從交易資料庫中挖掘客戶的購物行為

顏秀珍
輔仁大學資訊工程學系
sjyen@csie.fju.edu.tw

李御璽
銘傳大學資訊工程學系
leeys@mcu.edu.tw

王思穎
輔仁大學資訊工程學系
pingu90@csie.fju.edu.tw

摘要

本篇論文提出一個新穎的資料探勘方式，針對於某種耗材性商品，找出大多數客戶對此商品的消耗行為，使得我們可以利用此次的購買數量來推導出下次會再來購買的時間。由於所找出之客戶對此商品的消耗行為可能有許多種，而這些不同的行為常常與客戶的背景屬性有關，所以我們也考慮客戶背景屬性與客戶對商品之消耗行為的關聯性，使得我們可以根據顧客的背景屬性值以及此次購買商品的數量，正確的預測出此客戶下次會再來購買的時間。

關鍵詞：資料探勘，關聯規則，序列型樣，商品消耗率，購物行為

一、導論

資料探勘（data mining）技術[7, 8, 11, 12, 19]於近幾年來開始蓬勃發展，其目的是要從資料庫中找出隱含於其中的有用資訊。*關聯規則探勘（Mining association rules）* [1, 8, 9, 10, 14, 18]可以從資料庫中找出有哪些商品是常常同時被購買的，商家可以藉由這些找出的資訊來做一些增加獲利之決策。而*序列型樣探勘（Mining sequential patterns）*[2, 3, 6, 7, 15, 19]可以從資料庫中找出有哪些商品常常被以一定的順序購買，商家可以藉由此資訊來做商品之促銷。*量化關聯規則探勘（Mining quantitative association rules）*[4, 13, 16]就是架構於關聯規則探勘之上，除了找出有哪些商品常常同時被購買外，還會找出這些商品的數量資訊。而*時間間隔序列型樣探勘（Mining time-gap sequential patterns）*[17]則是架構於序列型樣探勘之上，除了找出商品常常被購買的順序外，時間間隔序列探勘還會找出商品與商品間被購買之時間間隔資訊。相對於序列型樣探勘，時間間隔序列型樣探勘多了時間間隔的資訊，讓商家能將所得到的較多資訊做更有效率的利用，以得到更大的利益。

時間間隔序列型樣探勘雖然已經考量商品購買的時間間隔，但若是針對於耗材性商品，其距離下次的購買時間間隔，通常與此次購買的數量有關。本篇論文則是針對於耗材性商品的此種特性提出一個新穎的探勘方式，以*商品消耗率*的概念，同時包含購買數量以及間隔時間的考量。對於某次購買與下次購買之間的*商品消耗率*被定義為在某次交易中平均每個商品可以被使用的時間，如式（1）所示：

$$\text{商品消耗率} = \frac{\text{距離下次購買的時間間隔}}{\text{某次的購買數量}} \quad (1)$$

在此篇論文中，我們以商品消耗率來作為購物行為的指標屬性，將每次交易的商品消耗率對應到一維空間上，再利用叢集演算法找出大部分交易所集中的區間，我們稱之為*消耗區間*。若是這些區間中的交易筆數有超過我們所設定的門檻值，則此區間即為*商品的頻繁消耗區間*。由於商品的頻繁消耗區間可能不只一個，這表示客戶的購物行為可能不只一種，此時對於來購買商品的客戶，我們便無法區別他的購物行為應該是屬於哪一種，所以我們必須找出造成不同購物行為的主要原因，在此篇論文中，針對於耗材性商品，我們認為影響客戶購物行為之主要原因便是客戶的背景屬性，而不同的商品可能會受到不同的背景屬性所影響。對於任一種商品，我們先分析客戶所有的背景屬性，找出對此商品消耗率較具影響力的背景屬性，以便更進一步得知商品消耗率與各個客戶背景屬性之間的關聯性，最後將此關聯性以規則的形式表達，稱之為*商品消耗率關聯規則*。

