第貳章 文獻探討

本章中我們將先探討有關資料挖掘相關議題，特別是以 Agrawal 的 Apriori 演算法做資料發掘的方法。之後我們再針對以全球資訊網為資料挖掘對象的現有演算法做資料整理，並討論其中與本研究的差別，以做為本研究資料挖掘方式的對照。

2.1 資料挖掘(Data Mining)

隨著資料收集科技與網路技術的進展，大量使用者所產生的資料被公司或組織有效率的收集起來。但是面對如此巨大的資料量，組織往往無法分析，或是僅能做粗淺的統計資料計算。這樣的現象引發了近年來 Data Mining 這個領域的發展。

資料挖掘(Data Mining)或被稱為資料庫裏的知識發現(Knowledge Discovery in Databases, KDD)是將隱性的、顯然還不知道且具有有用潛力的資訊經由一種不簡單(Nontrivial)的方式由資料中萃取出來的過程。(Frawley et al., 1991)資料挖掘演算法的技術經由學者們不斷的發展，已有十分豐富的文獻。各種針對特別資料庫所產生的挖掘演算法也一一的被提出，並且廣泛的被人們所討論。(Frawley et al.1991; Agrawal and Srikant,1994、Han and Fu, 1995)

本節第一部份首先將先介紹資料挖掘的分類，第二部份將對應用現有的資料挖掘技術所能找出的規則做分類的描述，第三部份將對各類資料挖掘的演算法做

介紹，尤其是對於本研究主要採用的 Apriori 演算法，在這部份會做更深入的介紹。

2.1.1 資料挖掘的分類

就原始資料的型態，我們可以將資料挖掘分成定量資料型式(Quantitative Type)與定性資料型式(Qualitative Type)兩種。定量型式的原始資料，是由數字所集合而成的數據，如某股的股價指數、某地的溫溼度或降雨量等。定量型式的原始資料可用統計方法及數值分析的方法來處理。而另一種定性型式的原始資料，他的集合就不一定是數字，主要是分類符號所組合而成的數據，如超商的購物車貨單、網頁的瀏覽路徑等。定性型式的資料在學習的處理上，可以分為二類(Quinlan,1986)，第一類是適應型系統(Adaptive System)，第二類是以概念(Concept)或規則(Rule)為呈現方式來獲取結構性的知識，也就是歸納型的學習過程。以下就適應型學習與歸納型學習做較詳細的說明：

一、適應型系統：

將學習網路的內容視為一個黑盒子。針對輸入的訓練資料，觀察得到結果是否正確。如不正確，就會調整學習網路的架構使其正確。經過訓練資料輸入過程的不斷調整之後，學習網路就可以進行測試。如果學習網路通過測試資料的測試，今後如果遇到新的資料，這個學習網路就會利用以往的經驗，自行推導出正確的結果，類神經網路是這類學習典型的例子。只是這個方式的產出是一個系統，並非一條一條的規則，組織在使用上並不方便。所以當資料挖掘遇到非線性或相當複雜的問題時，也可以使用類神經網路或其他適應型的系統來處理(Lewinson,1993)，不失為重要的一種方法。

二、歸納型系統：

歸納學習方式，因為它能找出 If… Than…Else 這種型態的規則或概念，所以是在資料挖掘領域中較常被應用的方式。常見的有關聯分析法、樹狀分類法與概念樹推導分析法三種。

關聯分析法主要是要找出各類資料模式之間的關聯性，藉由計算顆粒大小不同資料集合的出現次數，找出較常出現的資料模式(Pattern)，最後利用不同的資料模式的出現次數，就可以找到每個可能的規則（Agrawal, et al., 1992）。樹狀分類法是利用每個屬性值逐次來分類資料庫的資料，最後形成的決策樹，日後如有新資料的話，就可以利用決策樹上的決策點進行分類（Quinlan, 1986）。另一種歸納學習的方法是概念樹的分析法，這個方法利用一個的概念建立起一個多層的概念階層，藉由將資料不斷的抽象化，來達到表現出較高意義上的資料關係(Han,Cai and Cerocne,1993)。

2.1.2 資料挖掘可找出規則種類

資料挖掘技術經由學者們不斷的討論與發展，已可以找出種類豐富的規則。本節將針對能找到的常見規則在說明與整理：

a. 關聯規則：能找出資料庫中項目之間的關聯性。當資料為一個集合時，該規則代表在某個 itemset 出現的條件下，也同時會出現另一個 itemset 的狀況。在商業應用上，常會應用於客戶在所購賣物品中的關聯，例如可以找出這樣的規則：「有 75%的客戶在

