電子商務研究，2004 年秋季

第二卷，第三期，頁 333-352

快速模組拆解之關聯規則探勘-QMD

黃仁鵬

南台科技大學

黃南傑

南台科技大學

郭煌政

國立嘉義大學

許耀文

南台科技大學

摘要

近年來，客戶關係管理(CRM)是個相當熱門的議題，因為企業必須了解消費者購物行為與商品間的關聯關係，才能妥善安排商品陳列順序。如此可以提昇客戶滿意度，減少購物的搜尋時間。再者可以刺激購買商品數量，用以增加企業的利潤。所以在大型交易資料庫中，利用資料探勘技術找出有用的關聯法則，來提供企業的決策支援是非常重要的。本研究提出新的演算法 QMD (Quick Modulized Decomposition)來找出商品間的關聯法則。QMD 演算法的優點如下：1.只需掃描資料庫一次；2.利用模組化方式來提昇執行效率；3.利用遮罩(mask)與布林模式(Boolean)來產生拆解項目因子模組。上述得知，透過本演算法做關聯分析，其效能將優於以往 Apriori-Base 的演算法。此外，關聯法則的推導過程中，將不會重複產生多餘的候選項目組，因此更勝於拆解模式的演算法。快速得到正確、有效用的資訊，是企業在數位時代中最大的利器，由此能降低時間成本、快速反映市場需求，是提昇競爭力的最大利基。

關鍵字：資料探勘、關聯法則、Apriori 演算法、高頻項目集

QMD Algorithm for Mining Association Rules

Jen-Peng Huang
Southern Taiwan University of Technology
Nan-Jie Huang
Southern Taiwan University of Technology
Huang-Cheng Kuo
National Chiayi University
Yao-Wu Hsu
Southern Taiwan University of Technology

Abstract

Recently Customer Relationship Management is one of the hottest issues in cooperations. In order to properly arrange the positions of products, Cooperations need to understand customers' shopping behaviors and the associationships between products. In this way, we can increase the customers' satisfication and decrease the searching time during shopping. Besides, we can increase the quantity of purchase products and the profits. Thus, it is very important to use the technology of data mining to find the useful association rules and to provide the cooperation's decision supports.In this paper we propose a new algorithm QMD (Quick Modulized Decomposition) to find the association rules from large transaction databases. The merits of QMD algorithm are: 1. In data mining process it only needs to scan whole transaction database once. 2. Using Modulized method to increase the performance of data mining process. 3. Using mask and Boolean method to decompose the itemsets to sub-itemsets. 4. In decompostion process, we combine the same sub-itemsets and get the supports of each sub-itemset very efficiently and significantly shorten the processing time and cost.

Keywords： Data Mining、Association Rule、Apriori、Frequent itemsets

壹、前言

近年來電腦等高科技產業快速成長，進而加速資訊化的過程。使得資料庫(Database)中的資料呈現倍數性成長，對企業決策者而言，要從大量資料中找出有用的隱含性資料是件非常困難的事。然而這些隱藏在資料庫中有用的資訊對於企業管理方面或決策支援方面等都有莫大的幫助。因此資料庫內的知識探索(Knowledge discovery)議題也隨之

興起，資料探勘(Data Mining)的技術，更是重要的一環。

資料探勘的應用相當廣泛，例如在零售業中，我們可以藉由分析所有顧客的交易記錄來分析出顧客購買東西的習慣，然後將顧客經常一起購買的商品擺設在一起，以增加營業額。另外顧客關係管理（CRM）也是一種相當好的應用，可藉由分析顧客行為的資料，找出顧客的喜好、厭惡等資訊，作為高層決策人員在做決策時的參考。

資料探勘是指在大量資料儲存體中，探勘出隱藏、從未發現、且有用資訊的技術，所分析出具有價值的資訊提供決策者作有效的決策。Chen 等人(1997)指出資料探勘是一種有效率的方法與整合技術，可以從資料中找出先前不知道，卻隱含於其中的有用資訊。而 Cabena 等人(1997)指出資料探勘為將潛在或原本不知道的資訊從大型資料庫萃取出來的過程，並將萃取出來的資訊提供主管做決定性的決策。以上為各學者對資料探勘所下的定義，而如何有效的利用資料探勘技術從大量資料中找出其中所隱含的資訊為目前各研究學者努力的目標。

在本研究第二節中，介紹關聯法則的相關文獻。在第三節中，提出新的演算法－ QMD 演算法來改進 Apriori 演算法的缺點，輔以實例說明運作過程，並比較兩個演算法的執行效率。最後，在第四節中對本研究做結論。

貳、文獻探討

在眾多的資料探勘技術中，針對不同的應用領域、資料庫型態，目前已有許多不同的技術相繼被提出，每一種技術都有其特性及應用，其中主要的技術種類如下：描述與辨別(Characterization and Discrimination)、關聯分析(Association Analysis)、預測與分類(Classification and Prediction)、叢集分析(Cluster Analysis)、異常分析(Outlier Analysis)、趨勢分析(Evolution Analysis)等熱門技術。

Han 等人(2000)指出,關聯法則(Association Rules)是目前在資料探勘中最為成熟和利用最多的一個領域。關聯法則最早由 Agrawal 等人於 1993 年所提出，主要是被用來尋找資料庫中項目之間的關聯性，Brin 等人(1997)指出關聯法則最初被用於分析市場購物籃資料(Market Basket Data)的研究，藉由分析顧客之購買行為，找出相關商品集間彼此的關聯性，提供給決策者做為商品擺設、進貨、儲貨的參考，並有助於提昇商品的競爭力，以增進商品銷售週轉率提昇利潤。例如：『＂顧客可能在購買牛奶之後，會接著再買麵包＂，所以應該將牛奶類商品的陳設台靠近麵包類的陳設台，對於這樣的資訊就稱為＂關聯法則＂，表達方式為：牛奶→ 麵包[minsup=2%，minconf = 80%]』。在關聯法則分析中有兩個重要的參數，也就是支持度(support)跟信賴度(confidence)，這兩個參數是用來評估所找出的關聯法則是否能滿足使用者的期望。而常見求取關聯法則的演算法有 Apriori 演算法、DHP、AprioriTid、AprioriHybrid、Boolean、FP-Tree、ICI、AIM…等。

以下將介紹關聯法則的主要定義、Apriori 演算法的運作原理、Apriori 演算法的主要瓶頸與 DIC 演算法。

一、關聯法則探討

最早由 Agrawal 等人提出於 1993 年提出,主要是在大型交易資料庫中擷取項目(Item)之間的關聯性。關聯法則主要的問題定義如下:令 $I=\{i_1,i_2,\ldots,i_m\}$,I 為所有商品項目(Items)的集合,D 為資料庫中所有交易記錄的集合,T 為每筆交易記錄項目的集合,T 為 I 的子集合($T \subseteq I$),而 T 中的交易記錄內容是不考慮商品項目購買的數量,TID 為每一筆交易記錄的編號。

關聯法則以 $X \rightarrow Y$ 表示,其中 $X \subset I$,$Y \subset I$ 且 $X \cap Y = \varnothing$。然而,一個關聯法則衡量的規則是什麼?如何才能讓一個關聯法則成立?衡量關聯法則的標準有二,一是支持度(support)、二是信賴度(confidence),以下將詳細介紹這兩種衡量標準。

