使用加權移動視窗模式之圖書資料探勘

Data Mining for Library Borrowing History Records Based on the Weighted Sliding Window Model

蔡秀滿* 莊宛螢
明新科技大學
資訊管理研究所
新竹縣 新豐鄉
pauray@must.edu.tw

Pauray S.M. Tsai* and Wan-Ying Zhuang
Institute of Information Management
Minghsin University of Science and Technology
Hsin-Feng, Hsinchu 304,Taiwan
pauray@must.edu.tw

Received 17 April 2006; Revised 15 July 2006 ; Accepted 21 August 2006

摘 要

資料探勘是資料庫領域中一個熱門的研究課題。隨著各種應用的興起，使用資料探勘技術從大量資料中發掘出來的資訊，對於商業管理和決策支援等方面都有莫大的助益。在本論文中，我們以關連法則探勘中著名的 Apriori 演算法爲基礎，將加權移動視窗的觀念應用在圖書資料探勘的研究上。我們所提出的 “加權移動視窗模式” 可以讓使用者設定探勘的資料範圍、視窗個數、視窗時間的長短、以及各視窗的權重，這種讓使用者能夠對某些特定時段的資料給予較高權重的作法，將可以讓資料探勘的結果更符合使用者的需求。此外，我們也應用循序樣式探勘技術中著名的 AprioriAll 演算法，從歷史借閱記錄中找出讀者在不同時間點所借閱的書籍之先後關係，同樣地，我們也加入加權移動視窗的觀念，讓使用者可以自行決定相關參數。另外，我們將 “同好” 的觀念加入探勘所得到的推薦結果中，以期爲圖書借閱者提供更多的資訊。

關鍵詞 ： 資料探勘、關連法則、循序樣式、加權移動視窗

* 通訊作者

ABSTRACT

Data mining is a hot research topic in the database community. With the emergence of new applications, the information discovered from the mining will be useful for business management and decision support. In this paper, we propose the idea of the weighted sliding window and apply it on the research of library data mining, based on the well-known Apriori algorithm. The weighted sliding window model allows the user to specify the range of data, the number of windows, the size of a window, and the weight of each window. The approach of allowing users to specify a higher weight for more significant data will make the mining result closer to the user's requirement. We also apply the idea of the weighted sliding window on the consideration of the relationships among sequential borrowing records for a user, based on the well-known AprioriAll algorithm. Besides, the factor of peers is considered in the recommendation, in order to provide more information to users.

Keywords: Data Mining, Association Rule, Sequential Pattern, Weighted Sliding Window

一、前 言

資料探勘(data mining)在資料庫領域中的應用非常地廣泛[7-9,12]。其中，將資料探勘的技術應用在圖書館藏推薦系統已成爲一項重要的研究[2-4,6,11]。自古以來，文物書籍的館藏傳承著人類長久累積的知識，若無法有效發揮圖書館的功能而浪費寶貴的資源，將是一件非常可惜的事情。

隨著網際網路技術的快速進步以及 WWW 的大量使用，圖書館的數位化已成爲發展的趨勢，其中，推薦系統是最重要的服務之一。圖書館可以使用推薦系統建立個人化的功能，藉以提供讀者更多相關的資訊。過去以來，使用資料探勘技術在圖書館的研究爲數不少，其中論文[2]所提出的個人化館藏推薦系統是一項重要的研究成果。他們以關連法則探勘(association rule mining)的技術爲基礎，在讀者的借閱歷史資料中找出關連法則，再以這些結果分析讀者和借閱書籍之間的關係。根據讀者的興趣，利用推薦系統中常用的協力式過濾(collaborative filtering)可以找出適合的推薦書目清單，再以內容導向過濾(content-based filtering)的方式，將推薦清單依照讀者的興趣做個人化排序後再推薦給讀者。這套推薦系統已被實作在交通大學浩然圖書館的 myLibrary 個人資訊環境中。

論文[1]則是以圖書館的經費、行銷與營運爲出發點，使用貝氏分類法(naive Bayes classification)將資料分類，配合客戶關係管理(customer relationship management)的概念及閱覽者借閱館藏資源的借閱記錄，使用關連法則探勘的技術來發掘圖書館閱覽群組的潛在特徵。而論文[5]也是以個人化服務爲出發點，運用資料探勘的技術從同類讀者的歷史借閱記錄中發掘關連法則。綜合這些研究，主要都是著重在個人化服務的提供，使用各種方法來發掘出讀者間的喜好關係，進而產生推薦值排序給使用者參考。

使用購物籃分析的關連法則探勘技術，可以適當地被應用在讀者借閱記錄的歷史資料之分析，產生“建議那些書籍可以同時借閱”的法則。而循序樣式探勘(sequential pattern mining)的技術加入時間點的考量後，可以產生“建議後續每個階段可以借閱的書籍順序”的法則，對於讀者後續書籍的借閱提供了有用的資訊。先前相關研究所提出的圖書借閱資料探勘，都是將所有的借閱資料平等對待，忽略了以時間來考量的重要性。由於歷史借閱資料的數量通常非常的龐大，若將所有的借閱資料平等對待，將無法顯現出某些特定時段資料的重要性。因此，若能將探勘的資料分爲若干視窗，並且給予不同的權重來區分其重要性，將可以使探勘的結果更加顯著。在本論文中，我們提出 “加權移動視窗”(weighted sliding window)的觀念，讓使用者可以對探勘的資料設定視窗的個數、視窗時間的長短、以及各個視窗的權重，以期使資料探勘的結果能夠更符合使用者的需求。

在關連法則探勘方面，我們在 Apriori 演算法[7]中加入“加權移動視窗”的觀念，產生建議共同借閱的書籍。例如，我們可以對 SARS 期間的資料給予較高的權重，配合關連法則探勘來了解可能和 SARS 相關的書籍之間的關係。在循序樣式探勘方面，我們以 AprioriAll 演算法[8]爲基礎，加入移動時間視窗和時間限制條件，讓使用者可以指定序列中相鄰借閱書籍的最大平均時間間隔(maximum average time gap)和最小平均時間間隔(minimum average time gap)，發掘出不同時間點所借閱的書籍之先後關係。例如，我們可以將大學生在大一到大四期間每年借閱的書籍資料，在循序樣式探勘的方法上加入加權移動視窗的觀念，將更能了解學生循序學習和成長的歷程。

本論文共分爲六節。第二節介紹與本論文研究相關的方法。第三節說明我們所提出的加權移動視窗模式，以及如何將此模式應用在圖書資料的關連探勘。在第四節，我們使用一個範例來說明加權移動視窗模式之圖書資料探勘的方法。在第五節，我們討論視窗的個數與權重值的設定對探勘結果的影響。第六節爲結論。

二、相關的研究方法

在這一節中，我們介紹與本論文研究相關的兩個基本的探勘演算法：Apriori 演算法[7]和 AprioriAll 演算法[8]。我們將以此爲基礎，在下一節中提出使用加權移動視窗模式之圖書資料探勘演算法。