購買牛奶之時，又會連帶的購買麵包。」(Agrawal and Srikant,1994、Han and Fu, 1995)

b.特性規則：當使用者想瞭解研究的標的是否有具備某些特性時，可以應用特性規則尋找的方法來找尋特性規則。一個特性規則可能是「有 90%的感冒患者，會出現發高燒、頭痛與流鼻水的症狀。」(Han, Cai and Cercone, 1993)

c.區別規則：當使用者想要明顯分辨目標群體和對照群體之間所具有的不同特性時，就可以採用找區別規則的演算法來找到區別規則。例如說「能區分『感冒』和『水痘』之間不同的病癥是流鼻水和全身發癢。」(Agrawal, et al., 1992; Agrawal, Imielinski and Swan, 1993)就是一個區別規則。

d.演化規則：具有特性與區別規則兩種混合的屬性，一方面，演化規則會討論各階段資料具有什麼樣的特性；另一方面，演化規則也會討論現在和過去的資料如何區別。(Han, Koperski and Adhikary, 1996)

f.分群規則：資料庫中依選定的特性做分群。使得各群內的差異小，群與群之間的差異大。(Chen, Han and Yu, 1996、Ramkumar and Swami,1998)

e.序列模式：能找出計劃中連續動作的固定模式。例如「飛機航班之中所降落機場的規模是先降落幾次的小機場,之後再降落一次大機

場的模式」(Agrawal and Srikant,1995、Han, Yang and Kim, 1999)

在下個小節中，我們會對幾個與本研究較有關的演算法做較仔細的介紹。

2.1.3 資料挖掘的演算法

上一小上節所介紹的各式規則中，在背後都有一些特定的演算法去實作它。資料挖掘的演算法可藉由詳細定義資料庫內的資料，在其配適的應用領域中，以有限的發掘過程中控制每項參數，使其達到能找尋出隱含在資料庫內滿足使用者目的的規則。本節要介紹資料挖掘二個主要的演算法，Apriori 演算法與 Han 的概念樹演算法。

・Apriori 演算法

在找尋關聯規則中，最具代表性的做法就是 Agrawal 於 1994 年所提的 Apriori 演算法。這個演算法可分為兩個部份，第一部份是要在資料庫中找尋出所有的大項目組合(Large Itemset)，其基本精神就是利用掃描資料庫來計算各個候選項目組合(Candidate Itemset)的支援值(support)。一但 support 值超過某個最小 support 值(min. support)，這樣的候選項目組合即可被稱為大的項目組合。再經由 Join 兩個上一階的 Large Itemset，可以產生出下一階的候選項目組合。有了新一階的候選項目組合後，就再重覆的掃描下去，一直到再也找不到新的 Large Itemset 為止。第二部份則是從大項目組合中找出可能的關聯規則。這樣整個的演算過程就是 Apriori 演算法。圖 2-1 是 Apriori 演算法的示意圖。

