中山管理評論　1998年秋季號
第六卷第三期　pp.903-946

先前知識導向的資料發掘

The Prior Knowledge Approach to Data Mining

楊亨利　*Heng-Li Yang*
國立政治大學
National Chengchi University

林幸怡　*Hsing-Yi Lin*
運康科技公司
Infocomm Corporation

摘　要

資料發掘乃由資料庫中發掘非顯然的、隱含的、前所未知的，而可能有用的資訊的過程。本文首先從文獻中了解目前資料發掘領域的研究現況，從而由擴充先前知識的角度切入，利用企業法則、延伸的實體關係模式中的一般化、集合化、聚集化等抽象化觀念、延伸之資料字典及經驗法則等先前知識得出更合適的資料以供資料發掘，並對於概念樹導向歸納學習法做適當的修改，提出研究架構。並以假想的學校資料庫，發展出一套雛形系統，驗証本架構的可行性。最後並提出進一步的研究建議，以供後續研究參考。
關鍵詞：資料發掘、知識發現、先前知識、實體關係模式、資料抽象化

Abstract

Data mining is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. In the paper, we first survey previous research in data mining and discuss their prior knowledge. Then we propose a framework to extend prior knowledge including data abstractions (generalization, association, and aggregation) of the extended entity-relationship model, business rule, extended data dictionary and heuristics, in order to assist the process of data

mining. The Han's algorithms based on inductive learning using concept hierarchy trees are slightly modified. A prototype based on an artificial university database is then reported. Finally, we give some suggestions for future research.

Key Words: Data mining, Knowledge Discovery, Prior Knowledge, Entity-Relationship Model, Data Abstractions

壹、導　論

根據 Frawley., Piatetsky-Shapiro 與 Matheus 的定義 (Frawley et al., 1991)，知識發現 (Knowledge Discovery) 爲由資料庫中，發掘非顯然的、隱含的、前所未知的而可能有用資訊的過程。Grupe 與 Owrang (1995) 認爲資料發掘 (Data Mining) 是指由已存在的資料中發掘出新事實及發現專家尚且不知的新關係。而 Fayyad (1996) 則認爲資料庫中的知識發現爲指自資料庫中選擇合適資料、先前處理過濾、轉化與減化資料發掘至結果評估之一連串過程其定義實完整地包含一般所談的資料發掘之前後步驟。目的乃希望能由豐富的資料庫之中，幫助人類做資料的歸類或分析動作，從中找出企業未知的現象及關係，讓管理者得以充分利用手邊資訊，了解企業問題，提出改善、因應之道。

由於在廣大的資料海中發掘資料，本爲十分困難的動作，再加上使用者可能本身即對其需求不甚了解，因此不斷的試誤過程，可預想得知。如此不僅浪費使用者時間，也十分耗費系統資源。因此，本研究迥異於過去研究著重演算法效率提升，轉而希望能運用較多的先前知識，以幫助使用者確立問題，有效減少其試誤機會，並使得到的發掘結果合乎需求。

本文將提出一個利用先前知識輔助資料發掘過程的方法架構，探討先前知識之種類及使用時機，輔助資料發掘過程，以補其它方法之不足之處。同時，本文並將報導一個運用所提出的架構以進行實作之雛型。

貳、文獻探討

在資料發掘的文獻中，包含如圖 1 之五大類，即經「使用者溝通界面」自使用者處了解需求，由「資料庫」中獲取資料，並加入「應用領域知識」，經過「資料發掘方法」而「發掘出知識」的一連串過程。

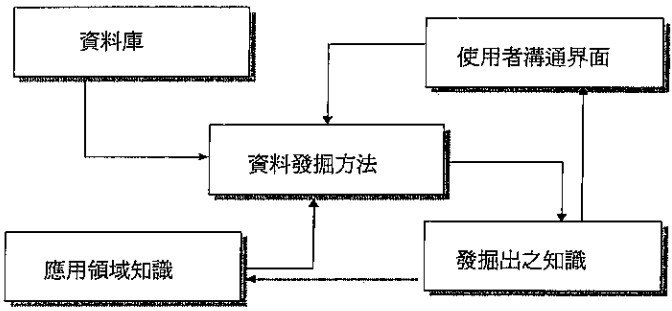

圖 1　資料發掘架構 (Frawley, et al., 1991)

一、資料庫

以目前之研究而言，對於資料庫本身相關議題上，分爲二大類。其一爲資料庫的設計與管理問題，包含雜訊處理（ Noise：指資料庫中之不正確或不確定資料)(Matheus, et al., 1993; Hong and Mao, 1991 等)，資料不完全情形處理 (Incomplete：指資料之應輸入而未輸入情形，即有 Null 值)(如 Quinlan, 1986; Frawley, et al., 1991 等) 及資料動態處理（ 指資料內容已過時)(如 Frawley, et al., 1991 等)。另一類研究主題，則針對發掘不同資料庫種類（ 如物件導向資料庫、空間資料庫等)(如 Han and Ng, 1994; Han, et al., 1994; Koporski, et al., 1996 等) 及不同的資料型態（ 如定量與定性資料之討論)(如 Paitesky-Shapiro, 1991; Smyth and Goodman, 1991)，而提出不同解決方式，這些並非本研究的重點。本研究所擬處理的爲傳統的關連式資料庫 (Relational Database)，並且假設其並無雜訊或不完全等情形。

二、發掘出之知識

資發掘所得之結果可以三個構面來探討，其一爲發掘出之知識用途，其二爲其表達方式，其三則爲後續處理方式。

(一) 知識用途

依使用者可能面臨的不同問題種類，所產生出之各種不同知識規則分爲六大類：

1. **特性規則** (Characteristic Rule)：

乃探討某一類別資料之相關屬性之特性，例如：所有研究生中，有 75% 乃出生於加拿大且 GPA 分數爲 excellent ；而另外 25%爲主修科學，出生於加拿大之外的國家學生，其 GPA 分數爲 good 。

2. **區別規則** (Discrimination Rule)：

爲資料庫中對於某些類別或資料項的特性比較。例如：只要主修科學的非加拿大國人，且其 GPA 成績爲 good 者，必爲研究生。而只要主修科目爲藝術或科學的加拿大人，且其 GPA 成績爲 average 者，必爲大學生。

3. **關聯規則** (Association Rule)：

此類規則最常被用在討論每項單品之間被購買的關聯性。如購買物品的消費行爲模式等。例如：所有交易清單中，出現換輪胎這二項者占 5.79%，而在請求換輪胎的交易行爲中，同時要求其它汽車服務者占 98.8%。

4. **分群規則** (Clustering Rule)：

統計中之分群規則，爲以群集分析方法，依各事物的特性分別出來，主要以其相對位置遠近 (Distance-Based Approach) 做爲分群依據。例如：以屬性值「種類」來看，資料庫內之資料可分爲三群：第一群爲屬電子設備，可能在美、中或日製造，可能有各種不同價位；第二群爲屬於在中國製造，價錢便宜或中等之服飾類；第三群爲屬深色，於美國或日本製造之各種價位的傢俱類。

5. **演進規則** (Evolution Rule)：

此規則由於表示資料在持續時間記錄下之變化趨勢，所以在結果呈現上，與上述之四種規則最不同點爲加入時間的因素於內 (Han, et al., 1995)。例如：作業系統產品的獲利率，在一年中之每個月相差不大，大約維持在 3.5%在右，而資料庫產品的獲利率則有明顯的成長趨勢。

(二) 表達方式

包含邏輯規則、決策樹、表格或圖形的表達方式。其中以邏輯規則最爲普遍，例如：上述特性規則中的例子，以規則方式表達即爲：

$$\forall(x)\text{graduate}(x)\rightarrow\text{Birth Place}(x)\in\text{Canada}\wedge\text{GPA}(x)\in\text{excellent}[75\%]\vee(\text{Major}(x)\in\text{science}\wedge\text{Birth\_Place}(x)\in\text{foreign}\wedge\text{GPA}(x)\in\text{good}[25\%]$$

(三) 後續處理方式

乃指發掘出來的結果是否儲存於資料庫中，以待日後有使用者查詢相同問題時可即時回應（如 Piatesky-Shapiro, 1991; Han, Fu and Ng, 1994）或更進一步與以前儲存下來的知識相比較，是否有衝突、矛盾等（如 Yoon and Kerschberg, 1993）。

三、資料發掘方式

機器學習的研究包含很多領域。Quinlan (1986) 將其分爲二個極端，均可用以作資料發掘的方式。其一爲適應性系統 (Adaptive Systems)，自我監視處理績效，並藉調整其內部參數而嘗試改進之。其二視學習爲獲取結構化知識，並以概念 (Concept) 或規則的方式呈現之。然而，在資料發掘領域中，除此二種外，尚有統計方式 (Fayyad and Uthurusamy, 1996; Fayyad et al., 1996)。以下分別說明之：

(一) 各種資料發掘方法簡介

1. 自我適應方式之發掘

此種方法以類神經最有名。類神經網路是以重覆學習的方法，將一串例子交與學習，使其歸納出一足以區分的模式 (Pattern)，學習效果十分正確並可做預測功能，爲十分重要的方法（如 Lewinson, 1993）。

2. 以擷取結構化知識方式之資料發掘

此種方式在資料發掘領域中，應用廣泛。包含樹狀分類法、關聯資料分析法與概念樹導向歸納學習法。

樹狀分類法爲利用各個屬性值，將資料庫中的資料予以分類，而形成決策樹；然後，可將新增資料依據決策樹的分類，歸至所屬類別之中。包含 ID3 (Quinlan, 1986; 1987)、CDP (Agrawal, et al., 1993a)、IC(Agrawal, et al., 1992)、C4.5 (Quinlan, 1993) 等均屬此歸納學習方法。

關聯資料分析法旨在找出各類資料之間的關聯性，希望能由資料庫中各式資料的分佈情形，了解其間彼此互動關係。包含 Agrawal 等人於 1992 年所提方法尋找交易資料之關聯規則 (Agrawal et al., 1992)、KID3 (Piatetsky-Shapiro, 1991) 等，而後又有一些效率改善的方法，如 AIS (Agawal, et al., 1993b) 等。

概念樹導向歸納學習則強調以每個屬性各自的概念樹建立，用來進行抽象化的動作，以發現出較高抽象層次意義上的資料關係如 (Han, Cai and Cerocne, 1992)(薛如芳，民國八十四年) 等。

3. 統計及數學方法

統計爲資料推測問題的核心方法 (Fayyad and Uthurusamy, 1996)。統計或許較少提供資料搜尋的架構，但對搜尋中假說的評估、搜尋結果的評核與瞭解卻甚爲重要 (Glymour et al., 1996)。例如， Agrawal 與 Srikant (1994) 用來檢測歸納出之規則的強度。不過，也有研究採統計爲資料搜尋方法，例如周立平 (民國八十四年) 採卡方檢定 (x^2) 判斷各類資料的相關性強度；Chan 與 Wang (1991) 利用機率統計推測方法尋找有雜訊、不完全的資料之間的相關性。

(二) 概念樹導向歸納學習法之探討

雖然資料發掘方式可分爲前述三大類，本研究乃專注於「以擷取結構化知識方式」的方法論，其原因爲我們希望解決商用資料庫面臨資料發掘所引發的相關議題，商用資料庫中之資料以定性者居多(例如：地址、電話等)，而類神經網路與統計方法主要在處理定量資料，對於定性資料需先以人爲對照方式預作轉換，在處理上較薄弱。另外，本研究目的是加入先前知識以輔助發掘，而類神經及統計與數學方式並不應用先前知識，故以歸納學習方法較爲合適。進一步，在三種「擷取結構化知識」的方法論中，本研究將採用概念樹導向歸納學習法，其主因爲此種方法所能處理的問題種類較爲完整。以下，本文將略加討論文獻中概念樹導向歸納學習法的相關課題。

1. 概念樹導向歸納學習法之策略

概念樹導向歸納學習法之主要精神爲將每個屬性均依專家知識或現實情況，各自建立概念樹 (Concept Hierarchy Tree)，目的即將每個屬性的屬性值依概念樹中之定義逐層提升，愈往上，則抽象程度愈高，屬性值數目愈少，有效減少資料量龐大的問題 (例如，「出生地」的屬性值爲「台北」、「台中」、「高雄」者，可提升爲「台灣」；爲「台灣」、「日本」、「韓國」者，可提升爲「亞洲」等，當然這前題爲系統必須先有此概念樹的定義)。

概念樹導向歸納學習法其實包含七個主要的學習策略，分別爲：

(1) 選取抽象化之屬性由單獨屬性選起，再選組合屬性 (Generalization on the Smallest Decomposable Components)。