2.1 圖書資料關連法則探勘

在圖書資料探勘的應用中，關連法則探勘的目的是從借閱資料中，找出相關連的書籍項目。在書籍的借閱資料中，單次的讀者借閱記錄告訴我們個別讀者的喜好，當累積大量的借閱記錄之後，我們就可以分析出整體讀者的借閱習慣。這些分析過的資訊，具有相當高的實用性，可以推薦給其他讀者作爲參考，以提升整體服務的品質。

2.1.1 相關定義

假設在圖書館借閱記錄資料庫中，每一筆借閱記錄包含借閱編號與一組被借閱的書籍項目，而一組書籍項目所成的集合稱之爲 “項目集” (itemset)。一個項目集 X 的 “支持個數”(support count)被定義爲 “支持項目集 X 的借閱記錄之總數”，而項目集 X 的 “支持度” (support)則是 “支持項目集 X 的借閱記錄之個數佔全部借閱記錄總數的比例”。最小支持度和資料庫中借閱記錄總數的乘積即是最小支持個數(minimum support count)。關連法則探勘的問題可以細分爲底下兩個子問題。首先，找出所有支持度大於或等於最小支持度的項目集，我們稱之爲“大型項目集”(large itemset)，一個包含 k 個項目的項目集被稱爲 k-項目集 (k-itemset)。接著，從大型項目集中產生信心水準大於或等於最小信心水準的關連法則。假設 Z 爲大型項目集，所有形式爲 $X \rightarrow Y$，滿足 $X \cup Y = Z$ 、 $X \cap Y = \phi$ 以及信心水準大於或等於最小信心水準的關連法則都會被產生。很明顯地，一旦所有大型項目集被發掘之後，關連法則的產生將變得非常直接。

2.1.2 Apriori 演算法

Apriori 的基本精神是使用前一個階段所發掘的大型項目集來產生下一個階段的大型項目集。也就是說，先找出所有大型 1-項目集，再利用大型 1-項目集找出候選 2-項目集，決定出所有大型 2-項目集之後，再利用大型 2-項目集找出候選 3-項目集，然後決定出所有大型 3-項目集，依此類推下去，直到下一個階段無任何大型項目集產生爲止。

候選 k-項目集是結合大型(k-1)-項目集產生的。令 W_1 和 W_2 是兩個大型(k-1)-項目集，我們用 $W_i[j]$代表項目集 W_i 中的第 j 個項目。假設項目集中的項目已依遞增的方式排序完成。若 W_1 和 W_2 的前 k-2 個項目皆相同，且 $W_1[k\text{-}1] < W_2[k\text{-}1]$，則 W_1 和 W_2 將被結合成一個候選 k-項目集，亦即$\{W_1[1], W_1[2], \ldots, W_1[k\text{-}1], W_2[k\text{-}1]\}$。Apriori 使用一個重要的性質來減少搜尋的空間：一個大型項目集的任何子集合也必定是大型項目集。因此，若項目集 W 有任何一個大小爲 k-1 的子集合不是大型(k-1)-項目集，則 W 必定不是大型 k-項目集，因此就可以將 W 刪除。

2.2 圖書資料循序樣式探勘

時間序列分析中，循序樣式探勘主要的目的是找出顧客在不同時間點所購買的物品先後之關係，其中最基礎且最著名的一個演算法爲－AprioriAll 演算法。我們可以將其應用在圖書借閱資料之分析，以發掘讀者在不同時間點所借閱的書籍之先後關係。

2.2.1 相關定義

一個序列(sequence)是項目集(itemsets)的有序串列。一個包含 k 個項目集的序列稱之爲 k-序列。給定一組序列 S，若序列 S 不被包含在其它序列中，則稱它爲 “最大序列”。每一個最大序列則代表一個循序樣式(sequential pattern)。對於每一位讀者，將他所有的借閱記錄根據借閱時間排序所得到的結果就是一個 “讀者序列”。一個序列 S 的“支持度”

被定義爲“包含 S 的讀者序列之總數佔全部讀者總數的比例”。若序列滿足使用者所設定的 “最小支持度”之限制，則稱之爲 “大型序列”(large sequence)。一旦所有的大型序列都被發掘之後，就可以很容易地產生所有的最大序列，而循序樣式探勘的問題即是找出所有的循序樣式，也就是所有的最大序列。

2.2.2 AprioriAll 演算法

AprioriAll 演算法主要包含下列五個步驟：

- 步驟一：排序階段。首先將圖書借閱資料庫根據讀者帳號和借閱時間做排序。依據排序之後的結果，即可產生讀者序列資料庫。
- 步驟二：大型項目集產生階段。每一個大型項目集就是一個大型 1-序列。我們可以應用 Apriori 演算法來產生大型 1-序列，項目集的支持度被定義爲 “曾在某次借閱中包含此項目集的讀者序列佔全部讀者總數的比例”。若項目集 x 出現在同一讀者一次以上的借閱記錄中，則支持個數只能增加一次。當大型項目集產生之後，可以將它們對應至一組連續的整數，以減少將來比對和計算的時間。
- 步驟三：轉換階段。根據下列幾個原則將讀者序列做轉換，以加速後續計算的速度：
  (1) 若借閱記錄不包含任何大型項目集，則將它從讀者序列中移除。
  (2) 若讀者序列不包含任何大型項目集，則將它從轉換後的資料庫移除，但是在進行相關計算時，讀者總數仍維持不變。
  (3) 將讀者序列中的每一筆借閱記錄，用被包含在借閱記錄中的大型項目集所成的集合來取代，表示方式爲 $\{I_1,I_2, ...,I_n\}$，其中 $I_j\,(1\le j \le n)$ 是大型項目集。例如，假設大型項目集有 (1)、(2) 和 (1, 2)，則借閱記錄 (1, 2) 將被大型項目集的集合 {(1) (2) (1, 2)} 所取代。
- 步驟四：大型序列產生階段。使用轉換後的資料庫來產生所有大型序列。AprioriAll 的基本精神和 Apriori 演算法類似，它是使用前一個階段所發掘的大型序列來產生下一個階段的大型序列。首先，我們找出所有大型 1-序列，再利用大型 1-序列產生候選 2-序列，藉由計算候選序列的支持度之後，即可產生大型 2-序列，然後再利用大型 2-序列產生候選 3-序列，再決定出大型 3-序列，依此類推下去，直到下一個階段無任何大型序列產生爲止。AprioriAll 演算法也是使用 “結合” 大型 (k-1)-序列的方式來產生候選 k-序列。令 X_1 和 X_2 是兩個大型 (k-1)-序列。我們用 $X_i[j]$ 代表序列 X_i 中的第 j 個項目集。若 X_1 和 X_2 的前 k-2 個項目集皆相同，則 X_1 和 X_2 將被結合成一個候選 k-序列，亦即 $<X_1[1], X_1[2], ..., X_1[k\text{-}1], X_2[k\text{-}1]>$。另外，候選 k-序列 $<X_1[1], X_1[2], ..., X_1[k\text{-}2], X_2[k\text{-}1], X_1[k\text{-}1]>$ 也會產生，這是因爲在序列中的項目集之位置不同即代表不同的序列，所以 X_1 和 X_2 結合時要分別考慮這兩個序列。AprioriAll 使用類似 Apriori 的性質來減少候選序列的個數：若候選序列 C 有任何大小爲 k-1 的子序列不是大型(k-1)-序列，則 C 必定不是大型 k-序列。
- 步驟五：最大序列產生階段。從所有大型序列中找出最大序列。

