適合市場區隔應用的多維度關聯規則探勘技術之研究

李御璽
銘傳大學資訊工程學系
leeys@mcu.edu.tw

顏秀珍
銘傳大學資訊工程學系
sjyen@mcu.edu.tw

摘要

根據調查顯示，75%的企業在面臨擬定策略時，常常無法獲得即時且有根據的決策資訊。什麼樣的資料、要透過什麼樣的方法，才能快速且即時的轉變成決策時有用的資訊，是現代企業所面臨最迫切性的問題。資料探勘(Data Mining)無疑是解決這些問題最有效的途徑之一。完整的資料探勘不單可以做到準確的目標市場行銷(Target Marketing)，也可以做到大量的客製化(Customization)。有鑑於此，本研究提出一個適合市場區隔(Market Segmentation)應用的多維度關聯規則(Multidimensional Association Rules)探勘技術，利用條件式資料集(Conditional Dataset)挖掘出包含客戶特徵與客戶購買行為的多維度關聯規則。它不需要多次掃瞄目標資料集，並結合集群(Clustering)技術將數值資料自動離散化(Discretization)。本研究所提之方法將產生兩種不同角度的探勘結果：其一是在不同的客戶特徵下，發掘出經常被購買的產品組合；另一個則是在不同的產品組合下，發掘出經常購買該產品組合的客戶特徵。這兩種探勘結果可以提供決策者做市場區隔和制訂更為精確的行銷策略。

關鍵詞：集群、條件式資料集、資料探勘、市場區隔、多維度關聯規則

一、導論

現今的企業處在競爭非常激烈的環境。以往以產品導向(Product-Oriented) 為主的市場，如今轉成以客戶導向(Customer-Oriented)為主[5]。因此，發掘客戶特徵與客戶行為就變成企業在區隔客戶和制訂行銷策略時，非常重要的資訊來源。資料探勘(Data Mining)無疑是解決這些問題最有效的途徑之一[6]。資料探勘有很多不同的技術，其中關聯規則(Association Rules)探勘是最普遍應用的技術之一。它可以挖掘出交易資料中的產品間之關聯性，以幫助企業進行交叉銷售(Cross-Selling)、目標市場行銷(Target Marketing)等商業活動。

傳統關聯規則探勘的演算法有很多。例如，Apriori 演算法[1]、DHP 演算法[4]、FP-Growth 演算法[3]等。然而，這些研究都著重在探勘單一維度的關聯規則(Single Dimension Association Rules)。單一維度的關聯規則只有發掘在交易資料中所購買產品間之關聯性，這樣的資訊是不夠的。這樣形式的規則不能描述像是“年齡 20 歲到 30 歲且薪水兩萬到三萬的人，會買筆記型電腦的機率相當高”。而企業想要得到更多像是客戶特徵和客戶行為的資訊，去幫助他們管理客戶關係。因此，探勘多維度關聯規則(Multidimensional Association Rules)，對企業而言，相對是更加重要的。

有鑑於此，本研究提出一個適合市場區隔(Market Segmentation)應用的多維度關聯規則探勘技術，利用條件式資料集(Conditional Dataset)挖掘出包含客戶特徵與客戶購買行為的多維度關聯規則。它不

