

逢 甲 大 學
資 訊 工 程 學 系
專 題 研 究 報 告

條件式關聯規則與序列規則探勘系統

學 生：資訊四丙 李兆偉
資訊四丙 廖健峰
資訊四丙 林榮章
資訊四丙 林宏軒
指導教授：林明言 老師

中華民國九十五年十一月

目錄

目錄.....	i
圖目錄.....	iii
表目錄.....	iv
摘要.....	v
第一章 導論.....	1
1.1 研究動機.....	1
1.2 研究目標.....	2
1.3 資料探勘概念簡述.....	3
第二章 系統環境建構.....	4
2.1 硬體配置.....	4
2.2 軟體環境.....	4
2.2.1 FreeBSD.....	4
2.2.2 JDK-1.5.0p1.....	7
2.2.3 JSP.....	7
2.2.4 JavaBean.....	8
2.2.5 Apache-2.1.4.....	8
2.2.6 MySQL-4.1.10a.....	8
2.2.7 Jakarta-Tomcat-5.0.30.....	9
2.2.8 PHP5.....	9
2.2.9 phpMyAdmin-2.6.1.3.....	9
2.2.10 lynx-2.8.6d11.....	9
2.3 資料庫規劃.....	10
2.3.1 ER Model.....	10
2.3.2 Data Mining Table.....	12
第三章 相關理論與技術研究.....	15
3.1 探勘關聯規則.....	15
3.1.1 關聯規則的問題分解.....	15
3.1.2 基本定義與符號.....	15
3.1.3 Apriori 探勘演算法.....	16
3.1.4 功能函數 apriori-gen.....	20
3.1.5 探索關聯規則.....	21
3.2 探勘循序樣式.....	22
3.2.1 循序樣式的基本簡介.....	22
3.2.2 研究問題敘述.....	22
3.2.3 探勘演算法 GSP.....	23
3.3 時間限制條件與樣式增長.....	25
3.3.1 時間限制型循序樣式.....	25
3.3.2 探勘演算法比較.....	25
3.3.3 相關術語說明.....	26

3.3.4 滿足時間限制條件的支持度定義.....	26
3.4 相關軟體操作.....	28
3.4.1 ARTool	28
第四章 每學期工作進度與重點.....	30
4.1 三年級上學期.....	30
4.1.1 相關環境建立.....	30
4.1.2 研讀書籍與文獻.....	30
4.1.3 定期開會討論.....	30
4.1.4 網頁系統架設.....	30
4.2 三年級下學期.....	31
4.2.1 資料庫與資料表建立.....	31
4.2.2 報名國科會專題研究計畫.....	31
4.2.3 關聯規則演算法實作.....	31
4.2.4 循序樣式概念初探.....	31
4.3 四年級上學期.....	32
4.3.1 系友專訪活動.....	32
4.3.2 循序樣式功能實作.....	32
4.3.3 專題發表相關準備.....	32
第五章 專題研究實作成果.....	33
5.1 系統流程.....	33
5.2 網頁式操作介面.....	34
5.2.1 系統首頁.....	34
5.2.2 Apriori 主要功能介紹	35
5.2.3 Sequential 主要功能介紹.....	40
5.2.4 顯示商品項目明細.....	43
5.2.5 顯示顧客購買明細.....	44
5.2.6 使用者上傳測試檔.....	45
5.2.7 查詢資料庫.....	46
第六章 總結.....	47
參考資料.....	49
附 錄.....	50
附錄 A 工作分配表	50

圖目錄

圖 1	ER Model Diagram	11
圖 2	Relational Schema	12
圖 3	Apriori 演算法 [8]	16
圖 4	Apriori 演算法實例	18
圖 5	候選項目產生方式例	19
圖 4	Join 的 SQL 指令	20
圖 7	刪除不合條件的候選項目	20
圖 8	產生規則的方法	21
圖 9	包含性檢驗範例 [9]	24
圖 10	Two pattern-growth forms [7]	26
圖 11	序列包含性關係的範例 [14]	27
圖 12	ARTool 測試資料	28
圖 13	ARTool 執行 Apriori 結果	29
圖 14	系統流程圖	33
圖 15	首頁 web 介面	34
圖 16	index.jsp - Apriori 主功能介面選擇	35
圖 17	search.jsp 執行限制條件而產生的 Apriori 結果	36
圖 18	association.jsp 顯示 Apriori 關聯式規則探勘的結果	37
圖 19	association_district.jsp 顯示 Apriori 限制型探勘介面	38
圖 20	association_choose.jsp 顯示商品選取介面	38
圖 21	association_district.jsp 顯示 Apriori 限制型探勘執行結果	39
圖 22	index.jsp - Sequential 主功能介面選擇	40
圖 23	sequential.jsp 執行 sequential + time constraints 而產生的結果	41
圖 24	Sequential 序列探勘結果	42
圖 25	showitem.jsp 所顯示的商品詳細資訊	43
圖 26	showitem.jsp 顯示哪些顧客購買了此商品	44
圖 27	sequential_showitem.jsp 顯示哪些顧客購買了此商品	44
圖 28	upload.htm 使用者自行上傳 Apriori 測試檔	45
圖 29	searchbyuser.jsp 上傳檔執行結果	45
圖 30	執行查詢資料庫結果	46

表目錄

表 1	符號說明.....	15
表 2	資料庫內交易資料.....	17
表 3	各種探勘演算法比較 [7].....	25

摘 要

「資料探勘」一詞在最近幾年非常熱門，是針對在資料庫中尋找特定資訊的一門技術，這項技術的功能，可以在大量的數據中，對資料進行分類、分群或找出高頻率樣式，如同在一堆黃土當中尋找有價值的金砂。目前很多學者紛紛投入此領域進行研究。透過資料探勘技術，可以得到有興趣的樣式或過去未知的知識。

資料探勘涉及許多學科領域，包括資料庫技術、資料倉儲、人工智慧、機器學習、神經網絡、統計學、模式識別、知識庫系統、知識獲取、資訊檢索、客戶關係管理、高性能計算和資料視覺化等。其應用也相當廣泛，例如在銀行業、保險業、信用卡公司等行業，近年來對於詐欺行為的偵測非常關心，因為詐欺行為每年都造成這些行業非常可觀的損失。使用資料探勘技術，可以從一些信用不良的客戶資料中找出相似特徵並預測可能的詐欺行為，達到減少損失之目的。

本小組的研究目標，在於參考資料探勘的知名演算法來進行實作，其成果可以用來找出大量資料中的頻繁項目集合。我們模擬管理者把交易資料存放於資料庫中，然後經由網頁式系統找出其關聯規則。在實務上，這個系統可以應用到許多行業，幫助其進行決策規劃。例如，零售業者為了要瞭解客戶的消費行為，哪些產品會一起被購買，或是客戶在買了某樣產品之後，在多少時間之內會買另一樣產品等等。本研究的假定情境，就是模擬零售業（購物籃）的交易記錄來進行資料探勘。

本專題前期的重點在於關聯規則的探勘，主要參數包含最小支持度與最小信賴度，從頻繁項目集合中推導出關聯規則；在後期的研究重點則著重於循序樣式，加入多種時間限制條件的探勘功能，包括最大時間間隔、最小時間間隔、滑動時間窗、持續期間等等。

利用本系統的輸出結果，業者可以更有效的決定銷售政策，或是店內商品的擺設方式，同時也可以用來評估商店的促銷活動或價格調整的成效。

關鍵字：資料探勘、頻繁項目、關聯規則、循序樣式、最小支持度、最小信賴度

第一章 導 論

1.1 研究動機

「資料探勘」一詞在最近幾年非常熱門，是針對在資料庫中尋找特定資訊的一門技術，這項技術的功能，可以在大量的數據中，對資料進行分類、分群或找出高頻率樣式，如同在一堆黃土當中尋找有價值的金砂，目前很多學者紛紛投入此領域進行研究。透過資料探勘技術，可以得到特定屬性的有價值訊息。

資料探勘的目的可以說是“從資料中探勘出知識”，其中的一個重要因素是從大量資料中探勘的特色。畢竟，探勘是一個很生動的術語，它抓住了從大量的、未經加工的材料中發現少量金塊這一過程的特色。舉例來說，商店用條碼掃描的每一筆交易記錄，這些大量的交易記錄就可以透過資料探勘來分析其關聯性，可以就此得知某些商品擺在一起可能會銷售比較好。而這只是一小部分的應用而已，還可應用到商務、科學和行政事務的電腦化等。

資料探勘涉及許多學科領域，包括資料庫技術、資料倉儲、人工智慧、機器學習、神經網絡、統計學、模式識別、知識庫系統、知識獲取、資訊檢索、客戶關係管理、高性能計算和資料視覺化等。又例如在銀行業、保險業、信用卡公司等行業，近年來對於詐欺行為的偵測非常關心，因為詐欺行為每年都造成這些行業非常可觀的損失。使用資料探勘技術，可以從一些信用不良的客戶資料中找出相似特徵並預測可能的詐欺行為，達到減少損失之目的。

由此可知，資料探勘的範圍相當廣泛，針對資料庫數量龐大且雜亂的資料經過適當的篩選與分析，就可以將處理過而獲得的資訊進行有效利用。設計者將其應用到軟體或網頁上，產生圖形化介面，可以讓使用者透過點選形式來方便使用。因為透過網際網路較容易表現資料探勘的技術，因而我們希望能夠藉由網頁來實作一個資料探勘的系統。透過本專題的製作，瞭解這項新興且先進的軟體技術。

1.2 研究目標

本小組的研究目標，在於參考資料探勘的知名演算法來進行實作，其成果可以用來找出大量資料中的頻繁項目集合。我們模擬管理者把交易資料存放於資料庫中，然後經由網頁式系統找出其關聯規則。在實務上，這個系統可以應用到許多行業，幫助其進行決策規劃。例如，零售業者為了要瞭解客戶的消費行為，哪些產品會一起被購買，或是客戶在買了某樣產品之後，在多少時間之內會買另一樣產品等等。本研究的假定情境，就是模擬零售業（購物籃）的交易記錄來進行資料探勘。利用本系統的輸出結果，業者可以更有效的決定銷售政策，或是店內商品的擺設方式，同時也可以用來評估商店的促銷活動或價格調整的成效。

本專題研究在實作上相當充滿挑戰性。舉市場購物籃應用為例，假設商品種類有 1000 種，而目的是探勘資料中的關聯規則，也就是哪些商品會同時一起被購買。考慮長度為 1 的項目集合，其可能有 1000 取 1 的組合方式，考慮長度為 2 種商品的項目，其組合有 1000 取 2 種可能性，依此類推，最後數目將接近天文數字。在現實中 1000 種商品並不過份，對於許多大賣場或批發商而言，商品種類都是動輒成千上萬種。如果欲探勘循序樣式，也就是哪些商品組合出現後會有哪些商品出現的順序關係，其困難度又要往上提升一個層級；因為循序樣式不只有組合，還加入排列的觀念。假設來源資料同樣包含 1000 種商品，那麼考慮長度為 1 個商品的樣式，其可能有 1000 取 1 種組合方式，考慮長度為 2 個商品的樣式，則有 1000 取 1 乘以 1000 取 1 加上 1000 取 2 組合的可能性，同樣依此類推。而且在循序樣式的概念中，相同商品項目可以多次出現在同一個樣式中，同一位顧客也可能有不同時間點的數筆交易，如此複雜程度將更加提升。

本網頁式系統的基本功能包含選擇演算法、選擇資料表，和設定最小支持度值、最小信賴度值等選項，使用者可依自己感興趣的參數值來設定。我們建立資料庫來儲存資料，把資料鍵入資料庫表單裡，再透過程式和 JSP 網頁做連結，並且把經過處理過後的資料顯示出來。針對不同使用者的需求，會有基本型介面與複雜型介面的雙重設計。本專題前期的重點在於關聯規則的探勘，主要參數包含最小支持度與最小信賴度，從頻繁項目集合中推導出關聯規則；在後期的研究重點則著重於循序樣式，加入多種時間限制條件的探勘功能，包括最大時間間距、最小時間間距、滑動時間窗、持續期間等等。