三、使用加權移動視窗模式之探勘方法

在這一節中，我們將以 Apriori 演算法和 AprioriAll 演算法爲基礎，加入“加權移動視窗”的觀念，讓使用者可以設定視窗的個數、視窗時間的長短、以及各個視窗的權重，使資料探勘的結果能夠更符合使用者的需求。

3.1 動視窗模式之圖書資料關連探勘

在移動視窗模式的環境中，關連法則探勘的工作是考慮離現在時間 T 以內的資料記錄，時間 T 的大小可以由使用者自行設定。我們所設計的加權移動視窗模式之架構主要有下列兩個性質：

(1) 移動視窗的大小是以時間來定義，而非以包含的借閱筆數來決定。以時間來定義的目的是爲了避免在不同的時間點，相同數量的借閱筆數所涵蓋的時間範圍差異太大。
(2) 使用者可以根據資料的重要性，將時間 T 區分爲若干視窗，並且對各視窗設定不同的權重。通常，愈接近目前時間的資料應該對整體探勘的結果具有較大的影響力，因此，使用 “加權移動視窗模式” 可以讓資料探勘的結果更符合使用者的需求。

假設使用者設定的探勘範圍是距離現在 T 時間內的借閱資料，T 被區分爲 w_1、w_2、… w_n 個區間，並且分別設定加權值爲 α_1 、 α_2 、… 和 α_n ，$\sum_{j=1}^{n}\alpha_j = 1$。假設在時間範圍 T 之內的借閱記錄總筆數爲 c，最小支持度爲 s（亦即最小支持個數爲 $c\times s$），則 “最小加權支持個數” 定義爲$(c\times s)/n$。令$\{x_1,x_2,...,x_k\}$爲包含 k 個書籍項目的大型 k-項目集。$\mathrm{CB}_j(\{x_1,x_2,...,x_k\})$被定義爲在視窗 w_j 中包含項目集$\{x_1,x_2,...,x_k\}$的借閱記錄之個數。$\sum_{j=1}^{n}(\mathrm{CB}_j(\{x_1,x_2,...,x_k\})\times\alpha_j)$ 被定義爲$\{x_1,x_2,...,x_k\}$的 “加權支持個數”。若$\{x_1,x_2,...,x_k\}$的加權支持個數大於或等於最小加權支持個數，則$\{x_1,x_2,...,x_k\}$是一個 “加權大型 k-項目集”。加權移動視窗模式之關連法則探勘的目的，就是要找出所有 “加權大型項目集”。

我們提出一個發掘“加權大型項目集”的演算法，稱之爲 W-Apriori。方法如下:

- 步驟一：根據使用者所提供的搜尋關鍵字以及探勘範圍，從原始資料庫中過濾出所有符合條件的借閱記錄，作爲後續探勘工作所要使用的資料庫。
- 步驟二：使用 2.1.2 節的 Apriori 演算法，找出所有支持個數大於或等於最小支持個數的大型項目集。
- 步驟三：假設$\{x_1,x_2,...,x_k\}$爲一個大型 k-項目集。根據使用者設定的探勘範圍、視窗大小和加權值，計算$\{x_1,x_2,...,x_k\}$的加權支持個數。若加權支持個數大於或等於最小加權支持個數，則 $\{x_1,x_2,...,x_k\}$ 是一個加權大型 k-項目集。
- 步驟四：假設所有加權大型項目集中，加權支持個數的最大值爲 X。以最大的加權支持個數爲基準，計算各加權大型項目集的推薦程度。若加權大型 k-項目集$\{x_1,x_2,...,x_k\}$的加權支持個數爲 Y，則$\{x_1,x_2,...,x_k\}$的推薦程度爲 Y/X。亦即推薦書

籍 $x_1,x_2,...$和 x_k一起被借閱的程度爲 Y/X。

3.2. 加權移動視窗模式之圖書資料循序樣式探勘

一個讀者序列被表示成 $<s_1,s_2,...,s_n>$，其中 s_j $(1\le j \le n)$ 代表此讀者在某次借閱中所借閱的書籍項目之集合。我們說序列 $<a_1,a_2,...,a_m>$ 被包含在讀者序列 $<s_1,s_2,...,s_n>$ 中，假如存在整數 $i_1<i_2< ...<i_m$ 滿足 $a_1\subseteq s_{i_1}$，$a_2\subseteq s_{i_2}$，...，$a_m\subseteq s_{i_m}$。在循序樣式探勘中，我們不希望包含 $a_1,a_2,...,a_m$ 的書籍項目集 $s_{i_1},s_{i_2}, ...,s_{i_m}$之借閱時間間隔太長或太短，因此允許使用者設定相鄰項目集 a_i 和 a_{i+1} 的最大平均時間間隔和最小平均時間間隔。對序列 $<a_1,a_2,...,a_m>$ 來說，讀者序列 $<s_1,s_2,...,s_n>$ 包含它之平均時間間隔爲 $\sum_{j=1}^{m-1}(s_{ij}$和$s_{i(j+1)}$的時間間隔$)/m-1$。假設最大平均時間間隔爲 T'，最小平均時間間隔爲 T''，亦即合理的平均時間間隔必須大於或等於 T''且小於或等於 T'。在 T' 和 T'' 的區間中又可區分爲 u_1、u_2、… u_k 個視窗，並且分別設定加權值爲 β_1、β_2、…、β_k，$\sum_{j=1}^{k}\beta_j=1$。假設最小支持個數爲 c，則“序列最小加權支持個數”定義爲 c/k。假設 $<x_1,x_2,...,x_n>$ 是一個大型序列，CS_j $(<x_1,x_2,...,x_n>)$被定義爲包含$<x_1,x_2,...,x_n>$且平均時間間隔落在視窗 u_j 的讀者序列之個數。$\sum_{j=1}^{n}$ $(CS_j(<x_1,x_2,...,x_n>)\times\beta_j)$被定義爲$<x_1,x_2,...,x_n>$的“序列加權支持個數”。若 $<x_1,x_2,...,x_n>$ 的序列加權支持個數大於或等於“序列最小加權支持個數”，則 $<x_1,x_2,...,x_n>$ 是一個“加權大型 n-序列”。加權移動視窗模式之循序樣式探勘的目的就是要找出所有“加權大型序列”。

我們提出一個發掘“加權大型序列”的演算法，稱之爲 W-AprioriAll。方法如下:

