量化信賴度計算法乃將交易中，項目集其同被購買的次數做為其項目集支持度之加權值；換言之，項目集在一筆交易中，若被共同購買 5 次，其對應整體項目集支持加權值則為 5，如此一來，便可避免支持度失真的情形。舉例來說：某一交易 T={A, A, A, A, A, B}。以此交易而言，在傳統信賴度計算法下，關聯規則“$A \rightarrow B$”的信賴度為：1/1＝100%；若應用本研究提出之數量化信賴度計算法，則關聯規則“$A \rightarrow B$”的信賴度為：1/5＝20%。由此一例子可以發現，原始交易中該消費者購買了 5 個 A 和 1 個 B，換言之，商品 A 和商品 B 之間的關聯性並不高，但應用傳統計算法所得之關聯規則信賴度卻為 100%，若決策者信賴此一關聯規則並擬定相關銷售策略，將造成相當大的損失；反觀應用本研究提出之數量化信賴度計算法所得之關聯規則信賴度為 20%，故能夠充分的表現出其購買情形，亦能夠避免錯估信賴度而造成決策錯誤的情形。由於數量化信賴度計算法將項目集購買數量納入考慮，故經由數量化信賴度計算法產生之關聯規則較具有代表性；此外，由於能夠較完整的呈現項目集的交易情形，故對於企業的決策者而言，其參考價值亦較高。

第 4 章 模擬實驗

本節將就各種不同的角度，針對本研究提出之 SMAG 演算法做整體的效率評估。此外，將針對 SMAG 演算法和單純隨機抽樣演算法就產生之高頻項目集數量和遺漏(Missed)之高頻項目集數量做一分析比較，以評估 SMAG 演算法之抽樣準確度和穩定度。

4.1 實驗環境

實驗中所使用的實驗平台為 Intel Pentium 1.7 GHz、512MB DDR RAM、作業系統為 Microsoft Windows XP、程式撰寫工具為 Java SE 1.4.1。在實驗中使用的參數之意義如表 4-1。本研究採用模擬方式產生交易資料庫，其參數設定如表 4-2 所示。

表 4-1. 模擬實驗中所有參數意義

D	原始資料庫所含的交易數量
N	資料庫包含的項目數量
S	自原始資料庫抽取的樣本數量
Min_sup(%)	最小支持度門檻值
Min_conf(%)	最小信賴度門檻值
Mpt_classes	動差保留法分類數量

表 4-2. 交易資料庫產生程式預設參數

商品數量 (N)	交易筆數 (D)	商品種類出現機率			各類商品出現機率			商品數量出現機率		
		<=5 種	6-21 種	>=22 種	高利潤	中利潤	低利潤	<=2 個	3-7 個	>=8 個
26	視需求而定	60%	30%	10%	20%	60%	20%	70%	20%	10%

為分析比較 SMAG 演算法的效能，我們將利用單純隨機抽樣演算法和 SMAG 演算法做比較。在模擬實驗中所使用的單純隨機抽樣演算法乃是應用單純隨機抽樣法自交易資料庫中抽樣樣本，並自樣本中利用 Apriori 演算法來採掘高頻項目集。值得注意的是，我們採用的單純隨機抽樣演算法並未如同 H. Toivonen 提出之 Sampling 演算法經過降低最小支持度門檻值的處理。Sampling 演算法在採掘高頻項目集的過程中，須經過一降低最小支持度門檻值的程序。雖然藉由降低最小支持度門檻值可以避免遺漏高頻項目集，但將造成採掘的高頻項目集數量非常多；而過多的高頻項目集會讓決策者較難從中選定特定銷售組合來擬定對企業本身有利的決策。鑑於以上所述，在模擬實驗中的單純隨機抽樣演算法將不採用該降低程序。此外，在 Sampling 演算法中，H. Toivonen 亦提出透過控制ε和δ來決定樣本的方法(式 2-1)，但由於相當難決定這兩個參數以產生適當的樣本數來和 SMAG 演算法比較，因此，模擬實驗中的隨機抽樣演算法將利用 3.2.1 節中介紹的單純隨機抽樣法之樣本決定公式來計算樣本數。

