

Enhancement of Library Operation and Service by Data Mining and Statistics

Student: An-Chi Wu Advisor: Dr. Hao-Ren Ke, Dr. Wei-Pang Yang

Institute of Computer and Information Science
National Chiao Tung University

ABSTRACT

With rapid advancement of Internet and World Wide Web (WWW), WWW has been becoming an important information source. The proliferation of WWW also encourages the development of electronic library. Many libraries are aiming to provide patrons better services by means of computer and telecommunication technology.

This thesis aims at discovering the community patterns of NCTU patrons via data mining. The data used in mining are the bibliographic and holding records of NCTU Library, as well as the transaction history logs.

We expect that NCTU Library can apply the research findings to its operation and service so that NCTU Library can play a more proactive role in the learning and researching of NCTU patrons. The community patterns of patrons that we want to discover in this thesis are the common patterns and sequences of collections, borrowed by patrons.

After we find out the community patterns, we want to achieve the following goals: (1) attract more patrons to borrow collections, (2) increase collection borrowing rate, (3) raise the loyalty of patrons, (4) assist the policy making on the acquisition of duplicate copy, and (5) encourage the rapid circulation of collections.

利用資料探勘的技術及統計的方法

增強圖書館的經營與服務

研究生：吳安琪

指導教授：柯皓仁博士，楊維邦博士

國立交通大學資訊科學研究所

摘要

隨著網際網路和全球資訊網的蓬勃發展，資訊傳播不再是傳統平面媒體的特權，人們已經漸漸習慣在全球資訊網上尋找自己要知道的資訊。全球資訊網的普及也促成了電子圖書館的日漸成熟，如何利用電腦與網路科技提昇圖書館對讀者的服務(即圖書館的電子化服務)亦是各圖書館努力追尋的目標。

本論文以交通大學圖書館的書目、館藏、交易歷史記錄為基礎，運用資料探勘(Data Mining)的技術探索交通大學讀者的社群(Community)特性，讓交通大學圖書館能夠在交通大學讀者學術、知識的吸收和產出中扮演更積極的角色。本論文所希望探索的讀者社群關係包含：館藏借閱的共同性及館藏借閱的順序性。

當我們發掘出讀者的社群關係後，希望能運用這些社群關係來達到下列目的：吸引讀者到館借閱、提昇館藏借閱率、提昇讀者忠誠度、協助館藏複本採訪政策及促進館藏流通率。

誌謝

感謝兩年來指導教授柯皓仁博士及楊維邦博士在各方面的細心指導與照顧，使我不只在課業與研究上獲得許多寶貴的經驗，在生活與做人處事上，更給了我不少的影響與啟示。

同時也要感謝交通大學圖書館館員們的協助與指導，使我對圖書館學有更進一步的認識。特別要感謝蔡淑琴小姐的幫忙，提供我各方面的資料，使我論文得以順利完成。

此外，還要感謝實驗室的學長們對我研究上的啟發與指導，提供我許多寶貴的建議。還有實驗室的夥伴們，對我這兩年來的關懷與照顧。

最後，要感謝我親愛的家人與朋友們長久以來的支持與鼓勵，使我能專心致力於研究，並得以順利完成學業。

目錄

英文摘要.....	I
中文摘要.....	II
誌謝.....	III
目錄.....	IV
表目錄.....	V
圖目錄.....	VI
第一章 簡介.....	1
第一節 研究動機.....	1
第二節 研究目的.....	3
第三節 論文架構.....	4
第二章 相關研究工作.....	5
第一節 資料探勘.....	5
第二節 相關規則探勘.....	6
第三節 相關規則探勘之延伸問題.....	13
第四節 循序模式探勘.....	14
第三章 資料探勘在圖書館上的應用.....	21
第一節 目標.....	21
第二節 方法.....	23
第四章 分析結果.....	30
第一節 探索讀者社群關係.....	30
第二節 吸引讀者到館借閱.....	34
第三節 提昇館藏借閱率.....	36
第四節 提昇讀者忠誠度.....	37
第五節 協助館藏複本採訪政策.....	38
第六節 促進館藏流通率.....	40
第五章 結論與未來研究方向.....	43
第一節 結論.....	43
第二節 未來研究方向.....	44
參考文獻.....	46
附錄.....	48

表目錄

表 2 - 3 - 1 : 各種類型相關規則探勘之延伸問題.....	14
表 4 - 1 - 1 : 1999 年讀者最常一起借閱的館藏	31
表 4 - 1 - 2(a) : 讀者最常一起借閱的二本書(一).....	32
表 4 - 1 - 2(b) : 讀者最常一起借閱的二本書(二)	33
表 4 - 1 - 3 : 讀者最常依序借閱的二本書.....	34
表 4 - 2 - 1 : 交通大學近一年來最熱門的二十本書	35
表 4 - 5 - 1 : 讀者經常重複借閱的書籍	39
表 4 - 5 - 2 : 讀者經常重複預約的書籍.....	40
表 4 - 6 - 1(a) : 交通大學八十九學年度第二學期熱門預約書(一)	41
表 4 - 6 - 1(b) : 交通大學八十九學年度第二學期熱門預約書(二)	42

圖目錄

圖 2 - 1 - 1 : 資料探勘流程圖.....	6
圖 2 - 2 - 1 : Apriori 例子	8
圖 2 - 2 - 2 : Apriori 演算法	9
圖 2 - 2 - 3 : 雜湊樹.....	10
圖 2 - 2 - 4 : subset 函數	11
圖 2 - 2 - 5 : TidApriori 例子.....	12
圖 2 - 4 - 1(a) : 交易序列.....	15
圖 2 - 4 - 1(b) : 大 1-序列對應編號	15
圖 2 - 4 - 2 : 轉換資料庫.....	15
圖 2 - 4 - 3 : AprioriAll 產生候集時連結的步驟	15
圖 2 - 4 - 4 : AprioriAll L_3 產生 C_4	16
圖 2 - 4 - 5 : 分類圖.....	17
圖 2 - 4 - 6 : GSP L_3 產生 C_4	18
圖 2 - 4 - 7 : Data-Sequence.....	19
圖 2 - 4 - 8 : 各個項目的時間陣列.....	20
圖 3 - 2 - 1 : 找出讀者特性	25
圖 3 - 2 - 2 : GSP 中儲存各個項目的時間陣列.....	26
圖 3 - 2 - 3 : 本論文中儲存各個項目的時間陣列.....	26
圖 3 - 2 - 4 : 吸引讀者到館借閱.....	27
圖 3 - 2 - 5 : 提昇館藏借閱率.....	27
圖 3 - 2 - 6 : 提昇讀者忠誠度.....	28
圖 3 - 2 - 7 : 複本採訪政策.....	29
圖 3 - 2 - 8 : 促進館藏流通率.....	29
圖 4 - 2 - 1 : 1999 年每月借書人數分佈圖	35
圖 4 - 4 - 1 : 1999 年借閱書籍本數的人數分佈圖	38
圖 4 - 4 - 2 : 1999 年借閱圖書數目為 10 本以下的人數分佈圖.....	38

第一章 簡介

第一節 研究動機

隨著網際網路和全球資訊網的蓬勃發展，資訊傳播不再是傳統平面媒體的特權，人們已經漸漸習慣在全球資訊網上尋找自己要知道的資訊。全球資訊網的普及也促成了電子圖書館的日漸成熟，如何利用電腦與網路科技提昇圖書館對讀者的服務(即圖書館的電子化服務)亦是各圖書館努力追尋的目標。

文獻上電子圖書館的相關名詞很多，諸如數位圖書館(Digital Library)、虛擬圖書館(Virtual Library)、無牆圖書館(Library without Wall)等皆是。

American Digital Library Federation曾在1998年提出對數位圖書館(電子圖書館)的定義[16][19]：

數位圖書館乃是擁有相關資源(包含軟硬體設備、網路、專業人士...等)以執行下列任務的機構：對數位形式的館藏進行挑選、組織、提供使用、解譯、傳播、保持完整性、長期保存等工作，並使這些數位形式的館藏能為特定讀者群快速且經濟地運用。

從這個定義上來看，電子圖書館與傳統圖書館的任務與功能其實並無不同，只不過電子圖書館所處理的乃是以數位形式的資料為主。根據以上定義，Cleveland[15]提出了電子圖書館的幾項特性，其中之一為：

數位圖書館是傳統圖書館的另一面貌，其館藏包含電子式(與數位式)以及印刷和其他(例如，影帶和聲音)媒材。

在胡述兆與吳祖善二位合著的「圖書館學導論」[18]中指出：「所謂的圖書館，就是運用科學方法，採訪、整理、保存各種印刷與非印刷的資料，以便讀者利用的機構。」而圖書館的功能則主要有[17]：(1) 收藏與保存，(2) 組織和呈現，

(3) 存取與檢索，(4) 分析、綜合與傳播。在前述定義中特別指明科學方法的運用，我們認為所謂的電子圖書館，正是要求館員運用最新的電腦與網路科技來進行各項(數位或傳統)資料的蒐集、組織、保存、傳播與利用，以完成圖書館被賦予的功能，這正與 American Digital Library Federation 提出的定義不謀而合。

在實務上，究竟要如何實現電子圖書館呢？陳亞寧在「另類圖書館：電子圖書館綜觀」[20]一文中提出以下推論：

- ◆ 電子圖書館是結合資料、檔案、典藏、資訊、知識、服務以及作業的集合體。
- ◆ ELIB = Electronic Data/File/Archive/ Information/Knowledge
 - + Automated/Electronic Service
 - + Automated/Electronic Operation

綜上所述，我們可以看出，電子圖書館的主要元素有三，即電子化館藏(E-Collection)、電子化作業(E-Operation)，以及電子化服務(E-Service)，因此若要實現電子圖書館，則必須從館藏、作業、服務等三方面的電子化著手。

圖書館的一切作為都是為了提供讀者良好的服務，協助讀者獲取資訊、運用資訊，從而產生知識。本論文即是著重於電子化服務，所謂的電子化服務，指的乃是運用電腦與網路科技來從事圖書館讀者服務。我們認為群體化乃是電子圖書館時代讀者服務的未來發展方向之一，所謂的群體化即是社群(Community)的概念，因為知識的產生有時並非光靠單一的個體就能達成的，而是得藉由具有相同興趣、專長的個體彼此激發靈感、分享心得和知識方能加速知識的產出。本論文的目的即在於運用資料探勘(Data Mining)的技術來探索社群的持性，從而達成群體化的電子圖書館讀者服務。

當圖書館想要既深且廣地運用電子技術來實施讀者服務時，首先一定要考量讀者到底需要什麼服務、現行的服務有什麼不足之處？我們覺得，面對圖書館龐大的館藏資訊，讀者往往不知哪些館藏是符合其需求的。以下茲列舉一些讀者常

遇到的情況：

1. 讀者難以決定要借哪些館藏：讀者在館藏查詢系統上檢索所得之館藏數量非常大且雜，讀者往往很難決定要借哪些館藏。
2. 讀者不知道有哪些相關館藏：或許讀者事先已預設好要借的館藏主題，但因不熟悉館藏系統的功能而漏找到一些相關的館藏，而這些相關的館藏卻是對讀者的學習或研究很有幫助的。
3. 館藏數量漸趨龐大，讀者不知如何利用這些資源：圖書館長期經營後必然會有館藏越來越多的傾向，此時有效率的管理是非常重要的，如果不適時地給予讀者適當的建議，那麼，可能會有許多值得一讀的館藏不會被讀者借閱。

有鑑於此，我們認為若能針對讀者的借閱行為傾向加以分析，藉由分析結果適時地給予讀者館藏借閱的建議，將可改善前述情況。

本論文應用資訊科技(Information Technology)及企業化模式(Business Model)來經營圖書館，利用資料探勘(Data Mining)的技術找出大多數使用者的借閱行為傾向，以改善前述讀者於借閱館藏時所遭遇到的困難，並進一步提供讀者適時適當的建議，提昇對讀者的服務。

資料探勘(Data Mining)也叫做資料庫探勘(Database Mining)或資料庫知識發掘(Knowledge Discovery in Database)。簡單地說，資料探勘是從儲存於資料庫(Database)、資料倉儲(Data Warehouse)或其他資訊儲存器(Information Repository)的大量資料中發掘出我們感興趣的知識之處理過程。例如在超級市場上，我們可以從商店的交易資料庫裡找出顧客的消費行為，藉這些行為來提昇商店的業績。

第二節 研究目的

本論文以交通大學圖書館的書目、館藏、交易歷史記錄為基礎，運用資料探勘的技術探索交通大學讀者的社群特性，並運用資料探勘的成果來提昇圖書館的

經營與服務，期能使交通大學圖書館在交大讀者學術、知識的吸收和產出中扮演更積極的角色。本論文擬探索的讀者社群關係包含：

1. 館藏借閱的共同性：有類似興趣的讀者通常所借閱的館藏也會很類似，如何藉由資料探勘的方法把館藏借閱的共同性找出來？
2. 館藏借閱的順序：讀者借閱館藏可能會先借入門的再借深入的，如何藉由資料探勘的方法把讀者借閱館藏順序的特性找出來？

當我們發掘出讀者的社群關係後，希望能運用這些社群關係達到下列目的：

1. 吸引讀者到館借閱：我們發現很多讀者從未借過館藏，要如何增加借閱的讀者人數？
2. 提昇館藏借閱率：我們發現有很多館藏是未曾被借閱過的，要如何才能把這些館藏推銷出去呢？
3. 提昇讀者忠誠度：我們發現很多讀者只借一、兩次就不再借閱，要如何提昇讀者的忠誠度，使讀者能夠持續地借閱？
4. 協助館藏複本採訪政策：圖書館針對一本書所採購的複本數往往有限，但有些熱門書讀者常常要預約很久才能借到，有些讀者因此放棄借閱。要如何找出哪些是熱門的書？哪些書該多買一些複本？
5. 促進館藏流通率：很多熱門的館藏借出後往往會到應還日才歸還，其他讀者要借閱便要預約很久才能借到，很多人因此放棄借閱。要如何增加這些館藏的流通率，讓更多的讀者可以借到這些館藏？

第三節 論文架構

本論文第二章介紹相關研究工作，第三章描述資料探勘在圖書館上的應用，第四章探討分析結果，第五章歸納出結論與未來研究方向。

第二章 相關研究工作

本論文係以資料探勘技術應用於圖書館書目、館藏、交易歷史記錄中，期能發掘讀者借閱館藏之共同性與順序性，進而由資料探勘之成果提昇圖書館之經營與服務。本論文所使用之資料探勘技術主要有：(一)相關規則探勘(Association Rules Mining)，(二)循序模式探勘(Sequential Patterns Mining)。

在本章中，第一節介紹資料探勘的技術，第二節介紹相關規則探勘，第三節說明相關規則探勘之延伸問題，第四節描述循序模式探勘之演算法。

第一節 資料探勘

所謂的資料探勘，簡單地說即是從儲存於資料庫(Database)、資料倉儲(Data Warehouse)或其他資訊儲存器(Information Repository)的大量資料中發掘出所感興趣的知識之處理過程。在[5]中提到，資料探勘包含下列幾個步驟：