- 步驟一：應用 3.1 節 W-Apriori 演算法，找出所有加權大型項目集，亦即所有大型 1-序列。將它們對應至一組連續的整數，以減少將來比對和計算的時間。
- 步驟二：使用 2.2.2 節的 AprioriAll 演算法，找出所有支持個數大於或等於最小支持個數的大型 n-序列($n\ge 2$)。
- 步驟三：假設$<x_1,x_2,...,x_n>$爲一個大型 n-序列($n\ge 2$)。根據使用者設定的最大時間間隔、視窗個數和加權值，計算 $<x_1,x_2,...,x_n>$ 的序列加權支持個數。若序列加權支持個數大於或等於“序列最小加權支持個數”，則 $<x_1,x_2,...,x_n>$ 是一個加權大型 n-序列。
- 步驟四：假設所有加權大型序列中，序列加權支持個數的最大值爲 X。以最大的序列加權支持個數爲基準，計算各加權大型序列的推薦程度。若加權大型 n-序列$<x_1,x_2,...,x_n>$的序列加權支持個數爲 Y，則$<x_1,x_2,...,x_n>$的推薦程度爲 Y/X，亦即推薦書籍 $x_1,x_2,...$和 x_n依序被借閱的程度爲 Y/X。

3.3 考慮同好的特徵

我們可以根據讀者的背景，以相同背景的讀者之借閱記錄來當作推薦參考的依據，如此將可以提供另一個層面的資訊給予讀者參考。同好是所有讀者中，和借閱查詢者具有相同背景或嗜好的族群。

依同好的特徵來推薦圖書資料之演算法如下：

- 步驟一：根據 3.1 節的 W-Apriori 演算法和 3.2 節的 W-AprioriAll 演算法所產生的加權大型項目集和加權大型序列，分別依同好類別區分其各別的加權支持個數。
- 步驟二：分別將加權大型項目集和加權大型序列的推薦程度乘以同好佔全體借閱者的比例，即爲同好之推薦程度。

四、範例說明

表一：探勘資料庫

編號	學號	讀者科系	讀者借閱資料序列
1	A001	資管	<(B501)[91/11];(B501,B502,B503)[91/12];(B802)[92/5]>
2	A003	資管	<(B501,B502,B503)[91/9];(B802)[94/2];(B501,B503)[94/3]>
3	A006	電子	<(B801)[92/5];(B701)[92/6];(B802,B804)[94/7];(B103)[94/9]>
4	A008	電子	<(B501,B502,B503)[91/12];(B503)[92/3];(B802)[92/3];(B804)[94/11]>
5	A010	電子	<(B501)[92/12];(B502,B503)[93/4];(B802)[93/6];(B501,B502,B503)[94/6]>
6	A011	資管	<(B501,B502,B503)[91/10];(B503,B802)[94/1];(B802)[94/7]>
7	A012	電子	<(B103)[90/9];(B501,B502)[90/12];(B503,B802)[94/8];(B802)[94/10]>
8	A014	休閒	<(B501,B502,B503)[92/9];(B502,B503)[93/1];(B103)[94/11];(B801)[94/12]>
9	A016	資管	<(B501,B802)[93/12];(B501,B502,B503)[93/12];(B802)[94/2]>
10	A017	資管	<(B501)[90/10];(B503)[91/12];(B802)[93/11];(B501,B502,B503)[94/5]>
11	A018	資管	<(B501,B502,B503,B802)[92/8];(B502,B503)[93/4]>
12	A020	電子	<(B801)[92/5];(B501,B502,B503)[93/1];(B502,B503)[94/4]>
13	A022	資管	<(B501,B502,B503)[91/11];(B503)[92/1];(B804)[92/1];(B802)[92/2]>
14	A023	休閒	<(B701)[90/11];(B501,B502,B503)[92/12];(B502)[93/4]>
15	A024	休閒	<(B501,B502)[93/9];(B501,B502,B503)[94/8]>

在本節中，我們以表一來說明使用加權移動視窗模式之圖書資料探勘的方法。假設表一是根據讀者搜尋的關鍵字「科技」和探勘範圍 90/1/1 至 94/12/31 所產生的探勘資料庫。在借閱資料中，書名包含關鍵字「科技」的書號如下：B103、B501、B502、B503、B701、B801、B802 和 B804。以編號 “1” 的讀者借閱資料序列來說，學號 “A001” 的讀者有三次借閱記錄，其中(B501)[91/11]表示他在 91 年 11 月借閱了編號 “B501” 這本書。探勘資料庫中，讀者總數爲 15、所有借閱之總筆數爲 50，假設最小支持度爲 0.16。

4.1 W-Apriori 範例

在 W-Apriori 演算法的步驟二，我們使用 Apriori 演算法產生候選項目集，如表二所示，最後所產生的大型項目集，如表三所示。

在步驟三，我們將探勘範圍分爲五個視窗：90 年、91 年、92 年、93 年和 94 年，

表二：候選項目集

項目集	支持個數	支持度	是否符合
(B103)	3	3/50=0.06	X
(B501)	20	20/50=0.4	O
(B502)	20	20/50=0.4	O
(B503)	23	23/50=0.46	O
(B701)	2	2/50=0.04	X
(B801)	3	3/50=0.06	X
(B802)	14	14/50=0.28	O
(B804)	3	3/50=0.06	X
(B501,B502)	15	15/50=0.3	O
(B501,B503)	14	14/50=0.28	O
(B501,B802)	2	2/50=0.04	X
(B502,B503)	17	17/50=0.34	O
(B502,B802)	1	1/50=0.02	X
(B503,B802)	3	3/50=0.06	X
(B501,B502,B503)	13	13/50=0.26	O

表三：所有大型項目集

	大型項目集	支持個數	支持度
大型 1-項目集	(B501)	20	20/50=0.4
	(B502)	20	20/50=0.4
	(B503)	23	23/50=0.46
	(B802)	14	14/50=0.28
大型 2-項目集	(B501,B502)	15	15/50=0.3
	(B501,B503)	14	14/50=0.28
	(B502,B503)	17	17/50=0.34
大型 3-項目集	(B501,B502,B503)	13	13/50=0.26

並且分別設定加權值如下：0.1、0.1、0.1、0.2 和 0.5，合計爲 1。最小加權支持個數=(最小支持個數/視窗個數)=8/5=1.6，每一個大型項目集的“加權支持個數”如表四所示。例如，大型項目集(B501)在 90 年、91 年、92 年、93 年和 94 年的支持個數分別爲 2、6、4、4 和 4，則它的加權支持個數爲 $2\times 0.1+6\times 0.1+4\times 0.1+4\times 0.2+4\times 0.5=4$。我們將大型項目集對應至一組連續的整數，以方便後續的相關計算。在步驟四，以加權大型項目集中加權支持個數最大者爲基準(亦即 (B503))，計算每一個加權大型項目集的推薦程度，如表五所示。以推薦順序爲“6”的加權大型項目集(B501,B503)爲例，它代表“科技管理”和“高科技行銷”被推薦一起借閱的程度爲 57.14%。另外我們可以發現，推薦順序爲“2”的加權大型項目集(B802)，它代表“藍芽科技”，沒有任何和它相關的大型 2-項目集與大型 3-項目集出現。這是因爲“藍芽科技”和其它書籍一起被借閱的次數非常低(如表二所示)，所以無法成爲大型 2-項目集或大型 3-項目集。

