A Data Mining System for Mining Library Borrowing History Records

Student: Yu-Min Tai Advisor: Dr. Hao-Ren Ke, Dr. Wei-Pang Yang

Institute of Computer and Information Science

National Chiao Tung University

ABSTRACT

With the rapid development of Internet, digitization has been a world trend. The proliferation of Internet also encourages the development of electronic libraries. In the era of new information technology, how to make use of computer technology to provide readers better services has been the target of all libraries.

The borrowing history of patrons is one excellent evidence to track patrons' interests, in view of this, we aim at finding the association of the collections in National Chiao Tung University (NCTU) Library by analyzing the borrowing history records of NCTU Library. Furthermore, we recommend the associated collections to patrons according to the findings. We expect that NCTU Library can play an active role in the knowledge discovery of NCTU patrons.

In order to achieve the above goal, this thesis chooses the suitable association rule algorithm H-Mine for mining library records and modifies H-Mine to generalized association rule mining and association rule mining with multiple minimum supports. We also implement a data mining system suitable for libraries, the ***Library Borrowing History Records Mining System***. Librarians can get the latest association rules by inserting new library borrowing history records into database, and find different association rules according to patrons of different departments and institutes. This

system also utilizes "New Classification Scheme for Chinese Libraries" to mine associated categories and collections. Furthermore, this system integrates the association rules into a persobalized system, PIE@NCTU (Personalized Information Environment for National Chiao Tung University Library), to recommend associated collections to patrons.

Keywords: Association Rule Mining, Generalized Association Rule Mining, Generalized Association Rule Mining with Multiple Minimum Supports, Mining System, Borrowing History Records

圖書館借閱記錄探勘系統

A Data Mining System for Mining Library Borrowing History Records

研究生：戴玉旻　　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

隨著網際網路的發展與電腦科技的日益進步，資訊數位化已成為世界的趨勢，電子圖書館也在這股資訊潮流下日漸成熟，而如何利用電腦技術以提昇圖書館對讀者的服務品質亦成為各圖書館努力的目標。

由於圖書館的借閱記錄有如讀者使用圖書館資源的最佳證據，因此本論文藉由分析交通大學圖書館的借閱記錄以了解讀者借閱館藏的關聯性，再根據以往讀者借閱的關聯性將館藏有效地推薦給其他讀者，讓交通大學圖書館在讀者探索知識的過程中扮演著積極主動的角色。

本研究根據圖書館借閱記錄的特性，選擇適合圖書館的相關規則演算法並加以改良應用至廣義相關規則探勘(Generalized Association Rule Mining)**及多重最小支持度廣義相關規則探勘**(Generalized Association Rule Mining with Multiple Minimum Supports)**，實作適合圖書館的資料探勘系統「圖書館借閱記錄探勘系統」。讓館員藉由輸入讀者借閱記錄得到最新的館藏借閱相關規則，針對不同系所的讀者找出不同的相關規則。亦應用「中國圖書分類法」找出讀者借閱關聯類別，且可針對不同階層的類別設定不同的最小支持度門檻值，探勘多重最小支持度廣義相關規則，並結合交通大學個人化數位圖書資訊環境** (PIE@NCTU) **將相關館藏推薦給讀者。**

關鍵字：相關規則探勘，廣義相關規則探勘，多重最小支持度廣義相關規則探勘，探勘系統，借閱記錄

誌謝

感謝指導教授柯皓仁教授與楊維邦教授的指導。不僅在我有疑問時，指引方向，提供解決方案，還教導我如何作研究，如何解決問題。也在我稍有怠惰時，盡責地督促，讓我不至於因懈怠而影響研究。除了在研究上的指導外，教授也會關心我的日常生活，關懷備至。

同時也要感謝圖書館資訊組蔡淑琴小姐熱情支援圖書館的相關資料及提供許多寶貴的經驗與意見，使得我的論文得以順利完成。並且感謝圖書館館員們，讓我對圖書館學及館員們的工作有更進一步的認識，以及國科會數位圖書館暨館際合作計畫室成員們給我的意見、協助與鼓勵。

此外，還要感謝實驗室的學長姐們在研究上及生活中的啟發與指導，提供我許多寶貴的建議及人生的經驗。謝謝實驗室的夥伴們的關懷與照顧。還有學妹莉君在個人化數位圖書資訊環境上的支援與協助。

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。在我遇到低潮時，給我加油打氣，聽我抱怨，提供一個舒適的避風港；在我忙於研究時，替我處理所有瑣事，成為最好的後援。讓我能專心致力於研究，並得以順利完成學業。

目錄

表目錄

圖目錄

第一章 圖書館記錄探勘系統簡介

第一節 研究動機及目的

隨著網際網路的發展與電腦科技的日益進步，資訊數位化已成為世界的趨勢，電子圖書館也在這股資訊潮流下日漸成熟，而如何利用電腦技術以提昇圖書館對讀者的服務品質亦成為各圖書館努力的目標。

圖書館的目的是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。然而早在 1979 年美國 Pittsburgh 大學的調查報告[7]中指出，圖書館的館藏資源只有少數被有效利用，值此電子圖書館時代，為了讓讀者快速、有效、完整地滿足其資訊需求，各圖書館必須善加應用資訊技術以有效幫助讀者使用館藏資源。

當圖書館想要廣泛地運用資訊技術實施讀者服務時，首要考量即是找出讀者迫切需求之服務及目前現行服務不足之處。在現今圖書館所提供的服務中，讀者往往會迷失在館藏檢索系統裡，因為檢索所得資源過多，讀者難以由書名、作者、出版社等簡要資訊抉擇要借閱的館藏；再加上館藏數量漸趨龐大，若讀者不熟悉檢索系統的功能，常會找不到所需要的資源。要解決此一困境，圖書館可以透過對讀者興趣的了解，幫助讀者找尋館藏資源。

卜小蝶在[24]一文中提到，圖書館借閱記錄是讀者使用圖書館資源的最佳「證據」，也是讀者積極滿足個人資訊需求的行為結果，這類資訊能反映使用者實際的資訊需求，因此對於掌握讀者興趣，進而作為加強圖書館資源利用的基礎具有一定的參考價值。除此之外，由於借閱記錄蘊含大量讀者與圖書館互動的歷史記錄，若能利用資料探勘(Data MIning)的技術從中挖掘隱藏有意義的資訊，不僅有助於讀者資訊需求的瞭解，還可視為加強圖書館資源利用的重要指標。

資料探勘(Data Mining)是知識管理的應用技術之一，其主要目的在探討如何從大量資料中，發掘出潛藏有用的資訊或規則，以提供決策參考之用。目前這類技術多半用於商業、醫學等資料量龐大又具有商機的領域上，如亞馬遜網路書店(http://www.amazon.com)由銷售記錄發掘顧客消費的消費關聯性，藉此將關連性產品推薦給有相似購買行為的顧客。運用資料探勘，經營者可更深入了解客戶需求，甚至可提供量身訂作之個人化服務，亦即採用客戶關係管理(Customer Relationship Management – CRM)的理念，以期增加客戶的滿意度與忠誠度，並確保在激烈商業競爭的環境裡獲利[27]。

群體化(Community)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、分享心得與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的技術來探索讀者社群的持性，從而達成群體化的電子圖書館讀者服務。為達此一目的，我們建置一套協助館員瞭解讀者的興趣及需求的探勘系統，讓館員藉由分析借閱記錄找出讀者經常一起借閱的館藏，再將關聯性館藏推薦給借閱同樣館藏的讀者。由於所找出的關聯館藏只佔了整個館藏資源微乎其微的一小部分，並不能有效地提供讀者意見，因此將中國圖書分類法加入分類階層探勘，並讓館員隨著分類階層設定不同的最小支持度門檻值，挖掘出最適宜的讀者借閱類別關聯性。如此一來，藉由借閱類別關聯性，根據讀者的借閱記錄分析興趣類別，進而推薦關聯類別的新進館藏給讀者，讓系統有效地提供更多更實際的借閱建議。

第二節 研究方法及目標

本論文以交通大學圖書館的借閱記錄為基礎，運用資料探勘的技術，探索讀者借閱館藏及類別的關聯性，並運用探勘的成果來提昇圖書館的經營與服務。

本研究根據圖書館借閱記錄的特性，設計一套適合圖書館的資料探勘系統

「圖書館借閱記錄探勘系統」，簡稱「圖書館資料探勘系統」。藉由資料倉儲(Data Warehousing)[11] 的技術處理圖書館借閱記錄的前置作業，以及運用相關規則探勘(Association Rule Mining)演算法 H-Mine[14] 的技術找出借閱館藏的關聯性，除此之外，並以 H-Mine 為基礎發展廣義相關規則探勘(Generalized Association Rule Mining)演算法及多重最小支持度廣義相關規則探勘(Generalized Association Rule Mining with Multiple Minimum Supports)演算法，以便找出借閱類別的關聯性。

在「圖書館借閱記錄探勘系統」發掘出讀者社群關係後，我們希望能運用這些成果達到以下目標：

- 提供讀者借閱館藏的建議：透過探勘讀者借閱關聯性，將關聯性館藏推薦給其他借閱同樣館藏的讀者。如：相關規則 " 5%有借閱過沉船的讀者也會借閱盜墓及老貓這二本書 "，若有讀者借閱過沉船、盜墓或是老貓中的其中一本，則推薦另二本給該讀者。在讀者檢索沉船的同時，也可推薦其他關聯館藏如盜墓給讀者。
- 推薦讀者新進館藏：藉著探勘借閱類別關聯性，經由該讀者的借閱記錄或是個人化系統中的興趣記錄，推薦讀者可能有興趣的關聯類別新書。如：經由讀者的借閱記錄或個人興趣記錄得知讀者喜好借閱電腦科學類的書，而探勘得知借閱類別關聯性 " 借閱電腦科學類書籍的讀者也同時會對語文類及企業管理類的書籍有興趣 "，因此若有電腦科學類、語文類及企業管理類的新書時，可發出新書通報推薦該讀者借閱。

第三節 論文架構

本論文第二章是敘述資料探勘相關研究工作，簡介產生相關規則的知名演算法 Apriori 及我們所選擇適用於圖書館借閱記錄的相關規則演算法 H-Mine；第三

章我們提出以 H-Mine **為基礎的廣義相關規則演算法** H-Mine(Generalized)**及多重最小支持度廣義相關規則演算法** H-Mine(MMS)**；第四章說明實作的「圖書館借閱記錄探勘系統」，簡稱「圖書館資料探勘系統」；第五章是圖書館資料探勘系統效益評估；第六章則歸納結論與未來研究方向。**

第二章 資料探勘相關研究工作

本論文是利用資料探勘(Data Mining)的技術，分析過去讀者借閱的歷史記錄，了解館藏的借閱關聯性，並且將關聯館藏推薦給讀者作為借閱時的參考。本論文運用資料探勘中的相關規則探勘(Association Rule Mining)及其延伸應用，包括廣義相關規則探勘(Generalized Association Rule Mining)與多重最小支持度相關規則探勘(Association Rule Mining with Multiple Minimum Supports)，藉此探勘借閱館藏的社群關係。

在本章中，第一節介紹資料探勘的技術；第二節說明相關規則探勘的二大類方法，包括 Apriori 演算法及 H-Mine 演算法；第三節描述相關規則探勘的延伸應用，說明廣義相關規則探勘及多重最小支持度廣義相關規則探勘的方法。

第一節 資料探勘

所謂的資料探勘，簡單來說即是從儲存於資料庫(Database)、資料倉儲(Data Warehousing)及資訊儲存器(Information Repositio ry)的大量資料中發掘出感興趣的知識(非瑣碎的、有隱含意義的、之前未知的、有潛力有用的)之處理過程，又稱資料庫探勘(Database Mining)、知識萃取(Knowledge Extraction)、資料考古(Data Dredging)及資訊收穫(Information Harvesting)等等[11]。資料探勘是資料庫知識探索(Knowledge Discovery in Database)的步驟之一，也是其中的主要核心步驟，因此有些學者將資料探勘與資料庫之知識探索二者視為同義詞。如圖 2 - 1 - 1 所示，整個知識挖掘的過程看似一個線性的過程，然而在過程中的每個步驟皆可返回，或是加入其他步驟[1]。資料庫之知識探索的過程主要包含以下四個步驟[11]：

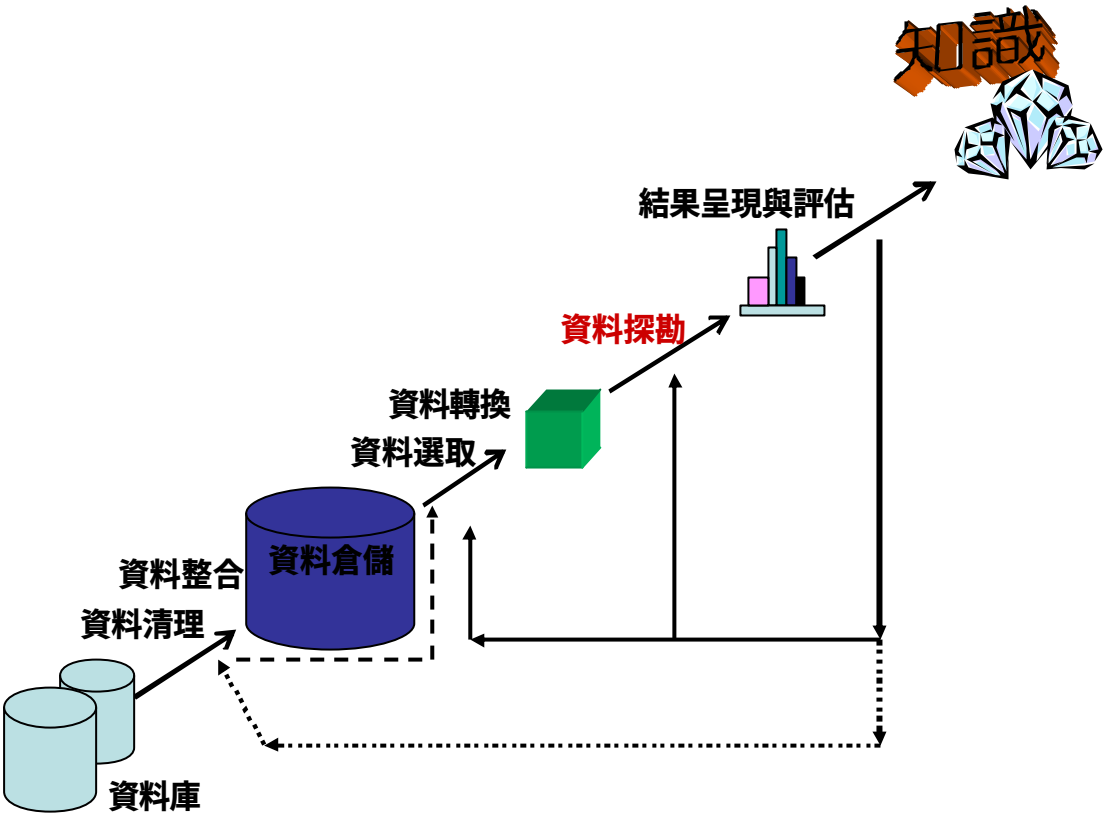


圖 2 - 1 - 1：資料庫之知識探索流程圖[11]

一、確定目標

明確地定義出問題所在及想要得到的結果。

二、預備資料

包含資料選取與資料前置處理二部分。這是最花費時間的部分，約佔整個知識探索過程的百分之六十，而預備資料的優劣亦會反應在知識探索的成效上。

- 資料選取：根據探勘的目標，從所有的資料中選擇適用的資料。
- 資料前置處理：又分為資料清理、資料整合、資料轉換、及資料簡化與量化。
  - 資料清理：資料庫中的資料可能會包含一些錯誤、遺失或是不完整的資料，為避免影響到知識探索的正確性，必須對這些資料特別處理，例如只保留資料中適用的部分、直接刪除有錯誤或是異常的資料、或是利用

數學統計或模糊理論方法來推論，針對不完整或前後不一致的資料作處理。

- 資料整合：資料整合包含以下幾種情形：
  - 資料可能來自不同的資料庫、資料倉儲或其他資訊儲存器，必須將不同來源的資料整合在統一格式的儲存器裡。
  - 整合不同來源的詮釋資料(Metadata)。
  - 將不同型態資料內容整合成一致且合理的值，如：描述日期的單位由民國年轉換成西元年。
  - 將不同格式(Format)的資料轉換成相同格式的資料，如：轉換欄位排列格式。
- 資料轉換：根據採用的資料探勘演算法之需求，對原始資料進行必要的轉換。轉換方式包含有：
  - 彙整(Aggregation)：將資料彙集加總。如每日營業額彙整成為每月的營業資料。
  - 正規化(Formalization)：依據特定範圍將屬性資料作刪減。
  - 歸納(Generalization)：將低階(Low Level)或是原始資料以較高階(High Level)的觀點重新定位。
  - 建立屬性(Attribute Construction)：以屬性的方式取代原本的表示法。如：以青少年，中年，老年屬性取代原本以年紀數字表示的資料。
- 資料簡化與量化：資料簡化是將資料中表示法過於複雜的部份簡化，以較簡單明瞭，但又不會影響分析結果的方式表示。如：變換過於精確的

單位為一般的單位，將錢的數量直接以千元為基本單位等。而資料量化則是使用多次元(Dimensionality)縮減、轉換或編碼等方法減少有效的變數或資料。

三、資料探勘

根據所定義的問題選擇適合的資料探勘演算法，在資料中找尋有用的特徵，並決定採用探勘模式及參數是否適當。資料探勘演算法包含觀念描述(Concept Description)、關連性(Association)、分類(Classification)、分群分析(Cluster Analysis)、及趨勢分析(Trend and Evolution Analysis)等等。

四、結果評估與呈現

依據一些量測的興趣度(Interestingness Measure)，評估真正令人感興趣的資料樣式，並且根據資料探勘演算法的結果，決定其適合的呈現方式，例如分類分群的結果較適合以圖表的方式表示，而關聯性則適合以規則的方式呈現。除此之外，尚須分析結果的適用性，期能應用到相關領域上。

第二節 相關規則探勘

相關規則探勘是資料探勘的方法之一。相關規則探勘經常運用在商店的交易記錄，針對使用者交易行為作關聯性分析，藉由交易商品的關聯性決定搭配促銷商品、商品架位等行銷策略，以提高購買率，增加商店業績。例如：“20%買牙刷的顧客也會同時買牙膏、毛巾和香皂”就是一個典型的相關規則。Agrawal 最早提出從交易資料庫中發掘出相關規則的演算法[2]，其後陸續有學者將相關規則的概念應用到其他領域，提出適用該領域的演算法。本篇論文將相關規則探勘應用在圖書館上，探討讀者借閱館藏及借閱類別的關聯性問題。

2.2.1 相關規則簡介

相關規則的正規敘述[1]**如下：**

令 $I=\{i_1,i_2,\ldots,i_m\}$ **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**。以圖書館的應用來看，每一本書就是一個交易項目，一個讀者在一段時間內來圖書館借閱館藏的集合即為一筆交易。**

一個相關規則(Association Rule)**表示成** $X \Rightarrow Y$**，其中** $X \subseteq I$**，** $Y \subseteq I$**，** $X \cap Y = \varnothing$ **。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%**。相關規則探勘定義為：給定交易記錄資料庫** ***D***，**在當中找出所有確信值和支持度大於最小支持度跟最小確信值的規則，其中最小支持度與最小確信值的門檻值由使用者設定。**

Agrawal **等學者**[2] **將相關規則探勘分為二個子問題：**

子問題一：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頻繁項目集(Frequent Itemset)**或是大項目集**(Large Itemset)**，反之稱為罕見項目集**(Infrequent Itemset)**或是小項目集**(Small Itemset)**。**

子問題二：用子問題一中所找到的頻繁項目集來產生所期望的規則。

此步驟的演算法非常直覺，即：對於任一頻繁項目集 *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⇒(*L*-a) 的確信值（也就是 support(*L*)/support(a)）大於最小確信值，則此規則即符合所求。

由於子問題二 Agrawal 已經在[3]中提出有效率的演算法，所以學者們不再進一步探討產生規則的方法，而是針對如何有效率地找出所有的頻繁項目集作研究，文獻上將此問題稱為探勘頻繁項目集(Mining Frequent Itemsets)。

2.2.2 探勘頻繁項目集

探勘頻繁項目集的方法主要分為二派，一是根據 Agrawal 所提出的 Apriori [3] 演算法為研究基礎的候選項目集產生及測試法(Candidate Generation and Test Method)；另一則是利用各種資料結構計算頻繁項目集的支持度，直接產生結果的頻繁項目集成長法(Frequent Pattern Growth Method)。

以下簡介產生頻繁項目集的二大類方法：

- 候選項目集產生及測試法

這類方法是以 Agrawal 所提出的 Apriori [3]演算法為研究基礎。Apriori 演算法以疊代(Iteration)的方式產生頻繁項目集。每一次疊代時產生所有相同長度的頻繁項目集，在第一次時疊代產生長度為 1 的頻繁項目集，第二次時則產生長度為 2 的頻繁項目集，依此類推。每一次產生的頻繁項目集當作下一次疊代的種子集(Seed Set)，由種子集來推論下一次疊代所有可能會出現的頻繁項目集，文獻上稱此可能出現的頻繁項目集為候選項目集(Candidate Itemset)。每一次疊代只要將所有交易和產生之候選項目集加以比對並計算它們的支持度，候選項目集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頻繁項目集。如此一直反覆，直到沒有新的頻繁項目集出現為止。圖 2 - 2 - 2 是一個簡單的例子，利用 Apriori 找出所有大於最小支持度 2 的項目集。圖 2 - 2 - 3 列出 Apriori 演算法。

交易資料庫 *D*

交易編號	項目集
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

最小支持度=2

C_1

項目集	支持度
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

項目集	支持度
{1}	2
{2}	3
{3}	3
{5}	3

C_2

項目集	支持度
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

項目集	支持度
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

項目集	支持度
{2 3 5}	2

L_3

項目集	支持度
{2 3 5}	2

L_1 = {large 1-itemsets}；//**產生長度為** 1 **的頻繁項目集**
for ($k=2; L_{k-1} \neq \varnothing; k++$) do begin
  C_k = apriori-gen(L_{k-1})；//**產生** k-**項目集之候選項目集**
  for all transactions $t \in D$ do begin
    C_t = subset(C_k,t); //**交易記錄中所有包含** C_k **的集合**
    for all candidates $c \in C_t$ do
      c.count++; //**計算支持度**
  end
  $L_k = \{c \in C_k | c.count \geq minsup\}$;//**產生長度為** k **的頻繁項目集**
end
Answer = $\cup_k(L_k)$;