1. 確定目標：雖然定出目標是很直覺的一件事，但在資料探勘中卻是不可缺少的一部份，所以必須明確地定義出問題所在及質疑的地方。
2. 預備資料：此步驟是資料探勘處理過程中最花時間的一個步驟，在整個資料探勘的過程中，大約佔了 60% 的時間。此步驟又分為三部份：
 - ◆ 資料選取：從所有的資訊中選擇所需要的資料。
 - ◆ 資料的前置處理：確保所選取資料的品質，使分析更容易。
 - ◆ 資料轉換：根據不同的資料探勘演算法，將資料轉換成演算法所適合的模式。
3. 資料探勘：選擇適當的資料探勘演算法來探勘上一個步驟所轉換出來的資料。
4. 分析結果：評估步驟 3 所探勘出來的資料。

5. 吸收知識：將步驟 4 所分析出來的資料加到企業化的組織或資訊系統裡。

圖 2 - 1 - 1 為整個資料探勘的流程圖。

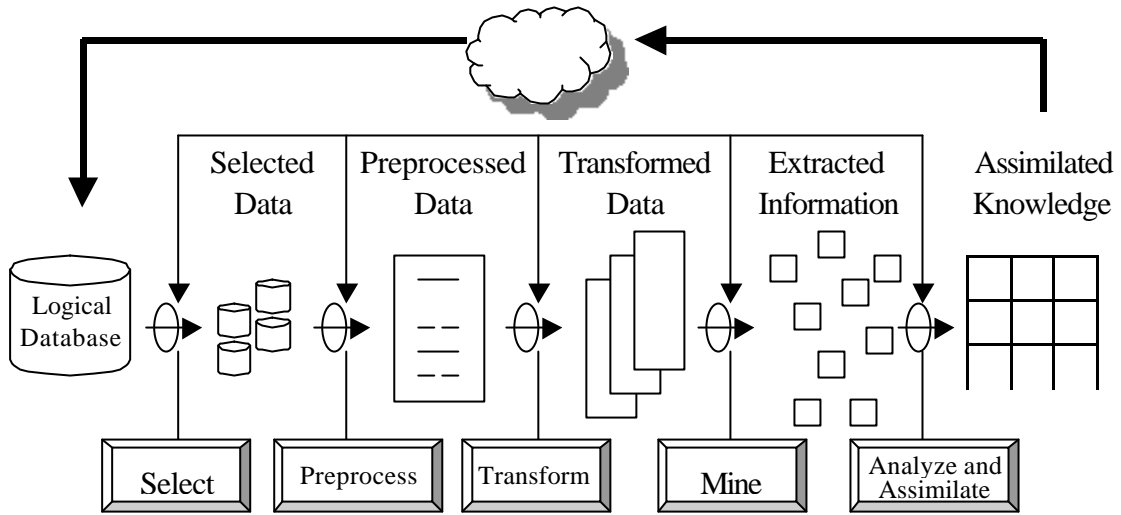


圖 2 - 1 - 1：資料探勘流程圖

現有資料探勘的研究課題很多，比較重要的包含相關規則探勘(Association Rules Mining)、循序模式探勘(Sequential Patterns Mining)、分類規則的歸納(Classification Rule)和分群(Clustering) 等[2]。

第二節 相關規則探勘

相關規則探勘問題是屬於資料探勘的範疇，最常以商店交易記錄資料庫為其應用領域，針對顧客的交易行為做分析，以增加商店的業績。其後陸續引入其他領域為研究主題應用，在不同領域時所採用的演算法和設計方式亦有些許差異。在本篇論文中，我們將相關規則探勘應用在圖書館上，探討讀者借閱的共同性問題。

2.2.1 簡介

從交易資料庫裡發掘出相關規則這個問題在[1]中首先被提出。例如：40%買牙刷的顧客也會買牙膏，這就是一個典型的相關規則。找出相關規則之後便可以採取適當的行銷策略促使顧客購買這些商品。

相關規則探勘的正規敘述 (Formal Statement) [1]如下:

令 $I = \{i_1, i_2, \dots, i_m\}$ 為一個文字符號 (Literal) 組成的集合, 每個文字符號稱為一個項目 (Item), 由一個或一個以上的項目所組成的集合稱為項目集 (Itemset)。令資料庫 D 是由一群交易 (Transaction) T 所組成的集合, 每個 T 為一個項目集, 代表交易記錄, $T \subseteq I$, 每個交易記錄有其唯一的識別碼, 稱為 TID 。如果 $X \subseteq I$ 且 $X \subseteq T$, 則定義為 T 包含 (Contain) X 。一個相關規則 (Association Rule) 表示成 $X \Rightarrow Y$, 其中 $X \subseteq I$, $Y \subseteq I$, $X \cap Y = \emptyset$ 。若 D 中包含 X 的交易裡有 $c\%$ 也同時包含了 Y , 我們就說規則 $X \Rightarrow Y$ 的確信值 (Confidence) 為 c ; 如果 D 裡包含 $X \cup Y$ 的交易記錄有 $s\%$, 我們就說規則 $X \Rightarrow Y$ 的支持度 (Support) 為 s 。相關規則探勘的問題為給定一個交易記錄資料庫 D , 在當中找出所有確信值和支持度大於最小確信值 (minconf) 跟最小支持度 (minsup) 的規則, 其中 minconf 與 minsup 的值由使用者給定。

[1]將本問題分成兩個子問題:

1. 子問題 1: 找到所有支持度大於最小支持度的項目集。為了探勘方便起見, 有時把某一項目集的支持度定義為包含此項目集的交易個數, 而不是原來的交易百分率。支持度大於最小支持度的項目集稱為大項目集 (Large Itemset), 反之稱為小項目集 (Small Itemset)。
2. 子問題 2: 用子問題 1 中所找到的大項目集來產生所期望的規則。此步驟的演算法非常直接, 即: 對於任一大項目集 L , 找出其所有非空子集合。對於每個非空子集合 a , 如果規則 $a \Rightarrow (L-a)$ 的確信值 (也就是 $\text{support}(L)/\text{support}(a)$) 大於最小確信值, 則此規則即符合所求。

由於在[2]中針對子問題 2 提出了直接且有效率的演算法, 所以通常並不對它做進一步的探討, 而是集中在探討子問題 1, 亦即如何快速地找出所有大項目集, 文獻上將此問題稱為”探勘大項目集”。

2.2.2 相關規則探勘演算法之相關研究

由於目前大多數的研究乃是以 Apriori[2]為基礎，因此接下來我們針對 Apriori 進一步地介紹。

Apriori 演算法以疊代 (Iteration) 的方式產生大項目集，每一次疊代時完整讀取資料庫並產生所有相同長度的大項目集，在第一次疊代時產生長度為 1 的大項目集，第二次時則產生長度為 2 的大項目集，依此類推。每一次產生的大項目集當作下一次疊代產生大項目集的種子集 (Seed Set)，由種子集來推論下一次疊代所有可能會出現的大項目集，文獻上稱此可能出現的大項目集為候選集 (Candidate Itemset)。每一次疊代只要將所有交易和產生之候選集加以比對並計算它們的支持度，候選集之中所有大於最小支持度的項目集所成之集合就是這一次疊代的大項目集。如此一直反覆，直到沒有新的大項目集出現為止。圖 2 - 2 - 1 是一個簡單的例子，利用 Apriori 演算法找出所有大於最小支持度 2 的項目集。圖 2 - 2 - 2 列出 Apriori 的演算法。

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

minisup=2

Itemsets	Support
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

Itemsets	Support
{1}	2
{2}	3
{3}	3
{5}	3

Itemsets	Support
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

Itemsets	Support
{1 2}	2
{2 3}	2
{2 5}	3
{3 5}	2

Itemsets	Support
{2 3 5}	2

Itemsets	Support
{2 3 5}	2

圖 2 - 2 - 1 : Apriori 例子[2]

```

L1 = {large 1-itemsets} ; //產生長度為 1 的大項目集
for (k=2;Lk-1≠∅;k++) do begin
  Ck = apriori-gen(Lk-1) ; //產生 k-項目集之候選集
  for all transactions t∈D do begin
    Ct = subset(Ck,t); //計算支持度
    for all candidates c∈Ct do
      c.count++;
  end
  Lk = {c∈Ck|c.count ≥ minsup}; //產生長度為 k 的大項目集
end
Answer = ∪k(Lk);

```

圖 2 - 2 - 2 : Apriori 演算法

圖 2 - 2 - 2 中的 apriori_gen 函數是用來產生候選集。由於任何大項目集的子集合也都是大項目集，因此 apriori_gen 以所有大 (k-1)-項目集(L_{k-1})為參數，傳回大 k-項目集的候選集(C_k)。 apriori_gen 分為兩個步驟來產生 C_k：

1. 連結(Join)：用 L_{k-1} 來連結 L_{k-1}

```

insert into Ck
select p.item1, p.item2, . . . , p.itemk-1, q.itemk-1
from Lk-1 p, Lk-1 q
where p.item1=q.item1, . . . , p.itemk-2=q.itemk-2, p.itemk-1<q.itemk-1

```

2. 刪除(Prune)：對於所有的 c∈C_k，只要任何一個子集合不在 L_{k-1} 中就刪除它

```

for all itemsets c∈Ck do
  for all (k-1)-subsets s of c do
    If (s∉Lk-1)then
      delete c from Ck;

```

圖 2 - 2 - 2 中的 subset 函數用來找出一個交易紀錄中所有包含於 C_k 的子集合。在實作上，C_k 儲存在雜湊樹 (Hash Tree) 資料結構，如圖 2 - 2 - 3 所示。將 C_k 儲存在雜湊樹使得比對候選項目集更有效率，此亦是 Apriori 重要的關鍵。

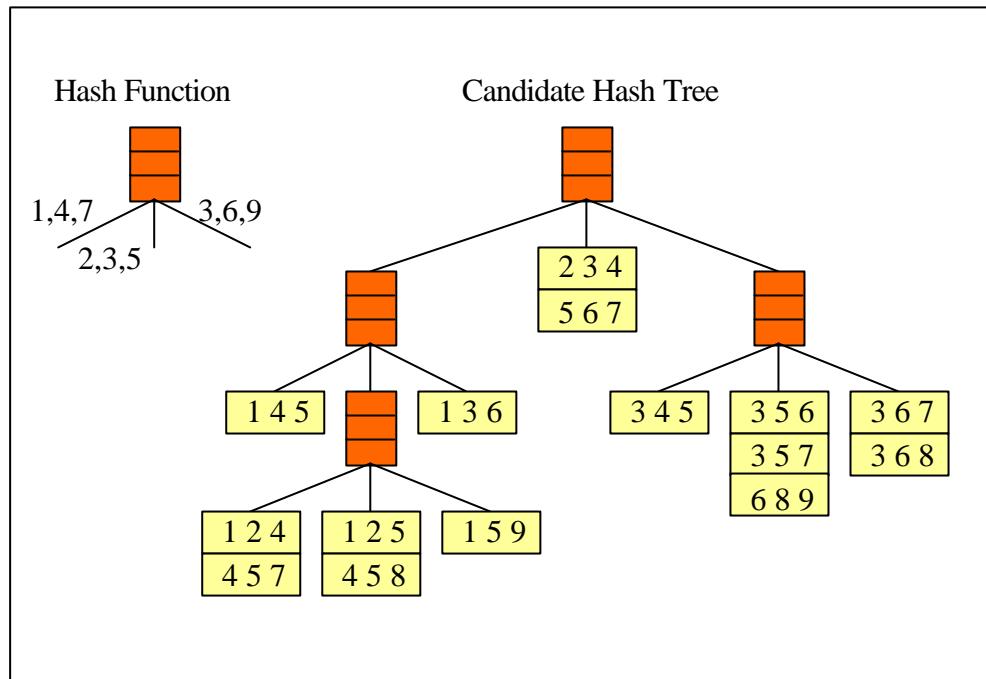


圖 2 - 2 - 3 : 雜湊樹[10]

雜湊樹的內部節點 (Interior Node) 是一個雜湊表 (Hash Table), 外部節點 (Exterior Node) 是一個陣列。雜湊樹根的深度 (Depth) 定義為 1, 深度 d 的內部節點指向深度 $d+1$ 的節點, 項目集儲存在外部節點裡。

當要新增一個項目集 c 時, 從根開始一直往下找, 找到 c 落在的外部節點裡, 然後加入。當在深度 d 時, 以項目集中第 d 個項目來執行雜湊 (Hashing) 的動作, 決定要走哪個樹枝 (Branch)。當外部節點裡儲存之項目集超過一給定的數量時, 外部節點就變成內部節點。

subset 這個函數從雜湊樹的根開始遞迴處理, 傳回所有包含於交易記錄 t 的項目集, 處理方法如下: (A) 如果正在處理的節點為外部節點, 則將該節點中包含於 t 的項目集加入答案中。(B) 如果正在處理的節點為內部節點, 且此節點是藉由雜湊項目 i 來的, 則項目 i 後面所有的項目都須執行雜湊的動作, 並將所得到的節點再遞迴, 運用上面所說的方式繼續執行。圖 2 - 2 - 4 用圖例說明 subset 的運作方式。

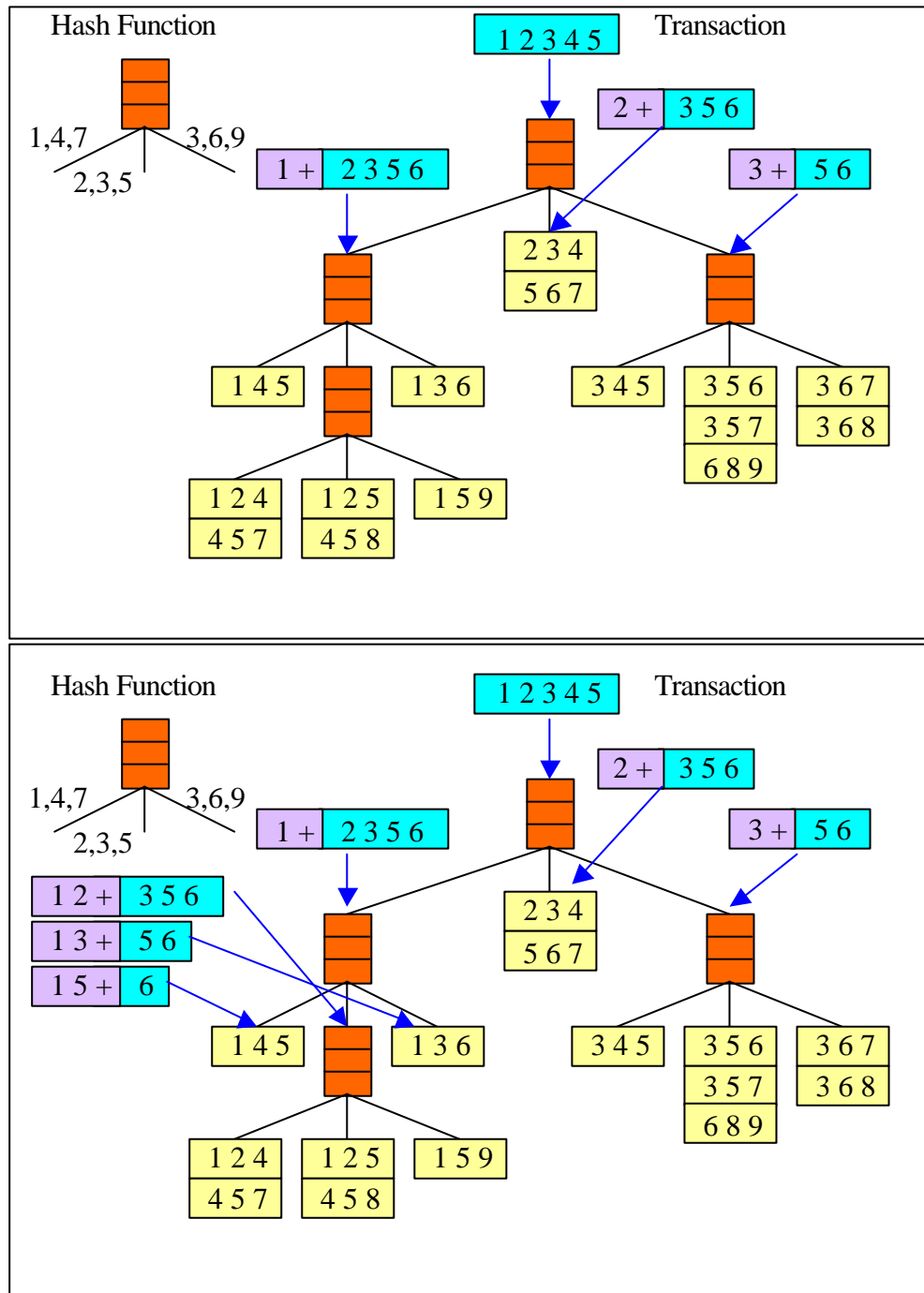


圖 2 - 2 - 4 : subset 函數[10]