(1)支持度(Support;s):支持度的定義為在 D 中包含 X 且包含 Y 交易記錄個數和 D 中所有交易記錄個數的比例,公式如(1)所示。

$$s = \frac{\text{資料項目X在資料庫D中出現的次數}(X_n)}{\text{資料庫D的總筆數}(D_m)}, (0 < s \le 1) \qquad (1)$$

(2)信賴度(Confidence;c):信賴度的定義為在 D 中包含 X 的交易記錄中也包含 Y 交易記錄所佔的比例,公式如(2)所示。

$$c(X \rightarrow Y) = \frac{\text{資料項目X與Y同時在資料庫D中出現的次出}(XY_n)}{\text{資料項目X在資料庫D中出現的次數}(X_n)}, (0 < c \le 1) \qquad (2)$$

其探勘關聯法則的工作主要可分為二個階段,(一)找出資料庫所有的高頻項目集(Frequent itemsets),也就是找出所有滿足最小支持度的項目組,若一個項目組含有 k 個項目,則稱為 k-項目組(k-itemsets),若 k-項目組滿足最小支持度,則稱為 k-高頻項目集;(二)根據第一階段產生的高頻項目集產生關聯法則;例如 BCE 為 3-高頻項目組,且 B、C、E⊆I,若關聯法則 BC→ E 滿足最小信賴度,表示此關聯法則成立。

二、Apriori 演算法

Apriori 演算法由 Agrawal 等人(1994)所提出,此一演算法是最具代表性的關聯法則演算法之一,而許多推導關聯法則技術的相關演算法,都是以 Apriori 為基礎加以改良或延伸,目前改進的方法有 AprioriTid、AprioriHybrid、Boolean、Partition、DIC、Cloum-Wise Apriori、Multiple-Leve…等。Apriori 演算法主要包含以下步驟:

(1)利用(k-1)-高頻項目集(L_{k-1})來產生候選項目集合(C_k)。

(2)掃描資料庫 D,計算所有候選項目集合的支持度,將所有支持

度大於等於最小支持度的候選項目集合選出來成為長度為 K 的高頻項目集(L_k)。

(3)重複上面(1)(2)步驟，直到無法再產生新的候選項目集合為止。

候選項目合併(Join)與修剪(Pruning)規則：

(1)由上述的步驟(1)找出有兩個 k-2 項目相同的(k-1)-高頻項目集，組合成 k-項目組。

(2)判斷(1)步驟中的 k-項目組，其所有的(k-1)-項目組之子集合是否都出現，假如成立則保留此 k-項目組。

三、Apriori 演算法的兩個瓶頸

(1)產生大量的候選項目集(Itemset)

在產在 2-候選項目時是由 1-頻繁項目兩兩合併產生，若 1-頻繁項目集中有 k 個項目，共會產生(k-1)+(k-2)+…+1 個 2-候選項目，即 *k*(k-1)/2* 個。假設 1-頻繁項目集中有 1000 個項目，則產生 45 萬個 2-候選項目。

(2)需要多次掃瞄資料庫

由⑴結果可知，因爲有大量的候選項目，而且每一個項目都必須掃瞄整個資料庫求取其支持度(support)，造成整體執行效率不佳。所以本研究的目的在於改善，產生頻繁項目集時所需花費的時間。

四、Dynamic Itemset Count(DIC 演算法)

在 Apriori 演算法中，候選項目集都是逐層產生的，而每一層所產生的候選項目集都必須搜尋完整個資料庫才能濾出該層的高頻項目集，所以每一個候選項目集必須花費大量的時間在資料庫的搜尋上，Brin 等人(1997)提出 DIC 演算法，其以 Apriori 演算法為基礎加以改進，主要目的就是要減少資料庫的存取時間。

DIC 演算法與 Apriori 演算法最大的不同在於其先將資料庫分為多個區塊，每個區塊中包含指定數量的交易記錄，搜尋資料庫時是以區塊為單位，在搜尋的過程中，在處理完成每一個區塊後，就馬上將每個區塊中所產生的潛在高頻項目集做結合，並且計算其支持度，若其支持度已經大於或等於使用者所訂定的最小支持度，就可以在不必等到搜尋完整個資料庫就濾出該層的高頻項目集，如此可以大幅減少搜尋資料庫的次數以提昇整體的效能。

DIC 演算法事先將資料庫分為多個區塊，就可以在不對資料庫掃描那麼多次的情況下，完成整個探勘的動作，而 Apriori 演算法卻必須要等到搜尋完整個資料庫才能濾出高頻項目集；在最差的情況下，DIC 演算法搜尋資料庫的次數頂多與 Apriori 演算法所需的次數相同。

利用 DIC 演算法進行探勘，能否有效的減少資料庫的掃描次數，

與分割資料庫的區塊數量有關，若區塊數量太少則仍須掃描資料庫多次，若區塊數量太多則誤判候選項目集為高頻項目集的可能性就會提高，反而會增加調整的時間。

參、研究方法

本研究針對 Apriori 演算法的缺點做改進,因為在每一個產生頻繁項目的階段都必須掃描資料庫 N 次，相當的費時，也造成整體執行效率不佳。所以本研究的目的在於改善找尋關聯法則時所需花費的時間。在本章節中詳細介紹整個 QMD 演算法的執行流程。一、將演算法運作過程繪製成流程圖,並依據流程圖詳細說明整個流程步驟。二、細說 QMD 之演算法。三、說明模組內部的拆解、配對、組合的演進過程。四、依照流程步驟透過實例來說明。五、與 Apriori 演算法執行效率比較。