由於商品消耗率以及客戶的屬性有些是屬於連續數值，需要做數值區間分割的動作，而傳統的方式大多是使用等寬分割或等深分割[4, 13]，並不能將資料集中的部分精確的切割出來，所以為了解決此問題，本篇論文利用叢集演算法來協助分割區間，確實將資料集中的區間切割出來，使得找出的商品消耗率關聯規則更具完整性。

二、商品消耗率關聯規則

在這一章節中，我們提出一個演算法來找出商品消耗率關聯規則，此演算法大致可以分

成三大部分：(1)資料表的過濾及轉換; (2) 找出*商品的頻繁消耗區間*; (3)找出*商品消耗率關聯規則*。

(一) 資料表的過濾及轉換

一般購物商場的資料表主要可分成兩個部分，第一部份為*客戶背景資料表*，其中記錄每個客戶的個人基本資料，如：客戶編號、姓名、性別、年齡…等，而第二部份為*客戶交易資料表*，其中記錄每一筆交易的詳細資料，如：交易編號、客戶編號、購買時間、購買商品種類及數量…等。由於資料表中包含許多我們演算法所不需要的資料，所以在演算法執行前，我們就需要將資料表作過濾，只留下有用的部分，再將其轉換成我們的演算法需要的輸入格式，以利於演算法的執行。

由於此演算法一次只針對一種商品作探勘，所以在商家決定要針對那種耗材性商品作探勘後，我們就將客戶交易資料表中其他商品的交易資訊都過濾掉，只留下所選擇的單一耗材性商品之交易資料，此種過濾後的資料表我們稱之為*單一商品客戶交易資料表*。單一商品客戶交易資料表原先是按照交易時間排序的，其資料表轉換步驟如下：

1. 將資料表依照客戶編號排序，如表一所示。
2. 對於同一個客戶，將前後兩筆資料的交易時間相減，即可得到兩次交易之間的時間間隔，在求出所有交易的間隔時間後，即可將購買時間欄位刪除，如表二所示。
3. 以間隔時間除以購買數量來計算出商品消耗率，計算完後將購買數量和間隔時間欄位都刪除，並給予每筆資料一個新的編號，此轉換完成的資料表稱為*商品消耗率資料表*，如表三所示。

(二) 找出商品的頻繁消耗區間

在此我們以商品消耗率來描述客戶的購物行為，將資料表中的商品消耗率當作是一維空間中的資料點，並根據其在空間中的分布，利用叢集演算法（clustering algorithm）找出商品消耗率所集中的區間，來得知客戶集中的購物行為，而這些商品消耗率所集中的區間，稱之為*商品的消耗區間*，簡稱為*消耗區間*。*消耗區間支持度*為此消耗區間中的資料點數與資料表中所有資料點數的比值，若消耗區間支持度有達到使用者所設定的*最小消耗區間支持度*，則為*商品的頻繁消耗區間*。

我們利用叢集演算法來找出消耗區間，而叢集演算法我們則參考 Martin Ester 等在 1996 年提出的 DBSCAN 演算法[5]。DBSCAN 演算法主要考量資料分布的密度，針對於每一個資料點，在使用者訂定的*半徑距離（e）*範圍內所包含的資料點數必須要超過*最少點數（MinPts）*才可形成叢集，而此資料點則為此叢集之中心點（core），若此叢集中有包含到其他叢集的中心點時，則將此兩個叢集結合成同一叢集，在所有叢集都已產生後，沒有被任何叢集所包含的資料點則被視為是雜訊（noise），不再考慮。在找尋消耗區間的過程中，資料點即為商品消耗率。在叢集演算法執行完後，我們可以找到所有叢集，這些叢集就是所有商品消耗率集中的區間，即為消耗區間。