在[2]中也提出 AprioriTid 演算法。AprioriTid 演算法跟 Apriori 相同，運用 apriori-gen 函數來決定下一次疊代的候選集。不同處為，Apriori 是每一次疊代都是以 D 對候選集來計算支持度，而 AprioriTid 是以 \bar{C}_k 來擔任這個工作。 \bar{C}_k 裡每個元素都是 $\langle TID, \{X_k\} \rangle$ 形式， X_k 是識別碼為 TID 的交易記錄中所有可能出現的

大項目集，此大項目集長度為 k 。如圖 2 - 2 - 5 的 \bar{C}_2 ，當 TID=200 時，其交易記錄為 {2 3 5}，所有可能出現長度為 2 的大項目集為 {2 3}, {2 5}, {3 5}，因此 $\{X_k\} = \{\{2 3\}, \{2 5\}, \{3 5\}\}$ 。圖 2 - 2 - 5 為 AprioriTid 的一個例子。當 $k=1$ 時 \bar{C}_k 與 D 相同，只不過每個項目 i 變成項目集 $\{i\}$ 。當 $k>1$ 時，對於所有的交易記錄 $t \in D$ ， $\langle t.TID, \{c \in C_k \mid c \subset t\} \rangle \in \bar{C}_k$ ，如果有一個交易記錄沒有包含任何候選 k -項目集，那麼 \bar{C}_k 裡將不會有這個交易記錄衍生出來的記錄。例如圖 2 - 2 - 5，TID=100 這筆交易不包含 C_3 的任何一個項目集，所以 \bar{C}_3 中不會有 TID=100 這筆交易。由此可知在 k 值大的時候 \bar{C}_k 中的項目集數目會比 D 小很多。而在 k 值小的時候，由於項目集的數目多，且 \bar{C}_k 是所有被這筆交易記錄所包含的 k -項目集的集合，所以 \bar{C}_k 裡的每一筆記錄都有可能比 D 裡的一筆交易記錄大。因此在探勘的初期 AprioriTid 會比 Apriori 慢，但是到後期因為 \bar{C}_k 的記錄變少變短，所以 AprioriTid 就比 Apriori 快許多。有鑑於此，作者提出了一個前幾次疊代用 Apriori，後幾次疊代用 AprioriTid 的演算法，稱為 AprioriHybrid。

Database		\bar{C}_1	L_1																																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th>TID</th><th>Items</th></tr> </thead> <tbody> <tr><td>100</td><td>1 3 4</td></tr> <tr><td>200</td><td>2 3 5</td></tr> <tr><td>300</td><td>1 2 3 5</td></tr> <tr><td>400</td><td>2 5</td></tr> </tbody> </table>	TID	Items	100	1 3 4	200	2 3 5	300	1 2 3 5	400	2 5	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th>TID</th><th>Set-of-Itemsets</th></tr> </thead> <tbody> <tr><td>100</td><td>{ {1}, {3}, {4} }</td></tr> <tr><td>200</td><td>{ {2}, {3}, {5} }</td></tr> <tr><td>300</td><td>{ {1}, {2}, {3}, {5} }</td></tr> <tr><td>400</td><td>{ {2}, {5} }</td></tr> </tbody> </table>	TID	Set-of-Itemsets	100	{ {1}, {3}, {4} }	200	{ {2}, {3}, {5} }	300	{ {1}, {2}, {3}, {5} }	400	{ {2}, {5} }	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th>Itemsets</th><th>Support</th></tr> </thead> <tbody> <tr><td>{1}</td><td>2</td></tr> <tr><td>{2}</td><td>3</td></tr> <tr><td>{3}</td><td>3</td></tr> <tr><td>{5}</td><td>3</td></tr> </tbody> </table>	Itemsets	Support	{1}	2	{2}	3	{3}	3	{5}	3					
TID	Items																																				
100	1 3 4																																				
200	2 3 5																																				
300	1 2 3 5																																				
400	2 5																																				
TID	Set-of-Itemsets																																				
100	{ {1}, {3}, {4} }																																				
200	{ {2}, {3}, {5} }																																				
300	{ {1}, {2}, {3}, {5} }																																				
400	{ {2}, {5} }																																				
Itemsets	Support																																				
{1}	2																																				
{2}	3																																				
{3}	3																																				
{5}	3																																				
C_2	\bar{C}_2	L_2																																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th>Itemsets</th><th>Support</th></tr> </thead> <tbody> <tr><td>{1 2}</td><td>1</td></tr> <tr><td>{1 3}</td><td>2</td></tr> <tr><td>{1 5}</td><td>1</td></tr> <tr><td>{2 3}</td><td>2</td></tr> <tr><td>{2 5}</td><td>3</td></tr> <tr><td>{3 5}</td><td>2</td></tr> </tbody> </table>	Itemsets	Support	{1 2}	1	{1 3}	2	{1 5}	1	{2 3}	2	{2 5}	3	{3 5}	2	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th>TID</th><th>Set-of-Itemsets</th></tr> </thead> <tbody> <tr><td>100</td><td>{ {1 3} }</td></tr> <tr><td>200</td><td>{ {2 3}, {2 5}, {3 5} }</td></tr> <tr><td>300</td><td>{ {1 2}, {1 3}, {1 5}, {2 3}, {2 5}, {3 5} }</td></tr> <tr><td>400</td><td>{ {2 5} }</td></tr> </tbody> </table>	TID	Set-of-Itemsets	100	{ {1 3} }	200	{ {2 3}, {2 5}, {3 5} }	300	{ {1 2}, {1 3}, {1 5}, {2 3}, {2 5}, {3 5} }	400	{ {2 5} }	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th>Itemsets</th><th>Support</th></tr> </thead> <tbody> <tr><td>{1 3}</td><td>2</td></tr> <tr><td>{2 3}</td><td>2</td></tr> <tr><td>{2 5}</td><td>3</td></tr> <tr><td>{3 5}</td><td>2</td></tr> </tbody> </table>	Itemsets	Support	{1 3}	2	{2 3}	2	{2 5}	3	{3 5}	2	
Itemsets	Support																																				
{1 2}	1																																				
{1 3}	2																																				
{1 5}	1																																				
{2 3}	2																																				
{2 5}	3																																				
{3 5}	2																																				
TID	Set-of-Itemsets																																				
100	{ {1 3} }																																				
200	{ {2 3}, {2 5}, {3 5} }																																				
300	{ {1 2}, {1 3}, {1 5}, {2 3}, {2 5}, {3 5} }																																				
400	{ {2 5} }																																				
Itemsets	Support																																				
{1 3}	2																																				
{2 3}	2																																				
{2 5}	3																																				
{3 5}	2																																				
C_3	\bar{C}_3	L_3																																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th>Itemsets</th><th>Support</th></tr> </thead> <tbody> <tr><td>{2 3 5}</td><td>2</td></tr> </tbody> </table>	Itemsets	Support	{2 3 5}	2	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th>TID</th><th>Set-of-Itemsets</th></tr> </thead> <tbody> <tr><td>200</td><td>{ {2 3 5} }</td></tr> <tr><td>300</td><td>{ {2 3 5} }</td></tr> </tbody> </table>	TID	Set-of-Itemsets	200	{ {2 3 5} }	300	{ {2 3 5} }	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr><th>Itemsets</th><th>Support</th></tr> </thead> <tbody> <tr><td>{2 3 5}</td><td>2</td></tr> </tbody> </table>	Itemsets	Support	{2 3 5}	2																					
Itemsets	Support																																				
{2 3 5}	2																																				
TID	Set-of-Itemsets																																				
200	{ {2 3 5} }																																				
300	{ {2 3 5} }																																				
Itemsets	Support																																				
{2 3 5}	2																																				

圖 2 - 2 - 5 : TidApriori 例子[2]

綜合以上的說明，我們可以了解：Apriori 演算法減少候選集的數目，並利用雜湊樹使得交易和候選集的比對時間較之前的演算法減少許多，缺點為資料庫搜尋次數仍然太多；AprioriTid 改進 Apriori 的缺點，在探勘後期減小資料庫，以降低磁碟讀取增加速度，但在初期因為項目集較多的關係會較慢。

第三節 相關規則探勘之延伸問題

相關規則探勘問題延伸出各式各樣的問題，其解決方法和基本的問題息息相關，且可以應用在更多各個不同的領域。以下茲列出目前的相關研究[10]：

1. 有些研究考慮到顧客多次交易的狀況，同一位顧客會有先後時間關係的交易記錄。此類研究的目的是想要在這些交易記錄中找出顧客最常買的商品序列(Sequence)，此序列是由項目集所組成，而每一個項目集內的項目是不考慮個數且沒有順序關係的[3][12][14]。
2. 支持度和確信值是用來控制產生規則多寡的兩個限制，除了這二個限制外，有些研究還定義更多的限制，繼而利用這些限制的特性來加速探勘[4][14]。
3. 在實際的應用環境中，資料庫內的交易記錄筆數會隨著時間而增加或減少，而資料探勘要耗費龐大的計算時間，若是每次資料庫一更新就重新探勘將會浪費很多的時間，相對地若只探勘資料庫更新的部份便可節省很多時間，因此有些研究希望只探勘資料庫增加或減少的部份就能得到所有的規則，此即漸進更新(Incremental Update)問題[7][9][11]。
4. 有些研究除了考慮交易記錄之間的順序關係外，也探討交易記錄之項目集內的項目順序關係[6]。
5. 有些研究將交易記錄中每一個交易項目的個數考慮進去[13]，或者將交易項目分類，以分類來當相關規則的項目[8][14]。

表 2 - 3 - 1 整理所有的研究方向，大致可依據項目、交易記錄和規則本身的

性質來區分成六大類。

交易記錄 +順序	找出同一使用者在不同交易記錄間循序購買的大 項目集(Sequential Pattern) [3][12][14]
規則+限制	用兩個以上的限制條件來探勘規則 [4][14]
規則+更新	資料庫變動時，僅針對變動的部分探勘，來更新既 有的規則[7][9][11]
規則+順序	有序交易記錄之探勘[6]
項目+數量	探勘的結果為項目名稱和其數量的對(Pair) [13]
項目+分類	探勘的結果可以是項目所屬的種類[8][14]

表 2 - 3 - 1：各種類型相關規則探勘之延伸問題[10]

第四節 循序模式探勘

本論文在探索讀者借閱的順序性方面與[3][12]同屬循序模式探勘，所以在此將這兩個演算法稍做整理介紹。

AprioriAll[3]屬於循序模式 (Sequential Pattern) 探勘之演算法，即同一使用者在不同時間點有許多交易記錄，這些交易形成一個序列，每一序列為項目集所形成的有序集(Order Set)，AprioriAll 的目的是要在這些交易序列中找出包含次數大於最小支持度的序列，在此所謂「包含」的定義為：如果存在 $i_1 < i_2 < \dots < i_n$ 及二個序列 $\langle a_1 a_2 \dots a_n \rangle$ 及 $\langle b_1 b_2 \dots b_n \rangle$ ，使得 $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \dots, a_n \subseteq b_{i_n}$ ，則序列 $\langle b_1 b_2 \dots b_n \rangle$ 包含 $\langle a_1 a_2 \dots a_n \rangle$ 。例如，序列 $\langle (7)(3\ 8)(9)(4\ 5\ 6)(8) \rangle$ 包含序列 $\langle (3)(4\ 5)(8) \rangle$ ，因為 $(3) \subseteq (3\ 8)$ 、 $(4\ 5) \subseteq (4\ 5\ 6)$ 且 $(8) \subseteq (8)$ ，然而 $\langle (3)(5) \rangle$ 並不包含 $\langle (3\ 5) \rangle$ ，反之， $\langle (3\ 5) \rangle$ 亦不包含 $\langle (3)(5) \rangle$ 。

以下舉例說明 AprioriAll 演算法。圖 2 - 4 - 1(a)為一以使用者編號為索引的交易序列資料庫，每個序列裡的項目集為此使用者每次交易的物品，亦即該次的交易記錄，例如使用者 2 曾交易三次，此三次交易分別為(10 20)，(30)，(40 50 70)。

現欲找出被大於 25% 的交易序列包含之序列，作法如下：首先，用 Apriori 演算法在資料庫中找出所有長度為 1 的大項目集，在 [2] 中，一個項目集的支持度 (Support) 定義為此項目集被交易包含的比例，在此，一個項目集的支持度則定義為所有使用者在許多交易中有任一交易包含此項目集的比例，圖 2 - 4 - 1(b) 為找到的大項目集，相當於長度為 1 的序列。接著將交易序列資料庫轉換成以這些大項目集來表示，並將每個大項目集對應到一個數字，圖 2 - 4 - 2 描述此步驟。

接下來，再以 AprioriAll 演算法來探勘出所有的大序列。AprioriAll 和 Apriori 大部分都相同，唯一不同的是它將 Apriori 產生候選集時連結 (Join) 的步驟更改成為有序性，如圖 2 - 4 - 3 所示。圖 2 - 4 - 4 為 L3 產生 C4 的過程。

Customer Id	Customer Sequence	Large Itemsets	Mapped To
1	<(30)(90)>	(30)	1
2	<(10 20)(30)(40 50 70)>	(40)	2
3	<(30 50 70)>	(70)	3
4	<(30)(40 70)(90)>	(40 70)	4
5	<(90)>	(90)	5

圖 2 - 4 - 1(a)：交易序列 [3]

圖 2 - 4 - 1(b)：大 1-序列對應編號 [3]

Customer Id	Original Customer Sequence	Transformed Customer Sequence	After Mapping
1	<(30)(90)>	<{(30)}{(90)}>	<{1}{5}>
2	<(10 20)(30)(40 50 70)>	<{(30)}{(40)(70)(40 70)}>	<{1}{2 3 4}>
3	<(30 50 70)>	<{(30)(70)}>	<{1 2}>
4	<(30)(40 70)(90)>	<{(30)}{(40)(70)(40 70)}{(90)}>	<{1}{2 3 4}{5}>
5	<(90)>	<{(90)}>	<{5}>