一、演算法流程執行步驟

1.將資料庫交易記錄與 MAP 模組載入。

2.讀取交易記錄;並判斷交易記錄長度是否有合適的 MAP 模組,"是"則跳至第四步驟，"否"則執行第三步驟。

3.利用特殊組合方式(遮罩與布林 AND 運算)產生符合此交易記錄的 N 項目因素(itemsets)，並將此新模組加入現有 MAP 模組中。

4.依據 MAP 模組使用遞迴結構遞迴產生所有的因素項目組合(itemsets)。

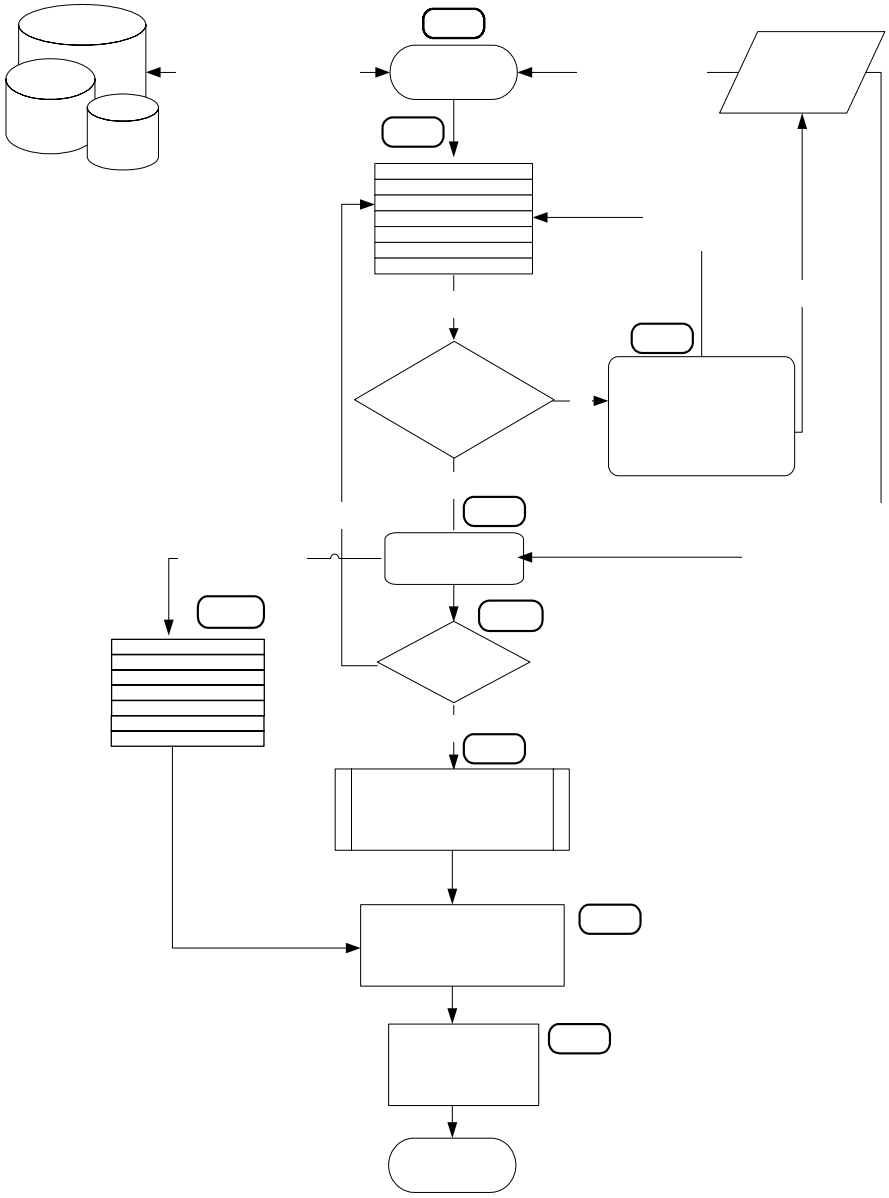
5.將步驟四產生的項目組合，存入因素項目表中；若此項目不存在表中，則將此項目加入，並給於初值 1；若此項目已存在表中，則將此項目值加 1。