4.2 傳統支持度計算法之效能評估

本節將以未使用 QSC 計算法之傳統支持度計算方式，就各種不同角度來分析本文所提出之演算法的效率和效果。圖 4-1 為採用傳統支持度計算法時，不同演算法的高頻項目集採掘效率之比較。在實驗中我們將最小支持度門檻值(Min_sup)訂為 10%；此外，SMAG 演算法之動差保留法分類數為 2。由圖 4-1

可以發現本研究提出之 SMAG 演算法不論採用比例配置或非比例配置，其採掘效率均較 Apriori 演算法為佳；和單純隨機抽樣演算法比較，在樣本數相同時，SMAG 演算法需花費些許的時間以動差保留法將資料分類導致採掘時間較長；但由於在相同的誤差邊界下時，單純隨機抽樣演算法所需的樣本數較 SMAG 演算法所採用的分層隨機抽樣法為多，因此，我們可以發現，當單純隨機抽樣演算法之樣本數由 1000 筆資料增加到 4000 筆資料時，其所花費的時間反而較 SMAG 演算法多。經過此實驗，証明本研究所提出之 SMAG 演算法能夠藉由有效的減少資料量來增加採掘效率。此外，在實驗中，我們可以發現採用比例配置分層隨機抽樣法的 SMAG 演算法較採用非比例配置的 SMAG 演算法有較佳的採掘效率。其主要的原因在於非比例配置的 SMAG 演算法會抽取較多的高利潤交易以避免遺漏含有高利潤的高頻項目集；但由於高利潤交易的長度通常較長，也就是包含的項目較多，因此，經由非比例配置之 SMAG 演算法所產生之樣本資料庫，其所含的交易之平均長度較比例配置之 SMAG 演算法所產生之樣本資料庫為長，故需花費較多的時間進行採掘。

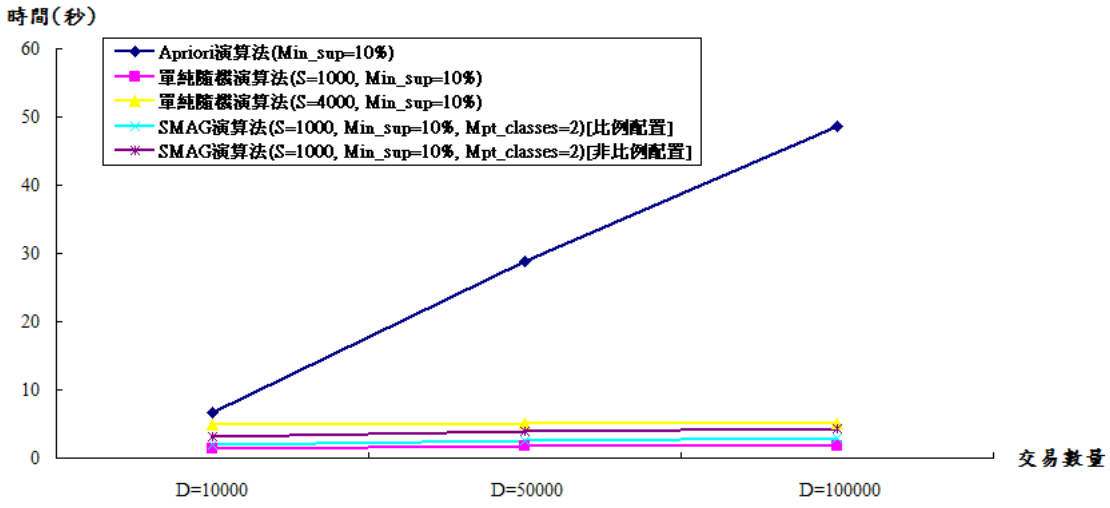

圖 4-1. 高頻項目集採掘效率之比較(傳統支持度計算法)

除分析 SMAG 演算法之採掘效率之外，本節亦將分析 SMAG 演算法之採掘準確度。首先，使用 SMAG 演算法和單純隨機抽樣演算法自資料庫中抽取樣本，並利用樣本採掘樣本高頻項目集；接著利用 Apriori 演算法自原始資料庫中採掘原始高頻項目集；比較樣本高頻項目集和母體高頻項目集便可分析不同演算法之採掘準確度。實驗中原始交易資料庫包含 10000 筆交易記錄；最小支持度門檻值(Min_sup)為 10%，動差保留法之分類數為 2。此外，由於不同的抽樣方法，其樣本決本公式不同；因此，實驗中我們將平均利潤期望誤差百分比訂為 3%，並分別利用(式 3-1)和(式 3-4)來決定單純隨機抽樣演算法和 SMAG 演算法在相同期望誤差百分比時所需之樣本數；由(式 3-1)計算出單純隨機抽樣演算法在平均利潤期望誤差百分比為 3%時，需抽取 4000 筆交易做為樣本；而在相同誤差百分比時，SMAG 僅需抽取 1000 筆交易。