圖 2 - 4 - 2：轉換資料庫 [3]

```

insert into Ck
select p.itemset1, .., p.itemsetk-1, q.itemsetk-1
from Lk-1 p, Lk-1 q
where p.itemset1=q.itemset1, .., p.itemsetk-2=q.itemsetk-2;

```

圖 2 - 4 - 3：AprioriAll 產生候集時連結的步驟

Large 3-Sequence	Candidate 4-Sequences (after join)	Candidate 4-Sequences (after pruning)
<1 2 3>	<1 2 3 4>	<1 2 3 4>
<1 2 4>	<1 2 4 3>	
<1 3 4>	<1 3 4 5>	
<1 3 5>	<1 3 5 4>	
<2 3 4>		

圖 2 - 4 - 4 : AprioriAll L_3 產生 $C_4[3]$

因為 AprioriAll 沒有考慮到時間的限制及分類，所以在[12]中提出了一個新的演算法，稱為 GSP (Generalized Sequential Patterns)。

GSP 加入了三個條件：

1. 時間限制 (Time Constraints): 使用者通常會希望限制一個順序模式裡，二個相鄰項目間有最大和(或)最小時間間隔。例如：以一個書商而言，如果有一位客戶今天買了一本書，三年後才又買了另一本書，書商會覺得這位讀者買這兩本書的時間間隔太久，這兩本書應該是沒什麼相關性的，所以書商對這樣的順序模式不感興趣。
2. 滑動的時間視窗 (Sliding Time Window): 在很多應用中，短時間內出現在兩筆不同交易中的項目可能是有相關性的，所以可以將這兩個項目做聯集加到順序模式裡視為同一筆交易。例如：有一位客戶今天訂了“Foundation”，明天訂了“Ringworld”，隔了一週又訂了“Foundation and Empire”和“Ringworld Engineers”，則可以將“Foundation”和“Ringworld”聯集起來，將 (Foundation, Ringworld) (Foundation and Empire, Ringworld Engineers) 加到順序模式裡。
3. 分類：很多資料庫都有對項目做分類，使用者如果想要得到不同層次間的模式，則需加入分類。例如：給定一個分類圖如圖 2 - 4 - 5，若我們想知道讀者借閱某一類別或某位作者的館藏後會借哪些書，則必須將類

別及作者也加入交易資料裡。舉例來說：如果有一位讀者借了“Foundation”及“Perfect Spy”二本館藏，其交易序列為 (Foundation)(Perfect Spy)，則我們必須將類別“Science Fiction”、“Spy”及作者“Asimov”、“Le Carre”也加入序列中，使序列擴充為 (Foundation, Asimov, Science Fiction)(Perfect Spy, Le Carre, Spy)，如此一來我們才能探勘出類別、作者及書目之間的關聯性。

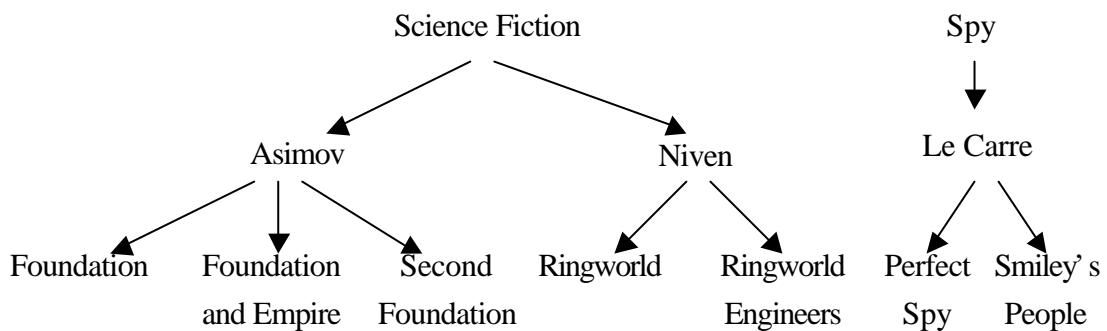


圖 2 - 4 - 5：分類圖[12]

GSP 演算法如下：

首先，GSP 和 AprioriAll 一樣，在資料庫中找出所有長度為 1 的大項目集，相當於長度為 1 的序列。接下來再依序產生候選集並計算候選集是否大於最小支持度。

GSP 分為兩個步驟來產生候選集 C_k

1. 連結(Join)：

用 L_{k-1} 來連結 L_{k-1} 。假設 C_{k-1} 中有兩個序列 S_1 和 S_2 ，如果 S_1 去掉第一個項目和 S_2 去掉最後一個項目後所得的序列相同，則將 S_1 和 S_2 連結起來成為 C_k 的序列，此序列即為 S_1 加上 S_2 的最後一個項目。如：

(1,2) (3) 和 (2) (3,4) 做連結後會產生 (1,2) (3,4)。

2. 刪除(Prune)：

對於所有的 $c \in C_k$ ，只要任何一個子集合不是連續的 $(k-1)$ -子序列或是連續的 $(k-1)$ -子序列小於最小支持度就刪除它。圖 2 - 4 - 6 即為 L_3 產生 C_4 的例子。

連續的子序列之定義為：給定一個序列 $s = \langle s_1 s_2 \dots s_n \rangle$ 和一個子序列 c ，如果下面任何一條件成立，則稱 c 為 s 的連續的子序列：

1. c 由 s 去掉第一個項目集之任何一個項目或最後一個項目集之任何一個項目所產生。

例如： $s = \langle (1,2)(3,4)(5)(6) \rangle$ ，則 $c = \langle (1)(3,4)(5)(6) \rangle$ 、 $\langle (2)(3,4)(5)(6) \rangle$ 或 $\langle (1,2)(3,4)(5) \rangle$ 。

2. c 由 s 中任何一個大於兩個項目的元素 s_i 去掉一個項目所產生。

例如： $s = \langle (1,2)(3,4)(5)(6) \rangle$ ，則 $c = \langle (1,2)(3)(5)(6) \rangle$ 或 $\langle (1,2)(4)(5)(6) \rangle$ 。

3. c 是 c' 的連續子序列，而且 c' 是 s 的連續子序列。

例如： $s = \langle (1,2)(3,4)(5)(6) \rangle$ ， $c' = \langle (2)(3,4)(5)(6) \rangle$ ，則 $c = \langle (2)(3,4)(5) \rangle$ 即是其中一個情況。

綜合上面三個情形，若 $s = \langle (1,2)(3,4)(5)(6) \rangle$ ，則 $\langle (2)(3,4)(5)(6) \rangle$ 、 $\langle (2)(3,4)(5) \rangle$ 、 $\langle (1,2)(3)(5)(6) \rangle$ 及 $\langle (3)(5) \rangle$ 等都是 s 的連續的子序列，但是 $\langle (1,2)(3,4)(6) \rangle$ 和 $\langle (1)(5)(6) \rangle$ 則不是。

Frequent 3-Sequence	Candidate 4-Sequences (after join)	Candidate 4-Sequences (after pruning)
$\langle (1, 2)(3) \rangle$	$\langle (1, 2)(3, 4) \rangle$	$\langle (1, 2)(3, 4) \rangle$
$\langle (1, 2)(4) \rangle$	$\langle (1, 2)(3)(5) \rangle$	
$\langle (1)(3, 4) \rangle$		
$\langle (1, 3)(5) \rangle$		
$\langle (2)(3, 4) \rangle$		
$\langle (2)(3)(5) \rangle$		

圖 2 - 4 - 6：GSP L_3 產生 C_4 [12]

GSP 在計算候選集是否大於最小支持度時，加入了時間限制和滑動的時間視窗，我們以一個簡單的例子來看：

假設最大的時間間隔為 30，最小的時間間隔為 5，滑動的時間視窗為 0，我們要找候選序列 (1,2)(3)(4)，圖 2 - 4 - 7 為資料序列，則找候選序列 (1,2)(3)(4) 的步驟如下：

首先要找(1,2)，我們發現(1,2)出現在時間為 10 的地方，接下來要找(3)，(3)出現在時間為 45 的地方，時間間隔為 $35(= 45-10)$ ，超過我們設定的 30，所以從時間 $15(= 45-30)$ 後開始尋找(1,2)，我們發現(1,2)又出現在 50 的地方，超過(3)出現的時間點 45，所以從 $55(= 50+5)$ 之後開始找(3)，(3)出現在 65，符合我們所設定的最大及最小時間間隔，接下來找(4)，我們發現(4)出現在時間為 90 的地方，也符合我們所設定的時間間隔，因此 (1,2)(3)(4) 存在資料序列中。

Transaction-Time	Items
10	1, 2
25	4, 6
45	3
50	1, 2
65	3
90	2, 4
95	6

圖 2 - 4 - 7：Data-Sequence

此外，GSP 在找單一個項目集時，加入滑動的時間視窗，並產生一個陣列儲存各個項目的時間。以下面的例子來看：

假設滑動的時間視窗為 7，圖 2 - 4 - 7 為資料序列，圖 2 - 4 - 8 為各個項目的時間陣列，則找出現時間為 20 之後項目集(2,6)的步驟如下：

首先要找 2，我們發現在時間 20 後 2 出現在時間為 50 的地方，接下來要找 6，6 出現在時間為 25 的地方，時間間隔為 25，超過我們設定的 7，所以從時間 $43(= 50-7)$ 後開始尋找，2 維持在時間為 50 的地方，6 出現在 95 的地方，超過我們設定的 7，所以從 $88(= 95-7)$ 之後開始找，2 出現在 90，6 維持在 95，符合我

們所設定的時間視窗，因此找到出現時間在 20 之後的項目集(2,6)。

Items	Times
1	→ 10 → 50 → NULL
2	→ 10 → 50 → 90 → NULL
3	→ 45 → 65 → NULL
4	→ 25 → 90 → NULL
5	→ NULL
6	→ 25 → 95 → NULL
7	→ NULL

圖 2 - 4 - 8：各個項目的時間陣列

第三章 資料探勘在圖書館上的應用

本論文中，我們採取現有資料探勘的演算法，將之應用在圖書館之電子化服務上，期望能夠找出讀者借閱館藏之社群性，並運用此特性提昇圖書館的經營與服務。本論文中採取的演算法為 Apriori 及 GSP 。

第一節 目標

本論文應用資訊科技及企業化的模式來經營圖書館，以交通大學圖書館的書目、館藏、交易歷史記錄為基礎，運用資料探勘的技術探索交通大學讀者的社群特性，並運用資料探勘的成果來提昇圖書館的經營與服務，期能使交通大學圖書館在交通大學讀者學術、知識的吸收和產出中扮演更積極的角色。本論文期望達成的目標包含：

1. 探索讀者社群關係 -- 本論文擬探索的讀者社群關係包含：
 - 甲、館藏借閱的共同性：具有相同興趣的讀者們所借的書常是類似的，將館藏借閱的相關性找出來，當下次有某位讀者借閱一本書時，我們便可以將同樣借過這本書的讀者常常借閱的書籍推薦給這位讀者，如此一來能節省讀者找資料的時間，二來也可以減少讀者因不熟悉館藏查詢系統之功能而在搜尋時沒找到這些相關書籍的機率。
 - 乙、館藏借閱的順序：讀者借閱館藏可能會先借入門的再借深入的，如果把讀者借閱館藏的順序特性找出來，則下次有某位讀者借入門的館藏時，即可推薦他借閱進階的館藏，讓讀者很容易地知道這本館藏的進階書籍有哪些。
2. 吸引讀者到館借閱：我們發現很多讀者從未借過館藏，要如何才能增加來借閱的人數呢？當我們發掘出讀者的社群關係後，便可得知讀者最常

借閱的書，這些館藏往往是大多數人都感興趣的，若我們能適時地建議讀者借閱這些熱門的館藏，讀者感興趣便會來借閱，如此一來即可增加借閱館藏的讀者人數。

3. 提昇館藏借閱率：我們發現有很多館藏是未曾被借閱過的，要如何才能把這些館藏推銷出去呢？因為讀者往往對相同主題的館藏感興趣，而在圖書館館藏編目過程中已經為每本書登錄其分類了。如果我們能夠在讀者借閱一本書時推薦這位讀者借閱相同類別的館藏，一來能使讀者能減少找相同類別資料的時間，二來也能增加讀者借閱這些館藏的機會。
4. 提昇讀者忠誠度：從統計資料中，除了發現有部份讀者未曾借過館藏之外，我們也發現許多讀者僅借閱過一、二次館藏。圖書館的經營就如同企業經營一般，除了吸引新顧客(即讀者)，還得想辦法留住老顧客；對於那些只來過一、二次的讀者，更需給予其繼續前來的動力。為此，我們利用資料探勘的技術發掘出讀者的社群關係，運用這些社群關係適時地建議讀者借閱其他相關的館藏，增加讀者繼續借閱館藏的機率，如此一來便可提昇讀者忠誠度，使讀者不會只借閱一、二次就不再來借閱。
5. 協助館藏複本的採訪政策：圖書館針對一本書所採購的複本數往往有限，但有些熱門書讀者常常要預約很久才能借到，有些讀者因此放棄借閱，所以有些時候圖書館不得已還是得購置複本，以滿足讀者借閱和教學研究的需求。但是，要如何判斷是否該購置複本呢？由於圖書館中的館藏如果有其他讀者預約，則讀者不能續借，必須先歸還館藏後再預約及借閱，所以我們推論：若有很多讀者重複借閱(同位讀者借閱同一本館藏二次或二次以上)或重複預約某一本館藏，則這些館藏可能需添購複本，因此我們協助館藏複本採訪政策的處理過程為：先運用資料探勘的技術探索出讀者重複借閱及預約的特性，再藉由圖書館中的借閱與預約記錄查詢這些館藏是否有很多人預約，最後查詢這些館藏是否持續有讀

者在借閱及預約，若是持續有讀者在借閱及預約，則這些書便是圖書館可以考慮添購複本的對象。

6. 促進館藏流通率：很多熱門的館藏借出後往往會到應還日才歸還，其他讀者要借閱便要預約很久才能借到，很多讀者因此而放棄借閱，所以我們統計出學期初熱門預約書，並縮短這些館藏的借閱期限，以增加這些館藏的流通率，讓更多讀者可以借閱這些熱門的預約書。

第二節 方法

前一節我們提出了所期望達成的目標，在本節中我們詳細說明所採用的方法。

3.2.1 準備工作

在本論文第二章中有提到：完整的資料探勘處理過程中，預備資料是最耗時的，預備資料對於資料探勘的結果是一個很大的影響因素。我們的預備資料如下：

1. 資料來源：交通大學圖書館的書目、館藏及交易歷史資料。書目、館藏資料包含 405,785 本圖書及 526,868 本期刊，共有約 93 萬冊的資料，其中期刊及碩博士論文等是不可外借的，因此交易歷史記錄裡沒有這些資料，而可外借的圖書不含複本共有 282,250 本。
2. 資料選取：由於我們取得資料時，交通大學交易歷史資料的記錄是從 1998 年 1 月 1 日至 2000 年 8 月 31 日，因此我們所選取的資料為交通大學圖書館 1998 年 1 月 1 日至 2000 年 8 月 31 日借閱及預約的交易歷史資料，裡面共包含 487,786 筆資料，其中包含圖書、影片及光碟等，而圖書資料共有 441,688 筆。
3. 資料的前置處理及轉換：由於本論文中所要探索的讀者社群關係有二種，因此，針對此二種不同的探勘在此步驟也有不同處理方式。