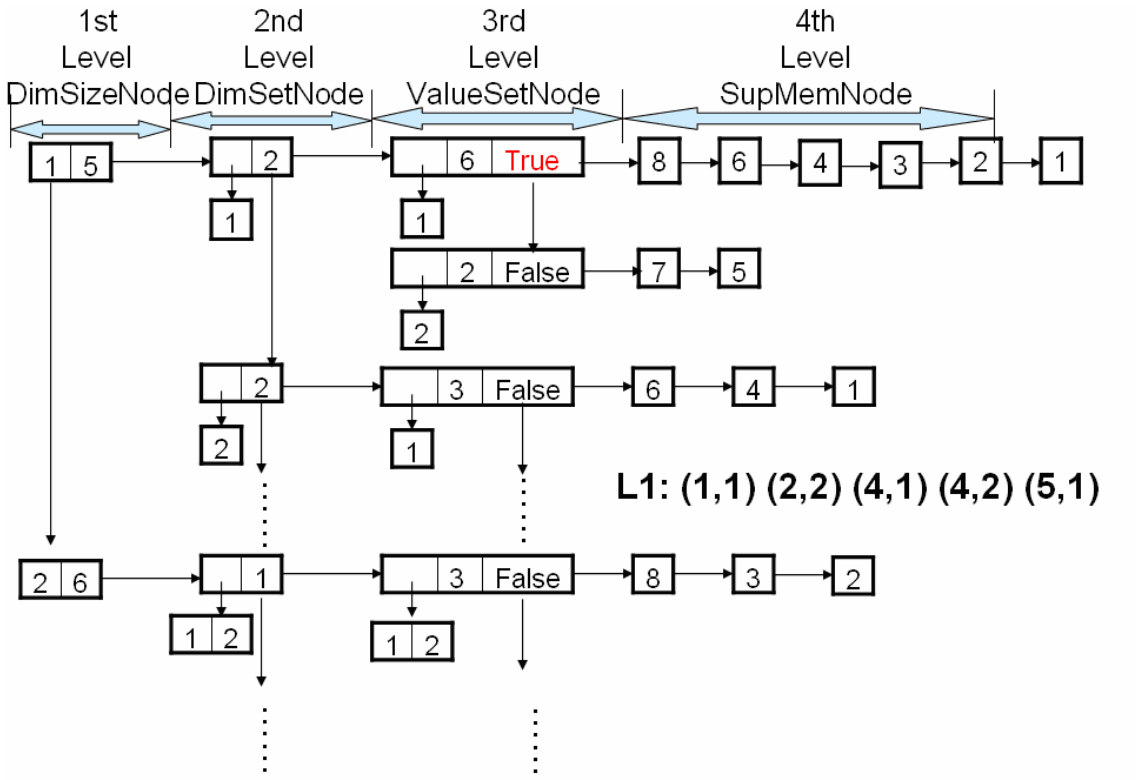

圖 1 MDIM 4 個 Level 的鏈結串列的架構

需要多次掃瞄目標資料集，並結合集群(Clustering)的技術將數值資料自動離散化(Discretization)。本研究所提之方法將產生兩種不同角度的探勘結果：其一是在不同的客戶特徵下，發掘出經常被購買的產品組合；另一個則是在不同的產品組合下，發掘出經常購買該產品組合的客戶特徵。這兩種探勘結果可以提供決策者做市場區隔和制訂更為精確的行銷策略。

本論文的章節安排如下。下一節我們將介紹相關的研究工作。第三節將介紹我們所提出探勘多維度關聯規則的方法。第四節則說明如何利用第三節所提出的方法，在不同的客戶特徵下，發掘出經常被購買的產品組合。第五節則說明如何利用第三節所提出的方法，在不同的產品組合之下，發掘出經常購買該產品的客戶特徵。在做結論之前，第六節將以實驗說明本研究所提之方法優於前人的研究。

二、相關研究

在傳統關聯規則探勘中，每一筆交易在交易資料集中只有包含購買產品的資訊。實際上，交易的資料(例如，信用卡的交易資料)往往包含更多的資訊，像是客戶性別、年齡、薪水和其他屬性等。當交易資料包含其它更多屬性(不僅有購買產品的屬性)時，我們稱為多維度的資料。而關聯規則中，包含兩個或是兩個以上的屬性(維度)，則可視為多維度關聯規則。

之前多維度關聯規則的相關研究大部分是將關聯式資料表轉變成像是交易資料表的形式，再利用 Apriori-Like 的演算法去探勘多維度關聯規則[2,7]。然而，這些方法基本上都需要使用者自行將數值屬性離散化，然後才能進行多維度關聯規則探勘的工作。不同於[2,7]，Xu 和 Wang [8]設計了 MDIM(Multi-Dimensional Indexing Mining)演算法，利用一個資料結構去儲存

表 1 多維度範例資料集

ID	Field 1	Field 2	Field 3	Field 4	Field 5
1	1	1	1	1	1
2	1	2	2	1	2
3	1	2	2	2	2
4	1	1	1	2	1
5	2	2	3	2	1
6	1	1	1	1	1
7	2	2	3	2	1
8	1	2	2	1	2

維度和值的資訊。MDIM 在掃描一次資料表後，產生 4 個 Level 的鏈結串列(Link List)的架構，如圖 1 所示。然後利用這樣的架構去產生頻繁項目集。1st-Level 的 DimSizeNode 是儲存長度和維度的組合數；2nd-Level 的 DimsetNode 是儲存維度的組合；3rd-Level 的 ValueSetNode 儲存在不同維度組合下值的組合以及出現在資料集中的筆數(Support Count)；4th-Level 的 SupMemNode 則儲存值的組合出現在資料集中的 ID。圖 1 是 MDIM 掃瞄表 1 多維度範例資料集後的結果。假設最小支持度(Minimum Support)定為 0.5，表示一個項目集(Itemset)至少要出現在 4 (=8*0.5)筆記錄中才能成為頻繁項目集(Frequent Itemset)。{(1,1),(4,1)}(意思代表第 1 個欄位值是 1 和第 4 個欄位值是 1)是長度為 2 的項目集。從表 1 中，我們可以發現有 4 筆記錄(ID 1、2、6 和 8)支持{(1,1),(4,1)}，大於等於我們定的最小支持度，所以{(1,1),(4,1)}是頻繁項目集。

MDIM 利用圖 1 的架構去挖掘出多維度的關聯規則，但需要很大的空間去儲存維度、值和交易 ID 的資訊。而且 MDIM 也只適用於類別型的資料，需要靠使用者具備相關的知識，將數值屬性離散化成類別型態，然後再利用 MDIM 的演算法去挖掘出多維度關聯規則。