(2) 無更高抽象屬性者，刪除 (Attribute Removal)。

(3) 隨著概念樹定義，將屬性往上抽象化 (Concept Tree Ascension)。

(4) 合併抽象化後所造成之相同記錄，並予以加總 (Vote Propagation)。

(5) 重覆以上步驟，直至每個屬性均小於屬性值個數之予許門鑑值之內 (Threshold Control on each Attribute)。

(6) 若以上之結果記錄數，仍大於總予許規則數之門鑑值，則再選取屬性做抽象化的動作 (Threshold Control on Generalized Relations)。此處之屬性選擇可選擇本身之屬性值種類最多者；抽象化後可能造成之結果數最少者；或抽象化後可能導致結果最明確者。

(7) 將最後之結果，轉換爲規則型態 (Rule Transformation)。

如此學習之執行結果，可依使用者需求將每個屬性的不同屬性值數目歸納至某一限量即「屬性值個數上限」(Attribute-Threshold) 之下，或將最後結果的規則數限定於某一數目即「關連表格列數上限」(Relation-Threshold) 以內，因此，其結果簡練故具高度可讀性，可讓使用者很容易接受 (Han, Cai and Cercone, 1992)。此法之主要特性爲 (1) 知識導向的學習，也就是可利用專家依領域知識建立概念樹；(2) 屬性導向的歸類，也就是每個屬性均根據其個自的概念樹往上抽象歸類，以簡化問題的複雜性 (Han, Cai and Cercone, 1993)。此類方法，最早被用於找尋如特性規則、區別規則。此外，其抽象化的概念，也可用於幫助尋找關聯規則、分群規則及演進規則。

2. 起始表格 (Initial Table) 之獲得方法

起始表格所指爲在確知使用者需求之後，於資料庫中將相關資料查詢出並置於暫存表格之內的一連串動作。而在起始表格的獲得方式，即如何獲知使用者需求上，目前研究中並無特別交代。均假設是已有問題存在，已知要發掘那類型 (如特性或區別) 規則，而後使用者或資料庫管理人員可依某種語法指定所需要的欄位去查詢而得到此起始表格，如 (Cai, et al., 1990; Han, et al., 1992 等)。

在 Dhar 與 Tuzhilin 的研究 (1993) 中，探討在建立起始表格的內容時，

可容許以使用者自訂字彙（ Vocabulary：爲特定名詞的定義，以屬性的條件限制，形成觀點來表示 ）及抽象觀念加以指定。他們的抽象觀念除了如同 Han 的概念樹之分類層級樹 (Classification Hierarchy) 外，另包含了對聚集屬性的抽象（ Dhar 與 Tuzhilin 稱之 Abstract Hierarchy — 例如：「年 / 月 / 日」的屬性可抽象化出「年 / 月」，更可進而抽象出「年」）。此外，他們可容許指定統計性函數屬性（ 如總薪水、平均薪水等 ）。

3. **意義性判斷**

目前在文獻中所提及判斷意義性之方法有下列二種。第一種爲用於關聯規則上之意義性判斷，其理念爲去除重覆之規則，使結果精簡，即達到意義性之效果。第二種意義性判斷，則是靠與使用者和系統互動完成，由使用者自行決定其所需資料。此種乃針對概念樹導向方法而做，其理念認爲應予使用者自行調整抽象層次，由其選擇及調整的過程中，找出最合乎其需求之結果 (Han and Fu, 1994)。

四、使用者界面

在以前文獻中，對於資料發掘使用者界面處理上，幾乎略過不談；而假設已經確認使用者需求後開始處理，重點於提出改良發掘效率的演算法。事實上，若無法探知使用者需求，不論採行何種快速的演算法，所得出之結果均不具效益，而且，使用者需求的明確程度，對於演算法的設計方式也具影響。以往大都假設使用者需求已很明確，甚至結果所需要求的抽象層次數目均由使用者自行決定（ 例如：屬性值個數或關連表格列數上限 ），但不論是何種歸納學習的方式，起始表格的資料內容，對於整個發掘結果具決定性影響。因此，若能進一步了解或揣摩使用者心態與意圖，以更多的資訊幫助確立使用者需求，對於整個資料發掘的完整性而言，應更具貢獻。

五、應用領域知識

應用領域知識，在許多研究中均十分強調其重要性（如： Smyth and Goodman, 1991 等 ）。但實際明確將之用於資料發掘程序之中者，尚不多見，唯有 Han 等人之概念樹爲明顯的應用。然而，並非所有領域知識均隱含於概念樹之中，也並非所有領域知識均合適以概念樹型態表達，例如企業運作法則 (Bussiness Rule)，以規則的方式要比以概念樹的表達更爲貼切。

若能將使用者需求配合上企業運作法則，妥善運用於此，不僅對於使用者需求的確認有所幫助，相信對於資料發掘結果而言，更能貼近現實世界的運作。

在資料庫的語意 (Semantics) 研究中，有很多語意資料庫模式 (Semantic Data Models)(Hull and King, 1981) 爲一系列重要研究。本研究採用 Chen (1976) 之原始實體關係模式 (ER Model 即 Entity-Relationship Model)，並加以延伸。在文獻中有很多的延伸的實體關係模式 (EER 即 Extended Entity Relationship)。根據以前文獻 (如 Mattos, 1988)，表達真實世界最普遍的抽象概念可包含一般化 (Generalization：乃係指不同實體的向上歸類，例如「研究生」與「大學生」均爲「學生」)、集合化 (Association ，包含：自然集合關係—例如所有「員工」所成的集合；索引集合關係—例如參與各「專案」的「員工」成的不同集合；列舉集合關係—例如列舉出所有常搭飛機的「員工」形成一個集合)、聚集化 (Aggregation ，例如：複合實體聚集化關係—如「汽車」是由「車輪」、「引擎」、「車身」等組合而成) 三大類。雖然，除此三種資料抽象化外，尚有很多其他抽象化的關係有待更進一步探討 (Goldstein and Storey, 1994; Storey, 1993)。但我們至少可考慮將此三種資料抽象化運用探詢使用者需求上，配合資料庫本身的語意，幫助得到更多隱含於實體間的關係。

六、綜合整理

根據上述之三種歸納學習方法—樹狀分類法、關聯資料分析法及概念樹導向歸納學習，加上資料發掘領域所談論的五類問題，形成一個二維矩陣表 1，即表示此三種研究派別對於上述五類問題的處理方式，整理如下：

表 1　文獻中歸納學習方法與資訊發掘架構之對應

	樹狀分類法	關聯資料分析法	概念樹導向歸納學習
使用者界面	文獻中未提及	文獻中未提及	•智慧型輔助： (Han, et al., 1996) (Han, et al., 1994) (Anwar, et al., 1992) •查詢語言： (Han and Fu, 1995) (Dhar andTuzhilin, 1993) (Han, et al., 1993) (Han, et al., 1992) (Cai, et al., 1991) (薛如芳, 1995)
資料庫	•雜訊處理： (Chan and Wong, 1991) (Quinlan, 1986) (Uthurusamy, et al., 1991) •空白資料： (Quinlan, 1986)	文獻中未提及	•雜訊處理： (Han, et al., 1993) (Cai, et al., 1991) (Cai, et al., 1990)
	•關聯式資料庫： (Agrawal, et al., 1933a) (Agrawal, et al., 1992) (Chan and Wong, 1991) (Quinlan, 1986)	•交易資料庫： (Agrawal and Srikant, 1995) (Park, et al., 1995) (Srikant and Agrawal, 1995) (Houtsma and Swami, 1995a ； 1995b) (Savasere, et al., 1995) (Agrawal and Srikant 1994) (Agrawal, et al., 1993b)	•關聯式資料庫： (Han, et al., 1995) (Dhar andTuzhilin, 1993) (Han, et al., 1993) (Han, et al., 1992) (Cai, et al., 1991) (Han, et al., 1991) (Cai, et al., 1990) (薛如芳, 1995) •物件導向資料庫、主動(Active) 資料庫：(Han, et al., 1994) •交易資料庫：(Han and Fu, 1995) •空間資料庫：(Han and Ng, 1994)
應用領域知識	文獻中未提及	文獻中未提及	•概念樹： (Han and Fu, 1995) (Dhar andTuzhilin, 1993) (Han, et al., 1993) (Han, et al., 1992) (Cai, et al., 1991) (Cai, et al., 1990) (薛如芳, 1995) •使用者自定字彙：(薛如芳, 1995)

發掘出之知識	•區別規則： (Agrawal, et al., 1933a) (Agrawal, et al., 1992) (Hong and Mao, 1991) (Chan and Wong, 1991) (Quinlan, 1986)	•關聯規則： (Agrawal and Srikant, 1995) (Park, et al., 1995) (Srikant and Agrawal, 1995) (Houtsma and Swami, 1995a ； 1995b) (Savasere, et al., 1995) (Agrawal and Srikant 1994) (Agrawal, et al., 1993b) (Piatetsky-Shapiro, 1991)	•特性、區別規則： (Dhar andTuzhilin, 1993) (Han, et al., 1993) (Han, et al., 1992) (Cai, et al., 1991) (Cai, et al., 1990) •關聯規則： (Han and Fu, 1995) •分群規則： (Han, et al., 1991) •進化規則： (Han, et al., 1995)
	•決策樹： (Agrawal, et al., 1992) (Hong and Mao, 1991) (Quinlan, 1986) •規則： (Agrawal, et al., 1933a) (Chan and Wong, 1991)	•圖形： (Piatetsky- Shapiro, 1991) •決策表、規則： (Agrawal and Srikant, 1995) (Park, et al., 1995) (Srikant and Agrawal, 1995) (Houtsma and Swami, 1995a;1995b) (Savasere, et al., 1995) (Agrawal and Srikant 1994) (Agrawal, et al., 1993)	•決策表、規則： (Han and Fu, 1995) (Dhar andTuzhilin, 1993) (Han, et al., 1993) (Han, et al., 1992) (Cai, et al.n, 1991) (Han, et al.e, 1991) (Cai, et al., 1990) (薛如芳, 1995) •圖形： (Han, et al., 1995)
資料發掘方法	•CDP 法：(Agrawal, et al., 1933a) •C4.5 法：(Quinlan, 1993) •IC 法：(Agrawal, et al., 1992) •INFERULE 法：(Uthurusamy, et al., 1991) •KD1 法：(Hong and Mao, 1991) •ID3 法：(Quinlan, 1986)	•SETM 法：(Houtsma and Swami, 1995a ； 1995b) •DHP 法：(Park, et al., 1995) •Partition 法：(Savasere, et al., 1995) •AprioriSome 法、Apriori-All 法：(Agrawal and Srikant, 1995) •Cumulate 法 、 EstMerge 法：(Srikant and grawal, 1995) •Apriori 法、 AprioriTid 法：(Agrawal and Srikant 1994) •AIS 法：(Agrawal, et al., 1993) •KID3 法：(Piatetsky-Shapiro, 1991)	•LCHR 、 LCLR 法： (Dhar andTuzhilin, 1993) (Han, et al., 1993) (Han, et al., 1992) (Cai, et al., 1991) (Cai, et al., 1990) •ML_T1LA 法、ML_TML1 法、ML_T2LA 法： (Han and Fu, 1995) •CLARANS 法： (Han and Ng, 1994) •FOIL6.2 法： (薛如芳, 1995)

本研究與國內的相關研究比較，如表 2 所示：

表 2　本研究與國人之前研究之比較

	薛如芳之研究 (1995 年)	周立平之研究 (1995 年)	本研究
使用者界面	下達 SQL 指令處理，輔以先前定義之應用領域知識	無	以圖形化界面方式點選，輔以先前知識庫中之知識處理
資料庫	爲關連式資料庫 以 SYBASE 軟體建置	爲關連式資料庫 現存之學校資料庫	爲關連式資料庫 以 ACCESS 軟體建置
應用領域知識	使用者自定字彙 (User-Defined Predicate) 屬性值抽象化定義	無	企業法則 經驗法則 延伸之資料字典(含使用者自定字彙) 資料抽象化觀念 屬性值抽象化定義 數值資料處理
資料發掘方法	歸納學習方法 以 FOIL6.2 爲學習系統	統計方式	歸納學習方法 將概念樹導向歸納學習方法加以改進
發掘出之規則	以規則方式呈現	以規則方式呈現	以規則方式呈現

參、研究架構

一、研究架構內容與用途

本研究提出一個新的架構，如圖 2 ，以延伸文獻探討之資料發掘架構。主要突破在於加入企業法則、一般化、集合化、聚集化等抽象化觀念、延伸之資料字典及經驗法則等先前知識。實體之抽象化觀念乃採 EER 模式並輔以資料字典，來延伸傳統關連資料庫所能表現之實體關係—即加以一般化、集合化、聚集化之語意豐富化，用以幫助探究使用者意圖。企業法則用以過濾問題。經驗法則用以確立問題範圍及路徑選擇，使資料發掘過程更見彈性。其基本假設爲對傳統關連資料庫，可由原資料庫設計人提供比較富語意的 EER 模式，並提供一些企業運作法則。其實以前文獻中 Han 等人也考慮到了抽象化的觀念，只不過他們只考慮到屬性值的抽象化而非實體關係的抽象化。此外，本研究並加入意義性判斷部份，讓發掘結果可讀性提高，更符合使用者需求。在屬性值抽象化定義部份仍爲事先建置概念樹於其中，並可依需求由使用者自行定義及選擇。在處理知識規則方面，專針對特性規則、區別規則做處理。並透過回饋的機制記錄使用者常使用的詢問組合，以幫助下位使用者發掘之用。