- ◆ 館藏借閱的共同性：由於此社群關係是要找出讀者常一起借閱的館藏，因此我們將讀者借閱及預約的每一本書之書目號視為一個項目(Item)，而每位讀者於一段時間內(如一學期或一學年)所借閱及預約的書目所成之集合即為一筆交易(Transaction)，資料庫 D 便是由這一群交易所組成的集合。舉例來說，若圖書館中有二位讀者“ A ”及“ B ”，A 在一段時間內借了“ 1343 ”及“ 253 ”這二本書，B 在一段時間內借了“ 3423 ”、“ 34636 ”及“ 9689 ”三本書，資料庫中則會有 {1343, 253} 和 {3423, 34636, 9689} 二筆交易。
- ◆ 館藏借閱的順序性：由於此社群關係在於找出讀者借閱某些書的順序性，因此我們先將讀者借閱及預約圖書的資料依時間排序，並將讀者借閱及預約的每一本書之書目號視為一個項目(Item)，同時間所借閱及預約的書目所成之集合即為項目集(Itemset)，則同一位讀者所借閱及預約的項目集依時間排序而成的序列即為讀者的交易序列(Sequence)。例如：圖書館中有一位讀者“ A ”在 10 月 1 日借了“ 3425 ”一本書，在 10 月 15 日又借了“ 9823 ”及“ 4875 ”，則其交易序列即為 { (3425) (9823, 4875) }。若將每位讀者所借過的書當成一筆交易序列，則交通大學圖書館交易歷史資料中 1998 年 1 月 1 日至 2000 年 8 月 31 日共有 14,652 筆交易序列，其中包含圖書、影片及光碟等，而圖書的資料共有 14,450 筆。

3.2.2 探索讀者社群關係

有了3.2.1的資料後，我們將交通大學圖書館的所有讀者當成一個大社群，選擇適當的資料探勘演算法來探索讀者的社群關係。本論文擬探索的讀者社群關係包含：

1. 館藏借閱的共同性：採用 Apriori 的方法在交通大學圖書館書目、館藏、

交易歷史資料中探勘，找出讀者借閱館藏的共同性。

2. 館藏借閱的順序：改進部份 GSP 的方法在交通大學圖書館書目、館藏、交易歷史資料中探勘，找出讀者借閱館藏的順序。

圖 3 - 2 - 1 為利用資料探勘的方法找出讀者特性的流程圖。

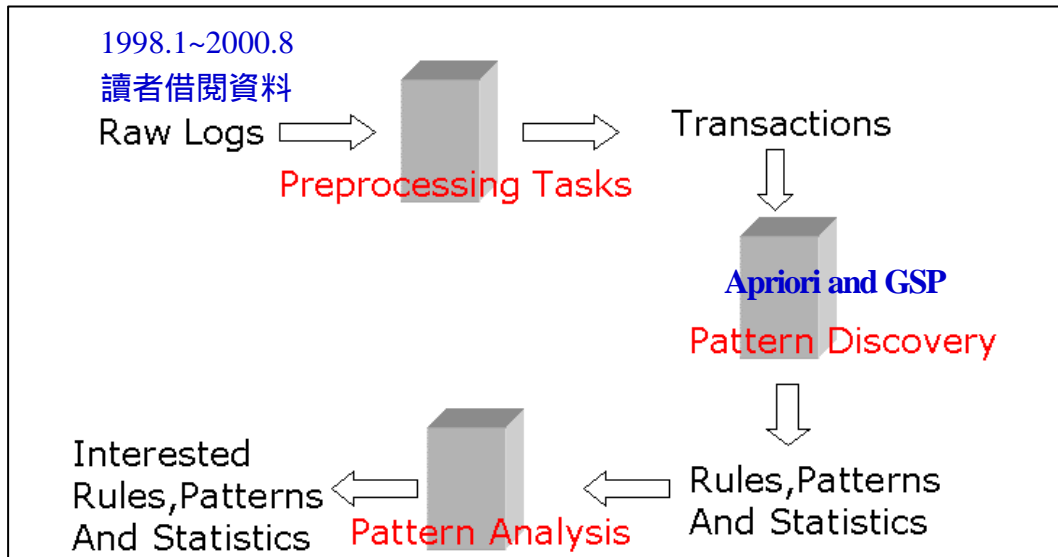


圖 3 - 2 - 1：找出讀者特性

在第二章第三節中，我們介紹了 GSP 演算法。在 GSP 演算法中為了加快探勘的速度而大量地使用記憶體，但由於交通大學圖書館可外借的館藏不計複本共有 282,250 本之多，使得我們在做探勘時有記憶體嚴重不足之現象，因此本論文根據圖書館館藏流通的特性，修改 GSP 演算法，大幅減少記憶體的用量，方得以順利地探勘出讀者借閱館藏的順序性。

本論文與 GSP 不同的地方如下：

- ◆ GSP 在找單一個項目集時，加入滑動的時間視窗，並針對每一位使用者產生個別的陣列儲存各個項目的時間，例如：假設圖書館中共有 8 本書，有一位讀者在時間點為 10 及 50 時借閱過編號 1 的館藏，在時間點為 45 及 65 時借閱過編號 4 的館藏，在時間點為 25 時借閱過編號為 6 的館藏，則其項目時間的陣列儲存如圖 3 - 2 - 2。

Items	Times
1	→ 10 → 50 → NULL
2	→ NULL
3	→ NULL
4	→ 45 → 65 → NULL
5	→ NULL
6	→ 25 → NULL
7	→ NULL
8	→ NULL

圖 3 - 2 - 2 : GSP 中儲存各個項目的時間陣列

- ◆ 由於交通大學圖書館中可外借的館藏不含複本有 282,250 本，但約有一半(51.92%)的讀者只借過少於 20 本館藏，借閱最多的讀者也不超過 600 本，若我們僅將讀者有借閱過之館藏和時間記錄到陣列中，則每一位讀者借閱館藏的時間陣列便可由原本 282,250 個項目減少為讀者借閱過之館藏數(1 ~ 600)。以圖 3 - 2 - 2 的例子來看，時間陣列的項目便由原本的 8 個減為 3 個。基於此一概念，本論文將陣列中時間為空集合之項目去除，並產生一個指標，將不為空集合之項目串連起來，如圖 3 - 2 - 3。

Items	Times
1	→ 10 → 50 → NULL
	↓
4	→ 45 → 65 → NULL
	↓
6	→ 25 → NULL
	↓
	NULL

圖 3 - 2 - 3 : 本論文中儲存各個項目的時間陣列

3.2.3 吸引讀者到館借閱

由於圖書館中的熱門書往往是大部份讀者均感興趣的館藏，因此，為了要吸引讀者到館借閱，減少未曾借閱館藏的讀者人數，我們從圖書館的交易歷史記錄中找出最熱門的二十本館藏，將這些館藏推薦給從未借閱館藏的讀者，若讀者對這些館藏感興趣便會來借閱，如此一來即可增加來借閱的讀者人數。

圖 3 - 2 - 4 為吸引讀者到館借閱的流程圖。

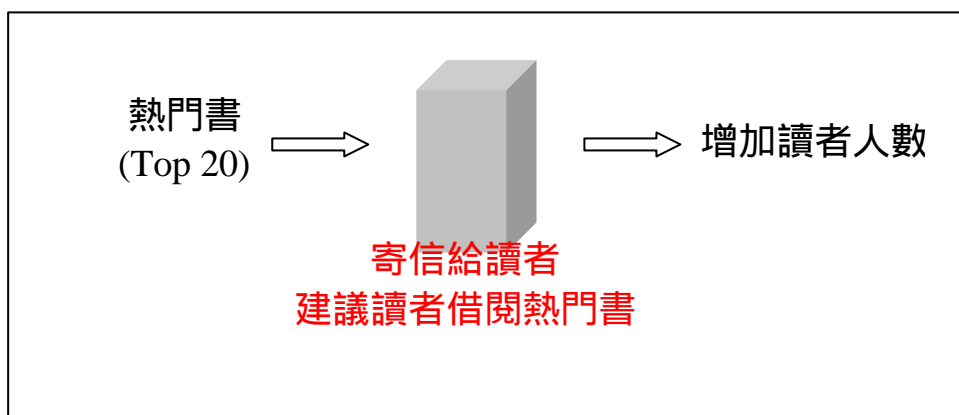


圖 3 - 2 - 4：吸引讀者到館借閱

3.2.4 提昇館藏借閱率

當有新進的館藏時，館員會針對每一本館藏的特性，依”中國圖書分類法“或“美國國會圖書分類法”編目，並給定幾個主題來做分類。為了要提昇館藏的借閱率，我們將館藏經由”中國圖書分類法“、“美國國會圖書分類法”及“主題”做分類，將同一類的館藏加到相關書籍中，當下次有某位讀者借閱一本書時，我們便可以推薦這位讀者借閱相關的館藏，藉此增加讀者借閱這些館藏的機會，將從未被借閱過的館藏推銷出去，提昇館藏借閱率。

圖 3 - 2 - 5 為提昇館藏借閱率的流程圖。

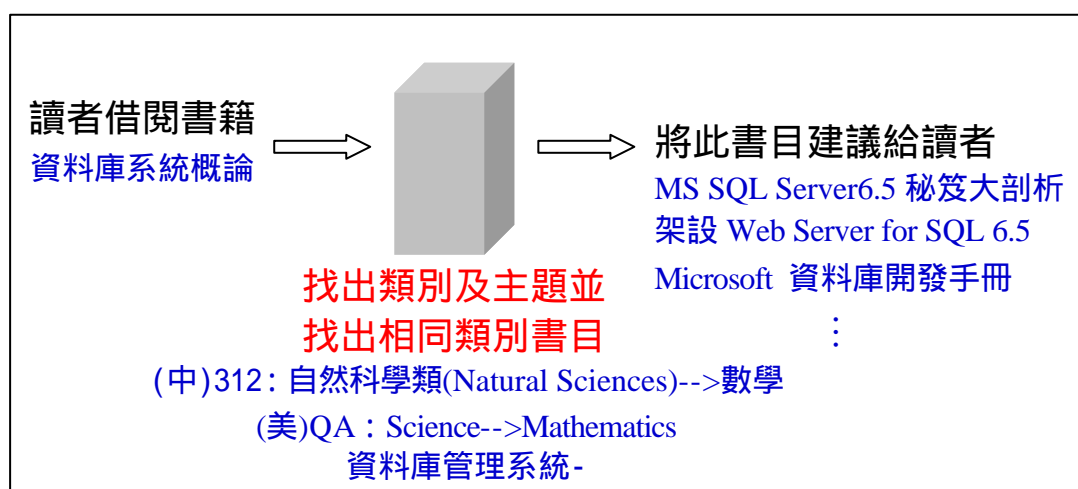


圖 3 - 2 - 5：提昇館藏借閱率

3.2.5 提昇讀者忠誠度

為了要提昇讀者忠誠度，我們藉由資料探勘發掘出讀者的社群關係 -- 館藏借閱的共同性及館藏借閱的順序性，運用這些社群關係適時地建議讀者借閱其他館藏，以增加讀者繼續借閱館藏之機會，如此一來便可以提昇讀者的忠誠度，使讀者不會只借一、兩次就不再來借閱。

圖 3 - 2 - 6為提昇讀者忠誠度的流程圖。

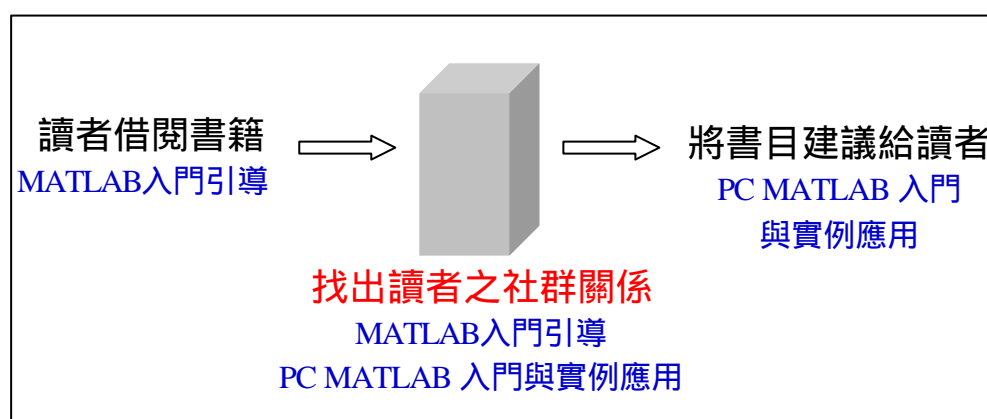


圖 3 - 2 - 6：提昇讀者忠誠度

3.2.6 協助圖書複本採訪政策

為了要判斷是否該購置複本，我們利用 Apriori 的方法在交通大學圖書館書目、館藏、交易史資料中探勘，以找出讀者重複借閱(同一位讀者借閱同一本館藏二次或二次以上)及重複預約館藏的特性，再將這些經常被重複借閱及預約的館藏交易歷史資料做分析，若發現這些館藏有多人預約且一直有人在借閱及預約，我們便可將這些館藏做為複本添購的依據之一。

圖 3 - 2 - 7為協助複本採訪政策的流程圖。

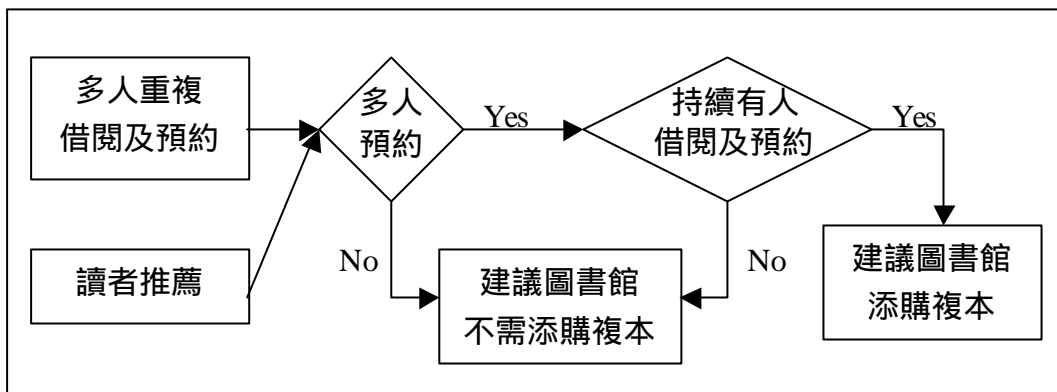


圖 3 - 2 - 7：複本採訪政策

3.2.7 促進館藏流通率

為了要促進館藏的流通率，我們統計出學期初最熱門的五十本預約書，並將這些館藏的借閱期限由原本的一個月縮短為二個星期，以增加這些館藏的流通率，讓讀者在預約後很短的時間內就可以借到這些熱門的預約書。