找出商品消耗區間後，我們需要設定*最小消耗區間支持度*（MinRSup）來找出頻繁消耗區間。由於每個消耗區間的大小都不相同，若是給予各個消耗區間相同的最小消耗區間支持度是不公平的，所以我們利用各個消耗區間的大小作為設定最小消耗區間支持度的依據，範圍較大的消耗區間其最小消耗區間支持度應該較高，而範圍較小的消耗區間其最小消耗區間支持度應該較低，以達到公平的目標。不過最小消耗區間支持度可調整的範圍仍需要加以限制，以免對於各個消耗區間的門檻標準落差太大，所以在此演算法中我們以兩個參數來設定最小消耗區間支持度：*最小支持度（MinSup）*和*支持度調整百分比（SupRate）*，其中最小支持度為設定最小消耗區間支持度的基準，而支持度調整百分比則是限制最小消耗區間支持度最大可調整的範圍，使得最小消耗區間支持度最小可為（MinSup - MinSup × SupRate），表示為 Min(MinRSup)；最大可為（MinSup + MinSup × SupRate），表示為 Max(MinRSup)。我們將最小的消耗區間之最小消耗區間支持度設定為 Min(MinRSup)；將最大的消耗區間之最小消耗區間支持度設定為 Max(MinRSup)，而消耗區間大小介於最大和最小之間的消耗區間，其最小消耗區間支持度則可依照其與最大和最小消耗區間的大小比例來推算出來。

(三) 找出商品消耗率關聯規則

當所找出的頻繁消耗區間只有一個時，代表大多數客戶對於此商品的消耗率都是差不多的，所以此頻繁消耗區間可反應出所有客戶的購物行為。而當所找出的頻繁消耗區間多於一個時，表示客戶的購物行為不只一種，此時對於來購買商品的客戶我們便無法區別他的購物行為應該是屬於哪一種，在此我們認為客

客戶編號	購買時間	購買數量
1	90/01/01	3
1	90/01/06	4
1	90/01/12	3
2	90/01/02	3
2	90/01/08	3
3	90/01/01	5
3	90/01/12	6
3	90/01/24	4
4	90/01/01	3
4	90/01/12	2
4	90/01/25	2
4	90/02/04	2
5	90/01/02	2
5	90/01/09	3
5	90/01/28	3
6	90/01/ 03	4
6	90/01/10	5
7	90/01/02	3
7	90/01/12	3
7	90/01/22	4
7	90/02/04	3
7	90/02/23	4
7	90/03/20	3
7	90/04/07	4

表一 以客戶編號排序的單一商品客戶交易資料表

客戶編號	間隔時間	購買數量
1	5	3
1	6	4
2	6	3
3	11	5
3	12	6
4	11	3
4	13	2
4	10	2
5	7	2
5	19	3
6	7	4
7	10	3
7	10	3
7	13	4
7	19	3
7	25	4
7	18	3

表二 計算間隔時間後的單一商品客戶交易資料表

編號	客戶編號	商品消耗率
1	1	1.67
2	1	1.5
3	2	2
4	3	2.2
5	3	2
6	4	3.67
7	4	6.5
8	4	5
9	5	3.5
10	5	6.33
11	6	1.75
12	7	3.33
13	7	3.33
14	7	3,25
15	7	6.33
16	7	6.25
17	7	6

表三 商品消耗率資料表

戶的購物行為與其背景屬性有關。所以我們要找出各個頻繁消耗區間與客戶背景屬性的關聯性，稱之為*商品消耗率關聯規則*。利用商品消耗率關聯規則，我們可以依據客戶的背景屬性來推斷出此客戶的購物行為。