表四：大型項目集的加權支持個數

大型項目集	對應的整數	支持個數	90 年借閱次數	*0.1	91 年借閱次數	*0.1	92 年借閱次數	*0.1	93 年借閱次數	*0.2	94 年借閱次數	*0.5	**加權支持個數**	**是否符合**
(B501)	1	20	2	*0.2*	6	*0.6*	4	*0.4*	4	*0.8*	4	*2*	**4**	**O**
(B502)	2	20	1	*0.1*	5	*0.5*	3	*0.3*	7	*1.4*	4	*2*	**4.3**	**O**
(B503)	3	23	0	*0*	6	*0.6*	5	*0.5*	5	*1*	7	*3.5*	**5.6**	**O**
(B802)	4	14	0	*0*	0	*0*	4	*0.4*	3	*0.6*	7	*3.5*	**4.5**	**O**
(B501,B502)	5	15	1	*0.1*	5	*0.5*	3	*0.3*	3	*0.6*	3	*1.5*	**3**	**O**
(B501,B503)	6	14	0	*0*	5	*0.5*	3	*0.3*	2	*0.4*	4	*2*	**3.2**	**O**
(B502,B503)	7	17	0	*0*	5	*0.5*	3	*0.3*	5	*1*	4	*2*	**3.8**	**O**
(B501,B502,B503)	8	13	0	*0*	5	*0.5*	3	*0.3*	2	*0.4*	3	*1.5*	**2.7**	**O**

表五：和「科技」相關的書籍之借閱推薦程度

排名	大型項目集	對應的書籍	加權支持個數	**加權推薦程度**
1	(B503)	高科技行銷	5.6	**5.6/5.6=100%**
2	(B802)	藍芽科技	4.5	**4.5/5.6=80.36%**
3	(B502)	科技行銷	4.3	**4.3/5.6=76.79%**
4	(B501)	科技管理	4	**4/5.6=71.42%**
5	(B502,B503)	科技行銷&高科技行銷	3.8	**3.8/5.6=67.86%**
6	(B501,B503)	科技管理&高科技行銷	3.2	**3.2/5.6=57.14%**
7	(B501,B502)	科技管理&科技行銷	3	**3/5.6=53.57%**
8	(B501,B502,B503)	科技管理&科技行銷&高科技行銷	2.7	**2.7/5.6=48.21%**

4.2 W-AprioriAll 範例

使用表一的探勘資料庫，假設最小支持個數爲 8。首先，根據表四所得到的加權大型項目集，將表一探勘資料庫中的資料依 AprioriAll 演算法的轉換階段，執行資料轉換的工作，其結果如表六所示。以學號 “A012” 爲例，讀者序列爲<(B103) [90/9] ; (B501,B502) [90/12] ; (B503,B802) [94/8] ; (B802) [94/10]>，其中(B103) 不是加權大型項目集，所以將它剔除，序列成爲< (B501,B502) [90/12] ; (B503,B802) [94/8] ; (B802) [94/10] >。接著以對應的整數來表示加權大型項目集，最後轉換的結果爲<{1,2,5}[90/12];{3,4}[94/8];{4}[94/10]>。

在步驟二，使用 AprioriAll 演算法找出所有候選 2-序列，如表七所示。符合最小支持個數的大型 2-序列爲<1,2>、<1,3>、<1,4>、<1,7>、<2,3>、<2,4>、<3,3>、<3,4>、<5,3>和<7,3>。表八是所有候選 3-序列，符合最小支持個數的大型 3-序列爲<1,3,4>。

在步驟三，假設最大平均時間間隔爲一年，最小平均時間間隔爲一個月，時間間隔區分爲 4 個視窗，分別爲 1 至 3 個月內、4 至 6 個月內、7 至 9 個月內、10 至 12 個月內，且加權值分別爲 0.5、0.3、0.1 和 0.1，合計爲 1。“序列最小加權支持個數” = (最小

支持個數/視窗個數)=8/4=2，大型 2-序列和大型 3-序列的 “序列加權支持個數” 分別如表九和表十所示。以學號 “A001” 的讀者序列爲例，< {1} [91/11] ; {1,2,3,5,6,7,8} [91/12] ; {4} [92/5] >包含序列<1,3,4>，其中項目集 1 和 3 的距離爲 1 個月，項目集 3 和 4 的距離爲 5 個月，取其平均值爲(1+5)/2=3，因此它對 <1,3,4> 的支持是隸屬於 “1 至 3 個月內” 的視窗範圍。由表九和表十可以得知加權大型 2-序列爲<1,3>、<1,4>、<2,4>和<3,4>，而加權大型 3-序列爲<1,3,4>。

表六：經過轉換後的探勘資料庫

學號	以對應整數來表示
A001	<{1}[91/11];{1,2,3,5,6,7,8}[91/12];{4}[92/5]>
A003	<{1,2,3,5,6,7,8}[91/9];{4}[94/2];{1,3,6}[94/3]>
A006	<{4}[94/7]>
A008	<{1,2,3,5,6,7,8}[91/12];{3}[92/3];{4}[92/3]>
A010	<{1}[92/12];{2,3,7}[93/4];{4}[93/6];{1,2,3,5,6,7,8}[94/6]>
A011	<{1,2,3,5,6,7,8}[91/10];{3,4}[94/1];{4}[94/7]>
A012	<{1,2,5}[90/12];{3,4}[94/8];{4}[94/10]>
A014	<{1,2,3,5,6,7,8}[92/9];{2,3,7}[93/1]>
A016	<{1,4}[93/12];{1,2,3,5,6,7,8}[93/12];{4}[94/2]>
A017	<{1}[90/10];{3}[91/12];{4}[93/11];{1,2,3,5,6,7,8}[94/5]>
A018	<{1,2,3,4,5,6,7,8}[92/8];{2,3,7}[93/4]>
A020	<{1,2,3,5,6,7,8}[93/1];{2,3,7}[94/4]>
A022	<{1,2,3,5,6,7,8}[91/11];{3}[92/1];{4}[92/2]>
A023	<{1,2,3,5,6,7,8}[92/12];{2}[93/4]>
A024	<{1,2,5}[93/9];{1,2,3,5,6,7,8}[94/8]>