圖 4-2 為 SMAG 演算法和單純隨機抽樣演算法遺漏之高頻項目集數量折線圖；橫軸為進行連續 30 次抽樣的抽樣別；縱軸為遺漏的高頻項目集數量。由圖 4-2 可以發現，採用非比例配置的 SMAG 演算法在連續 30 次的抽樣實驗中，幾乎未遺漏任何高頻項目集，其準確度相當高。此外，當單純隨機抽樣演算法其樣本數由 1000 筆增加為 4000 筆時，其準確度和穩定度均呈現小幅提昇；但和採用比例配置的 SMAG 演算法比較，當樣本數增加為 4000 筆，單純隨機抽樣演算法遺漏的高頻項目集數量雖然減少，但整體而言，其抽樣穩定度和準確度仍較 SMAG 演算法為弱。

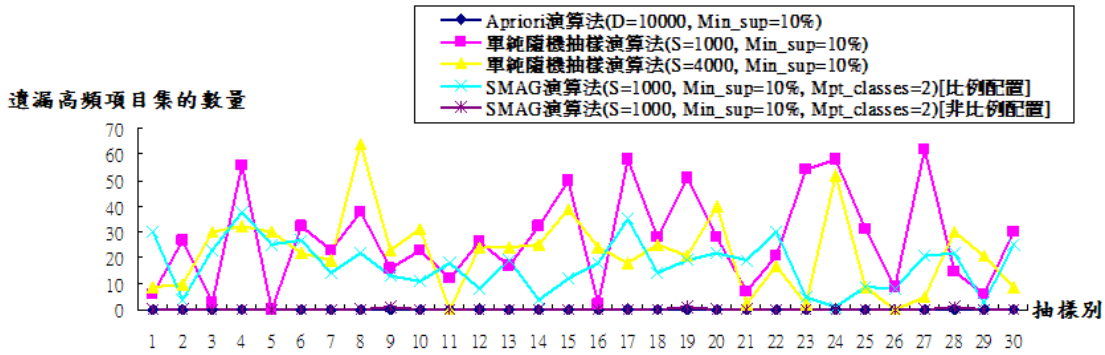

圖 4-2. 遺漏之高頻項目集數量折線圖(傳統支持度計算法)

圖 4-3 為 SMAG 演算法和單純隨機演算法採掘產生之高頻項目集數量折線圖。圖表的橫軸為進行連續 30 次抽樣的抽樣別；縱軸為高頻項目集的數量。由圖 4-2 和圖 4-3 可以發現，採用非比例配置之 SMAG 演算法雖具有相當高的準確度，但其產生之高頻項目集數量較多且採掘速度較慢；而比例配置之 SMAG 演算法則是準確度較低，但產生之高頻項目集數量較少且採掘速度較快。因此，若期望採掘準確度高，則應採用非比例配置之 SMAG 演算法；若期望能夠快速完成採掘過程，則應採用比例配置之 SMAG 演算法。此外，由圖中可以發現，當單純隨機抽樣演算法之樣本數增加時，其產生之高頻項目集數量亦較穩定，較少出現高頻項目集數量很多或很少的情形，但其穩定度仍較為不足。

整體而言，若將 SMAG 演算法以傳統支持度計算法應用於購物籃分析時，除能夠增加採掘效率，亦能夠準確的採掘出高頻項目集以做為決策參考。此外，由本節的實驗可以瞭解，在相同的誤差容忍範圍下，SMAG 演算法能夠以較少的樣本來進行採掘，故能夠節省較多採掘高頻項目集時所需的時間；此外，經過模擬實驗証明，SMAG 演算法在穩定度和準確度上都有很好的效果。