1.3 資料探勘概念簡述

以下是與資料探勘相關之簡介：

- 探勘關聯規則的技術
 - 用來發現資料庫中屬性間的有趣關聯，對購物籃分析已是一種普遍技術
 - 可以有一或多個輸出屬性，所有潛在的有趣組合都可以被找出來
- Apriori 方法會檢查資料庫(例如購物籃)中的內容並推論出規則，是探勘關聯規則的一個出色演算法（稍後在 3.1 節將會詳細介紹）
 - 這種演算法並不處理數值屬性的資料
- 親合力分析(affinity analysis)：決定事物在一起與否的一般化過程
- 典型的應用就是市場購物籃分析(market basket analysis)
 - 希望在購物期間，決定顧客可能同時購買的項目
- 市場購物籃分析的輸出是有關於顧客購買行為的關聯集合
 - 此關聯以特別的規則集合(即關聯法則)的形式表示
- 關聯法則和傳統的分類規則不同
 - 規則為單一屬性，但關聯法則可能包含一個或多個屬性值
- 例如，在兩種雜貨店儲存產品（牛奶與麵包）之中的顧客購買趨勢裡決定是否有任何有趣的關聯。可能的關聯性舉例：
 1. 假如顧客購買牛奶，他們也會買麵包
 2. 假如顧客購買麵包，他們也會買牛奶
- 第一個關聯性告訴我們：購買牛奶的顧客也可能會購買麵包
- 信賴度(confidence)：購買牛奶的事件會導致購買麵包的可能性有多大？
 - 假設總共有 10,000 筆顧客交易涉及牛奶的購買，其中有 5,000 筆包含有麵包的購買，則信賴度為 50% ($\text{confidence} = 5,000 / 10,000 = 50\%$)
- 考慮第二個關聯性，並不會給予和第一個關聯性相同的資訊
 - 因為兩個關聯性所組成的交易領域(顧客)並不相同
- 支持度(support)：針對特定的頻繁項目集合中，資料庫（或資料表）中包含其所有項目的交易紀錄之數目

第二章 系統環境建構

2.1 硬體配置

中央處理器：Pentium III 733 MHz

記憶體容量：768 MB

硬碟容量：200 GB

2.2 軟體環境

2.2.1 FreeBSD

相關簡介：

簡單地來說，FreeBSD 是一套可以在 Alpha/AXP, AMD64 及 Intel® EM64T, i386™ IA-64, PC-98, UltraSPARC® 上執行的 UN*X-like 作業系統，它是根據 U.C. Berkeley 所開發出來的 “4.4BSD-Lite”，並加上了許多

“4.4BSD - Lite2” 的增強功能。它同時也間接使用了 U.C. Berkeley 所開發出來並由 William Jolitz 移植到 i386 的 “Net/2”，也就是 “386BSD”，不過現在 386BSD 的程式碼只剩下極少數還留存在 FreeBSD 中。FreeBSD 已被廣泛地被世界各地的公司行號，ISP，研究人員，電腦專家，學生，以及家庭用戶所使用，用在工作，教育，以及娛樂上。

現今許多網路上的大型網站都是以 FreeBSD 作為其作業系統，例如：Yahoo!、Apache、Blue Mountain Arts、Pair Networks、Sony Japan、Netcraft、Weathernews、Supervalu、TELEHOUSE America、Sophos Anti-Virus 還有許多其他的網站。事實上，如果一家公司需要汲取大量的網際網路頻寬，它可能會執行 FreeBSD，因為其穩定性高。而 FreeBSD 的主要應用範圍可以在網

路伺服器方面，但是 FreeBSD 的應用並不侷限於此，它另外一方面的應用在於作為個人工作站的作業系統。專業的 Unix 工作站價格非常昂貴，但 FreeBSD 充分利用了個人電腦硬體廉價的優勢，以自己具備的優秀性能，使個人工作站擁有高性能的 Unix 工作站成為可能，且 FreeBSD 本身也提供了很好的機制用於簡化軟體的安裝和配置。

FreeBSD 的特點：

1. FreeBSD 是真正的 32 位元作業系統，不是任何 16 位元作業系統的升級版本。它是十分成熟的 BSD UNIX 向英特爾 386 體系的的處理器進行移植的結果，系統核心不包含任何 16 位元代碼，也不需要相容任何 16 位元軟體，從而提高了系統穩定性。且 FreeBSD 具有高效能核心架構、動態函式庫共用、絕佳的網路功能，比起其他商用 UNIX 系統毫不遜色。
2. FreeBSD 具有可調整的動態優先及搶佔式多任務能力。使多個應用程式能夠十分平滑的共用系統資源，即使在高負載下仍然能在不同任務間平緩切換，而不會發生由於個別任務獨佔系統資源，其他任務因此而發生停頓、鎖死現象，也絕不會造成整個系統鎖死。
3. FreeBSD 是多使用者作業系統，可以支援多個使用者同時使用 FreeBSD 系統，共用系統的磁碟、處理器等系統資源。每個使用者也可以同時啟動多個任務，使得工作效率更高。
4. FreeBSD 全面支援 TCP/IP 協定。FreeBSD 能夠十分方便的和其他支援的 TCP/IP 的系統集成在一起，用作 Internet／Intranet 伺服器，提供 NFS、ftp、email、www、路由和防火牆功能。
5. FreeBSD 中使用另一個著名的自由軟體 -- XFree86，來提供工業標準的 X 視窗系統(X11R6)，在 X 上可以執行多種圖形介面軟體，提供方便使用者使用圖形介面和應用軟體。
6. FreeBSD 的目標就是提供自由地可轉散發型、執行在熱門硬體上的作業系統。雖然系統安全性是一重大考量，FreeBSD 的主要目標是在人們最喜歡擁有的硬體上執行。目前最新版的 FreeBSD 6.0-RELEASE 支援

ALPHA、AMD64、I386、IA64、PC98、PPC、Sun 的 SPARC64。

7. FreeBSD 也經由它的 ports 收藏簡化了軟體管理。一般來說，為 UNIX 系統調適軟體所需要相當的專門知識。Ports 收藏藉由自動化和文件化的安裝、反安裝及組態設定數千個軟體套件程式，而大大地簡化了。數個其他的 BSD 作業系統已根據埠集合建立了它們自己的套件系統(packaging system)。
8. FreeBSD 能高效地滿足要求大量 RAM 的應用程式，又能最大效率地利用 RAM 來緩衝硬碟資料，提高讀、寫硬碟效率。
9. FreeBSD 支援各種語言和各種開發工具，如 C、C++、Fortran、perl、cvs、yacc...等，使得軟體開發和移植非常方便。
10. FreeBSD 開放原始碼，內附的眾多程式也是如此，大多數的軟體在 GNU 的規範下，你可免費的使用並修改它。
11. FreeBSD 具有動態共用連接庫的能力，使得應用程式能共用函式庫(類似 Windows 下的 DLL)，充分利用 RAM 和磁碟空間。

基於以上的特質，我們採用 FreeBSD 5.4-RELEASE 做為我們的作業系統，並額外安裝一些應用程式，使得操作上更加順手，使用也較為便利。

2.2.2 JDK-1.5.0p1

Java 簡介：

Java 原是 1991 年昇陽(Sun) 公司內部一項名為 Green 的發展計畫中，為了撰寫控制消費性電子產品軟體，所開發出來的小型程式語言系統，不過整項計畫並未獲得市場的肯定，因而沉寂了一段不短的時間。

但是沒有多久由於網際網路的蓬勃發展，誰也料想不到當初只是為了在不同平台系統下，執行相同軟體而開發的語言工具，卻意外地造成一種指標性趨勢。因此昇陽公司對 Green 計畫重新做的評估修正後，於 1995 年正式向外界發表名為「Java」的程式語言系統。

而 JDK 主要是負責編譯 Java 和執行 Java，因此如果想要執行 Java 程式，唯有安裝 JDK 才能夠順利地執行 Java 程式。

2.2.3 JSP

JSP 簡介：

JSP 是 Java Server Pages 的簡寫。JSP 技術能讓 Web 開發員和網頁設計員快速地開發容易維護的動態 Web 主頁。用 JSP 開發的 Web 應用是跨平臺的，即能在 Linux 下運行，也能在其他操作系統上運行。

JSP 技術使用 Java 編程語言編寫類 XML 的 tags 和 scriptlets，來封裝產生動態網頁的處理邏輯。網頁還能通過 tags 和 scriptlets 訪問存在服務端的資源(例如 JavaBeans)的應用邏輯。

JSP 將網頁邏輯與網頁設計和顯示分離，支持可重用的基組件的設計，使基 Web 的應用程式的開發變得迅速和容易。JSP 技術是 Servlet 技術的擴展。Servlet 是平臺無關的，100%純 Java 的 Java 服務端組件。

2.2.4 JavaBean

JavaBean 是一個可重複使用且跨越平臺的軟體元件，並且它可以在軟體開發工具如：Borland JBuilder、Oracle JDeveloper 或是 Sun ONE Studio 等等，以視覺化的方式來開發它。

它是利用 Java 語言撰寫而成，我們可以將它撰寫成各種 Java Class，將它放在元件 Applets 或 Java Class 中以便使用。JavaBean 在 JSP 上的應用，總共可分為 Page、Request、Session 與 Application 等四個部分。

一、**Page**：只對特定的網頁去存取Bean。

二、**Request**：只對同一個Request所提出的網頁去存取Bean。

三、**Session**：只對同一個使用者的Session所提出的網頁去存取Bean。

四、**Application**：可針對此系統中所有的使用者所提出的網頁去存取 Bean。

2.2.5 Apache-2.1.4

說到 Web Server，最有名氣的就是 Apache 和 IIS (Internet Information Services)了。但是比起穩定度及負荷度，就非 Apache 莫屬了。Apache 是 UNIX 系統中普遍使用的網頁伺服器軟體。目前網際網路中，有超過百分之五十的伺服器的使用 apache 來提供網頁瀏覽的服務。而我們也順便搭配 PHP 5 來使用 phpMyAdmin。

2.2.6 MySQL-4.1.10a

MySQL 是套著名的資料庫管理系統，它既免費功能又強大，不輸給一些知名的付費軟體，如 MS 的 SQL Server 和 Oracle，其效率也是深受好評，所以才能在眾多的資料庫管理系統中脫穎而出，也因此我們選擇它來當作我們後端的資料庫管理系統。

2.2.7 Jakarta-Tomcat-5.0.30

一套 Apache, 5.0.x branch 上的 Open-source Java web server，它是由 JavaSoft 和 Apache 開發團隊共同提出合作計畫(Apaches Jakarta Project) 下的產品。Tomcat 能支援 Servlet 2.4 和 JSP 2.0，並也是免費使用。搭配 JDK 和 Apache 之後就能執行 JSP 網頁，也是我們專題所採用的開發語言。

2.2.8 PHP5

為了能夠執行 phpMyAdmin 所以我們必須要有能支援 PHP 的環境，所以我們加灌了 mod_php5-5.0.3_2,1 模組，使得 Apache 能夠執行 PHP 的相關網頁語言程式，這在一般 UNIX 是蠻常見的組合 Apache + PHP + MySQL，尤其是在 FreeBSD 底下。

2.2.9 phpMyAdmin-2.6.1.3

phpMyAdmin 是套透過 PHP-scripts 語言所撰寫的工具，提供網頁化介面來管理 MySQL。透過它，我們可以輕易的建立/刪除資料庫和一些相關的資料庫操作等等，圖形化 GUI 的介面，省去了我們打指令的時間，也大大提升了對於 MySQL 資料庫的掌控，節省不少麻煩和瑣碎的事情，可說是一套相當好用的工具。

2.2.10 lynx-2.8.6d11

一個非圖形化，純文字介面的網頁瀏覽器，可以提供在沒有灌 X Window 之下能夠瀏覽 web，如果以速度為取向可以考慮 lynx 作為瀏覽器，因為基本上是沒有辦法讀取文字外的東西。

2.3 資料庫規劃

我們所探勘的資料庫是一個顧客購買商品的交易資料庫。在 2.3.1 節將描述我們的資料模型，2.3.2 節則是所用到資料表。

2.3.1 ER Model

2.3.1.1 ER Schema Diagram 描述與說明

賣場資料庫會紀錄一個賣場的顧客資訊、交易記錄、商品項目、分類類型等。

1. 我們會記錄所有在賣場中有購物行為的顧客(CUSTOMER)的資訊，包含該顧客的姓名、性別，以及唯一的識別號碼。
2. 對於賣場中的每樣商品項目(ITEM)，我們會記錄商品編號、商品名稱、價格，以及該商品隸屬的某個唯一分類類型。
3. 我們會記錄賣場的商品類型(種類)(CATEGORY)，每個類型有獨特的編號與名稱，同時我們也希望記錄該種類型的商品是擺放於賣場中的何種區域。
4. 我們欲追蹤、記錄賣場的所有交易紀錄(TRANSACTION)，以便將來可做統計與行銷策略之用途。我們保存每筆交易的行使顧客、發生時間、購買商品項目，並且有唯一獨特的序號。每一筆交易行為可能包含一種到好幾種的商品。