表七：候選 2-序列

候選 2-序列	支持個數	候選 2-序列	支持個數	候選 2-序列	支持個數	候選 2-序列	支持個數
<1,1>	6	<3,1>	3	<5,1>	2	<7,1>	2
<1,2>	9	<3,2>	6	<5,2>	5	<7,2>	5
<1,3>	13	<3,3>	9	<5,3>	9	<7,3>	8
<1,4>	9	<3,4>	9	<5,4>	7	<7,4>	7
<1,5>	5	<3,5>	2	<5,5>	1	<7,5>	1
<1,6>	6	<3,6>	3	<5,6>	2	<7,6>	2
<1,7>	8	<3,7>	5	<5,7>	4	<7,7>	4
<1,8>	5	<3,8>	2	<5,8>	1	<7,8>	1
<2,1>	3	<4,1>	4	<6,1>	1	<8,1>	1
<2,2>	6	<4,2>	3	<6,2>	4	<8,2>	4
<2,3>	10	<4,3>	5	<6,3>	7	<8,3>	7
<2,4>	8	<4,4>	3	<6,4>	6	<8,4>	6
<2,5>	2	<4,5>	3	<6,5>	0	<8,5>	0
<2,6>	3	<4,6>	4	<6,6>	1	<8,6>	1
<2,7>	5	<4,7>	4	<6,7>	3	<8,7>	3
<2,8>	2	<4,8>	3	<6,8>	0	<8,8>	0

表八：候選 3-序列

候選 3-序列	支持個數	候選 3-序列	支持個數	候選 3-序列	支持個數
<1,2,3>	1	<1,3,4>	8	<2,3,4>	4
<1,3,2>	2	<1,4,3>	3	<2,4,3>	2
<1,2,4>	3	<1,3,7>	2	<3,2,3>	0
<1,4,2>	2	<1,7,3>	1	<3,3,2>	0
<1,2,7>	1	<1,4,7>	2	<3,3,4>	3
<1,7,2>	1	<1,7,4>	3	<3,4,3>	3

表九：大型 2-序列的序列加權支持個數

候選 2-序列	支持個數	未滿 1 個月	1-3 個月內	*0.5	4-6 個月內	*0.3	7-9 個月內	*0.1	10-12 個月內	*0.1	超過 1 年	**序列加權支持個數**	**是否≥序列最小加權支持個數**
<1,2>	9	1	1	*0.5*	2	*0.6*	1	*0.1*	1	*0.1*	3	**1.3**	X
<1,3>	13	1	3	*1.5*	1	*0.3*	1	*0.1*	1	*0.1*	6	**2**	O
<1,4>	9	0	3	*1.5*	2	*0.6*	0	*0*	0	*0*	4	**2.1**	O
<1,7>	8	1	1	*0.5*	1	*0.3*	1	*0.1*	1	*0.1*	3	**1**	X
<2,3>	10	0	2	*1*	1	*0.3*	1	*0.1*	1	*0.1*	5	**1.5**	X
<2,4>	8	0	4	*2*	1	*0.3*	0	*0*	0	*0*	3	**2.3**	O
<3,3>	9	0	2	*1*	1	*0.3*	1	*0.1*	0	*0*	5	**1.4**	X
<3,4>	9	0	5	*2.5*	2	*0.6*	0	*0*	0	*0*	2	**3.1**	O
<5,3>	9	0	2	*1*	1	*0.3*	1	*0.1*	1	*0.1*	8	**1.5**	X
<7,3>	8	0	2	*1*	1	*0.3*	1	*0.1*	0	*0*	4	**1.4**	X

表十：大型 3-序列的序列加權支持個數

大型 3-序列	支持個數	未滿 1 個月	1-3 個月內	*0.5	4-6 個月內	*0.3	7-9 個月內	*0.1	10-12 個月內	*0.1	超過 1 年	**序列加權支持個數**	**是否≥序列最小加權支持個數**
<1,3,4>	8	1	4	*2*	0	*0*	0	*0*	0	*0*	3	**2**	O

表十一：加權大型序列之加權推薦程度

排名	加權大型序列	還原	**序列加權支持個數**	**加權推薦程度**	書籍
1	<3,4>	<(B503),(B802)>	3.1	**3.1/3.1=100%**	<(高科技行銷)，(藍芽科技)>
2	<2,4>	<(B502),(B802)>	2.3	**2.3/3.1=74.19%**	<(科技行銷)，(藍芽科技)>
3	<1,4>	<(B501),(B802)>	2.1	**2.1/3.1=67.74%**	<(科技管理)，(藍芽科技)>
4	<1,3>	<(B501),(B503)>	2	**2/3.1=64.52%**	<(科技管理)，(高科技行銷)>
4	<1,3,4>	<(B501),(B503),(B802)>	2	**2/3.1=64.52%**	<(科技管理)，(高科技行銷)，(藍芽科技)>

在所有加權大型序列中，擁有最大序列加權支持個數者爲<3,4>，因此以其爲基準，計算其它加權大型序列的加權推薦程度，如表十一所示。我們可以發現，推薦順序爲“1”的加權大型序列包含大型項目集(B503)和(B802)，它們分別代表“高科技行銷”和“藍芽科技”，但在表五中沒有出現項目集(B503,B802)。這是因爲“藍芽科技”和“高科技行銷”同時一起被借閱的次數非常低，所以無法成爲大型 2-項目集，但是在表十一中，序列<(B503),(B802)> 表示“高科技行銷”被借閱一段時間之後，再借閱“藍芽科技”的次數很高，所以才會有序列<(B503),(B802)>的出現。

4.3 考慮同好的影響

根據 3.3 節考慮同好特徵之演算法，分別對 W-Apriori 演算法和 W-AprioriAll 演算法所產生的加權大型項目集和加權大型序列，依同好類別區分其各別的加權支持個數，如表十二和表十三所示。接下來再依同好之加權支持個數佔全體讀者的比例，計算各科系同好之推薦程度，如表十四和表十五所示。我們可以發現，休閒系對項目集(B802)的加權支持個數爲 0，對序列<(B501),(B802)>、<(B502),(B802)>、<(B503),(B802)>和<(B501),(B503),(B802)>的加權支持個數亦爲 0，因此我們用“x”來作區別，不列入排名，表示它不會被列在推薦結果中。

表十二：各科系對加權大型項目集之加權支持個數

	加權大型項目集	資管系	電子系	休閒系	加權支持個數	加權推薦程度
W-Apriori	<(B501)>	2.1	1	0.9	4	71.42%
	<(B502)>	1.4	1.6	1.3	4.3	76.79%
	<(B503)>	2.6	2.1	0.9	5.6	100%
	<(B802)>	2.7	1.8	0	4.5	80.36%
	<(B501,B502)>	1.2	0.9	0.9	3	53.57%
	<(B501,B503)>	1.7	0.8	0.7	3.2	57.14%
	<(B502,B503)>	1.4	1.5	0.9	3.8	67.86%
	<(B501,B502,B503)>	1.2	0.8	0.7	2.7	48.21%

表十三：各科系對加權大型序列之加權支持個數

	加權大型序列	資管系	電子系	休閒系	加權支持個數	加權推薦程度
W-AprioriAll	<(B501),(B503)>	1.6	0.1	0.3	2	64.52%
	<(B501),(B802)>	1.3	0.8	0	2.1	67.74%
	<(B502),(B802)>	1.3	1	0	2.3	74.19%
	<(B503),(B802)>	1.6	1.5	0	3.1	100%
	<(B501),(B503),(B802)>	1.5	0.5	0	2	64.52%