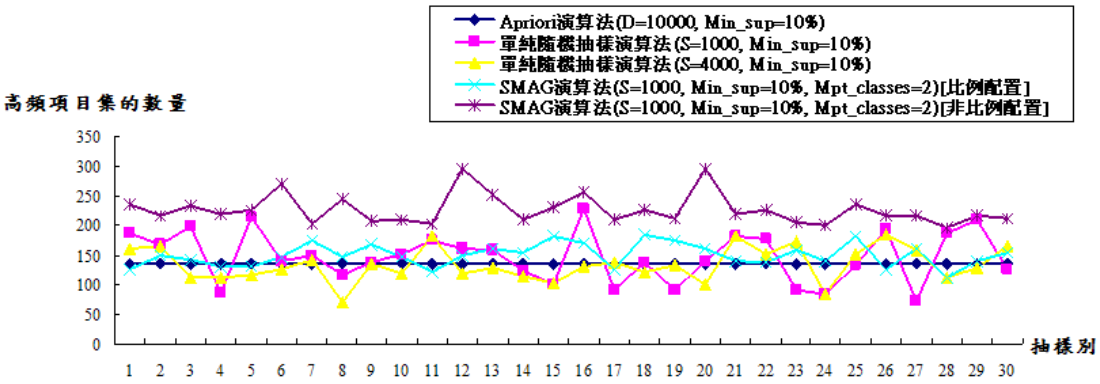

圖 4-3. 高頻項目集數量折線圖(傳統支持度計算法)

4.3 應用 QSC 計算法之效能評估

本節將探討本研究所提出之 SMAG 演算法應用 QSC 計算法時之效率和效果。圖 4-4 為以 QSC 計算法為支持度計算基礎時，各種演算法的高頻項目集採掘效率之比較。實驗中我們採用三個所含資料量不同的交易資料庫，其所含的交易數量分別為 10000、50000 和 100000；此外，實驗中最小支持度門檻值均訂為 10%；誤差邊界訂為 3%，由(式 3-1)計算出單純隨機抽樣演算法需抽取 4000 筆交易資料做為樣本；在相同誤差邊界下，藉由(式 3-4)可計算出 SMAG 演算法需抽取 1000 筆交易資料為樣本；SMAG 演算法的動差保留法分類數 Mpt_classes = 2。由圖 4-4 中可以發現，應用 QSC 計算法進行採掘高頻項目集時，由於必須考慮項目集購買數量，所以即使交易資料數量相同，其進行採掘時所需花費的時間仍較傳統方式來得久；因此，應用 QSC 計算法時，SMAG 演算法對於採掘高頻項目集的效率改善將更加明顯。

圖 4-4 可以証明，不論 SMAG 演算法或單純隨機抽樣演算法均能夠有效的增加採掘效率。此外，我們可發現，當樣本數 S = 1000 時，單純隨機抽樣演算法和 SMAG 演算法進行採掘高頻項目集所需的時間相差不多；但當單純隨機抽樣演算法之樣本數增加為 S = 4000 時，便需花費較 SMAG 演算法多的採掘時間。雖然在樣本數相同時，SMAG 演算法和單純隨機抽樣演算法進行採掘高頻項目

集所需花費的時間不多，但由於當誤差邊界相同時，SMAG 演算法需抽取的樣本數較少，因此，SMAG 演算法能夠較單純隨機抽樣演算法節省更多採掘高頻項目集時所需的時間。

