以廣義化關聯資料探勘方法設計物件導向資料庫壓縮技術之研究

李金鳳　　張簡尚偉　　王威澤
朝陽科技大學資訊管理系
台中縣霧峰鄉 413 吉峰東路 168 號
E-mail:{lcf, swc, s9014602}@cyut.edu.tw

摘要

資料探勘是從龐大的資料倉儲中挖掘有用的資訊，進而幫助企業網住有利的商業智慧，提高競爭能力。資料倉儲往往儲藏大量而複雜的資料，儘管隨著資訊技術的進步，資料儲存體的價格已大大地降低，但是在資訊容量永無止境增加的狀況下，如何提供有效而又兼具智慧的資料壓縮處理技術，裨始能節省資料儲存的成本，進而節省操作及壓縮資料的時間，卻是刻不容緩的事情。尤其近半年來資訊科技進步與網際網路驚人的擴充速度，傳統關聯模式或網路模式資料庫系統已不敷應付許多如 CAD/CAM、地理資訊系統、多媒體系統等進階的資料庫應用範疇。因此物件導向式資料庫(OODB)系統已日益廣受重視，並且對於進階的資料庫應用程式之技術具有相當的影響。本論文基於物件導向資料庫系統中物件繼承的廣義化機制，利用資料探勘的方法，採用 Class Inheritance Tree(CIT)來找出物件導向資料庫中的規則，對物件導向式資料庫進行資料壓縮研究與設計。其目標旨在降低廣義化物件資料的儲存空間，並利用部分解壓縮的技巧達到加速對物件存取的速率。

關鍵詞：資料探勘、關聯規則、資料壓縮、物件導向資料庫、廣義化機制

一、導論

扮演著企業資訊管理與系統核心的資料庫系統不斷演進，從檔案系統、階層式系統、網狀式系統、關聯式資料庫系統、物件導向資料庫系統等等，隨著資訊發展的日新月異，資料型態的多元化及 Web 網際網路的普及化正以旋風般的姿態影響應用系統的發展，技術的發展也逐漸地物件導向化，資料庫管理系統面臨新的挑戰。新一代的資訊系統所需之資料庫不但要能支援各式各樣的資料型態，也必須配合網路環境及物件導向技術的發展。資料庫大師 Dr. Stonebraker [11]預言物件導向式資料庫 (Object-Oriented DBMS,OODBMS)和物件關連式資料庫 (Object-Relational DBMS, ORDBMS)，將是廿一世紀新世代資料庫科技的主流。物件導向資料庫具備強大的物件導向模塑能力，可彈性地擴充各種資料型態，適合處理 Web 上各種複雜的資料，包括網頁（WebPage)、影像（Image)、圖形（Graphics）等，更是物件導向技術的最佳搭檔。在我們看到網際網路風起雲湧，數位多媒體網路興起，地理資訊系統（GIS）發展，和電腦輔助設計和製造(CAD/CAM)需求。我們似乎發覺愈來愈多新的資訊和系統的要求有賴物件技術來提供解答，結合文字、影像、聲音、資料庫的強勢行銷管道使得企業有能力創造新的經營契機，Dr. Stonebraker 的預言似乎要實現了。然而，在電腦取代人工之後，大量的資料的流轉與企業組織各個體系及流程作業間，隨著物件的增加，以致物件的尋找及資料的整理日趨

複雜，導致工作效率降低。資料蒐集對企業而言也許不是什麼難事，但是在資料庫茫茫大海中找出有用的資訊，就是一門很大的學問了。因此必須將處理資料的方式電腦化以提昇效率。系統中各種資料之間的關係相當密切且複雜，採用物件導向資料庫來設計，不僅程式設計師可以很容易地利用物件導向的特性及物件導向資料庫所擁有的物件關連特性來模塑系統架構及資料間的關係，同時設計系統取得資料的方式亦非常直接。