6.判斷是否還有交易記錄，有則跳回第二步驟；否則執行第七步驟。

7.等待使用者輸入最小支持度與最小信賴度。

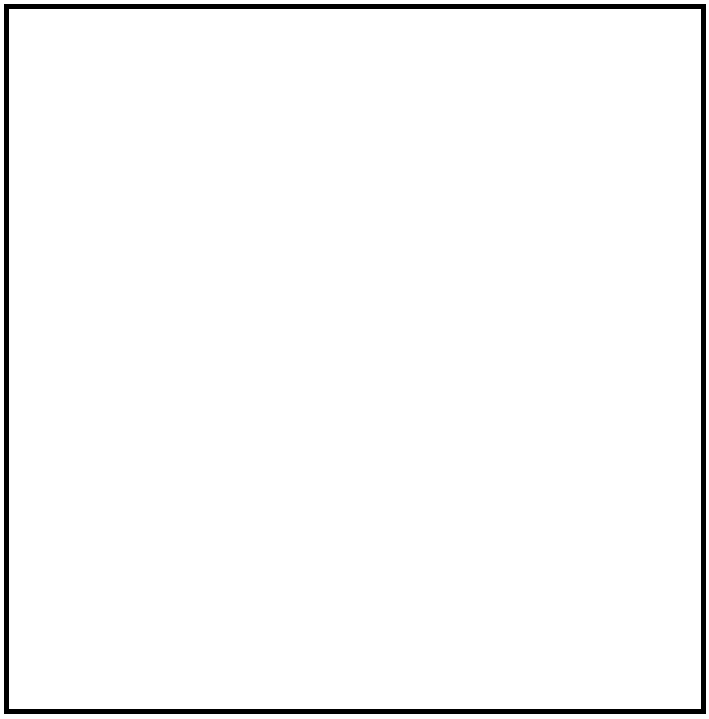
8.產生符合條件的高頻項目集合。

9.產生符合條件的關聯法則。



圖一　QMD 演算法流程圖

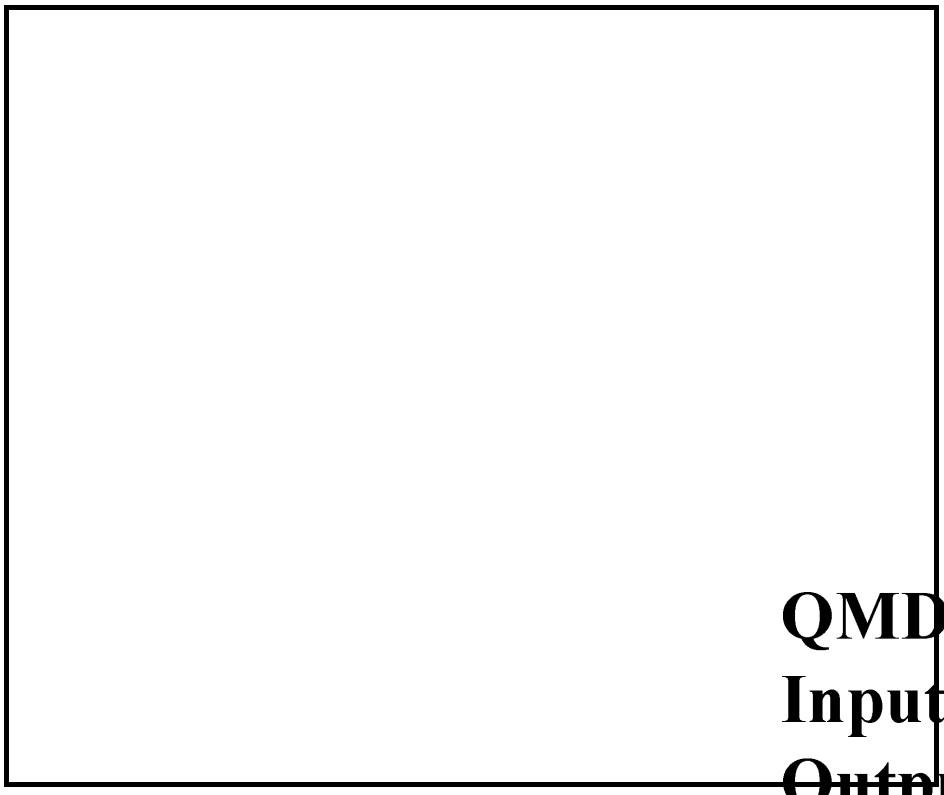
二、QMD 演算法說明



圖二　QMD 演算法

演算法程式說明：如圖二所示。(行 1)建立字母對照表，內容預設 52 個英文字母從大寫到小寫(A,B,…,Z,a,…,z)以供對映用，例如：交易記錄長度為一(X=1)會擷取到 A 代碼、長度為三(X=3)會擷取到 C 代碼。(行 2～行 3)設定初值，MM 為存放模組資訊用，初值為 null；FBT 所存放的是終值的二元樹，初值為 null。(行 4)開啟模組檔案，將檔案內容依序存放至 MM 中，以供對映用或新增模組用。(行 5～行 23)QMD 演算法主程式，資料庫內的交易記錄有幾筆，等於此區塊被執行的次數。(行 6)擷取讀入的交易記錄長度。(行 7～行 14)判斷是否有合適的 MAP 模組，"True"執行(行 8、9)由符合該交易記錄長度(n)的模組遞迴至長度一的模組，以取得 MAP 模組的資料內容，再將取得的資料內容取代為交易記錄的內容，"False"執行(行 11～行 13)先利用 produceNewModel 函式產生所需的新模組，其中 MM.max 為目前模組內最大的長度，n 為目前交易記錄所需的長度。在新模組產生後，再由新模組遞迴至長度一的模組，以取得 MAP 模組的資料內容，再將取得的資料內容取代為交易記錄的內容。(行 15～行 22)將對映後的項目集組合，存到 FBT 中計數，若此項目集已存在 FBT 中，則將此項目集之值加 1；若不存在，則將此項目集加入 FBT 中，並給予初值 1。重覆執行(行 5～行 23)直到沒有任何交易記錄為止。

QMD_Algor
Input：CM
Output：All
行1 create C
行2 modelM
行3 Final_Bi
行4 open Mo
行5 **forall** tra
行6 n = cou
行7 **if** (MM
行8 Mode
行9 items
行10 **else**
行11 **prod**
行12 Mode
行13 items
行14 **endif**



圖三　QMD 模組產生器演算法

如圖三所示。produceNewModel(MM.max,n)此函式即為所述的模組產生器。(行 25)讀取目前 MAP 模組中最大長度裡資料內容長度最長的項目，除了加入新模組用，也做為組合的依據。(行 26)取得字母對照表的字母，做為組合用。(行 27)透過組合方式產生新模組長度最長之項目(由上一個模組長度最長之項目與字母對照表的字母組合來)。(行 28)計算新模組最長項目之長度。(行 29～行 44)判斷新模組最長項目之長度是否大於等於 3，”True”執行(行 30～行 41) 使用特殊拆解方式來拆解其餘之子項目，”False”執行(行 43)直接將新模組最長之項目，再加上字母本身，成為新模組的資料內容。(行 30～行 33)使用遮罩方式來產生子項目。(行 34～行 41)判斷由遮罩所產生之子項目長度是否等於 2，”True”執行(行 35)將新模組最長之項目，加上字母本身，再加上遮罩所產生之子項目，成為新模組的資料內容。”False”執行(行 37～行 40)將遮罩之子項目彼此相互使用 Boolean 運算，直到產生所有長度二之項目。(行 40) 將新模組最長之項目，加上字母本身，再加上遮罩所產生之子項目與 Boolean 運算來之子項目，成為新模組的資料內容。(行 45)將產生的新模組加入 MM 中。

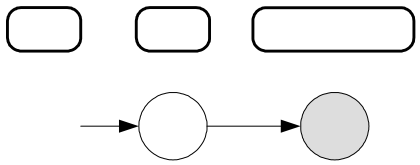
三、QMD 模組產生方式實例說明

此節詳細介紹 QMD 模組的拆解、配對、組合的演進過程，並舉一例子說明模組與交易記錄對映方式之運作過程。

(一)、模組組合產生方式

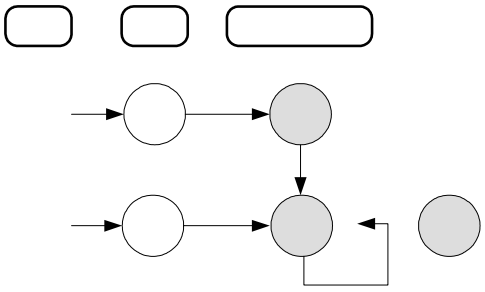
假設目前模組產生器，欲產生長度為一(X=1)的模組資料內容，此時 MAP 模組內並沒有任何模組，而模組產生器會依據長度來判斷，要從字母對照表中讀出何種字元做組合。

此例是欲產生長度為一的模組資料內容，所以從字母對照表中讀入 A 字母當做組合項目集的因子(所謂的字母對照表是指大寫字母「A」到小寫字母「z」的 52 個字母，當 Itemsets 的長度為一時，會讀取到大寫字母 A，當 Itemsets 的長度為 27 則會讀取到小寫字母 a，以此類推)，來產生模組的資料內容。讀 A 字母後，模組產生器隨即產生長度為一的 MAP 模組資料內容。此時，代號 A 只能產生一個項目集，其內容即為 A，如圖四所示。



圖四　MAP 模組產生圖(A)

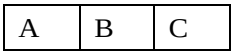
而交易記錄長度為二(X=2)，假設代號為 AB，可利用上一個 X=1 的內容來間接產生 AB 的項目組合內容，如圖五所示。【說明：代號 AB 比上一個 MAP 模組的代號多增加了一個項目 B，作法只需要將項目 B 與上一個 MAP 模組最長的資料內容結合(如：上一模組資料內容 A 與本身 B 結合成 AB)，最後將自己本身 B 加入，而此模組的資料內容第一個項目為長度最長之項目，如此以方便下一個模組抓取上一模組長度最長之項目，即完成組合這個動作。】



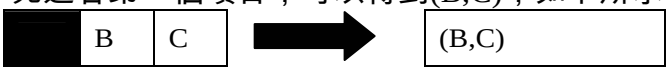
圖五　MAP 模組產生圖(AB)

而交易記錄長度為三(X=3)，假設代號為 ABC，可利用上一個 X=2 的內容來間接產生 ABC 的項目組合內容，如圖六所示。【說明：代號 ABC 比上一個 MAP 模組的代號多增加了一個項目 C，作法只需要將項目 C 與上一個 MAP 模組最長的資料內容結合(如：上一模組資料內容 AB 與本身 C 結合成 ABC)，並將自己本身 C 加入，由於此模組的長度超過三，所以必須要使用特殊之拆解方式(說明如下)來拆解出其他子項目集，將新產生項目 ABC 使用遮罩方式，產生 BC 與 AC，即完成組合這個動作。】