圖 3 - 2 - 8 為促進館藏流通率的流程圖。

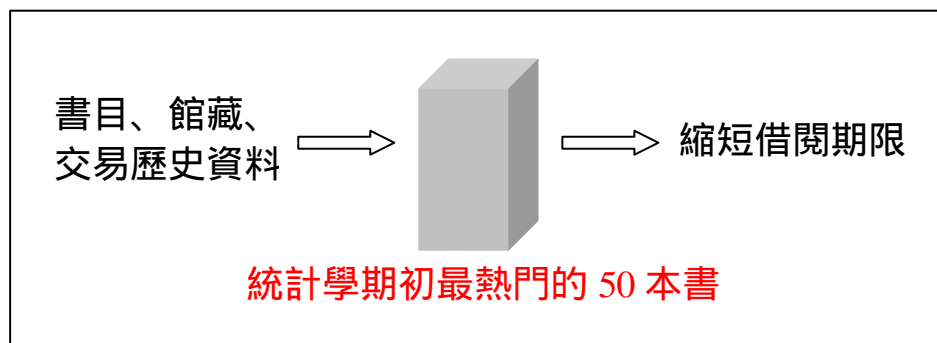


圖 3 - 2 - 8：促進館藏流通率

第四章 分析結果

本論文以交通大學圖書館 1998 年 1 月至 2000 年 8 月的書目、館藏及交易歷史記錄為基礎，運用資料探勘技術探索交大學讀者的社群特性來提昇圖書館的經營與服務，期能使交通大學圖書館在交通大學讀者學術、知識的吸收和產出中扮演更積極的角色，讓交個成功的圖書館。本章即利用資料探勘的技術及統計的方法來分析結果，探討是否達到以下幾項目的：(一)探索讀者社群關係，(二)吸引讀者到館借閱，(三)提昇館藏借閱率，(四)提昇讀者的忠誠度。(五)協助館藏複本採訪政策。(六)促進館藏的流通率。

第一節 探索讀者社群關係

讀者的社群關係通常分為二種：

1. 館藏借閱的共同性：利用 Apriori 的方法在圖書館交通大學圖書館書目、館藏、交易歷史資料中探勘，以找出讀者館藏借閱的共同性。

首先，我們針對國立交通大學圖書館 1999 年 1 月至 12 月的交易歷史資料做探勘，並假設每位讀者在這一年中所借閱過的館藏為一筆交易，則交易歷史資料中的借閱資料的交易筆數共 10,773 筆。我們將最小支持度設為 0.15%，結果顯示讀者最常一起借閱館藏的情形如表 4 - 1 - 1。由表 4 - 1 - 1 我們發現二個情形：

- ◆ 1999 年讀者最常一起借閱的館藏為“紅色警戒(上)”及“紅色警戒(下)”，我們探討其原因可能為：在 1999 年時，“紅色警戒”這個遊戲很熱門，所以帶動起欣賞這部影片的熱潮。
- ◆ 借閱某二部影片的人數比借閱某幾本書籍的人數來得多，探討其原因可能為：影片的數量比書籍少，而且影片通常都有分(上)、(下)

二集，因此我們認為影片和書籍屬於不同類型的館藏，應該將其分開做探勘。

書名	借閱人數
紅色警戒 (下)=Thin Red Line 紅色警戒 (上)=Thin Red Line	59
Borland C++使用手冊 Applied numerical analysis	31
The Godfather [videorecording]=教父 The Godfather Part 2 [videorecording]=教父第二集	28
The Empire Strikes Back [videorecording]=帝國大反擊 Return of the Jedi [videorecording]=絕地大反攻	27
Star Wars [videorecording]=星際大戰 The Empire Strikes Back [videorecording]=帝國大反擊	26
Star Wars [videorecording]=星際大戰 Return of the Jedi [videorecording]=絕地大反攻	23
空中英語教室 [書].1998-7 王牌對王牌 [DVD]=The Negotiator	18
這一夜,誰來說相聲 (上) [錄影資料] 這一夜,誰來說相聲 (下) [錄影資料]	17
仙河飲馬 淨土之春	16
哈啦瑪莉=There's something about Mary 怪醫杜立德=Dr. Dolittle	16

表 4 - 1 - 1：1999 年讀者最常一起借閱的館藏

在這次探勘的過程中我們遭遇到一些問題：1999 年借閱資料的交易筆數共 10,773 筆，若將最小支持度設超過 0.2%，則無法探索出讀者借閱館藏的共同性；反之，如果將最小支持度設小於 0.15%，則面臨記憶體不足的問題。為了解決這樣的問題，我們提出了二個方法減少交易筆數並讓交易資料更為有效：(一)將交易的時間由原本的一年拉長為 1998 年 1 月 1 日至 2000 年 8 月 31 日，也就是每位讀者所借閱過的館藏都視為同一筆交易；(二)若讀者只借過一本館藏，則不會探勘出同時借閱其他館藏，所以我們可將此讀者的交易記錄刪除，以減少交易記錄的筆數。

最後，我們還發現圖書館和商店最大的不同點是：在商店中同樣的物品會有好幾個，但在圖書館中同樣的館藏往往只有一本或少數幾本複本，若有讀者將館藏借走，則其他的讀者便不能同時借閱，因此我們假設讀者預約館藏即表示有興趣借閱，並將預約書資料加入交易記錄中。

在加入預約書資料，將資料量增加為 1998 年 1 月至 2000 年 8 月的資料並將只借過一本館藏的交易去掉後，我們的圖書交易共有 11,398 筆，且最長的交易有 542 個項目。由於圖書館讀者借閱館藏的期限大多為一個月，因此，若假設館藏只有一本，且讀者一借就是一個月，則二年半內最多只有 30 位讀者可借閱，因此我們將最小支持度(minsup)設為 0.21%(即 23 人，約是二年內可借閱的讀者人數)，並針對書籍部份的館藏做探勘，分析出讀者借閱館藏的共同性如表 4 - 1 - 2(a)及表 4 - 1 - 2(b)。

書名	借閱人數
精通 Borland C++ Builder:視覺化 C/C++程式設計.基礎篇 Borland C++ Builder 視窗程式設計經典	52
MPEG video:compression standard Digital video:an introduction to MPEG-2	51
CMOS circuit design, layout, and simulation Low-power cmos wireless communications:a wideband CDMA system design	42
MATLAB 入門引導 PC MATLAB 入門與實例應用	38
親蜜心事 是誰拿走了那一雙雪靴	37
XILINX FPGA/CPLD 數位邏輯設計實習 XILINX FPGA 數位邏輯設計	36
CDMA systems engineering handbook CDMA techniques for third generation mobile systems /	36
CDMA techniques for third generation mobile systems / Spread spectrum CDMA systems for wireless communications	35

表 4 - 1 - 2(a)：讀者最常一起借閱的二本書(一)

書名	借閱人數
Wideband CDMA for third generation mobile communications CDMA systems engineering handbook	34
FreeBSD 抓得住 INTERNET:伺服器架設與管理 FreeBSD 網路應用	34
精通 Borland C++ Builder:視覺化 C/C++程式設計.基礎篇 Borland C++ Builder 完全征服手冊	33
JPEG still image data compression standard Win 32 多緒程式設計:執行緒完全手冊=Multithreading Applications in Win 32	30
OpenGL reference manual:the official reference document to OpenGL, version 1.1 OpenGL programming guide:the official reference guide for OpenGL, release 1.1	29
RF power amplifiers for wireless communications / Microwave circuit design using linear and nonlinear techniques	28
CMOS circuit design, layout, and simulation Low-Power CMOS Design	26
CDMA techniques for third generation mobile systems / Smart antennas for wireless communications : IS-95 and third generation CDMA applications /	26
Win 32 多緒程式設計:執行緒完全手冊=Multithreading Applications in Win 32 神經網路入門與圖樣辨識	25
Numerical Recipes in C:The Art of Scientific Computing An introduction to wavelets	24
CMOS circuit design, layout, and simulation The design of CMOS radio-frequency integrated circuits	24
CMOS circuit design, layout, and simulation Phase-locked loops for wireless communications:digital and analog implementation	24
Visual C++ 6.0 程式開發手冊 Visual C++ 6.0 程式設計指南,應用程式架構篇	24
麵包樹上的女人 賣海豚的女孩	24
Delphi 4.0 徹底研究 煞死你的網頁設計絕招	24
Delphi 4.0 徹底研究 Delphi 4 Bible:應用與進階技巧篇	24

表 4 - 1 - 2(b) : 讀者最常一起借閱的二本書(二)

2. 館藏借閱的順序：在第三章第二節中，我們說明了如何改進 GSP 的方法，在圖書館交通大學圖書館書目、館藏、交易歷史資料中探勘，以找出讀者借閱館藏的順序性。

我們取得交通大學圖書館 1998 年 1 月至 2000 年 8 月的交易歷史資料，加入預約書資料並將只借過一本館藏的交易去掉後，我們的圖書交易共有 11,398 筆，且最長的交易有 542 個項目。我們假設最小支持度 (minsup) 為 0.21%，針對書籍部份的館藏做探勘，分析出讀者借閱館藏的順序性如表 4 - 1 - 3。

書名	借閱人數
MATLAB 入門引導 PC MATLAB 入門與實例應用	37
Wideband CDMA for third generation mobile communications CDMA systems engineering handbook	32
CDMA systems engineering handbook CDMA techniques for third generation mobile systems /	29

表 4 - 1 - 3：讀者最常依序借閱的二本書

第二節 吸引讀者到館借閱

2000 年 8 月時，交通大學圖書館的讀者人數共有 35,670 人，其中包含清華大學、中央大學及校外人士，而交通大學有效的讀者人數為 13,322 人，交通大學有效讀者在 1998/1/1~2000/8/31 有借閱館藏的人數為 9,092 人，佔交通大學有效讀者的 62.25%，我們發現有將近 40% 的讀者從未到圖書館借閱館藏，因此希望藉由統計的方法知道哪些館藏最常被借閱，進一步將這些館藏推薦給讀者，增加借閱館藏的讀者人數。

圖 4 - 2 - 1 為交通大學 1999 年每月借閱館藏的人數分佈圖，我們發現每個月有借閱館藏的讀者人數都未超過一半。

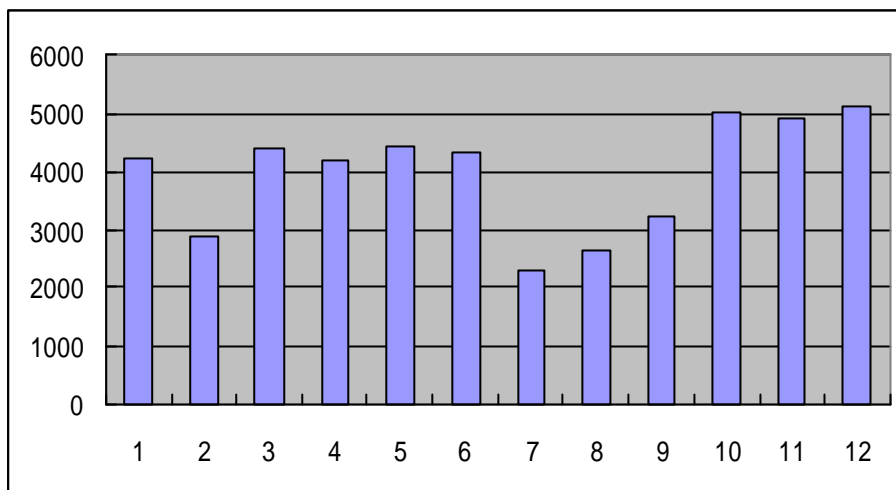


圖 4 - 2 - 1 : 1999 年每月借書人數分佈圖

以交通大學圖書館之交易歷史記錄為基礎，經由統計找出最熱門的二十本書如表 4 - 2 - 1。

書刊名	借閱次數
惡童日記:惡童三部曲.(一)	54
你愛我嗎?	53
潛水鐘與蝴蝶=Le Scaphandre et le Papillon	51
MATLAB 入門引導	50
最後 14 堂星期二的課=Tuesdays with Morrie	45
PC MATLAB 入門與實例應用	44
二人證據:惡童三部曲.(二)	43
第三謊言:惡童三部曲.(三)	41
牧羊少年奇幻之旅	40
遇見 100% 的女孩	40
Microsoft Windows 2000 先睹為快	39
VHDL 數位系統電路設計	39
WWW 文件設計:HTML 語言實務	39
我坐在琵琶河畔，哭泣。=By the River Pledra I sat down and wept	39
VHDL 與數位邏輯設計	38
向左走.向右走	38
Linux 技術參考手冊,周邊設備,網路篇	37
北京法源寺	37
Microwave engineering	36
Redhat Linux 隨裝即用(含 CLE)	36

表 4 - 2 - 1 : 交通大學近一年來最熱門的二十本書

我們將這二十本書經由寄發電子郵件的方式推薦給從未至圖書館借閱館藏的讀者，結果如下：

- ◆ 寄發信件數：798 封。
- ◆ 回收信件數：128 封。
- ◆ 有效信件：110 封。
- ◆ 感興趣、會來借閱的讀者人數：106 人。
- ◆ 不感興趣、不會來借閱的讀者人數：4 人。

結果顯示共有 96.36%的讀者對我們的建議很感興趣，並表示會來圖書館借閱，因此得知，藉由我們的方法可以有效地增加來借閱館藏的讀者人數。

第三節 提昇館藏借閱率

目前交通大學圖書館的館藏中有 405,785 本圖書，526,868 本期刊，共有九十多萬冊的資料，其中期刊和碩博士論文等都是不可外借的，而可外借圖書不含複本共有 282,250 本。這些可外借的圖書中，在 1998/1/1~2000/8/31 中有被借過的館藏，含複本有 143,956 本，不含複本的只有 123,647 本，佔了可借圖書的 43.8%，我們發現有一半以上的圖書都沒有被借過。因此，我們希望加入館藏的分類，當讀者借閱某一館藏時，我們便將相同類別的館藏推薦給他，藉此將館藏推銷出去，增加館藏被借閱的機會。

我們將館藏經由“中國圖書分類法”、“美國國會圖書分類法”及主題做分類，將同一類的館藏加到相關書籍中，然後將讀者借過的館藏之相關書目資料經由寄發電子郵件的方式推薦給讀者，結果如下：

- ◆ 寄發信件數：98 封。
- ◆ 回收信件數：31 封。
- ◆ 有效信件：29 封。

- ◆ 感興趣、會來借閱的讀者人數：28 人。
- ◆ 不感興趣、不會來借閱的讀者人數：1 人。

結果顯示共有 96.55%的讀者對我們的建議很感興趣，並表示會來圖書館借閱，因此得知，藉由我們的方法可以有效地將館藏推銷出去。

第四節 提昇讀者忠誠度

由第二節的統計資料我們知道交通大學有效讀者在 1998/1/1~2000/8/31 有借閱館藏的人數為 9,092 人，佔交通大學有效讀者的 62.25%。而這些人之中有 3,040 人借閱館藏數量少於十本，佔了有借閱館藏讀者之 33.44%，亦即有在借閱館藏的讀者中，很多都只借了幾本館藏就沒有再借閱了。

圖 4 - 4 - 1 為交通大學 1999 年借閱館藏數量的讀者人數分佈圖，我們發現很多讀者所借閱的館藏數均少於十本；圖 4 - 4 - 2 為交通大學 1999 年館藏借閱本數為 10 本以下的人數分佈圖，我們發現館藏借閱數目少於十本的讀者中，又以借閱一本的人數為最多；因此，本文藉由此章第一節所探索出的讀者社群關係後，將讀者借閱館藏的共同性及館藏借閱的順序性推薦給讀者，提供讀者有興趣的資料，讓讀者繼續來借書。