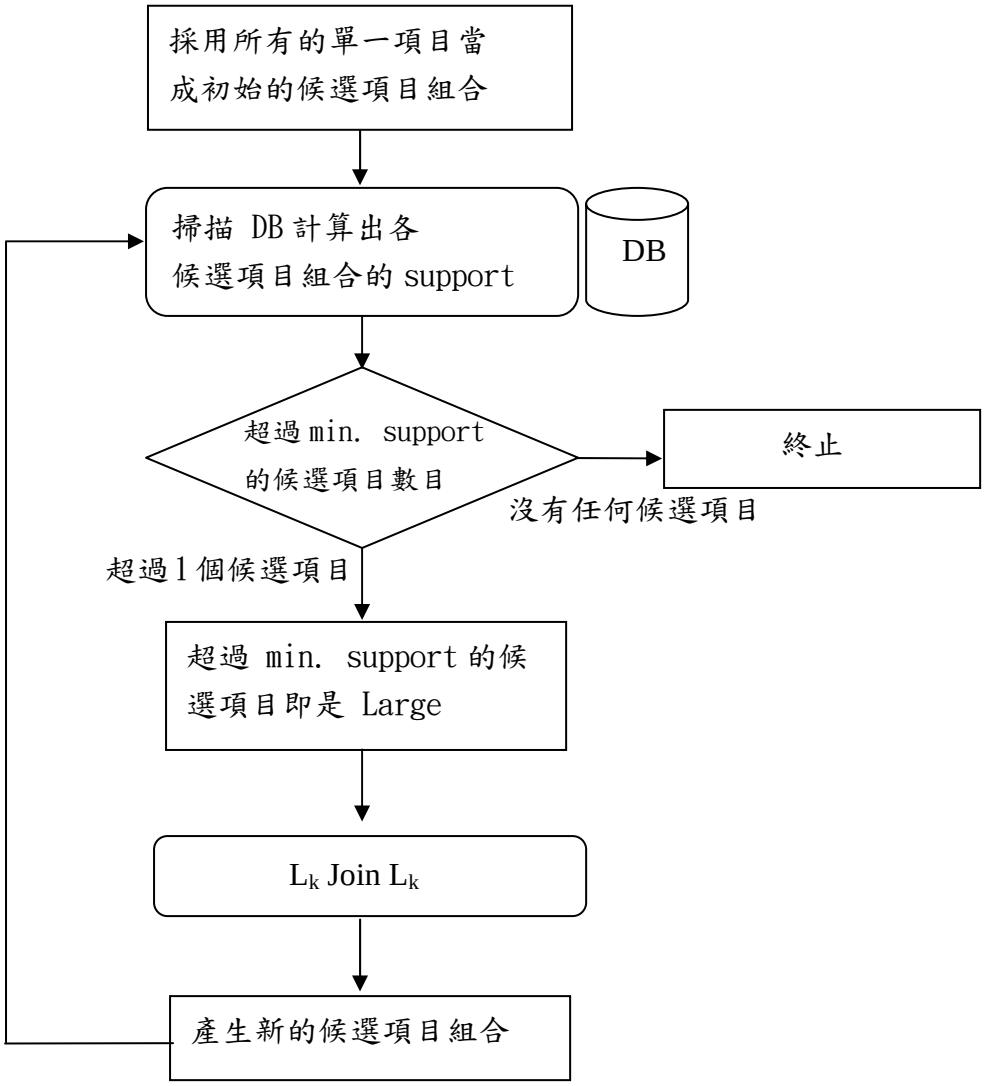

圖 2-1 Apriori 演算法的示意圖

在 Apriori 演算法中的找尋 Large Itemset 的部份中，往往佔了整個挖掘的大部份的時間。有許多的學者們提出改善的方法；學者 Savasere 在 1995 年提出了一個 Partition 的演算法，經由將資料邏輯上切分為幾個不重疊的資料庫，而加強了 Apriori 的效能。學者 Toivonen 在 1996 年使用了隨機抽樣的方法先快速大量的找出 Large Itemset，雖然說在第一輪中有可能會有某些項目會遺失，但是可再進行下一輪的掃描來取得遺失項目。學者 Pasquier 在 1999 年使用了最小

封閉方格法來處理關聯規則，也成功的加速了 Large Itemset 的尋找過程。

Apriori 演算法對於在找尋關聯規則上，是很重要的一個演算法。在我們的研究中，雖然在很多方面都因為所研究的標的不同，而有些許的差異。但其中演算法的觀念與 Apriori 的想法基本上還是一樣的。

・概念樹演算法

Han 的概念樹演算法是最近很受到重視的一種資料掘取演算法。該演算法主要精神是由各領域專家所建立概念階層(Concept Hierarchy)開始，這個概念階層代表資料意義的抽象層級。最高的概念階層是以 ANY 代表的全部概念，其下會有一至多層的概念層來表達一項資料的抽象層級概念，圖 2-2 就是概念樹的例子。其中描述了學生分類的抽象概念。

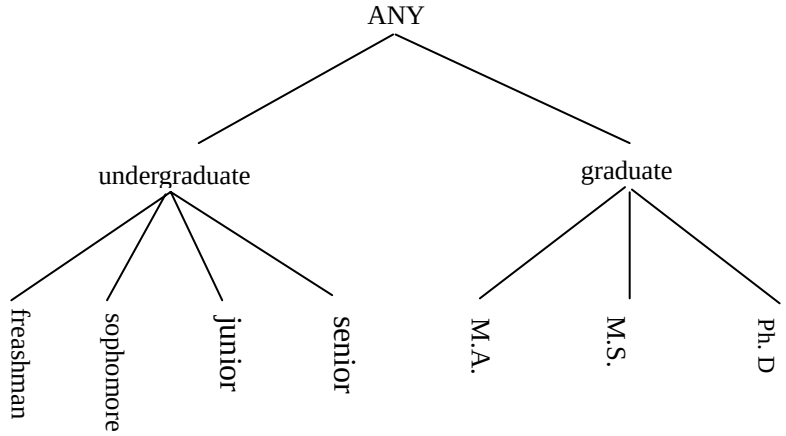

圖 2-2 一個概念樹的例子(資料出處：Han,Cai and Cerocne,1992)