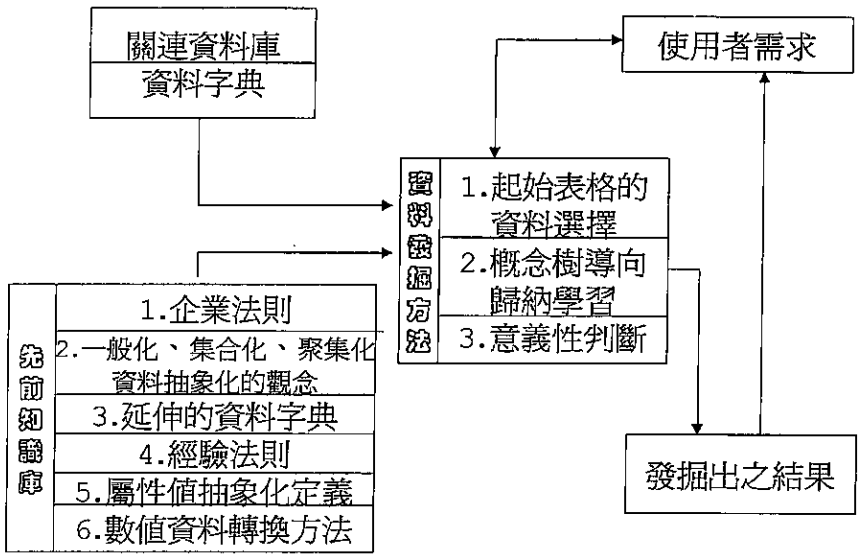

圖 2　研究架構（ 資料來源：本研究 ）

以下，將分項介紹本研究架構：

(一) 使用者需求界定

使用者對於專業知識的程度假設不同，所設計出的系統也將有所影響，在此，擬假設使用者能夠接受 EER 的呈現方式，且對於概念樹有基本認識。

1. **界面種類**

使用者需求界定主要目的是想明確定義使用者問題，使往後的資訊發掘程序得以順利進行。本研究採用呈現圖形化界面的方式將 EER 圖形呈現於使用者眼前，並輔以一些文字的條件設定，讓使用者點選並輸入其需求，由系統自動轉為資料庫查詢語言。並經由系統提供的問題擴充、範圍選擇及查詢路徑選擇等判斷機制，提供更具使用者親和性及更具智慧的溝通方式。

2. **問題種類**

本研究所探討的是目前資料發掘領域中的特性問題及區別問題。若使用者想進行區別規則探討 (例如：想了解登山社和合唱團的區別) 探討，則

使用者需指定其所想進行區別規則的分析屬性[1]爲何。由於在發掘過程中，區別規則也需經過與特性規則般的探討所有相關屬性的特性之後，才予以比較其差異，因此，在使用者自行選定問題種類後，即可由系統判定是否進行專屬區別問題演算法步驟。（ 例如：重覆標記等差別比較的動作 ）

由於資料發掘爲不斷的循環過程，本研究希望能借由圖形化的界面及內建的判斷機制，經由使用者對於問題的反覆發掘，能漸漸使其問題明確化，並可在反覆過程中，新增或刪除一些條件設定，找出更多的隱含資訊。

因此，使用者必須提供給系統的資訊包括：(1) 使用者有興趣了解的實體、(2) 分析屬性爲何（ 若使用者想進行區分規則，則系統將會要求使用者選擇一個分析屬性 ）、(3) 屬性值限制或其有興趣資料之條件限制、及 (4) 可允許之「屬性值個數上限」。

(二) 關連資料庫與其資料字典

在資料庫部份，本研究假設所處理的資料庫爲完整資料庫，並不考慮其虛值或雜訊的情況處理。而在資料庫種類選擇上，以附加 EER 語意之關聯式資料庫爲探討的模式。其內包含傳統的資料庫字典之定義[2]；也就是包含所有的關聯表格 (Relational Table) 名稱、其內屬性名稱、屬性的資料型態、資料長度。其目的爲將各個屬性予以定義，以方便之後發掘過程的使用。

(三) 先前知識庫

本研究著重於此部份先前知識處理上，將先前知識除 Han 等研究中常用的概念樹外，新增之發掘系統的先前知識包含六大部份。分別探討如下：

1. 企業法則

企業法則爲將一般企業運作規則，建置於系統之中，其內包含傳統資料字典中的屬性值範圍定義。其用處如下所述：

---

[1] 本文所稱的分析屬性指 Han 等人所研究中所提的特性屬性 (Characteristic Attribute) 及區別屬性 (Discrimination Attribute)。

[2] 本研究將傳統資料庫字典中有關屬性值範圍及主鍵值限制等均獨立出來，分別放在企業法則及延伸資料字典內；其理由爲各種資料庫管理軟體對系統檔案均有其自有的格式，若要直接使用，則需在本架構中另有轉換模組。

(1)縮小問題範圍：

於此假設資料庫內原有之資料必符合企業運作之規則限制。因此，在使用者所下之限制條件不合宜的情形下，可借助企業法則內之定義加入其它限制式，以將問題範圍予以縮小（如：想了解請假時數大於 70 小時之教職員特性，但若已有一企業法則規定除了女性職員請產假外，其它教職員之請假時數需在 70 小時以下，則可將問題予以縮小）。

(2)過濾問題：

若使用者所需了解的資訊爲本資料庫中未含或不可能成立者，也能透過企業法則的協助，立即予以回應，並請使用者修改其發掘條件（如：想了解請假時數大於 80 小時的教職員特性，但已有一企業法則規定所有員工請假時數不得大於 80 小時，則應予以拒絕）。

2. 三種資料庫抽象化觀念

由於往往在使用者詢問的問題背後，均有其動機存在 (Cuppens and Demolombe, 1988)，例如其欲了解各社團的特性，可能隱含想知道各社團組成份子的特性，若能在回答使用者問題的過程中，加入這些資訊，相信更能方便使用者使用。因此，若使用者對於本身問題不明確的情況下，本研究想透過所提前述 EER 之三種資料抽象化，推測使用者可能需求，再選出資料，進行發掘工作。

3. 延伸之資料字典

於本研究中，除傳統之資料字典外，尙包含其它延伸之資料字典，並將傳統資料字典內之主鍵值、外鍵值於此處獨立定義，其內應包含以下幾部份之定義。

(1)實際之資料庫與 EER 模式對應關係：此處所定義者爲關連式資料庫中之關聯表格與 EER 模式的實體或關係對應。

(2)導出值 (Derived Value) 與數值資料轉換公式定義：此處所包含的導出值爲指經驗法則內所定義的特殊資料項之轉換。例如：生日，轉換成年齡等；數值資料部份，則可存入幾種常用的數學計算公式，例如：平均值、標準差、中數、眾數等，以供以後需求時轉換之用。

(3)領域之專有名詞 (Vocabulary) 定義：專有名詞乃指資料庫設計者依其

領域知識，自行定義的名詞。其目的爲事先定義此應用領域內使用者可能用到的專業術語，並包含企業法則內專有名詞或特殊名詞定義。

(4) 屬性和其所屬概念樹的對應：每個關連表格內的屬性與其先前定義的概念樹彼此間的對應關係，用於在資料抽象化之時概念樹的取用選擇。另外，對於屬性層次的聚集化概念樹轉換函式對應也於此定義。

4. 經驗法則

(1) 一般化、集合化、聚集化三種關係之使用時機與方式—涉及單一實體時之問題範圍的選擇

由於本研究以資料抽象三種關係探究使用者之可能的需求意圖，並定出系統應支援的問題延伸方向，此種延伸方式，並非百分之百成立，僅爲經驗法則。其目的爲希望能幫助使用者獲得更多資訊，並期望使用者在反覆發掘的過程中，能藉由此種問題延伸及經驗法則內所述之屬性選擇建議，漸漸將其需求明確化。若使用者直接針對實體作出選擇，而不往下選擇實體之屬性，或所選擇的屬性均屬同一實體時，在此種使用者略過問題分析屬性設定的情況，本研究會視爲特性規則處理。此時，尚要判斷下列情況：

《一般化關係的延伸方向》

一般化關係爲最常用的實體關係。在探討某一實體時，若含有更高層的抽象實體 (Super-Class) 者，則常理上，均需將其所有更高層實體的屬性納入考量，並由使用者選擇其想進行特性規則或區分規則。若選擇區分規則，則由使用者再予選擇分析屬性。但若使用者有興趣之實體本身可分爲次實體型態(Sub-Type)時，使用者探詢意圖是否在次實體上，可能性並不像前述之由下往上考量情況來得強烈，因此，此種由上而下擴充問題的方式，系統並不自動將資料納入考量。

《集合化關係的延伸方向》

集合化關係中，集合 (Set) 爲由元素 (Component) 所組成。通常使用者於探究集合特性時，其可能想了解構成集合之元素的意圖較爲強烈。但由元素推估集合的特殊屬性情形，可能性較低，本系統將不自動處理該延伸方向。

《聚集化關係的延伸方向》

聚集化關係與集合化關係的情形十分類似，通常在聚集化關係中，使用者由「聚集」推往「元素」的可能意圖較強。反之，由「元素」推往「聚集」則視使用者自身需求而定。

《弱實體的加入時機》

由於弱實體 (Weak Entity) 並無單獨存在的可能，因此，若在使用者問題不明確的情況下，其有興趣的問題實體爲弱實體時，應將其主實體 (Parent Entity) 屬性，加入其內。但若使用者有興趣的問題實體爲一般實體時，則不將其弱實體屬性加入發掘資料中，也爲單向的延伸關係。

(2) 一般化、集合化、聚集化三種關係之使用時機與方式—涉及多個實體時之問題範圍的選擇

《一般化關係》

以一般化關係爲例，若使用者想了解「研究助理」與「教學助理」間的差異，只是此二實體間並無其它直接或間接關係存在，除了同與「研究生」實體間具一般化關係外。由於此問題爲區分二個原本即分開的實體，因此，此二實體各自的特有屬性，即爲二者結構上最大的不同。因此，除其由下往上與「研究生」、「學生」、「人」屬性連結時，屬性相同，可利用區別規則處理外，並將「研究助理」與「教學助理」結構或限制上的根本差異告知使用者。

《集合化關係》

若使用者有興趣的問題是了解「社團」(學校社團) 與「救國團」之間的差異，由於主問題之比較實體不同，因此，主問題是以將其結構上及其限制之不同資訊告知使用者。而在集合化的次問題延伸，由集合往元素了解其構成分子的差異性之時，應只取其相同之實體型態，例如：「社團」與「救國團」同樣有「學生」這種構成分子型態者，以區別規則處理。

《聚集化關係》

若使用者有興趣的問題是了解「學院」與「大學附屬教育研究單位」(如公共行政與企業中心) 之間的差異，由於主問題之比較實體不同，因此，主問題是以將其結構上及其限制之不同資訊告知使用者。而在聚集化的

次問題延伸，由聚集往組成了解其構成分子的差異性之時，應只取其相同之實體型態，例如：「學院」與「大學附屬教育研究單位」同樣有「建築物」這種構成分子型態者，以區別規則處理。

(3) 涉及多個實體或屬性之路徑選擇方式

此處所指爲使用者對於所想得知的問題已有方向，並可指出想了解的屬性或條件，且其所選定屬性的所屬實體彼此間具備除了前述所討論的一般化等三種關係外的其它關係。於此種情形下，系統需判斷的是如何爲使用者選擇合適的路徑，將其所需資料項予以連結，並且在此情形下，由於使用者需求已很明確，不用系統再將需求擴充，因此，起始表格中的屬性將只有使用者所選取者。

《選擇關係最直接的路徑》

路徑的選擇上，若使用者所選定的實體間有直接的路徑存在，無需透過第三個實體予以相連者，則直接採之爲當然路徑。若有一條以上之直接路徑，則列出由使用者選擇。

《選擇所有可能的路徑》

若使用者選定的屬性分屬於不同實體者，且實體與實體間並無直接路徑串連，均需透過第三個以上的實體才能連接者，表示此二實體的關係乃建立於第三個實體之上。因此，第三個實體，對於此二實體關係的連接十分重要，需將其間關係之相關屬性含入起始表格中。而在使用者選定的實體間有多條可能路徑的情形下，每條路徑所透過的中界實體並不同，其所帶來的涵義也不同，因此，需將之全盤考慮於內，才可幫使用者找出其所可能希望得知的資訊。