既然物件導向資料庫(OODB)系統日益受到重視且對進階的資料庫應用程式具有相當的影響，從關聯式資料庫系統中學習知識發掘的技術[1,2,7]，延伸應用於物件導向資料庫(OODB)系統中就顯得相當重要了[8,9]。物件導向資料模型和系統，在建構複雜資料庫時，收錄了大量豐富的資料結構和語義，如複雜資料物件、class/subclass 階層、class 組成階層、屬性繼承、方法和行動資料等。一個物件的屬性可能由子態 (subtype)繼承之，因而形成一個 class 構成繼承。一個物件 class 的廣義化應該主要執行它所擁有立即敘述屬性值。因此，物件的廣義化不僅帶來了系統的能力和彈性，卻也增加了實行的複雜度。此外，儘管計算機在記憶體的價格已隨科技技術的進步而大幅減低，然而在現今科技發達知識爆炸的時代，資料儲存的成本依然是資料庫應用的層面上所不可忽視的重要一環。因此，因應對資料永無止境的需求與增加，首要解決之道是提供有效的資料壓縮技術[4,6,10,12,13]，節省資料儲存的成本；此外，有智慧的壓縮技術不僅能提供高度的壓縮比例，更應該提供在查詢資料時不需要將壓縮資料予以整體還原再去搜尋的功能，以避免浪費大量的暫時記憶體來作為還原原始資料的儲存體。

本論文基於物件導向資料庫系統中物件繼承的廣義化機制，利用 Changchien 和 Lu [3]所提出的一種有效關聯資料探勘方法，Class Inheritance Tree (CIT)，進行資料壓縮研究與設計。其目標旨在利用物件資料之廣義化的繼承特性降低物件資料的儲存空間，並利用部分解壓縮的技巧達到加速對物件存取的速率。

本文以下章節內容包括: 文獻探討與回顧、研究方法及進行步驟、範例說明與實證結果、最後為研究結論與未來相關研究的方向。

二、文獻探討與回顧

本研究以 Changchien 和 Lu [3]提出一種有效的關聯資料探勘方法，Class Inheritance Tree (CIT)作為基本結構，並以物件資料庫中物件資料之廣義化的繼承特性設計一套可應用在物件導向資料庫壓縮技術之研究。以下茲將 CIT 的相關觀念與方法說明如下:

以 CIT 為基的資料探勘關聯方法參考了 rough set [5]的基本觀念以建立關聯規則。將以下表一做為說明 CIT 的四個基本步驟。

(一)產生條件等值類別(condition equivalent classes, CECs)與決定等值類別(decision equivalent classes, DECs):

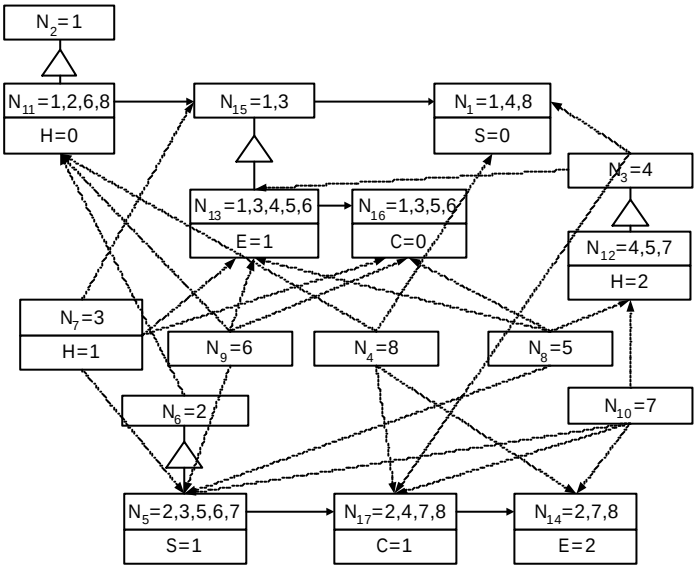
選擇屬性 “S”,“H”,“E”及 “C”做為條件屬性(condition attribute), 選擇屬性“Z” 做為決定屬性(decision attribute)。因此，共有 CEC_{S0}={1,4,8}, CEC_{S1}={2,3,5,6,7}, CEC_{H0}={1,2,6,8}, CEC_{H1}={3}, CEC_{H2}={4,5,7}, CEC_{E1}={1,3,4,5,6}, CEC_{E2}={2,7,8}, CEC_{C0}={1,3,5,6}, CEC_{C1}={2,4,7,8}等九個條件等值類別及有兩個決定等值類別，分別是 DEC_{Z0} = {1,3,6} and DEC_{Z1} = {2,4,5,7,8}。

(二)建立類別繼承樹(CIT)

根據 Changchien 和 Lu [3]所提出的 CIT 節點(node)定義及演算法，可知 CIT 的建立乃是建構在繼承的觀念上，依表一中所得的所有等值類別集(Equivalence Classes)，依據等值類別集(ECs)的規則，用樹狀架構把它們的關聯性表現出來，建構在一棵類別繼承樹，以做為下列關聯規則的找尋依據。