2.3.1.2 ER Schema Diagram

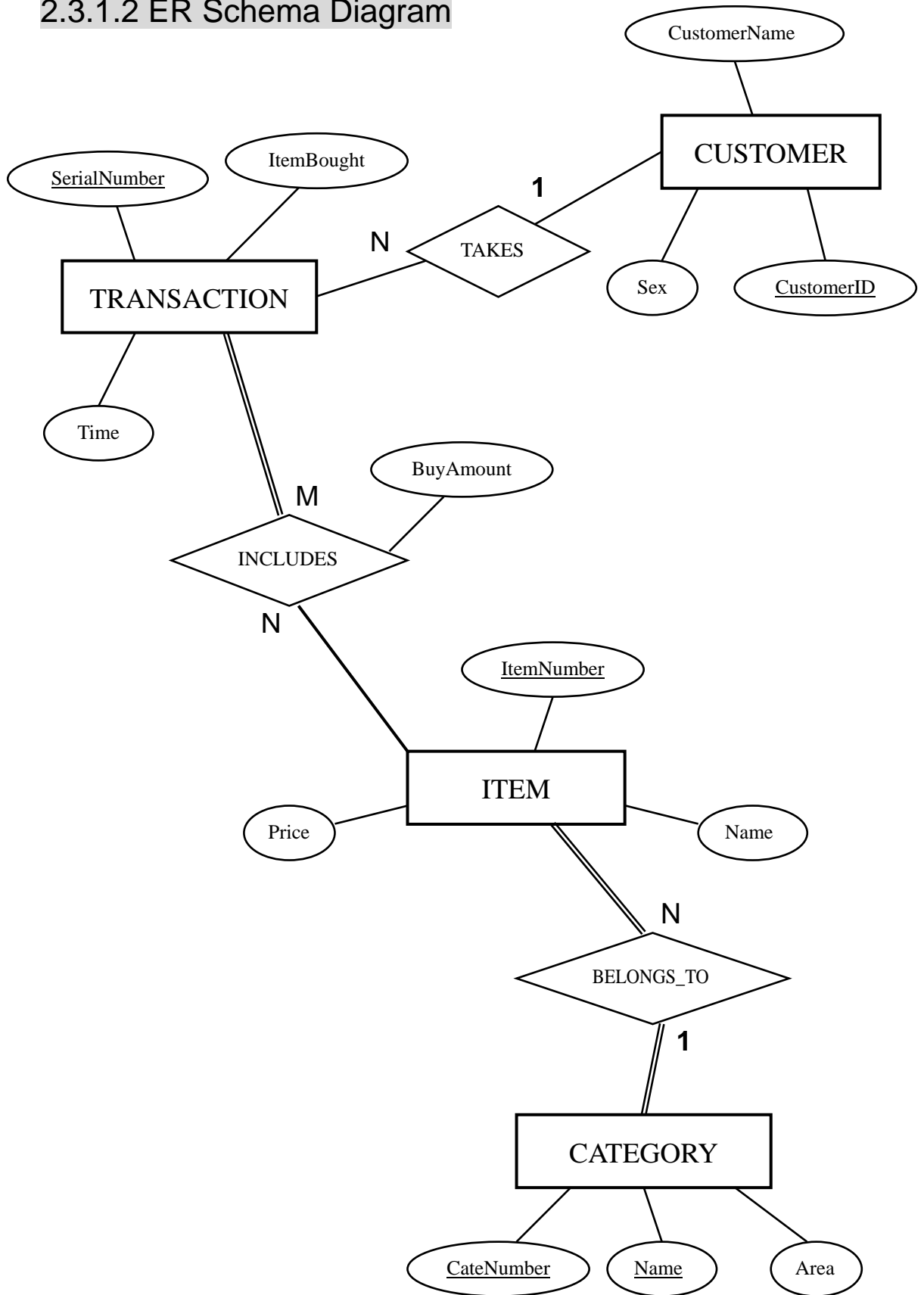


圖 1 ER Model Diagram

2.3.2 Data Mining Table

以下是我們所用到的資料表架構(Schema)，說明請見 2.3.2.2 節。

2.3.2.1 Relational Schema (Table)

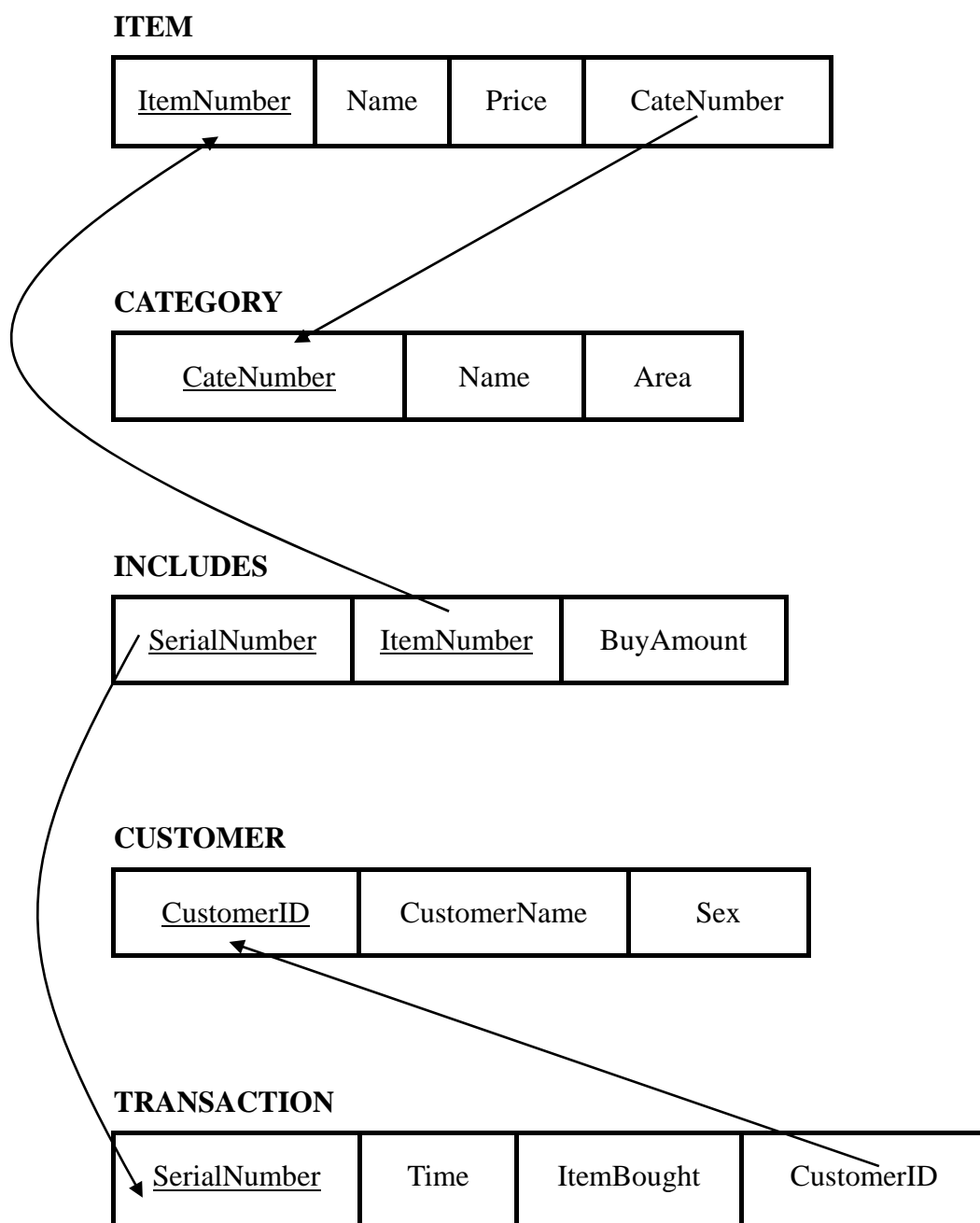


圖 2 Relational Schema

2.3.2.2 Relational Schema 描述與說明

Customer

屬性名稱	資料型態	說明與描述	範例
ID	INT	唯一識別每個不同顧客的整數編號	999
Name	VARCHAR(20)	顧客的姓名，將中文名字以英文拼音形式表示，順序為先姓後名	LI CHAO WEI
Sex	CHAR	顧客的性別，以'M'表示男性，以'F'表示女性	F

Includes

屬性名稱	資料型態	說明與描述	範例
SerialNumber	CHAR(8)	唯一識別每筆交易的序號，同時允許英文字母與數字，其中英文字母僅可能出現在前兩位	NB004869
ItemNumber	INT	代表每項商品的獨特整數號碼，從 1 號開始，範圍限制視賣場中的商品種類而定	777
BuyAmount	INT	一筆交易中購買某項特定商品的數量	10

Category

屬性名稱	資料型態	說明與描述	範例
ID	INT	商品種類的識別編號，每個號碼唯一表示某種商品類型	15
Name	VARCHAR(25)	商品種類的名稱	CRACKER
Area	INT	賣場區域的編號，唯一識別賣場中的每個商品放置區域	51

Item

屬性名稱	資料型態	說明與描述	範例
ID	INT	代表每項商品的獨特整數號碼，從 1 號開始，範圍限制視賣場中的商品種類數目而定	777
Name	VARCHAR(50)	商品的完整產品名稱，包含階層順序的從屬關係，例如廠商、品牌、商標、級別等	WEICHAUN LIN-FENG- YING 100% MILK
Price	DECIMAL(10,2)	商品的價格，最大精確度到小數點下兩位，但通常是整數	129.00
CateID	INT	商品種類的識別編號，每個號碼唯一表示某種商品類型	18

Transaction

屬性名稱	資料型態	說明與描述	範例
ID	CHAR(8)	唯一識別每筆交易的序號，同時允許英文字母與數字，其中英文字母僅可能出現在前兩位	AP004869
Time	DATE	交易發生的時間，包含年份、月份與日期，這裡假設每位顧客每天最多進行一筆交易	2005-12-12
CustomerID	INT	唯一識別每個不同顧客的整數編號	1001
ItemBought	VARCHAR(50)	顧客在交易中購買的所有商品列表，商品以其編號表示，並以逗號區隔每一項商品	10, 20, 35, 40

第三章 相關理論與技術研究

3.1 探勘關聯規則

3.1.1 關聯規則的問題分解

- 目的是找出所有(交易支持度)超過最小支持度值的項目集合(itemset)
 - 說明：一個項目集合的**支持度**，就是資料庫中包含此項目集合的交易之數目。而超過最小支持度的項目集合稱為頻繁(large)項目集合
- 接下來使用這些頻繁項目集合來產生所希望的規則
 - 假設：(ABCD) 和 (AB) 是頻繁項目集合，我們可以決定規則 $AB \rightarrow CD$ 是否成立，經由計算其信賴度(confidence)的方式
 - **信賴度** = $\text{support}(\text{ABCD}) / \text{support}(\text{AB})$ ，如果求得的信賴度值在最小信賴度以上，那麼這個規則就可以成立

3.1.2 基本定義與符號

表 1 符號說明

K-項目集合	含有k個項目的項目集合
L_k	頻繁k-項目集合的組合(擁有最小支持度)，集合中的每個成員有兩個欄位，分別存放: (1)項目集合 (2) 支持度計數
C_k	候選k-項目集合的組合(可能是頻繁項目集合)，集合中的每個成員也有兩個欄位，存放著: (1)項目集合 (2)支持度計數
\overline{C}_k	候選k-項目集合的組合，此外還包含所產生的交易紀錄之TID，跟這些候選項目集合關聯在一起。

- 一個項目集合的大小(size)就是其中所含的項目(item)之數量
- 一個大小為 k 的項目集合稱為 **k-項目集合**(k-itemset)
- 使用標記法 $c[1], c[2], c[3], \dots, c[k]$ 來表示一個 k-項目集合的每個組成元素

3.1.3 Apriori 探勘演算法

演算法的第一階段，簡單地計算每個項目的出現次數，用以決定長度為 1 的頻繁項目(L_1)。在每個後繼的階段，分為兩個步驟：

- 首先，用先前一階段求得的 L_{k-1} 來產生候選項目集合 C_k （產生候選人）
- 接著，對資料庫進行掃描並計算每個 C_k 的支持度（檢查是否夠頻繁）

```
1)  $L_1 = \{\text{large 1-itemsets}\};$ 
2) for (  $k = 2; L_{k-1} \neq \emptyset; k++$  ) do begin
3)    $C_k = \text{apriori-gen}(L_{k-1});$  // New candidates – see Section 2.1.1
4)   forall transactions  $t \in \mathcal{D}$  do begin
5)      $C_t = \text{subset}(C_k, t);$  // Candidates contained in  $t$  – see Section 2.1.2
6)     forall candidates  $c \in C_t$  do
7)        $c.\text{count}++;$ 
8)   end
9)    $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$ 
10) end
11)  $\text{Answer} = \bigcup_k L_k;$ 
```

圖 3 Apriori 演算法 [8]

Apriori 演算法範例：

假設資料庫中有 9 筆交易(表 2)，也就是 $|D|=9$ 。Apriori 假設交易中的項目按字典次序存放。圖 5 解釋 Apriori 演算法尋找 D 中的頻繁項目集合。