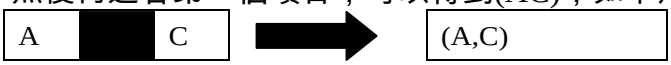
特殊之拆解方式說明：拆解長度三(A,B,C)的子項目



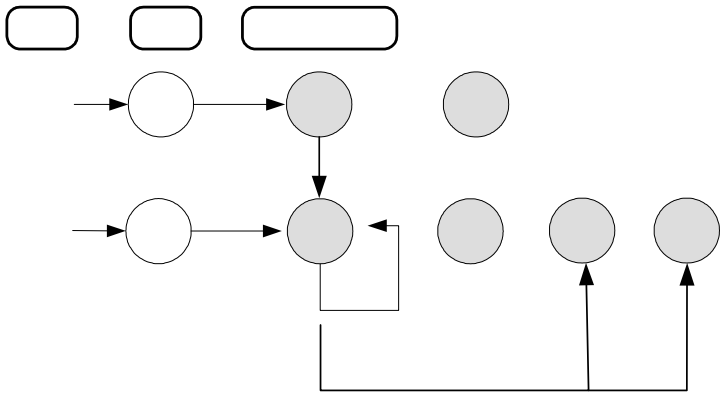
先遮著第一個項目，可以得到(B,C)，如下所示：



然後再遮著第二個項目，可以得到(AC)，如下所示：



Mark 的結束條件為 N-1 次，N 為項目長度。例如要拆解的為(A,B,C)長度三的項目所以 N 就為 3，遮罩的次數就是 2 次，並判斷所拆解出項目長度為二，所以就停止往下拆解。



圖六　MAP 模組產生圖(ABC)

而交易記錄長度為四(X=4)，假設代號為 ABCD，可利用上一個 X=3 的內容來間接產生 ABCD 的項目組合內容，如圖七所示。【說明：代號 ABCD 比上一個 MAP 模組的代號多增加了一個項目 D，作法只需要將項目 D 與上一個 MAP 模組最長的資料內容結合(如：上一模組資料內容 ABC 與本身 D 結合成 ABCD)，並將自己本身 D 加入，由於此模組的長度超過三，所以必須要使用特殊之拆解方式來拆解出其他

子項目集，將新產生項目 ABCD 使用遮罩方式與布林 AND 運算，產生 BCD、ACD、ABD、CD、BD、AD，即完成組合這個動作。】

特殊之拆解方式說明：拆解長度四(A,B,C,D)的子項目

步驟一：使用上述遮罩方式產生 BCD、ACD、ABD(因為使用遮罩所拆解出之子項目長度大於二，所以必須執行步驟二，以求出所有長度二之項目)。

步驟二：利用步驟一所產生之項目相互做布林 AND 運算，直到 AND 出所有長度二項目，說明如下。

(B,C,D) AND (A,C,D)，得到(C,D)

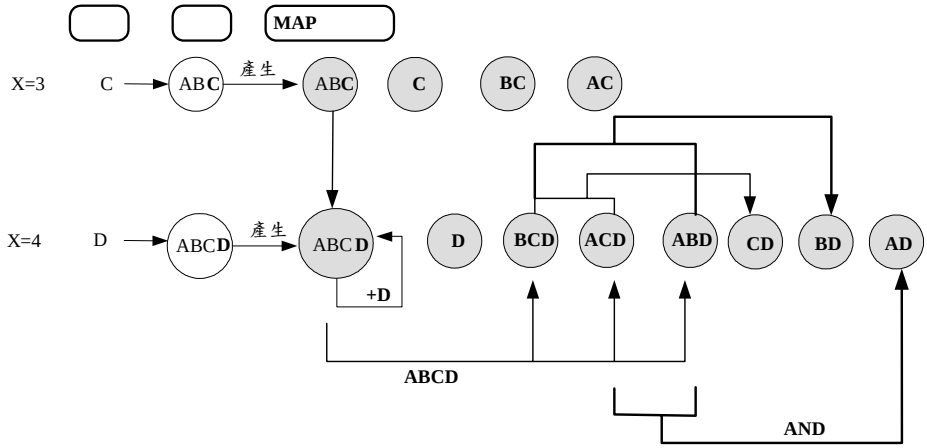
項目	A	B	C	D
B,C,D	0	1	1	1
A,C,D	1	0	1	1
	0	0	1	1

(B,C,D) AND (A,B,D)，得到(B,D)

項目	A	B	C	D
B,C,D	0	1	1	1
A,B,D	1	1	0	1
	0	1	0	1

(A,C,D) AND (A,B,D)，得到(A,D)

項目	A	B	C	D
A,C,D	1	0	1	1
A,B,D	1	1	0	1
	1	0	0	1

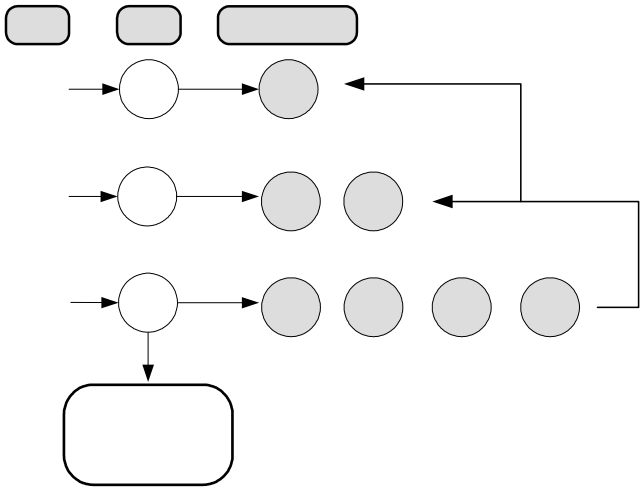


圖七　MAP 模組產生圖(ABCD)

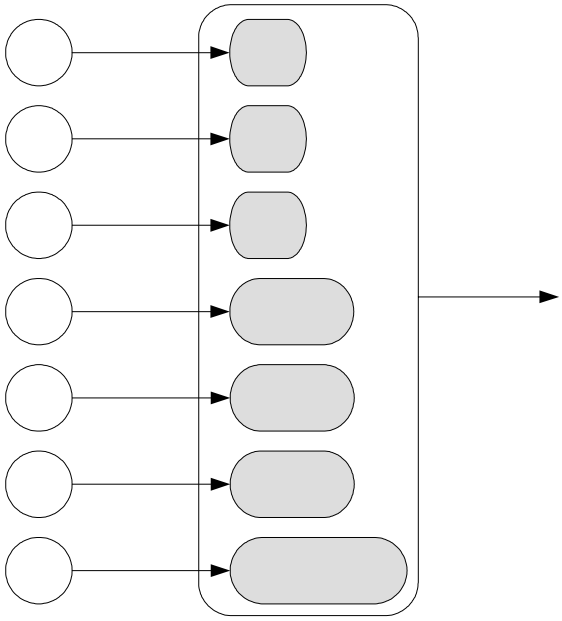
(二)、模組與交易記錄對映方式實例說明

假設一交易記錄為(P120,P220,P500)，數字表示為商品編號，其交易長度為三，所以對映到 X=3 的 MAP 模組內容，即 P120⇔A；

P220⇔B；P500⇔C，依據 MAP 模組的資料內容經由取代的方式加上遞迴的觀念，遞迴至長度一的模組，產生(P120,P220,P500)、(P500)、(P220,P500)、(P120,P500)、(P120,P220)、(P120)、(P220)如下圖的完整對映圖所示，圖八、圖九。(“⇔”表示『對映關係』)。



圖八　使用遞迴結構遞迴至長度一之模組



圖九　完整對映過程圖

四、完整實例說明

首先，步驟一，載入資料庫中的交易記錄以及 MAP 模組，表一為顧客購買商品項目的交易資料庫內容，TID 為顧客交易的編號，Itemsets 為顧客購買商品項目的交易記錄代碼細目；而表二為現有 MAP 模組的資料內容，在此假設模組內已有 3 個長度的模組資料內容。