5. 屬性值抽象化定義

以概念樹之方式表達此處之屬性值抽象化定義，仍延襲 Han 的概念樹導向歸納學習之設計，以屬性值間之一般化概念樹及屬性內之聚集化概念樹爲主。

(1) 一般概念樹的定義及儲存

本研究延用以前文獻中 Han 等人的作法，將屬性依概念樹層級分別儲存，並爲根據領域知識事先建置。

(2) 屬性聚集化概念樹的定義

此處爲定義屬性層次的聚集化所需使用的概念樹。如：「年 / 月 / 日」ㄱ「年 / 月」ㄱ「年」。

(3) 導出值概念樹的定義

導出值由於是動態才轉換、計算的數值，因此於上述屬性值的概念樹定義部份，並未將此部份涵蓋於內，而於此另外定義導出值的概念樹，並於資料字典處定義其與屬性的對應關係。

(4) 多重概念樹的定義

依使用者需求不同，針對同一個屬性抽象化的方式也可能不同，有些屬性可能有建立多個概念樹的需求。因此，若能提供使用者自建概念樹及選擇概念樹的功能，使用者將更明確知道系統抽象依據，對於結果的接受程度也較高。

6. 數值資料轉換方式

在數值資料方面，轉換的需求十分頻繁，且根據不同用途，可能有不同的轉換需求，不似定性資料般較著重於抽象概念。例如：「社團」的總人數屬性，於抽象化過程中，使用者較有興趣了解的並非登山社的總人數多少，而是運動類社團的平均人數、中數、眾數等多少，以方便與音樂類、服務類等社團比較。因此，若單純的概念樹提升方式，對於數值資料的處理上並不合適，且由於每種數值根據不同的使用時機，或不同的需求角度，可能有不同的轉換需求，有的使用者可能只需了解平均數，其它使用者可能需了解其分佈情形等。因此，若能提供幾種較常用的數值轉換方式，供使用者自行選擇，將更方便。

(四) 資料發掘方法

在資料發掘部分，加強先前研究中並無深入探討的使用者需求方面，即其前段起始表格的資料選擇方式；中段概念樹的選擇、建置；與後段意義性判斷部份。有關起始表格內屬性的抽象化步驟與程序，則採用 Han 等人的架構概念。對特性規則演算法的修改有二點：(1) 導出值與數值資的轉換；(2) 概念樹的建置與選擇；(3) 處理至表格內之所有屬性滿足屬性值個性上限即停止，進入意義性判斷部份。

1. 起始表格的資料選擇

起始表格獲取方式，若使用者指定問題不明確，則以 EER 中實體關係依前述經驗法則先加以延伸。若問題已明確，則對各種不同路徑加以選擇。若遇導出值，則加以轉換。

2. 概念樹導向歸納學習—知識規則的處理方式

《特性規則》

步驟 C.1 ：導出值與數值資料的轉換。

步驟 C.2 ：建置、選擇概念樹。

步驟 C.3 ：選擇屬性的概念樹開始向上抽象化。

步驟 C.4 ：提不上去之屬性，予以剔除。

步驟 C.5 ：合併計算次數 (Vote) 。

《區別規則》

區別規則延續特性規則之作法 (C.1~C.5)，將之以區分屬性分成二群，再加以處理。

步驟 D.1 ：依分析屬性將資料分成二群處理。

步驟 D.2 ：將重覆部份予以標記。

3. 意義性判斷 (屬性選擇)

意義性判斷部份，採用與使用者互動方式，提供使用者一個便利探視抽象層次變化對於結果造成之影響的環境，讓使用者自行決定再行抽象屬性，再重覆資料發掘的演算法部份，直至使用者滿意爲止。

此種作法之目的爲修改 Han 等人之演算法之繼續抽象至「關連表格列數上限」部份，由於本研究認爲：只需經由「屬性值列數上限」的把關，將資料筆數有效減少之後，即可由與使用者互動，獲得使用者較滿意之結果，而非需經使用者事先自行設定「關連表格列數上限」。使用者其實並無法得知此門檻值之些微變化，對於所能獲得之結果的影響，例如：若系統已找到五條規則，但因原設定關連表格列數上限爲 4 ，屬性值須往上再抽象一層，結果因概念樹的關係，造成只剩二條規則或甚至一條含 any (即屬性值之最高抽象化—任何值) 的情況，而對使用者反而較無意義。因此，本研究將

此部份設計成與使用者互動，由使用者自行決定是否繼續選擇屬性進一步抽象化。

(五) 發掘出之結果呈現

本研究之結果呈現，均採表格並轉爲規則的方式呈現。此外，若使用者滿意此次發掘結果，也希望其能將發掘之目的輸入，如此，系統可將其屬性組合記錄，提供日後使用者建議。

二、演算法整體流程概述

如圖 3 所示，整個流程簡述如下：使用者先輸入問題後，系統先檢查企業法則，以求縮小問題範圍或過濾問題。系統若有儲存以前使用者之屬性組合，將會作屬性選擇建議。而後系統將查詢關連資料庫及其資料字典，再利用經驗法則，以輔助確認問題種類及其涉及之實體關係；若涉及三種資料抽象化，則先列出涉及之次問題。而後，查核延伸之資料字典，以建立起始表格。再來進行導出值及數值資料轉換。若有多重概念樹，則先作建置及選擇後；否則，直接進行抽象化動作，最後經由與使用者互動之意義性判斷，確認結果並呈現，若由使用者並可能輸入目的，由系統儲存本次之屬性組合。

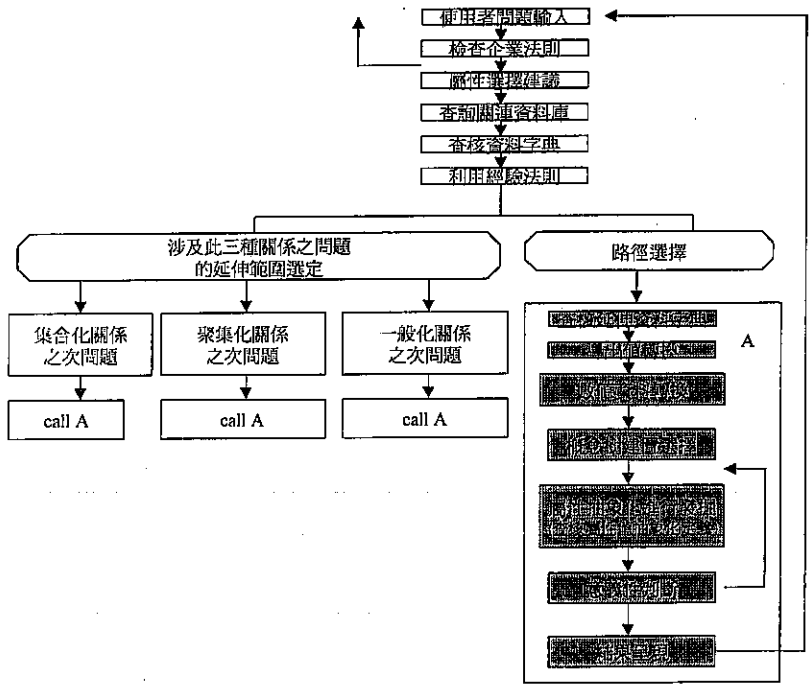

圖 3　演算法流程（資料來源：本研究）

肆、雛型系統實作

一、雛形系統設計

根據前述所提之研究架構及演算法，本研究配合資料取得及研究時間的限制選定假想之學校資料庫，進行雛形系統 (Prototype) 實作，以驗証此擴充先前知識於資料發掘過程之架構的可行性。

使用者界面處理上，以 Visual Basic 程式語言，於 Windows 3.1 環境中以視窗的圖形化界面，供使用者點選並輸入需求；資料庫內容，則以假想的學校資料庫，分別存於 19 個關連表格中，建置採用的資料庫管理系統軟體爲 Microsoft 公司的 Access 。

先前知識部份，本研究並沒有另外以獨立的知識庫處理所有先前知識，主要用於開發時間之限制考量；同時，與知識庫 (專家系統軟體) 的介面整合也非本研究的重點所在。知識庫若獨立存在將可便於管理與維護，然而本研究雖然沒有另外連接專家系統軟體來處理先前知識，但在雛型系統設計上，也對此管理與維護目的加以考量。在本雛型系統中，先前知識以各種不同方式分別呈現，其中與領域相關部份 (Domain Dependent)，均獨立存在，其獨立抽換性仍高；也就是本研究對於先前知識的作法爲將其中的一般化，集合化，聚集化資料抽象化的定義、延伸之資料字典、屬性值抽象化定義資料，均以表格的呈現方式存於 Access 資料庫中。此部份本身形成系統之資料庫，可由有權管理者以一般資料庫存取方式加以修改。而經驗法則由於是不受領域限制 (Domain Independent) 之處理原則，因此以程序性語言由 Visual Basic 的語法將機制置於程式內。至於企業法則的部份，其表達也以 Visual Basic 的獨立程式模組處理，由於一般常見的企業法則應爲規則型態，加上本雛型系統並非爲以邏輯語法的軟體開發，因此，所能處理的企業法則種類有限，爲本雛型系統的限制所在。在後續研究中，可將此部份以其它較適合的軟體開發，相信一定更合所用。

資料發掘過程上，本系統以 Visual Basic 的程式語法套用前述所提及的概念樹導向歸納學習法的資料發掘演算法，加以修改而成。唯對於圖 3 之「多重概念樹建置與選擇」的彈性作法並未實作；此外，對「屬性組合建議」，因系統知識累積不多，較無法提供有效建議。

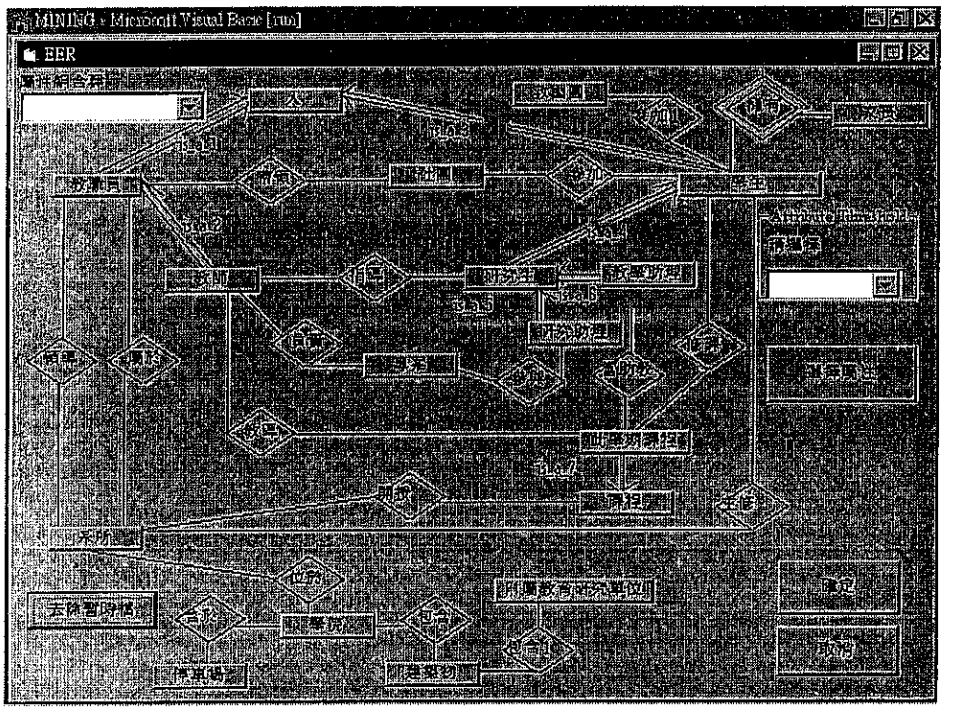

圖 4　系統主畫面

二、雛型系統介紹

(一) 使用者界面

1. 系統畫面簡介

本雛型系統的系統主畫面如圖 4 ，所示採圖形化界面方式，將 ER 中的所有實體及關係[3]，完整呈現於使用者面前，使用者可根據其需求點選有興趣的實體。在使用者點選有興趣之實體後，可點選圖 4 右邊之「選擇屬性」，進入圖 5 畫面。於圖 5 畫面之中，左半部之屬性爲使用者於圖 4 中所選擇實體的所有屬性，可供使用者選擇，若不選擇，則視爲全部選擇。圖 5 右邊欄位目的爲供使用者下屬性條件。「 Vocabulary 」部份爲本系統所定義之字彙部份，點選後其相對應的字彙定義，會自動出現於右下之空白欄位 (A 區)。 A 區也會自動出現系統根據使用者點選的實體、關係之屬性而轉換的建立起始表格的 SQL 查詢語法。此外，使用者可根據自己獨特的需

3　爲簡化起見,此處假設爲滿足正規化的 ER 圖。

求，於此圖之右下空白欄位處 A 區，自行輸入 SQL 查詢語言，進行查詢。當使用者「確定」其所輸入的條件後，系統將啓動企業法則查看使用者所下的條件，以縮小查詢範圍或當使用者之條件違背企業法則時，予以拒絕其發掘要求。