表 2 資料庫內交易資料

TID	List of item_IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

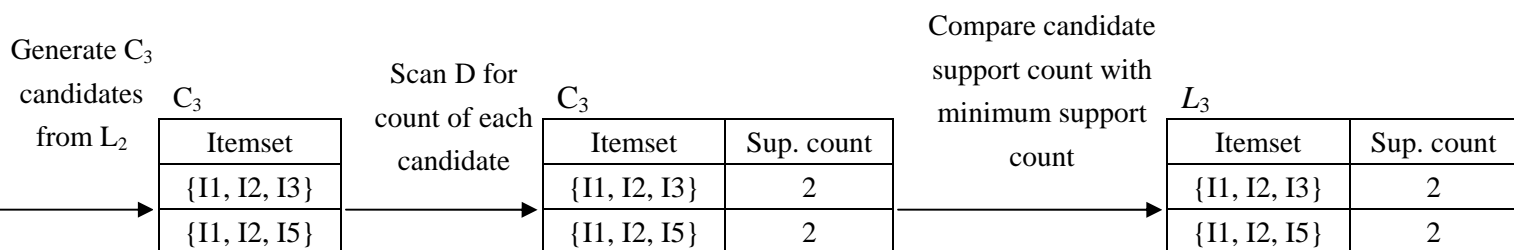
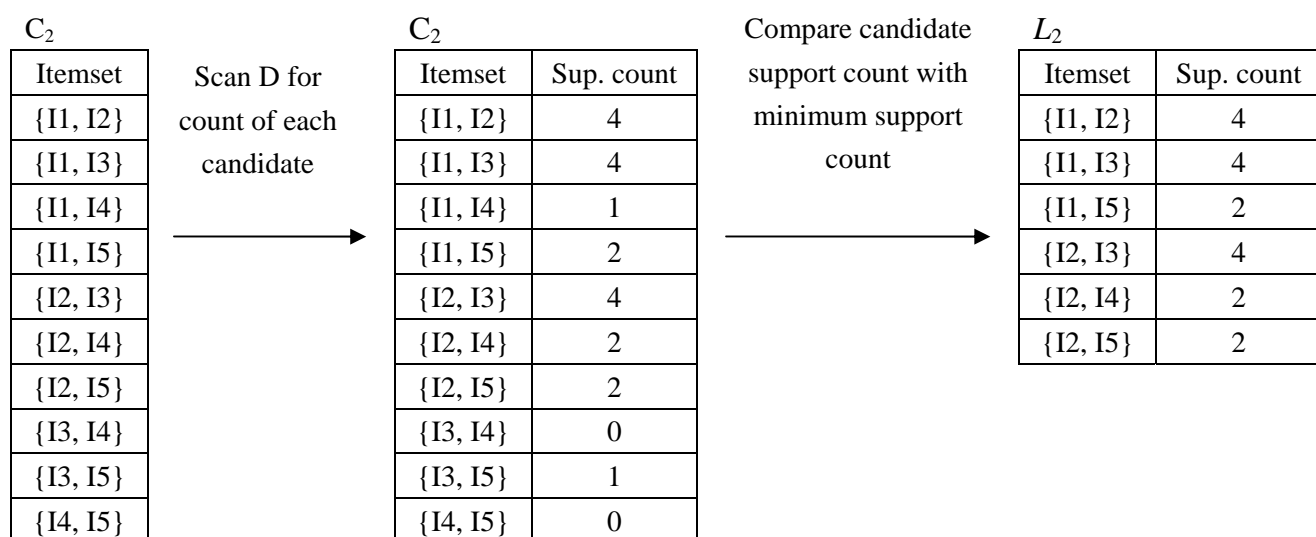
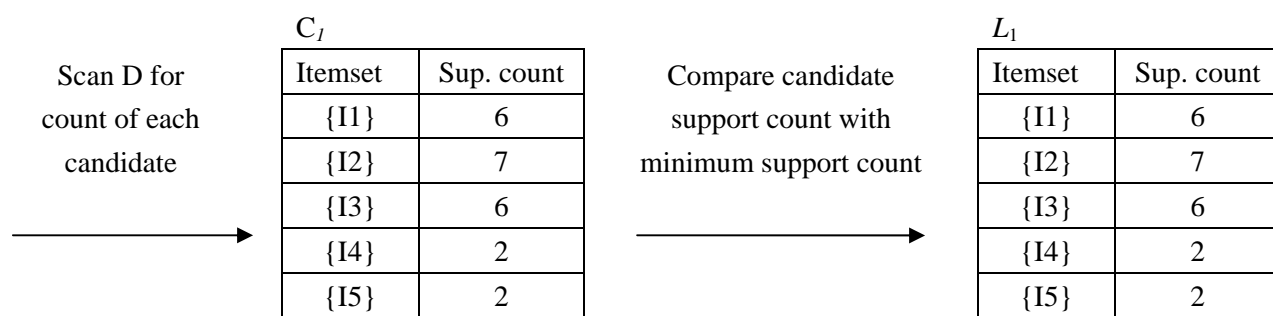


圖 4 Apriori 演算法實例

解說：

- (1) C_1 是掃描資料庫內所有的交易，並計算每個項目出現的次數計數
- (2) 假設最小交易支援度為 2 ($\min_sup = 2/9 = 22\%$)。並確定它是具有最小支持度 (support) 的 C_1 中的候選項目集合組成。
- (3) 使用 $L_1 \bowtie L_1$ 產生 C_2 ，並掃描 D 中交易，以計算 C_2 中每個候選項目集合的支援計數。
- (4) 產生 L_2 ，並確定它是具有最小支援度 (support) 的 C_2 中的候選項目集合組成。
- (5) 令 $C_3 = \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}$ 。根據 Apriori 性質(如下表)，頻繁項目集合的所有子集合必須是頻繁的，我們可以確定後 4 個候選不可能是頻繁的。因此，我們把它們從 C_3 刪除，這樣，在此掃描 D 確定 L_3 時就不必再求它們的計數值。注意的是，Apriori 演算法使用逐層搜索技術，給定 k -項目集合，我們只需要檢查它們的 $(k-1)$ -子集是否頻繁。

連接(Join)：

$$\begin{aligned} C_3 &= L_2 \bowtie L_2 = \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\} \\ &\quad \bowtie \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\} \\ &= \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\} \end{aligned}$$

修剪(Prune)：

$\{I1, I2, I3\}$ 的 2-項子集是 $\{I1, I2\}$, $\{I1, I3\}$ 和 $\{I2, I3\}$ 。

$\{I1, I2, I3\}$ 的所有 2-項子集都是 L_2 的元素。因此，保留 $\{I1, I2, I3\}$ 在 C_3 中。

$\{I2, I3, I5\}$ 的 2-項子集是 $\{I2, I3\}$, $\{I2, I5\}$ 和 $\{I3, I5\}$ 。

$\{I3, I5\}$ 不是 L_2 的元素，因而不是頻繁的項目。因此，由 C_3 中刪除 $\{I2, I3, I5\}$ 。

修剪後 $C_3 = \{\{I1, I2, I3\}, \{I1, I2, I5\}\}$ 。

圖 5 候選項目產生方式例

3.1.4 功能函數 apriori-gen

apriori-gen 函數以頻繁項目集合 L_{k-1} 當作輸入，執行後將會輸出長度為 k 的候選項目集合 C_k 。

第一步驟，從 L_{k-1} 中尋找所有「前($k-2$)項都相同」的兩個項目集合，將它們進行結合並加入 C_k 的集合中。

```
insert into  $C_k$ 
select  $p.item_1, p.item_2, \dots, p.item_{k-1}, q.item_{k-1}$ 
from  $L_{k-1} p, L_{k-1} q$ 
where  $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1};$ 
```

圖 6 Join 的 SQL 指令

第二步驟，對所有 C_k 的成員項目集合 c ，凡是 c 中有某些長度為 $(k-1)$ 的子集合不在 L_{k-1} 當中，則將 c 從 C_k 中刪除。

```
forall itemsets  $c \in C_k$  do
  forall  $(k-1)$ -subsets  $s$  of  $c$  do
    if  $(s \notin L_{k-1})$  then
      delete  $c$  from  $C_k;$ 
```

圖 7 刪除不合條件的候選項目

3.1.5 探索關聯規則

對於所有頻繁項目集合 la ，我們找出 la 的所有**非空子集合**。對每個這樣的子集合 a ，只要 $\text{support}(la)$ 除以 $\text{support}(a)$ 的商數超過最小信賴度(min-conf.)，我們就可以得到一個關聯規則的形式： $a \rightarrow (la - a)$

```
// Simple Algorithm
forall large itemsets  $l_k, k \geq 2$  do
    call genrules( $l_k, l_k$ );

// The genrules generates all valid rules  $\tilde{a} \Rightarrow (l_k - \tilde{a})$ , for all  $\tilde{a} \subset a_m$ 
procedure genrules( $l_k$ : large  $k$ -itemset,  $a_m$ : large  $m$ -itemset)
1)  $A = \{(m-1)\text{-itemsets } a_{m-1} \mid a_{m-1} \subset a_m\}$ ;
2) forall  $a_{m-1} \in A$  do begin
3)    $conf = \text{support}(l_k) / \text{support}(a_{m-1})$ ;
4)   if ( $conf \geq minconf$ ) then begin
5)     output the rule  $a_{m-1} \Rightarrow (l_k - a_{m-1})$ , with confidence =  $conf$  and support =  $\text{support}(l_k)$ ;
6)     if ( $m - 1 > 1$ ) then
7)       call genrules( $l_k, a_{m-1}$ ); // to generate rules with subsets of  $a_{m-1}$  as the antecedents
8)   end
9) end
```

圖 8 產生規則的方法

- 上述的程序可以使用一個遞迴的**深度優先(depth-first)**的形式來增進其效能，產生一個頻繁項目集合的子集合
 - 例如，給定一個項目集合 ABCD，我們首先考量子集合 ABC，接著考量子集合 AB，以此類推
- 如果頻繁項目集合 la 的某個子集合 a 沒有產生規則（也就是信賴度不夠），那麼 a 的所有子集合都不需要再考慮
 - 例如，如果 $ABC \rightarrow D$ 這個規則沒有足夠的信賴度，那麼我們便沒有必要再檢查 $AB \rightarrow CD$ 、 $AC \rightarrow BD$ 等規則是否會成立
- **理由**：一個項目集合 a 的所有子集合 a' ，其支持度(support)必大於或等於 a ，因此 $a' \rightarrow (la - a')$ 的信賴度只會變得更小，當然更不可能成立

3.2 探勘循序樣式

3.2.1 循序樣式的基本簡介

- 資料探勘，又被稱為「資料庫知識發現」，被定義為在大型資料庫中有效地探索有趣規則的研究技術
- 「循序樣式探勘」是資料探勘領域中的一個新興問題，其中輸入的資料是一組序列，稱為**資料序列**(data-sequence)
- 每個資料序列由一連串**交易**(transaction)所組成，而每筆交易則是由一群**項目**(item)所組成，每筆交易通常伴隨一個交易時間(transaction-time)
- 一個循序樣式是由一連串的**項目集合**(sets of items)所組成，可以是一個也可以是數個，研究之目的是要找尋出所有滿足最小支持度的樣式
- 循序樣式的**支持度**(support)，就是(資料庫中)包含有該樣式的資料序列的數目，或者是佔全部資料序列的百分比

3.2.2 研究問題敘述

- **項目集合**(itemset)是一個非空的項目(item)之集合
- **序列**(sequence)是一個排序過的項目集合的列表
- 序列 s 的表示式為 $\langle (s_1) (s_2) (s_3) \cdots (s_n) \rangle$ ，其中每個 s_j 是一個項目集合，也稱為該序列的一個**元素**(element)
- 序列元素的表示式為 (x_1, x_2, \cdots, x_m) ，其中每個 x_j 是一個**項目**(item)
- 每個項目只能夠在同一個序列元素中出現一次，然而可以在一個序列的不同元素中出現許多次
- 一個項目集合可被視為是一個只有**單一元素**的序列
- 一個序列的支持度，定義為在所有資料序列中，包含有該樣式的比例

3.2.3 探勘演算法 GSP

- 演算法分為多次操作階段對資料進行處理
- 第一個階段決定每一個項目(item)的支持度
 - 取得所有頻繁的項目，每個項目產生一個 1-元素-頻繁序列
- 後繼的每個階段使用前次階段的頻繁序列做為**種子集合**
 - 種子集合用來產生新的潛在頻繁序列，稱為**候選序列**
 - 每個產生的候選序列比種子序列多一個項目
 - 讀取資料庫以決定哪些候選序列是頻繁的
 - 最後將找到的頻繁序列(循序樣式)做為下一個階段的種子集合
- 需要詳細指明的兩個關鍵細節：
 - 候選序列產生：每個階段之前的候選序列如何產生的？
 - 計算候選序列：每個候選序列的支持度是如何決定的？