表一、具有 5-attribute **及** 8-tuple **的資料表**

ID	S	H	E	C	Z
1	0	0	1	0	0
2	1	0	2	1	1
3	1	1	1	0	0
4	0	2	1	1	1
5	1	2	1	0	1
6	1	0	1	0	0
7	1	2	2	1	1
8	0	0	2	1	1



圖一、依據表一的等值類別所建構的一棵類別繼承樹

(**三**)**找出下限近似規則** (lower approximation rules):

一組可以表示條件等值類別完全被包含於決定等值類別的事實之下限近似規則，此類的規則皆具有滿足最低支持度（minimal support**）及具有** 100% **的信賴度** (confidence, CF)**。因為**CEC_{H2} **完全被包含於** DEC_{Z1}**，因此可找出其中一條下限近似規則為** "If H=2 Then Z=1",**其**CF=100%**。**

(**四**)**找出上限近似規則** (upper approximation rules)**並計算信心度**(CF):

一組可以表示條件等值類部份被包含於決定等值類別的事實之上限近似規則，此類的規則也必須滿足最低支持度（minimal support**），其信賴度的計算公式如下。**

$$CF = \frac{Num(X \cap I(x))}{Num(I(x))} \qquad (1)$$

其中， $CF \in [0,1]$, $I(x)$**表示條件等值類別、**X **表示決定等值類別。**

因為 $CEC_{S0}=\{1,4,8\}$ **部份被包含於** $DEC_{Z1}=\{2,4,5,7,8\}$**，因此可找出其中一條下限近似規則為** "If S=0 Then Z=1", **其** CF=

Num({1,4,8}? {2,4,5,7,8})/ Num({1,4,8})= 2/3? 0.67。

(五)進一步去衍生更多的多維屬性關聯規則(multiple attributes association rules)**並計算信賴度**(CF):

以上所示皆是簡單的一維屬性關聯規則(single attributes association rules)**，使用者先定義出一個最低門檻值** *min-sup*(**最低支持度**)，**依據每一個事實只要有超過最低門檻值，就會被建立於一維屬性關聯規則中。再從所有具有常出現的項目集**(frequent itermset)**中，進一步去衍生出更多的規則，並且在規則間找尋可組合連結的部分以計算更多的多維屬性關聯規則並計算這些規則的信賴度**(CF)。

三、研究方法

資料庫儲存各企業間進行交易時的重要資訊以發展有效率資料庫知識發現的工具，對於資料庫、統計、機械學習和資料視覺化研究變得日益重要。本節是在物件導向資料庫系統中找出以廣義化為基礎的屬性（generalization-based attributes)，**並且將這些廣義化為基礎的屬性以** rough set **找出所有等值類別，並且利用一個有效的資料探勘方法**--CIT **結構**--**去發現在等值類別中的關聯規則。將這些演繹出的關聯規則取代部分物件，以壓縮重覆性資料進而達到節省空間的目標。也就是說，本目標的核心技術是利用資料壓縮的技術將物件導向資料庫中的物件之廣義化屬性轉化成在** CIT **結構中的一組等值類別，並利用快速搜尋將這些部分等值類別以相對應的一組關聯規則來取代之，以節省物件重複屬性儲存的空間。**

***step1*：將關聯表**(table)**中所有的屬性** $A=\{A_1, A_2, \ldots, A_N\}$，**建構出** Class Inheritance Tree(CIT) **的架構：**

(*1*)根據關聯表每個屬性 A_i **中具有相同值的記錄集合起來成為一個等值類別** (Equivalence Class; EC)，**並視每個記錄為所屬等值類別** (EC)**中的一個物件**(object)。**據此，建立出關聯表中所有的** $EC_1, EC_2, \ldots, EC_N$。

(*2*)依據 CIT **建構所有** ECs **的規則**[3]，**用樹狀架構把這些** ECs **的關聯性表現出來。在樹狀架構中，每一個節點**(node)**可能是由一個** EC **或多個** ECs **組成；若** ECs **之間有完全相同的** objects，**則會放在同一個** node **中。**