表十四：考慮同好之加權推薦程度(W-Apriori)

	加權大型項目集	資管系	排名	電子系	排名	休閒系	排名
W-Apriori	<(B501)>	(2.1/4)*71.42%=37%	3	(1/4)*71.42%=18%	5	(0.9/4)*71.42%=16%	2
	<(B502)>	(1.4/4.3)*76.79%=25%	5	(1.6/4.3)*76.79%=29%	3	(1.3/4.3)*76.79%=23%	1
	<(B503)>	(2.6/5.6)*100%=46%	2	(2.1/5.6)*100%=38%	1	(0.9/5.6)*100%=16%	2
	<(B802)>	(2.7/4.5)*80.36%=48%	1	(1.8/4.5)*80.36%=32%	2	0	x
	<(B501,B502)>	(1.2/3)*53.57%=21%	6	(0.9/3)*53.57%=16%	6	(0.9/3)*53.57%=16%	2
	<(B501,B503)>	(1.7/3.2)*57.14%=30%	4	(0.8/3.2)*57.14%=14%	7	(0.7/3.2)*57.14%=12%	3
	<(B502,B503)>	(1.4/3.8)*67.86%=25%	5	(1.5/3.8)*67.86%=27%	4	(0.9/3.8)*67.86%=16%	2
	<(B501,B502,B503)>	(1.2/2.7)*48.21%=21%	6	(0.8/2.7)*48.21%=14%	7	(0.7/2.7)*48.21%=12%	3

表十五：考慮同好之加權推薦程度(W-AprioriAll)

	加權大型序列	資管系	排名	電子系	排名	休閒系	排名
W-AprioriAll	<(B501),(B503)>	(1.6/2)*64.52%=0.52	1	(0.1/2)*64.52%=0.03	4	(0.3/2)*64.52%=0.10	1
	<(B501),(B802)>	(1.3/2.1)*67.74%=0.42	2	(0.8/2.1)*67.74%=0.26	3	0	x
	<(B502),(B802)>	(1.3/2.3)*74.19%=0.42	2	(1/2.3)*74.19%=0.32	2	0	x
	<(B503),(B802)>	(1.6/3.1)*100%=0.52	1	(1.5/3.1)*100%=0.48	1	0	x
	<(B501),(B503),(B802)>	(1.5/2)*64.52%=0.48	1	(0.5/2)*64.52%=0.16	1	0	x

五、討論

在本節中，我們使用表十六的借閱記錄資料庫來說明 Apriori 和 W-Apriori 演算法執行結果的差異，並且討論視窗的個數與權重值的設定對探勘結果的影響。在借閱記錄資料庫中，所有借閱的總筆數爲 50，假設最小支持度爲 0.28，則最小支持個數爲 14。使用 Apriori 演算法進行探勘，所產生的大型項目集如表十七所示。假設我們將探勘的資料範圍分爲五個視窗：90 年、91 年、92 年、93 年和 94 年，並且分別設定加權值如下：0.1、0.1、0.1、0.2 和 0.5，合計爲 1。最小加權支持個數=(最小支持個數/視窗個數)=14/5=2.8。使用 W-Apriori 演算法進行探勘，每一個大型項目集的 “加權支持個數” 如表十八所示。我們發現，(B501)、(B501,B502)、和(B501,B503)是大型項目集，但不是加權大型項目集。這是因爲它們出現在高權重值視窗的次數較少，而導致加權支持個數低於最小加權支持個數。反觀大型項目集(B802)，雖然它的支持個數是所有大型項目集當中最小者，但因爲它出現在高權重值視窗的次數很多，造成它具有最大的加權支持個數。從這個例子中我們可以很清楚的知道，使用加權移動視窗的方法可能產生和 Apriori 演算法不同的執行結果，這是因爲使用者依探勘的需求設定不同的資料權重所造成的影響。

表十六：借閱記錄資料庫

編號	學號	讀者科系	讀者借閱資料序列
1	A001	資管	<(B501)[91/11];(B501,B502,B503)[91/12];(B802)[92/5]>
2	A003	資管	<(B501,B503)[90/3];(B501,B502,B503)[91/9];(B802)[94/2]>
3	A006	電子	<(B801)[92/5];(B701)[92/6];(B802,B804)[94/7];(B103)[94/9]>
4	A008	電子	<(B501,B502,B503)[91/12];(B503)[92/3];(B802)[92/3];(B804)[94/11]>
5	A010	電子	<(B501)[92/12];(B501,B502,B503)[93/2];(B502,B503)[93/4];(B802)[93/6]>
6	A011	資管	<(B501,B502,B503)[91/10];(B503,B802)[94/1];(B802)[94/7]>
7	A012	電子	<(B103)[90/9];(B501,B502)[90/12];(B503,B802)[94/8];(B802)[94/10]>
8	A014	休閒	<(B501,B502,B503)[92/9];(B502,B503)[93/1];(B103)[94/11];(B801)[94/12]>
9	A016	資管	<(B501,B802)[93/12];(B501,B502,B503)[93/12];(B802)[94/2]>
10	A017	資管	<(B501)[90/10];(B503)[91/12];(B501,B502,B503)[93/5];(B802)[93/11]>
11	A018	資管	<(B501,B502,B503,B802)[92/8];(B502,B503)[93/4]>
12	A020	電子	<(B801)[92/5];(B501,B502,B503)[93/1];(B502,B503)[94/4]>
13	A022	資管	<(B501,B502,B503)[91/11];(B503)[92/1];(B804)[92/1];(B802)[92/2]>
14	A023	休閒	<(B701)[90/11];(B501,B502,B503)[92/12];(B502)[93/4]>
15	A024	休閒	<(B501,B502,B503)[90/8];(B501,B502)[93/9]; >

表十七：大型項目集

	大型項目集	支持個數	支持度
大型 1-項目集	(B501)	20	20/50=0.4
	(B502)	20	20/50=0.4
	(B503)	23	23/50=0.46
	(B802)	14	14/50=0.28
大型 2-項目集	(B501,B502)	15	15/50=0.3
	(B501,B503)	14	14/50=0.28
	(B502,B503)	17	17/50=0.34

表十八：大型項目集的加權支持個數(視窗個數=5)

大型項目集	90 年借閱次數 (權重= 0.1)	91 年借閱次數 (權重= 0.1)	92 年借閱次數 (權重= 0.1)	93 年借閱次數 (權重= 0.2)	94 年借閱次數 (權重= 0.5)	**加權支持個數**	**是否符合**
(B501)	4	6	4	6	0	**2.6**	**X**
(B502)	2	5	3	9	1	**3.3**	**O**
(B503)	2	6	5	7	3	**4.2**	**O**
(B802)	0	0	4	3	7	**4.5**	**O**
(B501,B502)	2	5	3	5	0	**2**	**X**
(B501,B503)	2	5	3	4	0	**1.8**	**X**
(B502,B503)	1	5	3	7	1	**2.8**	**O**