在這個步驟中，我們要找出各個頻繁消耗區間與客戶背景屬性的關聯性，因此我們要先將商品消耗率資料表(如表三)與客戶背景資料表(如表四)利用客戶編號屬性結合在一起，每一個商品消耗率，也就是資料點，所對應的客戶編號，都有其所屬的背景屬性值，我們稱其為此資料點的背景屬性值。對於每一個頻繁消耗區間，客戶的背景屬性值在此頻繁消耗區間中的*交易支持度*為在此頻繁消耗區間中，擁有此屬性值的資料點數與此頻繁消耗區間之所有資料點數的比值，若某一背景屬性值的交易支持度大於或等於使用者所設定的*最小交易支持度（MinTSup）*，則此屬性值就稱為在此頻繁消耗區間的*頻繁屬性值*。只包含一個屬性的頻繁屬性值，我們稱為 1-*頻繁屬性值*；而包含 k（k≥1）個屬性的頻繁屬性值稱為 k-*頻繁屬性值*，或稱為長度 k 的頻繁屬性值。

表四 客戶背景資料表

客戶編號	年齡	性別	家人數
1	22	女性	4
2	23	女性	3
3	21	男性	3
4	24	男性	4
5	25	女性	5
6	25	男性	2

客戶的背景屬性可分成兩種類型：❶類別屬性，如：性別、職業；❷數值屬性，如：年齡、家人數。對於類別屬性，我們可以直接計算此頻繁消耗區間中各種類別屬性值的資料點數，判斷此屬性值是否為頻繁屬性值。但是對於數值屬性，我們就不能依照其不同的屬性值直接計數，因為數值屬性的屬性值分布很廣，變化性很高，若是直接計數，各個計數值應該都十分小。例如：屬性年薪的屬性值可能從 0 到好幾百萬、千萬都有。所以對於數值屬性，我們需要將其數值做區間的分割，再對於分割出來的區間個別計數，來判斷此屬性值區間是否為頻繁屬性值。對於每一個頻繁消耗區間，我們將資料點投影在數值屬性的座標軸上，並利用 DBSCAN 演算法[5]對資料點在此座標軸上做分群，找出資料點在此座標軸上集中的區域，之後我們只要計算這些區域在此頻繁消耗區間的交易支持度，就可以判斷此屬性值區域在此頻繁消耗區間是否為頻繁屬性值。

(A) 找出 1-頻繁屬性值

對於每一個頻繁消耗區間，我們先找出所有的（k-1）-頻繁屬性值（k>1），再結合（k-1）-頻繁屬性值產生 *k-候選屬性值*。k-候選屬性值為包含 k 個屬性且有可能成為頻繁屬性值的背景屬性值。我們再計算 k-候選屬性值在此頻繁消耗區間的交易支持度，以決定其是否為此頻繁消耗區間的頻繁屬性值。找出 1-頻繁屬性值的方式如同前一節所描述，對於各個頻繁消耗區間，若背景屬性為類別屬性，我們可以分別計算此頻繁消耗區間中各種類別屬性值的交易支持度。若背景屬性為數值類型時，我們則需要先利用 DBSCAN 演算法來找出在此頻繁消耗區間中資料點集中的區域，再計算這些區域中的交易支持度，來判斷這些屬性值區域是否為頻繁屬性值。

範例一

假設最小交易支持度設定為 50%。若我們想要知道屬性性別與頻繁消耗區間[1.5，2.2]的關係，因為性別是類別屬性，我們可以分別計算此頻繁消耗區間中屬性值男性和屬性值女性的交易支持度，從表三中可知在此頻繁消耗區間中共有 6 個資料點，其編號分別為 1，2，3，4，5 和 11，其所對應的客戶編號分別為 1，1，2，3，3 和 6，由表四可得知，在此頻繁消耗區間中有 3 個資料點為女性，3 個資料點為男性，女性和男性的交易支持度皆為 3/6=50%，滿足最小交易支持度 50%，故此兩屬性值在此消耗區間中皆為頻繁屬性值。

對於每一個頻繁屬性值，我們也將在此消耗區間中擁有此頻繁屬性值之資料點的編號紀錄下來，我們稱其為此消耗區間中，此頻繁屬性值的 *編號串列*。例如：屬性值女性的編號串列為｛1，2，3｝，屬性值男性的編號串列是｛4，5，11｝。