***step2*：從** CIT **中找出所有的關聯規則：**

先固定其中一個 EC_i **視為** Decision Equivalence Class (DEC) ，**因此** DEC_i **定義為具有** $A_i = v_i$ **之等價類別，其他的** ECs **則被視作** Condition Equivalence Class(CEC)，**接著在** CIT **結構中找出全部都具有足夠的支持度的關聯規則與計算出每一個規則的信賴度** (confidences;CF)，**直到每個** EC **都被以** DEC **視之後，並且找到所有的規則** R **為止，其中** $R=\{A_{i_1}=v_{i_{1,j_1}}, A_{i_2}=v_{i_{2,j_2}}, \ldots, A_{i_k}=v_{i_{k,j_k}} \leftarrow A_i = v_i \mid \text{for } i=1, 2, \ldots, N, \text{ for } v_{i_{1,j_1}}? \text{ Domain}(A_{i_1}), \text{and } i_? \{1, 2, \ldots, N\} \text{ with } i_? i\}$，**計算信賴度為** $CF_i = value_{CF_i}$

***step3*：利用關聯規則建構資料壓縮規則：**

根據 Goh *et al*.[6]**提出的壓縮觀念，本研究針對關聯規則的特性分成以下三類如公式**（2) , (3) and (4）**的壓縮規則，這些壓縮規則將來被存放在** metadata **中以作為一組** meta-rules，**系統根據這些** meta-rules，**可以對資料進行壓縮以減少資料存放的空間。**

（1) **針對每一個屬性** $A_i=?$，**若找到** s **個** ECs，**而且** $s? \textit{min-sup}$，for $i = 1, 2, \ldots, s$，**則可以用下列的表示法對資料壓縮：**

$$t(Z_1, Z_2, \ldots, Z_{i-1}, ?, Z_{i+1}, \ldots, Z_N) \leftarrow t?(Z_1, Z_2, \ldots Z_{i-1}, Z_{i+1}, \ldots, Z_N) \quad (2)$$

其中 $Z_1, Z_2, \ldots Z_{i-1}, Z_{i+1}, \ldots, Z_N$ **表示變數值，**? **表示在第** i **個屬性的值是常數。假設在一組具有** s **個常出現的項目集** (frequent itermset) **的物件，如** $t(a_{11}, a_{21}, \ldots, ?, \ldots, a_{N1})$，$t(a_{12}, a_{22}, \ldots,$

?, …, a_{N2})，…，t(a_{1s}, a_{2s}, …, ?, …, a_{Ns})

每個物件共佔的 byte 數為:

B（Z_1）＋B（Z_2）＋…＋B（Z_{i-1}）＋B（?）＋B（Z_{i+1}）＋…＋ B（Z_N）

根據公式(2)，這 s 個物件，可分別被替換成

t?(a_{11}, a_{21}, …, $a_{i-1,1}$, $a_{i+1,1}$,…, a_{N1})，

t?(a_{12}, a_{22}, …, $a_{i-1,2}$, $a_{i+1,2}$, …, a_{N2})，…，

t?(a_{1s}, a_{2s}, …, $a_{i-1,s}$, $a_{i+1,s}$, …, a_{Ns})

因此約可節省 s* B（?）的空間。

（2）對於每個一個 CEC 對應一個 DEC 的關聯規則，若找到 s 個 ECs，而且 s? *min-sup*，且其中 A_{j1} 為具有 CEC 的屬性，A_{j2} 為具有 DEC 的屬性， A_{j1}?A_{j2}，則可以用下列的表示法對資料壓縮：

t(Z_1, Z_2, …, Z_{j1-1}, ?, Z_{j1+1}, …, Z_{j2-1}, ?, Z_{j2+1}, …, Z_N) ← t?(Z_1, Z_2, …, Z_{j1-1}, Z_{j1+1},…, Z_{j2-1}, Z_{j2+1},…, Z_N) （3）

其中 Z_1, Z_2, …Z_{j1-1}, Z_{j1+1}, …, Z_{j2-1}, Z_{j2+1}, …, Z_N 表示變數值，?與?表示在第 j1 與第 j2 個屬性的值是常數。

假設在一組具有 s 個常出現的項目集(frequent itermset) 的物件，如

t(a_{11}, a_{21}, …, ?, …,?,…, a_{N1})，t(a_{12}, a_{22}, …, ?, …,?,…, a_{N2})，…，t(a_{1s}, a_{2s}, …, ?, …,?,…, a_{Ns})

每個物件共佔的 byte 數為:

