第二章 文獻探討

第一節 資料探勘

一、資料探勘定義

資料探勘是電腦應用領域的新名詞。然而當人類還在茹毛飲血的上古時代早已進行著資料探勘的行為，為了快速並準確捕獲獵物，人類的祖先必須細心觀察獵物的習性，並預測獵物的行為，才能戰勝獵物、存活下去。同樣的，在講究即時、競爭激烈的網路時代，如果能事先破解消費者的行為模式，將會是贏得電子商務的關鍵因素之一。

知識的貢獻大於資訊，資訊的貢獻大於資料，資料探勘就是從大型的資料庫中萃取出隱藏的預測資訊(Inmon, 1996)。在商業上可以有效地協助企業利用資料倉儲做為預測未來的趨勢以及行為分析，也是建立企業智慧最主要的方法之一，因此有人稱之為資料庫的知識探勘、資料考古學(Data Archaeology)或資料型態分析(Data Pattern Analysis) (梁定澎，2004)。若能由資料中找出顯著且有用的決策模式或決策法則，進而正確地預測未來的行為，將能增加更多企業智慧，資料探勘可以創造出許多新的知識，透過這些知識的發現、儲存、分享、利用，是重要的知識產生工具。下表 2-1 為本研究整理出各個不同研究者對於資料探勘的定義及論述。

表 2-1 資料探勘的定義與研究整理表

學者（年代）	資料探勘的定義
Frawley（1991）	從資料中萃取出隱藏的、先前未知的有用資訊。
Fayyad（1996）	可萃取出資料中有效的、嶄新的、具潛在效益的資訊之非細瑣過程，其最終目標是瞭解資料的型態。
Berry & Linoff（1997）	為了發現有意義的模式或規則，以自動或半自動的方式，來勘查、分析大量資料所進行的流程。
Lewis & Bernstein & Kifer (2001)	資料探勘是一個發掘知識的過程，可以在一個大型資料集合結構中找到所需要的特徵知識。
Gnardellis & Boutsinas（2001）	認為資料探勘為資料庫中知識發掘過程的一個步驟。

資料來源：本研究自行整理

二、資料探勘的發展歷程

資料探勘的發展歷程，主要可分成四個階段，如下表 2-2 所示(Kimball, 1996)。

表 2-2 資料探勘的發展歷程

年代	內容
1960s	資料蒐集磁帶、軟碟、硬碟
1980s	資料查詢、資料庫(SQL 語言)
1990s	資料統計、資料倉儲、OLAP
2000s	資料分析、資料探勘

資料來源：本研究自行整理

由表 2-2 我們可以看出資料探勘的技術隨著時間的演進不斷發展及進步，在早期 1960 年代時，對於管理者而言，資料只要能夠方便攜帶及存取則能足夠滿足管理者的需求，但到了 1980 年代後，管理者漸漸了解到資料能夠用來查詢及記錄，才能增加管理上的效率，如當時所開發的資料庫(SQL 語言)。在 1990 到 2000 年後，資料並不只是用來保存而是用來分析與應用，找出有效之資訊提供給決策管理者，資料不僅僅是管理者所關

心的議題，更是各個研究領域研究者所要深入探討的，就如影像分析、線上及時處理、資料庫系統及人工智慧等等研究領域，都需要擷取資料來做歷史行為的分析並建立相關預測模型及規則，找出有用之資訊，也就是我們現今所謂的資料探勘。

三、知識發現的過程

在知識發現的過程中，我們必須先從資料庫中預先萃取有用的歷史資料來進行資料的準備過程，這些準備過程包括：資料的選取(Data Selection)、淨化(Cleaning)、欄位擴充(Enrichment)及編碼(Coding) (Liu & Guo, 2001)。

(一). 資料選取

例如要從一大型資料中找出 Jonson 預購買何種類型之雜誌。如下表 2-3 所示。

表 2-3 資料選取示意表

ID	Name	Address	Date	Magazine
U00001	Bill	Taiwan	04-15-94	Car
U00002	Andy	Taiwan	06-15-96	Car
U00003	Marry	Taiwan	02-23-98	Music
U00004	Italy	China	04-15-94	Comic
U00005	**Joson**	USA	03-12-93	House

資料來源：本研究自行整理

(二). 淨化

在歷史資料中未必所有的資料都為完整的資料，有時因為填寫基本資料而有錯誤或是輸入人員的疏失而造成原始資料出現遺漏值(Null)，這將會影響預測模型的準確度度下降，所以我們必須透過淨化的動作改善遺漏值所造成的影響。下表 2-4 為遺漏值之示意表。

表 2-4 遺漏值之示意表

ID	Name	Address	Date	Magazine
U00001	Bill	Taiwan	04-15-94	Car
U00002	Andy	Taiwan	06-15-96	Car
U00003	Marry	Taiwan	02-23-98	Music
U00004	Italy	China	**null**	Comic
U00005	Joson	USA	03-12-93	House