三、多維度關聯規則探勘

關聯規則是從交易資料集中，發掘客戶同一時間所購買項目(Item)間之關聯性。假設 D 代表交易資料集，$I = \{i_1, i_2, \ldots, i_m\}$代表資料集中所有項目的集合，則每一筆交易 T 都是由 I 中部分的項目所組成(滿足 $T \subseteq I$)。通常，要衡量一條規則"A=>B"(A 和 B 是項目的集合，滿足 $A \subset I$, $B \subset I$ and $A \cap B = \varnothing$)是不是我們要的關聯規則，最常見的方式是利用支持度(Support)和信賴度(Confidence)這兩項指標。關聯規則必需滿足使用者所定義的最小支持度(Minimum Support)與最小信賴度(Minimum Confidence)。支持度代表 A 和 B 在交易資料集一起出現的機率；信賴度則代表在 A 出現的條件下，A 和 B 一起出現的機率，其公式如下所示:

$$\text{支持度} = P(A \cup B)$$

$$\text{信賴度} = \frac{P(A \cup B)}{P(A)}$$

然而，探勘多維度關聯規則並不像探勘傳統關聯規則，其探勘的資料來源並非交易資料庫，而是關聯式資料庫或是資料倉儲。因此，它不僅僅只有單純的產品購買資訊，如表 2 所示。

表 2 多維度資料集

ID	Occupation	Sex	Age	Salary	Paper	Printer
1	Businessman	M	34	38000	1	1
2	Sales Clerk	F	33	36000	1	0
3	Student	M	22	21000	0	1
4	Student	F	23	22000	1	0
5	Businessman	M	34	37000	1	1
6	Sales Clerk	M	24	27000	1	1
7	Sales Clerk	M	30	28000	0	1
8	Businessman	M	28	35000	1	1
9	Student	F	24	25000	1	0
10	Sales Clerk	F	27	32000	1	1

在表 2 中，Occupation、Sex、Age、Salary 是客戶特徵；Paper、Printer 則是產品購買資訊。此外，多維度資料通常包含兩種類型的屬性：數值屬性和類別屬性。在表 2 中，Occupation、Sex、Paper、Printer 屬於類別屬性；Age、Salary 則屬於數值屬性。由於關聯規則探勘需運作在類別型的資料上，因此我們通常會先將數值資料轉變成類別型式，才去探勘多維度的關聯規則。但如何將數值資料有效的離散化(Discretization)，則是一個重要的議題。在本研究中，我們將提出一個集群(Clustering)的方法來自動離散化數值資料。

假設多維度的資料集有 n 個維度(我們可視每一個屬性為一個維度)及 m 筆記錄，則 d_i 表示在多維度資料集中第 i 個維度；v_{ij} 表示在多維度資料集中第 i 個維度的第 j 筆記錄的值；$I_{ij} = (d_i, v_{ij})$ 表示在多維度資料中的一個項目(Item)，且 $1 \leqq i \leqq n$ 且 $1 \leqq j \leqq m$。項目集(Itemsets)是由項目所組成。例如，在表 2 中(2,1)是一個項目，表示第 2 個欄位(也就是"Sex"欄位，ID 欄位可以忽略)且值是 1 的這個項目。這個項目出現在 6 筆資料(ID 1、3、5、6、7 和 8)中，出現次數為 6。假如我們把 Occupation 中的"Sales Clerck"轉成用"2"代替，則{(1,2)(2,1)}是長度為 2 的項目集，出現在 3 筆資料中(ID 1、5 和 8)。如何探勘如表 2 的多維度資料，我們以下將分成全部都是類別屬性時的作法，以及當有數值屬性時應如何處理兩部分，分別加以說明。

3.1 全部都是類別屬性時的作法

表 3 為全部都是類別屬性的多維度資料集。為簡化說明起見，表 3 中所有屬性的屬性值均已編碼成數字。每個屬性其下的屬性值均從數字 1 開始去編碼。此外，為了探勘方便起見，我們會將表 3 的資料集轉換成類似傳統交易資料集的型式，如表 4 所示。

表 3 全部都是類別屬性的多維度資料集

ID	Field 1	Field 2	Field 3	Field 4	Field 5
1	1	1	1	1	1
2	2	2	1	2	1
3	2	2	2	2	1
4	1	1	2	1	1
5	2	3	2	1	2
6	1	1	1	1	1
7	2	3	2	1	2
8	2	2	1	2	1

表 4 轉換後的多維度資料集

ID	Items
1	(1,1) (2,1) (3,1) (4,1) (5,1)
2	(1,2) (2,2) (3,1) (4,2) (5,1)
3	(1,2) (2,2) (3,2) (4,2) (5,1)
4	(1,1) (2,1) (3,2) (4,1) (5,1)
5	(1,2) (2,3) (3,2) (4,1) (5,2)
6	(1,1) (2,1) (3,1) (4,1) (5,1)
7	(1,2) (2,3) (3,2) (4,1) (5,2)
8	(1,2) (2,2) (3,1) (4,2) (5,1)