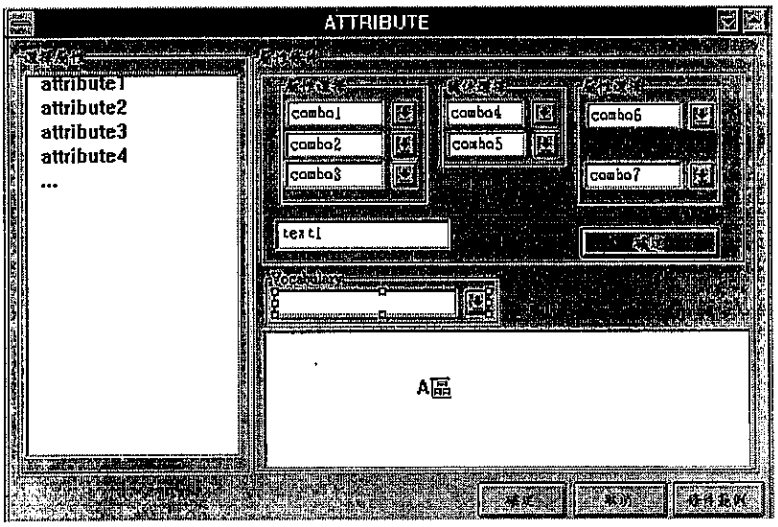

圖 5　屬性選擇，條件輸入畫面

2. 屬性條件限制輸入

於圖 5 之右半部，爲提供使用者下屬性限制之輔助介面。其上半部之屬性選擇部份可分三排，第一排 (combo1 、 combo4 、 combo6) 放置實體或關係名稱；第二排 (combo2 、 combo5) 放置其對應之屬性名稱；第三排 (combo3 、 combo7) 則予許使用者下達對應比率 (cardinality)。在本雛型中，此對應比指實體應至少參與關係的次數限制 (即最小值)，至於至多的限制其實作方式應雷同。

其使用方式有以下幾種：

(1) 單一實體之屬性條件：選定 combo1 ，則 combo2 中將出現其對應的所有屬性。「確定」後，系統將會把該實體所對應的屬性以 SQL 語法寫入 A 區，若使用者要進行區別問題分析，則須於 A 區輸入屬性值。

(2) 單一關係之屬性條件：可於 combo4 中針對有關 EER 模式中的關係作選擇，同樣的選定 combo4 (如「修課」) 後，並可於 combo5 中選擇其所應的屬性 (如「成績」)。

(3) 單一實體中單一屬性的屬性值出現個數條件：若使用者對於單一屬性的屬性值出現個數有興趣時（例如想了解各種籍貫中，有相同籍貫超過 5 個人的那些學生特性），使用者除了在 combo1 、 combo2 分別選擇實體（如「人」）及屬性（如「籍貫」）外，並可在 combo3 中選擇其所希望限制的屬性值出現個數（如「5」）。

(4) 單一實體中多個屬性的屬性值出現個數條件：若使用者對於同一實體中多個屬性值之組合的共同出現次數限制有興趣者（例如想了解同一「姓名」與「地址」出現二次以上者），則其可所選擇置於 combo2 中之屬性名稱（如「姓名」、「地址」），這將出現於圖 5 的 text1 中。並可選擇 combo3 個數（如「2」）。

(5) 實體↔關係↔實體之指定：若使用者想指定實體與實體之間的關係（例如：學生主修系所的關係），即可分別在 combo1 及 combo6 中選擇實體名稱（如「學生」與「系所」），於 combo4 中選擇其相對的關係名稱（如「主修」），如此，本雛型系統會將其關連轉爲 SQL 指令表達。

(6) 實體↔關係↔實體的對應比率條件限制：若使用者想加入實體間彼此參與情形的下限限制，則需於 combo1 、 combo4 、 combo6 中分別點選實體及關係，並在 combo3 或 combo7 中點選其參與條件的下限個數。例如想了解至少有 5 個學生有雙主修情形的系所，則 combo1 爲「學生」、 combo6 爲「系所」、 combo3 爲「2」、 combo7 爲「5」。

3. 介面設計理由

本雛型系統的主要介面爲圖 4 及圖 5 二個畫面，圖 4 是直接將 EER 視爲介面，由使用者在其上點選，主要是假設使用者對於 EER 的模式有基本認識與了解，能在其上找到所需要的資訊。而在圖 5 的左半部，則是集合其所點選的實體所包含的所有屬性，提供一處讓使用者可一次選擇的環境。另外，在圖 5 的右半部，較牽涉到屬性限制條件下法，主要根據使用者直覺與習慣設計 (combo1~combo7)，將 EER 的實體↔關係↔實體概念，引用過來。在其下的對應比率爲指此實體參與此關係最少次數的情形，易於對照了解。事實上，由於圖 5 的右半部爲一較多重用途的介面設計，在雛型系統設計之初，也考慮過其它方式，其中包含是否以 Enable / Disable 方式（即將某畫面於某情境下可用選項啓動，而不可用選項予以顏色淡化不准選用）引導

使用者使用或是直接將多種使用方式置於不同畫面之中，分開處理的方式。有關 Enable / Disable 方式，由於在多重用途之下， Enable / Disable 的切換須十分彈性，以滿足所有可能，但如此一來，真正有效引導使用者的目的不易達成。而多重畫面的方式，則由於畫面過多，使用較繁瑣而放棄。因此對於此圖 5 畫面，目前是以提供指引完整訊息的方式，幫助使用者處理。

(二) 先前知識

1. 企業法則

由於本雛型系統並未使用邏輯語言或使用知識庫開發，因此，受限於 Visual Basic 所能處理的邏輯能力，本雛型系統所能處理的企業法則種類也受限。其所能處理的企業法則種類如下所述。其中 (1) 至 (3) 實際儲存在有關資料庫實體關係之資訊中 (表 3 及表 4)。而 (4) 至 (9) 則以 Visual Basic 的模組儲存：

(1) 單一屬性的候選鍵限制 (Key Constraints) (如：學號必須唯一)。

(2) 複合鍵 (Composite Key) 限制 (如：沒有兩個同「姓名」的人住在同一「地址」)。

(3) 實體參與關係之至少及至多的對應比率 (如：學生至多主修 2 個系所，至少主修 1 個系所)。

(4) 數值型態的屬性值限制 (如：[社團].[人數] >= 30 ，其意義爲每個社團參與人數至少 30 人)。

(5) 文字型態的屬性值限制 (如：[教師].[最高學歷]='碩士' or '博士')。

(6) 單一條件 (IF A Then B) 之屬性限制 (其中 B 可爲單純的 and 或單純的 or 邏輯算式) (如： if [教師].[到職員] >= 1/1/81 then [教師].[最高學歷] = '碩士' or '博士'，其意義爲「於民國 81 年 1 月 1 日起到職的教師，其最高學歷須爲博士或碩士」)。

(7) 複合條件的屬性限制 (If A and B Then C ，其中 C 可爲含單純的 and 或單純 or 的邏輯算式) (如：[專案].[委託單位] = '國科會' and [教師].[最高學歷] = '碩士' or '博士' then [學生].[學號] = [研究生].[學號] and [學生].[年級] = ' 2 '，其意義爲「若專案的委託單位爲國科會，且參與專案的教師學歷爲博士或碩士，則參與專案的學生必爲二年級的研究

生」)。

(8)以實體間之關係爲條件的屬性值限制或對參與其它關係的要求 (如：if [教師].[身份証字號] = [系所].[領導教師 id] then [教師].[職稱] = '教授'，其意義爲「系所的領導教師，其職稱必爲教授」)。

(9)計算值的企業法則 (如：[研究生].[學號] in (select [研究生].[學號] from [研究生], [修課], [學生] where [研究生].[學號] = [學生].[學號] and [學生].[學號] = [修課].[學號] group by [研究生].[學號] having avg ([修課].[成績]) >= 70 ，其意義爲「研究生的修課成績平均必至少 70 分」)。

2. 資料庫抽象化觀念

本系統之有關資料庫抽象化基本資訊，均以關連表格的方式加以儲存，其定義如表 3 至表 8 。

表 3　實體之基本資訊 sys_table_1

實體名稱	主鍵值	複合鍵
人	身份証字號	{(姓名,地址)}
學　生	學　號	

＊此表記錄各實體之鍵值，如人有二個鍵「身份証字號」與「姓名，地址」(爲複合鍵)，其中，前者爲主鍵。

表 4　關係之基本資訊 sys_table_2

關係名稱	實體名稱	MAX	MIN
指　導	教　師	3	0
指　導	研究生	2	1

＊此表記錄各實體參與關係之對應比率限制，如實體「教師」參與「指導」關係至多 (MAX) 只能 5 個，至少 (MIN) 則沒限制 (爲 0)。

表 5　一般化抽象關係 sys_table_3

上一層	下一層
人	學　生
人	教職員

＊此表記錄一般化之階層關係，如「人」之下有「學生」及「教職員」。

表 6　集合化抽象關係 sys_table_4

集合	元素
社　團	學　生
社　團	教職員

＊此表記錄集合（如「社團」）是由那些元素（如「學生」、「教職員」）組成。

表 7　聚集化抽象關係 sys_table_5

聚集	組合
學　院	建築物
學　院	停車場

＊此表記錄聚集（如「學院」）中包含什麼組合（如「建築物」與「停車場」）。

表 8　弱實體關係 sys_table_6

Parent	Child
學　生	家　長
學　院	系　所

＊此表記錄弱實體（如「家長」）依賴之主實體（如「學生」）。

3. 延伸之資料字典

如前述，延伸之資料字典部份，包含實際之資料庫與 EER 模式對應關係（表 9 至表 11）、導出值與數值資料轉換公式定義、領域之專有名詞定義（如：北部爲[人].[籍貫] = '台北' or [人].[籍貫] = '基隆' or [人].[籍貫] = '桃園' or [人].[籍貫] = '新竹'）、屬性和其所屬概念樹的對應（表 12 及表 14）等四部份。在導出值轉換部份，本雛型系統中，均以功能函式的方式轉值並呼叫，若將來有其它的轉換需求，只需將之定義爲功能函式，或加入原來的功能函式之內即可。儲存方式可由表 13 查出其對應之副函式(如呼叫 time-point ，即執行 yu$ = year (now) -year (datevalue (yu$)，代表「人」的「生日」屬性依系統日期轉換爲「年齡」)。

表 9　EER 實體與對應關連表格名稱 sys_table_7

實體名稱	關連表格名稱
人	人
教職員	教職員

＊注意實體與關連表格名稱可能不一定相同，唯此處假設 EER 中的實體已正規化，因此一個實體即對應其關連表格。

表 10　EER 之關係與所對應關連表格名稱 sys_table_8

關係名稱	關連表格名稱
修　課	修　課

＊此表記錄那些關係(如「修課」)是獨立對應一個關連表格。

表 11　EER 隱含之關係與所對應關連表格外鍵名稱 sys_table_9

關係名稱	關連表格	FK
參　與	研究助理	專案名稱

＊此表記錄那些關係（如「參與」）是以外鍵方式成爲於那個關連表格的屬性（如「研究助理」表格中的「專案名稱」）。

表 12　一般概念樹的對應 sys_table_10

實體名稱	屬性名稱	概念樹名稱
人	籍　貫	籍　貫
課　程	課程名稱	課程名稱

＊此表記錄各實體屬性（如[人].[籍貫]）之對應的概念樹名稱（如「籍貫」）。

表 13　轉換值之概念樹對應 sys_table_11

屬性名稱	轉換後屬性	副函式
生日	年齡	time_point
性別	性別	boolean_change

＊此表記錄那些屬性（如「生日」透過那些副函式（如 time_point）予以轉換值（如「年齡」）

表 14　聚集化之概念樹對應 sys_table_12

屬性名稱	轉換後屬性	何副函式
起用日期	起用年月	ttime
起用年月	起用年	ttime

＊此表記錄聚集化的屬性值（如「起用日期」是透過何副函式（如 ttime）轉換成何觀念（如「起用年月」）

4. 經驗法則

本系統之經驗法則，用於判斷問題種類，決定問題是否延伸、問題範圍選擇及路徑選擇。若使用者只針對圖 4 中的多個實體做選擇而不選擇關係時，由系統主動幫使用者找出所有可能路徑，如圖 6 所示。若使用者本身已決定其有興趣發掘的路徑之情形下，也可以由使用者自行在圖 4 處，將完整路徑點選出來，如此，系統將針對使用者選定的路徑進行發掘，不再由系統尋找路徑。另外，若使用者只確定部份路徑，剩餘部份均只選擇實體，則系統將保留使用者選擇的前段路徑，主動幫其找出後段路徑之所有可能，同樣置於如圖 6 中，由使用者自行選擇。