表一　原始資料庫 D

TID	Itemsets
T001	P001,P003,P004
T002	P002,P003,P005
T003	P001,P002,P003,P005
T004	P002,P005

表二　MAP 模組

交易記錄長度	交易記錄代表內容	所有的交易項目集組合
X=1	A	A
X=2	AB	AB,B
X=3	ABC	ABC,C,BC,AC

進入步驟二，讀入第一筆交易記錄，T001(P001,P003,P004)，此筆交易記錄長度為三，隨即至 MAP 模組中尋找是否有合適的模組資料內容，判斷”是”即跳到步驟四，依據 MAP 模組的資料內容使用遞迴的方式，產生(P001, P003, P004)、(P004)、(P003,P004)、(P001,P004)、(P001, P003)、(P001)、(P003)等對映後的項目集組合。步驟五，將步驟四之結果存至因素項目集中，如表三所示。步驟六，判斷尚有記錄，跳回步驟二讀取下一筆交易記錄。

表三　因素項目表(讀入 T001 對映後項目集組合)

X=3		X=2		X=1	
Itemsets	Sup	Itemsets	Sup	Itemsets	Sup
P001,P003,P004	1	P001,P003	1	P001	1
		P001,P004	1	P003	1
		P003,P004	1	P004	1

步驟二～步驟六，讀入第二筆交易記錄，T002(P002,P003,P005)，拆解方式與上述相同，因素項目表更新後如表四。

表四　因素項目表(讀入 T002 對映後項目集組合)

X=3		X=2		X=1	
Itemsets	Sup	Itemsets	Sup	Itemsets	Sup
P001,P003,P004	1	P001,P003	1	P001	1
P002,P003,P005	1	P001,P004	1	P002	1
		P003,P004	1	P003	2
		P002,P003	1	P004	1
		P002,P005	1	P005	1
		P003,P005	1		

步驟二，再讀取下一筆交易記錄 T003(P001,P002,P003,P005)。判斷長度為四，因為 MAP 模組中沒有合適的長度，所以進入步驟三。利用模組產生器，產生此筆交易記錄所需的新模組。產生後將新產生的模組加入 MAP 模組中，如表五所示。

表五　新 MAP 模組

交易記錄長度	交易記錄代表內容	所有的交易項目集組合
X=1	A	A
X=2	AB	AB,B
X=3	ABC	ABC,C,BC,AC
X=4	ABCD	ABCD,D,BCD,ACD,ABD,CD,BD,AD

取得新模組後進入步驟四，依據新 MAP 模組的資料內容使用遞迴之方式，產生(P001,P002,P003,P005)、(P005)、(P002,P003,P005)、(P001,P003,P005)、(P001,P002,P005)、(P003,P005)、(P002,P005)、(P001,P005)、(P001,P002,P003)、(P003)、(P002,P003)、(P001,P003)、(P001,P002)、(P002)、(P001) 等對映後的項目集組合。步驟五，將結果存至因素項目集中，如表六所示。步驟六，判斷尚有記錄，跳回第二步驟，讀取下一筆交易記錄。

表六　因素項目表(讀入 T003 對映後項目集組合)

X=4		X=3		X=2		X=1	
Itemsets	Sup	Itemsets	Sup	Itemsets	Sup	Itemsets	Sup
P001,P002, P003,P005	1	P001,P002,P003	1	P001,P002	1	P001	2
		P001,P002,P005	1	P001,P003	2	P002	2
		P001,P003,P004	1	P001,P004	1	P003	3
		P001,P003,P005	1	P001,P005	1	P004	1
		P002,P003,P005	2	P003,P004	1	P005	2
				P002,P003	2		
				P002,P005	2		
				P003,P005	2		

步驟二～步驟五，再讀取下一筆交易記錄 T004(P002,P005)，拆解方式與上述相同，因素項目表更新後如表七。

表七　因素項目表(讀入 T004 對映後項目集組合)

X=4		X=3		X=2		X=1	
Itemsets	Sup	Itemsets	Sup	Itemsets	Sup	Itemsets	Sup
P001,P002, P003,P005	1	P001,P002,P003	1	P001,P002	1	P001	2
		P001,P002,P005	1	P001,P003	2	P002	3
		P001,P003,P004	1	P001,P004	1	P003	3
		P001,P003,P005	1	P001,P005	1	P004	1
		P002,P003,P005	2	P003,P004	1	P005	3
				P002,P003	2		
				P002,P005	3		

P003,P005	2

步驟六，判斷已沒有任何交易記錄，進入步驟七等待使用者輸入最小支持度與最小信賴度。步驟七，在此假設使用者設定的最小支持度為 50%(最低交易次數為 50%*4=2)、信賴度為 80%。步驟八，由因素項目表中找出符合最小支持度的大項目集合，整理如下(表八)。

第三大項目集合(L3)：(P002,P003,P005)。第二大項目集合(L2)：(P001,P003)、(P002,P003)、(P002,P005)、(P003,P005)。第一大項目集合(L1)：(P001)、(P002)、(P003)、(P005)。

表八　大項目集合

X=3		X=2		X=1	
Itemsets	Support	Itemsets	Support	Itemsets	Support
P002,P003,P005	2	P001,P003	2	P001	2
		P002,P003	2	P002	3
		P002,P005	3	P003	3
		P003,P005	2	P005	3

根據大項目集合(L3、L2、L1)，產生下列關聯法則。

☑ P002,P003→ P005(100%>80%)

☑ P003,P005→ P002(100%>80%)

☑ P001→ P003(100%>80%)

☑ P002→ P005(100%>80%)

☑ P005→ P002(100%>80%)

求解關聯法則，由高頻項目集檢測是否通過最小信賴度，實例如下：

1.舉例：P002,P003→ P005<符合>

$$p(P005 \mid P002, P003) = \frac{p(P002, P003, P005)}{p(P002, P003)} = \frac{2}{2} = 100\% > 80\%$$

2.舉例：P002→ P003<不符合>

$$p(P003 \mid P002) = \frac{p(P002, P003)}{p(P002)} = \frac{2}{3} = 67\% < 80\%$$

五、演算法執行效率比較

為了驗證 QMD 演算法的執行效率，本研究分別設計了幾個實驗的資料庫，來檢測本研究提出的方法，並將實驗的結果與 Apriori 演算法做效率比較，再將實驗的結果做相關的評估與說明。本研究利用以下的環境來進行資料探勘效率的測試：

(一) 實驗環境說明

實驗平台

(1) CPU：PentiumⅣ 1.7GHz

(2) Memory：512Mbytes

(3) OS：Windows 2000 Sever

(4) Data Base：MS SQL Sever 2000

(5) Programming Language：Java JDK 1.4

實驗資料庫

本研究實驗為求公正，實驗資料庫是由 IBM generator 產生。產生測試資料庫參數如表九所示。

表九　定義參數

D	資料庫的總交易量(筆數)
L	資料庫中單筆交易記錄最大的項目數
I	產生的頻繁項目集平均最大的項目數
N	交易資料庫中項目的總數

本研究用來實驗資料庫有 3 個，資料庫筆數(D)介於 50K 至 100K，包含的項目個數為 500，交易記錄長度(L)最長均為 10 個項目，平均最大頻繁項目集之項目數(I)為 4，根據此命名原則，用來測試的 3 組資料庫如表十所示。