資料來源：本研究自行整理

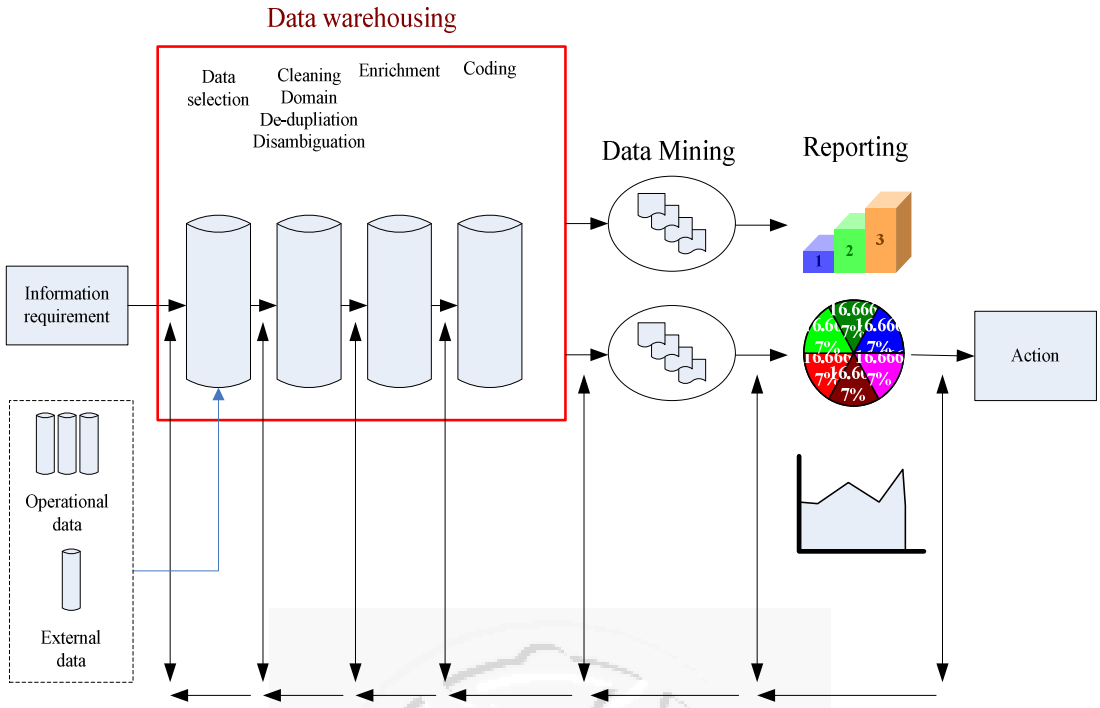
(三). 欄位擴充

在做資料探勘前，我們必須事先發現所要研究資料的維度是否不足夠，若不足夠常常會造成預測的不卻定性及不準確性。所以研究中必須要熟悉所探討的資料，並加以調整維度。

(四). 編碼

資料探勘中有許多的演算法，資料的屬性必須要相容於某演算法所建立之模型，才能趨動研究者所建構之預測模型。資料的編碼目前在資料探勘領域中也是非常重要的一項過程，常常也會因為編碼的不正確而降底了預測模型的準確度。

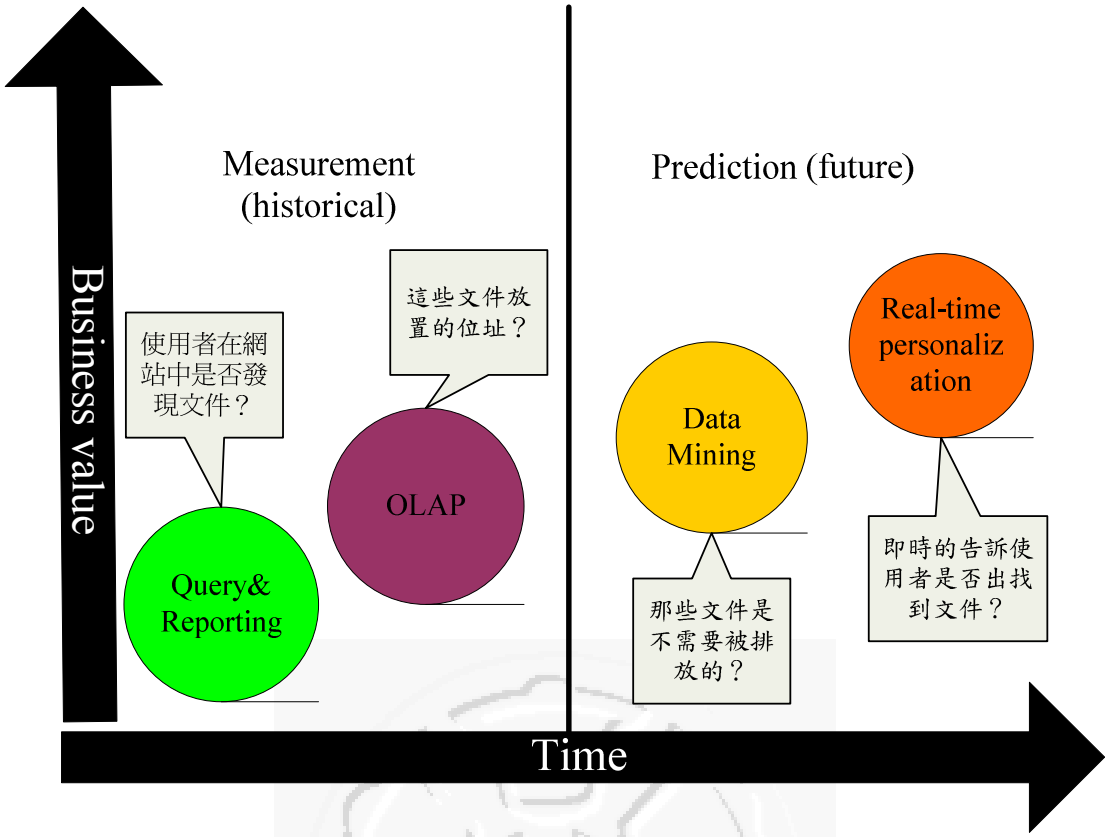
以上幾項步驟，多數學者則稱為資料倉儲四步驟，透過以上四項資料前處理(Preprocessing)，則可有效排除資料的不完全性，提升預測模型的效率及準確度。由上述資料的準備過程，本研究以圖形化的方式來加以描述。如下圖 2-1 所示。



資料來源：本研究自行整理

圖 2-1 資料探勘前處理示意圖

由以上文獻之分析及探討，我們更可以知道資料探勘在任何領域之定位及扮演何種角色。Fayyad（1996）認為資料庫知識發現流程是說明在資料庫中取得知識的過程，因此對於不確定、不完整及有干擾性的資料要預先處理，而資料探勘只是資料庫知識發現流程中的一步驟。從資料庫中找尋知識，不但結合了資料庫、知識庫、而且也包括人工智慧、統計學等方面相關的應用。透過知識的探勘，從龐大的資料庫中探勘出有意義的資訊，以幫助決策者做最有利的決策。由此可知資料探勘之技術，可因研究目的的不同來做不同定位的表現，就如本研究網路使用者是否找尋到文件為例，如下圖 2-2 所示。



資料來源：本研究自行整理

圖 2-2 資料探勘定位圖

四、資料探勘的各項任務與技術

(一). 資料探勘各項任務

資料探勘是利用統計與人工智慧的演算法，從龐大的歷史資料中，找出隱藏的規律及建立精準的模型，用以預測未來，在商業上可有效提供市場行銷以及顧客管理所需根據之規則型態，根據Berry and Linoff (2000)在“Mastering Data Mining“一書中，可將資料探勘區分為六大任務。

1. 分類(Classification)

由目前的分類推論出一套最有效的規則來識別群體的類型，從已知類別的物件集合中，依據其屬性(可能影響物件類別的變數)建立一個分類模式(如決策樹或決策法則)來描述物件屬性與類別之關係，再根據這些特徵，對其他未經分類或是新的資料作預測，分

類的目的是產生一個分類模型來描述物件屬性與類別之間的關係，事實上，分類不光是提供單純「會」或「不會」的答案而已。還可以透過分類結果的純度(Purity)來提供分類機率，這些分類機率可以提供我們作為名單排序或是篩選顧客時的重要依據。而分類模型最常使用的表示法包括類神經網路、決策樹(Decision Tree)、樹歸納法(Decision Tree)、決策法則(Decision Rules)及規則法則歸納法(Rule Induction)。

2. 推估(Estimation)