3.2.3.1 候選序列的產生

- 一個有 k 個項目的序列稱為一個“**k-序列**”(k-sequence)
- 給定 L_{k-1} ，我們希望產生所有 L_k 集合的超集合，即 C_k
- 定義：給定一個序列 $s = \langle s_1 s_2 s_3 \cdots s_n \rangle$ 以及一個子序列 c ， c 被稱為 s 的**連續子序列(contiguous)**，如果 c 滿足底下任一條件：
 - c 是從 s 當中刪去 s_1 或 s_n 的一個項目而得來
 - c 是從 s 當中，刪去某個至少 2 個項目的元素 s_i 的 1 個項目而得來
 - c 是 c' 的連續子序列，而 c' 是 s 的連續子序列 (層遞關係)
- 舉例： $s = \langle (1, 2) (3, 4) (5) (6) \rangle$ ，則 $\langle (2) (3, 4) (5) \rangle$ 、 $\langle (1, 2) (3) (5) (6) \rangle$ 、 $\langle (3) (5) \rangle$ 都是 s 的連續子序列。
 - 然而 $\langle (1, 2) (3, 4) (6) \rangle$ 和 $\langle (1) (5) (6) \rangle$ 則不是連續子序列
- 若一個資料序列包含序列 s ，則也會包含 s 的任何連續子序列
- 候選序列分為兩個階段產生：
 - **連結階段(Join)** 與 **修剪階段(Prune)**

3.2.3.2 檢查資料序列是否包含一特定序列

- **包含性檢驗(Contains test)**：檢查資料序列 d 是否包含一候選序列 s ，演算法在以下兩個階段之間交替切換：
- **前向階段(Forward)**：找出連續的兩個元素，只要剛找到的元素的結束時間，和前一個元素的開始時間的時間差，小於最大時間間隔(max-gap)
- **反向階段(Backward)**：演算法逆向進行並「拉拔」(pull up)之前一個元素。如果 s_i 是目前元素且其結束時間為 t ，則找出在時間點 $t - (\text{max-gap})$ 之後，包含有 s_{i-1} 的第一組交易
- 此程序會反覆進行，在前向與反向兩階段之間不斷切換，直到所有的元素被找到為止，否則表示 d 並沒有包含 s

3.2.3.3 包含性檢驗範例

Transaction-Time	Items	Item	Times
10	1, 2	1	→ 10 → 50 → NULL
25	4, 6	2	→ 10 → 50 → 90 → NULL
45	3	3	→ 45 → 65 → NULL
50	1, 2	4	→ 25 → 90 → NULL
65	3	5	→ NULL
90	2, 4	6	→ 25 → 95 → NULL
95	6	7	→ NULL

圖 9 包含性檢驗範例 [9]

最大間距(Max Gap)=30，最小間距(Min Gap)=5，合併時間總和(Sliding

Window)=0，候選序列為 $\langle (1, 2) (3) (4) \rangle$

- 首先找到(1, 2)在時間點 10，接著找到(3)在時間點 45
- 由於兩元素的間距過大(35 天)，切換反向階段並「拉拔」(1, 2)
- 在時間點 15 ($t - \text{max-gap} = 45 - 30$) 之後搜尋(1, 2)的首次出現
- 找到(1, 2)在時間點 50，返回前向階段並找到(3)在時間點 65
- (3)和(1, 2)之間的最大間距限制滿足，繼續找到(4)在時間點 90
- (4)和(3)之間的最大間距限制滿足，檢驗完成！（有包含此候選序列）

3.3 時間限制條件與樣式增長

3.3.1 時間限制型循序樣式

針對一個資料序列 $ds \langle e_1 e_2' \dots e_n' \rangle$ 以及一個序列 $s = \langle e_1 e_2 \dots e_w \rangle$ ，如果存在有一群整數 $1 \leq u_1 < l_2 \leq u_2 < \dots < l_w \leq u_w \leq n$ 而且符合下列 5 種條件，我們便稱 ds 包含序列 s ：

(1) $e_i \subseteq (e_{l_i}' \cup \dots \cup e_{u_i}')$ and $t_{u_i} - t_{l_i} \leq swin, 1 \leq i \leq w$ // 合併時間總和

➤ex: $swin=2$, 則 $\langle_3(c)_5(a, f) \rangle$ 包含樣式 $\langle(a, c) \rangle$, $5-3 \leq 2$

(2) $t_{u_i} - t_{l_{i-1}} \leq maxgap, 2 \leq i \leq w$ // 最大間距

➤ex: $maxgap=15$, 則 $\langle_3(c)_5(a, f)_{18}(b) \rangle$ 包含樣式 $\langle(a, c)(b) \rangle$, $18-3 \leq 15$

(3) $t_{l_i} - t_{u_{i-1}} \geq mingap, 2 \leq i \leq w$ // 最小間距

➤ex: $mingap=3$, 則 $\langle_3(c)_5(a, f)_{18}(b) \rangle$ 包含樣式 $\langle(a, c)(b) \rangle$, $18-5 \geq 3$

(4) $t_{u_w} - t_{l_1} \leq duration$ // 總時間

➤ex: $duration=25$, 則 $\langle_3(c)_5(a, f)_{18}(b) \rangle$ 包含樣式 $\langle(a, c)(b) \rangle$, $18-3 \geq 25$

(5) 確切間隔 (最大間隔=最小間隔), $t_{l_i} - t_{u_{i-1}} = exact\ gap, 2 \leq i \leq w$

3.3.2 探勘演算法比較

表 3 各種探勘演算法比較 [7]

主要架構	Algorithm	min. gap	max. gap	Exact gap	sliding window	duration	提出年份
Apriori-like	GSP	✓	✓		✓		1996
	PSP	✓	✓		✓		1998
	cSPADE	✓	✓	✓		✓	2000
	CCSM	✓	✓	✓			2004
Pattern-growth	prefix-growth	✓	✓	✓		✓	2002
	DELISP	✓	✓		✓		2002

- GSP (Generalized Sequential Patterns) (廣義化循序樣式)
- PSP (Prefix tree for Sequential Patterns) (前序樹型式之循序樣式)
- cSPADE (constrained Sequential Pattern Discovery using Equivalence classes)
- CCSM (Cache-based Constrained Sequence Miner) (快取式限制型序列探勘)
- prefix-growth (Mining sequential patterns with prefix-monotone constraints)
- DELISP (DELimited Sequential Pattern) (限制式之循序樣式)

3.3.3 相關術語說明

- **頻繁項目**：資料庫中的項目 x 如果滿足 $x.\text{sup} \geq \text{minsup}$ (最小支持度)，則我們稱其為「頻繁」的項目
- **第一類樣式 (Type-1)、第二類樣式 (Type-2)、Stem、Prefix**：

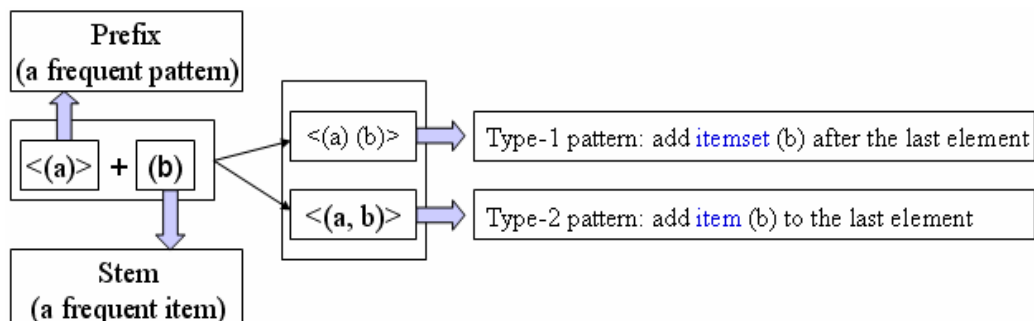


圖 10 Two pattern-growth forms [7]

3.3.4 滿足時間限制條件的支持度定義

- 一個序列 s 如果滿足 $s.\text{sup} \geq \text{minsup}$ ，則 s 是一個符合時間限制條件的循序樣式，其中 $s.\text{sup}$ 是序列 s 在資料庫中的支持度，而 minsup 是使用者指定的最小支持度門檻
- s 的支援度是將包含 s 的資料序列數目除以 $|\text{DB}|$ 所得到的商數計算而來
- 一個資料序列 $ds = \text{sid} / \langle {}_{i1}e_1' \quad {}_{i2}e_2' \quad {}_{i3}e_3' \quad \dots \quad {}_{im-1}e_{m-1}' \quad {}_me_n' \rangle$ 被認為**包含有一個**序列 $s = \langle e_1 \ e_2 \ e_3 \ \dots \ e_w \rangle$ ，如果存在一組整數 $l1, u1, l2, u2, l3, u3, \dots, lw, uw$ 而且 $1 \leq l1 \leq u1 < l2 \leq u2 < l3 \leq u3 < \dots < lw \leq uw \leq n$ ，如此以致於下列四個條件成立：
 - (1) $e_i \subseteq (e_{l_i}' \cup \dots \cup e_{u_i}')$, $1 \leq i \leq w$, // 傳統包含性定義
 - (2) $t_{u_i} - t_{l_i} \leq \text{swin}$, $1 \leq i \leq w$, // 合併時間總和
 - (3) $t_{u_i} - t_{l_{i-1}} \leq \text{maxgap}$, $2 \leq i \leq w$ and // 最大時間間距
 - (4) $t_{l_i} - t_{u_{i-1}} > \text{mingap}$, $2 \leq i \leq w$. // 最小時間間距
- 假設 t_j 、 maxgap 、 mingap 和 swin 皆為正整數， mingap 和 swin 的數值可以為 0，而且 $\text{mingap} < \text{maxgap}$

- 序列 $s = \langle e_1 e_2 e_3 \dots e_w \rangle$ 被認為**包含在**資料序列 $ds = \langle e_1' e_2' e_3' \dots e_n' \rangle$ 當中，如果 e_i 中的所有項目，都可以在合併 e_{li}' 和 e_{ui}' 之間的所有元素而形成的元素中被發現，其中 $1 \leq i \leq w$ ，而且 $swin$ 、 $maxgap$ 、 $mingap$ 等限制條件都能滿足

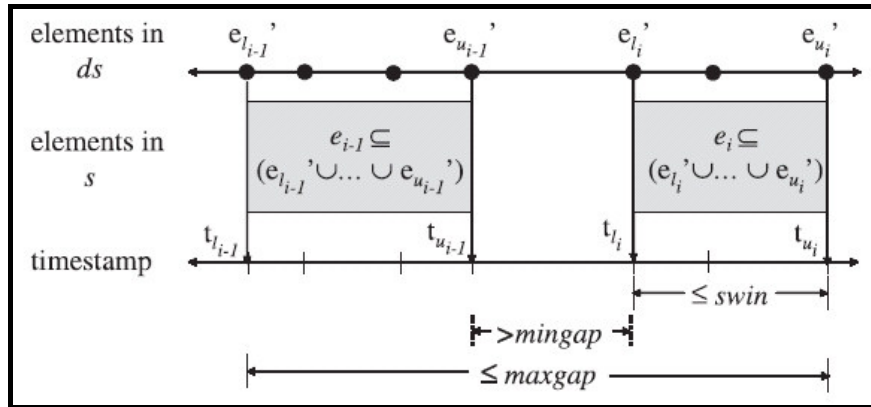



圖 11 序列包含性關係的範例 [14]

3.4 相關軟體操作

3.4.1 ARTool

ARTool 是用 Java 這個程式語言所撰寫的工具程式，把資料探勘的演算法用 Java 實作出來，利用 java 的 GUI 介面方便使用者以滑鼠點選使用。C 語言實作出來的資料探勘演算法，要自行輸入指令和參數才能顯示出結果，和利用 Java 實作的相比較，確實是 Java 所撰寫的 GUI 介面比較方便。

以下是一個測試範例，包含 9 筆交易並詳細記錄每筆交易的購買項目，購買商品的代號依照 ARtool 所要求的格式輸入進去，然後再存成 input.asc 檔案。



```
input.asc - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
1 I1
2 I2
3 I3
4 I4
5 I5

BEGIN_DATA
1 2 5
2 4
2 3
1 2 4
1 3
2 3
1 3
1 2 3 5
1 2 3
END_DATA
```

圖 12 ARTool 測試資料

點選到 標籤 頻繁項目集合(Frequent Itemsets)，然後選擇演算法為：「Apriori」，最小支持度值(Minimum support)設定為 0.2，之後按下開始按鈕，稍待片刻就可以執行出以下畫面中的結果。

Itemset	Support
I1	0.6666666666666666
I2	0.7777777777777778
I3	0.6666666666666666
I4	0.2222222222222222
I5	0.2222222222222222
I1, I2	0.4444444444444444
I1, I3	0.4444444444444444
I1, I5	0.2222222222222222
I2, I3	0.4444444444444444
I2, I4	0.2222222222222222
I2, I5	0.2222222222222222
I1, I2, I3	0.2222222222222222
I1, I2, I5	0.2222222222222222

Have fun using ARTool!

Opened database: C:\ARTool\input.db
Cache file is: C:\ARTool\input.cache

Executing algorithm laur.dm.ar.Apriori on database C:\ARTool\input.db for minimum support 0.2... done!
Time elapsed (ms): 125
4 passes were performed over the database

Reading cache contents... done!
13 itemsets were found
Displaying results... done!