(B) 由 k-頻繁屬性值找出 (k+1)-候選屬性值（k ³ 1）

我們將 k-頻繁屬性值或是 k-候選屬性值簡單表示成：（R , B_1I_1 , B_2I_2 , … , B_kI_k），其中 R 代表頻繁消耗區間值；B_1，B_2，…和 B_k 代表 k 個不同的客戶背景屬性，而 I_j 為屬性 B_j 的屬性值。例如：在消耗區間[1.5, 2.2]中，性別女性，且年齡介於 21 到 23 歲之間為 2-頻繁屬性值，則此 2-頻繁屬性值可以表示為：（[1.5, 2.2],性別女性,年齡[21，23]）。若有兩個 k-頻繁屬性值：(R_i, A_1I_1, A_2I_2,…, A_kI_k)和(R_j, B_1J_1, B_2J_2,…, B_kJ_k)，要由此兩個 k-頻繁屬性值來產生出(k+1)-候選屬性值（$k \geq 1$）需要符合下列條件：

1. 此兩個 k-頻繁屬性值的頻繁消耗區間值必須是相同的，即 $R_i=R_j$。
2. 對於此兩個 k-頻繁屬性值，前 k-1 個背景屬性以及其屬性值要相同，最後一個背景屬性不相同，即 $A_m = B_m$，$I_m = J_m$ 且 $A_k \neq B_k$，其中 $1 \leq m < k$，則可產生(k+1)-候選屬性值：($R_i, A_1I_1, A_2I_2, \ldots, A_{k-1}I_{k-1}, A_kI_k, B_kJ_k$)。

對於產生之(k+1)-候選屬性值，若其所有長度為 k 之子集皆為 k-頻繁屬性值，則保留；否則，予以刪除。對於保留之(k+1)-候選屬性值，我們將產生此(k+1)-候選屬性值的兩個 k-頻繁屬性值之編號串列做交集，其結果則為此(k+1)-候選屬性值的編號串列，而此編號串列中的編號個數，即為擁有此(k+1)-候選屬性值

的資料點數。

(C) 找出所有商品消耗率關聯規則

對於每一個頻繁屬性值，其代表的意義為該頻繁消耗區間的重要屬性值，然而擁有此屬性值的資料點數可能很多，而落在此消耗區間中的資料點數可能只有其中一小部份，因此，相對於整體而言，此頻繁屬性值在此消耗區間中的重要性也必須被打折扣，所以此資訊仍需要經過*信任度（confidence）*的評估。對於某一 k-頻繁屬性值的客戶，其行為是在某一頻繁消耗區間的信任度為在此頻繁消耗區間中擁有此頻繁屬性值的資料點個數與整個資料庫中擁有此頻繁屬性值的資料點個數之比值。

對於某一頻繁消耗區間 R 中的 k-頻繁屬性值（$R, B_1I_1, B_2I_2, \ldots, B_kI_k$），若其信任度有達到使用者所設定的*最小信任度*，則此 k-頻繁屬性值可以產生*商品消耗率關聯規則*，表示為：$B_1 = I_1$ AND $B_2 = I_2$ AND…AND $B_k = I_k \Rightarrow$ 消耗率 R。例如：（[1.5，2.2），性別女性，年齡[21，23]）為 2-頻繁屬性值，若其信任度有達到最小信任度，則其所產生的商品消耗率關聯規則為：性別=女性 AND 年齡=21　23 歲 ⇒ 消耗率[1.5，2.2]，其意義為性別女性且年齡介於 21　23 歲的客戶，其單位時間會消耗 1.5　2.2 個商品。

(四) 探勘結果之應用

商品消耗率關聯規則可分為前題（antecedent）與結論（consequent）兩部分，前題為客戶的背景屬性值，結論則為此商品的消耗區間。當我們要預測某一客戶對此商品的消耗率時，符合此客戶之背景屬性值的規則可能不只一條，此時我們應該選取哪條規則來預估此客戶的行為呢？我們有以下的考量順序：