表十　實驗資料庫內容

Database Name	L	N	I	D
L10N500I4D100K	10	500	4	100K
L10N500I4D80K	10	500	4	80K
L10N500I4D50K	10	500	4	50K

實驗設計

實驗 1：支持度對執行效率的影響。

實驗目的：測試支持度的改變對 QMD 演算法與 Apriori 演算法執行效率的影響。

實驗方法：本研究測試 D100K 的實驗資料庫，而資料庫中交易記錄最長為 10、平均頻繁項目集為 4、總項目個數為 N500。實驗中分別以 1%至 0.1%等 5 個最小支持度進行測試，並記錄產生所有頻繁項目集所需的時間，比較 QMD 演算法與 Apriori 演算法在不同支持度下的執行效率，實驗結果之比較圖，分別以最小支持度與程式執行時間為橫軸與縱軸。

實驗 2：交易記錄數量對執行效率的影響。

實驗目的：測試交易記錄數量的改變對 QMD 演算法與 Apriori 演算法執行效率的影響。

實驗方法：本研究測試 D50K 至 D100K 的實驗資料庫，而資料庫中交易記錄最長為 10、平均頻繁項目集為 4、總項目個數為 N500。實驗中以最小支持度 1%進行測試，並記錄產生所有頻繁項目集所需的時間，比較 QMD 演算法與 Apriori 演算法在不同資料庫中是否依然都

能保持良好的效能，實驗結果之比較圖，分別以資料庫總筆數與程式執行時間為橫軸與縱軸。

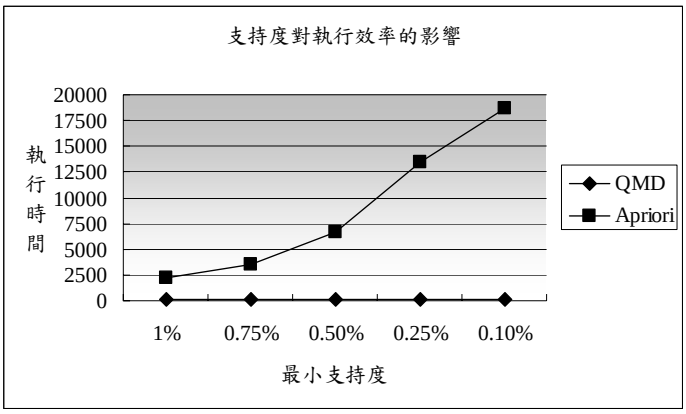
(二)數據效能評估

實驗 1：支持度對執行效率的影響。

依據資料庫 L10T7N500I4D100K，經過實驗，產生下圖的實驗結果：

表十一　QMD 與 APRIORI 演算法效率比較表

Minsup	1%	0.75%	0.50%	0.25%	0.10%
QMD	94sec	94sec	94sec	94sec	94sec
Apriori	2234sec	3512sec	6731sec	13528sec	18644sec
效率比	23.77	37.36	71.61	143.91	198.34



圖十　QMD 與 APRIORI 演算法效率比較圖

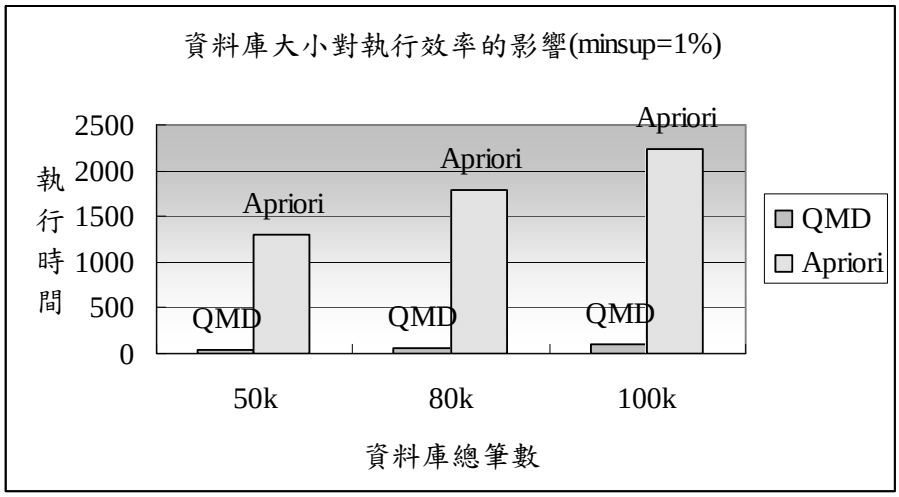
由圖十與表十一可知本研究 QMD 演算法執行效率平穩，不會因為支持度的不同而影響執行效率，而 Apriori 演算法很明顯的可以看出會因為支持度的不同而影響執行效率，由上結果可知，QMD 演算法執行速度平均較 Apriori 演算法快，且平均快約 95 倍。

實驗 2：交易記錄數量對執行效率的影響。

依據資料庫 D50K 至 D100K，經過實驗，產生下圖的實驗結果：

表十二　QMD 與 APRIORI 演算法效率比較表

資料庫大小	50k	80k	100k
QMD	40sec	61sec	94sec
Apriori	1301sec	1797sec	2234sec
效率比	32.52	29.45	23.76



圖十一　QMD 與 APRIORI 演算法效率比較圖

由圖十一與表十二可知隨著資料庫大小的增加，各演算法所花費的執行時間也明顯的變多，但是不論資料庫多大，由上結果可知，QMD 演算法執行速度平均較 Apriori 演算法快，且平均快約 29 倍。

綜合以上兩組的實驗結果可知 QMD 演算法執行效率相當優異，不論在支持度大小的實驗或是資料庫大小，皆比 Apriori 演算法來的有效率，也驗證了 QMD 演算法的優越執行效率。

肆、結論

一般用來作資料探勘的資料庫都非常龐大，當進行資料探勘時需要攏長的時間掃描資料庫，尤其是利用以 Apriori-Base 的演算法，來做關聯分析時其效率是經常被詬病。由於 Apriori 最大的缺點就於執行效率不佳，也就是產生每個候選項目時，會重複掃瞄資料庫來計算出候選項目的支持度，使得執行時間相當長。因此，針對此缺點，本研究提出 QMD 演算法改善，1.只需掃描資料庫一次，並利用模組方式來提昇執行效率；2.利用遮罩與布林運算來產生拆解項目因子模組。

透過本演算法做關聯分析，其效能將優於以往 Apriori-Base 的演算法。此外，關聯法則的推導過程中，將不會重複產生多餘的候選項目組，因此更勝於拆解模式的演算法，並快速得到正確、有效用的資訊。由此能降低時間成本、快速反映市場需求，是企業提昇競爭力的最大利基。

參考文獻

Agrawal R., Imilienski T., and Swami A., "Mining Association Rules between Sets of Items in Large Databases, "In Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, pp.207-216, May

1993.

Agrawal R.and Srikant R., "Fast Algorithm for Mining Association Rules in Large Databases," In Proc. 1994 In't Conf. VLDB, pp. 487-499, Santiago, Chile, Sep. 1994.