圖 13 ARTool 執行 Apriori 結果

ARTool 還有其他更進階的功能，使用者如果點選 關聯規則(Association Rules) 標籤，在 basis 頁面可設定 最小信賴度值(Minimum confidence)。Advanced 頁面可以設定 Antecedent must contain、Consequent must contain、Items to ignore、Maximum antecedent size、Minimum consequent size 等參數值，使用者可以針對特殊需求，利用此工具程式探勘所希望的資訊與規則。

第四章 每學期工作進度與重點

4.1 三年級上學期

4.1.1 相關環境建立

我們首先組裝一台電腦當作研究用的主機兼伺服器。在硬體配置完成後，從安裝作業系統 FreeBSD 開始，接著陸續安裝網頁伺服器 Apache、Java 開發工具 JDK、資料庫管理系統 MySQL、支援模組 PHP5 等必備程式工具。針對專題研究可能產生的需求，我們也架設電子郵件系統 OpenWebMail，並且建立一個組員專屬的留言板（討論區）。

4.1.2 研讀書籍與文獻

針對老師所推薦的書籍進行學習，並且研讀教授提供的數篇資料探勘知名論文。由於學校並未開設此門課程，所以我們等於是從零開始學習這門技術；一開始當然沒有什麼概念，連「信賴度」與「支持度」的意思都搞不清楚。然而在不斷的碰觸與複習，並且使用市場購物籃的案例來幫助學習之後，總算逐漸建立較為清楚、正確的概念，即使是艱深的研究論文也變得容易理解了。

4.1.3 定期開會討論

我們小組每一週或每隔兩週會定期與教授開會，針對研究過程中的困難進行提出並請求老師給予指導，此外各人也會針對書籍與文獻的讀後感進行簡報介紹。在大家的交流討論與教授的指點迷津下，無論是在觀念上與行動上都越來越能夠掌握正確的方向。

4.1.4 網頁系統架設

初步建立好展示用的網頁式使用者介面，並導入最基本的關聯規則探勘功能。這個學期尚未觸及系統核心的程式碼實作，重點在於基本觀念的建立，以及測試資料的產生。（當然，因為課業忙碌而無法撥出時間研究專題也是一項原因）

4.2 三年級下學期

4.2.1 資料庫與資料表建立

產生測試用的輸入資料，包含一千筆交易到一萬筆交易的數個測試檔供選擇（尚且先不考慮每筆交易的時間）。另一方面為了模擬賣場情境，我們替商品項目建立商品名稱，並且與輸入資料進行結合，提供更人性化的顯示內容。

4.2.2 報名國科會專題研究計畫

在指導老師的建議之下，我們報名欲參加國科會的大專生專題研究計畫，然而最後並未獲選。究其原因，我們在當初報名的那個時期，對於資料探勘的觀念尚未有全面性的清楚認識，特別是關於循序樣式的部分，這使得研究計畫書沒能強調出自我特色與價值，以致於未獲評審委員的青睞。雖然如此，我們並沒有因此氣餒，因為我們很明白自己的研究主題是充滿挑戰性與前瞻性的技術，而且此結果反而更加深我們將專題研究做得盡善盡美的決心。

4.2.3 關聯規則演算法實作

針對關聯規則最著名的 Apriori 演算法進行實作，使用程式語言為 Java 與 JSP。而我們認為比較困難的部份除了 HashTree 的資料結構之外，就是 JSP 和 Java 的整合部分。由於 JSP 和 Java 的程式碼有時是由兩位不同組員分別撰寫，所以往往會出現各自測試都沒問題，但整合之後卻出現錯誤的情況… 外加上 JSP 並沒有一套有效的 IDE 軟體來協助開發，所以在除錯方面可謂困難重重，我們只能憑著 Tomcat 的幾行模糊錯誤訊息來找出問題所在。所幸最後終能順利完成實作！

4.2.4 循序樣式概念初探

在關聯規則的程式碼大致完工後，我們接著將其整合至網頁式系統當中。隨著各項功能的運作測試，以及必要的錯誤修正後，我們也設法改善網頁介面，使其更加美觀並更加人性化。在學期末我們開始研究循序樣式的觀念，並嘗試導入時間限制條件的探勘功能至系統中，不過初探的結果是遭遇許多的挫折。

4.3 四年級上學期

4.3.1 系友專訪活動

基於系上規定，我們小組在暑假期間至桃園南崁的長榮航空公司進行系友專訪，訪談對象為擔任電算本部協理的方國憲學長。透過此行我們學習到資料倉儲的基本觀念與知識，瞭解其與資料探勘之間的密切關係，並認識長榮航空在開發資料倉儲技術的傲人成就。在訪談過程中，我們小組和方學長（以及數位長榮航空工程師），雙方彼此分享經驗和展示成果，此外也針對學業以外的其他方向（例如就業、職場、工作等）進行交流。雖然在程式碼撰寫上並沒有直接幫助，但此專訪活動對我們相當具有意義，除了增廣見聞外也獲得了許多寶貴的訊息。

4.3.2 循序樣式功能實作

針對我們在循序樣式的觀念上的疑惑，指導老師和研究生學長數次為我們進行解釋和說明。像是加入「時間間距」(Max./Min. Gap) 和「合併時間總和」(滑動時間窗) (Sliding Window)等參數後，所需要考慮的兩種型態的樣式增長方式，除了相關文獻上的敘述外，學長也很熱心地對我們詳細解說。我們最後是參考研究生學長的演算法，使用「樣式增長」(Pattern-growth)架構方法，順利實作出具備各式時間限制條件的循序樣式探勘功能。

4.3.3 專題發表相關準備

在實作時間限制條件考量，加入循序樣式探勘功能後，我們的網頁式系統可謂接近完成，僅剩下參酌教授的建議，進行細部修改與微調的善後工作，同時盡量統一專有名詞的翻譯用字。配合系上公布的相關時程表，我們除了著手於正式（最終版）書面報告的製作之外，也開始針對發表時的口頭報告進行準備，並分配各位組員所負責的內容。

第五章 專題研究實作成果

5.1 系統流程

本系統可讓使用者登錄網頁，利用GUI介面來對網頁產生指令，這些需求會被送到遠端伺服器中，經由 JavaBean 的技術來對需求做搜尋或相關處理，最後再將結果傳回網頁並展示於使用者，可以循環式的執行此動作，參考圖14。

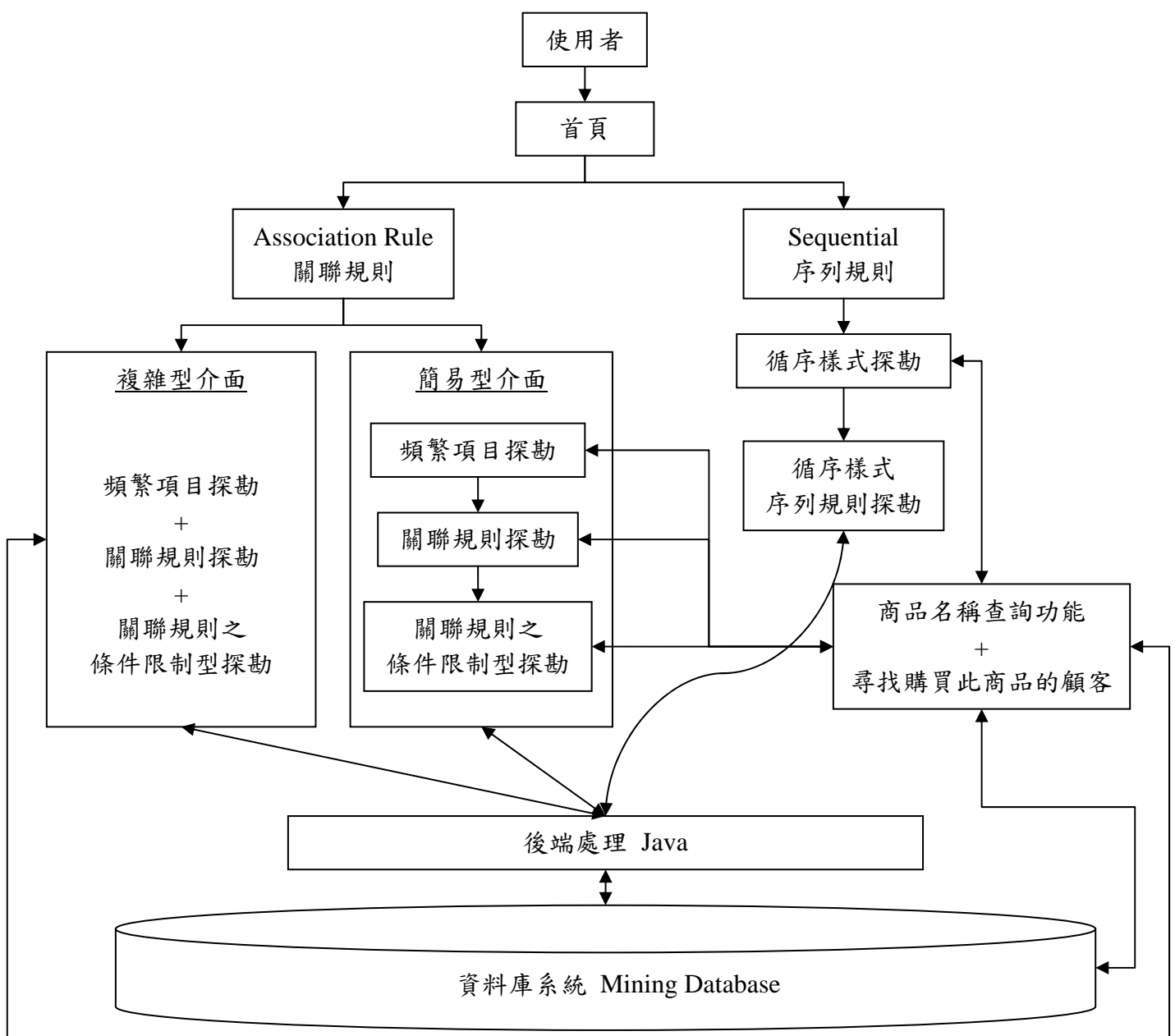


圖 14 系統流程圖

5.2 網頁式操作介面

5.2.1 系統首頁

本系統最主要的功能即為資料探勘，此使用者介面(User Interface)可供使用者選擇資料表、日期限制、演算法、Minimum Support (最小支持度)、Confidence (可信度) 以及輸出介面的型式 (設計有簡易型和複雜型兩種)。

Apriori :

Data Mining	
選擇探勘類型	關聯規則 ▾
選擇資料表	Transaction9 ▾ [說明]
日期限制	<input type="checkbox"/> 請選擇 ▾ to 請選擇 ▾
最小支持度 (Min Sup.)	次數 ▾ 2
信賴度 (Min Conf.)	75% ▾
操作介面	<input checked="" type="radio"/> 簡易型 <input type="radio"/> 複雜型
確定 清除	

《 [Data Mining 簡介](#) | [使用方法](#) | [查詢資料庫](#) | [上傳測試檔](#) 》

Program Design by [FTX](#) Team

Sequential :

Data Mining	
選擇探勘類型	序列規則 ▾
選擇資料表	Seq_Transaction4 ▾ [說明]
最小支持度 (Min Sup.)	次數 ▾ 2
最小間距 (MinGap)	3
最大間距 (MaxGap)	15
合併時間總和 (Sliding window)	2
總時間 (Duration)	25
確定 清除	

《 [Data Mining 簡介](#) | [使用方法](#) | [查詢資料庫](#) | [上傳測試檔](#) 》

Program Design by [FTX](#) Team

圖 15 首頁 web 介面

5.2.2 Apriori 主要功能介紹

我們選擇 Transaction9 資料表做為示範，表示此測試資料共有 9 筆交易紀錄，演算法選擇 Apriori，代表我們希望從中探勘關聯規則。另外將最小支持度(Minimum Support)的次數設為 1，可信度設為 77%，並使用簡易型的操作介面，日期限制則暫不設定。

Data Mining	
選擇探勘類型	關聯規則 ▾
選擇資料表	Transaction9 ▾ [說明]
日期限制	<input type="checkbox"/> 請選擇 ▾ to 請選擇 ▾
最小支持度 (Min Sup.)	次數 ▾ 1
信賴度 (Min Conf.)	77% ▾
操作介面	<input checked="" type="radio"/> 簡易型 <input type="radio"/> 複雜型
確定 清除	

《 [Data Mining 簡介](#) | [使用方法](#) | [查詢資料庫](#) | [上傳測試檔](#) 》

Program Design by [FTX](#) Team

圖 16 index.jsp - Apriori 主功能介面選擇

5.2.2.1 Apriori - 搜尋頻繁項目集合

接著按下「確定」鈕，則可顯示出依所設條件而找出的頻繁項目集合，如下圖：

Apriori Algorithms -- Frequent Itemsets

Next >> [關聯式規則探勘](#) (conf = 77%)