我們將第一節探索出來的讀者社群關係經由寄發電子郵件的方式推薦給讀者，結果如下：

- ◆ 寄發信件數：368 封。
- ◆ 回收信件數：134 封。
- ◆ 有效信件：132 封。
- ◆ 感興趣、會來借閱的讀者人數：127 人。
- ◆ 不感興趣、不會來借閱的讀者人數：5 人。

結果顯示共有 96.21 %的讀者對我們的建議很感興趣，並表示會來圖書館借

閱，因此得知，藉由我們的方法可以有效地提昇讀者的忠誠度。

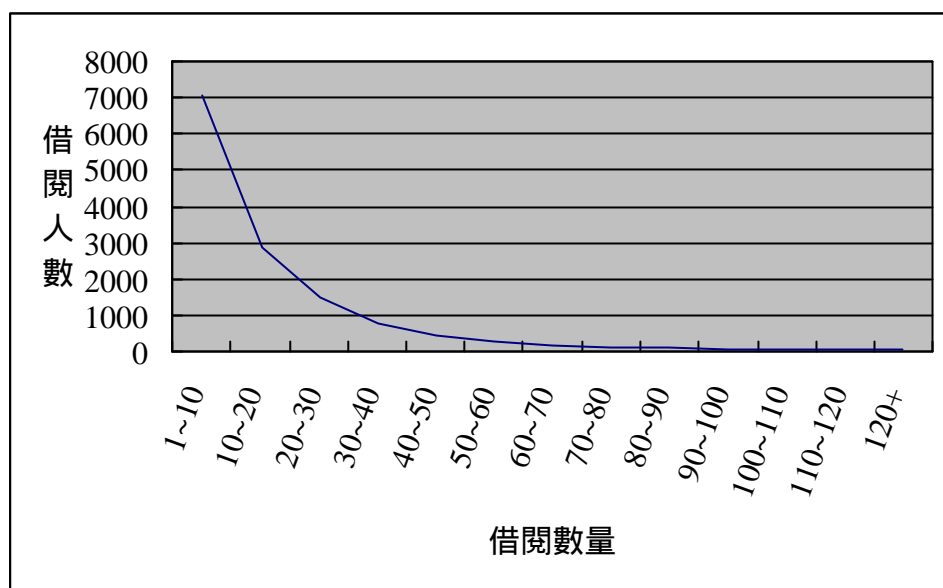


圖 4 - 4 - 1 : 1999 年借閱書籍本數的人數分佈圖

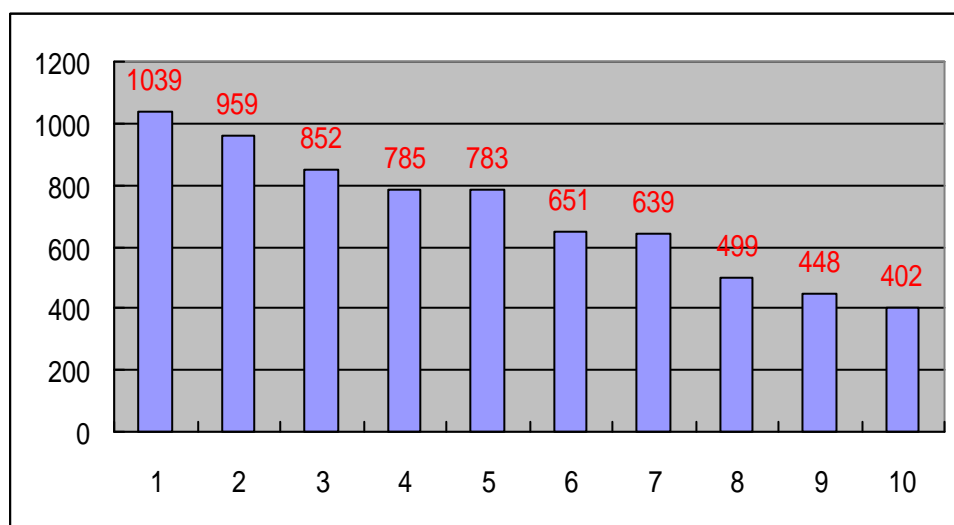


圖 4 - 4 - 2 : 1999 年借閱圖書數目為 10 本以下的人數分佈圖

第五節 協助館藏複本採訪政策

交通大學目前採行的複本採訪原則如下：

- ◆ 電腦類買二本複本以上：由於交通大學多數的系所均為理工類，對電腦書籍的需求比較高，所以電腦類的館藏會買二本以上。

- ◆ 讀者推薦：若讀者覺得哪些書常借不到又要預約很久，便可填申請單建議圖書館採買複本，此時圖書館的館員就會到資料庫裡去查詢，看看這本書是否有很多人借閱及預約，如果有很多人借閱及預約則採買複本。

圖書館目前採行的方法都是需先經由人力判斷哪些書需採買複本，若是讀者借不到館藏時未向圖書館建議採買複本而自行購買，則圖書館便少了一些採買複本依據，如果可以藉由資料探勘得知哪些館藏需加買複本，也可提供圖書館業務參考的依據。

圖書館中的館藏如果有其他讀者預約，則讀者不能續借，必須先歸還館藏後再預約及借閱；若沒有其他讀者預約，也只能續借一次，要再續借則需先歸還再借閱。所以我們推論出，若有很多讀者重複借閱及預約某一本書時，則表示此館藏可能需要多本複本提供更多的讀者借閱。我們利用 Apriori 的方法將圖書館現有的借閱及預約資料做探勘，以找出讀者重複借閱及預約館藏的特性。經過資料探勘後，我們發現從 1998 年 1 月到 2000 年 8 月為止，讀者經常重複借閱的館藏如表 4 - 5 - 1，讀者經常重複預約的館藏如表 4 - 5 - 2。

書名	重複借閱人數	複本數
神經網路入門與圖樣辨識	14	6
Microwave solid-state circuits and applications	14	7
Operating system concepts	13	15
高等工程數學.(下)	12	6
Molecular biology of the cell	11	5
Semiconductor physics and devices:Basic principles	11	5
Introduction to algorithms	11	6
Semiconductor devices:an introduction	11	5
高等工程數學=Advanced Engineering Mathematics.(上)	10	6

表 4 - 5 - 1：讀者經常重複借閱的書籍

書名	重複預約人數	複本數
RF power amplifiers for wireless communications /	8	3
CDMA systems engineering handbook	8	3
Numerical Recipes in C:The Art of Scientific Computing	7	4
計算機結構:計量接近(下)=Computer Architecture A Quantitative Approach	7	2
二人證據:惡童三部曲.(二)	7	4
軟體工程	6	2

表 4 - 5 - 2：讀者經常重複預約的書籍

從表 4 - 5 - 1 中我們發現，很多讀者重複借閱的館藏均有很多本複本，因為多本複本才能供多位讀者重複借閱，讀者如果一直借不到館藏則會重複預約，所以我們認為，若有很多讀者重複預約某一館藏，則表示此館藏需要有多一點的複本，以提供更多的讀者借閱，減少讀者借閱不到的情況。當我們找出讀者經常重複預約的館藏後，必須分析這些館藏是否有很多人在預約，以及是否持續有人在借閱及預約，因此我們將表 4 - 5 - 2 的館藏交易歷史資料做分析，發現“RF power amplifiers for wireless communications /”、“CDMA systems engineering handbook”及“Numerical Recipes in C:The Art of Scientific Computing”這三本館藏一直有很多人在預約，而非只熱門半年或一年，所以我們推斷這些書需要多買一些複本，如此一來便可減少讀者借不到書的情況。

第六節 促進館藏流通率

很多熱門的館藏借出後往往會到應還日才歸還，若某一館藏只有一本，每個讀者又都到應還日才歸還，如此一來，假設讀者借閱期限為一個月，則一年便只有十二位讀者可借閱到此館藏，而其他讀者要借閱便要預約很久才能借到，很多讀者因此而放棄借閱。為了解決這個問題，我們統計出學期初最熱門的五十本預約書，並縮短這些館藏的借閱期限，以增加這些館藏的流通率，讓更多讀者可以借閱這些熱門的預約書。

我們提出此一辦法並與交通大學圖書館館員討論，大家一致認為此方法為一可行之辦法，因此目前交通大學正試用此一辦法。

表 4 - 6 - 1(a)及表 4 - 6 - 1(b)列出交通大學八十九學年度第二學期初最熱門的五十本預約書。

排行	預約次數	書刊名
1	13	DreamWeaver 4 魔法書
2	12	資料庫系統原理-Fundamentals of Database Systems
3	12	數位影像處理-活用 Matlab
4	11	MATLAB 入門及應用
5	10	Probability and statistical inference
6	10	行銷管理學
7	10	材料分析=Materials Analysis
8	9	C++程式語言入門與應用
9	9	Flash 5 特效大全
10	9	TCP/IP 網路管理
11	9	以 MATLAB 透視 DSP
12	9	白色巨塔-The Hospital
13	8	Internetworking with TCP/IP Vol.III:Client-server programming and applications
14	8	Introduction to algorithms
15	8	Linux 的核心與程式設計
16	8	VHDL 與數位邏輯設計
17	8	Wireless CMOS frequency synthesizer design
18	8	乞丐囡仔
19	8	訊號與系統
20	8	蛋白質女孩
21	8	策略行銷管理
22	8	電路程式設計 OrCAD Pspice 9.0
23	7	AUTO LISP 參數化繪圖
24	7	An introduction to database systems /
25	7	Flash 5 動感世界-進階應用
26	7	MATLAB 入門引導
27	7	Perl 學習手冊=Learning Perl
28	7	RF power amplifiers for wireless communications /
29	7	Simulation of communication systems
30	7	Space-time processing for CDMA mobile communications /
31	7	VLSI digital signal processing systems : design and implementation /
32	7	VLSI 製造技術

表 4 - 6 - 1(a)：交通大學八十九學年度第二學期熱門預約書(一)

排行	預約次數	書刊名
33	7	Wideband CDMA for third generation mobile communications
34	7	抓住你的 PhotoImpact 6.x 中文版
35	7	愛回到最初
36	7	新洞悉 UNIX - - 系統程式設計篇
37	7	總裁獅子心:嚴長壽的工作哲學=The Success Story of Stanley Yen
38	6	ASP 網頁製作徹底研究
39	6	CMOS circuit design, layout, and simulation
40	6	Data mining solutions : methods and tools for solving real-world problems /
41	6	Dreamweaver 4 築夢天堂
42	6	Elementary classical analysis
43	6	Fundamentals of microfabrication
44	6	Harry Potter and the Chamber of Secrets /
45	6	Java : how to program /
46	6	LINUX 核心研究篇=Linux Kernel Internals
47	6	MATLAB 5 專業設計技巧
48	6	MATLAB 程式設計,基礎篇
49	6	MATLAB 程式應用及問題範例
50	6	Matlab 程式語言入門

表 4 - 6 - 1(b) : 交通大學八十九學年度第二學期熱門預約書(二)

第五章 結論與未來研究方向

本章總結本論文以及說明未來的研究方向，第一節說明本論文運用資料探勘的技術及統計的方法來提昇圖書館的經營與服務，第二節說明本論文未來可能的研究發展方向。

第一節 結論

本論文應用資訊科技及企業化的模式來經營圖書館，利用資料探勘的技術探索交通大學讀者的社群特性，並運用資料探勘的成果及統計的方法來提昇圖書館的經營與服務，使交通大學圖書館在交通大學讀者學術、知識的吸收和產出中扮演更積極的角色。

本論文達成的目標包含下列六點：

1. 探索讀者社群關係：

甲、館藏借閱的共同性：相同興趣的讀者所借閱的館藏也會很類似，因此我們利用資料探勘技術之相關規則探勘找出讀者常一起借閱的館藏，即館藏借閱的共同性。

乙、館藏借閱的順序性：讀者在借閱館藏時往往會先借入門的館藏再借進階的館藏，因此本論文利用資料探勘技術之循序模式探勘找出讀者常借閱的順序，即館藏借閱的順序性。

2. 吸引讀者到館借閱：很多讀者從未到圖書館借閱過館藏，經由本論文所提出的方法可以成功地讓更多的讀者到館借閱，增加來借閱館藏的讀者人數。

3. 提昇館藏借閱率：很多館藏是未曾被借閱過的，經由本論文的分類及推薦後，可以成功的讓讀者借閱這些館藏，增加館藏被借閱的機會。

4. 提昇讀者忠誠度：很多讀者只借閱過一、兩次館藏就不再到館借閱，經由資料探勘的技術找出讀者的社群關係後，我們便可以適時地給予讀者適當的館藏建議，經由這樣的方法，我們可以成功地讓讀者繼續到館借閱，提昇讀者的忠誠度。
5. 協助館藏複本的採訪政策：圖書館針對一本書所採購的複本數往往有限，但有些熱門書還是必須多買一些複本，本論文利用資料探勘的技術找出須要購買複本的館藏，提供館員購買複本的依據。
6. 促進館藏的流通：很多熱門的館藏借出後往往會到應還日才歸還，其他的讀者要預約很久才能借到，因此我們提出縮短熱門預約書借閱期限的概念，提借給圖書館參考並開始試用，以促進館藏的流通率，讓更多讀者可以借閱。

第二節 未來研究方向

本論文未來的研究方向有下列幾點：

1. 探討 GSP 演算法中的時間條件和讀者的社群關係之關聯性：在 GSP 的演算法中提到了時間限制及滑動的時間視窗兩個條件，本論文中並沒有針對這兩個條件做探討，未來的研究方向將針對這兩個條件來探索讀者的社群關係和此兩個條件間的關聯性。
2. 採用類似“Amazon”的方式推薦讀者相關的館藏：由於“中國圖書分類法”及“美國國會圖書分類法”的分類中，有些層次所包含的書目很廣，有些層次所包含的書目則很少，因此在做分類時，若是以“中國圖書分類法”及“美國國會圖書分類法”做分類，我們往往會不知道哪一些類別該分到第幾層，所以採用主題(Subject)做為分類的依據似乎會來得更方便，未來的研究方向可以針對這個分類方式做探討，找出最適當的分類法則。

3. 加入讀者分類及館藏的分類：由於不同的讀者會有不同的借閱興趣，因此若能針對讀者做分類，將相同興趣的讀者歸在同一類別裡，我們便可更精確地提供讀者有用的館藏資料，推薦給讀者他所感興趣的館藏。另外，由於讀者借閱館藏的目的不同時所借閱的館藏也會不同，如教科書與休閒書，教科書可能只有在修課時會借閱，而休閒類的書則隨時會借閱，如果我們將這些不同類型的館藏分開做探勘，則可針對讀者在不同的時間給予不同的建議。
4. 找出一個更適用於圖書館的資料探勘演算法：本論文中採用 Apriori 演算法及改進 GSP 的演算法提昇圖書館的經營與服務，此兩個演算法原本都是應用在商店交易及其他領域中，然而圖書館的特性與商店是有所不同的，例如：商店中同樣的物品會有很多個，圖書館中相同的書卻只有一本或少數幾本複本，未來期望能針對這些不同的特性提出一個更適合應用在圖書館上的資料探勘演算法。
5. 整合一個實作的系統：由於資料探勘應用在圖書館上所需要的記憶體很龐大，因此必須在工作站上執行，但我們的工作站上並沒有資料庫的系統，所以無法將資料庫及資料探勘結合起來，未來期望能找出一個更適合應用在圖書館上的資料探勘演算法，減少記憶體並將其與資料庫結合，整合出一個實作的系統。
6. 加入漸進更新(Incremental Update)之資料探勘演算法：由於資料探勘是很耗時的，若有新的資料進來就要重新探勘一次會浪費很多時間成本。在[7][9][11]中有提出如何只針對新進的資料做探勘而不將全部的資料重新探勘一次，未來的系統裡期望加上此類演算法，節省時間的成本。
7. 結合個人化資訊環境：本論文中除了探索讀者的社群關係外，也針對個人所借閱的館藏給予不同的推薦，若本系統能和個人化資訊環境結合，便能針對個人的興趣給予不同的建議，提供讀者個人化的服務。