5. 屬性值抽象化定義

在本系統中之儲存方式爲將每個概念樹定義分開儲存，即每個概念樹儲存在一個獨立的關連表格內。至於屬性聚集化概念樹的定義部份，則由於其所對應的是轉換函式，因此，將之與轉換函式名稱定義做對照，以方便呼叫。導出值概念樹則專門儲存導出值轉換後的抽象化概念樹。其定義如表 15。

表 15　人.籍貫之屬性值抽象定義 sys_table_13

抽象概念	次概念	層次級
北　部	台　北	1
北　部	基　隆	1

＊此表記錄屬性值一般化之抽象樹定義，如「北部」之下含有「台北」、「基隆」。

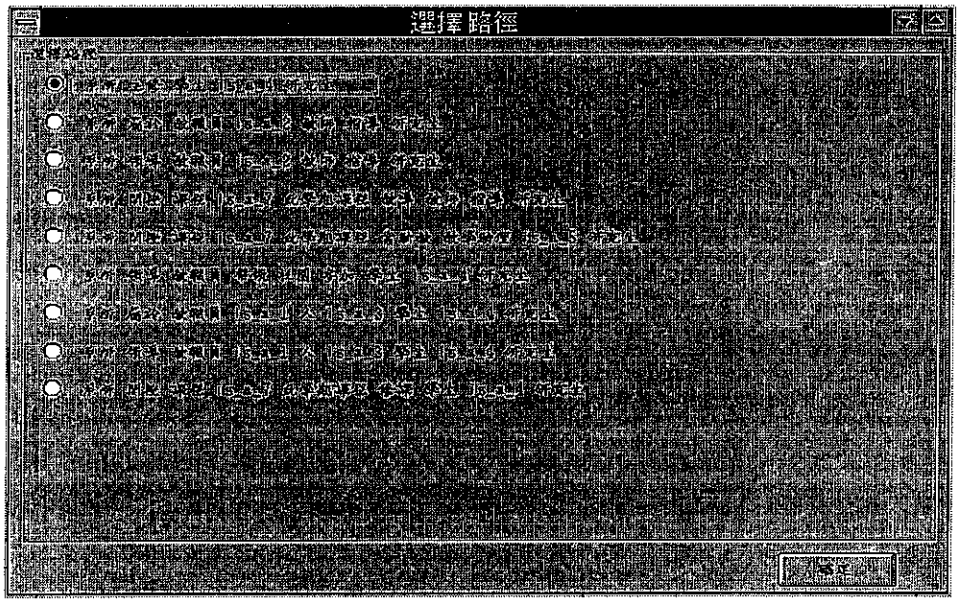

圖6　路徑選擇畫面

6. 數值資料轉換方式

本系統之數值處理公式只展示平均值之計算公式。例如：將年齡轉換爲平均年齡、薪水轉換爲平均薪水、人數轉換爲平均人數等。其定義也以功能函式的方式，只需轉入值，利用此處公式，即可轉換爲平均值結果轉出。事實上，其它數值處理公式的轉換作法也與平均值類似，將來若有需求，十分方便加入定義之內。

(三) 資料發掘

資料發掘方法，主要以概念樹導向歸納學習法加以修改而成。在系統根據經驗法則找出應包含於起始表格中之資料後，透過屬性值抽象化定義，將資料經過加以抽象化，合併，次數加總等動作，完成資料歸納動作。

在所有資料項均滿足先前使用者定義之屬性值列數上限值後，系統提供使用者進行意義性判斷功能，由使用者自行決定將何屬性再予以抽象化，若使用者所選取的屬性已無法再進行抽象化，屬性將被去除。若使用者想回復意義性判斷之前的資料，系統也提供讓使用者放棄所有更改，回復資料的功能，允許其不斷的嘗試，直至使用者找到其滿意的結果爲止。

(四) 結果說明

在使用者完成前述意義性判斷的工作，找出滿意的結果時，系統將之轉換爲結果判斷的型態（將在以下例二說明）。此時系統將依使用者原來是選擇特性規則或區別規則而對結果判讀的格式有所區別。若使用者接受此次發掘結果，則系統將要求使用者輸入其發掘目的，並記錄此次發掘結果的屬性組合。

(五) 系統維護與擴充

本雛型系統目前並未提供，包含先前知識等之內部資料的定義與維護畫面，但使用者仍可直接至 Microsoft 的 Access 軟體或利用 Visual Basic 本身所提供的 Data Manager 來做系統維護與修改的動作。未來對於本雛型系統的維護與擴充上，可能仍須建置這些畫面，以方便維護之用，也就是在系統主畫面上區分爲系統維護與進行資料發掘二個選項進行。對系統維護可以建置的畫面透過 Visual Basic 的 Data Manager 進行。

(六) 實　例

在此，我們舉三個例子，加以說明。唯得先說明的是：由於假想的資料筆數不多，發掘的結果可能較不具一般社會意義。

例一：單一實體一般化延伸

例如，使用者想發掘研究生的特性，此問題種類爲特性問題。可於圖 4 中點選「研究生」實體，並選 " attribute-threshold " =3 。此時，系統檢查企業法則後，發現無企業法則與之相關組合。注意在圖 4 的左上角有建議的屬性組合下拉式視窗。而後系統將查詢關連資料庫及其資料字典，再利用經驗法則，確認此問題涉及一般化資料抽象觀念，系統會將所有研究生的屬性（包含學生與人的屬性）列於圖 5 。使用者在圖 5 中未選擇特定屬性，也不針對屬性下條件限制。而後，系統查核延伸之資料字典，以建立起始表格。進行性別與年齡導出值轉換（原資料庫存的是 Boolean 型態的性別；而存生日不存年齡）。再來進行抽象化動作，其發掘後經第一次抽象化後的起始表格如圖 7 所示。

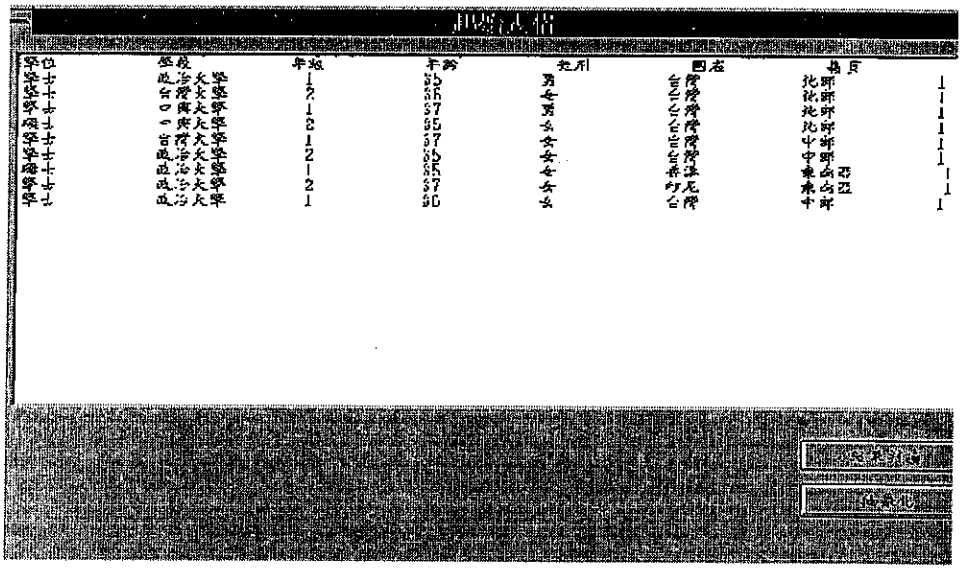

圖 7　研究生特性發掘之第一次抽象化

圖 7 爲起始表格經過第一次屬性值抽象化後的結果。其內的「學位」、「學校」分別爲「研究生」實體的屬性，代表此位研究生以前的最高學歷及在那個學校獲得此最高學歷。「年級」爲實體「學生」的屬性。「年齡」爲實體「人」的屬性「生日」轉換而成。其後的「性別」、「國名」、「籍貫」均爲實體「人」的屬性中，經過第一次抽象化後的結果仍未被刪除的屬性。

圖 8 爲圖 7 中之資料再次抽象化至均滿足 attribute threshold 後，再經過意義性判斷，去除「年齡」、「學校」、「年級」三個屬性後合併的結果。注意此圖中右下角有一可與使用者溝通之「意義性判斷」視窗，使用者可自行決定是否繼續抽象化。假設使用者決定接受此結果，其判讀如下：

∀(x) 研究生 (x) → (學位∈學士 ∧ 國名∈台灣 ∧ 籍貫∈北部 [33.33]) ∨ (學位∈碩士 ∧ 國名∈台灣 ∧ 籍貫∈北部 [11.11]) ∨ (學位∈學士 ∧ 國名∈台灣 ∧ 籍貫∈中部 [33.33]) ∨ (學位∈碩士 ∧ 國名∈香港 ∧ 籍貫∈東南亞 [11.11]) ∨ (學位∈學士 ∧ 國名∈印尼 ∧ 籍貫∈東南亞 [11.11])

也就是表示，研究生中，學位屬於學士，國名屬於台灣，籍貫屬於北部，占全部之 33.33%；學位屬性碩士，或名屬於台灣，籍貫屬於北部，占全部之 11.11%；學位屬於學士，國名屬於台灣，籍貫屬於中部，占全部之 33.33%；學位屬於碩士，國名屬於香港，籍貫屬於東南亞，占全部之

11.11%；學位屬於學士，國名屬於印尼，籍貫屬於東南亞，占全部之11.11%。

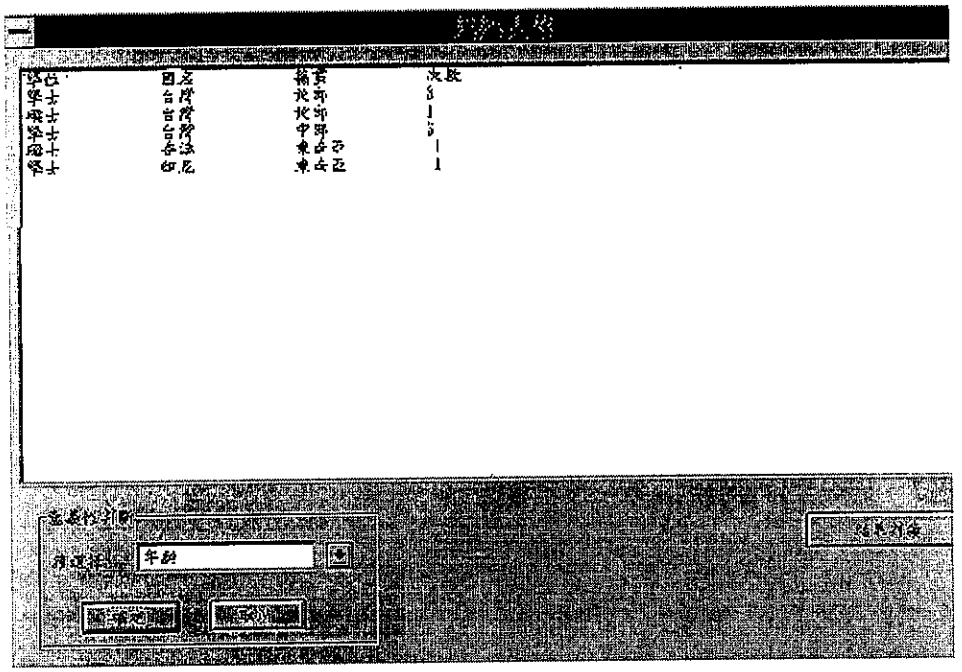

圖 8　研究生特性發掘最終結果

例二：參與關係數目限制

假設使用者想要發掘「參加了至少有兩個社員之社團的研究生特性」。同樣地，此問題種類爲特性問題。對使用者所需步驟而言，此問題與例一不同之處，在於使用者會於圖 5 中對屬性下條件限制，如圖 9 。而系統也會額外地去檢查關係的對應比率之企業法則（「社團」對「參加」的下限，若要求至少 30 個成員參加，則使用者等於沒下限制 ）。圖 10 則爲使用者接受此問題發掘結果。圖 11 爲結果判讀畫面。圖 12 爲發掘目的輸入畫面。