Brin S., Motwani R., Ullman J.D., and Tsur S., "Dynamic Itemset Counting and Implication Rules for Market Basket Data," ACM SIGMOD Conference on Management of Data, pp.255-264, 1997.

Brin S., Motwani R., and Silverstein C., "Beyond Market Baskets: Generalizing Association Rules to Correlations," 1997 ACM SIGMOD Conference on Management of Data, pp.265-276, 1997.

Cabena P., Hadjinian P., Stadler R., Verhees J. and Zanasi A., "Disovering Data Mining From Concept to Implementation, "Prentice-Hall Inc., 1997.

Chen M.S., Han J., and Yu P.S., "Data Mining: An Overview from a Database Perspective, "IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, 1996.

Han J. and Kamber M., "Data Mining: Concepts and Techniques, "Morgan Kaufmann Publishers, 2000.

作者簡介

黃仁鵬博士，畢業於美國奧克拉荷馬大學，現任教於南台科技大學資訊管理學研究所，主要研究領域為資料庫系統、專家系統、演算法、超級電腦分析、資料探勘，電子郵件信箱為 **jehuang@mail.stut.edu.tw**。

郭煌政博士，畢業於美國凱斯西儲大學，現任教於國立嘉義大學資訊工程學系研究所，主要研究領域為即時資料庫管理系統、資料探勘，電子郵件信箱為 **hckuo@mail.ncyu.edu.tw**。

黃南傑研究生，畢業於南台科技大學資訊管理所，主要研究領域為資料採勘，電子郵件信箱為 **m9190102@webmail.stut.edu.tw**。

許耀文研究生畢業於南台科技大學資訊管理所，主要研究領域為資料探勘，電子郵件信箱為 **m9190216@webmail.stut.edu.tw**。

「電子商務研究」徵稿簡則

(1)本學報為學術刊物，由國立台北大學資管所負責編輯出版，每年三月、六月、九月、十二月各出版一期，全年接受投稿。

(2)為提升稿件作業效率，本學報採取線上投稿與審稿方式，投稿稿件必須為 word 檔或 pdf 檔，網址為 http://ECstudies.thesis.com.tw，或由 http://www.mis.ntpu.edu.tw 進入。投稿後，作者即可利用自訂的密碼，隨時關心稿件處理現況。本學報並將盡力加速審稿作業，目標設定為百分之八十稿件在投稿六週內給予作者第一次審查結果回覆。

(3)本學報稿件至少經過兩位審查者匿名審查。

(4)來稿中英文不拘，以不超過二十頁為原則。中文稿件應附英文論文題目、摘要、關鍵字，英文稿件應附中文論文題目、摘要、關鍵字。作者姓名、職稱、服務單位等資訊請於網站上直接輸入，勿出現於文稿內。

(5)為維護學術倫理，來稿應未刊於其他刊物或書籍，且文責自付。稿件一經刊載，非經本學報同意，不得另行發表於其他刊物。稿件投稿後，視同授權本學報複製或儲存該論文，**稿件一經接受後，本學報擁有該稿件轉載與儲存於網站、光碟、資料庫、或發行於其他書籍、刊物、各類媒體及其他各項權利。**本學報接受曾在研討會發表之論文，但必須無著作權疑慮。為鼓勵學術知識分享，在合理範圍內與非營利前提下，作者可自行複製刊登於本學報之論文，基於學術研究之必要，讀者可複製論文以供學術研究或課堂討論之用。**論文接受後需將「著作權讓與同意書」(可於 http://ecstudies.thesis.com.tw 取得，自行列印簽名)寄至本所。**

(6)本學報為獨立公正之學術發表園地，為維持本學報之財務獨立自主性，本學報將於稿件接受後向作者酌收論文刊登費。本學報編輯委員由各大學電子商務領域資深研究者組成，目前總編輯為國立臺北大學方文昌教授。

(7)本學報以促進電子商務學術研究為目的，舉凡理論推演、實證驗證、觀念發展、個案研究、系統發展、技術研發方面的論文，無論針對電子商務的管理層面或技術層面，均為本學報刊登之對象。詳細徵稿主題請參考本學報之網站。

「電子商務研究」評審程序

1. 本學報稿件評審工作由電子商務相關研究領域中，具有博士學位或助理教授以上之學者擔任之。與本學報研究領域相同之學者可自願申請擔任審查委員，其資格審核由編輯委員決定之。

2. 稿件一律採取雙向匿名審查，每篇稿件至少由兩位審查委員進行評審，除了陳述審查意見外，並於下述四項審查建議中勾選一項：

   接受

   小幅修改

   大幅修改

   退稿

3. 針對審查意見之處理方式如下。

		評審二之審查建議			
		刊登	小幅修正	大幅修正	退稿
評審一之審查建議	刊登	刊登	刊登	修改後刊登	第三位評審
	小幅修正	刊登	刊登	修改後刊登	不同意刊登
	大幅修正	修改後刊登	修改後刊登	不同意刊登	不同意刊登
	退稿	第三位評審	不同意刊登	不同意刊登	不同意刊登

說明：審查結果若為修改後刊登，則必須修改至審查委員認同已具刊登水準方可刊登。

4. 稿件送交第三位評審後，取兩位對作者最有利之審查委員意見進行判斷，如落於「刊登」區，則刊登，如落於「修改後刊登」，則必須修改至審查委員認同已具刊登水準方可刊登，如落於「第三位評審」與「不同意刊登」區，則退稿。

5. 所有審查意見，均置放於本學報網站，供作者參考，本學報並將以電子郵件說明稿件處理方式。本學報對於稿件之刊登與否保有最後決定與調整權。

「電子商務研究」論文刊登費說明

一、為鼓勵本學報讀者投稿，本學報訂戶刊登之稿件，完稿頁數二十頁內免收刊登費。超過二十頁時，每額外增加一頁之刊登費為 500 元。

二、非本學報訂戶，完稿頁數二十頁以內收新台幣 7000 元，低於二十頁以 7000 元計算。刊登頁數超過二十頁時，每額外增加一頁之刊登費為 500 元。

三、刊登費收費標準得視狀況調整，每年調整次數至多不超過一次，於每年一月份時公佈。

四、刊登費收取目的，在於期使各稿件均能以較合適的篇幅長度發表。作者若確實認為其論文需較大篇幅方能表現，則可以支付刊登費的方式處理。

五、**請踴躍投稿，共同灌溉此份屬於全體電子商務研究者的學報。**

「電子商務研究」訂閱辦法

一、本學報於每年發行四期。

二、為簡化本學報作業，本學報不零售，訂閱以年度為單位：

訂閱 2004 年(預定出版四期)，費用為一千二百元

三、訂閱時，請以郵政劃撥方式，戶名：國立臺北大學，劃撥帳號：19456481 , 並請於劃撥單背面註明「訂閱電子商務研究」。劃撥後，請保留收據，將劃撥單上的局號、收據編號等資訊 e-mail 至 ecs@cm1.hinet.net，以通知本學報。

四、本學報鼓勵海外訂閱，海外訂閱與台灣地區訂閱同價，請先以電子郵件聯絡，再以等值外幣或其他方式付費。

五、**請踴躍訂閱，共同灌溉此份屬於全體電子商務研究者的學報。**