B（Z_1）＋B（Z_2）＋…＋B（Z_{j1-1}）＋B（?）＋B（Z_{j1+1}）＋…＋B（Z_{j2-1}）＋B（?）＋B（Z_{j2+1}）+…＋ B（Z_N）

根據公式(3)，s 個物件被替換成所節省的空間為 s*[B（?）＋B（?）]。

（3）對於每一具有 m 個 CEC 對應一個 DEC 的關聯規則，若找到 s 個 ECs，若 CEC 為 k_1、k_2 、…、k_m（k_1<k_2<…<k_m），DEC 為 k_u，則可以用下列的表示法對資料壓縮：

t(Z_1, …, Z_{k_1-1}, ?$_1$, Z_{k_1+1},…, Z_{k_2-1}, ?$_2$, Z_{k_2+1}, …, Z_{k_m}, ?$_m$, $Z_{k_{m+1}}$, …, Z_{k_u}, ?$_u$, $Z_{k_{u+1}}$,…, Z_N) ← t?(Z_1, …, Z_{k_1-1}, Z_{k_1+1}, …, Z_{k_2-1}, Z_{k_2+1}, …, Z_{k_m-1}, Z_{k_m+1},…, Z_{k_u-1}, Z_{k_u+1}, …, Z_N) （4）

其中 Z_1, Z_2, …Z_{j1-1}, Z_{j1+1}, …, Z_{j2-1}, Z_{j2+1}, …, Z_N 表示變數值，在第 k_j 個屬性的值是常數 ?$_j$, for j=1, 2, …, m。

假設在一組具有 s 個常出現的項目集(frequent itermset) 的物件，如

t(a_{11}, …, $a_{k_1-1,1}$, ?$_{11}$, $a_{k_1+1,1}$,…, $a_{k_2-1,1}$, ?$_{21}$, $a_{k_2+1,1}$, …, $a_{k_m,1}$, ?$_{m1}$, $a_{k_{m+1},1}$, …, a_{k_u1}, ?$_{u1}$, $a_{k_{u+1},1}$,…, a_{N1})，t(a_{12}, …, $a_{k_1-1,2}$, ?$_{12}$, $a_{k_1+1,2}$,…, $a_{k_2-1,2}$, ?$_{22}$, $a_{k_2+1,2}$, …, a_{k_m2}, ?$_{m2}$, $a_{k_{m+1},2}$, …, a_{k_u2}, ?$_{u2}$, $a_{k_{u+1},2}$,…, a_{N2})，…，t(a_{1s}, …, $a_{k_1-1,s}$, ?$_{1s}$, $a_{k_1+1,s}$,…, $a_{k_2-1,s}$, ?$_{2s}$, $a_{k_2+1,s}$, …, $a_{k_m,s}$, ?$_{ms}$, $a_{k_{m+1},s}$, …, a_{k_us}, ?$_{us}$, $a_{k_{u+1},s}$,…, a_{Ns})

每個物件共佔的 byte 數為:

B（Z_1）＋B（Z_2）＋…＋B（Z_{k_1-1}）＋B（?$_1$）＋B（Z_{k_1+1}）＋…＋B（Z_{k_2-1}）＋B（?$_2$）＋B（Z_{k_2+1}）＋…+B（Z_{k_m}）＋B（?$_m$）＋B($Z_{k_{m+1}}$)+…+B（Z_{k_u}）＋B（?$_u$）＋B($Z_{k_{u+1}}$)+…＋ B（Z_N）

根據公式(3)，s 個物件被替換成所節省的空間為 $s*[\sum_{i=1}^{m-1} B(?_i)]$。

Step4：從 DEC 分類建構關聯規則中，對資料進行壓縮：

依照每一個具有 $A_{i_k}=v_{i_k,j_k}$ 的 DEC_i 做分類，算出每一個規則可壓縮資料的空間大小。計算每個規則可壓縮的空間方法如下：

$$N(C_i)=N(A_{i_1}=v_{i_1,j_1} \cap A_{i_2}=v_{i_2,j_2} \cap \ldots \cap A_{i_k}=v_{i_k,j_k}) * value_{CF_i} \quad (5)$$

$$S(C_i)=N(C_i) * [B(A_{i_1}=v_{i_1,j_1}) + B(A_{i_2}=v_{i_2,j_2}) + \ldots + B(A_{i_k}=v_{i_k,j_k}) + B(A_i=v_i)] \quad (6)$$

其中，

$N(C_i)$表示符合可被壓縮規則 "$A_{i_1}=v_{i_{1,j_1}}$, $A_{i_2}=v_{i_{2,j_2}}$, …, $A_{i_k}=v_{i_{k,j_k}}$ ✍ $A_i=v_i$" 的資料個數。$B(A_{i_k}=v_{i_{k,j_k}})$表示在 CEC_{i_k} 中，每單位資料所佔的 byte 數。