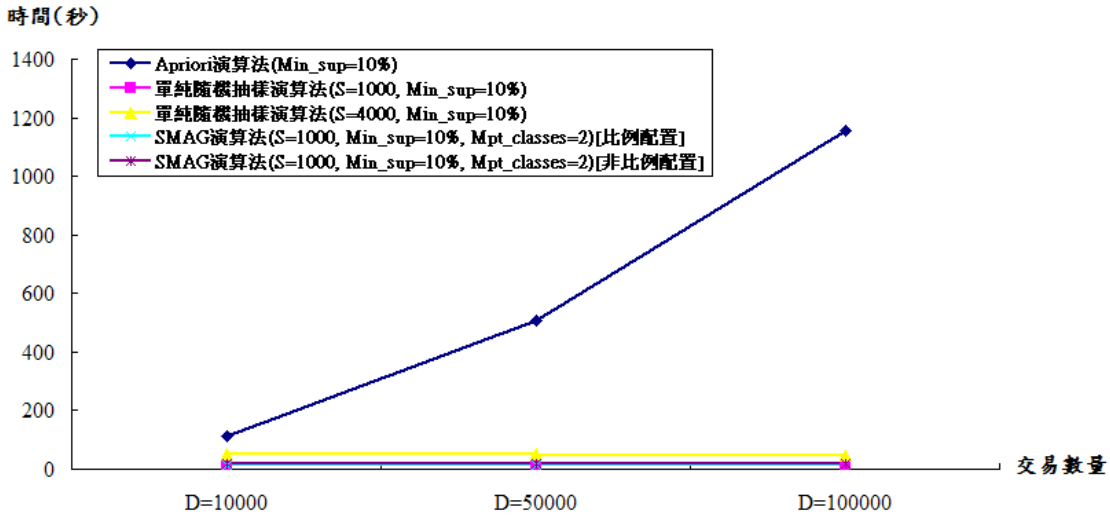

圖 4-4. 採掘效率之比較(QSC 計算法)

如同在 4.2 節的實驗，本節我們同樣將分析在應用 QSC 計算法時，各種演算法的採掘準確度；和 4.2 節不同的是，本節在計算項目集支持度時乃是應用本研究提出 QSC 計算法。在實驗中，我們同樣使用 SMAG 演算法和單純隨機抽樣演算法自資料庫中抽取樣本，並自樣本中採掘樣本高頻項目集；再和利用 Apriori 演算法自原始資料庫中出之原始高頻項目集比較以分析採掘準確度。原始交易資料庫包含 10000 筆交易記錄；最小支持度門檻值(Min_sup)為 10%，動差保留法之分類數為 2。此外，實驗中我們仍將平均利潤期望誤差百分比訂為 3%，因此，我們可由(式 3-1)計算出單純隨機抽樣演算法需抽取 4000 筆交易做為樣本；而在相同誤差百分比時，SMAG 需抽取 1000 筆交易。

圖 4-5 為應用 QSC 計算法時，演算法遺漏之高頻項目集數量折線圖；其中，橫軸為進行連續 30 次抽樣的抽樣別；縱軸為遺漏的高頻項目集數量。由圖中可以發現，非比例配置之 SMAG 演算法和樣本數為 4000 時的單純隨機抽樣演算法在連續 30 次的抽樣中，僅各遺漏一個高頻項目集，因此，証明其準確度相當高。

此外，由圖 4-5 我們亦可發現，應用 QSC 計算法採掘高頻項目集時，比例配置之 SMAG 演算法和樣本數為 1000 時的單純隨機抽樣演算法其效果相差不多，其準確度之改善並未如同應用傳統支持度計算法時明顯；但在穩定度上，比例配置之 SMAG 演算法仍有較好的效果。整體而言，在應用 QSC 計算法採掘高頻項目集時，單純隨機抽樣演算法和 SMAG 演算法其遺漏的高頻項目集數量都很少，換言之，二種演算法均能夠有效的採掘出所有的高頻項目集。

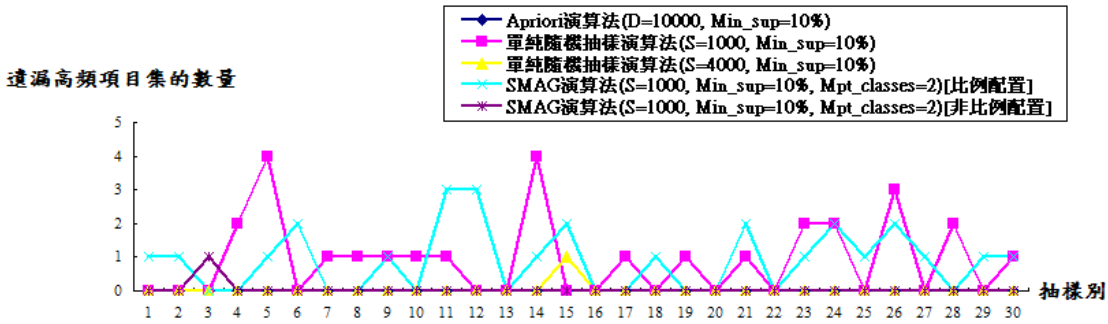

圖 4-5. 遺漏之高頻項目集數量折線圖(QSC 計算法)