推估與分類的差異在於分類問題是預測「類別變數」，而推估問題主要是用來推估「連續變數」。透過輸入資料我們可以用來推估一些未知的連續性變數，在商業上的應用例如信用額度等。

3. 預測(Prediction)

預測與分類和推估相當接近，其差別在於預測是去推估未來的數值及趨勢，以歷史資料來預測未來的走向，例如根據過的課程開設來預測未來課程數量的趨勢是增加還是減少，藉以做為調整教育資源分源參考資料。

4. 群集化(Clustering)

利用一些特性的組合將這些物件分成集群的過程，根據物件的相似性(或不相似性)，將這些物件分成群集，使得每個群集內的成員具有高度的相似性，而不同群集間之物件具有高度的不相似性。但是與「分類」不同的是，群集不像分類問題中有那麼明確的分析標的，必須透過各產業領域知識專家或是借用其他像敘述統計或線上分析處理的工具，才能進一步找出這群相近的顧客中，有哪些共同的特性。當中在群集化分析常用的演算法包括：

K-means、EM以及自我組織映射圖網路(SOM, Self-Organization Map)。

5. 關聯分組(Affinity Grouping)

關聯分組就是從歷史資料中，找出哪些物件/事件總是相伴發生，在大多數的時候我們會稱之為關聯規則或者是購物籃分析。在關聯規則當中最有名的案例，就是NCR為美國最大的連鎖零售商Wal-Mart所分析的案例，他們發現在星期四的晚上，大多數購買尿布的消費者，也會一併購買啤酒。這個規則之所以有名其實是在於它突顯出利用資料探勘可以發覺出許多利用傳統行銷人員經驗所無法找出的規律，而事後的研究發現，這些同時購買尿布與啤酒的人，大多數是年輕的夫婦，趁週四買週末狂歡要用的啤酒時，也順便購買安置家中幼兒的尿布。關聯分組可以成功的找出事物之間隱藏的關聯性，而找出這種關聯性有助於我們研判針對單一事物做改變時所帶來的整體影響。

6. 描述(Description)

描述是資料探勘過程中的重要附加價值的一種，通常在進行資料探勘的前置作業時透過資料視覺化，能夠將資料特性呈現在資料探勘者的面前。人的眼睛有時候是最好的資料探勘工具，透過良好的資料視覺化程序，我們可以發覺許多演算法無法判別的規則型態。

(二). 資料探勘各項技術

在資料探勘領域中最普遍使用的幾項技術，如表 2-5 所示(Berry, et al., 1997)。

表 2-5 常見的資料探勘技術整理表

技術	描述
關聯分析	「同質分組」的一種形式，著眼點在於找出可以一起販售的商品組合，它能顯示商品組合的售出率有多高並且形成規則。當交易是非匿名時，它可以加上一個模組進行跨時性分析。
記憶基礎理解	比對新資料和現存資料庫原有資料的相似度，以決定新資料的類別或預測值的工作。它是以距離函數判斷資料間的相似度，再以數個最相似的資料投票來決定歸類結果。
群集偵測	針對要分析的資料，利用幾何學、統計、類神經網路等方法，將資料分成多個群內同質、群間異質的群組，使各群組的特徵能有效突顯出來。
連結分析	搜尋資料間的關聯，並且據此發展出分析模型，這是圖像理論在資料探勘中的應用。
決策樹	主要用在資料分類上，屬於監督式的資料探勘方法。它能將訓練資料集的紀錄區分為獨立的子群，每一子群都有自己的規律，彼此是互斥的。同時在樹的發展過程中，獲得清楚易懂的分類規則並找出關鍵屬性。
類神經網路	一種平行分散式的計算模式，以大量簡單的相連人工神經元，模仿生物神經網路的資訊處理系統，使電腦能夠模擬人類的神經系統結構，進行資料的處理。

資料來源：本研究自行整理

第二節 資料倉儲

一、資料倉儲之定義

資料倉儲 ，由資料倉儲之父 W.H.Inmon 於 1990 年提出，主要功能乃是將組織透過資訊系統之線上交易處理(OLTP)經年累月所累積的大量資料，透過資料倉儲理論所特有的資料儲存架構，作一有系統的分析整理，以利各種分析方法如線上分析處理、資料探勘之進行，並進而支援如決策支援系統(DSS)、主管資訊系統(EIS)之建立，幫助決策者能快速有效的自大量資料中，分析出有價值的資訊，以利決策擬定及快速回應外在環境變動，幫助建構商業智慧(BI)。

一般來說，資料倉儲可由關聯式資料庫，或專為資料倉儲開發的多維度資料庫建立，若由關聯式資料庫建立而成，其架構可分為星狀及雪花狀架構，包含數個維度資料表及一個事實資料表，若以規模分類則又可分為資料倉儲及資料超市。

資料倉儲的建制不僅只是資訊工具技術面的運用，在規劃和執行面更需對產業知識、行銷管理、市場定位、策略規劃等相關條件有深入的了解，才能真正發揮資料倉儲以及後續分析工具的價值，提升組織競爭力。

資料倉儲就是將現有資料、歷史資料、外部資料等經過有系統的整合，把資料選取、分類、轉換達到最佳化後，融入一個設計好的關連式資料庫中，以利資料分析者存取，大量改進效率。Data Mining 及 OLAP 都是由資料倉儲中取出資料而進行分析。因不同研究者的研究重點及動機不同而有不同的看法，整理如表 2-6 所示。

表 2-6 資料倉儲定義整理表

作者(年代)	內容綱要
Inmon(1996)	「資料倉儲是以主題導向、整合性、非揮發性及時間變異性所聚集之資料，以支援管理決策。
Kimball(1996)	資料倉儲是一群靜態的資料，由日常的交易系統中取得，另外形成一個倉儲，因此資料倉儲可說是支援決策的資料庫。
Hoven(1998)	資料倉儲是一套經過改良的決策支援系統，它產生高階的、整合的、系統的、結構化的資料，使其可以被解釋、查詢、報告、分析以協助商業的決策
Murtaza(1998)	資料倉儲可以產生整合性、非揮發性的組織資料，並轉換成提供決策支援的資訊。
Lewis，Bernstein & Kifer (2001)	通常是非常大型的，資料的取得包括了數個不同時間的資料庫來源。

資料來源：本研究自行整理

由此可知我們知道，資料倉儲是一種資訊系統的資料儲存理論，此理論強調利用某些特殊資料儲存方式讓所包含的資料特別有利於分析處理，以產生有價值的資訊。利用資料倉儲方式所存放的資料，具有一但存入，便不隨時間而更動的特性，同時存入的資料必定包含時間屬性，通常一個資料倉儲皆會含有大量的歷史性資料，並利用特定分析方式自其中發掘出特定資訊。