(1) 先找前題符合客戶之背景屬性值且屬性值最多的商品消耗率關聯規則。

(2) 若是同時有多條符合(1)之消耗率關聯規則，則優先考慮信任度較高者。

對於某一客戶，當我們找到其所符合的商品消耗率關聯規則後，即可得知其所對應的消耗區間，而此消耗區間即表示每個商品約略可供應此客戶使用的時間，所以我們只要利用此消耗區間與客戶此次購買商品之數量兩者資訊，經由簡單的乘法運算，即可以預估客戶下次會再來購買此商品的時間，計算方式如式（2）所示，而商家便可以利用此計算出來的資訊，做為商品庫存之管理以及個人化商品促銷的活動，以增進其利益之獲得。

距離下次購買的時間間隔 =
找出之消耗區間×此次購買的數量　(2)

若我們預測某一客戶的商品消耗區間為[1.5，2.2]。即每個商品約可供應她使用 1.5　2.2 天的時間，所以若是她今天購買了 3 個商品，我們則可以預估她將在（1.5×3）　（2.2×3）天，即 4.5　6.6 天用完此商品而再來購買，而商家就可以約略在這個時段寄發此商品的相關資訊及促銷活動給此客戶，以掌握其時效性。

三、實驗報告

這個章節中，我們對此篇論文所提出的演算法做執行效能的評估，由於真實資料難以取得，所以我們利用程式參數的設定，模擬真實環境，產生出人工資料來做實驗，其人工資料的產生方式以及實驗的結果詳述如下。

表五 產生人工資料之參數設定

參數	參數解釋
TD	商品消耗率資料表中的資料筆數
CN	類別屬性的個數
NN	數值屬性的個數
FC	各個頻繁消耗區間中，頻繁類別屬性值的個數
FN	各個頻繁消耗區間中，頻繁背景屬性值的個數
RN	頻繁消耗區間的個數
DS	資料分散度
MRS	最小消耗區間支持度
MTS	最小交易支持度

(一) 人工資料的產生

由於我們的演算法是針對單一商品的交易資料來分析，所以我們只針對一種商品來產生出此商品的商品消耗率以及客戶的背景屬性值，所有需要設定的參數值列於表五中，其中 DS 是資料分散度，其所代表的意義為每 100 個資料點，平均分布於多少個商品消耗率數值，例如：當我們設定資料分散度為 80%，此時若是有 100 個資料點，則會分布於 100×80%=80 個商品消耗率數值，也就是有 80 個不同的消耗率數值；有 20 個資料點的消耗率數值與其他 80 個資料點的某些消耗率數值相同。RN 為頻繁消耗區間的個數，由於我們

設定每個頻繁消耗區間都產生一條包含屬性值最多的規則，所以 RN 也是包含屬性值最多的規則個數。

(二) 實驗結果分析

表六 演算法之參數設定

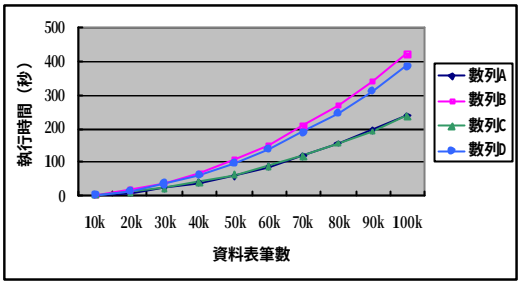
參數	參數解釋
MS	最小支持度
SR	支持度調整百分比
MTS	最小交易支持度
C	最小信任度