假設我們定的最小支持度為 0.3(代表只要項目集在表 3 中出現至少 3 次就是頻繁項目集)，則探勘步驟如下：

Step 1: 找出長度為 1 的頻繁項目集

首先，掃瞄一次資料集，找出所有長度為 1 的頻繁項目(刪除出現次數少於 3 次的項目) L_1={(1,1):3 (1,2):5 (2,1):3 (2,2):3 (3,1):4 (3,2):4 (4,1):5 (4,2):3 (5,1):6}，其中，(Itemset):count 表示項目集:出現次數。

DB_Count	(5,1)	(1,2)	(4,1)	(3,1)	(3,2)	(1,1)	(2,1)	(2,2)	(4,2)
(5,1)		3	3	4	2	3	3	3	3
(1,2)			2	2	3			3	3
(4,1)				2	3	3	3		
(3,1)						2	2	2	2
(3,2)						1	1	1	1
(1,1)							3		
(2,1)									
(2,2)									3
(4,2)									

圖 2 三角矩陣 DB_Count

Step2:壓縮和排序資料

根據 L_1，每一筆交易都要去除不是頻繁的項目。這個步驟主要是要壓縮資料以加速探勘的速度。接著，L_1 中的每個項目會根據其出現的次數做排序，假如次數一樣時，則根據維度和值的大小作排序。因此，我們可以得到排序後的 L_1 = {(5,1):6 (1,2):5 (4,1):5 (3,1):4 (3,2):4 (1,1):3 (2,1):3 (2,2):3 (4,2):3}。所有的交易都要依據這個排序後的 L_1 重新排序。例如，第一筆交易重新排序後會變成，(5,1) (4,1) (3,1) (1,1) (2,1)，其它交易依此類推。排序後的資料集，如表 5 所示。

表 5 壓縮及排序後的多維度資料集

ID	Items
1	(5,1) (4,1) (3,1) (1,1) (2,1)
2	(5,1) (1,2) (3,1) (2,2) (4,2)
3	(5,1) (1,2) (3,2) (2,2) (4,2)
4	(5,1) (4,1) (3,2) (1,1) (2,1)
5	(1,2) (4,1) (3,2)
6	(5,1) (4,1) (3,1) (1,1) (2,1)
7	(1,2) (4,1) (3,2)
8	(5,1) (1,2) (3,1) (2,2) (4,2)

排序的目的在於要加速整體探勘的速度。不同的排序方法會產生相同的探勘結果，但執行的效能不同。

Step3: 掃瞄資料表直接產生 L_2 並切割出條件式資料集

根據 $L_1 \times L_1$ 的結果，我們可以建立一個三角矩陣 DB_Count。然後，掃瞄一次資料集去計數三角矩陣，如圖 2 所示(三角矩陣中無填值者計數為 0)，並產生所有長度為 2 的頻繁項目集(共 15 個) L_2={(5,1)(1,2)}:3, {(5,1)(4,1)}:3, {(5,1)(3,1):4, …}。接著，根據 L_2 去切割出 15 個條件式資料集，並將這 15 個條件式資料集放入 CDB 中。{(5,1)(1,2)}的條件式資料集，如表 6 所示。

表 6 {(5,1)(1,2)}的條件式資料集

Prefix	Postfix Dataset
{(5,1)(1,2)}	(3,1) (2,2) (4,2) (3,2) (2,2) (4,2) (3,1) (2,2) (4,2)

Step 4: 遞迴探勘條件式資料集並產生長度 k (k>2) 的頻繁項目集

假如 CDB 中尚有條件式資料集，則從 CDB 中取出一個資料集。然後，探勘此資料集中 Postfix Dataset 的部分，產生長度比目前 Prefix 長度多 1 的頻繁項目集及新的條件式資料集並放入 CDB 中。例如，從 CDB 中取出的是{(5,1)(1,2)}的條件式資料集。探勘此資料集中 Postfix Dataset 的部分後，發覺(2,2):3 及(4,2):3 的次數有達到最小支持度。因此，產生長度為 3 的頻繁項目集{(5,1)(1,2)(2,2)}:3 和{(5,1)(1,2)(4,2)}:3 及其條件式資料集並放入 CDB 中。{(5,1)(1,2)(2,2)}的條件式資料集如表 7 所示。

表 7 {(5,1)(1,2) (2,2)}的條件式資料集

Prefix	Postfix Dataset
{(5,1)(1,2)(2,2)}	(4,2) (4,2) (4,2)

持續遞迴探勘 CDB 中的條件式資料集，直到 CDB 為空集合為止。這個演算法，本研究稱之為 CMDAR (Categorical-Type Multidimensional Association Rule) 演算法。由於篇幅的限制，在此省略詳細演算法的列表。

3.2 針對數值屬性的作法

CMDAR 演算法只能處理全部都是類別屬性的多維度資料集。若多維度資料集中有數值屬性，則本研究將利用以下所提之集群方法，逐一離散化每個數值屬性。假設多維度資料集中包含 Age 及 Salary 這兩個數值屬性，如表 8 所示。自動離散化這兩個數值屬性的步驟如下：