有些的概念階層可以被自動或半自動的發現。數值性的屬性可以被組織成一個離散的階層概念，這個階層可以利用資料庫統計的方法自動建構出來(Han,Cai

and Cerocne,1993)。Han 的概念樹演算法在資料掘取的過程中，就是依造這種概念樹來不斷的向上累計來進行一般化(Generalized)，一直到一般化的個數滿足預設的一個門檻值為止。最後再整理成人類能瞭解的邏輯規則。這種方法因為整合了領域的專家，所以找出的規則並不是單純經由資料比對出來的零碎資料，而是真正與領域相結合的知識。

2.2 網路上的規則挖掘(Web Mining)

隨著近十年來全球資訊網的發展，在 WWW 上的資料挖掘已成為一個引起學者們注意的領域，雖然 Web Mining 這個名詞我們廣泛的使用，但是它並沒有一個很正式的定義。廣義的來說，Web Mining 可以泛指一切把未知、有用且隱藏在網路中資訊所找出來的活動。Web Mining 一般而言可以分為以下三個種類(Cooley,1997)：

一、針對網路資料內容的挖掘(Web Content Mining)：對於實際存在於網站中的資料當做原料來進行資料發掘。包含以 HTML 語法格式資料的原始網頁資料發掘(Web Page Content Mining)；或是對使用者利用搜尋引擎所得的查詢結果的資料進行的資料發掘(Search Result Mining)。

二、針對網站的鏈結結構的挖掘(Web Structure Mining)：包括發掘網站間的超鏈接結構的組成，以及網站被別個超鏈接參照間的關係。

三、針對網站的使用狀況的挖掘(Web Usage Mining)：包括網站伺服器 Log 檔的資料發掘(Chen,Park and Yu,1998;Pei,et al.,2000;陳仕昇等，民 88)，

這類的資料挖掘目的在於找出使用者在網站中的瀏覽行為模式，以用來對網站整體的架構進行調整或是調整符合使用者需求的服務。

我們的研究在這個分類中是屬於網站的使用狀況的挖掘。在下個小節中我們將介紹網站使用狀況的挖掘和幾個有關的文獻。

2.2.1 網站使用狀況的挖掘(Web Usage Mining)

對網站使用狀況的挖掘，往往是經營者為了強化其對使用者的瞭解而進行的。所以挖掘的內容都是與使用者行為有關的資料。常見的是使用者瀏覽網頁路徑的挖掘。在實做 Web Usage Mining 時，一般可以分三個階段(Cooley et al.,1999)：

一、前置處理(Preprocessing)：將可使用的 Log 檔，轉換成可進行演算法處理的資料。期間包含資料清洗(data cleaning)、使用者判定(user identification)、任務判定(session identification)、填滿路徑(path complete)及交易判定(transaction identification)直到 Log 檔已轉換成可進行演算法運算為止。

二、演算法(Mining Algorithm)：網站使用狀況挖掘的主要核心，針對問題的本身的特點所發展出的在有限個步驟間就可以找到規則的一個方法。

三、模式分析(Pattern Analysis)：在找出規則之後，還要一個評估規則的方法，來分析找到的模式是否合理或是有用。

在實行完這三個階段後，研究者就可以針對找出的使用者行為模式，進行網站版面的修改或是新增使用者有需求的功能，甚至可以在使用者的行為路徑中安排劇情式的廣告來達到強化網站效能的作用。以下我們列舉一些相關的文獻，並且說明我們的研究與其不同之處：

一、最大向前參考序列的演算法：Chen 等人(1998)提出一個 MF(Maximal Forward)的演算法，將 Log Data 轉成最大向前參考序列，演算法強調 Web 上的應用，要找出使用者瀏覽網頁的模式，但因預設使用者瀏覽網頁是不能有迴圈的。這和一般瞭解的網頁瀏覽行為較不相同。

二、以停留網頁時間做門檻值的演算法：Hsieh 和 Chang (2001)提出一個 ITIM(Integrated Transaction Identification Module)演算法，主要是將使用者停留網頁的時間當作門檻值，來過濾出使用者有興趣網頁，進而求得最常瀏覽路徑。

三、在多維資料庫中找尋序列模式的演算法:學者余宗慶與陳彥良(2002)提出了一個在多維資料庫中找尋序列模式的演算法。這個演算法可以針對本研究相同的網頁-動作路徑來進行序列資料的挖掘，但是由於本研究所採用混合式的掃描方法與其不同，能夠尋找出的規則也有所不同。

四、從 Log 檔中找出關聯規則及序列模式的演算法：(Cooley et al., 1997)該演算法計算 support 的方式是看 Sequential pattern 是否在一筆交易中出現，但是因為並未考慮到單筆交易可以出現重覆次數，及並未考量使用者在網頁中的動作，所可會容易忽略到使用者記錄中所更隱藏的內含。