實驗所需要用到的參數如表六所示，利用 MS 與 SR 這兩個參數我們便可以計算出最小消耗區間支持度。我們針對四種不同的參數設定，如表七所示，我們將數列 A 的參數設定視為基準；數列 B 的參數中總背景屬性的個數是數列 A 的兩倍，也就是整個資料表大小約略是數列 A 的兩倍；數列 C 的參數中頻繁屬性值的個數是數列 A 的兩倍，也就是所產生出來的最長規則的長度是數列 A 的兩倍；數列 D 的參數中頻繁消耗區間的個數是數列 A 的兩倍，也就是總規則個數是數列 A 的兩倍。我們將表五中的參數 TD 從 10K 增加到 100K，而其他參數都相同的情況下來執行演算法，演算法設定之參數 MS 為 20%、SR 為 0%、MTS 為 70%、C 為 80%，我們可以得到執行時間的曲線圖如圖一所示，由圖中我們可以發現，針對於同一組數列，隨著資料筆數增加，程式的執行時間也隨之緩慢增加，呈現出線性成長，並沒有因為資料增加而有執行時間暴漲的情況，表示此演算法十分穩定。

表七 人工資料參數

參數	CN	NN	FC	FN	RN
數列 A	4	4	2	2	2
數列 B	8	8	2	2	2
數列 C	4	4	4	4	2
數列 D	4	4	2	2	4

而對於不同數列，以數列 A 為基準，數列 B 的資料庫大小是數列 A 的兩倍；數列 D 的總規則個數是數列 A 的兩倍，我們預估所需要的執行時間應該都是數列 A 的兩倍，但從實驗結果發現，此兩者所需要的時間還略少於數列 A 所需時間的兩倍，由此可知我們的演算法對於不同類型的資料，在速度上都十分穩定。而數列 C 的最長規則長度是數列 A 的兩倍，預估所需要的時間也要略長於數列 A，但從實驗結果發現，數列 C 所需要的時間和數列 A 幾乎完全相同，由此可知在長度 2 以後的規則都只需要做結合和編號串列交集的動作，速度十分快，對於整體的速度並沒有任何影響。



圖一 演算法的執行時間

四、結論與未來工作

在這篇論文中，我們提出一個新穎的資料探勘方式，針對於某種耗材性商品，以商品消耗率的概念，同時包含商品的購買數量以及間隔時間，找出大多數客戶的購物行為，並考量客戶的背景屬性，以區別出不同背景之客戶的購物行為，找出背景屬性與購物行為之間的關聯規則，使得所得到的資訊更加精確、詳盡，最後便可利用客戶的背景屬性以及此次購買商品的數量來推導出客戶下次會再來購買此商品的時間，提供商家更充足的資訊來做及時的廣告及促銷活動，以增進利益的獲得。而對於演算法效能的評估，從實驗數據中我們可以發現，隨著資料點數的增加，程式的執行時間也隨之緩慢增加，呈現出線性成長，並沒有因為資料增加而有執行時間暴漲的情況，表示此演算法十分穩定。

由於此演算法一次只能處理一種耗材性商品，若有許多耗材性商品同時需要探勘，將需要反覆處理，較為麻煩，所以在未來的工作中，我們希望可以同時處理多種耗材性商品。由於同時處理多種商品將會需要使用許多的記憶體空間以及更久的執行時間，所以發展較有效率的演算法來降低記憶體的使用量以及減少執行的時間將是未來工作的首要任務。

誌謝

這篇論文的研究成果是國科會計劃（NSC 92-2213-E-030-019, NSC 92-2213-E-130-007, 和 NSC91-2622-E-130- 002-CC3）的一部份。我們在此感謝國科會經費支持這個計劃的研究。

五、參考文獻

[1] R. Agrawal, R. Srikant: "Fast Algorithm for Mining Association Rules", *International Conference on Very Large Data Bases*, pp. 487-499, Sept. 1994.

[2] R. Agrawal, R. Srikant: "Mining Sequential Patterns", *International Conference on Data Engineering (ICDE)*, pp. 3-14, March 1995. Expanded version available as IBM Research Report RJ9910, October 1994.