二、資料倉儲的特性

資料倉儲大師Bill Inmon(1996)在「建構資料倉儲（Building the Data Warehouse）」一書中之描述，資料倉儲包含以下四個主要特性：主題導向、整合性、非揮發性及時間變異性，以下分別說明其意義(Kimball, 1996)。

(一).主題導向(Subject Oriented)

資料倉儲會將資料自然的以相同的種類或主題聚集在一起，因此它會以這些高層次且不重複的主題為主要的處理對象，而有別於交易作業系統的流程導向。資料倉儲所欲解決的問題是決策分析的問題，而非交易導向的問題。

(二).整合性(Integrated)

資料倉儲內的資料必須具有相當整合性，在一企業中，具有多個資料庫或不同的系統平台是普遍的事。而資料倉儲便是要整合企業的多個資料庫，跨越不同的平台，透過資料轉換過程，要讓欄位名稱、測量變數、編碼方式、日期時間等等主題屬性具有一致性的格式。

(三).非揮發性(Non-Volatile)

當資料放到資料倉儲中後，便不易異動、修正或更新。換個方式說，資料一旦被新增之後，便難以被更動，只會被查詢，因此它具有唯讀的性質和累積性。

(四).時間變異性(Time Variant)

日常的作業系統每天都有新資料增加，為維持資料倉儲的可用性，必須在某些特定的時間點到作業系統中擷取新資料，這樣才能確保資料倉儲中的資料是最具時效性的。

簡單地說，資料倉儲本身是一個非常大的資料庫，它儲存著由組織作業資料庫中整合而來的資料，特別是指從線上交易系統（On-Line Transactional Processing，簡稱 OLTP，處理一般交易的線上資訊系統）所得來的資料，另外還包括少部分相關的外部資料。

而企業可以以顧客為中心來整合資料，來加以分析顧客關係管理的相關主題，如顧客流失問題、顧客分群問題、顧客利潤問題、關聯分析、交叉行銷問題等。資料倉儲可以作為資料探勘和 OLAP 等分析工具的資料來源，由於存放於資料倉儲中的資料，必需經過篩選與轉換，因此可以避免分析工具使用錯誤的資料，而得到不正確的分析結果。

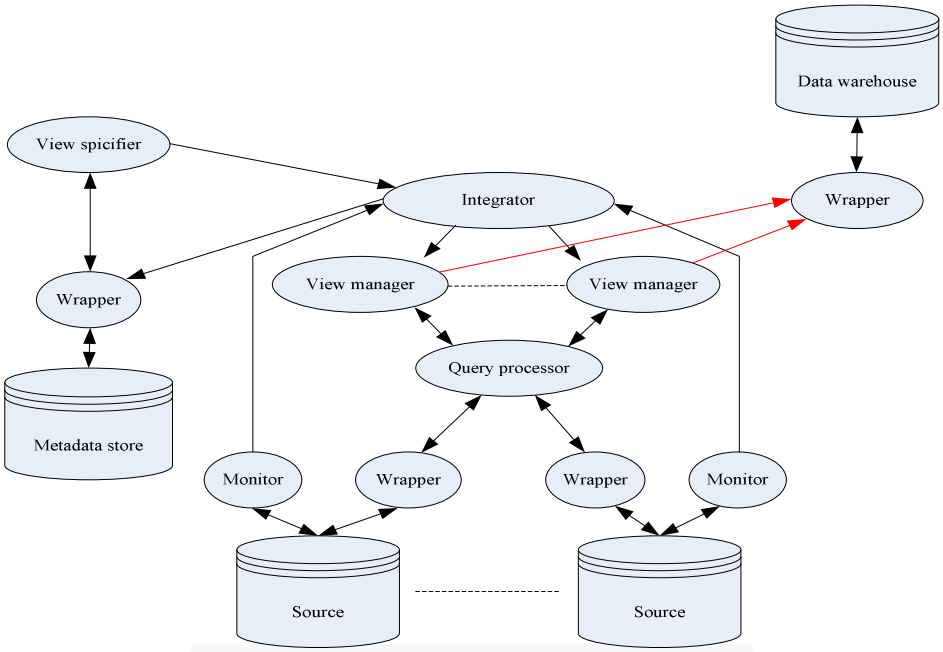
資料探勘和 OLAP 同為分析工具，其差別在於 OLAP 提供使用者一便利的多維度觀點和方法，以有效率的對資料進行複雜的查詢動作，其預設

查詢條件由使用者預先設定而資料探勘，則能由資訊系統主動發掘資料來源中，未曾被查覺的隱藏資訊，和透過使用者的認知以產生知識。

三、資料倉儲的架構

資料倉儲為整合性資訊的儲存體，其內部的資訊可供查詢及分析。資料倉儲可由多個分散式、自主性、異質性的資料來源中搜集並維護相關的資訊。當來源(Source)初始化或其中資料被修改時，對應的資料可自 Source 擷取、處理並轉換成內部型態後，再與已存放於資料倉儲中的資訊互相整合。在資料倉儲中，因為資訊已事先存於系統中，查詢及資料分析皆可以有效率反應（Berson, Smith, & Thearling, 1999）。

資料倉儲主要的概念是輔助高階查詢擷取、篩選、整合相關資訊。相對於傳統系統的被動式查詢(當需要查詢時才做運算)，資料倉儲使用主動式查詢(當來源更動時，即作出相對應的反應)，因此執行查詢時並不需要在 Data Source 處重新計算，而是直接由資料倉儲中取出資訊(Jiawei, & Micheline, 2001)。也正因為資料倉儲中的資訊已存放在系統中，所以 Data Source 和系統資訊間存在 Consistence 的問題。下圖 2-3 為資料倉儲系統架構圖。



資料來源：(Jiawei, & Micheline, 2001)

圖 2-3 資料倉儲系統架構圖

另外，建構資料倉儲包含三個部份(Chaudhuri, & Dayal, 1997)：

(一)、Business Analysis and Modeling

此部份是資料倉儲的顧問主要工作項目。必須了解每種不同產業的競爭環境、企業價值 ，進而塑模出有利於企業的 Model 來分析。當然，也必須了解資料倉儲的設計方式，例如：Star Schema。如果只是先分析外在資料，再用工具單純抓進來組 Star Schema，對資料倉 儲的價值不大，所以每個人應多了解產業競爭結構，企業內的流程，組織的管理與決策， 對此方面的設計才有助益。

(二)、ETL (Extraction、Transformation、Transportation、Loading)

此方面比較偏技術，通常是工程師來做。

(三)、DWA (Data Warehouse Administrator)