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** $c \in C_k$，**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;
```

■ **頻繁項目集成長法**

頻繁項目集成長法的最大特色就是利用各種資料結構計算頻繁項目的支持度，直接產生結果。如 FP-growth [10]**及** Tree-projection [4]**均是利用樹狀結構儲存頻繁項目，並運用樹狀結構計算各頻繁項目的支持度，而** H-Mine [14]**則是運用** Pei **等學者所提出的** H-Struct**，利用動態調整** H-Struct **的連結計算各頻繁項目的支持度，以求出最後的頻繁項目集。**

由於 Apriori **在項目重疊性高、交易項目長的項目集或是在支持度相當小的情況下，效能並不夠好，因此，**Han **等學者**[4][10][14]**分析可能造成效能不彰的因素，並針對缺失提出新演算法。**Han **等學者**[10]**分析** Apriori **效能不彰的原因為二，一是在上述情況下，**Apriori **必須處理龐大數目的候選項目集，所以拖垮整體效能，另一則在產生在長項目集時，必須耗費大量時間來掃瞄資料庫並檢查比對候選項目集。表** 2 - 2 - 1 **是二類頻繁項目集產生方法的比較。**

方法	候選項目集產生及測試法	頻繁項目集成長法
優點	1. 不需再另建資料結構儲存交易項目 2. 演算法較簡單直覺	1. 有效減少重複讀取資料庫的次數 2. 不需要額外產生候選項目集
缺點	1. 花費時間產生可能不是結果的候選項目集 2. 一再重複地掃瞄資料庫	1. 需額外花費時間空間建立資料結構儲存交易項目資料 2. 演算法較複雜

表 2 - 2 - 1：候選項目集產生及測試法與頻繁項目集成長法之比較

由於圖書館的資料量龐大，若是使用候選項目集產生及測試法勢必會因為重複掃描資料庫而拖垮探勘的效能，且由 Han [10]得知，在支持度小、交易量大時，FP-growth 的效能都比 Apriori 佳。圖書館交易記錄的資料量大且讀者借閱的館藏重疊性不高，在設定高支持度時，不易找到頻繁項目集，因此，我們得設定較低的支持度，但如此一來，頻繁項目成長法就較候選項目集產生及測試法適用於圖書館的資料探勘。

現有的頻繁項目成長法主要有三種演算法，分別為 FP-growth [10]、Tree Projection [4] 及 H-Mine [14]，由 Han [10]得知，FP-growth 的效能不管在支持度或是資料量的擴充性上都略勝於 Tree Projection 一籌，但是，將 FP-growth 用在交易項目分散、稀疏、重複性很低的資料集上時，表現又不如 H-Mine 出色。

由於圖書館讀者借閱的期限大多為一個月，若館藏無複本，平均一年只會被借出 12 次，且館藏有複本的數量很少，館藏被借出後其他讀者便無法借閱相同書籍，故圖書館的館藏流通率較一般商店商品流通率低，讀者借閱同一館藏的重疊性亦較顧客購買相同商品重疊性低。

總結以上分析，圖書館資料的特性是資料量大、借閱重疊性低，故在支持度小時才會有比較好的結果，所以，我們採用 H-Mine 為圖書館借閱記錄資料探勘系統的核心演算法。表 2 - 2 - 2 是 H-Mine 與 FP-growth 的優劣比較。

頻繁項目集成長法	H-Mine	FP-growth
優點	重新調整資料結構 H-Struct 的連結即可，不會一直重複產生小的投影資料庫(Projected Database)。	在不同交易中若有共同前幾個項目即可共用相同項目的節點，不需再重複產生相同的樹狀節點。
缺點	不同交易中即使項目十分相似仍無法共用資料結構。	重複產生小的投影資料庫(Projected Database)及附帶的頻繁項目樹 (FP-Tree: Frequent Pattern Tree)。
最適合資料集	交易項目分散、稀疏、重複性很低的資料集，可共用項目少的資料集。	交易項目多且重複性、壓縮性很高的資料集。

表 2 - 2 - 2：H-Mine 與 FP-growth 之比較

由於 H-Mine 較適用於探勘圖書館的借閱資料，故本論文乃以 H-Mine 為基礎發展圖書館資料探勘系統。以下進一步介紹 H-Mine 的運作方式。

H-Mine[14]是由 Jian Pei 等學者所提出的頻繁項目集成長法，H-Mine 運用一資料結構 H-Struct 儲存交易中的頻繁項目集，並透過 H-Struct 動態調整連結，進行資料探勘。這個方法的最大特點就是在 H-Struct 中調整連結以達到如在投影資料庫(Projected Database)中探勘的效果，這個方法的記憶體空間複雜度是可預期的，而 Apriori、FP-growth 在記憶體的使用上則無法預期。

H-Mine 演算法針對資料量大小設計不同的演算法。當所有交易均可置入主要記憶體時，使用 H-Mine(Mem) 演算法；當交易量龐大以致於無法置入主要記憶體時，則是使用 H-Mine 演算法。

直接舉例說明 H-Mine(Mem)。表 2 - 2 - 3 前二欄是假設的交易資料庫，最小支持度設定為 2。

交易編號	項目集	頻繁投影
100	c, d, e, f, g, i	c, d, e, g
200	a, c, d, e, m	a, c, d, e
300	a, b, d, e, g,	a, d, e, g
400	a, c, d, h	a, c, d

表 2 - 2 - 3：相關規則探勘之交易資料庫[14]

H-Mine 演算法的關鍵步驟就是建立資料結構 H-Struct。首先，掃瞄整個資料庫找出符合最小支持度且長度為 1 的單一項目頻繁項目集，得到{a:3, c:3, d:4, e:3, g:2}，a:3 表示交易資料庫中有 3 筆資料包括項目 a。將單一頻繁項目集以標頭表格(Header Table) H 表示，每個頻繁項目包含三個欄位：項目編號、項目支持度及連結位置，其中頻繁項目集的順序為任意順序，而本例是依字母順序排列。再將資料庫中每筆交易項目只保留頻繁項目，並按照頻繁項目集順序排列，得到每筆交易的頻繁投影(Frequent Projection)，而頻繁投影的集合則稱為投影資料庫(Projected Database)。每一個存在 H-Struct 中的頻繁項目包含二個欄位：項目編號及連結位置。H-Struct 的資料結構如圖 2 - 2 - 4。

當交易中所有頻繁投影均可置入主要記憶體時，在每筆投影中的第一個頻繁項目都會以佇列的方式經由連結位置連結起來，而標頭表格 H 中的連結位置則是儲存每一個佇列第一個項目的位置。如圖 2 - 2 - 4，標頭表格 H 的項目 a 即是存放 a 佇列第一個項目的指標 (指向交易投影 200 中第一個項目 a 的指標)，a 佇列則連結 200，300 及 400 三筆交易的投影。

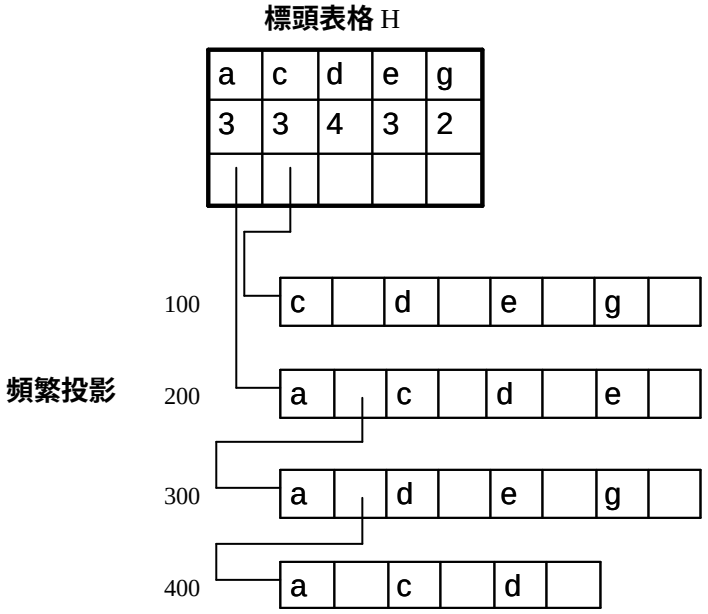


圖 2 - 2 - 4：H-Struct[14]

顯然地，要建立這個 H-Struct 只要掃瞄整個資料庫二次，一次是計算所有的頻繁項目集，另一次則是在每筆交易中保留頻繁項目，建立投影資料庫。之後整

個探勘過程全都在 H-Struct 上進行，不會再使用到原本的交易資料庫。

如圖 2 - 2 - 4 探勘由標頭表格 H 表示的五個項目，分為五個子集，亦可視為五個虛擬投影資料庫，在各自的子集中探勘，找出屬於個別的頻繁項目集。先從第一個子集 a 投影資料庫(屬於 a 佇列的頻繁投影)開始，所有屬於 a 投影資料庫的交易已被連結成 a 佇列，所以，掃瞄 a 投影資料庫時，只要利用 a 佇列的連結即可瀏覽整個 a 投影資料庫。

探勘 a 投影資料庫中的頻繁項目時，必須先建立屬於 a 投影資料庫的標頭表格 H_a，如下圖 2 - 2 - 5。標頭表格 H_a 的欄位就如同標頭表格 H 的欄位。但是，項目支持度是記錄頻繁項目在 a 投影資料庫中出現的次數，如項目 c 在 a 投影資料庫出現 2 次，故在 H_a 中項目 c 的項目支持度為 2。

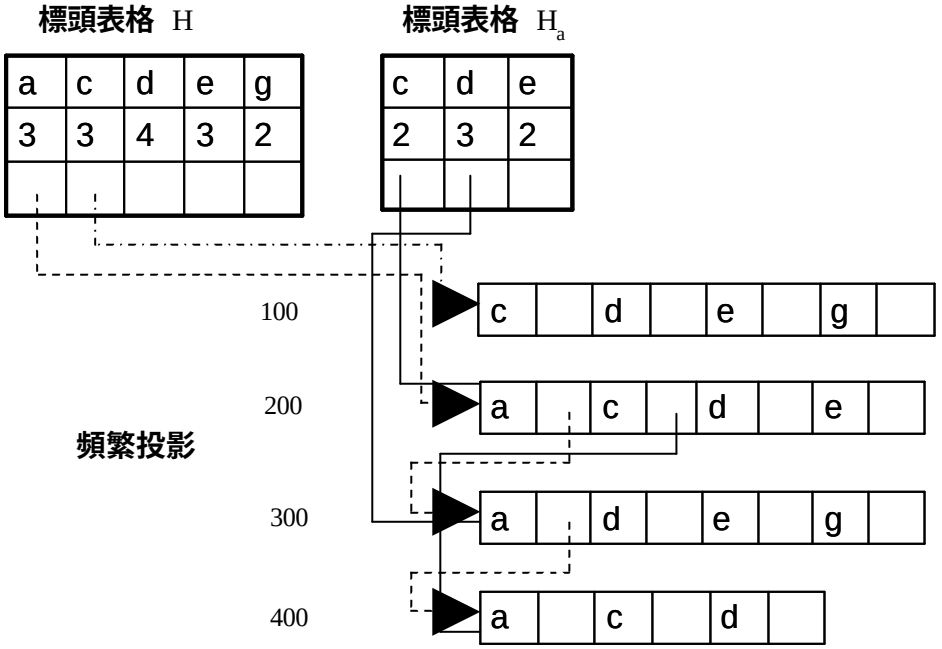


圖 2 - 2 - 5：標頭表格 H_a 及 ac 佇列[14]

掃瞄整個 a 投影資料庫，找出除了項目 a 以外的頻繁項目集，得到{c:3, d:3, e:2}，建立標頭表格 H_a，即可輸出部分結果頻繁項目集{ac:3, ad:3, ae:2}，並建立 H_a 與頻繁投影的連結。如圖 2 - 2 - 5，對 a 投影資料庫而言，交易投影 200 及 400 的第一個項目均為 c，故使標頭表格 H_a 中項目 c 的連結位置指向 200，200 項目 c 的連結位置指向 400，成為 ac 佇列。

同樣地，探勘 ac 投影資料庫中的頻繁項目時，利用 ac 佇列的連結瀏覽 ac 投影資料庫，並建立屬於 ac 投影資料庫的標頭表格 H_{ac}。標頭表格 H_{ac} 中只有項目 d 是頻繁項目，因此只有 acd:2 是結果頻繁項目集。搜尋 ac 為首的頻繁項目集就此結束。

結束以 ac 為首頻繁項目集的搜尋後，回到標頭表格 H_a，下一步尋找包含 ad 但不包含 c 的頻繁項目集。因為在探勘 ac 投影資料庫時就已經找出所有跟 c 有關的項目集，所以在接下來的探勘就不需要再考慮那些已經探勘過的項目。

ad 佇列從標頭表格 H_a 項目的連結位置開始，除了在標頭表格 H_a 一建立就連結好的交易投影 300 外，還必須加入屬於 ac 投影資料庫但是包含有項目 d 的交易。如圖 2 - 2 - 6，交易投影 300 的連結位置指向 200，200 的連結位置指向 400。

在調整連結位置之後，ad 佇列收齊所有頻繁投影中含有項目 ad 的完整集合。經由追蹤 ad 佇列的連結，找到唯一的頻繁項目 e，因此只有 ade:2 是結果頻繁項目集。同樣地，搜尋包含 ad 的頻繁項目集也就此結束。

搜尋包含 ae 的頻繁項目集時，由於項目 e 並無連結，表示並無以 ae 為首的結果頻繁項目集。搜尋 ae 亦就此結束。

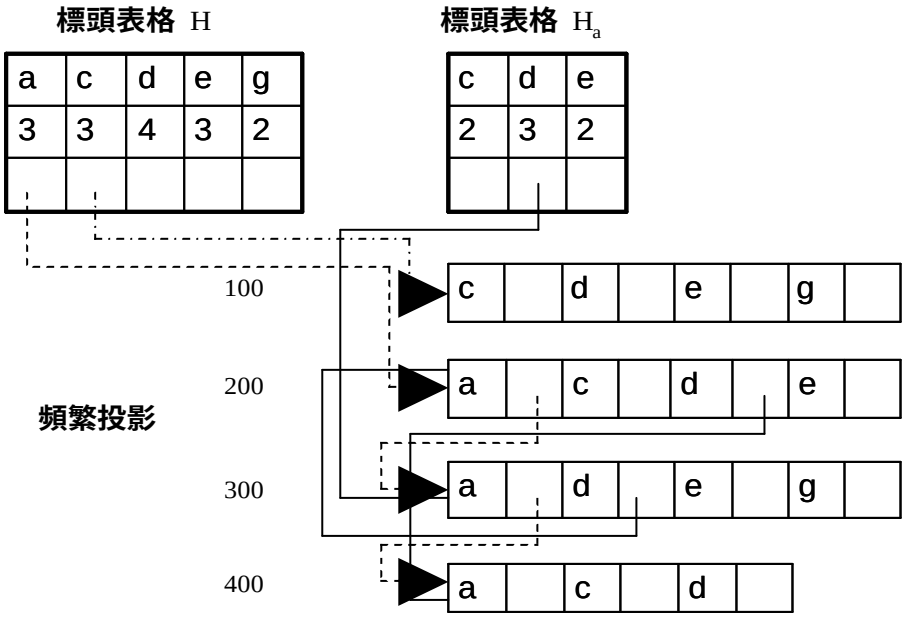


圖 2 - 2 - 6：標頭表格 H_a 及 ad 佇列[14]

結束所有以 a 為首頻繁項目集的搜尋後，回到標頭表格 H，追蹤 c 頻繁項目集的搜尋，此時 c 佇列包含了所有第一個項目為 c 的頻繁投影，但是 a 佇列中仍有同時包含項目 a 及 c 的頻繁投影，即那些在 a 佇列中的投影亦有可能會有以 c 為首的頻繁項目。因此，必須再追蹤 a 佇列的連結，將屬於 a 佇列頻繁投影在項目 a 之後的項目，即第二個頻繁項目（第一個項目均為 a)， 與各自歸屬的佇列連結起來。如圖 2 - 2 - 7 所示，頻繁投影 acde、acd 插入 c 佇列，而頻繁投影 adeg 插入 d 佇列。

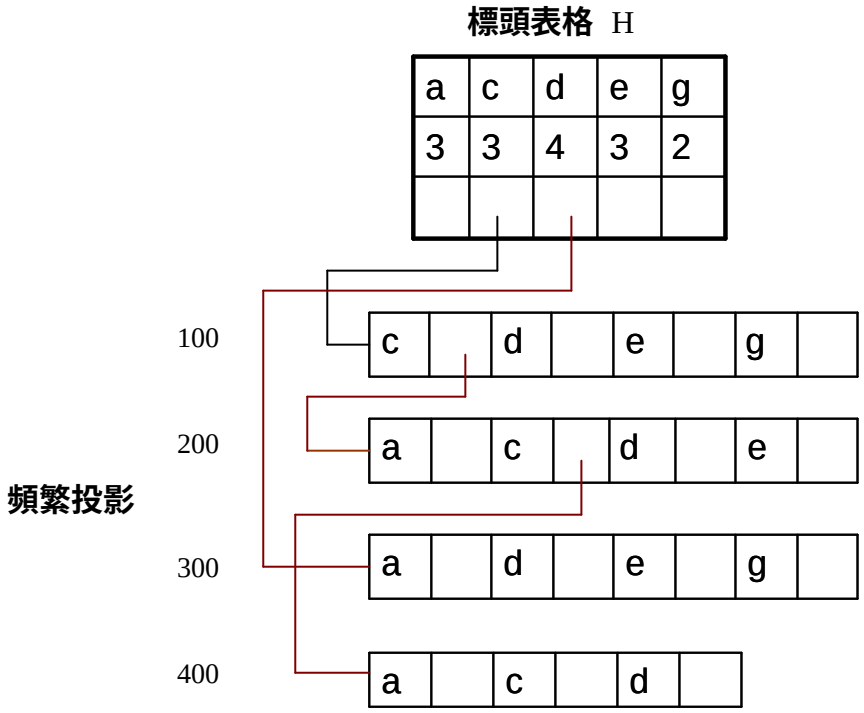


圖 2 - 2 - 7：調整探勘 a 投影資料庫後的連結位置[14]

依照探勘 a 投影資料庫的模式，探勘 c 投影資料庫。先找出頻繁項目中含有項目 c，但是不含有項目 a，因為與 a 項目有關的頻繁項目集已在 a 投影資料庫探勘完畢。建立標頭表格 H_c，並仿照上面遞迴的方式探勘出以 c 為首的頻繁項目集。接下來，依序探勘 d、e 投影資料庫，即可找出所有的結果頻繁項目集。

當交易中所有頻繁投影均可置入主要記憶體時，依照上述方法即可找出所有的結果頻繁項目集。但是，若是資料量龐大到無法置入主要記憶體時，就得將整個資料庫分為幾個子資料庫，分別利用 H-Mine(Mem) 探勘子資料庫，最後再整

合(Merge)結果，即 Jian Pei等學者所提出適合探勘大資料庫的 H-Mine[14]演算法。

H-Mine 演算法分為以下四個步驟。步驟一：先掃瞄過整個資料庫找出所有單一項目的頻繁項目集。步驟二：將整個交易資料庫的頻繁投影分成幾個子資料庫，讓每一個子資料庫所有交易的頻繁投影都可以置入主記憶體中。步驟三：將每個子資料庫的頻繁投影利用 H-Mine(Mem) 演算法找出符合各自最小支持度的頻繁項目集，各自最小支持度的算法為使用者設定整個資料庫的最小支持度乘以該子資料庫占整個資料庫的比例。步驟四：將每個子資料庫中的頻繁項目集(為區域頻繁項目集(Local Frequent Itemsets))及各項目集的支持度收集起來，掃瞄最後一次資料庫，確定區域頻繁項目集滿足整體最小支持度，得到整體頻繁項目集(Global Frequent Itemsets)。

H-Mine 與其他分割關聯演算法如 Partitioned Apriori [15]最大的不同在於 H-Mine 所需要的空間是可預測的，所需要的空間只有交易頻繁項目投影及標頭表格，空間負擔比 Apriori 所耗費的空間小許多。且 H-Mine 一開始就直接考慮整體頻繁項目，並不會耗費時間去運算那些不屬於整體頻繁項目但屬於區域頻繁項目的項目集，相反地，Partitioned Apriori 在分配不均的子資料庫中，會耗費許多時間計算及產生無用的候選項目集。

第三節 相關規則探勘之延伸問題

相關規則探勘延伸出各式各樣的問題，包括將項目分類加入探勘項目的廣義相關規則探勘[18][19]、依照項目特性設定不同支持度的多重最小支持度相關規則探勘[12][20]、相關規則更新[6][15] 及考慮交易項目序列順序的循序探勘[17]等等問題，其解決方法和基本問題息息相關，且可以應用在更多不同的領域。

以下簡介本論文會應用到的二種延伸問題：廣義相關規則探勘及多重最小支持度相關規則探勘。

2.3.1 廣義相關規則探勘(Generalized Association Rule Mining)

廣義相關規則是將項目分類 (Taxonomy) **的資訊加入相關規則探勘，一方面可將與分類有關的資訊反映在相關規則探勘上，不會只侷限在項目相關規則，使相關規則更有意義，另一方面還可得知不同階層間項目的關聯性。**

廣義相關規則的正規敘述[18][19]**如下：**

令 $I=\{i_1,i_2,\ldots,i_m\}$ **為一個文字符號組成的集合，每個文字符號稱為一個項目，由一個或一個以上的項目所組成的集合稱為項目集。令** $J=\{j_1,j_2,\ldots,j_p\}$**為由** I **延伸之廣義項目**(Generalized Items)**的集合。令** Γ **為由文字符號** I **及其廣義項目** J **所組成的非環狀有向圖，** Γ **上的邊線**(Edge)**代表著** is-a **的關係，而** Γ **代表分類** (Taxonomy) **的集合。如果在** Γ **上有一邊線從** p **指向** c**，我們則稱** p **為** c **的母體**(Parent)**，稱** c **為** p **的子體**(Child) **，也表示** p **為** c **的廣義延伸**(Generalization)**。如果在有遞移封閉性**(Transitive Closure) **的** Γ **上，有一邊線從** $\hat{x}$ **指向** x**，我們則稱** $\hat{x}$ **為** x **的祖先**(Ancestor)**，稱** x **為** $\hat{x}$ **的後裔**(Descendant)**。令資料庫** D **是由一群交易** T **所組成的集合，每個** T **為一項目集，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***。**

一個廣義相關規則(Generalized Association Rule)**表示成** $X \Rightarrow Y$**，其中** $X, Y \subseteq I \cup J$**，** $X \cap Y = \varnothing$ **，而且** Y **中項目並無包含** X **中任何項目的祖先。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%**。條件限制** Y **中並無包含** X **中任何項目祖先的原因是規則** "x=>ancestor(x)" **是保證** 100%**正確的，但是這樣的規則卻是已知且累贅的。廣義相關規則探勘定義為：給定交易記錄資料庫** D **及分類**Γ**，在當中找出所有支持度和確信值大於最小支持度跟最小確信值的廣義相關規則，其中最小支持度與最小確信值的門檻值由使用者給定。**

廣義相關規則探勘，直覺地來說，將交易中的項目加上分類項目，再利用探勘相關規則演算法探勘即可。如圖 2 - 3 - 8，Srikant **等學者所提出的基本演算法**[18]，**將交易中每個項目的廣義項目加入交易，並移除同筆交易中重複出現的廣義項目，而加入廣義項目的交易稱為延伸交易**(Extended Transactions) **。最直覺的探勘方法即是直接將延伸交易利用相關法則演算法找出廣義相關法則。**