$S(C_i)$ 表示此壓縮規則總共可以壓縮 byte 數。

以下舉例來說明計算規則總共壓縮的空間的方法：

範例一：

假設有三個可壓縮的關聯規則已被建立如下表二。三個關聯規則可被壓縮的資料個數分別為 $N(A_1=v_1)=120$, $N(A_2=v_2)=250$, $N(A_1=v_1, A_2=v_2)=80$，以及在條件等值類別與決定等值類別的資料所佔的 byte 數分別為 $B(A_1=v_1)=2$, $B(A_2=v_2)=1$, $B(D=d)=1$。則從 DEC 分類建構關聯規則中，找出壓縮規則所壓縮的空間如 S(C) 欄所示。

四、範例說明與實證結果

本節是以一個小型物件式資料庫說明依據所提出的演繹法則做實驗設計與實驗結果之解析，並以實證方法之可行性。

範例二：有一物件資料庫內容如表三至表六之產品資料表、顧客資料表、繼承於顧客資料表的 VIP 資料表及交易資料表。將從表四中找出規則並壓縮之。

首先，建構 CIT 結構，圖二是根據類別繼承樹之部分圖。根據 CIT 中可將物件廣義化屬性利用繼承的結構將此一物件導向資料庫中的物件轉化成一組物件所成的等值類別集合並計算每一等值類別所佔的 byte 數，如表七中顧客資料表所有的 ECs 內容所示。

其次，依序把每一個 EC 視為 Decision Equivalence Class(DEC)，其他的 EC 就為 Condition Equivalence Classes(CECs)，從 CIT 中找出全部的規則與算出其信賴度(confidences)。直到每個 EC 都當過 DEC 並且找到所有的規則為止。本實驗分別以 A1、A2 與 A3 當 DEC，在 CIT 結構中找到的規則如表八所示。

接著，利用關聯規則建構資料壓縮規則，根據本研究所提出針對關聯規則的特性的壓縮規則，滿足最小壓縮空間達 30bytes 以上的規則共有 69 個，從 DEC 分類為依據將所有壓縮量從大排到小排序壓縮規則並計算所有規則可壓縮的空間，其結果如表九所示，這些規則將來可至於 metadata 中，以 Meta-rules 呈現作為資料壓縮與資料關聯性分析的準則。由表九，列舉一個關聯公式R1：D1→D1具有 CF=100% 來說明其壓縮的情況。根據公式(2)，可推導以下的壓縮規則，並且列出所有應用此壓縮則後的物件

T(OID, CUSNAME, PRONAME, AMOUNT, DATE) ← T?(OID, CUSNAME, PRONAME, AMOUNT) ，其中 T(OID, CUSNAME, PRONAME, AMOUNT, DATE)表示原物件的綱要，T?(OID, CUSNAME, PRONAME, AMOUNT) 表示壓縮後物件的綱要，由此規則所示，可發現物件中 DATE 屬性的資料值已被壓縮。

T?(11, 04, 01, 1)
T?(12, 04, 02, 2)
T?(13, 04, 03, 3)
T(14, 04, 01, 1, 2)
T(15, 04, 02, 2, 2)
T(16, 04, 01, 1, 3)
T(17, 04, 02, 2, 3)
T?(18, 05, 01, 1)
T?(19, 05, 02, 1)
T(20, 05, 01, 1, 2)
T(21, 05, 02, 1, 2)
T(22, 05, 03, 3, 2)
T(23, 05, 01, 1, 3)
T(24, 05, 03, 3, 3)
T?(25, 06, 01, 3)
T?(26, 06, 02, 3)
T?(27, 06, 03, 3)
T(28, 06, 01, 3, 2)
T(29, 06, 02, 3, 2)
T(30, 06, 03, 3, 2)
T(31, 06, 01, 3, 3)
T(32, 06, 02, 3, 3)
T(33, 06, 03, 3, 3)
T?(34, 07, 01, 1)
T?(35, 07, 02, 2)
T?(36, 07, 03, 3)
T(37, 07, 02, 2, 2)
T(38, 07, 03, 3, 2)
T(39, 07, 01, 2, 3)
T(40, 07, 02, 2, 3)

表二: **計算關聯規則總共壓縮的空間**

Association Rules	N(C)	S(C)
$A_1=v_1 \rightarrow D=d$，CF=1	120*1=120	120*(2+1)=360
$A_2=v_2 \rightarrow D=d$，CF=0.6	150*0.6=150	150*(1+1)=300
$A_1=v_1, A_2=v_2 \rightarrow D=d$，CF=1	80*1=80	80*(2+1+1)=320