類似 DBA 的管理，例如資料庫存取處理、平行處理、查詢最佳化、校能調效、資料庫控制。

四、資料倉儲之優缺點比較

藉由以上對由資料倉儲技術深入的探討，本研究針對其優缺點做一比較表，如下表 2-7 所示。

表 2-7 資料倉儲優缺點比較表

有形優點
◆ 降低對電腦主機的依賴，並有效降低企業成本。
◆ 整合企業各部門的資料及減少作業成本。
◆ 促使企業內部各種規格的文件趨向單一格式，便於儲存及查閱。
無形優點
◆ Data Warehouse 的資料來源可由各個部門取得，因此可充份運用組織資源。
◆ 確保日常作業中的資料不受干擾。
◆ 提昇企業的效率與生產力。
◆ 可充分運用組織所操控的資訊，獲得顯著之競爭優勢。
◆ 提供更快的資訊以供企業決策，並加強決策支援、加快決策速度、並增加所作決策的品質。
◆ 對於需要時效性的企業問題，較能提供即時且有效的資訊。
◆ 由於資料可信賴，使得企業的模擬假設更接近事實。
◆ 使用者不知道資料存放的處所，可由一個單一且統一的查詢界面取得所需的資訊。
◆ 可由 Data Warehouse 中的企業歷史資料中，預測未來趨勢與結果，並可從事策略性的重大決策。
缺點
■ Data Warehouse 的開發往往是一項極為龐大的投資。開發單一的 Data Warehouse 不但風險高，而且須投入大量的企業資源。
■ 開發的時間較傳統的系統開發時間長。

資料來源：本研究自行整理

五、資料倉儲的種類

現階段的資料倉儲可供選擇的種類有(Ponniah, 2001)：

(一)、企業資料倉儲(Enterprise Data Warehouse)

此一資料倉儲系統乃是企業資料最核心的處理部份，所有的資料將被視為共同資產來管理，由企業統合規劃，並藉由應用軟體來分離或共享。

(二)、企業資料超市(Data Mart)

此資料倉儲所涵蓋的範圍比資料倉儲小，其資料的涵蓋範圍為企業的

部份資料。事實上，此種倉儲模式為資料倉儲的一個子集，一般僅提供部份資訊，給某一群使用者或某一部門使用，以符合企業內部的特殊需求。

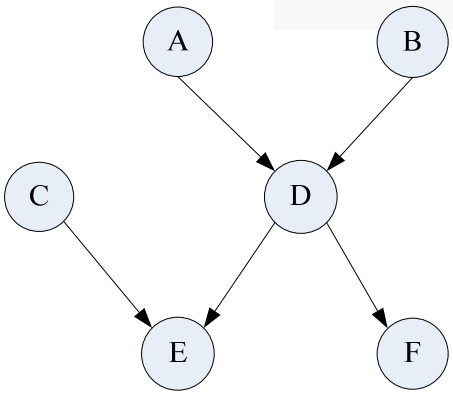
(三)、虛擬資料倉儲(Virtual Data Warehouse)

Virtual Data Warehouse 不另建特殊的資料庫，而是使用現有的資料庫，並輔助以一些特殊資料倉儲軟體、硬體工具，來進行有效資料處理。即利用一些中介工具，來存取遠端的實體倉儲。因此建構快速、風險低，可做到即時的資料有效性。

第三節 貝氏網路

貝氏網路為一種機率性專家系統，其型態為圖形模式(Graphical Modal)，可說明變數間的機率關係。而一般常見的所謂機率網路(Probabilistic Network)、因果網路(Causal Network)與知識地圖(Knowledge Map)等等均是貝氏網路的別名。圖形模式結合統計方法後，經常運用在資料分析問題上。

一、貝氏網路的基本架構型態



A	T	F
	0.6	0.4

B	T	F
	0.2	0.8

C	T	F
	0.34	0.66

D	F	
	T	F
T	0.18	0.82
F	0.78	0.22

A	B	D	
		T	F
T	T	0.2	0.8
T	F	0.55	0.45
F	T	0.7	0.3
F	F	0.37	0.63

C	D	E	
		T	F
T	T	0.5	0.5
T	F	0.85	0.15
F	T	0.1	0.9
F	F	0.77	0.23

資料來源：本研究自行整理

圖 2-4 貝氏網路基本架構及機率表

圖 2-4 表示貝氏網路含有 A、B、C、D、E、F 等六個節點，且每個節點皆為二元節點(Binary Node)；即各節點的狀態(State)均為(True , False)。而圖 2-4 中，右方的機率表格為條件機率表(Condition Probability Table；CPT)。節點 A 的兩個狀態(True , False)的機率別為(0.6 , 0.4)，因節點 A 為根源因子(Root Cause)，所以並無包含其父代節點(Parent Node)的條件機率表。節點 E 的父代節點為 C 與 D，故節點 E 有包含其父代節點的條件機率表。而該機率表表示的意涵，舉例來說，當節點 C 的狀態為 T，節點 D 的狀態為 F 時，節點 E 的狀態為 T 的機率為 0.85 節點 E 的狀態為 F 的機率為 0.15。

貝氏網路具有以下幾點之性質(Lauritzen, et al.,1994)：

(一). 貝氏網路是由一群隨機變數當作網路節點所組成。

(二). 有關係的任一對網路節點間皆以有方向性的連結或箭號所串聯。

(三). 每一個節點均有一個條件機率表。

(四). 以定量的方式說明該節點之父代節點對此節點的影響。

(五). 網路中不允許有方向性循環(Directed Cycles)的存在。

具由文獻之整理，本研究歸納出貝氏網路之四項優點(David, 1995; Shachter, 1988)：

(一). 貝氏網路可以輕易的處理不完整資料(Incomplete Data)的問題。貝氏網路在建構時，使用者可以主觀的設定未知的參數值，如事前機率(Prior Probability)等。因此若建構網路時資料不完整，使用者可主觀的填入資料彌補資料的完整性。

(二). 貝氏網路可以允許使用者對因果關係進行學習。學習能力是貝氏網路一項極重要的能力，特別是當網路建構者對欲求解問題的領

域知識不足時。

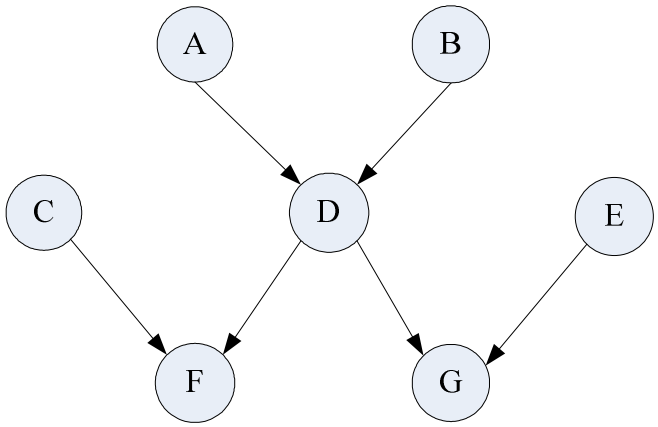
(三). 貝氏網路可以結合貝氏統計方法輕易的結合領域知識(Domain Knowledge)與資料。在現實世界建立系統時，系統建構者很重視關於問題的領域知識，特別是關於此問題的資料量很龐大或昂貴時。貝氏網路將關於問題的知識以因果關係表示之，並以機率表示因果關係的強度。因此我們可以貝氏統計方法結合問題的領域知識與資料。