```
L1 := {frequent 1-itemsets};
k:=2; //k represents the pass number
while(Lk ≠ f ) do
   Ck := New candidate of size k generated from Lk-1.
   for all transactions t ∈ D do begin
        Add all ancestors of each item in t to t, removing any duplicates.
        Increment the count of all candidates in Ck that are contained in t.
    End.
    Lk := All candidates in Ck with minimum support.
   k := k+1;
End while
Result = ∪k Lk;
```

圖 2 - 3 - 8：**廣義相關規則基本演算法**[18]

廣義相關規則基本演算法的第一步驟，先找出長度為 1 **的頻繁項目集。其中包括分類中的葉節點**(Leaf Node，**即一般項目**)**及內部節點**(Internal Node，**即廣義項目**)。**只要一般項目及廣義項目的支持度大於使用者定義的最小支持度都是屬於頻繁項目集。之後，就如同** Apriori [3]**的步驟，每次疊代利用前次頻繁項目集找出候選項目集，並且掃瞄每筆交易及其對應的廣義項目**(**即延伸交易**)**，找出各個項目的支持度。**

然而，廣義相關規則基本演算法並不夠有效率，可再利用廣義關聯法則及分類的特性進一步地改善廣義關聯演算法，因此 Srikant **等學者還提出三個可以使基本演算法最佳化的規則，發展出累積演算法**(Cumulate)[18]。

累積演算法的三個最佳化規則：

1. 過濾掉要加到交易中的廣義項目。沒有必要把分類中所有交易項目的廣義項目加入交易裡。相反地，我們只需要將在疊代的某一輪(Pass) 有用到的候選項目的廣義項目加入交易中即可。事實上，如果原本的項目並不在任何頻繁項目集中出現，該項目亦可直接從交易中刪除。舉個例子說明，假設"外套" 的母體是"外衣"， 而"外衣"的母體是"衣服"。令{衣服, 鞋子}為唯一滿足最小支持度的項目集。之後，若任何交易中有包含"外套"，以"衣服"取代"外套"。我們不需要在交易中保留"外套"這個項目，也不需要在交易中加入"外衣"這個項目。

2. 預先計算廣義項目。與其在每次追蹤每一個項目時就瀏覽整個分類階層，可以預先計算每一個項目的廣義項目集合。同時，我們也可以從分類階層中丟棄那些從未出現在候選項目集的廣義項目。

3. 刪除項目集中同時包含一個項目及由該項目延伸而來的廣義項目。

利用以上的三個最佳化規則，再配合原本的廣義相關規則基本演算法，即是廣義相關規則累積演算法。演算法如圖 2 - 3 - 9。粗體字是與基本演算法相異的部分，即是應用最佳化規則改善演算法的部分。

Compute Γ^*, the set of ancestors of each item, from Γ. //Optimization 2
L_1 := {frequent 1-itemsets};
k:=2; //k represents the pass number
while($L_k \neq f$) do begin
  C_k := New candidate of size k generated from L_{k-1}.
  If (k=2) then
    Delete any candidate in C_2 that consists of an item and its ancestor.
    // Optimization 3
  Delete any ancestors in Γ^* that are not present in any of the candidate in C_k; // Optimization 1
  for all transactions $t \in D$ do begin
    Add all ancestors of each item in t to t, removing any duplicates.
    Increment the count of all candidates in C_k that are contained in t.
  End.
  L_k := All candidates in C_k with minimum support.
  k := k+1;
End while
Result = $\cup_k L_k$;

圖 2 - 3 - 9：**廣義相關規則累積演算法**(Cumulate) [18]

2.3.2 **多重最小支持度相關規則探勘**(Association Rule Mining with Multiple Minimum Supports)

決定相關法則實用與否的關鍵在於最小支持度設定的適當與否。一般的相關規則探勘都是在單一支持度下產生規則，然而只用一個最小支持度並無法表示所有不同特性項目的支持度。因此，Liu [12]**等學者提出為不同特性的項目訂定不同支持度的想法，並設計一個架構在** Apriori **上的多重最小支持度相關規則探勘演算法，以符合現實情況之需求。**

例如在超級市場的交易資料中，有些項目集雖然支持度較低(亦即較不常被購買) ，但是可以產生相當高的利潤，若要把此類相關規則找尋出來，則必須將最小支持度設定得比較低，以利產生有用的規則，如：

食物處理器 ⇒ 烹調平底鍋 (支持度 = 0.5%，**確信值** = 60%)

但是假如所有項目都訂定同一的最小支持度，則下列沒有意義的規則也會產生：

麵包，起司 ⇒ 牛奶 (支持度 = 5%，確信值 = 68%)

知道 5%顧客一起買此三種食品是沒有用的，因為這些食品在超級市場同時被買的機率是十分頻繁，這樣的規則太繁瑣，並非是使用者會想要知道的。若要使這類的規則變得非常有用，則需要將最小支持度定得比較高才有意義，但是相對地會無法產生“ 食物處理器 ⇒ 烹調平底鍋 ”此類相關規則。由此例可以發現，必須針對項目的特性設定不同的支持度，才能產生有意義且適用的相關規則。

多重最小支持度相關規則的定義和相關規則的差別在於前者修改了最小支持度的定義。在多重最小支持度相關規則中，一個規則的最小支持度為出現在該規則內所有項目的最小支持度之最小值。每一項目的最小支持度稱為最小項目支持度(Minimum Item Supports)，簡寫為 MIS。將經常出現的項目設定高的最小支持度，將罕見但價值高的規則設定低的最小支持度，可以使規則更符合現實需求。

多重最小支持度相關規則探勘演算法的精神在於 Liu [12]等學者歸納出多重最小支持度規則的特性：排序封閉的特性(Sorted Closure Property)，此一特性是從 Apriori 向下封閉的特性(Downward Closure Property) 延伸而來。Apriori 向下封閉的特性是指若一個項目集滿足最小支持度，那麼它所有的子集也都會滿足最小支持度。但是向下封閉的特性並不能適用於多重最小支持度相關規則探勘。例如資料庫中有四個項目：1, 2, 3, 4；各自的最小支持度分別為 MIS(1)=10%, MIS(2)=20%, MIS(3)=5%, MIS(4)=6%。如果我們找出{1,2}的支持度為 9%，由於並不滿足 1 或 2 的最小支持度，因此{1,2}不屬於頻繁項目集，對 Apriori 而言，項目集{1,2,3}及{1,2,4}也會跟著不可能成為頻繁項目集，但是事實上項目集{1,2,3}及{1,2,4}的最小支持度分別為 5%及 6%，是符合頻繁項目集的。因此，Liu 提出了排序封閉的特性應用在刪除候選項目集上。

多重最小支持度相關規則演算法的關鍵在於探勘規則時必須將所有項目依項目最小支持度遞增排列。如上例，四個項目的最小支持度分別為 MIS(1)=10%, MIS(2)=20%, MIS(3)=5%, MIS(4)=6%**，就得排列成** 3, 4, 1, 2**。這樣的排列有助於解決不符合** Apriori **向下封閉的特性。**

令 L_k **表示頻繁** k **項目集的集合。每一個項目集** c **的表示法為，**<c[1], c[2], … ,c[k]> **，且** MIS(c[1])≤MIS(c[2])≤ … ≤MIS(c[k])**。** Liu[12] **提出多重最小支持度相關規則演算法** MSappriori **如圖** 2 - 3 - 10(a)**。**

```
M=sort(I, MS); //according to MIS(i)'s stored in MS
F=init-pass(M, T); //make first pass over database T, only keep frequent 1-itemsets
L1 := {<f>|f∈ F, f.count ≥ MIS(f)};
for(k=2;Lk ≠ f ;k++) do // k represents the pass number
   if(k=2) then C2=level2-candidate-gen(F)
   else Ck=candidate-gen(Lk-1)
   end
   for each transaction t ∈ T do begin
        Ct=subset(Ct,t);
      for each candidate c∈ Ct do c.count++;
   End.
   Lk := {c∈ Ck |c.count ≥ MIS(c[1])};
   //itemset must larger than the minimum MIS
end for
Result = ∪k Lk;
```

圖 2 - 3 - 10 (a)**：多重最小支持度相關規則探勘演算法**[12]

首先，將所有項目依最小項目支持度遞增排列，存成 MS**。掃瞄一次資料庫，找出長度為** 1 **的候選項目集** F **及頻繁項目集** L_1**，其中候選項目集** F **的每個項目都必須符合最小項目支持度之最小值** minMIS **(以** minMIS **表示所有最小項目支持度之最小值)，而頻繁項目集** L_1 **的每個項目都必須符合各自的最小項目支持度。類似** Apriori **的步驟，產生候選項目集。當欲產生長度為** 2 **候選項目集時，必須利用長度為** 1 **的候選項目集，即還沒經過各自最小項目支持度測試的項目集**

F，來找出長度為 2 候選項目集，以避免錯失一些項目集。

例如第一次掃瞄 100 筆的資料庫得到 3.count=6, 4.count=3, 1.count=9 及 2.count=25 。而 MIS(1)=10%, MIS(2)=20%, MIS(3)=5%, MIS(4)=6%, minMIS=5%。因此，依序排列，F={<3>, <1>, <2>}，而 L_1={<3>,<2>}。因為 4.count 小於 minMIS，所以不在 F 中。而 1 不在 L_1 中是因為 1.count 小於項目 1 的最小支持度 10%。而經由候選項目集 F，得到 C_2={<3,1>,<3,2>}。(其中<1,2>不屬於 C_2 是因為<1,2>支持度只有 9，而項目集的最小支持度為 10%。) 若是經由 L_1 產生 C_2，則會因不適用 Apriori 的向下封閉特性，而失去<3,1>這個亦符合候選資格的項目集。

產生 C_2 候選項目集 level2-candicate-gen(F) 步驟如圖 2 - 3 - 11(b)。

```
for each item f in F in the same order do
    if f.count ≥ MIS(f) then
      for each item h in F that is after f do
            if h.count ≥ MIS(f) then
              insert <f, h> into C2
```

圖 2 - 3 - 12 (b)：產生 C_2 候選項目集 level2-candicate-gen(F) 步驟[12]

至於產生其他候選項目集 candicate-gen(L_{k-1})的方式，則是類似 Apriori 的 apriori-gen 方式分為二步驟，連結(Join)與刪除(Prune)。連結步驟與 Apriori 的連結步驟完全相同，至於刪除步驟，則是利用排序封閉特性(Sorted Closure Property)。刪除步驟的目的在於刪除一些已知不可能符合最小支持度的項目集，因此，依據向下封閉特性，必須刪除項目集中其子集不屬於候選項目集的項目集。但是，為了符合多重最小支持度的特性，有個情形是例外不能刪除的，就是當該子集內不含有項目集的第一個項目時，是不需要刪除的。這是因為第一個項目代表著項目集的最小支持度，而當其子集不在候選項目集時，除了最小的二個項目最小支持度相等(MIS(c[1])=MIS(c[2]))，我們無法確定它是否會不滿足最小

項目的支持度 MIS(c[1])**，只能確定該子集不滿足** MIS(c[2]) (MIS(c[1])< MIS(c[2]))**。**

例如令 L_3 **為**{<1,2,3>, <1,2,5>, <1,3,4>, <1,3,5>, <1,4,5>, <1,4,6>, <2,3,5>}**，項目在項目集內已排序過。經過連結步驟，得到** C_4={<1,2,3,5>, <1,3,4,5>, <1,4,5,6>}**。項目集**<1,4,5,6>**會在刪除步驟裡被刪除，因為**<1,5,6>**不屬於候選項目集，且**<1,5,6>**包含該項目集的最小項目** 1**。而**<1,3,4,5>**卻不會被刪除，雖然**<3,4,5>**不屬於候選項目集，但是其最小支持度為** MIS(3)**，最小支持度可能會大於整個項目集**<1,3,4,5>**的最小項目支持度** MIS(1)**，而我們無法確定**<3,4,5>**是否會大於** MIS(1) **，所以除非我們知道項目** 1 **與** 3 **的項目最小支持度相等，否則我們就不刪除候選項目集**<1,3,4,5>**。**

產生 C_k ($k \neq 2$)**候選項目集的刪除步驟如圖** 2 - 3 - 13(c)**：**

```
for each itemset c ∈ Ck do
    for each (k-1)-subset s of c do
        if(c[1] ∈ s) or (MIS(c[2])= MIS(c[1])) then
            if(s ∉ Ck-1) then delete c from Ck;
```

圖 2 - 3 - 13 (c)**：產生** C_k ($k \neq 2$)**候選項目集的刪除步驟**[12]

第三章 以 H-Mine 為基礎之廣義相關規則演算法

本論文針對圖書館借閱記錄的特性，選擇適合的演算法 H-Mine 來探勘圖書館借閱記錄，期望能夠找出借閱館藏的關聯性，並且讓館員能夠針對不同身份不同系所的讀者，找出不同的相關規則。此外，本論文亦將廣義相關規則探勘及多重最小支持度相關規則探勘的觀念與 H-Mine 演算法結合，提出 H-Mine(Generalized) 及 H-Mine(MMS) 演算法，以探勘借閱類別的關聯性。

在本章中，第一節介紹我們所提出的廣義相關規則演算法 H-Mine(Generalized)；第二節說明如何加入多重最小支持度的概念探勘廣義相關規則，進而提出多重最小支持度廣義相關規則演算法 H-Mine(MMS)。

第一節 廣義相關規則演算法 H-Mine(Generalized)

H-Mine(Generalized) 演算法是根據 H-Mine 演算法結合廣義相關演算法的概念而提出的延伸演算法。主要動機是因為在探勘圖書館借閱記錄時，由於探勘所得的讀者借閱關聯館藏只佔了微乎其微的一小部分，對館藏龐大的圖書館而言效益不大，因此本論文進一步地找出讀者借閱類別的關聯性，藉由推薦讀者有興趣關聯類別的新書，更有效地提供讀者借閱建議。

H-Mine(Generalized) 演算法是將交易資料中的所有項目根據其隸屬分類階層將分類項目(即廣義項目)加入每筆交易資料，成為延伸交易，再將延伸交易利用改進 H-Mine 演算法找出廣義相關法則。詳述步驟如下：

步驟一：根據項目隸屬分類階層加上其廣義項目，得到延伸交易。例如以圖書館中文館藏為例，若採用「中國圖書分類法」做為分類階層，若有本書的分類號為 448.82，則將含有該本書的交易加上分類廣義項目 4XX、44X、448、448.8 及 448.82。

步驟二：將交易的每一項目，包括原本交易項目及廣義項目，都當成一般項目，找出長度為 1 **的單一頻繁項目集，並據以建立** H-Struct **資料結構的標頭表格** H，H **中每個項目包含三個欄位：項目編號、項目支持度及連結位置，其中頻繁項目集的順序為任意順序。**

步驟三：刪除多餘項目，重新調整頻繁項目集，找出新標頭表格 H'**。因為本方法將廣義項目當成一般項目運作，這樣一來分類項目的子體**(Child)**與母體**(Parent)**或是後裔**(Descendant)**與祖先**(Ancestor)**就可能會同時存在標頭表格內，若是子體(後裔)與母體(祖先)的支持度又是相同，母體(祖先)所代表的意義就已經隱含在子體(後裔)中，則母體(祖先)則成了多餘的頻繁項目，沒有必要再保留。重新調整頻繁項目集的方法是掃瞄** H **中的所有頻繁項目，若是有二個頻繁廣義項目皆屬於同一大類分類項目，則判斷是否有一項目為另一項目的母體(祖先)。若是，又二者支持度相同，則刪除母體(祖先)項目，表示母體(祖先)資訊已經隱含在子體(後裔)中，則沒有必要再保留母體(祖先)。**

步驟四：將資料庫中每筆交易項目只保留經過調整的所有頻繁項目，包含原本交易項目及廣義項目，即新標頭表格 H' **內的頻繁項目集，並依照頻繁項目集順序排列，得到每筆交易的頻繁投影。如同** H-Mine**，每一個存在** H-Struct **中的頻繁項目亦包含二個欄位：項目編號及連結位置。**H-Struct **即是包含標頭表格及頻繁投影的資料結構。**

步驟五：將新標題表格 H' **的每一項目，找出每個頻繁投影的第一個項目與之對應連結，得到完整** H-Mine **的** H-Struct **資料結構。**

步驟六：由於標頭表格內仍有母體(祖先)與子體(後裔)同時存在，但二者支持度卻不相同的情形，為確保在同一個結果頻繁項目集內不會有母體(祖先)及子體(後裔)同時出現的情形，本步驟將要輸出的結果頻繁項目集進行最後測試。利用之前步驟三調整頻繁項目集的方法，判斷每一個頻繁項目集中的所有廣義項目

是否有其他廣義項目為其子體(後裔)的情形，確認每一結果頻繁項目集皆是最精簡的。最後則輸出所有精簡後的頻繁項目集。

步驟七：類似 H-Mine **方法，標頭表格** H' **的每個項目可各自組成一個投影資料庫。在每個投影資料庫中重複執行二、三、五及六的步驟：找出頻繁項目集，得到各自的標頭表格** H**、經由分類測試後調整成為** H'**標頭表格、根據標頭表格重新與頻繁投影連結、測試結果頻繁項目集並輸出精簡後的頻繁項目集、再遞迴一層層深入探勘，直到所有標頭表格的頻繁項目集完全找完。**

本論文中我們所提出的 H-Mine(Generalized)**演算法，是利用** H-Mine **演算法，再加上二個最佳化的條件，一是調整標頭表格中母體(祖先)與子體(後裔)同時出現，且支持度又相同的項目，刪除母體(祖先)的項目，只保留子體(後裔)的項目；另一則是，在印出結果頻繁項目集時，必須測試同一結果頻繁項目集內的所有廣義項目是否有子體(後裔)包含母體(祖先)的情形，確定頻繁項目集是最精簡的。**H-Mine(Generalized)**演算法如下，其中粗體字則是改變原本演算法的部分。**

```
Add generalized items to transactions. //Step1
H := {frequent 1-itemsets}; //Step2: get Header Table H
H' := re-adjust H; //Step3: delete redundant itemsets in H
Construct frequent projections according to H'. //Step4
Link frequent projections and H' . //Step5
Mine(H' );
Function Mine(itemset H)
{
    For each itemset i∈ H do begin //Step6
        Delete any item in itemset i that consists of a taxonomy item and its ancestor.
        Print out final frequent itemset i.
    End.
    Traverse_projected_db(H); //Step7
}
Function Traverse_projected_db(itemset I)
{
    For each itemset i∈ I do begin
        Traverse i as i-projected database
        Generate new Header Table Hi.
        Re-adjust Header Table, get Hi'.
        //Delete any item in Hi that which next item is the descendant of this item
        Link frequent projections and Hi'.
        Mine(Hi');
    End.
}
```

第二節 多重最小支持度廣義相關演算法 H-Mine(MMS)

H-Mine(MMS)**演算法是根據** H-Mine **演算法結合廣義相關規則演算法與多重最小支持度演算法的概念，提出的延伸演算法。由於我們想要找出讀者借閱館藏與類別的關聯性，但是因為在所有分類階層設定相同的支持度，不足以展現出分類的效果，因此，提出** H-Mine(MMS) **演算法，將分類法的分類階層作進一步設定，針對不同階層的類別給予不同最小支持度，探勘最適宜的相關規則。**

H-Mine(MMS)演算法是利用 H-Mine 演算法的 H-Struct 資料結構，增加一個儲存該項目最小支持度的欄位，再設定各標頭表格探勘時的最小支持度，改進成為適用於多重最小支持度的廣義相關規則演算法。詳述步驟如下：

步驟一：根據項目隸屬分類加上各階層分類廣義項目，得到延伸交易。

步驟二：掃瞄整個交易資料庫，找出長度為 1 的頻繁項目候選集，即項目支持度必須大於所有項目的最小項目支持度之最小值(minMIS)。此頻繁項目集即為 H-Struct(MMS)的標頭表格 H。H-Struct(MMS)標頭表格的項目包含四個欄位：項目編號、項目支持度、最小項目支持度(MIS: Minimum Item Support)及連結位置。

步驟三：刪除多餘廣義項目，重新調整頻繁項目集，找出新標頭表格 H_{re}。

步驟四：將標頭表格 H_{re} 中的每一個頻繁項目之最小項目支持度值存入欄位，並將標頭表格依照各項目的最小項目支持度遞增排列，得標頭表格 H'。

步驟五：將資料庫中每筆交易只保留經過調整的頻繁項目，即標頭表格 H' 內的頻繁項目集，並按頻繁項目集順序排列，得到每筆交易的頻繁投影。每一個存在 H-Struct(MMS)中的頻繁項目包含二個欄位：項目編號及連結位置。H-Struct(MMS)即是包含標頭表格及頻繁投影的資料結構。

步驟六：將標頭表格 H'，找出每個頻繁投影的第一個項目與之對應連結，則得到完整 H-Mine(MMS)的 H-Struct(MMS)資料結構。

步驟七：由於標頭表格 H'內仍有母體(祖先)與子體(後裔)同時存在，但二者支持度卻不相同的情形，為確保在同一個結果頻繁項目集內不會有母體(祖先)及子體(後裔)同時出現的情形，本步驟將要輸出的結果頻繁項目集進行刪減，確定頻繁項目集是最精簡的。最後則測試是否符合各自的最小項目支持度，輸出標頭表格中所有符合的頻繁項目集。

步驟八：類似 H-Mine(Generalized)方法，標頭表格 H' 的每個項目可各自組

成一個投影資料庫。在每個投影資料庫中重複執行二、三、四、六及七的步驟：找出頻繁項目集，其中各標頭表格探勘時的最小支持度為隸屬於各投影資料庫項目的最小項目支持度(如：探勘 a 投影資料庫時，最小支持度為 a 的最小項目支持度；探勘 bc 投影資料庫時，最小支持度為 b 的最小項目支持度)，得到各自的標頭表格 H、經由分類測試調整及重新排序成為 H'標頭表格、根據標頭表格重新與頻繁投影連結、測試頻繁項目集找出精簡且符合各自最小項目支持度的結果頻繁項目集、再遞迴一層層深入探勘，直到所有標頭表格的頻繁項目集完全找完。

以下舉例說明 H-Mine(MMS)，為求精簡起見，本例中不將分類廣義項目加入探勘，直接說明如何應用多重最小支持度的概念。下表前二欄是假設的交易資料庫 TDB，項目 a, b, c, d, e, f, g, h, i, k, m 的項目最小支持度分別為 3, 2, 3, 4, 4, 2, 2, 2, 2, 2, 2。

交易編號	項目集	頻繁投影	依最小項目支持度排序之頻繁投影
100	c, d, e, f, g, i	c, d, e, g	g, c, d, e
200	a, c, d, e, m	a, c, d, e	a, c, d, e
300	a, b, d, e, g,	a, d, e, g	g, a, d, e
400	a, c, e, h	a, c, e	a, c, e

表 3 - 2 - 1：多重最小支持度相關規則探勘之交易資料庫

首先，步驟一，掃瞄資料庫找出符合最小項目支持度之最小值 2 的單一項目頻繁項目集，得到{a:3, c:3, d:3, e:4, g:2}。步驟二：將標頭表格所有項目依各最小項目支持度遞增排列，則得到{g:2, a:3, c:3, d:3, e:4}，即是 H-Struct(MMS) 的標頭表格 H。步驟三：經由項目最小支持度測試，由於項目 d 不符合所要求的最小項目支持度，故單一結果頻繁項目集為{g:2, a:3, c:3, e:4}。步驟四：找出頻繁投影並將標頭表格與之連結，如圖 3 - 2 - 1，標頭表格 H 的項目 g 即是存放 g 佇列第一個項目的指標(指向交易投影 100 中第一個項目 g 的指標)，g 佇列則連結 100 及 300 二筆交易的投影。

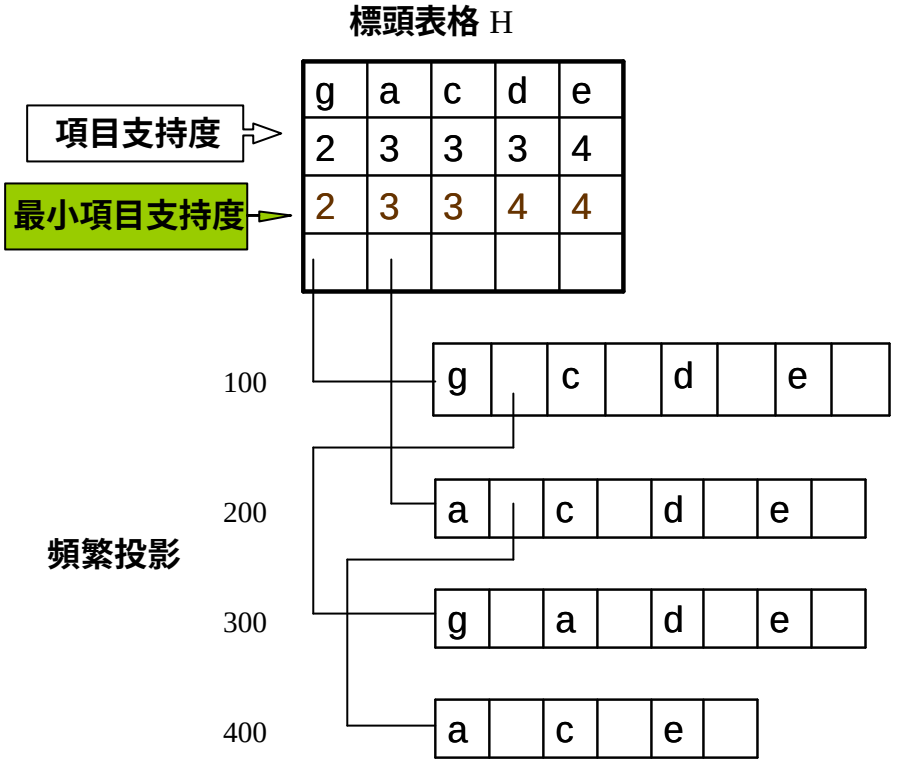


圖 3 - 2 - 1：H-Struct(MMS)範例

如圖 3 - 2 - 1，由於 d 不滿足項目最小支持度，d 投影資料庫的其他項目集亦不可能滿足項目最小支持度，因此探勘經由標頭表格 H 的除 d 外的四個項目，分為四個子集，可視為四個虛擬投影資料庫，在各自的子集中探勘，找出屬於個別子集的頻繁項目集。

先從第一個子集 g 投影資料庫（屬於 g 佇列的頻繁投影）開始，首先掃瞄整個 g 投影資料庫，建立屬於 g 投影資料庫的標頭表格 H_g。標頭表格 H_g 的欄位就如同標頭表格 H 的四個欄位。但是，項目支持度是存放頻繁項目在 g 投影資料庫中出現的次數，且標頭表格 H_g 的最小支持度為 g 項目最小支持度 MIS(g)。建立 g 投影資料庫的標頭表格 H_g{d:2, e:2}，輸出部分結果頻繁項目集{gd:2, ge:2}，並建立 H_g 與頻繁投影的連結，如圖 3 - 2 - 2。對 g 投影資料庫而言，交易投影 100 及 300 的第一個項目均為 d，因為 a 與 c 在 g 投影資料庫中並非頻繁項目。標頭表格 H_g 中項目 d 的連結位置指向 100，100 項目 d 的連結位置指向 300，成為 gd 佇列。

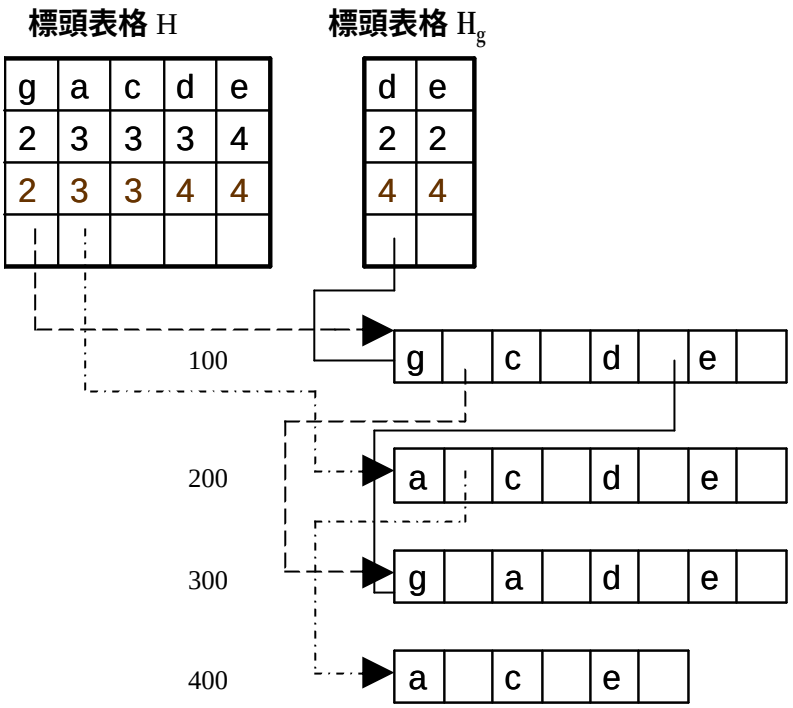


圖 3 - 2 - 2：標頭表格 Hg 及 gd 佇列

同樣地，探勘 gd 投影資料庫中的頻繁項目時，利用 gd 佇列的連結瀏覽 gd 投影資料庫，並建立屬於 gd 投影資料庫的標頭表格 H_{gd}，最小支持度亦是 g 項目最小支持度 MIS(g)，如圖 3 - 2 - 3。標頭表格 H_{gd} 中只有項目 e 是頻繁項目，因此只有 gde:2 是結果頻繁項目集。搜尋 gd 為首的頻繁項目集就此結束。

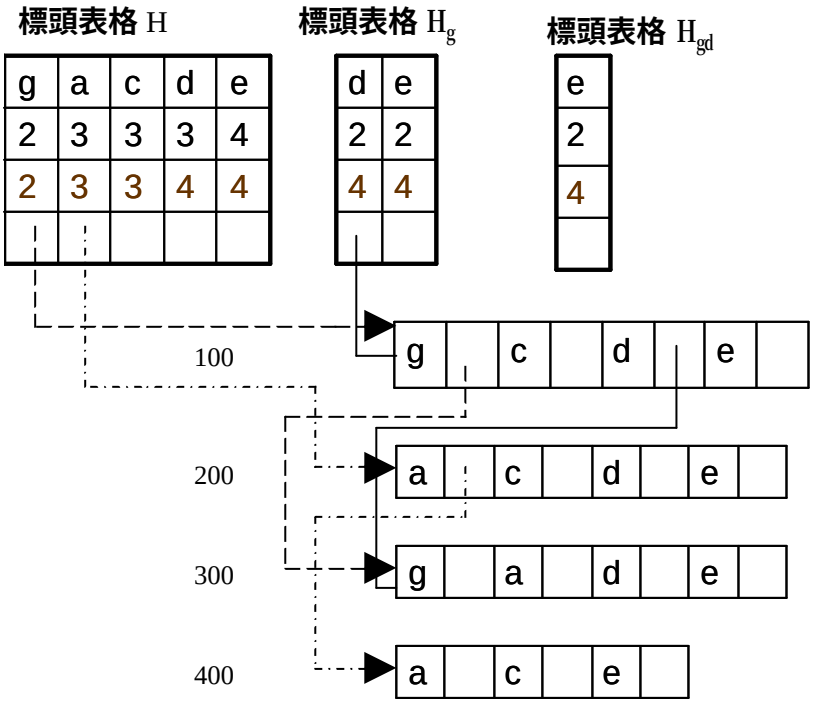


圖 3 - 2 - 3：標頭表格 H_{gd}

結束以 gd 為首頻繁項目集的搜尋後，回到標頭表格 H_g，因為 e 為 H_g 的最後一個項目，並不會有以 e 為首的項目集，搜尋 H_g 亦就此結束。

結束所有以 g 為首頻繁項目集的搜尋後，回到標頭表格 H，追蹤 a 頻繁項目集的搜尋，此時 a 佇列只包含了所有第一個項目為 a 的頻繁投影，但是 g 佇列中仍有同時包含項目 g 及 a 的頻繁投影，那些在 g 佇列中的投影亦有可能會有以 a 為首的頻繁項目。因此，必須再追蹤 g 佇列的連結，將屬於 g 佇列的每個頻繁投影中項目 g 之後的項目，即第二個頻繁項目(第一個項目均為 g)，與各自歸屬的佇列連結起來。如圖 3 - 2 - 4，頻繁投影 gcde 插入 c 佇列，而頻繁投影 gade 插入 a 佇列。

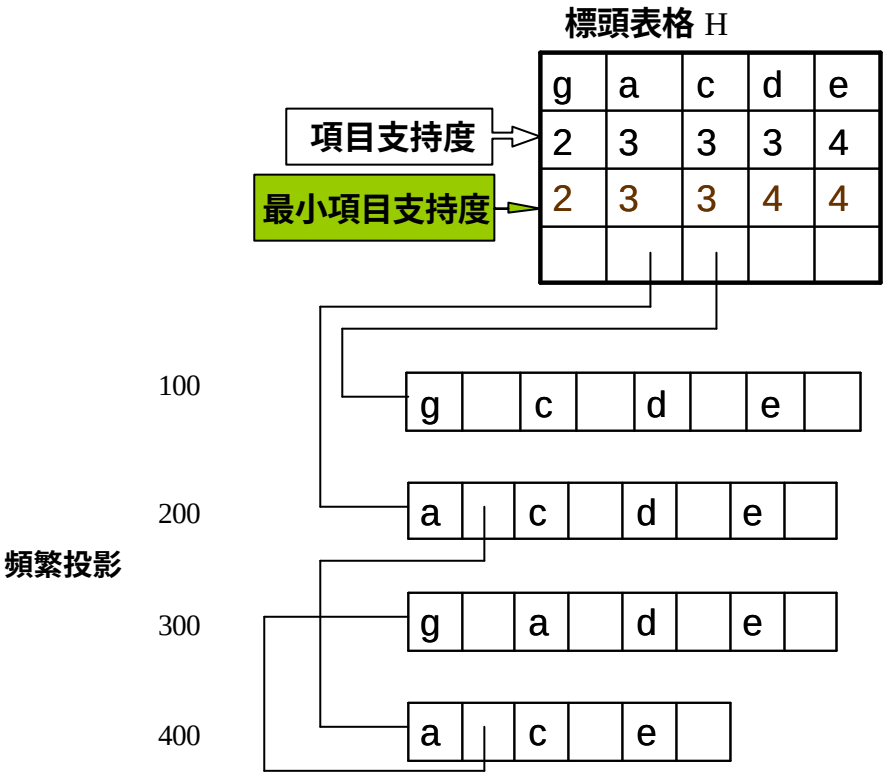


圖 3 - 2 - 4：調整探勘 g 投影資料庫後的連結位置

依照探勘 g 投影資料庫的模式，探勘 a 投影資料庫。先找出頻繁項目中含有項目 a，但是不含有項目 g，因為與 g 項目有關的頻繁項目集已在 g 投影資料庫探勘完畢。建立標頭表格 H_a，並仿照上面遞迴的方式探勘出以 a 為首的頻繁項目集。接下來，依序探勘 c 投影資料庫(不需探勘 d 投影資料庫是因為 d 並不滿

足項目最小支持度，其他以 d 為首的項目集亦不可能符合，至於 e 投影資料庫不需要探勘是因為 e 為最後一個項目，並不會有以 e 為首的項目集)，即可找出所有的結果頻繁項目集。

H-Mine(MMS)**演算法如下，而粗體字則是改變原本演算法的部分。**