表三、產品資料表

PRODUCT		
OID	NAME	PRICE
01	Cookie	30
02	Drink	15
03	Milk	50

表四、顧客資料表

CUSTOMER			
OID	NAME	ADDR	BIRTHDAY
04	Alice	TaiChung	1970/06/20
05	Bob	Taipei	1953/01/05

表五、繼承於顧客資料表的 VIP **資料表**

VIP (inherits CUSTOMER)					
OID	NAME	ADDR	BIRTHDAY	CARDNO	DUETIME
06	John	Taipei	1972/08/06	V11000	2002/07/31
07	Marry	TaiChung	1979/03/13	V11001	2002/07/31

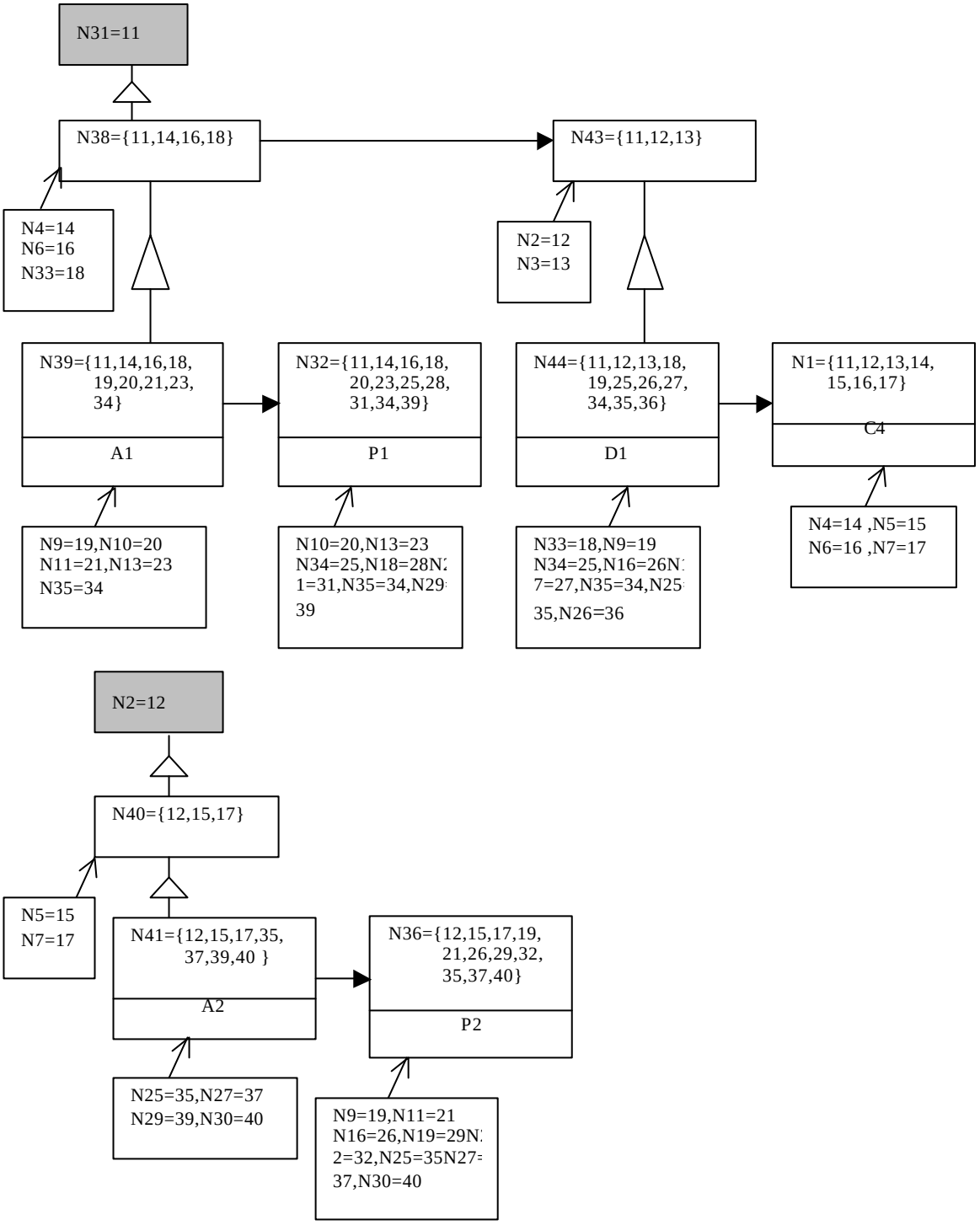
表六、交易資料表

TRANSACTION				
OID	CUSNAME	PRONAME	AMOUNT	DATE
11	04	01	1	2000/06/01
12	04	02	2	2000/06/01
13	04	03	3	2000/06/01
14	04	01	1	2000/06/02
15	04	02	2	2000/06/02
16	04	01	1	2000/06/03
17	04	02	2	2000/06/03
18	05	01	1	2000/06/01
19	05	02	1	2000/06/01