圖 4-6 為 SMAG 演算法和單純隨機演算法採掘產生之高頻項目集數量折線圖。圖表的橫軸為進行連續 30 次抽樣的抽樣別；縱軸為高頻項目集的數量。綜合觀察圖 4-5 和圖 4-6，我們可以發現樣本數為 4000 的單純隨機抽樣演算法，在應用 QSC 計算法進行採掘高頻項目集時，其準確度非常高；由圖 4-5 可以發現其幾乎未遺漏任何的高頻項目集，此外，由圖 4-6 可以發現其產生的高頻項目集數量和原始高頻項目集數量幾乎相同，故証明該演算法在應用 QSC 計算法進行採掘高頻項目集時，具有非常顯著的效果。經由分析後我們發現，由於抽取的樣本數較多，亦較能夠完整呈現出項目集之的購買關聯程度，故能夠有效的採掘高頻項目集。比較單純隨機抽樣演算法和 SMAG 演算法，當樣本數同樣為 1000 時，SMAG 演算法和單純隨機抽樣演算法均能夠有效的採掘高頻項目集，但 SMAG 演算法有較佳的穩定度；此外，雖然當樣本數增加為 4000 時，單純隨機抽樣演算法能夠準確的採掘高頻項目集，但其所需花費的採掘時間較 SMAG 演算法為多。

由本節的實驗可以發現，應用 QSC 計算法進行採掘高頻項目集時，不論單純隨機抽樣演算法或 SMAG 演算法均能夠大量的節省採掘高頻項目集時所需花費的時間。此外，經本節的實驗証明，在相同的樣本數時，SMAG 演算法較單純隨機抽樣演算法有更佳的採掘效率和效果。

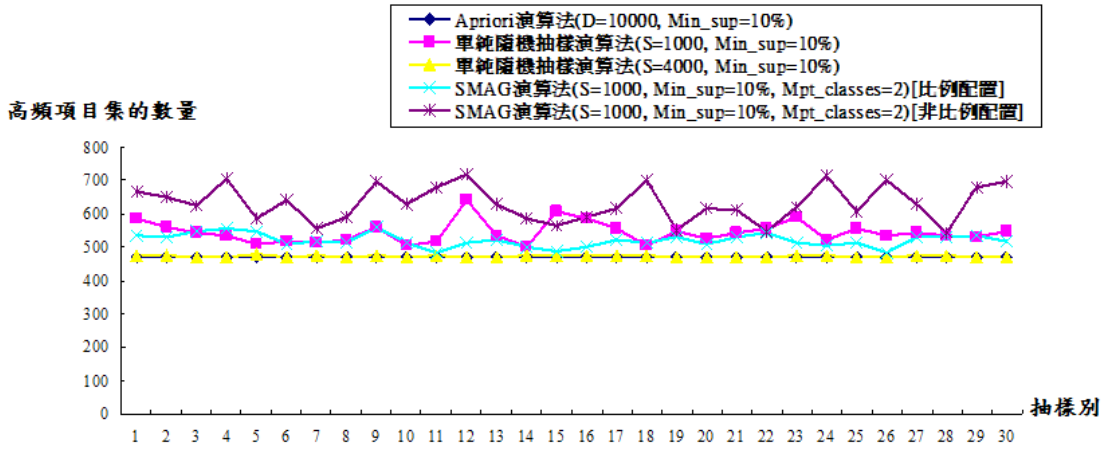

圖 4-6. 高頻項目集數量折線圖(QSC 計算法)

第 5 章 結論與未來研究

5.1 結論

就購物籃分析而言，許多相關之採掘演算法均嘗試應用不同的資料結構，期望能夠增加採掘效率；經實驗証明，這些演算法雖然能夠有效的減少採掘高頻項目集所需的時間，但是卻忽略了造成採掘效率低落的主要原因，乃在於資料庫中所含的資料量過於龐大，因此需花費大量的時間才能夠從中採掘出具有價值的資訊。本研究嘗試利用和其餘演算法不同的思考方式，期望能夠藉由減少資料量的方式來縮短採掘高頻項目集所需花費的時間以增加效率。

由於傳統購物籃分析在計算項目集支持度和關聯規則信賴度時未將購買數量納入考慮，造成支持度和信賴度的失真，亦使得採掘出之項目集和關聯規則的參考價值大為降低。鑑於以上所述，本研究提出數量化支持度計算法(QSC)和數量化信賴度計算法，期望能夠將商品購買數量納入計算以避免失真的情形。