```
Add generalized items to transactions. //Step1
H := {frequent 1-itemsets}; //Step2: H 的支持度為所有項目最小支持度的最小值
H_re := re-adjust H; //Step3: delete redundant itemset in H
H' := sort H_re according to each item of MIS in ascending order; //Step4
Construct frequent projections according to H'. //Step5
Link frequent projections and H' . //Step6
Mine(H' );
Function Mine(itemset H)
{
    For each itemset i∈ H do begin //Step7
        Delete any item in itemset i that consists of a taxonomy item and its ancestor.
        If the support of Itemset i is larger than the minimum item supports of all
        items in that itemset,
            Print out frequent itemset i.
    End.
    Traverse_projected_db(H); //Step8
}
Function Traverse_projected_db(itemset I)
{
    For each itemset i∈ I do begin
        Traverse i as i-projected database
        Generate new Header Table H_i. //支持度為該項目的最小支持度，MIS(i)
        Re-adjust Header Table, get H_i'.
        //Delete any item in H_i that which next item is the descendant of this item
        Link frequent projections and H_i'.
        Mine(H_i');
    End
}
```

第四章 圖書館借閱記錄探勘系統之實作

在本章中，我們將介紹以第三章演算法為基礎設計而成的系統—圖書館借閱記錄探勘系統(A Data Mining System for Mining Borrowing Library History Records)，簡稱圖書館資料探勘系統(Library Borrowing History Records Mining System)。

本系統所提供的主要功能為：

- 讓館員藉由讀者借閱記錄可得到最新的館藏借閱相關規則。
- 針對不同系所的讀者探勘，探勘不同的相關規則。
- 應用「中國圖書分類法」找出讀者借閱關聯類別。
- 結合「中國圖書分類法」，在分類階層中設定不同的支持度探勘相關規則。
- 將結果轉換成封閉式頻繁項目集[12,20,21,22]，並導入交通大學個人化數位圖書資訊環境 (PIE@NCTU) 系統[25]中，結合 PIE@NCTU 的智慧型查詢及個人借閱歷史記錄將關聯館藏推薦給讀者。

在本章中，第一節介紹系統運作流程、系統建置需求及系統功能。第二節說明在 PIE@NCTU 個人化系統中結合資料探勘的結果。

第一節 圖書館資料探勘系統說明

4.1.1 系統流程

本系統是以圖書館的借閱歷史記錄為資料來源，先經過預備資料的過程，包含資料清理、資料轉換，再經由資料探勘，利用關聯演算法找出相關規則。最後，將所得到的規則導入個人化數位圖書資訊環境 PIE@NCTU，針對個別的讀者需求適時提供借閱建議。系統流程如圖 4 - 1 - 1。

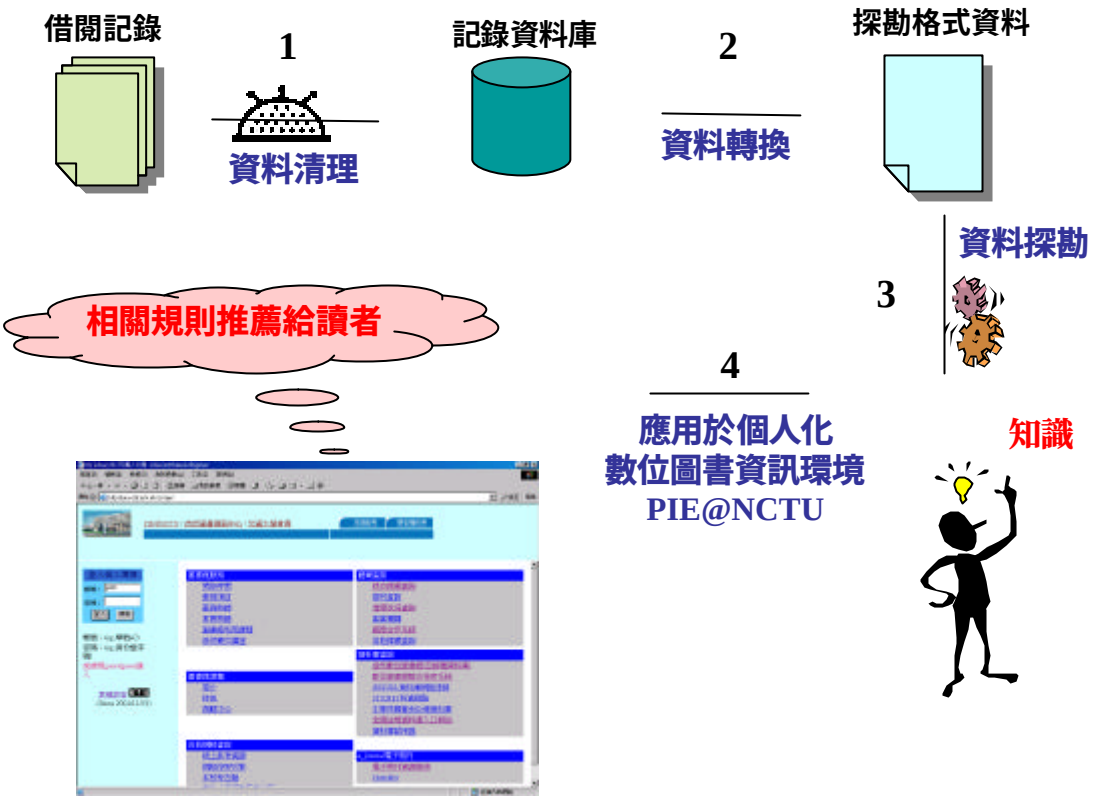


圖 4 - 1 - 1：圖書館資料探勘系統流程圖

■ 借閱記錄資料格式

本系統提供館員將符合格式的借閱記錄存入借閱記錄資料庫 Mining_Lib，便於探勘相關規則。借閱記錄資料的欄位包括讀者證號、身份類別號、系所代碼、性別、書目識別號、交易日期、登錄號、館藏登錄日期、索書號、書名十個欄位。每筆記錄以四行表示，格式如下：

讀者證號、身份類別號、系所代碼、性別、書目識別號、交易日期
登錄號、館藏登錄日期
索書號
書名

表 4 - 1 - 1：借閱記錄檔格式

借閱記錄範如表 4 - 1 - 2：

xxxxxxxx A 317 M 141755 Jan 2 2001 12:00AM C97310 May 7 1999 12:00AM 083.6 1154 v.18 **小畢的故事**
xxxxxxxx A 500 F 315427 Jan 2 2001 12:00AM X238388 Jul 13 2000 8:22AM 522.1 6691 **如何考上國小老師**
xxxxx C 610 202671 Jan 2 2001 12:00AM X126680 May 8 1999 12:00AM 550.1 7472 **經濟學概論**
xxxxxxxx B 216 M 334440 Jan 2 2001 12:00AM F303188 Nov 17 2000 3:15PM NA1455.F53 A 223 1995 v.3 Alvar Aalto. Band III, projekte und letzte bauten = Vol.III, projects et dernieres oeuvres /

表 4 - 1 - 2：借閱記錄範例

■ **預備資料**

本論文第二章中提到，在完整的資料探勘處理過程中，預備資料是最耗時的，且預備資料對於資料探勘結果的優劣影響甚鉅。在此，預備資料包含了資料清理及資料轉換二個步驟。

◆ **資料清理：將取得的借閱記錄資料存入交易資料庫中。將每筆記錄中十個欄位依序取出，存入對應的資料庫表格欄位。**

◆ **資料轉換：依照不同的相關規則演算法，將資料轉換成適合的模式。**

● **相關規則演算法**

所要找的相關規則是讀者常一起借閱的館藏，因此將讀者借閱館藏的書目識別號視為一個項目，而每位讀者於一段時間內(可由館員自行設定，預設為一年)借閱的館藏之書目識別號所成的集合視為一筆交

易，這一群交易所組成的集合便成為我們的交易資料庫。舉例來說，若將期間設為一年，圖書館中有二位讀者“A”及“B”，A 先借了書目識別號為“1343”及“253”二本書，半年後借“3466”及“96893”二本書，再一年後，A 又借了“3423”、“34636”及“9689”三本書，則 A 的交易有{1343, 253, 3466, 96893}和{3423, 34636, 9689}二筆交易。若 B 在一年內借了“3423”、“34656”及“9689”三本書，則 B 的交易有{3423, 34656, 9689}。

- 廣義相關規則演算法及多重最小支持度廣義相關規則演算法

由於廣義相關規則演算法及多重最小支持度廣義相關規則演算法均是探勘館藏資源及其分類廣義項目，故二者的資料轉換是完全相同的。而廣義相關規則與相關規則的最大不同在於廣義相關規則多加了項目的類別資訊，因此廣義相關演算法的資料轉換乃是先依照相關規則演算法的資料轉換方式，找出每筆交易的項目集，並且將交易內的每個項目依照使用者要求加入不同深度的類別資訊。

依照交通大學圖書館目前的分類標準，中文書是以「中國圖書分類法」，西文書則是採用「美國國會圖書分類法」，我們目前先著重於中文書探勘，於是應用中國圖書分類法[28]為本系統廣義相關規則探勘的項目分類階層。

「中國圖書分類法」[28]將人類知識分為十大類，以十個阿拉伯數字代表，其中 0 為總類，1 為哲學類，2 為宗教類，3 為科學類，4 為應用科學類，5 為社會科學類，6-7 為史地類，8 為語文類及 9 為藝術類。每一大類又分為十小類，共得 100 小類，小類之下以目細分之。

舉例說明本系統應用的廣義相關規則探勘的資料轉換。若有筆交易為{3423, 34636, 9689}，其中“3423”的分類號為 312.91695，“34636”

的分類號為 312.932**，“** 9689 **”的分類號為** 550.91**，且分類階層科學類**(3XX)**到小數點後三位，社會科學類**(5XX)**到小數點後一位，交易則成為**{3XX, 31X, 312, 312.9, 312.91, 312.916, 3423, 312.93, 312.932, 34636, 5XX, 55X, 550.9, 9689}**。**

- **資料探勘**

本系統提供三種資料探勘演算法，包括相關規則演算法，廣義相關規則演算法及多重最小支持度廣義相關規則演算法。

- **結果呈現與評估**

經由本系統三種演算法探勘直接得到結果頻繁項目集。可再經由系統將結果頻繁項目集轉換成封閉式頻繁項目集(Closed Frequent Itemsets)**或是再設定確信值**(Confidence) **得到相關規則。最後，我們將封閉式頻繁項目集導入個人化數位圖書資訊環境** PIE@NCTU**，在讀者搜尋館藏及瀏覽借閱歷史記錄時，提供借閱的建議。**

4.1.2 系統建置需求

- **硬體設備：所需要的硬體設備為一部個人電腦，建議基本配備如下：**
  - **中央處理器**(CPU)**：**AMD Athlon 700 **或** Intel Pentium III 700
  - **記憶體：**128MB
  - **硬碟：**30GB
- **軟體設備**
  - **作業系統：**Microsoft Windows 2000 Professional、Microsoft Windows 2000 Server **或** Microsoft Windows XP Professional
  - **資料庫：**Microsoft SQL Server 7.0

4.1.3 系統功能

圖書館資料探勘系統依功能分為七大類，分別為檔案(File)、資料庫(Database)、轉換(Transform)、探勘(Mining)、產生規則(GenRules)、系統(System)、離開(Exit)。系統的起始畫面如圖 4 - 1 - 2：

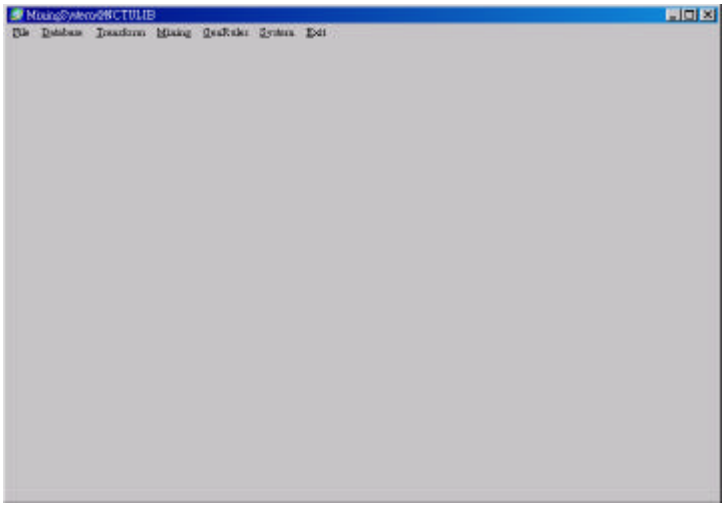


圖 4 - 1 - 2：系統起始畫面

- 檔案(File) ：檔案的功能如圖 4 - 1 - 3，包括顯示內容(Show)及另存新檔(Save to File)。顯示內容為呈現系統內所有會用到的各種檔案格式內容，檔案的格式包括借閱記錄檔、資料轉換過程資訊檔(.infor)、探勘過程資訊檔(.datainfor)、探勘結果檔案(.result) 及封閉式結果檔案(.closed)。另存新檔則是讓館員將資料探勘的結果儲存起來。

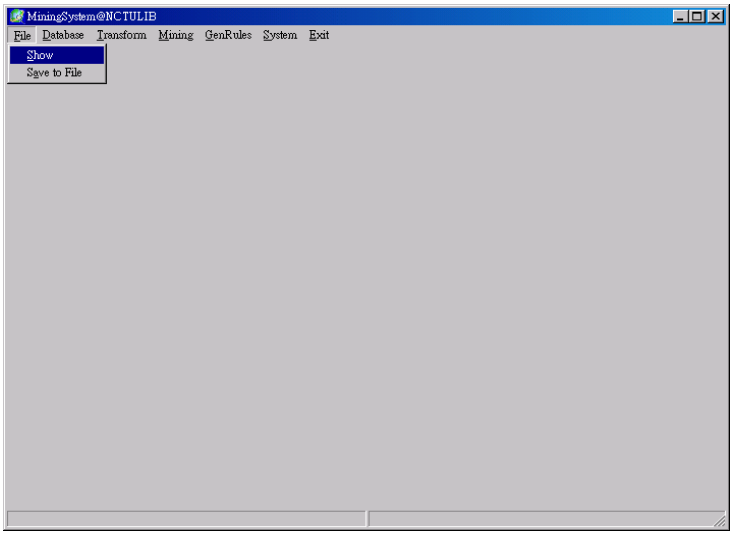


圖 4 - 1 - 3：檔案功能

■ **資料庫**(Database) **：資料庫的功能包括插入資料、刪除資料及清理資料庫。**

◆ **插入資料**(Insert Data) **：本研究將借閱記錄資料儲存至所建置的借閱記錄資料庫** Mining_Lib **中**(**見** 39 **頁**)**。插入資料是要讓使用者可以隨時更新借閱記錄資料庫，如圖** 4 - 1 - 4**。將如表** 4 - 1 - 1 **格式的借閱記錄存入借閱記錄資料庫中，以便於之後的轉換資料及規則探勘。**

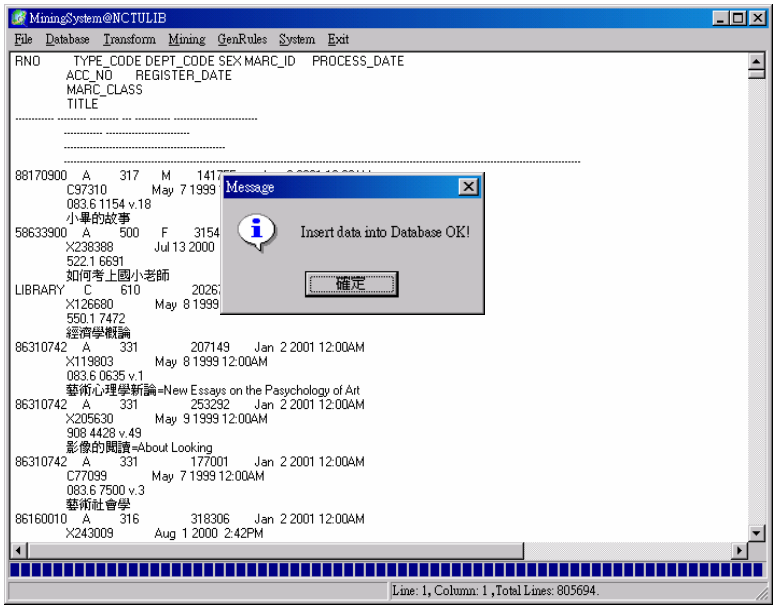


圖 4 - 1 - 4**：插入資料**

◆ **刪除資料**(Delete Data) **：選定欲刪除的借閱資料年份，刪除該年份的資料，如圖** 4 - 1 - 4**。**

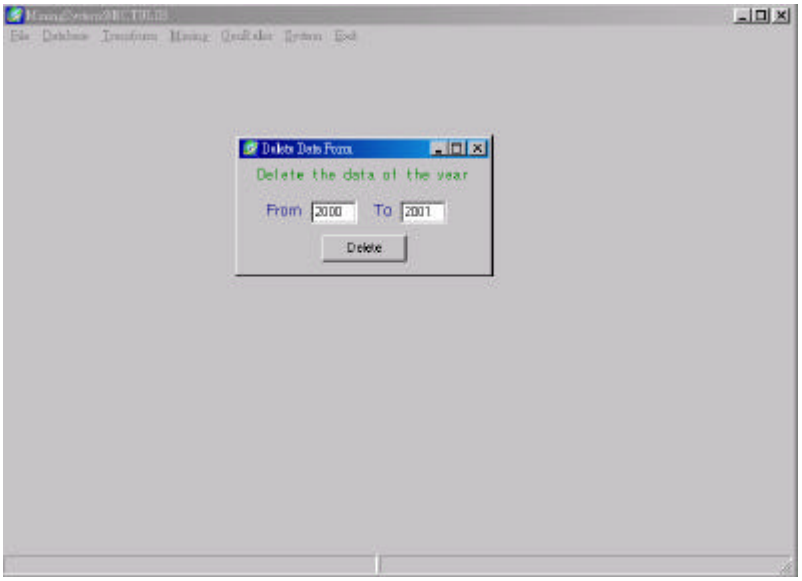


圖 4 - 1 - 5**：刪除資料**

◆ **清理資料庫**(Clear Database) **：如　　圖** 4 - 1 - 6**，按下** OK **按鈕，將整個交易資料庫的借閱記錄資料全部刪除。**

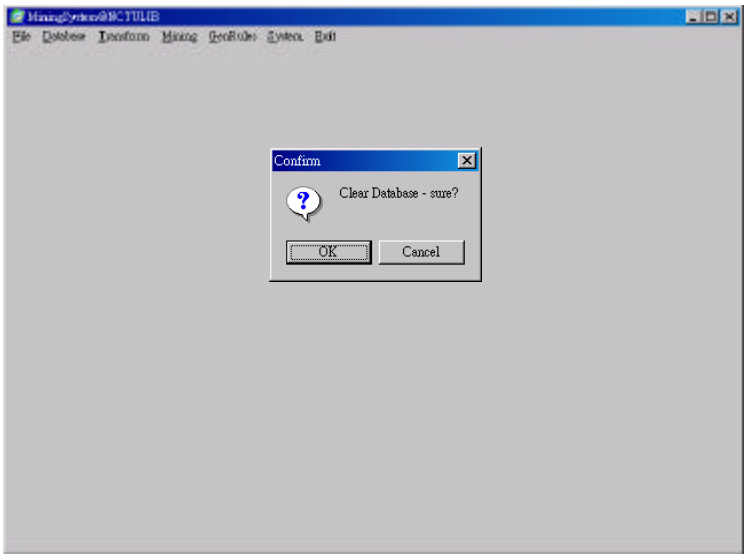


圖 4 - 1 - 6**：清理資料庫**

■ **轉換**(Transform) **：轉換的功能包括轉換資料庫及特殊轉換。**

◆ **轉換資料庫**(Data Transform Database Mining_Lib) **：將整個借閱記錄資料庫** Mining_Lib **的全部資料轉換成符合探勘格式的資料檔。如圖** 4 - 1 - 7**，每一行代表一筆交易，預設借閱間隔於一年內的視為同一筆交易，若間隔超過一年，則分為二筆交易。**

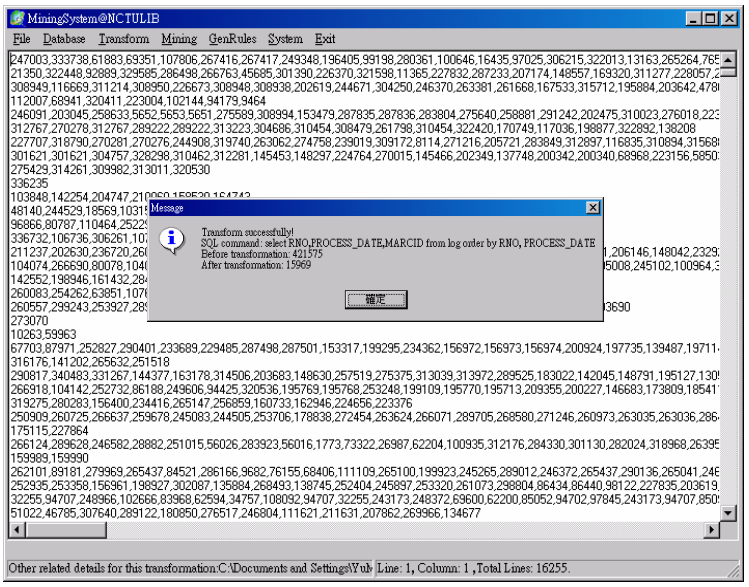


圖 4 - 1 - 7**：轉換資料庫**

◆ **特殊轉換**(Special Transform Form) **：如圖** 4 - 1 - 8**，讓使用者自行輸入欲轉換的起始、終止日期，及多久時間內的資料視為同一筆的資訊，再針對使用者要求轉換。**

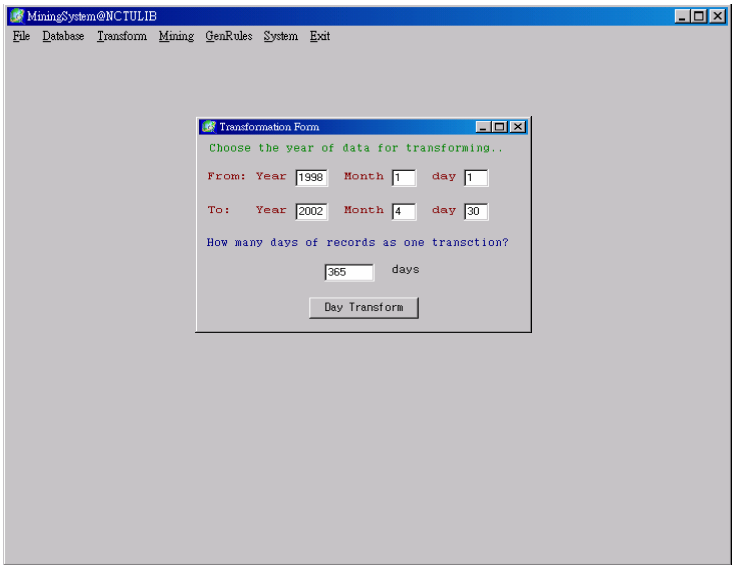


圖 4 - 1 - 8**：特殊轉換**

■ **探勘**(Mining) **：探勘的功能包括相關規則探勘、身份類別探勘、廣義相關規則探勘及多重最小支持度廣義相關規則探勘。**

◆ **相關規則探勘**(Association Mining) **：如圖** 4 - 1 - 9**，讓使用者填入最小支持度及選擇欲探勘檔案，按下** Mining **按鈕，產生頻繁項目集。**

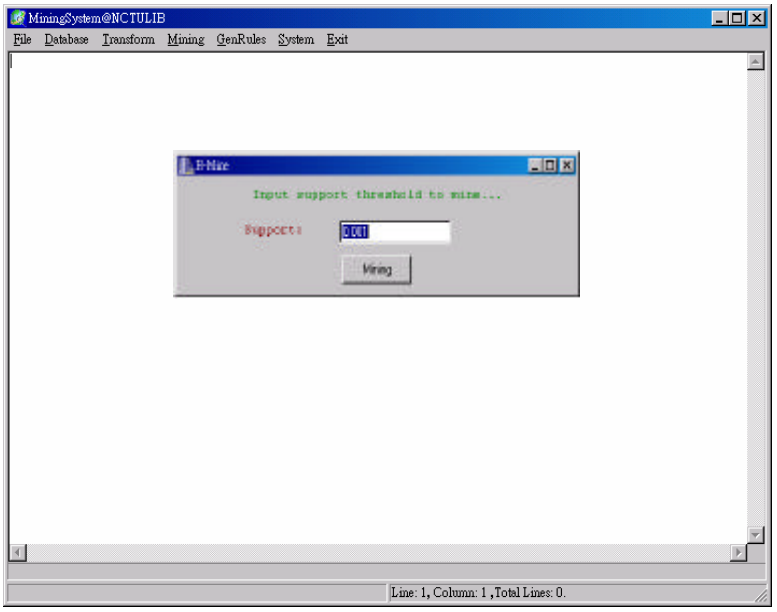


圖 4 - 1 - 9**：相關規則探勘**

輸入支持度 0.001，選擇已由借閱記錄轉換成探勘格式的 2001 年交易記錄，相關規則探勘結果畫面如圖 4 - 1 - 10 (相關規則探勘分析如附錄一)。相關規則結果以頻繁項目集的方式表示，如探勘結果頻繁項目集“211326,210033: 14.”，表示項目 211326 與項目 210033 在 2001 年的借閱資料中同時出現 14 次。

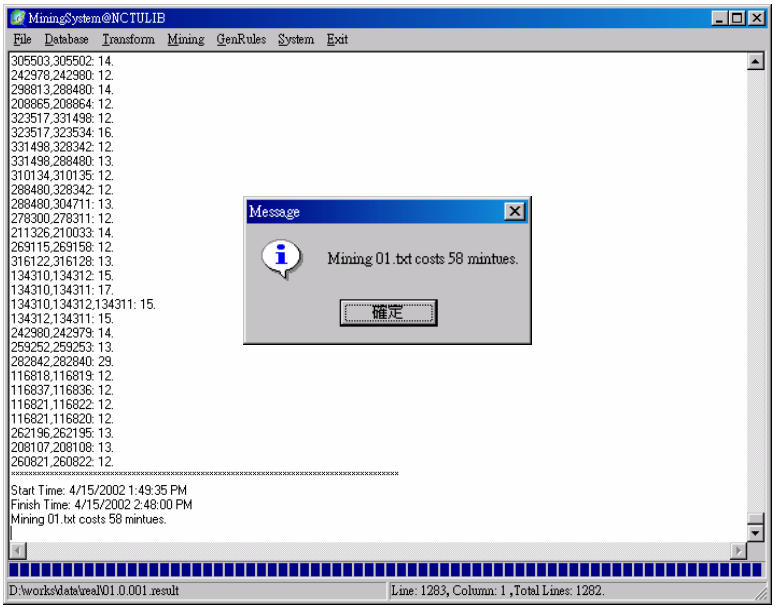


圖 4 - 1 - 10：相關規則探勘結果