參考文獻

- [1] R. Agrawal, T. Imielinski and A. Swami. “Mining association Rules between Sets of Items in Large Databases”. *Proc. of the 1993 ACW SIGMOD Conference*, 1993.
- [2] R. Agrawal and R. Srikant. “Fast Algorithms for Mining Association Rules”. *Proc. of the 20th VLDB Conference*, 1994.
- [3] R. Agrawal and R. Srikant. “Mining Sequential Patterns”. *IEEE*, 1995.
- [4] R. J. Bayardo Jr. and R. Agrawal, “Mining the Most Interesting Rules”. *Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining*, 1999.
- [5] P. Cabena, P. Hadjinian, R. Stadler, J. Verhees, and A. Zanasi. “Discovering Data Mining From Concept to Implementation”. *Prentice Hall PTR, Upper Saddle River, NJ*, 1998.
- [6] M. S. Chen, J. S. Park, and P. S. Yu. “Efficient Data Mining for Path Traversal Patterns”. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, VOL. 10, NO. 2, MARCH/APRIL 1998.
- [7] D. W. Cheung, S.D. Lee, and B. Kao. “A General Incremental technique for Maintaining Discovered Association Rules”. *Proc. of the 15th Int’l Conf. on Database Systems for Advanced Applications*, 1997.
- [8] J. Han and Y. Fu. “Discovery of multiple-level association rules from large databases”. *21st VLDB Conf.*, 1995.
- [9] M. Y. Lin and Suh-Yin Lee. “Incremental Update on Sequential Patterns in Large Databases”. *IEEE*, 1998.
- [10] K. H. Lin. “An Efficient Mining Approach for Large Reference Sequences and Its Application on the Web”. *Institute of Computer and Information Science National Chiao Tung University*, 2000.
- [11] N. L. Sarda and N. V. Srinivas. “An Adaptive Algorithm for

- Incremental Mining of Associatin Rules”. *IEEE*, 1998.
- [12] R. Srikant and R. Agrawal. “Mining Sequential Patterns: Generalizations and performance improvements”. *IBM Research Division Almaden Research Center*, 1995.
- [13] R. Srikant and R. Agrawal. “Mining quantitative associatin rules in large relational tables”. *ACM SIGMOD Conf. Management of Data*, June 1996.
- [14] R. Srikant, Quoc Vu, and R. Agrawal. “Mining Association Rules with Item Constraints”. *IBM Almaden Research Center*, 1997
- [15] Cleveland, G. (1998), “Digital Libraries: Definitions, Issues and Challenges”, UDT Occasional paper #8. Available at <http://www.ifla.org/VI/5/op/udtop8/udtop8.htm>.
- [16] Digital Library Federation (1998), “A Working Definition of Digital Library”, Available at <http://www.clir.org/diglib/dldefinition.htm>.
- [17] 吳政勳 (1997/04), 「電子圖書館時代的大學圖書館」, *大學圖書館*, 1 卷 2 期, 頁 111-122。
- [18] 胡述兆, 吳祖善 (1994/11), 「圖書館學導論」, *漢美圖書有限公司*, 1994 年 11 月二版二刷, 頁 1。
- [19] 柯皓仁, 「電子圖書館實施經驗談 – 以交通大學圖書館為例」, *書苑季刊*, 第 47 期, pp.10 – pp.33, 2001.
- [20] 陳亞寧 (1999/03), 「另類圖書館：電子圖書館綜觀」, *資訊傳播與圖書館學*, 5 卷 3 期, 頁 141-150。

附錄 利用資料探勘演算法所探勘出來的社群關係

書名	借閱人數	
	共同性	順序性
二人證據:惡童三部曲.(二) 第三謊言:惡童三部曲.(三)	71	70
精通 Borland C++ Builder:視覺化 C/C++程式設計.基礎篇 Borland C++ Builder 視窗程式設計經典	52	<24
MPEG video:compression standard Digital video:an introduction to MPEG-2	51	<24
惡童日記:惡童三部曲.(一) 二人證據:惡童三部曲.(二)	49	44
Delphi 4.0 徹底研究 Delphi 4 Bible:應用與進階技巧篇	47	<24
自動控制系統.(上) 自動控制系統.(下)	46	<24
惡童日記:惡童三部曲.(一) 第三謊言:惡童三部曲.(三)	45	42
雍正皇帝.上,九王奪嫡 雍正皇帝.下,雕弓天狼	43	<24
CMOS circuit design, layout, and simulation Low-power cmos wireless communications:a wideband CDMA system design	42	<24
康熙大帝.上,奪宮 康熙大帝.下,奪宮	39	37
MATLAB 入門引導 PC MATLAB 入門與實例應用	38	37
高等工程數學=Advanced Engineering Mathematics.(上) 高等工程數學.(下)	38	<24
親蜜心事 是誰拿走了那一雙雪靴	37	<24
XILINX FPGA/CPLD 數位邏輯設計實習 XILINX FPGA 數位邏輯設計	36	<24
CDMA systems engineering handbook CDMA techniques for third generation mobile systems /	36	29
CDMA techniques for third generation mobile systems / Spread spectrum CDMA systems for wireless communications	35	<24
Wideband CDMA for third generation mobile communications CDMA systems engineering handbook	34	32
Win 32 多緒程式設計:執行緒完全手冊=Multithreading Applications in Win 32 深入 MFC	34	<24

書名	借閱人數	
	共同性	順序性
精通 Borland C++ Builder:視覺化 C/C++程式設計.基礎篇 Borland C++ Builder 完全征服手冊	33	<24
蘇菲的世界.上 蘇菲的世界.下	33	32
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第二部	33	31
康熙大帝.上,驚風密雨 康熙大帝.下,驚風密雨	32	32
計算機結構:計量接近(上)=Computer Architecture A Quantitative Approach 計算機結構:計量接近(下)=Computer Architecture A Quantitative Approach	32	27
名流劍客沒羽箭.第二部 名流劍客沒羽箭.第三部	32	31
FreeBSD 抓得住 INTERNET:伺服器架設與管理 FreeBSD 網路應用	31	<24
黎明篇.上 黎明篇.下	31	29
名流劍客沒羽箭.第四部 名流劍客沒羽箭.第五部	31	28
JPEG still image data compression standard Win 32 多緒程式設計:執行緒完全手冊=Multithreading Applications in Win 32	30	<24
黎明篇.下 野望篇.上	30	28
仙河飲馬 淨土之春	30	29
伊達政宗.(一),黎明之卷 伊達政宗.(二),人取之卷	30	30
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第三部	30	30
名流劍客沒羽箭.第二部 名流劍客沒羽箭.第四部	30	30
名流劍客沒羽箭.第二部 名流劍客沒羽箭.第五部	30	28
名流劍客沒羽箭.第三部 名流劍客沒羽箭.第四部	30	30
名流劍客沒羽箭.第三部 名流劍客沒羽箭.第五部	30	28
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第四部	30	28
Visual C++ 5 視窗程式設計經典 深入 C++ Builder 設計	29	<24

書名	借閱人數	
	共同性	順序性
OpenGL reference manual:the official reference document to OpenGL, version 1.1 OpenGL programming guide:the official reference guide for OpenGL, release 1.1	29	<24
江湖奇譚.第一部 江湖奇譚.第二部	29	<24
江湖奇譚.第二部 江湖奇譚.第三部	29	<24
劍玄錄.第一部 劍玄錄.第二部	29	27
國家競爭優勢=The competitive advantage of nations.(上) 國家競爭優勢=The competitive advantage of nations.(下)	29	<24
德川家康:破曉之前. 1 德川家康:亂世鴛鴦. 2	29	26
FreeBSD 抓得住 INTERNET:伺服器架設與管理 抓住你的 PhotoImpact 4.2 中文版	28	<24
Win 32 多緒程式設計:執行緒完全手冊=Multithreading Applications in Win 32 PC 影像處理技術.(一),圖檔壓縮篇	28	<24
RF power amplifiers for wireless communications / Microwave circuit design using linear and nonlinear techniques	28	<24
江湖奇譚.第一部 江湖奇譚.第三部	28	<24
大地飛鷹.第二部 大地飛鷹.第三部	28	27
劍玄錄.第三部 劍玄錄.第四部	28	27
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第五部	28	28
Wideband CDMA for third generation mobile communications CDMA techniques for third generation mobile systems /	27	<24
Flash 4 搖擺閃客族 Flash 4 十全大補	27	<24
線性代數 通訊系統	27	<24
TCP/IP 網路管理 FreeBSD 網路架站實務	27	<24
康熙大帝.上,奪宮 康熙大帝.上,驚風密雨	27	27
康熙大帝.下,奪宮 康熙大帝.上,驚風密雨	27	<24
劍玄錄.第一部 劍玄錄.第三部	27	<24

書名	借閱人數	
	共同性	順序性
大地飛鷹.第三部 大地飛鷹.第四部	27	26
Windows 95 程式設計(II) Windows 95 程式設計(I)	26	<24
CMOS circuit design, layout, and simulation Low-Power CMOS Design	26	<24
CDMA techniques for third generation mobile systems / Smart antennas for wireless communications : IS-95 and third generation CDMA applications /	26	<24
策謀篇.上 策謀篇.下	26	<24
天夢飄香 仙河飲馬	26	<24
大地飛鷹.第一部 大地飛鷹.第二部	26	<24
大地飛鷹.第一部 大地飛鷹.第三部	26	<24
大地飛鷹.第四部 大地飛鷹.第五部	26	<24
劍玄錄.第一部 劍玄錄.第四部	26	<24
劍玄錄.第二部 劍玄錄.第三部	26	<24
劍玄錄.續.第三部 劍玄錄.續.第四部	26	<24
飄香箭雨.續.第一部 飄香箭雨.續.第二部	26	<24
飄香箭雨.續.第一部 飄香箭雨.續.第三部	26	<24
飄香箭雨.續.第一部 飄香箭雨.續.第四部	26	<24
飄香箭雨.續.第二部 飄香箭雨.續.第三部	26	<24
飄香箭雨.續.第二部 飄香箭雨.續.第四部	26	<24
飄香箭雨.續.第三部 飄香箭雨.續.第四部	26	<24
雍正皇帝:恨水東逝.上 雍正皇帝:恨水東逝.下	26	<24
伊達政宗.(二),人取之卷 伊達政宗.(三),醍醐夢之卷	26	<24

書名	借閱人數	
	共同性	順序性
Win 32 多緒程式設計:執行緒完全手冊=Multithreading Applications in Win 32 神經網路入門與圖樣辨識	25	<24
Win 32 多緒程式設計:執行緒完全手冊=Multithreading Applications in Win 32 The Windows NT device driver book:a guide for programmers	25	<24
Windows 98 程式設計奧秘=Windows 98 programming secrets Windows 98 程式設計奧秘=Windows 98 programming secrets.2	25	<24
舞 . 舞 . 舞.(上) 舞 . 舞 . 舞.(下)	25	<24
Numerical Recipes in C:The Art of Scientific Computing An introduction to wavelets	24	<24
麵包樹上的女人 賣海豚的女孩	24	<24
CMOS circuit design, layout, and simulation The design of CMOS radio-frequency integrated circuits	24	<24
CMOS circuit design, layout, and simulation Phase-locked loops for wireless communications:digital and analog implementation	24	<24
Visual C++ 6.0 程式開發手冊 Visual C++ 6.0 程式設計指南,應用程式架構篇	24	<24
Delphi 4.0 徹底研究 煞死你的網頁設計絕招	24	<24
德川家康:亂世鴛鴦. 2 德川家康:戀慕秋雨. 3	24	<24
伊達政宗.(一),黎明之卷 伊達政宗.(三),醜聞夢之卷	24	<24
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第二部 名流劍客沒羽箭.第三部	30	30
名流劍客沒羽箭.第二部 名流劍客沒羽箭.第三部 名流劍客沒羽箭.第四部	30	30
名流劍客沒羽箭.第二部 名流劍客沒羽箭.第三部 名流劍客沒羽箭.第五部	30	28
名流劍客沒羽箭.第二部 名流劍客沒羽箭.第四部 名流劍客沒羽箭.第五部	30	28
名流劍客沒羽箭.第三部 名流劍客沒羽箭.第四部 名流劍客沒羽箭.第五部	30	28
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第二部 名流劍客沒羽箭.第四部	30	28

書名	借閱人數	
	共同性	順序性
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第三部 名流劍客沒羽箭.第四部	30	28
惡童日記:惡童三部曲.(一) 二人證據:惡童三部曲.(二) 第三謊言:惡童三部曲.(三)	29	28
江湖奇譚.第一部 江湖奇譚.第二部 江湖奇譚.第三部	28	<24
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第二部 名流劍客沒羽箭.第五部	28	28
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第三部 名流劍客沒羽箭.第五部	28	28
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第四部 名流劍客沒羽箭.第五部	28	28
大地飛鷹.第一部 大地飛鷹.第二部 大地飛鷹.第三部	26	<24
劍玄錄.第一部 劍玄錄.第二部 劍玄錄.第三部	26	<24
飄香箭雨.續.第一部 飄香箭雨.續.第二部 飄香箭雨.續.第三部	26	<24
飄香箭雨.續.第一部 飄香箭雨.續.第二部 飄香箭雨.續.第四部	26	<24
飄香箭雨.續.第一部 飄香箭雨.續.第三部 飄香箭雨.續.第四部	26	<24
飄香箭雨.續.第二部 飄香箭雨.續.第三部 飄香箭雨.續.第四部	26	<24
Wideband CDMA for third generation mobile communications CDMA systems engineering handbook CDMA techniques for third generation mobile systems /	25	<24

書名	借閱人數	
	共同性	順序性
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第二部 名流劍客沒羽箭.第三部 名流劍客沒羽箭.第四部	30	28
名流劍客沒羽箭.第二部 名流劍客沒羽箭.第三部 名流劍客沒羽箭.第四部 名流劍客沒羽箭.第五部	30	28
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第二部 名流劍客沒羽箭.第三部 名流劍客沒羽箭.第五部	28	28
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第二部 名流劍客沒羽箭.第四部 名流劍客沒羽箭.第五部	28	28
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第三部 名流劍客沒羽箭.第四部 名流劍客沒羽箭.第五部	28	28
飄香箭雨.續.第一部 飄香箭雨.續.第二部 飄香箭雨.續.第三部 飄香箭雨.續.第四部	36	<24
名流劍客沒羽箭.第一部 名流劍客沒羽箭.第二部 名流劍客沒羽箭.第三部 名流劍客沒羽箭.第四部 名流劍客沒羽箭.第五部	28	28