在本研究中，提出了一個以分層隨機抽樣和動差保留法為演算基礎的 SMAG 演算法，來處理資料探勘中最被廣為討論的購物籃分析。經模擬實驗証明，本研究提出之 SMAG 演算法具有下列優點：

1. 能夠快速且有效的減少資料量
2. 使用較少的記憶體空間
3. 減少磁碟進行 I/O 所需的時間
4. 採掘出之項目集和關聯規則具有較高的參考價值

在實驗中我們發現，SMAG 演算法不論應用傳統支持度計算法或本研究提出之 QSC 計算法時，均能夠藉由準確且有效的減少資量來增加其採掘效率；和 Apriori 演算法比較，當交易資料庫中所含的資料量愈多時，SMAG 演算法便能

夠較 Apriori 演算法節省愈多的時間。

現代的企業每天須面對競爭對手不斷的挑戰，若無法利用有效的方式來擬定正確的銷售策略將無法在商場上取得有利的位置。資料探勘能夠幫助企業自大量的資料中採掘出有意義的資訊以做為擬定策略之用；但若花費過多的時間來進行資料探勘，便無法有效且立即的產生對於決策支援有幫助的資訊，而失去進行資料探勘的意義。本研究提出之 SMAG 演算法能夠快速且有效的自龐大的原始資料採掘出對於企業決策有幫助的高頻項目集和關聯規則；此外，應用本研究提出之數量化支持度計算法和數量化信賴度計算法所得之高頻項目集和關聯規則亦具有較高的參考價值。

5.2 未來研究

本研究乃是期望能夠利用抽樣方法來減少龐大的原始資料以增加採掘高頻項目集的效率；經模擬實驗後，証明有良好的效果。不過，在實驗過程中，我們發現，採用非比例配置分層隨機抽樣法的 SMAG 演算法來進行採掘高頻項目集時，雖然幾乎未遺漏任何的高頻項目集，但所產生的高頻項目集數量會較原始高頻項目集多；換言之，採用非比例配置分層隨機抽樣法的 SMAG 演算法會將在原始資料庫中不為高頻項目集的候選項目集視為高頻項目集，而造成高頻項目集較多的情形。產生這種情況的原因在於本研究乃是利用交易利潤做為分類依據，又交易利潤高通常代表交易長度較長；因此，當採用非比例配置分層隨機抽樣法的 SMAG 演算法藉由各個群集的變異情形來分配樣本數時，由於交易利潤高的群集其變異數較大，其所分配需抽取的樣本數亦較多；故造成樣本資料中交易利潤較高、交易長度較長、資料較離散的情形。由於樣本資料較離散，所以非比例配置分層隨機抽樣法的 SMAG 演算法所採掘出的高頻項目集數量才會比較多。因此，在未來期望能夠針對此問題做更深入的探討，期望能使 SMAG 演算法得到更快和更穩定的採掘效率。

由於購物籃分析中，對於如何訂定最小支持度門檻值和最小信賴度門檻值並無一較明確的規範，因而造成進行購物籃分析時的盲點；在購物籃分析中，最小支持度門檻值和最小信賴度門檻值是非常重要的二個參數；利用這二個參數才能夠判定項目集是否為高頻項目集和判定關聯規則是否具有足夠的信賴度來做為決策之參考。換言之，若這二個參數不具有足夠的代表性和關鍵性，則採掘演算法的採掘效率再高，都是沒有意義的；因為採掘出來的高頻項目集和產生的關聯規則極可能都不具代表性，而無法提供決策者任何參考價值。因此，在未來研究中，我們期望能夠針對如何訂定具有代表性和關鍵性的門檻值做相關研究，用以幫助決策者在進行購物籃分析前，能夠藉由快速且正確的訂定最小支持度門檻值和最小信賴度門檻值，以減少擬定錯誤決策的機率。

由於不同資料集合之間的資料特徵均不相同，故較難以直接(Straightforward)的方式來立即的產生代表性高的門檻值；因此，我們初步認為可以將歷史資料做為訓練資料(Training Data)以建立選擇模式(Selection Model)；再利用已建立的模式為新增交易資料產生門檻值。由於採用歷史資料做為訓練資料，故建立的模式對於該企業而言具有相當的代表性，因此，若應用於建立新增交易資料的門檻值，其誤差程度應該不大，故所產生的門檻值亦必定具有一定的準確率。如此一來，便能夠幫助決策者快速且正確的選擇最小支持度門檻值和最小信賴度門檻值。值得一提的是，在建立選擇模式時，我們可以嘗試選擇許多不同的相關分類技術來做為建立模式之用，如：類神經網路、基因演算法和貝氏法則等。