表 8 多維度資料集中的兩個數值屬性

ID	Age (20~35)	Salary (20000~40000)
1	34	38000
2	33	36000
3	22	21000
4	23	22000
5	34	37000
6	24	27000
7	30	28000
8	28	35000
9	24	25000
10	27	32000

Step 1: 切割空間

首先，我們會讓使用者針對每個屬性設定切割此屬性值的最小單位 *u*。假設使用者設定切割 Age 的最小單位是 5(年)，則 Age 將會被切割成三個單元：U_1 = [20, 25), U_2 = [25, 30) and U_3 = [30, 35)。假設使用者設定切割 Salary 的最小單位是 10000(元)，則 Salary 將會被切割成兩個單元 U_1 = [20000, 30000), U_2 = [30000, 40000)。其中，切割的範圍是採右開放區間。

當切割完畢，我們會再讓使用者設一個高密度的門檻值 *d*(一般來說，*d* 必須小於或等於最小支持度)。也就是說，當單元內的資料個數大於"*d**資料筆數"時，則稱此單元為高密度單元。

圖 3 是假設 *d* 設為 0.3(代表一個單元至少要有 0.3*10 = 3 筆記錄才會成為高密度單元)時的情形。對 Age 來說，U_1 及 U_3 是高密度單元；對 Salary 來說，U_1 及 U_2 都是高密度單元。

圖 3 兩個屬性的高密度單元

表 9 多維度資料集

T_ID	Occupation	Sex	Age	Salary	Paper	Printer
1	2	1	34	38000	1	1
2	3	2	33	36000	1	0
3	1	1	22	21000	0	1
4	1	2	23	22000	1	0
5	2	1	34	37000	1	1
6	3	1	24	27000	1	1
7	3	1	30	28000	0	1
8	2	1	28	35000	1	1
9	1	2	24	25000	1	0
10	2	2	27	32000	1	1
		C		N		P

Step 2: 合併高密度單元形成最後離散化結果

我們選擇最左邊的單元當作我們的起始點，然後向右找相鄰的高密度單元，直到碰到非高密度的單元才停止。所有相鄰的高密度單元會合併成一個區塊。下一個起始點就從最接近這個區塊且並未包含在任何高密度單元的區塊開始。

以 Age 為例，U_1 是起始單元也是高密度單元。U_2 相鄰於 U_1 卻不是高密度單元，所以 U_2 不會和 U_1 合併成一個區塊，U_1 會自成一個區塊。下一個開始的單元是 U_3。但由於右邊沒有相鄰高密度單元，所以 U_3 也會自成一個區塊。此外，為了能夠提供更精確的資訊，我們會針對每一個區塊找到它的最小與最大的範圍。因此最後 Age 會離散化為[22,25)及[30,35)；Salary 因合併 U_1 及 U_2 會離散化為[21000,38001)。

Step3: 利用最小支持度刪除支持度不夠的區塊

假如區塊支持度小於最小支持度，則我們會刪除這個區塊。例如，假設最小支持度為 0.4(區塊中的資料筆數最少需有 4 筆)，則 Step 2 中的所有區塊均可保留(對 Age 來說[22,25)及[30,35)均有 4 筆資料在其中；對 Salary 來說[21000,38001)則有 10 筆資料

表 10 轉換後的多維度資料集

T_ID	Itemsets
1	(1,2) (2,1) (3,34) (4,38000) (5,1) (6,1)
2	(1,3) (2,2) (3,33) (4,36000) (5,1)
3	(1,1) (2,1) (3,22) (4,21000) (6,1)
4	(1,1) (2,2) (3,23) (4,22000) (5,1)
5	(1,2) (2,1) (3,34) (4,37000) (5,1) (6,1)
6	(1,3) (2,1) (3,24) (4,27000) (5,1) (6,1)
7	(1,3) (2,1) (3,30) (4,28000) (6,1)
8	(1,2) (2,1) (3,28) (4,35000) (5,1) (6,1)
9	(1,1) (2,2) (3,24) (4,25000) (5,1)
10	(1,2) (2,2) (3,27) (4,32000) (5,1) (6,1)

表 11 壓縮及排序後的多維度資料集

T_ID	Itemsets
1	(2,1) (1,2) (3,34) (4,38000) (5,1) (6,1)
2	(2,2) (3,33) (4,36000) (5,1)
3	(2,1) (3,22) (4,21000) (6,1)
4	(2,2) (3,23) (4,22000) (5,1)
5	(2,1) (1,2) (3,34) (4,37000) (5,1) (6,1)
6	(2,1) (3,24) (4,27000) (5,1) (6,1)
7	(2,1) (3,30) (4,28000) (6,1)
8	(2,1) (1,2) (3,28) (4,35000) (5,1) (6,1)
9	(2,2) (3,24) (4,25000) (5,1)
10	(1,2) (2,2) (3,27) (4,32000) (5,1) (6,1)