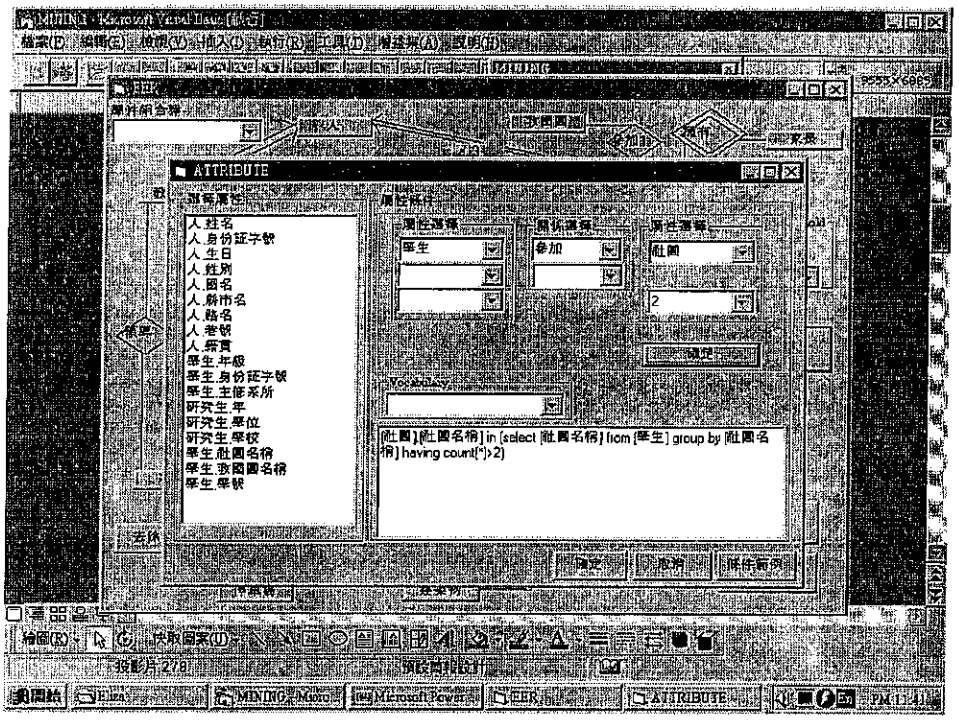

圖 9 「參加了至少有兩個社員之社團的研究生」之條件限制

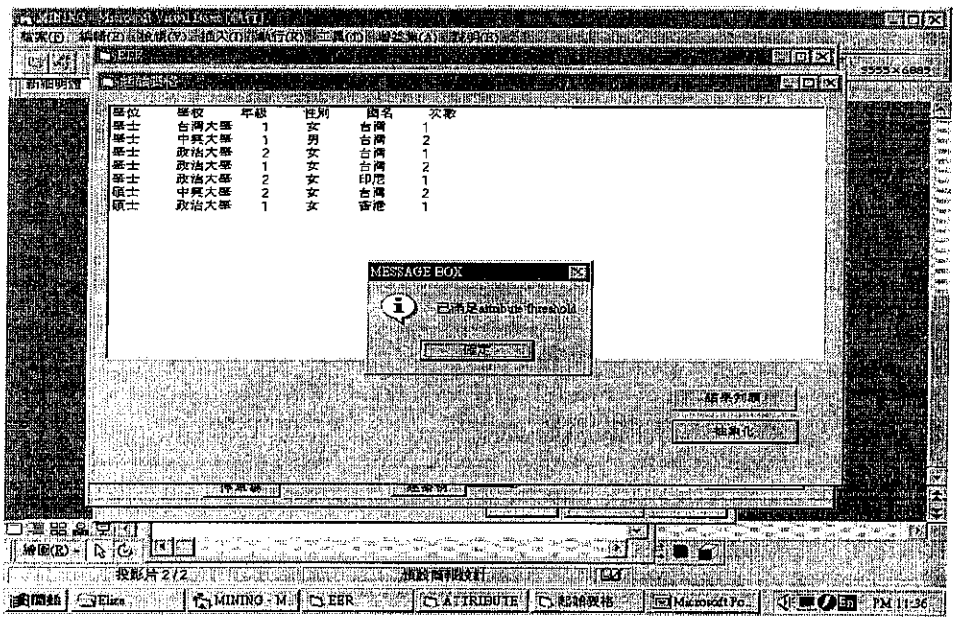

圖 10 「參加了至少有兩個社員之社團的研究生」之問題發掘結果

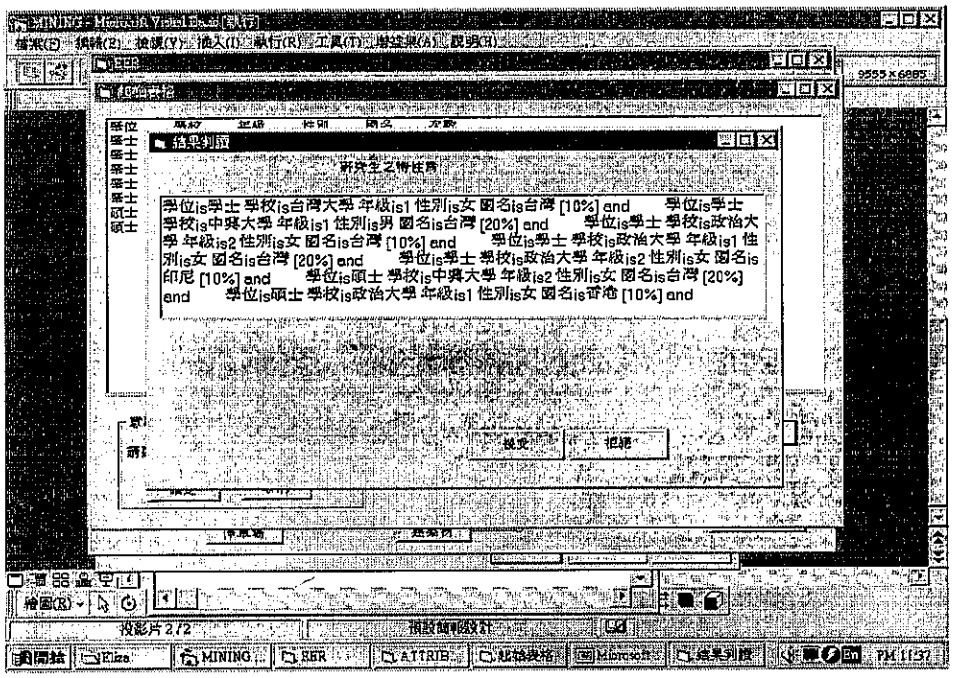

圖 11　「參加了至少有兩個社員之社團的研究生」之問題結果判讀

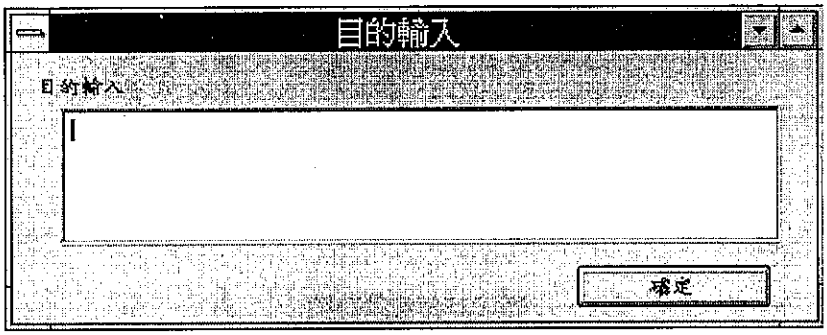

圖 12　「參加了至少有兩個社員之社團的研究生」問題發掘目的輸入畫面

例三：單一實體屬性值出現數目限制

在此最後一例中，我們只講解使用者輸入方式。假設使用者想瞭解「各種籍貫中，有相同籍貫超過 5 個人的那些學生特性」。則於圖 4 中選擇「學生」實體，並於圖 5 針對「學生」的屬性「籍貫」下條件，如圖 13 所示。

(七) 雛型系統的執行效率

由於本雛型的主要目的在於架構先前知識使用的機制，對於系統效率上

並未做特別考量。由於本雛型系統在先前知識的處理上，所處理的爲系統表格，並不涉及資料內之總資料筆數，在其時間複雜度計算上，會影響其效率者爲 EER 之實體數與關係數。但實體數與關係數相對於 n (資料庫內之應用領域資料總筆數) 而言，幾可視爲常數。而在屬性值的抽象化上，本研究所採用的爲 Han 等人所發展出之演算法，其執行效率爲 O(nlog(n)) (Han, et al., 1994)。因此，本雛型系統的時間複雜度應同爲 O(nlog(n))。

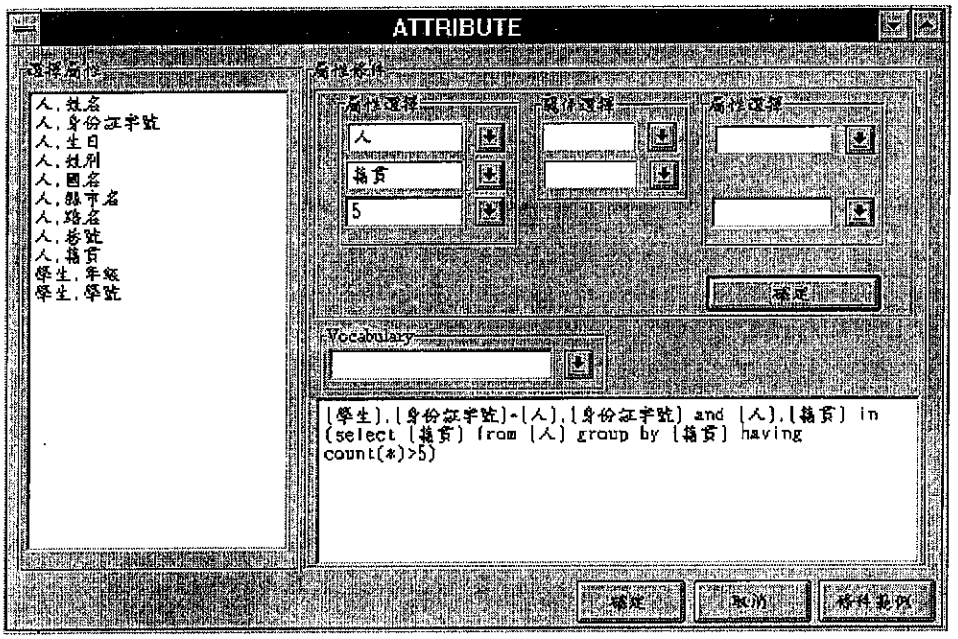

圖 13　「相同籍貫超過 5 個人的學生特性」輸入圖例

伍、結論與建議

一、結　論

本研究的主要貢獻在於提供一套資料發掘中，運用先前知識的處理機制，包含先前知識的種類，個別的用途與如何綜合運用，使資料選擇更合乎需求。具體而言，達成的成果如下:

(1) 資料發掘領域的相關文獻整理。

(2) 運用先前知識 (EER 實體層面抽象化觀念、經驗法則、企業法則、及

屬性值抽象化觀念，含數值及聚集化概念樹 ）提出新的資料發掘架構，將 Han 的資料發掘方法修正，並在結果判讀後，加入屬性組合累積系統知識。

(3) 雛形系統實作，包含：使用者介面輔助實體及屬性選擇、 使用者下達屬性條件限制之輔助機制、先前知識種類探討及運用機制、導出值處理與轉換、數值資料處理機制等。

本研究所提出的資料發掘架構之優點爲：

(1) 加入實體抽象化程度：以前研究中只著重屬性值抽象化程度部份，本研究在屬性層面外，另加上實體層次的抽象化程度。

(2) 運用企業法則：運用企業法則過濾或縮小使用者問題，簡省系統資源。

(3) 提供經驗法則：利用經驗法則爲使用者延伸及判斷問題處理方式，使所得結果更符合使用者可能需求。

(4) 使用者需求介面：具較親和性的使用者需求介面提供，及屬性限制輸入機制，讓使用者較易下達其需求。

(5) 系統的獨立性，各部份模組可抽換性高：雛型系統與應用領域知識部份分別獨立，可經由抽換應用領域知識，並加以定義先前知識的方式，將此雛型系統運用至其它領域之中，系統的獨立性高。

(6) 系統的擴充性：以本系統的架構，可將雛型系統加入知識庫，專門處理先前知識，也可透過 ODBC 等方式與其它資料庫連接，擴充性高。

二、建　議

對於後續研究的建議上，有以下幾點：

(1) 繼續探討其它先前知識種類及其使用機制：在本研究所探討的先前知識上，其種類仍有限，尤其在經驗法則的規則數上，仍有待後續研究加以補強，讓系統更符合一般使用者需求用法。此外，本研究目前只加入三種常見的資料抽象化觀念，或許可以再探索可否將其它有關資料庫語意 (Hull and King, 1987) 部份的相關研究，加入資料發掘中，使得對於語意方面的處理能夠更完整。

(2) 使用者介面的改進：相較以前的研究，本研究在使用者介面上最大的貢獻在加入 EER 介面及一些協助使用者點選的協助（如圖 5），更重要的，本系統會依經驗法則考量三種資料抽象化觀念，加以問題延伸。所以，使用者並不需要真正地懂得某種語言即可進行一般的資料發掘。但是以目前雛型而言，圖 5 若拿掉 A 區，並無法作到關聯完全性 (Relational Complete) 的要求。對於一些複雜的查詢，使用者必須自行與 A 區輸入補充的 SQL 語言，而且，目前雛型由於時間限制，並未涵蓋一個 SQL 的翻譯器 (Compiler) 去查核 A 區的語法，日後研究者，可考慮二種加強作法：(1) 加入一個 SQL 翻譯器，去查核使用者自下的 SQL 語言；或 (2) 考慮如 QBE (Query By Example) 的作法，以涵蓋更多的查詢種類，而達關連完全性的要求。