商品長度快速連結： [1](#) [2](#) [3](#) [4](#)

PS: 「點選」該項目名稱，即可知道該項目詳細資料

「」：觀看有購買該列全部商品的顧客

搜尋條件：最小支持度 (Minimum Support) = 1

項目名稱	Sup.
晶工牌溫熱開飲機 	6
大家櫃電動熱水瓶 	7
歌林空氣清淨機 	6
Panasonic 液晶電視 	2
西屋數位倍頻影音光碟機 	2
晶工牌溫熱開飲機, 大家櫃電動熱水瓶 	4
晶工牌溫熱開飲機, 歌林空氣清淨機 	4
晶工牌溫熱開飲機, Panasonic 液晶電視 	1
晶工牌溫熱開飲機, 西屋數位倍頻影音光碟機 	2
大家櫃電動熱水瓶, 歌林空氣清淨機 	4
大家櫃電動熱水瓶, Panasonic 液晶電視 	2
大家櫃電動熱水瓶, 西屋數位倍頻影音光碟機 	2
歌林空氣清淨機, 西屋數位倍頻影音光碟機 	1
晶工牌溫熱開飲機, 大家櫃電動熱水瓶, 歌林空氣清淨機 	2
晶工牌溫熱開飲機, 大家櫃電動熱水瓶, Panasonic 液晶電視 	1
晶工牌溫熱開飲機, 大家櫃電動熱水瓶, 西屋數位倍頻影音光碟機 	2
晶工牌溫熱開飲機, 歌林空氣清淨機, 西屋數位倍頻影音光碟機 	1
大家櫃電動熱水瓶, 歌林空氣清淨機, 西屋數位倍頻影音光碟機 	1
晶工牌溫熱開飲機, 大家櫃電動熱水瓶, 歌林空氣清淨機, 西屋數位倍頻影音光碟機 	1

-- 共 19 筆 --

Next >> [關聯式規則探勘](#)

圖 17 search.jsp 執行限制條件而產生的 Apriori 結果

5.2.2.2 Apriori - 關聯式規則探勘

先前所展示者為測試資料表中的頻繁項目集合，表示這些項目集合的支持度超越了期望門檻值，使用者接著可以進一步探勘出「關聯規則」。例如：藉由分析買了牛奶是否也會同時買麵包的信賴度值等…以下即為執行關聯規則探勘後的結果：

Apriori Algorithms -- Association Rule

搜尋條件：可信度 (Confidence) 大於等於 77%
 Next >> [限制型探勘](#)

[《 回首頁 》](#)

conf. 排序方式：

項目名稱	conf.	sup.
西屋數位倍頻影音光碟機 => 晶工牌溫熱開飲機	100 %	2
Panasonic 液晶電視 => 大家櫃電動熱水瓶	100 %	2
西屋數位倍頻影音光碟機 => 大家櫃電動熱水瓶	100 %	2
晶工牌溫熱開飲機, Panasonic 液晶電視 => 大家櫃電動熱水瓶	100 %	1
大家櫃電動熱水瓶, 西屋數位倍頻影音光碟機 => 晶工牌溫熱開飲機	100 %	2
晶工牌溫熱開飲機, 西屋數位倍頻影音光碟機 => 大家櫃電動熱水瓶	100 %	2
歌林空氣清淨機, 西屋數位倍頻影音光碟機 => 晶工牌溫熱開飲機	100 %	1
歌林空氣清淨機, 西屋數位倍頻影音光碟機 => 大家櫃電動熱水瓶	100 %	1
大家櫃電動熱水瓶, 歌林空氣清淨機, 西屋數位倍頻影音光碟機 => 晶工牌溫熱開飲機	100 %	1
晶工牌溫熱開飲機, 歌林空氣清淨機, 西屋數位倍頻影音光碟機 => 大家櫃電動熱水瓶	100 %	1
西屋數位倍頻影音光碟機 => 晶工牌溫熱開飲機, 大家櫃電動熱水瓶	100 %	2
歌林空氣清淨機, 西屋數位倍頻影音光碟機 => 晶工牌溫熱開飲機, 大家櫃電動熱水瓶	100 %	1

[《 回首頁 》](#)

圖 18 association.jsp 顯示 Apriori 關聯式規則探勘的結果

5.2.2.3 Apriori - 關聯規則之條件限制型探勘

限制型探勘的功能，能夠讓使用者自行篩選探勘後的商品項目以進行分析：

Association Rule -- 限制型探勘

Define: 左項 => 右項

設定值：信賴度(conf.) = 77%，最小支持度(sup.) = 1

左項項目：

右項項目：

《 [回首頁](#) 》

圖 19 association_district.jsp 顯示 Apriori 限制型探勘介面

點選「選擇項目」即會彈出視窗供勾選商品，將視窗關閉則即完成選取動作：

Association Rule -- 限制型探勘

Define: 左項 => 右項

設定值：信賴度(conf.) = 77%，最小支持度(sup.) = 1

點選「選擇項目」即會彈出視窗供勾選商品，將視窗關閉則即完成選取動作：

Data Mining -- 網頁對話			
<input type="checkbox"/> 0 大同電鍋	<input type="checkbox"/> 1 晶工牌溫熱開飲機	<input type="checkbox"/> 2 大家電電動熱水瓶	<input type="checkbox"/> 3 歌林空氣清淨機
<input type="checkbox"/> 4 Panasonic 液晶電視	<input type="checkbox"/> 5 西屋數位倍頻影音光碟機	<input type="checkbox"/> 6 歌林平面電視	<input type="checkbox"/> 7 飛利浦音響
<input type="checkbox"/> 8 幸福MP3/CD手提機	<input type="checkbox"/> 9 LG組合音響	<input type="checkbox"/> 10 MP3隨身聽	<input type="checkbox"/> 11 聲寶CD手提機
<input type="checkbox"/> 12 HITACHI日立組合音響	<input checked="" type="checkbox"/> 13 Esonic液晶顯示器	<input checked="" type="checkbox"/> 14 西屋奈米雙門冰箱	<input type="checkbox"/> 15 LG液晶顯示器
<input type="checkbox"/> 16 TECO東元冰箱	<input type="checkbox"/> 17 西屋液晶顯示器	<input checked="" type="checkbox"/> 18 GoldStar洗衣機	<input type="checkbox"/> 19 大同旋風大烤箱
<input type="checkbox"/> 20 大同微波爐	<input type="checkbox"/> 21 國際電子鍋	<input checked="" type="checkbox"/> 22 西屋臭氧超音波洗衣機	<input type="checkbox"/> 23 西屋清菌光+光觸媒除濕機
<input type="checkbox"/> 24 男/女滑板鞋	<input checked="" type="checkbox"/> 25 卡通兒童	<input type="checkbox"/> 26 庫洛魔法使	<input checked="" type="checkbox"/> 27 女運動鞋
<input type="checkbox"/> 28 義式成人/兒童室內皮拖鞋	<input type="checkbox"/> 29 男多功能運動鞋	<input type="checkbox"/> 30 男牛皮工作鞋	<input type="checkbox"/> 31 休閒男拖鞋/健康造型拖鞋
<input type="checkbox"/> 32 女休閒運動鞋	<input type="checkbox"/> 33 男運動鞋	<input checked="" type="checkbox"/> 34 男牛皮核心氣墊休閒鞋	<input type="checkbox"/> 35 男正式平口/打摺西褲
<input type="checkbox"/> 36 高級羊毛西服+高級羊毛西褲	<input type="checkbox"/> 37 男長袖中格正式襯衫	<input type="checkbox"/> 38 歐風縐花長袖襯衫	<input type="checkbox"/> 39 平口正式西褲
<input type="checkbox"/> 40 超細透氣西褲	<input type="checkbox"/> 41 華爾滋立體織花領帶	<input type="checkbox"/> 42 紳仕自動皮帶	<input type="checkbox"/> 43 運動外套/男休閒背心

http://140.134.26.172:8080/mining/association_choose.jsp 網際網路

圖 20 association_choose.jsp 顯示商品選取介面

經過以上指定動作，點選「確定」按鈕，則會產生出限制項目後的結果：

Association Rule -- 限制型探勘

Define: 左項 => 右項

設定值：信賴度(conf.) = 77%，最小支持度(sup.) = 1

左項項目：

右項項目：

搜尋條件：

左項 [包含]： 歌林空氣清淨機
右項 [不包含]： 大家櫃電動熱水瓶

項目名稱	conf.	sup.
歌林空氣清淨機, 西屋數位倍頻影音光碟機 => 晶工牌溫熱開飲機	100 %	1
大家櫃電動熱水瓶, 歌林空氣清淨機, 西屋數位倍頻影音光碟機 => 晶工牌溫熱開飲機	100 %	1

《[回首頁](#)》

Program Design by [FTX](#) Team

圖 21 association_district.jsp 顯示 Apriori 限制型探勘執行結果

5.2.3 Sequential 主要功能介紹

我們選擇 Seq_Transaction4 資料表做為示範，其測試資料有 4 位顧客，19 筆交易紀錄。演算法選擇 Sequential，此外將最小支持度(Minimum Support)的次數設為 2，最小間距設為 3，最大間距設為 15，合併時間總和設為 2，總時間設為 25。

Data Mining	
選擇探勘類型	序列規則
選擇資料表	Seq_Transaction4 [說明]
最小支持度 (Min Sup.)	次數 2
最小間距 (MinGap)	3
最大間距 (MaxGap)	15
合併時間總和 (Sliding window)	2
總時間 (Duration)	25
確定 清除	

《 [Data Mining 簡介](#) | [使用方法](#) | [查詢資料庫](#) | [上傳測試檔](#) 》

Program Design by [FTX](#) Team

圖 22 index.jsp - Sequential 主功能介面選擇

5.2.3.1 Sequential - 搜尋頻繁項目集合

根據上頁所設定的條件，執行後產生以下的結果：

Mining Sequential Patterns -- with Time Constraints

MinSupport = 2, MinGap = 3, MaxGap = 15, Sliding window = 2, Duration = 25

Next >> [序列規則探勘\(Sequences Rule\)](#)

PS. 「點選」該項目名稱，即可知道該項目詳細資料
「」：觀看有購買該列全部商品的顧客

項目名稱	Sup.
(晶工牌溫熱開飲機) 	3
(晶工牌溫熱開飲機) (大家櫃電動熱水瓶) 	2
(晶工牌溫熱開飲機) (Panasonic 液晶電視) 	2
(晶工牌溫熱開飲機 , 歌林空氣清淨機) 	2
(晶工牌溫熱開飲機 , 歌林空氣清淨機) (大家櫃電動熱水瓶) 	2
(大家櫃電動熱水瓶) 	3
(大家櫃電動熱水瓶) (晶工牌溫熱開飲機) 	2
(大家櫃電動熱水瓶) (Panasonic 液晶電視) 	2
(大家櫃電動熱水瓶) (西屋數位倍頻影音光碟機) 	2
(大家櫃電動熱水瓶) (西屋數位倍頻影音光碟機) (Panasonic 液晶電視) 	2
(歌林空氣清淨機) 	3
(歌林空氣清淨機) (大家櫃電動熱水瓶) 	2
(歌林空氣清淨機) (西屋數位倍頻影音光碟機) 	2
(歌林空氣清淨機 , Panasonic 液晶電視) 	2
(Panasonic 液晶電視) 	3
(西屋數位倍頻影音光碟機) 	3
(西屋數位倍頻影音光碟機) (Panasonic 液晶電視) 	2

-- 共 17 筆 --

《 [回頂端](#) | [回 sequential 首頁](#) | [回首頁](#) 》

圖 23 sequential.jsp 執行 sequential + time constraints 而產生的結果

5.2.3.2 Sequential - 序列規則探勘

此外我們也可對 Sequential 做序列規則探勘，推論出在每筆序列樣式中，各個項目集合之間的關係，其畫面如下：

[GoBack](#)