◆ 身份類別探勘(User Category Mining) ：先選擇身份類別配置檔(Configuration File)，讓系統明瞭隸屬於各學院中系所的分配。身份類別配置檔的格式是先以一行表示學院，該行以下的每一行以 1 開始編號隸屬於各學院的系所。學院的表示法是二個欄位，以 Tab 為欄位間隔，第一個欄位為編號，學院編號固定為 0，第二個欄位則是學院名稱。而系所則以三個欄位表示，以 Tab 為欄位間隔，第一個欄位為編號，每個學院中的系所編號以 1 為始，第二個欄位是系所名稱，第三個欄位是系所代碼。例如：若電機資訊學院中有資工系、電信系及資科系，系所代碼為 311、313 及 323，工學院有土木系及環工所，系所代碼為 316 及 219，則身份類別配置檔如下：

0 電機資訊學院
1 資工系 311
2 電信系 313
3 資科系 323
0 工學院
1 土木系 316
2 環工所 219

選擇身份類別配置檔所在位置，如圖 4 - 1 - 11 所示，讓系統讀取結構配置檔，建立學院系所類別的可複選選項，如圖 4 - 1 - 12。

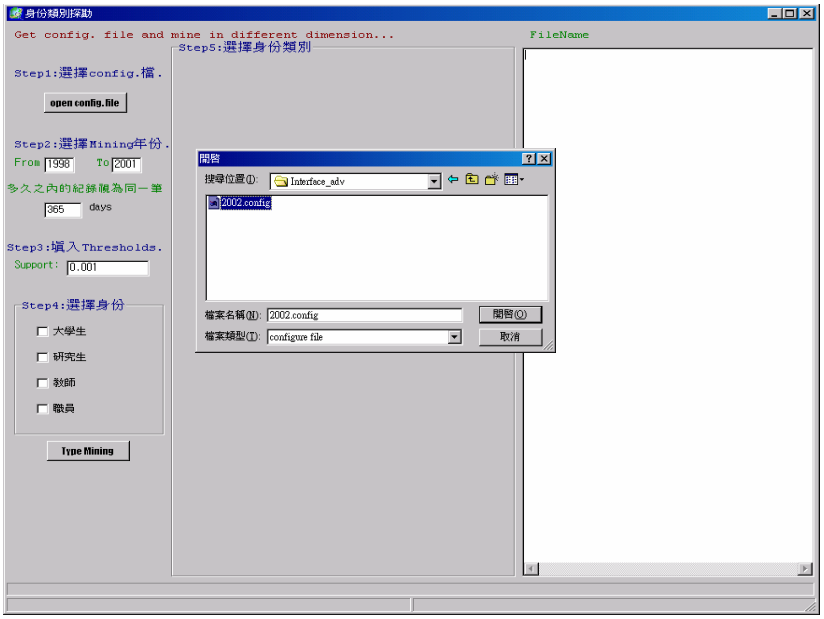


圖 4 - 1 - 11：選擇身份類別配置檔

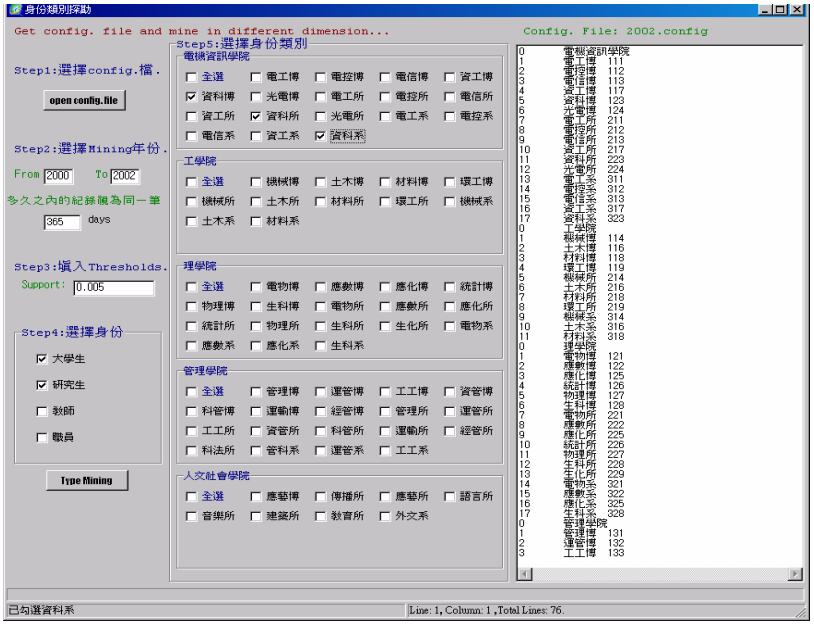


圖 4 - 1 - 12：選擇身份探勘資訊

選擇探勘哪段時間內的借閱資料及多久時間內的資料視為同一筆的資訊，並選擇欲探勘的身份類別及系所類別。如圖 4 - 1 - 12，選擇 2000 至 2002 年的借閱記錄，一年內視為同筆交易，以 0.005 為最小支持度，針對資科系、資科所及資科博的大學生及研究生進行探勘。

選擇探勘資料、身份及最小支持度等相關資訊後，按下 Type Mining 的按鈕後，系統即會自動從資料庫中找到相關資料，並且經由相關規則探勘找出頻繁項目集，顯示在右邊白底方框裡。如圖 4 - 1 - 13，顯示資科學生在 2000 年至 2002 年的借閱記錄探勘結果頻繁項目集。身分類別相關規則探勘分析如附錄二。

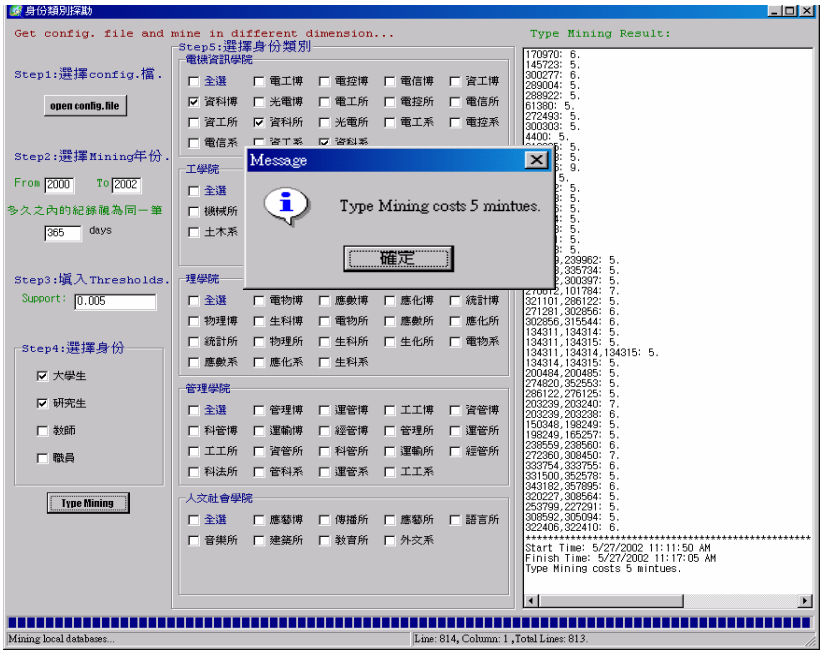


圖 4 - 1 - 13：身份探勘結果

◆ 廣義相關規則探勘(Generalized Mining) ：選擇探勘哪段時間內的借閱資料及多久期間內視為同一筆的相關資訊，再針對中國圖書分類法中的每一類設定探勘深度，按下 Generalized Mining 的按鈕，即可探勘出廣義相關規則。例如探勘 2001 年 11 月 1 日到 2002 年 4 月 30 日的借閱資料，科學類及應用科學類探勘到小數點後第三位，其他類則探勘至小數點後二位，最小支持度設定為 0.0125，則探勘畫面如圖 4 - 1 - 14：

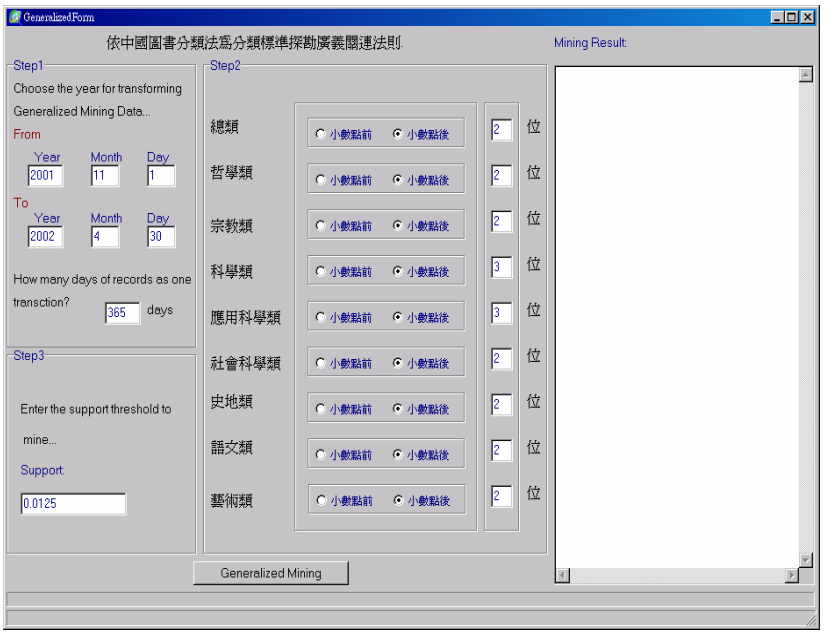
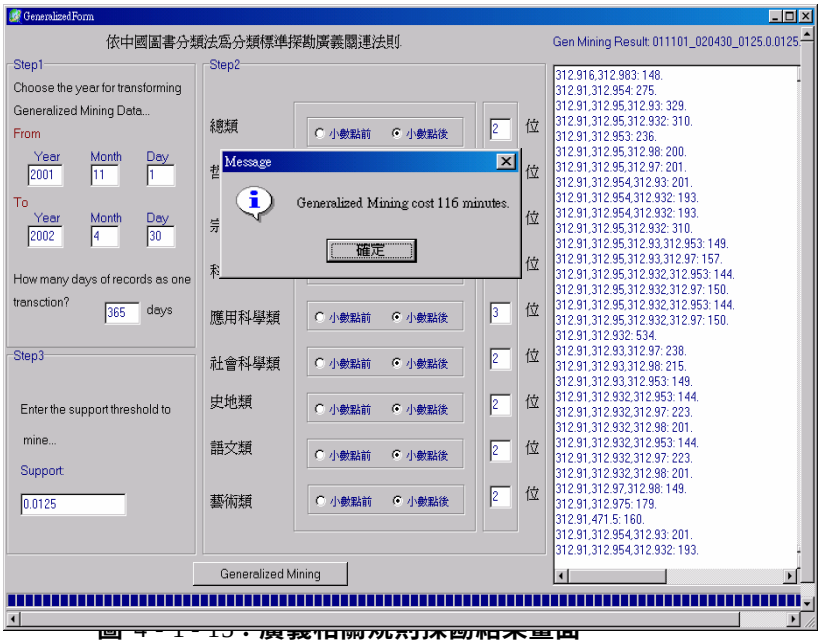


圖 4 - 1 - 14：廣義相關規則探勘

按下 Generalized Mining **的按鈕，即可探勘出廣義相關規則之頻繁項目集。探勘廣義相關規則結果畫面，如圖** 4 - 1 - 15**，探勘結果以頻繁項目集的方式表示，如頻繁項目集“** 312.91,312.95,312.932,312.97: 150**”，表示分類項目** 312.91、312.95、312.932 **及** 312.97 **在** 2001 **年** 11 **月** 1 **日至** 2002 **年** 4 **月** 30 **日的類別交易記錄中同時出現** 150 **次。廣義相關規則探勘分析如附錄三。**



◆ **多重最小支持度廣義相關規則探勘**(Mulitple Supports Mining) **：選擇探**

勘哪段時間內的借閱資料及多久期間內視為同一筆的相關資訊，再針對中國圖書分類法中的每一類設定探勘深度及每一層設定最小支持度值，按下 Multiple Supports Mining 的按鈕，即可探勘出多重最小支持度廣義相關規則。例如探勘 2001 年 11 月 1 日到 2002 年 4 月 30 日的借閱資料，科學類及應用科學類探勘到小數點後第三位，其他則探勘至小數點後二位，最小支持度各階層設定分別為小數點前三位最小支持度設定為 0.25，小數點前二位設定為 0.2，小數點前一位設定為 0.15，小數點後一位最小支持度設定為 0.1，小數點後二位最小支持度設定為 0.05，小數點後三位最小支持度設定為 0.01，一般項目最小支持度設定為 0.00125，則探勘畫面如圖 4 - 1 - 16：

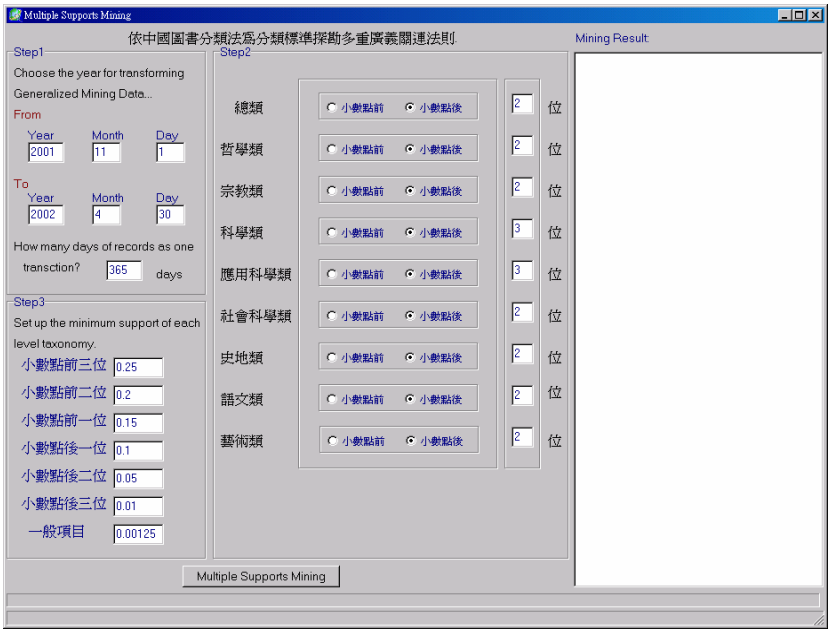


圖 4 - 1 - 16：多重最小支持度相關規則探勘

按下 Multiple Support Mining 的按鈕，即可探勘出多重最小支持度廣義相關規則。探勘結果畫面如 圖 4 - 1 - 17，探勘結果包括一般項目頻繁項目集與廣義分類項目頻繁項目集。多重最小支持度廣義相關規則探勘分析如附錄四。

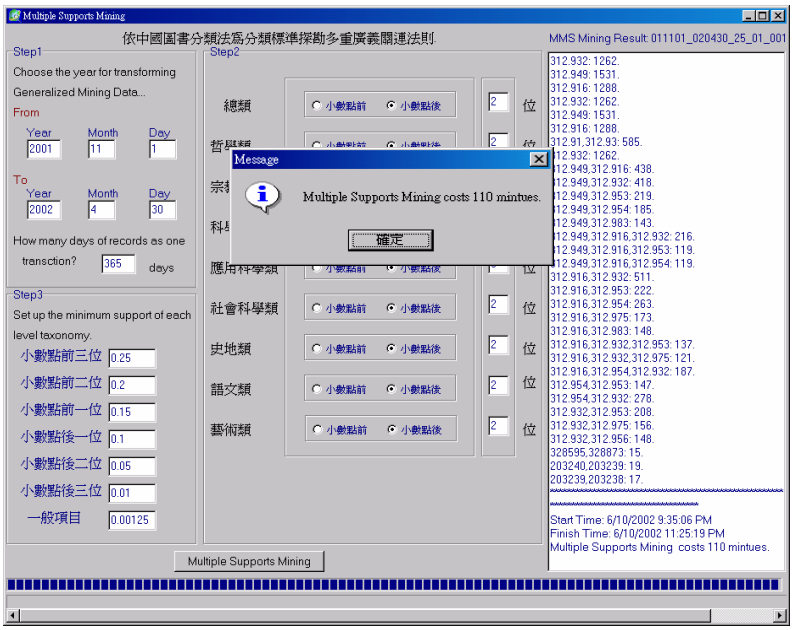


圖 4 - 1 - 17：多重最小支持度相關規則探勘結果

- **產生規則**(GenRules) **：產生規則的功能包括產生相關規則及產生封閉式頻繁項目集。**
  - **產生相關規則**(Generate Rules) **：讓使用者輸入確信值及選擇之前探勘所得到的結果頻繁項目集檔案，系統即可產生相關規則，如圖** 4 - 1 - 18 **所示，相關規則“** 282842 => 282840 conf:0.88 **”，表示借館藏** 282842 **的讀者其中有** 88%**會同時也借館藏** 28842**。**

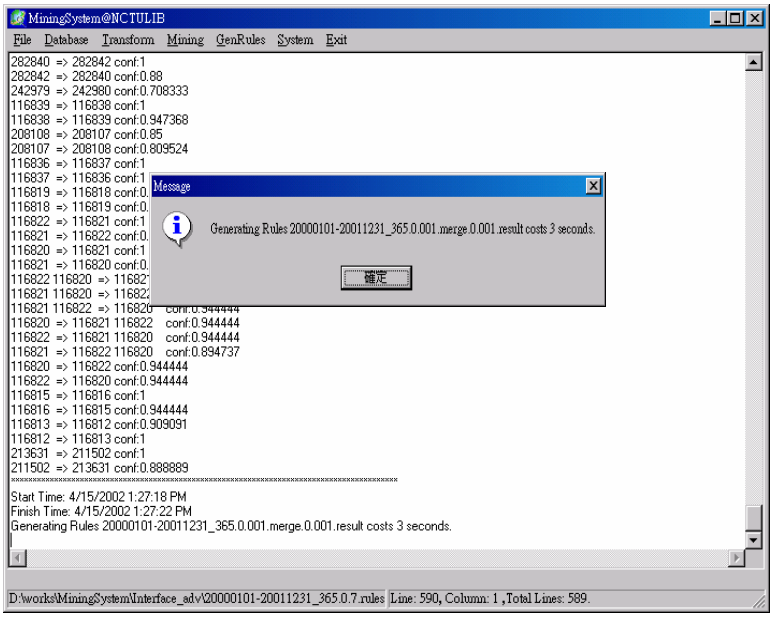


圖 4 - 1 - 18：相關規則結果

◆ **產生封閉式頻繁項目集**(Generate Closed Itemsets) **：利用此功能，使用者只要選擇之前探勘所得到的結果頻繁項目集檔案，即可得到封閉式頻繁項目集，如圖** 4 - 1 - 19**。封閉式頻繁項目集的優點是在不失去任何跟頻繁項目集有關資訊的情況，仍可以最精簡的方式表達所有的頻繁項目集。本論文即是利用封閉式頻繁項目集將結果與個人化數位圖書資訊環境** PIE@NCTU **結合，針對讀者特性提供借閱建議。**

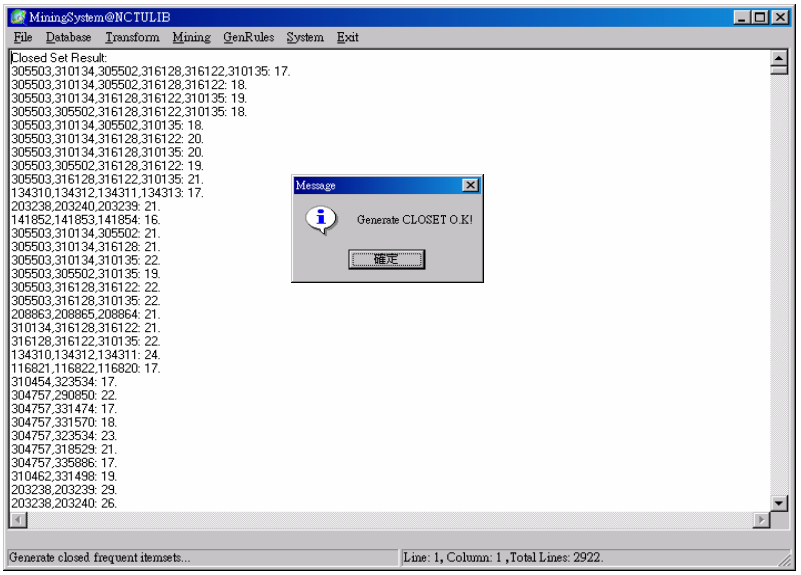


圖 4 - 1 - 19**：封閉式頻繁項目集結果**

■ **系統**(System) **：系統的功能包括探勘資料資訊及系統記憶體使用量。**

◆ **探勘資料資訊**(Data Information) **：輸入最小支持度及探勘格式資料，即可預估探勘此資料需要多少系統資源，結果如圖** 4 - 1 - 20**。**

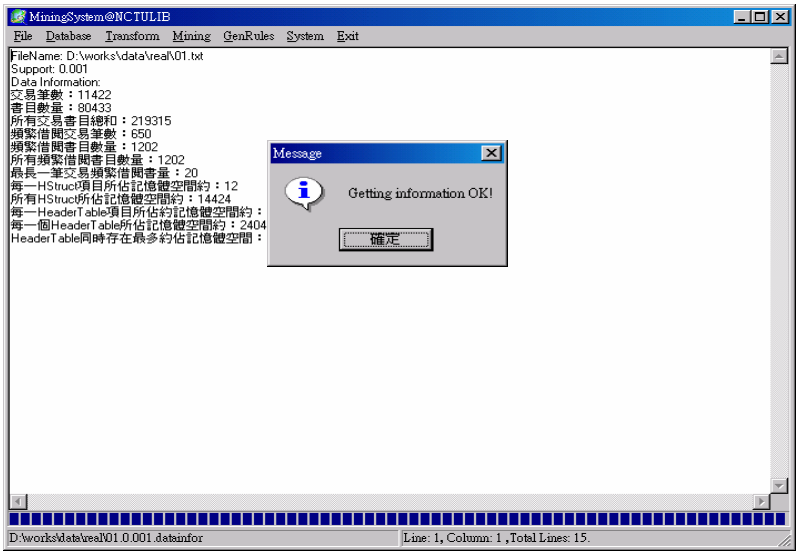


圖 4 - 1 - 20**：探勘資料資訊**

◆ **系統記憶體使用量**(System Memory Usage) **：顯示目前系統記憶體資源使用量，如圖** 4 - 1 - 21 **所示。**

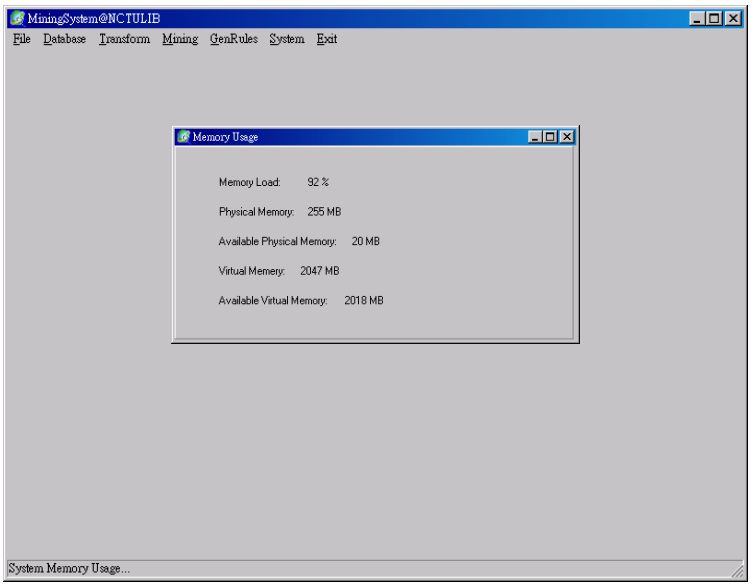


圖 4 - 1 - 21**：系統記憶體使用量**

■ **離開**(Exit) **：如圖** 4 - 1 - 22**，選擇** Exit**，離開本系統。**

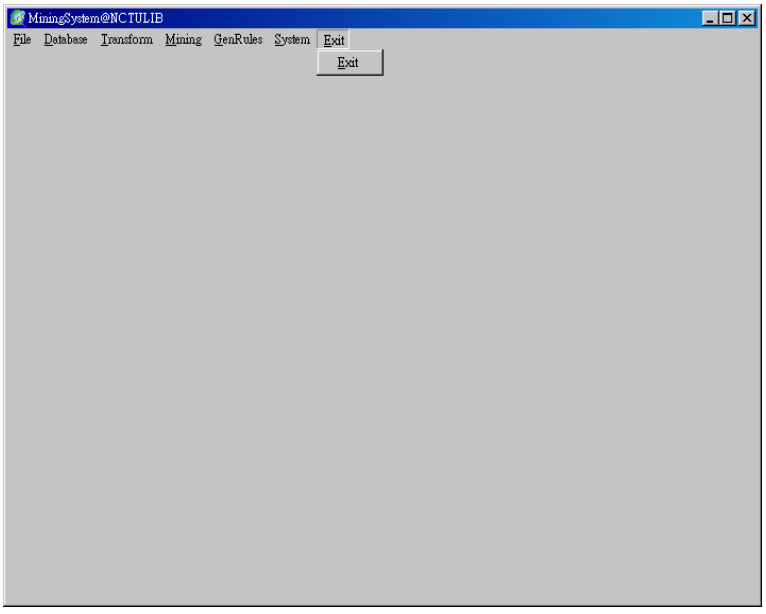


圖 4 - 1 - 22**：離開系統**

第二節 應用於個人化數位圖書資訊環境

我們將探勘結果導入個人化數位圖書資訊環境 PIE@NCTU **中，當讀者瀏覽其借閱記錄時，即可得知是否有相關的借閱館藏。**

如圖 4 - 2 - 23 為 PIE@NCTU 的個人登入畫面，讀者可輸入在交通大學圖書館的帳號及密碼。若無帳號進入，只是參觀系統，在帳號密碼欄位均輸入 guest 即可登入。

圖 4 - 2 - 23：個人化數位圖書資訊環境 PIE@NCTU 登入畫面

圖 4 - 2 - 24：PIE@NCTU 智慧型查詢畫面

在圖 4 - 2 - 23 的左邊選單中選擇智慧型查詢選項，輸入欲查詢館藏的資訊，如圖 4 - 2 - 24 所示，利用類似一般圖書館檢索系統查詢館藏的方法，即可得知所需館藏的相關資訊。不同的是，PIE@NCTU 會根據使用者的興趣將檢索

結果排序。檢索結果會根據系統計算出來的興趣依不同的等級排列，等級越高，與使用者感興趣的館藏就越可能相關。再由檢索所得之結果畫面，選擇「推薦」即可得知是否有相關連館藏。例如：檢索關鍵字「廚房」，得到結果畫面如圖 4 - 2 - 25。PIE@NCTU 將檢索結果分為三個等級，選擇與興趣最接近的結果「溫馨廚房咖啡座」，按下右邊推薦的地球圖示，如圖 4 - 2 - 26，則會顯示關聯館藏「海德堡之吻」、「香水婚紀念日」及「花內褲排排掛：鄭華娟的歐式家庭生活手記」。

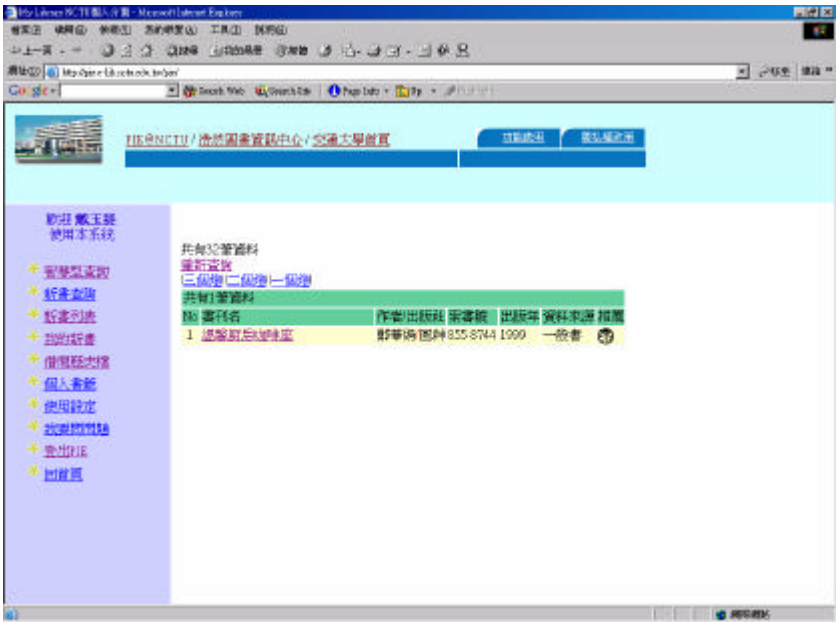


圖 4 - 2 - 25：PIE@NCTU 智慧型查詢結果畫面

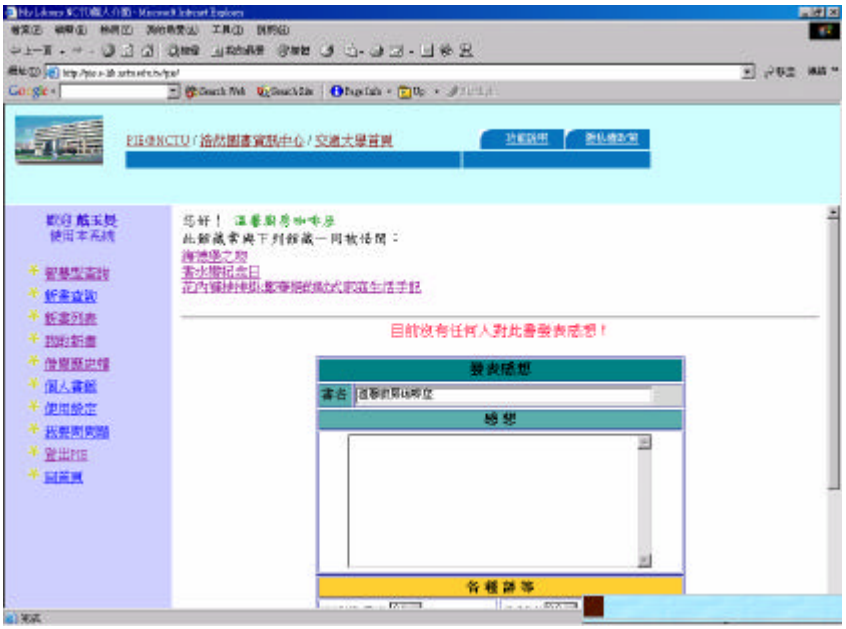


圖 4 - 2 - 26：PIE@NCTU 智慧型查詢之推薦關聯館藏畫面

在圖 4 - 2 - 23 的左邊選單中選擇借閱歷史檔選項，既可得知該帳號曾經借

閱過的所有館藏，亦可選擇查詢時間，查詢一段時間內曾經借閱過的館藏。借閱歷史檔畫面如圖 4 - 2 - 27：

圖 4 - 2 - 27：PIE@NCTU 借閱歷史檔畫面

在借閱歷史檔畫面的記錄中，按下共享按鈕，不但可以得知推薦的相關館藏，還可閱讀其他讀者針對該館藏的心得，讀者也可以針對該筆記錄發表心得。但是，若無相關館藏，則不會顯示在畫面中，只會有其他讀者的心得及發表心得的畫面。例如圖 4 - 2 - 27 中，選擇曾借閱過的館藏「臥虎藏龍」，則會推薦「駭客任務」及「永不妥協」給讀者，推薦館藏畫面如圖 4 - 2 - 28：

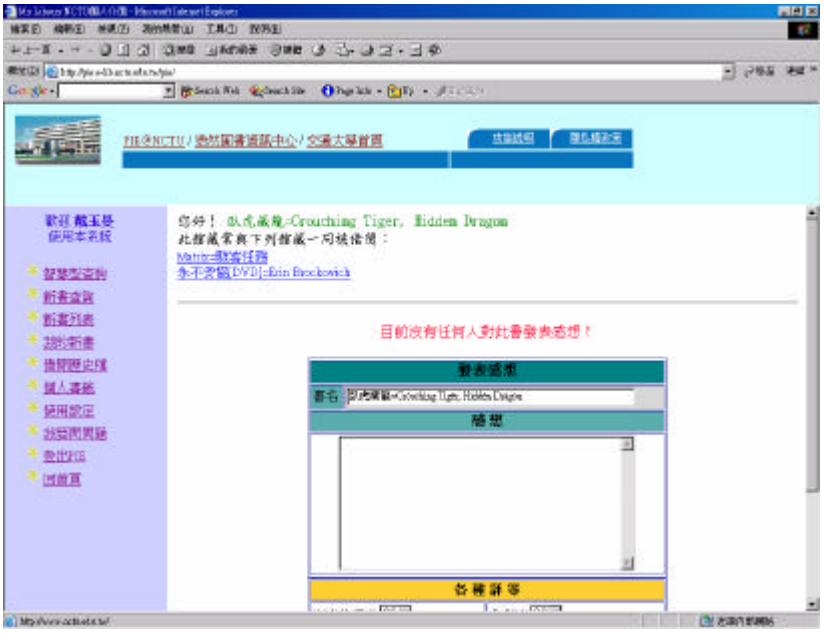


圖 4 - 2 - 28：PIE@NCTU 借閱歷史檔之推薦相關館藏畫面

第五章 圖書館借閱記錄探勘系統評估

本章將針對本論文所應用在圖書館借閱記錄探勘系統的演算法 H-Mine、H-Mine(Generalized) 及 H-Mine(MMS) 進行效益分析。第一節介紹系統評估實驗的環境，包括軟硬體設備及探勘資料；第二節將說明相關規則、廣義相關規則及多重最小支持度廣義相關規則探勘效益評估；第三節則是 H-Mine(Generalized) 與 H-Mine(MMS) 的效益分析；最後一節則是系統效益評估總結。

第一節 實驗環境

本節中說明「圖書館借閱記錄探勘系統」系統評估的實驗環境，包括：硬體設備、軟體設備及探勘資料。

5.1.1 硬體設備

實驗所使用的硬體設備為一部個人電腦，配備如下：

- **中央處理器**(CPU)：AMD Athlon XP 1600
- **記憶體**：512MB
- **硬碟**：75GB

5.1.2 軟體設備

- **作業系統**：Microsoft Windows 2000 Server
- **資料庫**：Microsoft SQL Server 7.0

5.1.3 探勘資料

本系統評估以交通大學圖書館 2001 年 5 月 1 日至 2002 年 4 月 30 日的借閱交易記錄為基礎，針對相關規則探勘、廣義相關規則探勘及多重最小支持度廣義相關規則探勘演算法作系統效益評估。總計在此期間的記錄包括使用者 15,409

位，借閱交易記錄共有 341,050 筆。

第二節 探勘效益評估

將借閱資料分為一個月、三個月、半年、九個月及一年的資料量，相同讀者借閱間隔一年內的資料視為同一筆交易，設定最小支持度為 0.0005、0.001 及 0.005，進行相關規則探勘。相關規則探勘資料詳細資訊如表 5 - 2 - 1。

時間間隔	借閱資料時間	借閱記錄筆數	資料轉換資料量(KB)	交易筆數
一個月	2002. 4. 1 ~ 2002. 4. 30	29213	203	5672
三個月	2002. 2. 1 ~ 2002. 4. 30	85449	584	8909
半年	2001. 11. 1 ~ 2002. 4. 30	179309	1219	11494
九個月	2001. 8. 1 ~ 2002. 4. 30	262759	1780	12869
一年	2001. 5. 1 ~ 2002. 4. 30	341050	2309	15409

表 5 - 2 - 1：相關規則借閱資料詳細資訊

針對系統效率分析，如圖 5 - 2 - 1，探勘時間會隨著借閱資料量增加，時間遞增的趨勢呈線性關係。當借閱資料量小時，如一個月或是三個月的資料量，支持度設定值的大小並不會影響探勘時間。因為資料量小，探勘時間快，即使支持度設小，相對地探勘時間也不至於增加太多。但是當資料量大時，如半年以上的資料量，則會因為支持度變小，而增加探勘時間。且由表 5 - 2 - 2 得知，最小支持度越小時，探勘所得的頻繁項目集越多，故探勘所需花費的時間亦隨之增加。

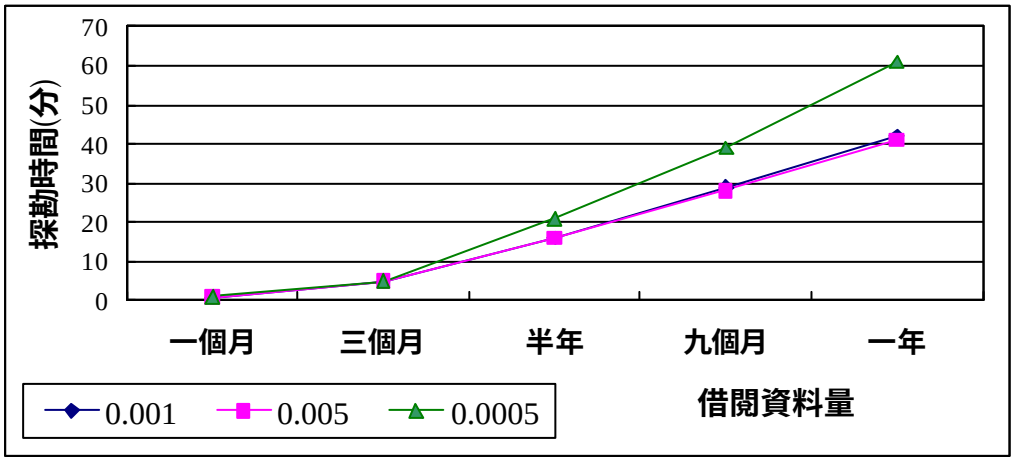


圖 5 - 2 - 1：H-Mine 資料量與探勘時間分析

最小支持度 時間間隔	0.0005	0.001	0.005
一個月	20	0	0
三個月	48	0	0
半年	352	7	0
九個月	533	25	0
一年	913	48	0

表 5 - 2 - 2：相關規則頻繁項目集個數

進一步分析 H-Mine 探勘的結果，我們以資料時間間隔為一年的資料及最小支持度為 0.0005 的交易分析探勘結果。在這一年的借閱資料中，有借出記錄的館藏有 94468 本，而符合最小支持度為 0.0005 的單一頻繁項目集卻只有 7649 個。對 H-Mine 演算法而言，單一頻繁項目集只佔了所有借閱項目的 8.10%。且如表 5 - 2 - 3 所示，當項目集長度越長時，滿足支持度的項目集個數越少，項目集支持度平均也越小。我們以單一頻繁項目集項目預估各長度組合個數，如預估長度為 2 的頻繁項目集個數為 $C_2^{7649} = \frac{7649(7649-1)}{2\times 1} = 29249776$。其中各長度頻繁項目集個數佔其預估項目集個數的比率隨著長度增加而減少。亦得知，各長度頻繁項目集皆只佔整個記錄項目的一少部分，故印證前述的圖書館借閱館藏重疊性低，必須在支持度低時才找出好結果。

項目集長度	項目集個數	支持度平均	預估項目集個數	實際佔預估百分比