(四). 貝氏網路可以結合其他型態的模式提供有效且具原則性的方法，以避免某些特殊的例外資料所造成的誤差。

貝氏網路為圖形模式(Graphical Model)的一種，所以是 Variable Space 的模型。網路所描述的問題發生結構上或參數的變化時，可以只對有變動的部分做修正，不需考慮其他沒有變動的部分。

二、貝氏網路的數學模式

貝氏網路由問題的領域知識與資料所構成，因此若以 B 表示貝氏網路(Enrique, et al,.1997)，則 B＝（D，P）。其中 D 表示一有方向性的非循環圖形(Directed Acyclic Graph；DAG)，即表示問題的領域知識。如圖 2-5：



資料來源：本研究自行整理

圖 2-5 簡單之貝氏網路結構示意圖

P 為 DAG 中某一變數的聯合機率分配(Joint Probability Distribution；JPD)，即表示關於問題的資料。如下：

(一)、 首先根據訓練樣本，使用貝氏機率法則，計算各屬性相對不同分類結果的條件機率：

$P(C = c_j \mid A_1 = a_1, A_2 = a_2, \cdots, A_n = a_n)$

貝氏定理：$P(c_j \mid a_1, a_2, \cdots, a_n) = \dfrac{P(a_1, a_2, \cdots, a_n \mid c_j)p(c_j)}{P(a_1, a_2, \cdots, a_n)}$

$= P(a_1, a_2, \cdots, a_n \mid c_j)p(c_j)$.....（式1）

屬性獨立：$P(a_1, a_2, \cdots, a_n \mid c_j)p(c_j) = \prod_{i=1}^{n} P(a_1 \mid c_j)$.....（式2）

(二)、 應用樸素貝氏分類，進行預測新訓練樣本所應歸屬的類別C_{NB}可表示如下：

$$c_{NB} = \arg_{c_j = C^{\max P(c_j \mid a_1, a_2, \cdots, a_n)}}$$

$$= \arg_{c_j = C^{\max P(c_j) \prod_{i=1}^{n} P(a_1 \mid c_j)}} \quad \text{.....（式3）}$$

其中，

a_n：表示與目標值相關聯的屬性值

c_j：表示目標屬性

$p(c_j)$：表示目標屬性值的機率

$P(a_1, a_2, \cdots, a_n \mid c_j)$：表示每項屬性值在目標屬性$c_j$中出現的機率

若訓練資料嘗試推算兩個條件機率$P(a_1, a_2, \cdots, a_n \mid c_j)$與$p(c_j)$，假設有n項與目標值相關之屬性值，則就有n(n-1)項關聯性需要考慮。假設與目標值相關聯的屬性值，彼此為相互獨立，則從訓練樣本中選取的目標值，即為目標值的機率c_{NB}，而c_{NB}即為個別屬性值機率的相乘積。

第四節 派翠西網路

Petri Nets在1962年由Petri1所提出（Petri, 1962），它可以做為很多系統的圖形及數學塑模(Modeling)工具，尤其是對於分析和描述具有同步的（Concurrent）、分散式的（Distributed）、非定性的（Nondeterministic）、動態的、非同步（Asynchronous）或隨機（Stochastic）等性質的資訊系統（Jefferson,1985；Flanagan,& Qadeer, 2002）。以一個圖形工具而言，Petri Nets如同流程圖（Flow Charts）及網路（Networks）一樣可以用來作為傳達視覺化訊息（Visual Communication）。

一、基礎派翠網路

以較為嚴謹的方式來定義派翠網路，它是一個具有方向性的加權雙分圖（Directed Weighted Bipartite Graph），由兩種節點所組成：一種稱之為位置（Place），另一種稱之為轉換（Transition）。位置與轉換之間可有箭頭加以連接，箭頭的方向可能由位置指向轉換或由轉換指向位置。一個派翠網路是由五個部分所組成(5-Tuple)， $PN=(P,T,F,W_0)$正式的定義如下（Murata, 1989）：

$P= \{p_1, p_2,..., p_m\}$ 是一組有限(Finite)的places 集合(Set)；

$T= \{t_1, t_2,..., t_n\}$ 是一組有限的Transitions 集合；

$F \subseteq (P \times T) \cup (T \times P)$ 是一組Arcs 集合（Set）（流程關係）；

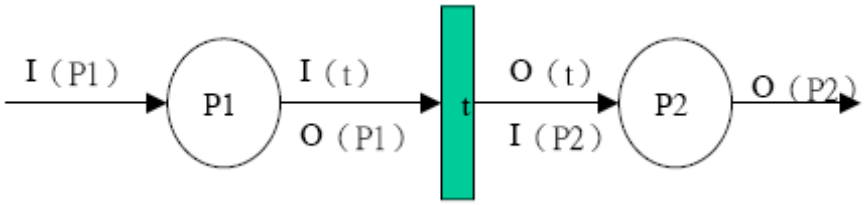
$W：F \rightarrow \{1,2,3,...\}$ 是一個權重函數（Weight Function）；

$M_0：P \rightarrow \{0,1,2,3,...\}$ 初始(Initial) P_N呈現的（Marking）；

其中$P \cap T=\varnothing$ and $P \cup T \neq \varnothing$

I為輸入函數，$I : T \rightarrow P^{\infty}$或$I : P \rightarrow T^{\infty}$，表示由轉變輸入位置的集合或由位置輸入轉變的集合；O 為輸出函數$O：T \rightarrow P^{\infty}$或$O：P \rightarrow T^{\infty}$，表示由轉變

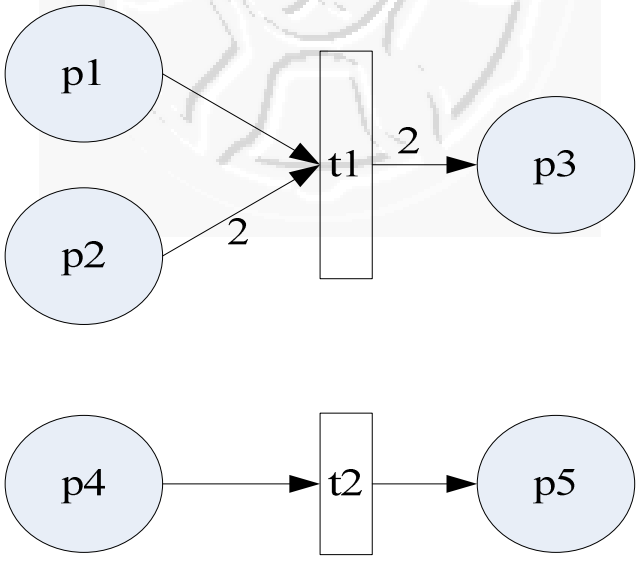
輸出位置的集合或由位置輸出轉變的集合；(P∞為位置群組；T∞為轉變群組)。上述之關係可由下圖2-6 表示：