在其中)。由於篇幅的限制，在此省略詳細演算法的列表。

四、在不同的客戶特徵下，發掘經常被購買的產品組合

決策者需要不同的角度去幫助他們做決策。在這一節中，我們將探討如何利用第 3 節所提出的方法，發掘在不同的客戶特徵下，經常被購買的產品組合。基本上，多維度資料可以切割成兩個部份：客戶特徵與客戶購買產品的記錄。客戶特徵又可再細分成客戶的類別屬性及數值屬性，如表 9 所示。表 9 中，所有屬性的屬性值均已編碼成數字。同時，C(類別屬性)+N(數值屬性)是客戶特徵；P 是客戶購買產品的記錄(屬性值為 0 者代表沒有購買該項產品)。表 10 為轉換後的多維度資料集。由於 C 與 P 均為類別屬性，因此我們將這兩個部分合併成 CP，並利用 3.1 節所提出的方法探勘出排序後的 L_1={(2,1):6 (1,2):4

(2,2):4 (5,1):8 (6,1):7}(假設最小支持度定為 0.4)。

由於我們的目標是要先找到客戶特徵，然後再依據不同的特徵，找到不同的購買行為，所以即使(5,1)和(6,1)的支持度高於其它項目，他們仍是要放在排序的最後面((5,1)和(6,1)代表的是購買產品 5 和 6)。

此外，在客戶特徵中 N(數值屬性)的部份，我們可讓使用者定義數值屬性的重要性(也就是數值屬性的排名)，而且數值屬性和類別屬性是可以交叉排序的。例如，我們把所有類別屬性 C 放在數值屬性 N 之前，且我們認為數值屬性 Age 的重要性又比 Salary 高。依此假設，我們可以得到最終排序後的 L_1= {(2,1):6 (1,2):4 (2,2):4 (3,X) (4,Y) (5,1):8 (6,1):7}。其中，(3,X)=(3,[22,25))(3,[30,35)) 和 (4,Y)=(4,[21000, 38001))代表欄位 3 (Age)和欄位 4 (Salary)的集群結果(利用 3.2 節的集群演算法)。所有的交易都必須依照這個做排序。因此，第一筆交易會變成(2,1) (1,2) (3,34) (4,38000) (5,1) (6,1)，其他交易依此類推。表 11 為壓縮及排序後的多維度資料集。

根據 $L_1 \times L_1$ 的結果，我們可以建立一個三角矩陣 DB_Count。然後，掃瞄一次資料集去計數三角矩陣，並產生所有長度為 2 的頻繁項目集。接著，根據 L_2 去切割出多個條件式資料集，並將這些條件式資料集放入 CDB 中。{(5,1)(1,2)}的條件式資料集，表 12 列出{(2,1)(3,[30,35)}的條件式資料集。

表 12 {(2,1)(3,[30,35)}的條件式資料集

Prefix: {(2,1)(3,[30,35)}	
TID	Postfix Dataset
1	(3,34)(4,38000)(5,1)(6,1)
5	(3,34)(4,37000)(5,1)(6,1)
6	(3,24)(4,27000)(5,1)(6,1)
7	(3,30)(4,28000) (6,1)
8	(3,28)(4,35000)(5,1)(6,1)

根據我們的目標，我們不會產生只有產品組合的型樣，所以我們不需要產生{(5,1)}和{(6,1)}的條件式資料集。

假如目前 CDB 中尚有條件式資料集，則從 CDB 中取出一個資料集。然後，探勘此資料集中 Postfix Dataset 的部分，產生長度比目前 Prefix 長度多 1 的頻繁項目集及新的條件式資料集並放入 CDB 中。我們持續遞迴探勘 CDB 中的條件式資料集，直到 CDB 為空集合為止，即可找出所有的多維度關聯規則。我們將這個演算法稱之為 CNMDAR1 (Categorical-Type and Numerical-Type Multidimensional Association Rule I)。

五、在不同的產品組合之下，發掘經常購買該產品的客戶特徵

這一節中，我們將探討如何利用第 3 節所提出的方法，發掘在不同的客戶特徵下，經常被購買的產品組合。由於 C 與 P 均為類別屬性，因此我們將這兩個部分合併成 CP，並利用 3.1 節所提出的方法探勘出排序後的 L_1={(2,1):6 (1,2):4 (2,2):4 (5,1):8 (6,1):7}(假設最小支持度定為 0.4)。

由於我們的目標是要先探勘出經常被購買的產品組合，然後再依據不同的產品組合，找到客戶特徵。因此，(5,1)和(6,1)要放在排序的最前面((5,1)和(6,1)代表的是購買產品 5 和 6)。

此外，在客戶特徵中 N (數值屬性)的部份，我們可讓使用者定義數值屬性的重要性(也就是數值屬性的排名)，而且數值屬性和類別屬性是可以交叉排序的。例如，我們把所有類別屬性 C 放在數值屬性 N 之前，且我們認為數值屬性 Age 的重要性又比 Salary 高。依此假設，我們可以得到最終排序後的 L_1= {(5,1):8 (6,1):7 (2,1):6 (1,2):4 (2,2):4 (3,X) (4,Y)}。其中，(3,X)=(3,[22,25))(3,[30,35))和(4,Y)=(4,[21000, 38001))代表欄位 3 (Age)和欄位 4 (Salary)的集群結果(利用 3.2 節的集群演算法)。所有的交易都必須依照這個做排序。因此，