7	1	8	3.03×10^{23}	3.30×10^{-20}%
6	8	8.5	2.77×10^{20}	2.89×10^{-18}%
5	33	8.85	2.18×10^{17}	1.51×10^{-14}%
4	83	9.14	1.43×10^{14}	5.80×10^{-11}%
3	151	9.45	7.46×10^{11}	2.02×10^{-8}%
2	637	9.86	2.93×10^{7}	2.17×10^{-3}%

表 5 - 2 - 3：以時間間隔為一年的資料，最小支持度為 0.0005 探勘結果分析

由於 H-Mine 演算法最為人所質疑之處在於探勘時需要耗費記憶體儲存頻繁項目集，因此本研究針對記憶體耗費狀況作分析。如圖 5 - 2 - 2 所示，最小支持度越小時，因為頻繁項目集增多，所需耗費的記憶體亦隨之遞增。且由表 5 - 2 - 2 得知，最小支持度越大，頻繁項目集越少，所需耗費的記憶體量亦隨之減少。

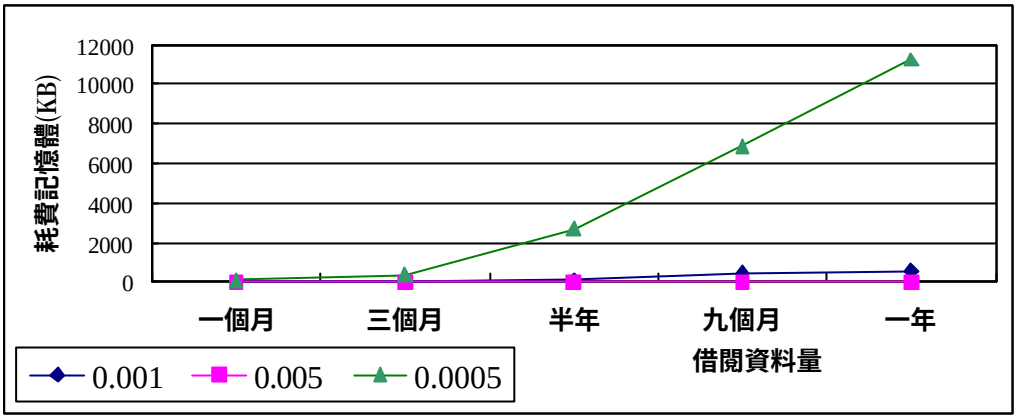


圖 5 - 2 - 2：H-Mine 資料量與記憶體耗費量分析

廣義相關規則探勘與多重最小支持度廣義相關規則探勘資料量如同相關規則探勘資料量設定，分為一個月、三個月、半年、九個月及一年的資料量，相同讀者借閱間隔一年內的資料視為同一筆，階層分類設定科學類及應用科學類設定為小數點後三位，其他類別則設定為小數點後二位。廣義相關規則探勘最小支持度設定分別為 0.0125、0.025 及 0.05。多重最小支持度廣義相關規則探勘各階層最小項目支持度設定分別為 0.25, 0.2, 0.15, 0.1, 0.05, 0.025, 0.0125 及 0.5, 0.4, 0.3, 0.2, 0.1, 0.05, 0.025。廣義相關規則探勘資料詳細資訊如表 5 - 2 - 4。

時間間隔	借閱資料時間	借閱記錄筆數	資料轉換資料量(KB)	交易筆數
一個月	2002. 4. 1 ~ 2002. 4. 30	29213	647	5672
三個月	2002. 2. 1 ~ 2002. 4. 30	85449	1876	8909
半年	2001. 11. 1 ~ 2002. 4. 30	179309	3941	11494
九個月	2001. 8. 1 ~ 2002. 4. 30	262759	5717	12869
一年	2001. 5. 1 ~ 2002. 4. 30	341050	7422	15409

表 5 - 2 - 4：廣義相關規則借閱資料詳細資訊

針對系統效率分析 H-Mine(Generalized) 與 H-Mine(MMS)。由圖 5 - 2 - 3 及圖 5 - 2 - 4 所示，探勘時間會隨著借閱資料量增加及最小支持度變小而增加。當借閱資料量小時，即使因增加廣義項目探勘而增加項目集數量，支持度設定值的大小仍不會影響探勘時間。但是，半年以上的資料量，則會因為支持度變小，符合最小支持度的頻繁項目集量變多，且又加入廣義項目去探勘，而大幅增加探勘時間。即最小支持度設定的影響亦會隨著資料量遞增，所需增加的探勘時間遞增。

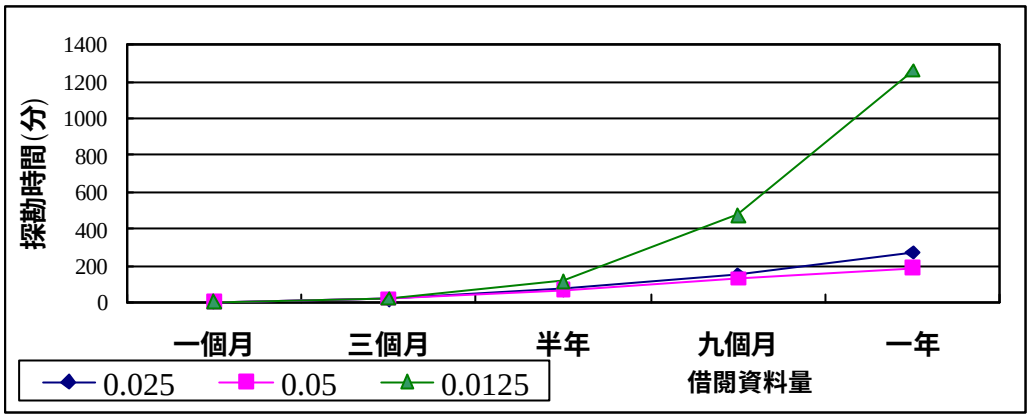


圖 5 - 2 - 3：H-Mine(Generalized)資料量與探勘時間分析

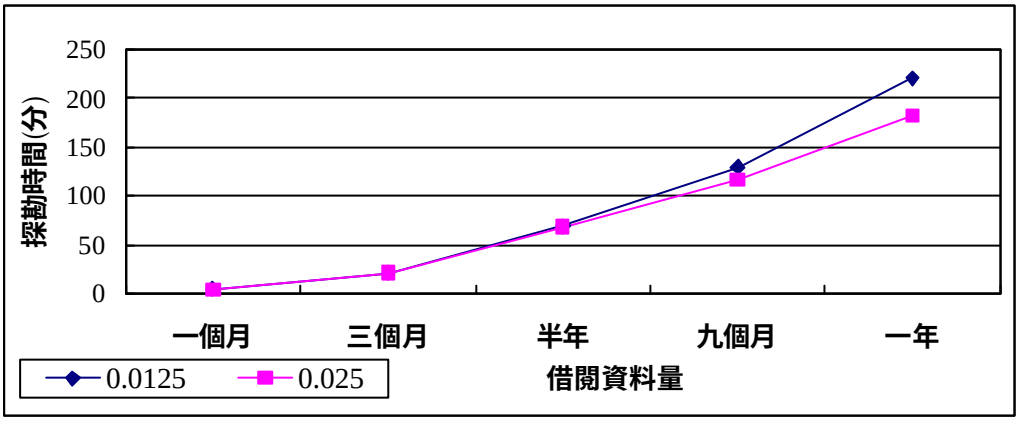


圖 5 - 2 - 4：H-Mine(MMS)資料量與探勘時間分析

H-Mine(Generalized) 及 H-Mine(MMS) 的記憶體使用量分析如圖 5 - 2 - 5 與圖 5 - 2 - 6 所示。記憶體消耗量隨著資料量遞增呈線性成長。最小支持度越小時，因為頻繁項目集增多，所需耗費的記憶體亦隨之遞增，且會因為最小支持度設定的影響，需耗費的記憶體的差距會隨著資料量遞增而小幅度增加。

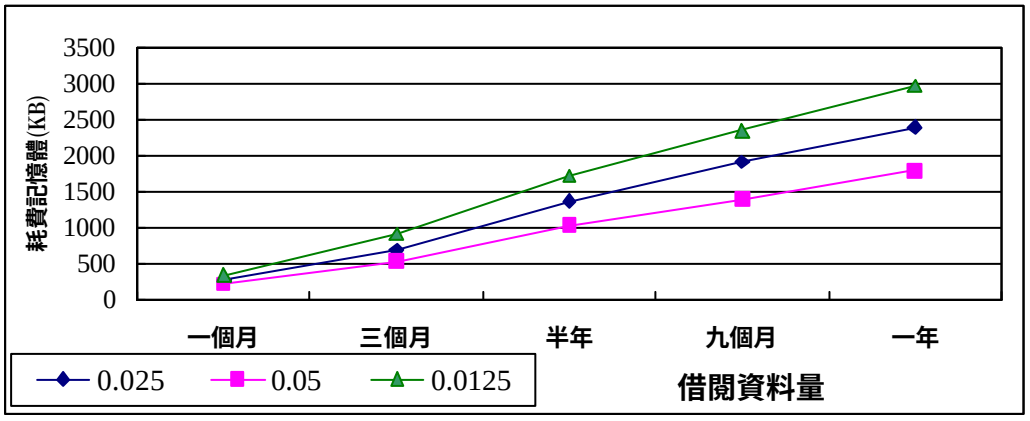


圖 5 - 2 - 5：H-Mine(Generalized)資料量與記憶體耗費量分析

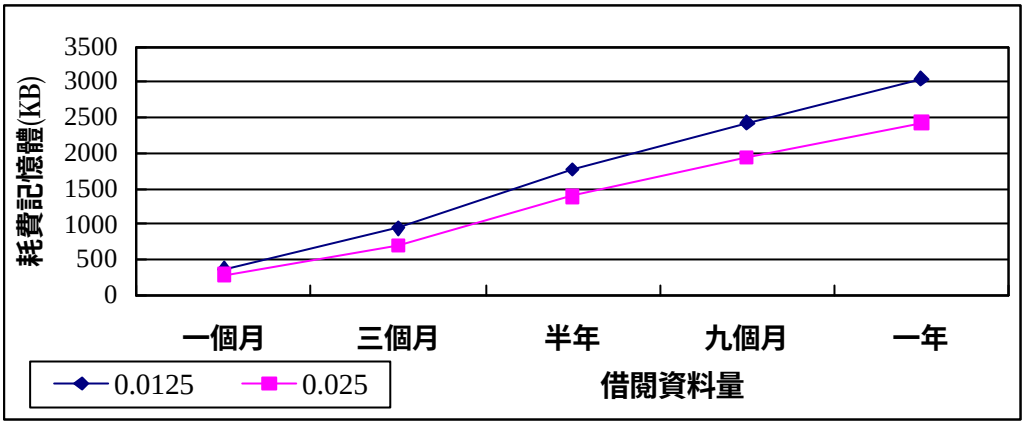


圖 5 - 2 - 6：H-Mine(MMS)資料量與記憶體耗費量分析

第三節 H-Mine(Generalized) 及 H-Mine(MMS) 效益評估

在本節中，我們將針對 H-Mine(Generalized) 及 H-Mine(MMS) 演算法進行探勘時間及耗費記憶體量效益分析。資料量及階層分類設定如同第二節，H-Mine(Generalized) 的最小支持度設定為 0.0125 及 0.025，而 H-Mine(MMS)各階層最小項目支持度設定分別為 0.25 、0.2 、0.15 、0.1 、0.05 、0.025 、0.0125 及 0.5 、0.4 、0.3 、0.2 、0.1 、0.05 、0.025。

針對探勘時間分析，如下圖 5 - 3 - 7 所示，H-Mine(Generalized) 演算法與 H-Mine(MMS) 演算法在資料量小時，支持度設定對探勘時間的影響極小。而隨著資料量的增加，H-Mine(Generalized) 所需要探勘時間遞增的速率較 H-Mine(MMS) 快。因為在同樣的一般項目最小支持度設定值下，多重最小支持度廣義相關演算法探勘廣義項目時會設定較一般項目高的最小項目支持度，這樣一來，符合廣義項目的頻繁項目集會比廣義相關演算法探勘時少，所需的探勘時間就會因廣義項目支持度較高而遞減。因此，以多重最小項目廣義探勘演算法探勘可以較有效率的方式，探勘適量且適宜的廣義相關規則。

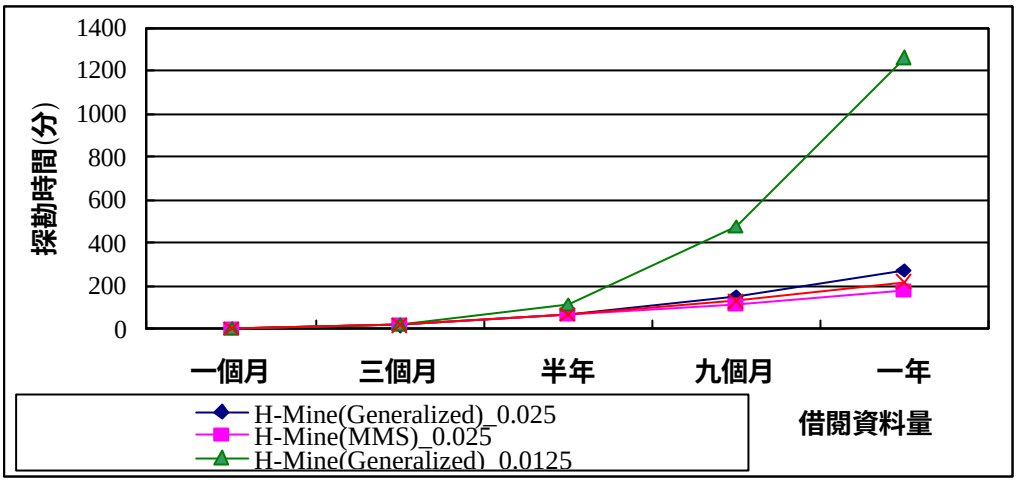


圖 5 - 3 - 7：H-Mine(Generalized) vs. H-Mine(MMS) 探勘時間分析

H-Mine(MMS)與 H-Mine(Generalized)的記憶體使用量，如圖 5 - 3 - 8 所示，皆會隨著資料量增加呈線性成長。在相同最小支持度下，H-Mine(MMS)與 H-Mine(Generalized) 的記憶體耗費量幾乎相同。隨著資料量遞增，H-Mine(MMS) 所耗費的記憶體量卻略多於 H-Mine(Generalized)。這是因為即使 H-Mine(MMS) 的廣義頻繁項目集較 H-Mine(Generalized) 少，在頻繁項目集數量上耗費的記憶體較少，但是由於 H-Mine(MMS) 的每個標題表格中的每個頻繁項目均會多一個儲存最小項目支持度的欄位，因此所需耗費的記憶體總量加總起來，H-Mine(MMS) 所需要的量仍多於 H-Mine(Generalized)。

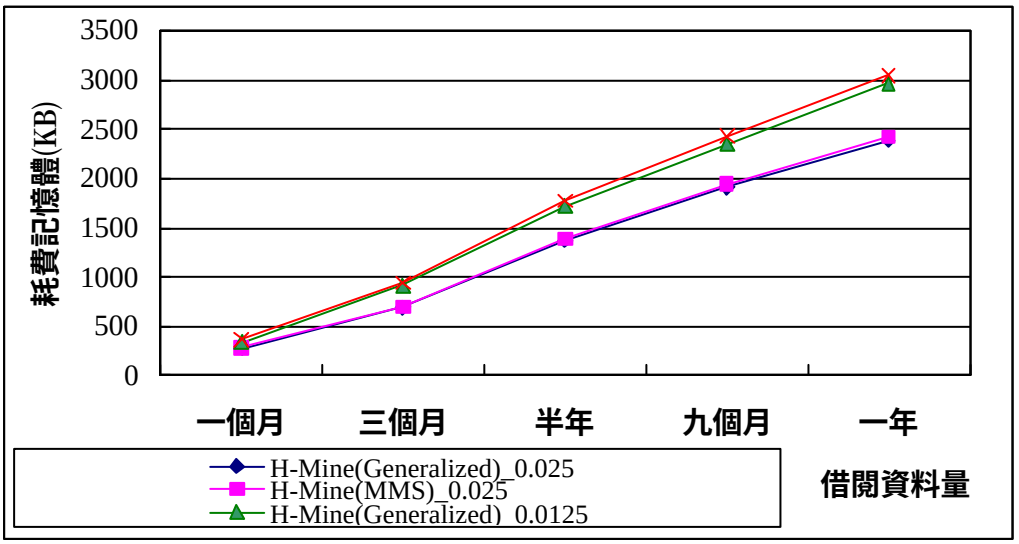


圖 5 - 3 - 8：H-Mine(Generalized) vs. H-Mine(MMS) 記憶體耗費量分析

第四節 系統效益評估總結

本研究依據我們所應用在「圖書館借閱記錄探勘系統」的演算法 H-Mine、H-Mine(Generalized) 及 H-Mine(MMS) 進行效益評估。

針對探勘時間分析，三種演算法探勘時間皆隨著資料量增加而增加，且因支持度的影響，探勘時間增加的速率亦會隨著資料量增加而遞增。但在資料量小時，探勘時間並不會因支持度大小而受影響。

就記憶體耗費量而言，三種演算法的記憶體耗費量會隨著時間增加而遞增。亦會因支持度的影響，耗費記憶體增加的速率亦會隨著資料量增加而小幅度增加。

就 H-Mine(MMS) 與 H-Mine(Generalized) 進行探勘時間分析，在相同一般項目最小支持度設定值下，由於 H-Mine(MMS) 會在廣義項目設定較一般項目高的最小支持度，頻繁項目集會因而減少，故 H-Mine(MMS) 所需要耗費的探勘時間較 H-Mine(Generalized) 少。同理，隨著其中資料量的增加，H-Mine(Generalized) 所需要探勘時間遞增的速率較 H-Mine(MMS) 快。

就 H-Mine(MMS) 與 H-Mine(Generalized) 的記憶體耗費量而言，皆會隨著資料量增加呈線性成長。資料量小時，二者記憶體耗費量幾乎相同，且 H-Mine(MMS) 的每個標題表格中的每個頻繁項目均會多一個儲存最小項目支持度的欄位，隨著資料量遞增，H-Mine(MMS) 所耗費的記憶體量會略多於 H-Mine(Generalized)。

整體而言，探勘時間及記憶體耗費量與資料量的關係均呈線性關係，皆在可容忍範圍內。因此，我們認為本研究所提出的演算法均適用於圖書館借閱資料的資料探勘系統上。

第六章 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。第一節總結本論文運用資料探勘相關規則技術所實作的資料探勘系統「圖書館借閱記錄探勘系統」；第二節說明本論文所研究之主題的未來研究發展方向。

第一節 結論與討論

由於圖書館借閱記錄具有資料量大、借閱重疊性低，且必須在支持度小時才會有比較適宜的結果等特性，根據這些特性，本論文認為在諸多相關規則演算法中以 H-Mine 較適用於圖書館借閱記錄探勘，因此本論文將 H-Mine 應用於圖書館借閱記錄探勘系統中。除此之外，本論文並修改 H-Mine，使其能應用於廣義相關規則探勘及多重最小支持度廣義相關規則探勘。

此外，本論文實作出適合圖書館的資料探勘系統「圖書館借閱記錄探勘系統」，系統的特色為：

- 讓館員輸入最近讀者借閱記錄得到最新的館藏借閱相關規則。
- 針對不同系所的讀者探勘找出相關規則。例如：針對電資學院的學生探勘相關規則，所得的規則以與電腦科學相關者居多。
- 應用「中國圖書分類法」探勘讀者借閱關聯類別，以提昇館藏之借閱率。
- 針對「中國圖書分類法」中不同階層的類別設定不同的最小支持度門檻值，探勘多重最小支持度廣義相關規則。
- 結合交通大學個人化數位圖書資訊環境 PIE@NCTU，將探勘結果導入個人化環境中，當讀者在瀏覽借閱記錄及利用智慧型查詢檢索時，即可將關聯館藏推薦給讀者，作為借閱時的參考，藉此提昇圖書館的服務品質及讀者閱讀的興趣。

在「圖書館借閱記錄探勘系統」發掘出讀者社群關係後，我們運用這些成果達到以下目的：

- 提供讀者借閱館藏的建議：透過探勘讀者借閱關聯性，找出讀者的社群關係，將關聯性館藏推薦給借閱同樣館藏的讀者，作為借閱時之參考。經由本系統所提供的借閱建議，讓讀者瞭解可能有興趣的關聯館藏，增加讀者繼續到館借閱的機會，增進讀者的忠誠度。
- 推薦讀者新進館藏：藉著探勘借閱類別關聯性，經由該讀者的借閱記錄或是個人化系統中的興趣記錄，推薦讀者可能有興趣的關聯類別新書。經由系統探勘類別關聯性，讀者得知可能有興趣的新書，增加讀者借閱館藏的機會，促進館藏流通，提昇館藏借閱率。

第二節 未來研究方向

本論文運用相關規則演算法及其延伸問題來實作圖書館的資料探勘系統。未來研究可分二大方向進行：(1) 增加系統功能 (2) 提昇系統效率。

- 增加系統功能
  - 加入漸進更新(Incremental Update)之演算法

    由於資料探勘是很耗時的，若是有新資料就得重新探勘所有資料將會很浪費時間，而在[6,7,15]中提出如何只針對新進資料作探勘，而不需將所有資料都重新探勘，未來的系統希望加上這類演算法，節省時間成本。
  - 運用其他資料探勘演算法

    本系統只運用了資料探勘演算法中的相關規則演算法，而資料探勘還包含了其他演算法，如循序模式(Sequential Patterns)探勘，分類(Classification)，分群分析(Cluster Analysis)，及趨勢分析(Trend and Evolution

Analysis)等等，未來系統中可加入這些探勘演算法，根據讀者的借閱記錄資訊瞭解讀者的同質性，將讀者分群，或是依據讀者的借閱記錄預測可能有興趣館藏等功能。

◆ 增加探勘項目

本論文只分析出圖書與圖書間的關聯性，未來還可以設計探勘讀者與圖書的關係及讀者與讀者的關係。藉由分析讀者與圖書的關係，可瞭解該讀者所喜好的圖書類別，亦可獲知讀者興趣，進而針對讀者可能的興趣作相關館藏推薦。探勘讀者與讀者間的關係有助於瞭解讀者的同質性，可藉此將讀者分群組，將相同群組所借閱的館藏互相推薦給同群組的其他人。

◆ 增加廣義相關規則演算法的階層分類

本論文只運用「中國圖書分類法」當成廣義相關演算法的分類階層，未來還可運用「美國國會圖書分類法」針對英文書籍探勘廣義相關規則找出英文書籍類別關聯性。

◆ 增加以多重最小支持度相關規則演算法探勘不同類項目

本系統是應用多重最小支持度廣義相關規則探勘，未來還可針對不同類別的館藏設定不同的最小項目支持度探勘，如 DVD、VCD、CD、錄影帶、錄音帶與書籍設定不同的最小項目支持度探勘相關規則。

■ 提昇系統效率

◆ 封閉式頻繁項目集

目前本論文所採用的方法是先產生所有頻繁項目集，再計算所有頻繁項目集的封閉性，刪去多餘的項目集，保留封閉式頻繁項目集。但是這樣一來，會花費不必要的時間在探勘已經包含在其他項目集中的子項目集。若是可以找出適合的封閉式相關規則演算法應用在 H-Mine 上，如 Pei 等學

者所提出的 CLOSET[13]演算法或是 Zaki 等學者所提的 CHARM[21,22,23]演算法，直接探勘封閉式相關規則，即可省去刪除多餘項目集的步驟，以有效率的方式探勘封閉式項目集。

◆ 廣義相關規則演算法

目前系統是利用 H-Mine 演算法，再加上二個最佳化的條件，一是調整標頭表格中母體與子體或祖先與後裔同時出現且支持度又相同的項目，刪除母體(祖先)的項目，只保留子體(後裔)的項目；另一則是，在列出頻繁項目集時，必須測試項目集中的所有項目是否有子體包含母體或後裔包含祖先的情形，確定頻繁項目集是最精簡的。然而，即使採用了我們的最佳化原則探勘，這樣的作法仍耗費了許多時間，因此，未來希望可以找到更有效率的廣義相關規則演算法，以提昇系統效能。

參考文獻

[1] P. Adriaans, and D. Zantinge. "Data Mining," Addison-Wesley, Harlow, 1996.

[2] R. Agrawal, T. Imielinski and A. Swami. "Mining Association Rules between Sets of Items in Large Databases," Proc. of the 1993 ACM SIGMOD Conference, 1993.

[3] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules," Proc. of the 20th VLDB Conference, 1994.