資料來源：本研究自行整理

圖 2-6 Petri Net 元件之關係

Petri Net 的表示法可分為結構表示法及圖形表示法，其各組成元件之圖形表示已如上述，以下說明二種表示法之關係。若有一Petri Net，其結構表示式為I(t1)={P1 ,P2 ,P2}，O(t1)={P3 ,P3}，I(t2)={P4}，O(t2)={P3 ,P5}，則其圖形表示如下圖2-7 所示。其中，有向弧之權數代表轉變所需的位置數。例如，P2 至t1 權數為2 代表t1 至少需2 個記號在P2；t1 至P3 權數為2 代表t1會產生2個記號在P3省略即代表權數為1。

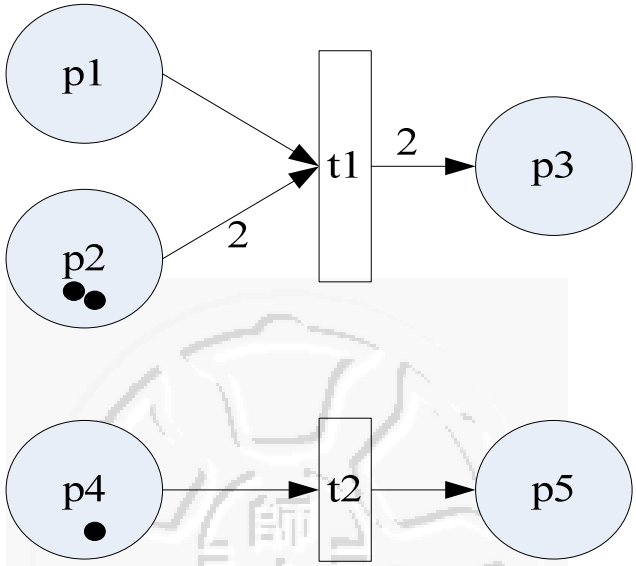


資料來源：本研究自行整理

圖 2-7 基本派翠網路架構圖

在Petri Net 中，若一轉變t 的所有輸入位置，至少都具有與其有向弧(位置到轉變)加權數相等的記號時，則稱t 處於致能狀態。處於致能狀態的

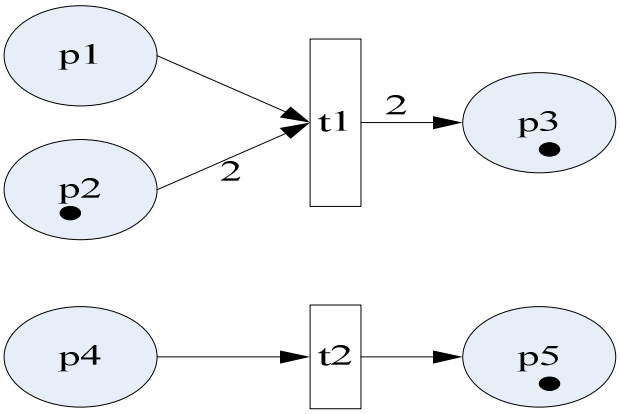
轉變可以被激發，當致能被激發時，將從t 的每一個輸入位置中移走記號，並產生新的記號至輸出位置中。至於移走與產生的記號數目，則與相對應的有向弧權數相等。如圖2-8，t2 處於致能狀態，而t1 則否，因為t1 須有一個P1 以及二個P2 才能致能，而圖2-8 中之P1 並沒有記號存在，因此，t1 必須等待。



資料來源：本研究自行整理

圖 2-8 Petri Net 致能實例

圖2-9 為t2 激發後的情形。t2 被激發後，產生新的記號至其輸出的位置。因其有二個輸出位置，P3 及P5，且其權數均為1，因此，分別產生一個記號至P3 和P5 中。



資料來源：本研究自行整理

圖 2-9 Petri Net 激發實例

本研究將使用Petri Net 模擬使用者在網站所做的動作路徑行為，以圖形化方式讓管理者清楚了解使用者於網站內所找尋文件及文件上傳的行為。

第五節 關聯分析法

關聯式法則可由大量資料中找出關聯性較高的資訊，其主要目的在於找出項目間的關聯規則。假設A、B為資料庫中的屬性集合，則項目集合AB其支持度為A、B同時出現的交易筆數佔資料庫所有交易筆數的比率。信賴度則為A、B同時出現的交易筆數對只包含A的交易筆數之比率，當項目集於交易記錄中出現頻率高於最小支持度與最小信賴度門檻值即為關聯規則，而支持度與信賴度之定義如下：(Agrawal, et al , 1993)

$$\text{支持度} = \frac{\text{交易記錄中同時出現屬性集合}A\text{與屬性集合}B\text{的數目}}{\text{資料庫中所有交易筆數}} = \frac{\#(A \cap B)}{N}$$

$$\text{信賴度} = \frac{\text{交易記錄中同時出現屬性集合}A\text{與屬性集合}B\text{的數目}}{\text{交易記錄中包函屬性集合}A\text{的次數}} = \frac{\#(A \cap B)}{\#A}$$

以下介紹關聯式資訊探勘技術常用的Apriori 演算法(Agrawal & Srikant 1994)。

一、Apriori演算法

Apriori(Agrawal & Srikant 1994)是關聯法則裡最基本的演算法，將資料庫中的交易屬性形成項目集，檢驗項目集是否通過最小支持度及最小信賴度門檻值。Apriori 演算法依序組合項目之間關係形成規則，以表2-8 購買項目為例並建構關聯規則。首先掃描資料庫計算各屬性出現次數並設定最小支持度為30%及最小信賴度設定為60%，產生候選項目集(Candidate

Itemset)，並存入C1 表格中，接著計算候選項目集，找出通過門檻值的大項目集(Large Itemset)，存入L 1 表格中。以相同方法組合產生候選2-項目集，儲存於表格C2，再掃描資料庫，判斷候選2-項目集是否通過使用者訂定之最小支持度與最小信賴度門檻值，將符合之候選2-項目集形成真正的2-項目集，儲存於表格L2 中。進一步結合2-項目集產生候選3-項目集，再掃描資料庫判斷候選3-項目集是否通過門檻值。依相同方式結合及掃描資料庫至無法再產生更高項目集為止。當項目集無法再結合產生更高次項的候選項目集時，必須存入規則表格中，而組合產生的更高次項候選項目集若無法通過門檻值成為大項目集時，仍需退回前項目集，加以組合成規則並存入規則表格中，重覆此一程序直到找出所有規則為止。如下圖2-10 所示。