項目名稱	conf.	sup.
(晶工牌溫熱開飲機) => (大家電電動熱水瓶)	66%	2
(大家電電動熱水瓶) => (晶工牌溫熱開飲機)	66%	2
(晶工牌溫熱開飲機) => (Panasonic 液晶電視)	66%	2
(Panasonic 液晶電視) => (晶工牌溫熱開飲機)	66%	2
(晶工牌溫熱開飲機, 歌林空氣清淨機) => (大家電電動熱水瓶)	100%	2
(大家電電動熱水瓶) => (晶工牌溫熱開飲機, 歌林空氣清淨機)	66%	2
(大家電電動熱水瓶) => (晶工牌溫熱開飲機)	66%	2
(晶工牌溫熱開飲機) => (大家電電動熱水瓶)	66%	2
(大家電電動熱水瓶) => (Panasonic 液晶電視)	66%	2
(Panasonic 液晶電視) => (大家電電動熱水瓶)	66%	2
(大家電電動熱水瓶) => (西屋數位倍頻影音光碟機)	66%	2
(西屋數位倍頻影音光碟機) => (大家電電動熱水瓶)	66%	2
(歌林空氣清淨機) => (大家電電動熱水瓶)	66%	2
(大家電電動熱水瓶) => (歌林空氣清淨機)	66%	2
(歌林空氣清淨機) => (西屋數位倍頻影音光碟機)	66%	2
(西屋數位倍頻影音光碟機) => (歌林空氣清淨機)	66%	2
(西屋數位倍頻影音光碟機) => (Panasonic 液晶電視)	66%	2
(Panasonic 液晶電視) => (西屋數位倍頻影音光碟機)	66%	2
(大家電電動熱水瓶) => (西屋數位倍頻影音光碟機, Panasonic 液晶電視)	66%	2
(西屋數位倍頻影音光碟機) => (大家電電動熱水瓶, Panasonic 液晶電視)	66%	2
(Panasonic 液晶電視) => (西屋數位倍頻影音光碟機, 大家電電動熱水瓶)	66%	2
(大家電電動熱水瓶, 西屋數位倍頻影音光碟機) => (Panasonic 液晶電視)	100%	2
(大家電電動熱水瓶, Panasonic 液晶電視) => (西屋數位倍頻影音光碟機)	100%	2
(Panasonic 液晶電視, 西屋數位倍頻影音光碟機) => (大家電電動熱水瓶)	100%	2

-- 共 24 筆 --

圖 24 Sequential 序列探勘結果

5.2.4 顯示商品項目明細

如果使用者認為產生的結果還不夠詳細，那麼可以藉著點選商品名稱來瞭解此商品更詳細的訊息。我們在這裡點選商品「晶工牌溫熱開飲機」做為示範，則會顯示如下圖所示的資訊：

商品編號	1
商品名稱	晶工牌溫熱開飲機
價格	1888.00

《[顯示有購買此商品的所有顧客](#) | [回首頁](#)》

圖 25 showitem.jsp 所顯示的商品詳細資訊

5.2.5 顯示顧客購買明細

使用者還可以進一步針對購買此項商品的顧客來做分析，由於演算法關係，我們可分為 Apriori 類的搜尋和 Sequential 類的搜尋，如下圖所示：

[Apriori：每筆顧客編號不同]

顯示有購買此商品的所有顧客

顧客編號	購買時間	購買項目
1	2005-01-01	晶工牌溫熱開飲機, 大家櫃電動熱水瓶, 西屋數位倍頻影音光碟機
4	2005-01-04	晶工牌溫熱開飲機, 大家櫃電動熱水瓶, Panasonic 液晶電視
5	2005-01-05	晶工牌溫熱開飲機, 歌林空氣清淨機
7	2005-01-07	晶工牌溫熱開飲機, 歌林空氣清淨機
8	2005-01-08	晶工牌溫熱開飲機, 大家櫃電動熱水瓶, 歌林空氣清淨機, 西屋數位倍頻影音光碟機
9	2005-01-09	晶工牌溫熱開飲機, 大家櫃電動熱水瓶, 歌林空氣清淨機

《[回首頁](#)》

圖 26 showitem.jsp 顯示哪些顧客購買了此商品

[Sequential：多筆顧客編號相同]

顯示有購買以下商品的顧客

1. 晶工牌溫熱開飲機 2. 歌林空氣清淨機 3. 大家櫃電動熱水瓶

顧客編號	購買時間	購買項目
1	2005-01-03	歌林空氣清淨機
	2005-01-05	晶工牌溫熱開飲機, 歌林平面電視
	2005-01-18	大家櫃電動熱水瓶
	2005-01-31	晶工牌溫熱開飲機
	2005-02-14	歌林平面電視
2	2005-01-06	晶工牌溫熱開飲機, 歌林空氣清淨機
	2005-01-10	大家櫃電動熱水瓶
	2005-01-17	西屋數位倍頻影音光碟機
	2005-01-18	晶工牌溫熱開飲機
	2005-01-24	歌林空氣清淨機, Panasonic 液晶電視

--- 共搜尋到 2 位顧客 ---

《[回首頁](#)》

Program Design by [FTX](#) Team

圖 27 sequential_showitem.jsp 顯示哪些顧客購買了此商品

5.2.6 使用者上傳測試檔

為了能增進和使用者的互動，我們增設此功能，讓使用者能依照預設的輸入格式，將測試檔案上傳至伺服器中，透過我們所提供的 Apriori 演算法的運算，將探勘成果呈現出來。

📌 注意事項

1. 測試檔內全部為**數字**格式，並且檔名需使用**英文命名**
2. 測試檔格式為「**顧客編號** **時間編號** **數量(依後面的商品數量決定)** **商品編號**」
Ex: **1 1 3 1 2 3**
 2 2 4 1 2 3 4 (範例檔)
3. 目前只提供 Apriori 演算法
4. 上傳檔案不可超過 **1 MB**，且為文字檔格式(.txt)

Minimum Support 次數

可信度 (Confidence)

請選擇要上傳的檔案

《[回首頁](#)》

Program Design by [FTX](#) Team

圖 28 upload.htm 使用者自行上傳 Apriori 測試檔

執行結果：

Apriori Algorithms -- Frequent Itemsets

Next >> [關聯式規則探勘](#) (conf = 75%)

商品長度快速連結: [1](#) [2](#) [3](#)

搜尋條件：最小支持度 (Minimum Support) = 2

項目名稱	Sup.
1	6
2	7
3	6
4	2
5	2
1, 2	4
1, 3	4
1, 5	2
2, 3	4
2, 4	2
2, 5	2
1, 2, 3	2
1, 2, 5	2

-- 共 13 筆 --

Next >> [關聯式規則探勘](#)

圖 29 searchbyuser.jsp 上傳檔執行結果

5.2.7 查詢資料庫

本系統的資料皆儲存於資料庫(Database)當中，使用者可以利用此功能來查詢資料庫裡面的內容，亦可從中新增、修改或刪除資料。操作畫面如下圖所示：

[\[首頁\]](#) 請選擇想要顯示的資料表：

Transaction9

 顯示方式：☒ 原始資料 ☐ 套用日期商品相關資訊

Transaction9 內容

《[第一頁](#) | [前一頁](#) | [後一頁](#) | [最後一頁](#)》 每頁顯示 50 筆資料 頁數：1/1

快速跳頁

Cid	Tid	Items
1	1	1 2 5
2	2	2 4
3	3	2 3
4	4	1 2 4
5	5	1 3
6	6	2 3
7	7	1 3
8	8	1 2 3 5
9	9	1 2 3

《[第一頁](#) | [前一頁](#) | [後一頁](#) | [最後一頁](#)》
《[回首頁](#)》

Program Design by [FTX](#) Team

圖 30 執行查詢資料庫結果

第六章 總 結

在這次專題研究中，我們設計並建立了一個具備多項條件限制的，以網頁為基礎的關聯規則與序列規則探勘系統。我們輔以市場購物籃的情境做為背景，以實際展示進行資料探勘後的結果。針對關聯規則與序列規則的探勘，我們首先讓使用者選擇輸入檔案並設定參數，採用循序漸進式的方式，一個步驟接著一個步驟完成資料探勘並呈現出結果。關聯規則探勘使用 Apriori 架構之演算法，序列規則探勘使用 Pattern-growth 架構的方法論，發展網頁式系統的工具為 Java 與 JSP。

在關聯規則的部分，首先選擇資料表並選擇性指定日期限制，接著設定最小信賴度與最小支持度值，即可從資料中找尋出頻繁項目集合，並進一步推導出關聯規則。本系統設計有兩項額外功能，針對任何頻繁項目以及頻繁項目集合，我們可以讓使用者檢視其在資料表中的出現位置（進而得知特定顧客的購物狀況與相關資訊）；另外針對關聯規則，我們提供限制型探勘功能，使用者可以指定或排除規則的左項和右項（前項和後項）的商品項目，並取得篩選與過濾之後的關聯規則。在序列規則探勘的部分，基本形式如同關聯規則的步驟，首先指定參數，接著找尋出循序樣式並推導出序列規則。然序列規則增加了時間因素，因此參數（條件限制）方面比較複雜，包含最小間距、最大間距、滑動時間窗、總時間等。循序樣式的結果頁面，同樣可以查閱購買單向商品，或者整組樣式所有商品的顧客交易紀錄。

本系統具有相當簡潔且人性化的網頁式操作介面，首頁提供有資料探勘的觀念介紹，以及使用方式說明。使用者可以針對資料庫內容進行檢視，也可以自行上傳符合格式的測試檔案。在關聯規則的部分，並設計有簡易型和複雜型兩種介面，提供不同專業程度的使用者選擇較為適合的模式。本系統提供探勘關聯規則與序列樣式的基本功能與條件設定，同時具備有篩選與逆向查閱的功能，具備高度的完成度與實用性。其測試過的最大輸入資料為 10,000 筆交易紀錄，可以正確無誤並且相當有效率地執行探勘動作。在序列規則的探勘方面，針對時間因素的較複雜情況，本系統可以推導出前項包含一個項目集合（項目）的序列規則。如何設法延伸此功能，使其可以推導出無限數量的所有可能的序列規則，將是一個頗具難度的挑戰。同時，如何在數量更為龐大的資料庫、資料表（例如 50,000 筆或者 100,000 筆資料）中保持高效率進行探勘，也將是一項重要的努力發展目標。

最後感謝林明言老師的熱心指導，使我們能順利的完成此專題。在專題製作期間，也曾遇到硬體損壞的問題，老師總是能在第一時間給予我們支援。而在資料探勘的基礎觀念或是到程式實作方面的問題，老師運用他豐富的知識和經驗耐心地為我們解說，並給予我們很大的信心和鼓勵，讓我們能順利地克服萬難，真的是受益良多。因此我們也很慶幸能找到林明言老師當我們的專題指導老師，他對我們的付出我們感受在心，也非常由衷的感謝。其次也感謝組員們的辛勞，大家平常都很忙碌，但還是抽空撥出時間做專題，遇到困難彼此互相討論，發揮團隊合作的精神，使得這專題能成功順利的完成，感謝大家的奉獻和努力，這一路上辛苦了！

參考資料

- [1] 王波，FreeBSD 入門與應用。機械工業出版社，2001 年 2 月
- [2] 王俊斌，FreeBSD 架設管理與應用。博碩文化，2003 年 6 月
- [3] 林上傑、林康司，JSP 2.0 技術手冊。碁峯，2004 年 4 月
- [4] 榮欽科技，Java 2 入門與實務應用。碁峯，2004 年 9 月
- [5] FreeBSD 4.X，5.X 及 6.X 常見問答集，<http://www.freebsd.org/zh/FAQ/>
- [6] OHaHa's 學習心得，<http://ohaha.ks.edu.tw/>
- [7] 張家汶，具時間限制之高效率序列樣式探勘演算法，逢甲大學資訊工程學系，碩士論文，2006 年
- [8] Agrawal R. and Srikant R. Fast Algorithms for Mining Association Rules. In *Proc. of the 20th Int'l Conference on Very Large Databases*, Santiago, Chile, September 1994.
- [9] Agrawal R. and Srikant R. Mining Sequential Patterns. In *Proc. 1995 Int. Conf. Data Engineering (ICDE'95)*, Taipei, Taiwan, Mar. 1995.
- [10] Agrawal R. and Srikant R. Mining Sequential Patterns: Generalizations and Performance Improvements. *Proceedings of the 5th International Conference on Extending Database Technology*, Avignon, France, 1996.
- [11] Jiawei Han. Data Mining: Concepts and Techniques. Morgan Kaufmann. Aug. 2000
- [12] Michael Lucas 著，藝立協譯，FreeBSD 完全探索。上奇，2003 年 9 月
- [13] Lin, M. Y. and Lee, S. Y. Efficient Mining of Sequential Patterns with Time Constraints by Delimited Pattern-growth. *Knowledge and Information Systems*. Volume 7, Issue 4, May 2005.

附 錄

附錄 A 工作分配表

●：主要負責 ◎：次要負責或合作 ○：參與協助

工 作 \ 組 員	林榮章	李兆偉	林宏軒	廖健峰
軟硬體環境配置	●	○	◎	○
導論與概述撰寫	◎	●	○	●
Data Mining Table	◎	●		
ER Model		●		
Association Rules 觀念介紹	●	●		
Sequential Patterns 觀念介紹	○	●	○	○
Time Constraints 觀念介紹	◎	●		
FreeBSD 相關介紹	●		◎	
相關書籍與文獻研讀	●	●	●	●
Java & JSP 相關	●		○	
MySQL 相關	●			
ARTool 介紹	○	○	○	●
JavaBean 介紹	●	●	○	○
資料庫建立	●	◎	●	●
演算法(程式碼)實作	●	●	◎	◎
JSP 網頁製作	●	○	○	●
報告修訂與整合	●	●		