至於利用此種訓練方法來建立門檻值選擇模式是否可行，則需進過進一步的研究、驗証和實驗，故未來希望能夠針對此問題做深入的研究，期望能夠找出有效解決此問題的方法。

參考文獻

[1] A. Tabatabai, “Edge Location and Data Compression for Digital Imagery, ” Ph.D. dissertation, School of Elect. Engrg., Purdue University, Dec. 1981.

[2] C. C. Aggarwal and P. S. Yu, “A New Approach to Online Generation of Association Rules, ” IEEE Trans. On Knowledge and Data Engineering, Vol. 13, No. 4, pp. 527-540, 2001.

[3] C. C. Aggarwal, C. Procopiuc, and P. S. Yu, “Find Localized Associations in Market Basket Data, ” IEEE Trans. On Knowledge and Data Engineering, Vol. 14, No. 1, pp. 51-62, 2002.

[4] D. I. Lin and Z. M. Kedem, “Pincer-Search: An Efficient Algorithm for Discovering the Maximum Frequent Set, “ IEEE Trans. on Knowledge and Data Engineering, Vol. 14, No. 3, pp. 553-556, 2002.

[5] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang, “Finding Interesting Associations without Support Pruning, “ IEEE Trans. On Knowledge and Data Engineering, Vol. 13, No. 1, pp. 64-78, 2001.

[6] F. Berzal and J. C. Cubero, “TBAR: An efficient method for association rule mining in relational databases,” Data and Knowledge Engineering, Vol. 37, No. 1, pp. 47-64, 2001.

[7] G. Szego, “Orthogonal Polynomials, ” Vol. 23, 4th ed., Amer. Math. Soc., Providence R. I., 1975.

[8] H. Toivonen, “Sampling Large Databases for Association Rules, “ Proc. Int’l Conf. Very Large Data Bases, pp. 134-145, 1996.

[9] J. Han, Y. Fu, “Mining Multiple-Level Association Rules in Large Databases, “ IEEE Trans. on Knowledge and Data Engineering, Vol. 11, No. 5, pp. 798-805, 1999.

[10] J. R. Quilan, “C4.5: Programs for Machine Learning, ” Morgan Kaufmann, 1993.

[11] J. R. Quilan, “Induction of decision trees, ” Machine Learning, pp. 81-106, 1986.

[12] J. S. Park, M. S. Chen, and P. S. Yu, “An Effective Hash-Based Algorithm for Mining Association Rules, “ Proc. ACM SIGMOD Int’ l Conf. Management of Data, pp. 175-186, 1995.

[13] L. Breiman, J. Friedman, R. Olshen, and C. Stone, “Classification of Regression Trees, ” Wadsworth, 1984.

[14] R. Agrawal, R. Srikant, “Fast Algorithms for Mining Association Rules, ” Proc. Int’ l Conf. Very Large Data Bases, pp. 487-499, 1994.

[15] R.T. Ng and J. Han, “Efficient and Effective Clustering Methods for Spatial Data Mining,” Proc. Int’l Conf. Very Large Data Bases, pp. 144-155, 1994.

[16] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: An Efficient Data Clustering Method for Large Databases, ” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 103-114, 1996

[17] V. Ganti, J. Gehrke, and R. Ramakrishnan, “Mining Very Large Databases, “ IEEE Computer Society, Vol. 32, No. 8, pp. 38-45, 1999.

[18] V. Ganti, R. Ramakrishnan, and J. Gehrke, “Clustering Large Datasets in Arbitrary Metric Spaces,” Proc. Int’l Conf. Data Engineering, pp. 502-511, 1999.

[19] W. H. Tsai, “Moment-preserving thresholding: A New Approach, ” Computer

Vision, Graphics, and Image Processing, Vol. 29, 377-393, 1985.

[20] W. Mendenhall, L. Ott, and R. L. Scheaffer, “Elementary Survey Sampling, ” Wadsworth, 1986.