表2-8 關聯分析法之例子

Transaction	Items Purchased
1	(Orange Juice、Soda)
2	(Milk、Orange Juice、Window Cleaner)
3	(Orange、Juice、Detergent、Soda)
4	(Window Cleaner、Soda)
5	(Soda、Potato Chips)

資料來源：本研究自行整理

Step 1: Generate large1-itemsets

All1-itemsets(C1)

1-Itemset	Support
Orange Juice	60%
Soda	80%
Milk	20%
Window Cleaner	40%
Detergent	20%
Potato Chips	20%

Large1-itemsets(L1)

1-Itemset	Support
Orange Juice	60%
Soda	80%
Window Cleaner	40%

Step 2: Combine cadnidate2-itemsets

Candidate2-itemsets(C2)

1-Itemset	Support
(Orange Juice,Soda)	
(Orange Juice,Window Cleaner)	
(Soda,Window Cleaner)	

Step 3: Generate large3-itemsets

Candidate2-itemsets(C2)

1-Itemset	Support
(Orange Juice,Soda)	40%
(Orange Juice,Window Cleaner)	20%
(Soda,Window Cleaner)	20%

Large2-itemsets(L2)

1-Itemset	Support
(Orange Juice,Soda)	40%

Step 4: Construct association rules from large 2- itemsets

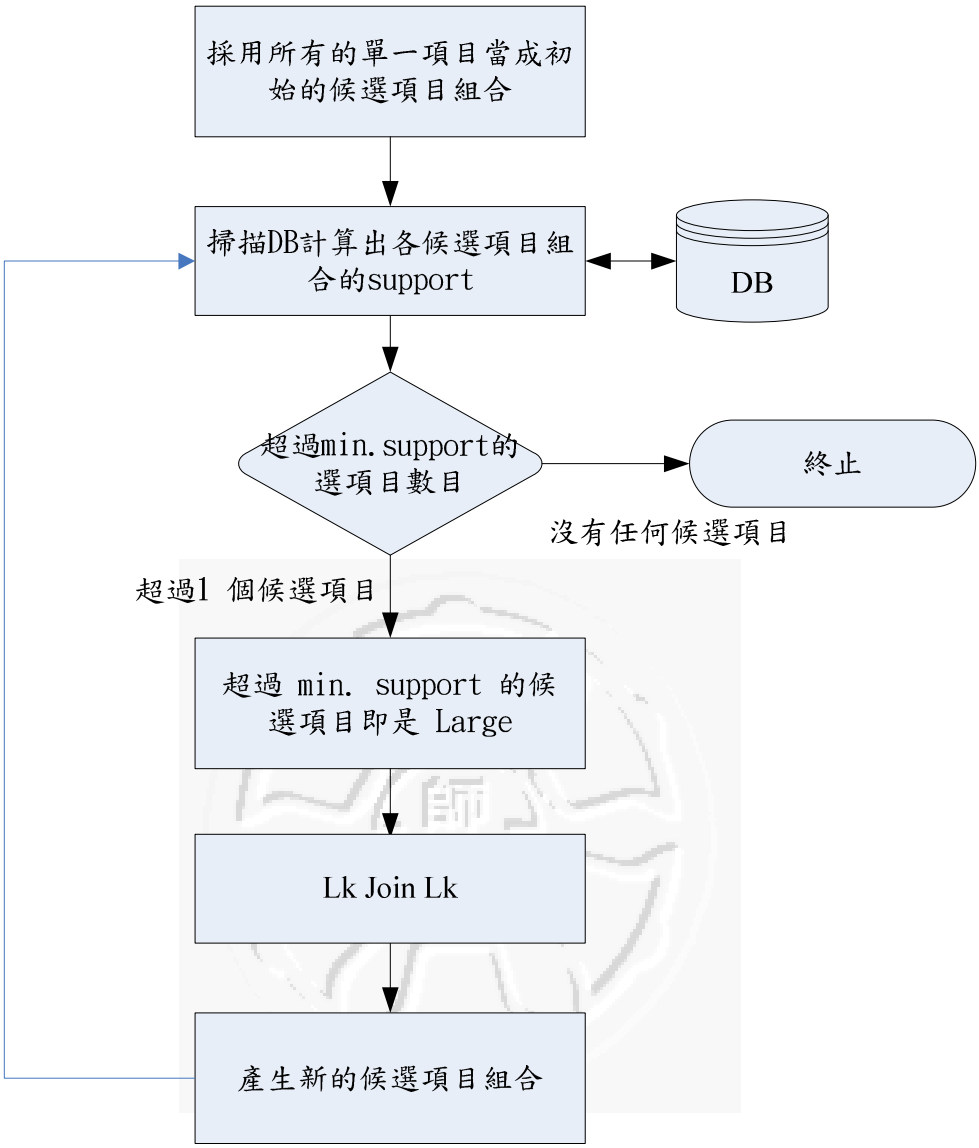
Orange Juice→Soda(confidence=66.67%)

Soda→Orange Juice(confidence=50%)

An association rule is generated : Orange Juice→Soda(confidence=66.67%)

圖2-10 關聯法則示意圖

由上圖2-10 本研究規劃出關聯規則法之流程圖，如下圖2-11 所示。



資料來源：修改自Agrawal & Srikant 1994

圖 2-10 為關聯法則之流程圖

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。但與Apriori 的想法基本上還是一樣的。以下則為關聯演算法則。所下表2-9 所示。

表2-9 關聯規則演算法

insert into C_k
select p.litemset$_1$, p.litemset$_2$,…, p.litemset$_{k-1}$, q.litemset$_{k-1}$
from L_{k-1} p, L_{k-1} q
where p.litemset$_1$=q.litemset$_1$,…, p.litemset$_{k-2}$=q.litemset$_{k-2}$;
for all sequences c∈ C_k **do**
  for all (*k-1*)-subsequences s of *c* **do**
    if ($s \notin L_{k-1}$) **then** **delete** *c* from C_k;
L_1 = {large 1-sequences}; // Result of the phase
for ($k=2$; $L_{k-1} \neq \Phi$; k++) **do**
  begin
  C_k = *New candidate generate from* L_{k-1}
  foreach customer-sequence c in the database **do**
    Increment the count of all candidates in C_k that are contained in *c*
L_k = Candidates in C_k with minimum support.
End Answer=Maximal Sequences in $U_k L_k$;
L_k the set of all large k-sequences
C_k the set of candidate k-sequences

資料來源：(Savasere 1995)