[4] R. C. Agarwal, C.C Aggarwal, and V. V. V. Prasad. "A Tree Projection Algorithm for Generation of Frequent Item Sets," Journal of Parallel and Distributed Computing 61, 350-371, 2001.

[5] M. S. Chen, J. Han, and P. S. Yu. "Data Mining: An Overview from a Database Perspective," IEEE Transactions on Knowledge and Data Engineering, 1996.

[6] 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.

[7] C. I. Ezeife, and Y. Su. "Mining Incremental Association Rules with Generalized FP-Tree," Candian Conference on AI 2002.

[8] A. Kent et al. "Use of Library Materials: the University of Pittsburgh Study," PA.: Pittsburgh University, 1979.

[9] J. Han, J. Chiang, S. Chee, J. Chen, Q. Chen, S. Cheng, W. Gong, M. Kamber, K. Koperski, G. Liu, Y. Lu, N. Stefanovic, L. Winstone, B. Xia, O. R. Zaiane, S. Zhang, and H. Zhu, "DBMiner: A System for Data Mining in Relational Databases and Data Warehouses," Proc. CASCON'97: Meeting of Minds, Toronto, Canada, November 1997

[10] J. Han, J. Pei, and Y. Yin. "Mining Frequent Patterns without Candidate Generation," Proc. of the ACM-SIGMOD 2000 Conference on Management of Data, Dallas, May 2000.

[11] Jiawei Han and Micheline Kamber. "Data Mining: Concepts and Techniques," Morgan Kaufmann, 2000.

[12] Bing Liu, Wynne Hsu and Yiming Ma. "Mining Association Rules with Multiple Minimum Supports." ACM SIGKDD 1999, Pages 337 – 341 .

[13] J. Pei, J. Han, and R. Mao. "CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets," Proc. 2000 ACM-SIGMOD Int. Workshop on Data Mining and Knowledge Discovery (DMKD'00)}, Dallas, TX, May 2000

[14] J. Pei, J. Han, H. Lu, S. Nishio, S. Tang, and D. Yang. "H-Mine: Hyper-Structure

Mining of Frequent Patterns in Large Databases," Proc. 2001 Int. Conf. on Data Mining (ICDM'01)}, San Jose, CA, Nov. 2001.

[15] N. L. Sarda and N. V. Srinivas. "An Adaptive Algorithm for Incremental Mining of Associatin Rules". IEEE, 1998.

[16] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In VLDB' 95, pages 432–443.

[17] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and performance improvements," IBM Research Division Almaden Research Center, 1995.

[18] R. Srikant and R. Agrawal, "Mining generalized association rules." VLDB, 1995.

[19] R. Srikant and R. Agrawal, "Mining generalized association rules." Future Generation Computer Systems, 1997.

[20] M. C. Tseng, and W. Y. Lin. "Mining Generalized Association Rules with Multiple Supports." Data Warehousing and Knowledge Discovery 2001 .

[21] M. J. Zaki and C. J. Hsiao. "CHARM: An Efficient Algorithm for Closed Association Mining," In Technical Report 99-10, Computer Science, Rensselaer Polytechnic Institute, 1999.

[22] M. J. Zaki. "Generating Non-Redundant Association Rules," 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, August 2000

[23] M. J. Zaki and C. J. Hsiao. "CHARM: An Efficient Algorithm for Closed Itemset Mining," In Proc. 2002 SIAM Int. Conf. Data Mining, Arlington, VA, April 2002.

[24] **卜小蝶，「以圖書借閱記錄探勘加強圖書資源利用之探討」，中國圖書館學會會報，第** 66 **期，頁** 59-72，2001**.**

[25] **楊雅雯，「個人化數位圖書資訊環境—以** PIE@NCTU **為例」，交通大學資訊科學研究所碩士論文，**2001.

[26] **吳安琪，「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」，交通大學資訊科學研究所碩士論文，**2001.

[27] **柯皓仁，楊雅雯，吳安琪，楊維邦，「圖書館自動化系統中個人化及群體化資訊服務之設計」.**

[28] **賴永祥，「中國圖書分類法」，**2001 **增訂八版，文華圖書管理資訊股份有限公司.**

附錄一：相關規則探勘結果(部分)

借閱資料量：1998.05.01~2002.04.30　　**最小支持度：**0.001

館藏名稱	借閱次數
龍槍傳奇.第五部, 龍槍傳奇.第六部, 龍槍傳奇.第二部, 龍槍傳奇.第三部, 龍槍傳奇.第四部	30
龍槍傳奇.第五部, 龍槍傳奇.第六部, 龍槍傳奇.第二部, 龍槍傳奇.第四部, 龍槍傳奇.第一部	28
龍槍傳奇.第五部, 龍槍傳奇.第六部, 龍槍傳奇.第二部, 龍槍傳奇.第三部,龍槍傳奇.第一部	27
名流劍客沒羽箭.第三部, 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	27
龍槍傳奇.第五部, 龍槍傳奇.第六部, 龍槍傳奇.第二部, 龍槍傳奇.第三部, 龍槍傳奇.第四部, 龍槍傳奇.第一部	26
大地飛鷹.第四部, 大地飛鷹.第三部, 大地飛鷹.第二部, 大地飛鷹.第一部, 大地飛鷹.第五部	25
龍槍傳奇.第五部, 龍槍傳奇.第六部, 龍槍傳奇.第二部, 龍槍傳奇.第四部	33
龍槍傳奇.第五部, 龍槍傳奇.第六部, 龍槍傳奇.第三部, 龍槍傳奇.第四部	32
龍槍傳奇.第五部, 龍槍傳奇.第六部, 龍槍傳奇.第二部, 龍槍傳奇.第三部	31
名流劍客沒羽箭.第三部, 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	29
龍槍傳奇.第五部, 龍槍傳奇.第六部, 龍槍傳奇.第二部, 龍槍傳奇.第一部	29
龍槍傳奇.第二部, 龍槍傳奇.第三部, 龍槍傳奇.第四部, 龍槍傳奇.第一部	28
Harry Potter and the sorcerer's stone, Harry Potter and the Chamber of Secrets, Harry Potter and the prisoner of Azkaban, Harry Potter and the goblet of fire	28
飄香箭雨.第一部, 飄香箭雨.第二部, 飄香箭雨.第三部, 飄香劍雨.第四部	27
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	27
伊達政宗.(二),人取之卷, 伊達政宗.(三),醍醐夢之卷, 伊達政宗.(一),黎明之卷, 伊達政宗.(四),黃金日本島之卷	26

館藏名稱	借閱次數
大地飛鷹.第四部, 大地飛鷹.第三部, 大地飛鷹.第二部, 大地飛鷹.第五部	26
劍玄錄.第四部, 劍玄錄.第三部, 劍玄錄.第二部, 劍玄錄.第一部	25
白玉老虎.第四部, 白玉老虎.第一部, 白玉老虎.第二部, 白玉老虎.第三部	24
劍玄錄.續.第一部, 劍玄錄.續.第四部, 劍玄錄.續.第三部,劍玄錄.續.第二部	24
二人證據:惡童三部曲.(二), 惡童日記:惡童三部曲.(一), 第三謊言:惡童三部曲.(三)	53
Harry Potter and the sorcerer's stone, Harry Potter and the Chamber of Secrets, Harry Potter and the prisoner of Azkaban	35
Harry Potter and the sorcerer's stone, Harry Potter and the Chamber of Secrets, Harry Potter and the goblet of fire	33
Harry Potter and the sorcerer's stone, Harry Potter and the prisoner of Azkaban , Harry Potter and the goblet of fire	32
龍槍傳奇.第二部, 龍槍傳奇.第三部, 龍槍傳奇.第一部	31
龍槍傳奇.第二部, 龍槍傳奇.第四部, 龍槍傳奇.第一部	30
Harry Potter and the Chamber of Secrets , Harry Potter and the prisoner of Azkaban, Harry Potter and the goblet of fire	30
飄香箭雨.第二部, 飄香箭雨.第三部, 飄香劍雨.第四部	29
飄香箭雨.續.第一部, 飄香箭雨.續.第二部, 飄香箭雨.續.第四部	28
江湖奇譚.第三部, 江湖奇譚.第二部, 江湖奇譚.第一部	28
劍玄錄.第三部, 劍玄錄.第二部, 劍玄錄.第一部	27
德川家康:戀慕秋雨.３, 德川家康:歸雁.４, 德川家康:雌伏之虎.５	25
白玉老虎.第一部, 白玉老虎.第二部, 白玉老虎.第三部	25
劍玄錄.續.第四部, 劍玄錄.續.第三部, 劍玄錄.續.第二部	25
黎明篇.上, 野望篇.上, 野望篇.下	23
仙河飲馬, 淨土之春, 天夢飄香	23
紅色警戒 (下)=Thin Red Line, 紅色警戒 (上)=Thin Red Line	87
二人證據:惡童三部曲.(二), 第三謊言:惡童三部曲.(三)	83
二人證據:惡童三部曲.(二), 惡童日記:惡童三部曲.(一)	64
惡童日記:惡童三部曲.(一), 第三謊言:惡童三部曲.(三)	61
Harry Potter and the sorcerer's stone, Harry Potter and the Chamber of Secrets	45
伊達政宗.(二),人取之卷, 伊達政宗.(一),黎明之卷	43
蘇菲的世界.上, 蘇菲的世界.下	40

館藏名稱	借閱次數
MATLAB 入門引導, PC MATLAB 入門與實例應用	38
三少爺的劍.第一部, 三少爺的劍.第二部	37
Verilog 硬體描述語言:a guide to digital design and synthesis, Verilog 硬體描述語言數位電路設計實務	36
德川家康:破曉之前.１, 德川家康:亂世鴛鴦.２	35
VHDL 數位系統電路設計, VHDL 與數位邏輯設計	35
MATLAB 入門及應用, Matlab 程式語言入門	33
鋼穴.上,鋼穴.下	33
Matrix=駭客任務, 8MM [DVD]=8 厘米	32
計算機結構:計量接近(下)=Computer Architecture A Quantitative Approach,計算機結構:計量接近(上)=Computer Architecture A Quantitative Approach	32
MATLAB 5 專業設計技巧, MATLAB 入門及應用	31
大地飛鷹.第二部, 大地飛鷹.第一部	31
飄香箭雨.續.第三部, 飄香箭雨.續.第四部	31
Verilog 硬體描述語言:a guide to digital design and synthesis 精通 Verilog 數位系統設計與合成	30
裸陽.下, 裸陽.上	30
仙河飲馬, 淨土之春	30
Wideband CDMA for third generation mobile communications CDMA systems engineering handbook	29
Latex 使用介紹:功能豐富的排版系統, LATEX 排版系統實務入門	29
你愛我嗎?, 好想結個婚:都會男女愛情極短篇	29
社會科學研究方法.(上), 社會科學研究方法.(下)	29
永不妥協[DVD]=Erin Brockovich, 綠色奇蹟 [DVD]=The Green Mile	29