(3) 效率提升：由於本研究較著重於先前知識的運用機制上，對於資料發掘的效率上著墨較少，在真實世界中，由於資料發掘的資料量均十分龐大，因此，如何能在顧及效率的情形下，有效使用先前知識，爲後續研究可持續努力之處。

(4) 獨立的知識庫建立：若能有一獨立的知識庫，對於先前知識的管理與運用上，相信更方便。尤其在企業法則部份，本研究所能處理的企業法則種類十分有限，若能搭配另一個邏輯語言或專門處理知識的專家系統軟體，必能處理更多種類的企業法則，使系統的機制更爲健全。

(5) 本架構與資料倉儲的結合：若能利用資料倉儲處理完後的資料做爲發掘的資料來源，除了資料更爲豐富外；而且由於本研究假設的爲不含雜訊的資料庫，若透過 OLAP (On-Line Analytical Processing) 將原資料庫內的雜訊加以處理，即可直接以處理過的資料進行發掘的動作，也可節省不少先前對於資料的處理時間。此外在資料倉儲內，也有使用抽象化的概念，在各個不同層次檢視資料的情形，此點與本研究內之概念樹抽象化概念十分近似。在未來研究中，或許可試著了解此二套作法結合的可能性，甚或如本研究般加入其它實體層次的抽象化觀念於其中，以期能有更佳的處理效果。

(6) 結果判讀的加強：在結果判讀的部份，本研究與先前研究均是以規則的方式呈現，對於一般使用者而言，此種方式可能造成某種程度的不便，因此，在未來研究中，可加以探討是否能以更白話，甚至自然語

言 (Natural Language) 的方式，將結果整理成幾句話，幫助使用者一目了然地抓住整個概念。另外，甚至可試圖以使用者發掘需求的動機爲回答方向 (Intensional Query Answer)，直接切合使用者需求，相信對於結果判讀上能更有效益。

(7) 未來研究之實務應用：最後在未來研究上也有一些實務與理論結合的研究議題。例如，以審計爲例，由於會計資料本身不含雜訊，也許可利用資料發掘觀念，提早發現財務運作的不合理或異常情形 (例如：尋找各個明細科目的費用特性規則，以了解各項支出的情形)，以幫助審計人員提高其工作效率。或許也可以發掘演進規則，因爲演進規則加入時間的因素考量，可將各公司長期的營運資料加入判讀；了解由於時間因素所造成的變化情形，以幫助判斷異常情形。另方面，目前網際網路 (Internet) 的應用日廣，也許可思考是否可將資料發掘的觀念置於網際網路內的資料尋找上，但以目前的現實情況而言，由於各資料庫的種類不同 (可能爲層級式資料庫、網路資料庫或物件導向資料庫)，且其資料庫定義綱目 (Schema) 並無法取得，又大多爲非結構性資料，雖然這是一項十分值得處理的議題，但在理論研究與實務技術配合上仍有許多待突破之處。

參考文獻

周立平， 1995 ，從資料庫中發現法則:用學生修課的資料作分析，淡江大學資訊工程研究所碩士論文。

薛如芳， 1995 ，以歸納學習自關聯式資料庫中發掘知識，交通大學資訊工程研究所碩士論文。

Agrawal, R. and R. Srikant. 1995. Mining sequential patterns. ***IEEE 11th International Conference on Data Engineering***, Taipei, Taiwan, March: 3-14.

Agrawal, R. and R. Srikant. 1994. Fast algorithms for mining association rules. ***Proceedings of the 20th VLDB Conference***, Santiago, Chile, September: 487-499.

Agrawal, R., T. Imielinski. and A. Swami. 1993a. Database mining: a performance perspective. ***IEEE Transactions on Knowledge and Data Engineering***, 5(6): 914-925.

Agrrawal, R., T. Imilinski and A. Swami. 1993b. Mining association rules between

sets of items in large databases. ***Proc. of the 1993 ACM SIGMOD***, Washington, May: 207-216.

Agrawal, A., S. Ghosh ,T. Imielinski, B. Iyer and A. Swami. 1992. An interval classifier for database mining applications. ***Proceedings of the 18th VLDB Conference***, Vancouver, Canada, August: 560-573.

Anwar, T.M., H.W. Beck and S.B. Navathe. 1992. Knowledge mining by imprecise querying: a classification-based approach. ***IEEE 8th International Conference on Data Engineering*** , Phoenix, Arizona: 622-630.

Cai, Y., N. Cercone and J. Han. 1991. Attribute-oriented induction in relational databases. in G. Piatetsky-Shapiro and W.J. Frawley (Eds.). ***Knowledge discovery in databases***, 213-228. California: AAAI/MIT Press.

Cai, Y., N. Cercone and J. Han. 1990. An attribute-oriented approach for learning classification rules from relational databases. ***IEEE 6th International Conference on Data Engineering***, Los Angeles, California: 281-288.

Chan, K.C.C. and A.K.C. Wong. 1991. A statistical technique for extracting classificatory knowledge from databases. in G. Piatetsky-Shapiro and W.J. Frawley (Eds.). ***Knowledge discovery in databases***, 107-124. California: AAAI/MIT Press.

Chen, P. P. 1976 The entity-relationship model- toward a unified view of data. ***ACM Transaction of Database Systems***, 1(1):9-36.

Cuppens, F. and R. Demolombe. 1988. Cooperative answering: a methodology to provide intelligent access to databases. ***Proc. 2nd Int. Conference Expert Databases Systems***, Fairfax, VA, April :621-643.

Dhar, V. and A. Tuzhilin. 1993. Abstract-driven pattern discovery in databases. ***IEEE Transactions on Knowledge and Data Engineering***, 5(6):926-938.

Fayyad, U. 1996. Data mining and knowledge discovery: making sense out of data. ***IEEE Expert***, 11(5):20-25.

Fayyad, U. and R. Uthurusamy. 1996. Data mining and knowledge discovery in databases. ***Communication of the ACM***, 39(11):24-26.

Fayyad, U., G. Piatetsky-Shapiro and P. Smyth. 1996. The KDD process for extracting useful knowledge from volumes of data. ***Communication of the ACM***, 39(11):27-34.

Frawley, W. J., G. Paitetsky-Shapiro and C.J. Matheus. 1991. Knowledge discovery in databases: an overview. 1991. in G. Piatetsky-Shapiro and W.J. Frawley (Eds.). ***Knowledge discovery in databases***, 1-30. California: AAAI/MIT Press.

Glymour, C., D. Madigan, D. Pregibon and P. Smyth. 1996. Statistical inference and data mining. ***Communication of the ACM***, 39(11):35-41.

Grupe, F.H. and M.M. Owrang. 1995. Data base mining discovering new knowledge and cooperative advantage. ***Information Systems Management***,

12(4):26-31.

Goldstein, R.C. and V.C. Storey. 1994. Materialization. ***IEEE Transactions on Knowledge and Data Engineering***, 6(5):835-842.

Han, J., Y. Cai and N. Cercone. 1993. Data-driven discovery of quantitative rules in relational databases. ***IEEE Transactions on Knowledge and Data Engineering***, 5(1):29-40.

Han, J., Y. Cai and N. Cercone. 1992. Knowledge discovery in databases: an attribute-oriented approach. ***Proceedings of the 18th VLDB Conference***, Canada, August:547-559.

Han, J., Y. Cai and N. Cercone. 1991. Concept-based data classification in relational databases. ***Workshop Notes of 1991 AAAI Workshop on Knowledge Discovery in Databases (KDD'91)***, Anaheim, CA, July:77-94.

Han, J., Y. Cai, N. Cercone, and Y. Huang. 1995. Discovery of data evolution regularities in large databases. ***Journal of Computer and Software Engineering***, 3(1):41-69.

Han, J. and Y. Fu. 1995. Discovery of multiple-level association rues from large databases. ***Proc. of 1995 International Conference on Very Large Data Bases (VLDB'95)***, Zich, Switzerland, September:420-431.

Han, J. and Y. Fu. 1994. Dynamic generation and refinement of concept hierarchies for knowledge discovery in databases. ***AAAI'94 Workshop on Knowledge Discovery in Databases (KDD'94)***, Seattle, WA, July:157-168.

Han, J., Y. Fu and R. Ng. 1994. Cooperative query answering using multiple layered databases. ***Proc. 2nd int'l Conf. on Cooperative Information Systems (CoopIS'94)***, Toronto, Canada, May:47-58.

Han, J., Y. Huang, N. Cercone, and Y. Fu. 1996. Intelligent query answering by knowledge discovery techniques. ***IEEE Transactions on Knowledge and Data Engineering***, 8(3):373-390.

Han, J., S. Nishio and H. Kawano. 1994. Knowledge discovery in object-oriented and active databases. in F. Fuchi and T. Yokoi (Eds.), ***Knowledge Building and Knowledge Sharing***, 221-230. Ohmsha, Ltd, and IOS Press.

Han, J. and R. Ng. 1994. Efficient and effective clustering method for spatial data mining. ***Proc. of 1994 Int'l Conference on Very Large Data Bases (VLDB'94)***, Santiago, Chile, September:144-155.

Houtsma, M. and A. Swami. 1995a. Set-oriented data mining in relational databases. ***Data & Knowledge Engineering***, 17(3):245-262.

Houtsma, M. and A. Swami. 1995b. Set-oriented mining for association rules in relational databases. ***IEEE 11th International Conference on Data Engineering***, Taipei, Taiwan, March:25-33.

Hong, J. and C. Mao. 1991. Incremental discovery of rules and structure by hierarchical and parallel clustering. in G. Piatetsky-Shapiro and W.J.

Frawley (Eds.). ***Knowledge discovery in databases***, 177-194. California: AAAI/MIT Press.

Hull, R. and R. King. 1987. Semantics database modelling: survey, applications and research issues. ***ACM Computing Surveys***, 19(3):201-260.

Koperski, K., J. Adhikary and J. Han. 1996. Spatial data mining: progress and challenges. ***1996 SIGMOD'96 Workshop on Research Issues on Data mining and Knowledge Discovery***, Montreal, Canada, June.

Lewinson., L. 1993. Data mining: intelligent technology gets down to business. ***PC AI***, 7(6):16-23.

Matheus, C.J., P.K. Chan and G. Piatetsky-Shapiro. 1993. Systems for knowledge discovery in databases. ***IEEE Transactions on Knowledge and Data Engineering***, 5(6):903-913.

Mattos, N. M. 1988. Abstraction concepts: the basis for data and knowledge modeling. ***Proc. of the Seventh International Conference on Entity-Relationship Approach***, Italy, November:331-350.

Park, J.S., M.S. Chen and P.S. Yu. 1995. An effective hash-based algorithm for mining association rules. ***Proc. of the 1995 ACM SIGMOD***, San Jose, California, May :207-216.

Piatetsky-Shapiro, G. 1991. Discovery, analysis, and presentation of strong rules. in G. Piatetsky-Shapiro and W.J. Frawley (Eds.). ***Knowledge discovery in databases***, 229-238. California: AAAI/MIT Press.

Quinlan, J.R. 1993. ***C4.5: programs for machine learning***. Morgan Kaufmann Publishers: San Meteo, Calif.

Quinlan, J.R. 1987. Simplifying decision trees. ***International Journal of Man-Machine Studies***, 27(3):221-234.

Quinlan, J.R. 1986. Induction of decision trees. ***Machine Learning***, 1(1):81-106.

Savasere, A., E. Omiecinski and S. Navathe. 1995. An efficient algorithm for mining association rules in large databases. ***Proceedings of the 21st VLDB Conference***, Zurich, Switzerland, September:432-444.

Storey, V.C. 1993. Understanding semantic relationships. ***Very Large Data Bases Journal***, 2(4):455-488.

Smyth, P. and R.M. Goodman. 1991. Rule induction using information theory. in G. Piatetsky-Shapiro and W.J. Frawley (Eds.). ***Knowledge discovery in databases***, 159-176. California: AAAI/MIT Press.

Srikant, R. and Agrawal, R., "Mining Generalized Association Rules," Proceedings of the 21st VLDB Conference, 1995, pp.407-419.

Uthurusamy, R., U.M. Fayyad and S. Spangler. 1991. Learning useful rules from inconclusive data. in G. Piatetsky-Shapiro and W.J. Frawley (Eds.). ***Knowledge discovery in databases***, 141-158. California: AAAI/MIT Press.

Yoon, J. P. and L. Kerschberg. 1993. A framework for knowledge discovery and evolution in databases. ***IEEE Transactions on Knowledge and Data Engineering***, 5(6):973-979.