五、可重複序列挖掘網路瀏覽規則的演算法：陳仕昇、許秉瑜與陳彥良(民 88)提出了 Next Pass Large Threshold 與 Next Pass Large Sequence 演算法，增進了一般瀏覽規則無法重覆計算 support 的缺點。由於本研究重覆計算的方法是由此演算法改造而成，所以在這裏我們要仔細的介紹一下這個演算法的核心概念。可重複序列演算法的核心概念有二，一是門檻值的計算不同。一般不重覆計算的演算法，在門檻值的計算上，只需將資料總數乘上百分比型態的 Min.Support 即可。但是在重覆計算的演算法中，單一資料的 support 數可能會大於 1，所以我們要先計算出所有可能 Support 值的總數，這個總數在每次掃描階數不同時(Pattern 不一樣長時)，皆會不同。算出總數後，乘上百分比型態的 Min.Support 就得到了門檻值了。第二個核心概念是往後看 X 步的想法，圖 2-3 說明瞭可能發生的狀況。

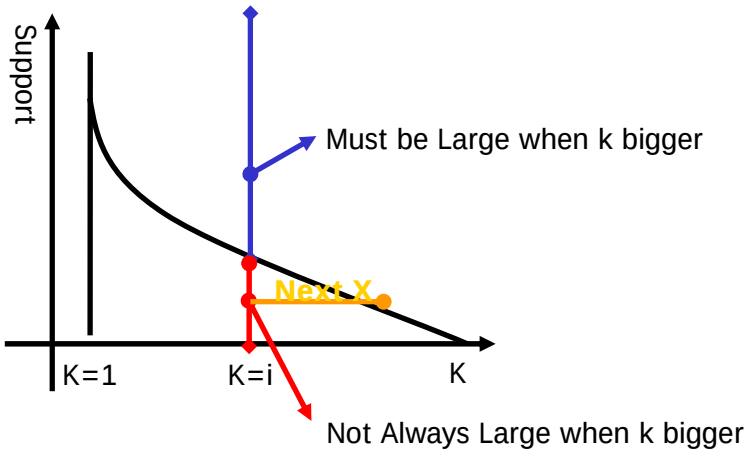

圖 2-3 Next Pass Large Threshold 與

Next Pass LargeSequence 演算法往後看 X 步的概念

在普通的演算法中，只要你計算出的 support 值小於門檻值，那就不能算 Large，在下一輪產生 CS 時就沒辦法加入生成下一輪的 CS。但是在重覆計算的演算法中，這樣做可能會遺失掉一些可能的 Large 的 Pattern。假設我們計算一個 Pattern，在重覆計算的狀況下它的 support 為 5，但是這一階的門檻值要 6 才會 Large，所以我們不會算這個 pattern Large。而下一階的門檻值可能降低成 3，而由剛剛不 Large 的 Pattern 所生成的下一階的 Pattern，就有可能 support 會超過 3，應該被認為 Large 才是。為了儘量避免遺失資料，所以第二個核心的想法就被引入使用，當一個 Pattrn 被掃描完之後，狀況不再是只有 Large 或不 Large 了。而是多了一個 NextPassX Large 的狀況，這說明瞭下 X 步，這個 Pattern 所生成的子序列是還可能 Large 的。所以，這類的 Pattern 雖然不能加入找尋規則的行列中(不算 Large)，但是卻可以加入生成下一階的 CS。陳仕昇提出重覆計算的想法，但是因為沒有考量到使用者動作的構面，所以還是無法找出與之相關的規則。

六、將購買物品與瀏覽路徑結合的演算法：Yun 與 Chen(2000)提出一個 WTM(Web Transaction Mining)演算法，主要是將使用者在購物網站中所購買的物品與瀏覽路徑相結合，以求得使用者可能在哪些網頁購買特定商品。

七、將時間與瀏覽網頁次數加入探勘法則中的演算法：Zhang 等人(2000)所提出的一篇文獻中，則是將使用者瀏覽網頁的次數與時間加入瀏覽序列探勘中，來預測使用者下一步可能點選的網頁，這樣可以將網頁預先下載至使用者電腦中，以增加網頁瀏覽的速度。但由於並未考量使用者行為，所以也無法找出相關的規則。

綜合來說，由於本研究討論的是一個過去較少人談到的模式，對於相關的論文很少有能力再多處理使用者動作路徑的部份，唯一可以採用的多維資料庫演算法，又因為缺少重覆計算能力，以及掃描方式不儘相同，跟本研究有些許相異之處。