城邦暴力團.壹,城邦暴力團.貳	27
Latex 使用介紹:功能豐富的排版系統, Latex 入門與應用實務	26
荊楚爭雄記.上,荊楚爭雄記.下	26
精通 Java 2,基礎篇,精通 Java 2,進階篇	25
仙河飲馬, 魔女國	24
仙河飲馬, 聖域干戈	24
VHDL 與數位邏輯設計, VHDL 與數位電路設計	23
裸陽.上, 鋼穴.上	23
MATLAB 5 專業設計技巧, Matlab 程式語言入門	23
甘露,廚房	23
喜歡, 彷彿	23

附錄二：身份類別相關規則探勘結果(部分)

借閱資料量：2001.01.01~2002.4.30　　**最小支持度：**0.001

身份類別：電機資訊學院的大學生及研究生

館藏名稱	借閱次數
德川家康:破曉之前.１, 德川家康:戀慕秋雨.３, 德川家康:歸雁.４, 德川家康:雌伏之虎.５, 德川家康:流星.６, 德川家康:雙鶴圖.８, 德川家康:瘋狂之夜.７	5
伊達政宗.(一),黎明之卷,伊達政宗.(二),人取之卷,伊達政宗.(三),醍醐夢之卷,伊達政宗.(四),黃金日本島之卷,伊達政宗.(五),蒼穹之鷹之卷	6
康熙大帝.上,玉宇呈祥, 康熙大帝.下,玉宇呈祥, 康熙大帝.上,亂起蕭牆, 康熙大帝.下,亂起蕭牆, 乾隆皇帝:風華初露.上	5
大地飛鷹.第三部, 大地飛鷹.第五部, 大地飛鷹.第一部, 大地飛鷹.第四部, 大地飛鷹.第二部	5
柴可夫斯基:第一號鋼琴協奏曲; 拉赫曼尼諾夫:第二號鋼琴協奏曲 布拉姆斯：匈牙利舞曲;德佛札克：斯拉夫舞曲、詼諧奇想曲 拉威爾：鋼琴協奏曲、加斯帕之夜;普羅高菲夫：第三號鋼琴協奏曲 莫札特:鋼琴協奏曲第十七、二十一、六號	5
康熙大帝.上,玉宇呈祥, 康熙大帝.下,玉宇呈祥, 康熙大帝.上,亂起蕭牆, 康熙大帝.下,亂起蕭牆	8
德川家康:亂世鴛鴦.２, 德川家康:破曉之前.１, 德川家康:戀慕秋雨.３, 德川家康:歸雁.４	5
德川家康:雌伏之虎.５, 德川家康:流星.６, 德川家康:雙鶴圖.８, 德川家康:瘋狂之夜.７	7
曹操大傳.陸,赤壁之戰, 曹操大傳.肆, 山崩雲裂, 曹操大傳.貳,頭角崢嶸, 曹操大傳.伍,馳騁沙場	5
第二基地.下, 第二基地.上, 基地與帝國.下, 基地與帝國.上	6
黑暗精靈.第五冊,旅居 Sojourn(上),黑暗精靈.第四冊,流亡 Exile(下), 黑暗精靈.第六冊,旅居 Sojourn(下),黑暗精靈.第三冊,流亡 Exile(上)	5
第三謊言:惡童三部曲.(三), 二人證據:惡童三部曲.(二), 惡童日記:惡童三部曲.(一)	9
Verilog硬體描述語言:a guide to digital design and synthesis,Verilog硬體描述語言數位電路設計實務,精通 Verilog數位系統設計與合成	7
RF power amplifiers for wireless communications, Microwave circuit design using linear and nonlinear techniques, Fundamentals of RF circuit design : with low noise oscillators	6

館藏名稱	借閱次數
Linux 完整安裝與設定, Linux 完整安裝與設定(CD-ROM-1), Linux 完整安裝與設定(CD-ROM-2)	5
布拉姆斯:第一.三號弦樂四重奏, 德布西：海、牧神午後前奏曲;拉威爾：死公主孔雀舞曲, 葛利格：皮爾金組曲;西貝流士：憂傷圓舞曲、芬蘭頌	5
VHDL 與數位邏輯設計, VHDL 數位系統電路設計, VHDL 與數位電路設計	5
沈船,盜墓,老貓	5
Verilog 硬體描述語言:a guide to digital design and synthesis, Verilog 硬體描述語言數位電路設計實務	18
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	17
VHDL 與數位邏輯設計,VHDL 數位系統電路設計	14
VHDL 與數位邏輯設計,VHDL 與數位電路設計	13
WinSock 網路程式設計之鑰=Key to WinSock Network Programming, 深入 Internet WinSock 設計	12
Verilog 硬體描述語言:a guide to digital design and synthesis, 精通 Verilog 數位系統設計與合成	11
RF power amplifiers for wireless communications, Microwave circuit design using linear and nonlinear techniques	11
Verilog 硬體描述語言數位電路設計實務, 精通 Verilog 數位系統設計與合成	11
C++ Builder 5 徹底研究, 精通 C++ Builder 5.0	10
圖控式程式語言 LabVIEW, LabVIEW 基礎篇=LabVIEW For Everyone	10
Delta-Sigma data converters:theory, design, and simulation, Top-down design of high-performance sigma-delta modulators	9
Effective C++國際中文版, More Effective C++國際中文版	9
數位影像處理-活用 Matlab, 以 MATLAB 透視 DSP	9
Harry Potter and the sorcerer's stone, Harry Potter and the Chamber of Secrets	9
JBuilder 入門學習手冊, Java 程式設計快樂上手-使用 JBuilder 5	9
LINUX 核心研究篇=Linux Kernel Internals,Linux 的核心與程式設計	8
Visual C++入門進階:從 C++物件導向到視窗程式設計, Visual C++ 6 視窗程式設計經典	7

附錄三：廣義相關規則探勘結果(部分)

借閱資料量：2001.05.01~2002.04.30　　**最小支持度：**0.025

科學類、應用科學類階層設定為小數點後三位，其他類設定為小數點後二位

類別	借閱次數
應用科學類, 社會科學類, 科學類, 語文類, 總類, 數學	457
應用科學類, 社會科學類, 科學類, 語文類, 普通叢書, 數學	440
應用科學類, 社會科學類, 科學類, 語文類, 總類, 電腦科學	424
應用科學類, 社會科學類, 科學類, 語文類, 數學, 商學總論	420
應用科學類, 社會科學類, 科學類, 語文類, 現代叢書, 數學	410
應用科學類, 社會科學類, 科學類, 語文類, 數學	736
應用科學類, 科學類, 語文類, 總類, 數學	731
應用科學類, 科學類, 語文類, 普通叢書, 數學	703
應用科學類, 科學類, 語文類, 總類, 電腦科學	686
應用科學類, 社會科學類, 科學類, 語文類, 電腦科學	685
應用科學類, 科學類, 語文類, 現代叢書, 數學	661
應用科學類, 科學類, 語文類, 普通叢書, 電腦科學	660
應用科學類, 科學類, 語文類, 現代叢書, 電腦科學	623
應用科學類, 科學類, 語文類, 現代叢書:通俗用, 電腦科學	622
應用科學類, 社會科學類, 科學類, 總類, 數學	612
社會科學類, 科學類, 語文類, 總類, 數學	594
應用科學類, 社會科學類, 科學類, 普通叢書, 數學	578
應用科學類, 科學類, 語文類, 數學, 商學總論	571
社會科學類, 科學類, 語文類, 普通叢書, 數學	571
應用科學類, 社會科學類, 科學類, 總類, 電腦科學	569
社會科學類, 科學類, 語文類, 總類, 電腦科學	544
應用科學類, 社會科學類, 科學類, 現代叢書, 數學	539
應用科學類, 社會科學類, 科學類, 普通叢書, 電腦科學	537
應用科學類, 社會科學類, 科學類, 現代叢書:通俗用, 數學	537
社會科學類, 科學類, 語文類, 現代叢書, 數學	536
應用科學類, 科學類, 語文類, 電腦科學, 商學總論	534
社會科學類, 科學類, 語文類, 現代叢書:通俗用, 數學	534
應用科學類, 社會科學類, 科學類, 語文類, 總類	532
社會科學類, 科學類, 語文類, 普通叢書, 電腦科學	522
應用科學類, 科學類, 特種文藝, 總類, 數學	519
應用科學類, 社會科學類, 科學類, 語文類, 普通叢書	513

類別	借閱次數
應用科學類, 社會科學類, 科學類, 現代叢書, 電腦科學	503
應用科學類, 科學類, 特種文藝, 普通叢書, 數學	503
應用科學類, 社會科學類, 科學類, 現代叢書:通俗用, 電腦科學	501
社會科學類, 科學類, 語文類, 現代叢書, 電腦科學	493
社會科學類, 科學類, 語文類, 現代叢書:通俗用, 電腦科學	492
應用科學類, 科學類, 特種文藝, 總類, 電腦科學	487
應用科學類, 社會科學類, 科學類, 特種文藝, 數學	484
應用科學類, 科學類, 語文類, 數學, 企業管理	481
應用科學類, 社會科學類, 科學類, 語文類, 現代叢書	478
應用科學類, 社會科學類, 科學類, 語文類, 現代叢書:通俗用	476
應用科學類, 科學類, 特種文藝, 現代叢書, 數學	476
應用科學類, 科學類, 特種文藝, 現代叢書:通俗用, 數學	475
應用科學類, 科學類, 特種文藝, 普通叢書, 電腦科學	472
應用科學類, 科學類, 語文類, 史地類, 數學	469
應用科學類, 科學類, 總類, 數學, 商學總論	464
應用科學類, 社會科學類, 科學類, 語文類, 商學總論	462
應用科學類, 社會科學類, 科學類, 特種文藝, 電腦科學	450
應用科學類, 科學類, 語文類, 藝術類, 數學	450
應用科學類, 科學類, 語文類, 電腦科學, 企業管理	449
應用科學類, 科學類, 特種文藝, 現代叢書, 電腦科學	447
應用科學類, 科學類, 特種文藝, 現代叢書:通俗用, 電腦科學	446
應用科學類, 科學類, 語文類, 數學, 哲學類	443
科學類, 語文類, 總類, 數學, 哲學類	440
應用科學類, 科學類, 語文類, 史地類, 電腦科學	435
應用科學類, 科學類, 總類, 電腦科學, 商學總論	434
應用科學類, 科學類, 普通叢書, 數學, 商學總論	433
社會科學類, 科學類, 特種文藝, 總類, 數學	432
科學類, 語文類, 總類, 史地類, 數學	430
應用科學類, 科學類, 小說, 總類, 數學	425
科學類, 語文類, 普通叢書, 數學, 哲學類	425
應用科學類, 科學類, 語文類, 藝術類, 電腦科學	423
科學類, 語文類,普通叢書,史地類,數學	421
應用科學類, 科學類, 語文類, 電腦科學, 哲學類	419
社會科學類, 科學類, 特種文藝, 普通叢書, 數學	417
應用科學類, 科學類, 語文類, 數學	1325
應用科學類, 科學類, 語文類, 電腦科學	1235

類別	借閱次數
科學類, 語文類, 總類, 數學	1094
科學類, 語文類, 普通叢書, 數學	1051
應用科學類, 科學類, 總類, 數學	1015
科學類, 語文類, 總類, 電腦科學	1003
社會科學類, 科學類, 語文類, 數學	999
科學類, 語文類, 現代叢書, 數學	980
科學類, 語文類, 現代叢書:通俗用, 數學	978
科學類, 語文類, 普通叢書, 電腦科學	965
應用科學類, 科學類, 總類, 電腦科學	951
應用科學類, 科學類, 普通叢書, 數學	949
社會科學類, 科學類, 語文類, 電腦科學	922
科學類, 語文類, 現代叢書, 電腦科學	902
科學類, 語文類, 現代叢書:通俗用, 電腦科學	901
應用科學類, 科學類, 普通叢書, 電腦科學	892
應用科學類, 科學類, 現代叢書, 數學	890
應用科學類, 科學類, 現代叢書:通俗用, 數學	885
應用科學類, 科學類, 語文類, 總類	871
應用科學類, 社會科學類, 科學類, 語文類	849
應用科學類, 科學類, 特種文藝, 數學	845
應用科學類, 科學類, 語文類, 普通叢書	839
應用科學類, 科學類, 現代叢書, 電腦科學	838
應用科學類, 科學類, 現代叢書:通俗用, 電腦科學	833
社會科學類, 科學類, 總類, 數學	815
應用科學類, 科學類, 語文類, 現代叢書	790
應用科學類, 科學類, 特種文藝, 電腦科學	790
應用科學類, 科學類, 語文類, 現代叢書:通俗用	788
應用科學類, 社會科學類, 科學類, 商學總論	780
科學類, 特種文藝, 總類, 數學	772
社會科學類, 科學類, 普通叢書, 數學	767
社會科學類, 科學類, 總類, 電腦科學	748
科學類, 特種文藝, 普通叢書, 數學	746
社會科學類, 科學類, 現代叢書, 數學	720
社會科學類, 科學類, 現代叢書:通俗用, 數學	717
應用科學類, 社會科學類, 科學類, 總類	709
科學類, 特種文藝, 總類, 電腦科學	709
應用科學類, 社會科學類, 數學, 商學總論	706

附錄四：多重最小支持度廣義相關規則探勘結果(部分)

借閱資料量：2001.05.01~2002.04.30

科學類、應用科學類階層設定為小數點後三位，其他類設定為小數點後二位

分類各階層最小項目支持度分別為：0.25, 0.2, 0.15, 0.1, 0.05, 0.025, 0.0125

類別	借閱次數
程式語言,系統程式設計與程式,高階程式語言	948
資料處理及電腦科學,各種應用程式套裝軟體,介面與通訊	932
應用科學類,科學類	4629
應用科學類,數學	3932
現代叢書:通俗用,電腦科學	1792
程式語言,資料處理及電腦科學	1183
電腦應用及其程式,資料處理及電腦科學	1102
電腦應用及其程式,程式語言	1072
電腦應用及其程式,介面與通訊	1047
程式語言,系統程式設計與程式	1042
資料處理及電腦科學,各種應用程式套裝軟體	975
各種應用程式套裝軟體,介面與通訊	952
電腦應用及其程式,系統程式設計與程式	950
電腦應用及其程式,高階程式語言	944
高階程式語言,介面與通訊	933
程式語言,各種應用程式套裝軟體	926
高階程式語言,各種應用程式套裝軟體	846
系統程式設計與程式,各種應用程式套裝軟體	831
高階程式語言,各種應用程式套裝軟體	826
電腦應用及其程式,特殊電腦方法	815