表 13 壓縮及排序後的多維度資料集

T_ID	Itemsets
1	(5,1) (6,1) (2,1) (1,2) (3,34) (4,38000)
2	(5,1) (2,2) (3,33) (4,36000)
3	(6,1) (2,1) (3,22) (4,21000)
4	(5,1) (2,2) (3,23) (4,22000)
5	(5,1) (6,1) (2,1) (1,2) (3,34) (4,37000)
6	(5,1) (6,1) (2,1) (3,24) (4,27000)
7	(6,1) (2,1) (3,30) (4,28000)
8	(5,1) (6,1) (2,1) (1,2) (3,28) (4,35000)
9	(5,1) (2,2) (3,24) (4,25000)
10	(5,1) (6,1) (1,2) (2,2) (3,27) (4,32000)

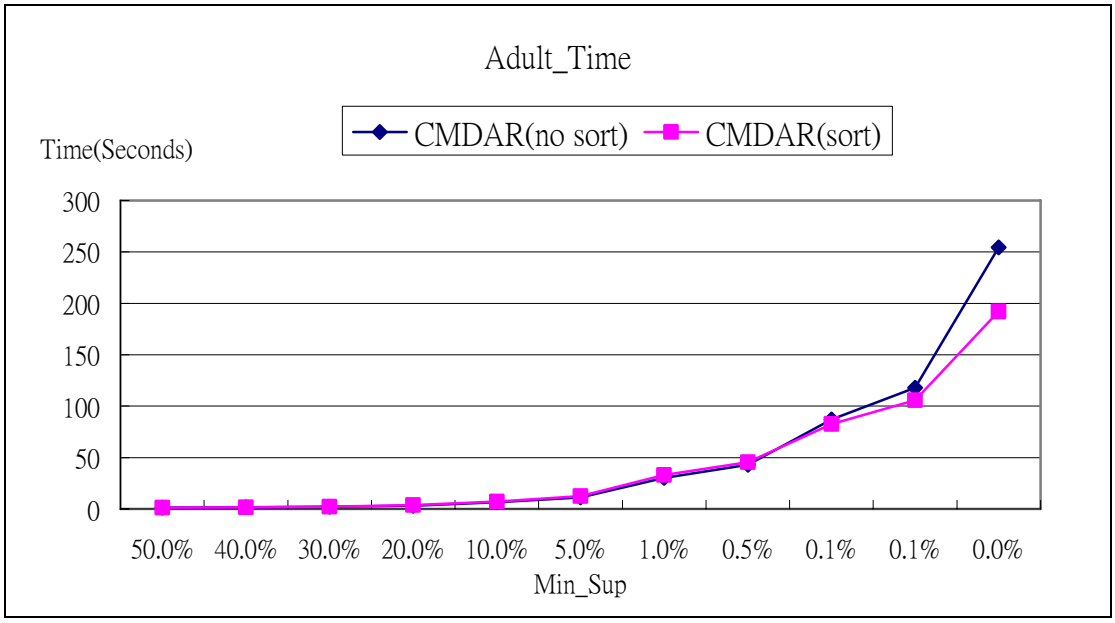

圖 4 CMDAR 的執行時間

表 10 的第一筆交易會變成(5,1) (6,1) (2,1) (1,2) (3,34) (4,38000)，其他交易依此類推。表 13 為壓縮及排序後的多維度資料集。

根據我們的目標，我們只需產生{(5,1)}、{(6,1)}的條件式資料集，所以我們不會產生只有客戶特徵的型樣。之後的做法與第 4 節相同，我們就不再贅述。我們將這個演算法稱之為 CNMDAR2 (Categorical-Type and Numerical-Type Multidimensional Association Rule II)。

六、實驗結果

在本節中，我們利用 Adult Dataset 來比較我們所提方法與 MDIM 演算法的差異。Adult Dataset 可在 UCI (http://mlearn.ics.uci.edu/MLSummary.html)的網站中找到。