[3] R. Agrawal, R. Srikant: "Mining Sequential Patterns: Generalizations and Performance Improvements", *International Conference on Extending Database Technulogy (EDBT)*, pp. 3-17, March 1996. Expanded version available as IBM Research Report RJ 9994, December 1995.

[4] R. Agrawal, R. Srikant: "Mining Quantitative Association Rules in Large Relational Tables", *ACM International Conference on Management of Data (SIGMOD)*, pp. 1-12, June 1996.

[5] Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu: "A Density-Based Algorithm for Discovering Clusters in Large Spatial Database with Noise", *ACM Conference on Knowledge Discovery in Data (KDD)*, pp. 226-231, August 1996.

[6] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Qiming Chen, Umeshwar Dayal, and Meichun Hsu: "FreeSpan : Frequent Pattern-Projected Sequential Pattern Mining", *ACM Conference on Knowledge Discovery in Data (KDD)*, pp. 355-359, 2000.

[7] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu: " PrefixSpan : Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", *International Conference on Data Engineering (ICDE)*, pp. 215-224, 2001.

[8] J. Han, J. Pei, H. Lu, S. Nishio, S. Tang, and D. Yang: "H-Mine: Hyper-structure Mining of Frequent Patterns in Large Databases", *Proc. The 2001 IEEE International Conference on Data Mining (ICDM'01)*, San Jose, California, Novermber 29-December 2, 2001.

[9] Jiawei Han, Jian Pei, Yiwen Yin: "Mining frequent patterns without candidate generation", *ACM International Conference on Management of Data (SIGMOD)*, May 2000.

[10] Bing Liu, Wynne Hsu, Yiming Ma: "Mining Association Rules with Multiple Minimum Supports ", *ACM Conference on Knowledge Discovery in Data (KDD)*, pp. 337-341, 1999.

[11] Chang-Hung Lee, Cheng-Ru Lin, and Ming-Syan Chen : "On Mining General Temporal Association Rules in a Publication Database", *Proc. of the First IEEE International Conference on Data Mining (ICDM-01)*, November 29 - December 2, 2001.

[12] Bamshad Mobasher, Honghua Dai, Tao Luo, Miki Nakagawa: "Effective Personalization Based on Association Rule Discovery from Web Usage Data", *Proc. of the 3rd ACM Workshop on Web Information and Data Management (WIDM01)*, November 2001.

[13] Wei Wang, Jiong Yang, Philip S. Yu: "Efficient Mining of Weighted Association Rules (WAR)", *ACM Conference on Knowledge Discovery in Data (KDD)*, pp. 270-274, 2000.

[14] Show-Jane Yen and A.L.P. Chen (2001). "A Graph-Based Approach for Discovering Various Types of Association Rules," *IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)*, Vol.13, No.5, pp. 839-845, 2001.

[15] Show-Jane Yen and Chung-Wen Cho (2001). "An Efficient Approach for Updating Sequential Patterns Using Database Segmentation," *International Journal of Fuzzy Systems (IJFS) Special Issue on Soft Computing and Data Mining*, Vol.3, No. 2, pp. 422-431, 2001.

[16] Show-Jane Yen, Yue-Shi Lee, Si-Wei Chen: "Mining Quantitative Association Rules from Transaction Database", *Proceedings of 10th National Conference on Fuzzy Theory and Its Applications*, pp. D520-D525, 2002.

[17] Show-Jane Yen and Yue-Shi Lee. "Mining Time-Gap Sequential Patterns from Transaction Databases," *Journal of Computers*, Vol. 14, No. 2, pp. 30-46, June 2002.

[18] Mohammed J. Zaki: "Generation Non-Redundant Association Rules", *ACM Conference on Knowledge Discovery in Data (KDD)*, pp. 34-43, 2000.

[19] Mohammed J. Zaki: "SPADE: An Efficient Algorithm for Mining Frequent Sequences", *Machine Learning Journal, special issue on Unsupervised Learning*, pp. 31-60, Vol. 42 Nos. 1/2, Jan/Feb 2001.