表十九：大型項目集的加權支持個數 (視窗個數=10)

大型項目集	90 年 [1-6 月] 借閱次數 (權重 =0.05)	90 年 [7-12 月] 借閱次數 (權重 =0.05)	91 年 [1-6 月] 借閱次數 (權重 =0.05)	91 年 [7-12 月] 借閱次數 (權重 =0.05)	92 年 [1-6 月] 借閱次數 (權重 =0.05)	92 年 [7-12 月] 借閱次數 (權重 =0.05)	93 年 [1-6 月] 借閱次數 (權重 =0.05)	93 年 [7-12]月 借閱次數 (權重 =0.15)	94 年 [1-6 月] 借閱次數 (權重 =0.15)	94 年 [7-12 月] 借閱次數 (權重 =0.35)	**加權支持個數**	**是否符合**
(B501)	1	3	0	6	0	4	3	3	0	0	**1.3**	**X**
(B502)	0	2	0	5	0	3	7	2	1	0	**1.3**	**X**
(B503)	1	1	0	6	2	3	6	1	2	1	**1.75**	**O**
(B802)	0	0	0	0	3	1	1	2	3	4	**2.4**	**O**
(B501,B502)	0	2	0	5	0	3	3	2	0	0	**0.8**	**X**
(B501,B503)	1	1	0	5	0	3	3	1	0	0	**0.8**	**X**
(B502,B503)	0	1	0	5	0	3	6	1	1	0	**1.05**	**X**

視窗個數和權重的設定也會影響探勘的結果。當視窗個數=1 時，權重值亦爲 1，此時 Apriori 所產生的大型項目集和 W-Apriori 所產生的加權大型項目集完全相同，而且項目集的支持個數也等於加權支持個數。若視窗個數大於 1，且權重值平均分配給每個視窗，則 Apriori 所產生的大型項目集也會和 W-Apriori 所產生的加權大型項目集完全相同，但項目集的加權支持個數等於支持個數乘以權重值。當視窗個數大於 1，但視窗權重值不相同時，視窗個數的多寡會影響權重值的分配，進而影響探勘的結果。我們以相同的例子來說明，假設將探勘的資料範圍分爲十個視窗：90 年[1-6 月]、90 年[7-12 月]、91 年[1-6 月]、91 年[7-12 月]、92 年[1-6 月]、92 年[7-12 月]、93 年[1-6 月]、93 年[7-12 月]、94 年[1-6 月]、和 94 年[7-12 月]，並且分別設定加權值如下：0.05、0.05、0.05、0.05、0.05、0.05、0.05、0.15、0.15 和 0.35，合計爲 1。最小加權支持個數=(最小支持個數/視窗個數)=14/10=1.4。使用 W-Apriori 演算法進行探勘，大型項目集的 “加權支持個數” 如表十九所示，其中只有(B503)和(B802)是加權大型項目集。在這個例子中我們可以發現，當視窗個數太多時，會增加權重值設定的困難度(因爲每個視窗分配到的權值將變得更小)，而且也可能造成符合條件的項目集之個數變少。綜合以上的討論，我們建議在設定參數時，視窗的個數不要超過十個，因爲會造成權重值的分配過於繁細。視窗個數設定在 3 到 6 個之間應該是比較好的選擇。

六、結論

在本論文中，我們提出加權移動視窗的觀念，將其應用在圖書資料探勘的研究上。“加權移動視窗模式” 可以讓使用者設定探勘的資料範圍、視窗的個數、視窗時間的長短、以及各視窗的權重，如此可以讓資料探勘的結果更符合使用者的需求。視窗的個數和視窗時間的長短決定了探勘資料的範圍。例如對五年內的資料進行探勘，可以設定每

個視窗的時間爲一年，視窗的個數爲 5。而視窗權重的設定，通常對於愈接近現在時間的視窗會給予較高的權重，代表這段期間的資料對探勘的結果具有較大的影響力，當然也有例外的情況。例如，在三年前有一批和讀者興趣相關的新書出版，我們也可以對包含那段時間的視窗設定比其它視窗更高的權重。因此，使用者可以根據探勘的目的來設定適合的參數值。如同在關連法則探勘的研究領域中，最小支持度的設定也沒有一定的標準，完全依使用者的需要來決定。

本論文的主要貢獻是應用現有的探勘技術，再加入 “加權移動視窗模式” 的觀念，從現有的借閱記錄中，發掘出 “哪些書籍較常被共同借閱”，以及 “書籍被借閱的先後順序之關係”。在關連法則探勘方面，我們以 Apriori 演算法爲基礎，加入“加權移動視窗”的觀念，產生建議共同借閱的書籍和加權推薦程度。在循序樣式探勘方面，則以 AprioriAll 演算法爲基礎，加入移動時間視窗和時間限制條件，讓使用者可以指定序列中相鄰借閱書籍的最大平均時間間隔和最小平均時間間隔，產生建議書籍借閱的順序和加權推薦程度。此外，我們也考慮同好間的影響力，希望可以爲圖書借閱者提供更多的資訊。未來我們將製作一個整合我們的方法爲基礎的系統，對所提出的 W-Apriori 和 W-AprioriAll 演算法的效能與實際效益進行驗證和評估。

誌　謝

本論文之研究獲得國科會專題研究計畫 NSC 94-2213-E-159-003 的補助。

參考文獻

[1] 王毓菁，*圖書館閱覽者群組潛在特徵探勘資訊系統*，華梵大學工業管理學系碩士班學位論文, 2002。

[2] 余明哲，*圖書館個人化館藏推薦系統*，國立交通大學資訊科學研究所碩士論文, 2003。

[3] 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*，國立交通大學資訊科學研究所碩士論文, 2001。

[4] 陳揮明，*數位圖書館上個人化檢索與推薦服務之設計與實作*，南華大學資訊管理學研究所碩士論文, 2004。

[5] 曹健華，*應用資料探勘技術於數位圖書館之個人化服務及管理*，南華大學資訊管理學研究所碩士論文, 2003。

[6] 戴玉旻，*圖書館借閱記錄探勘系統*，國立交通大學資訊科學研究所碩士論文, 2002。

[7] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” *Proceedings of the VLDB Conference*, pp. 487-499, 1994.

[8] R. Agrawal and R. Srikant, “Mining Sequential Patterns,” *Proceedings of IEEE International Conference on Data Engineering*, pp. 3-14, 1995.

[9] J. Han, J. Pei and Y. Yin, “Mining Frequent Patterns without Candidate Generation,” *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 1-12, 2002.

[10] S. Ko and J. Lee, “User Preference Mining Through Collaborative Filtering and Content Based Filtering in Recommender System,” *Lecture Notes in Computer Science*, Vol. 2455, pp. 244-253, 2002.

[11] B. J. Mirza, B. J. Keller and N. Ramakrishnan, “Studying Recommendation Algorithms by Graph Analysis,” *Journal of Intelligent Information System*, Vol. 20, No. 2, pp. 131-160, 2003.

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