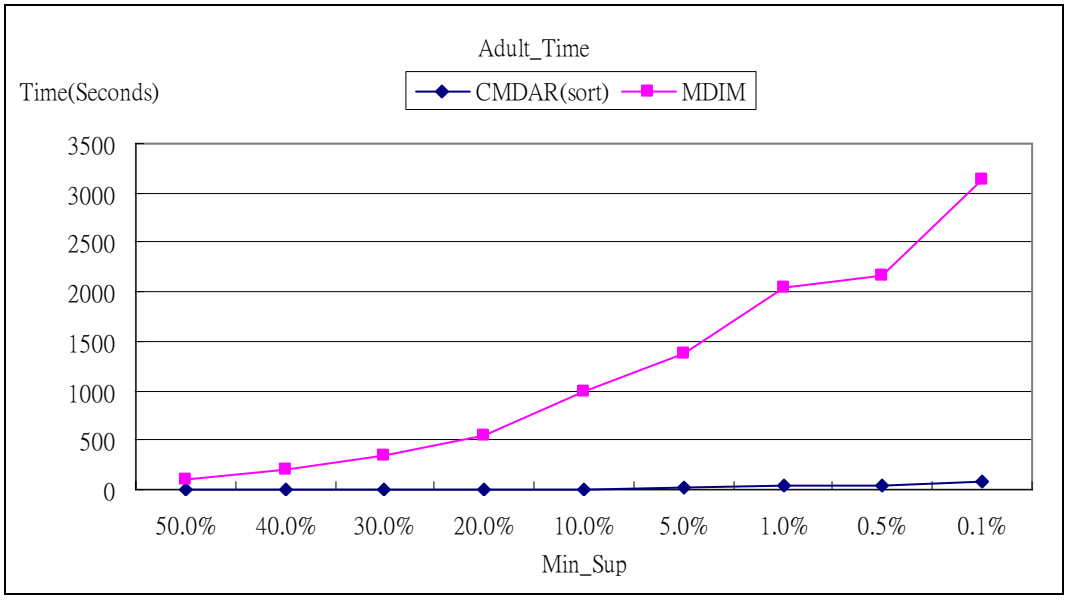

圖 5 CMDAR 與 MDIM 在執行時間上的比較

Adult Dataset 有 48,842 紀錄，14 條件屬性(包含 6 個數值屬性和 8 個類別屬性)及 1 個分類目標屬性。由於 MDIM 演算法只能處理類別型態的資料，所以我們的方法也排除了針對數值屬性的處理。同時，由於 MDIM 演算法也沒有將類別屬性區分成客戶特徵(C)及客戶購買產品的記錄(P)，所以我們最後採用 3.2 節中所提出的 CMDAR 演算法與之比較。此外，我們也去除 Adult Dataset 中數值屬性的部分，只保留 Adult Dataset 中的類別屬性。

圖 4 為 CMDAR 演算法在 L_1 部分排序與否的實驗結果。我們可以看出 CMDAR(sort)演算法比 CMDAR(no sort)演算法要快。尤其當最小支持度越小時，依照出現次數大小排序 L_1，演算法會更有效率。圖 5 為 CMDAR(sort)演算法與 MDIM 演算法在探勘執行時間上的比較。由圖 5 也可明顯看出 CMDAR(sort) 演算法的執行效率遠遠優於 MDIM 演算法。

七、結論

本研究提出一個適合市場區隔(Market Segmentation)應用的多維度關聯規則(Multidimensional Association Rules)探勘技術，利用條件式資料集(Conditional Dataset)挖掘出包含客戶特徵與客戶購買行為的多維度關聯規則。它不需要多次掃瞄目標資料集，並結合集群(Clustering)技術將數值資料自動離散化(Discretization)。本研究所提之方法將產生兩種不同角度的探勘結果：其一是在不同的客戶特徵下，發掘出經常被購買的產品組合；另一個則是在不同的產品組合下，發掘出經常購買該產品組合的客戶特徵。這兩種探勘結果可以提供決策者做市場區隔和制訂更為精確的行銷策略。實驗的結果顯示，本研究所提出之方法，其執行效率遠優於現行已提出的 MDIM 演算法。

八、誌謝

這篇論文的研究成果是國科會計劃(NSC 95-2221-E-130-013 和 NSC 95-2221-E-130-025)的一部份。我們在此感謝國科會經費支持這個計劃。

九、參考文獻

[1] R. Agrawal and R. Srikant. “Fast Algorithm for Mining Association Rules,” *Proceedings of International*

Conference on Very Large Data Bases (VLDB), pp. 487-499, 1994.

[2] J. Chiang and C. C. Wu. "Mining Multi-dimension Association Rules in Multiple Database Segmentation," *Proceedings of International Conference on Information Management (ICIM)*, 2005.

[3] J. Han, J. Pei, Y. Yin and R. Mao. "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach," *Data Mining and Knowledge Discovery*, Vol. 8, No. 1, pp. 53-87, 2004.

[4] J. S. Park, M. S. Chen and P. S. Yu. "Using a Hash-based Method with Transaction Trimming for Mining Association Rules," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, Vol. 9, NO. 5, pp. 813-825, 1997.

[5] C. Rygielski, J. C. Wang and D. C. Yen. "Data Mining Techniques for Customer Relationship Management," *Technology in Society*, Vol. 24, No. 4, pp. 483-502, 2002.

[6] M. J. Shaw, C. Subramaniam, G. W. Tan and M. E. Welge. "Knowledge Management and Data Mining for Marketing," *Decision Support Systems*, Vol. 31, No. 1, pp. 127-137, 2001.

[7] P. S.M. Tasi and C. M. Chen. "Mining Interesting Association Rules from Customer Databases and Transaction Databases," *Information Systems*, Vol. 29, pp. 685-696, 2004.

[8] W. X. Xu and R. J. Wang. "A Novel Algorithm of Mining Multidimensional Association Rules," *Proceedings of International Conference on Intelligent Computing (ICIC)*, pp. 771-777, 2006.