20	05	01	1	2000/06/02
21	05	02	1	2000/06/02
22	05	03	3	2000/06/02
23	05	01	1	2000/06/03
24	05	03	3	2000/06/03
25	06	01	3	2000/06/01
26	06	02	3	2000/06/01
27	06	03	3	2000/06/01
28	06	01	3	2000/06/02
29	06	02	3	2000/06/02
30	06	03	3	2000/06/02
31	06	01	3	2000/06/03
32	06	02	3	2000/06/03
33	06	03	3	2000/06/03
34	07	01	1	2000/06/01
35	07	02	2	2000/06/01
36	07	03	3	2000/06/01
37	07	02	2	2000/06/02
38	07	03	3	2000/06/02
39	07	01	2	2000/06/03
40	07	02	2	2000/06/03

表七: 顧客資料表所有的 ECs 內容如下：

物件所成的等值類別集合	**等值類別所佔的 BYTE 數**
C4={11,12,13,14,15,16,17}	B(C4)=2
C5={18,19,20,21,22,23,24}	B(C5)=2
C6={25,26,27,28,29,30,31,32,33}	B(C6)=2
C7={34,35,36,37,38,39,40}	B(C7)=2
P1={11,14,16,18,20,23,25,28,31,34,39}	B(P1)=2
P2={12,15,17,19,21,26,29,32,35,37,40}	B(P2)=2
P3={13,22,24,27,30,33,36,38}	B(P3)=2
A1={11,14,16,18,19,20,21,23,34}	B(A1)=2
A2={12,15,17,35,37,39,40}	B(A2)=2
A3={13,22,24,25,26,27,28,29,30,31,32,33,36,38}	B(A3)=2
D1={11,12,13,18,19,25,26,27,34,35,36}	B(D1)=8
D2={14,15,20,21,22,28,29,30,37,38}	B(D2)=8
D3={16,17,23,24,31,32,33,39,40}	B(D3)=8



圖二、類別繼承樹之部分圖

假設每一物件之屬性 CUSNAME**、屬性** PRONAME **與屬性** AMOUNT **中資料單位所佔的空間均為** 2bytes**，屬性** DATE **的資料單位空間均為** 8bytes**。應用所有過** metadata **中的壓縮規則之後，資料被壓縮後的壓縮比如下：**

CR=**壓縮後資料所佔的空間/壓縮前資料縮所佔的空間＝**2(10+4) /30(3*2+8)?6.7%

表八、以 A1、A2 與 A3 當 DEC，所得的規則及其信賴度

A1={11,14,16,18,19,20,21,23,34}

A2={12,15,17,35,37,.39,40}

A3={13,22,24,25,26,27,28,29,30,31,32,33,36,38}

A1			A2			A3		
OID	CECs	CF	OID	CECs	CF	OID	CECs	CF
11	C4	0.43	12	P2	0.45	13	P3	1
11	D1	0.36	12	C4	0.43	13	C4	0.14
11	P1	0.63	12	D1	0.18	13	D1	0.45
11	C4∩D1	0.33	12	C4∩D1	0.33	13	C4∩D1	0.33
11	C4∩P1	1	15	D2	0.2	22	C5	0.29
11	D1∩P1	0.75	15	C4∩D2	0.5	22	D2	0.5
11	C4∩D1∩P1	1	15	C4∩P2	1	22	C5∩D2	0.3
14	D2	0.3	15	D2∩P2	0.5	22	C5∩P3	1
16	D3	0.33	15	C4∩D2∩P2	1	22	D2∩P3	1
18	C5	0.86	17	D3	0.33	22	C5∩D2∩P3	1
19	P2	0.18	17	C4∩D3	0.5	24	D3	0.44
19	C5∩P2	1	17	D3∩P2	0.67	24	C5∩D3	0.5
19	D1∩P2	0.25	17	C4∩D3∩P2	1	24	D3∩P3	1
19	C5∩D1∩P2	1	35	C7	0.29	24	C5∩D3∩P3	1
20	C5∩D2	0.67	35	C7∩D1	0.33	25	C6	1
20	C5∩P1	1	35	C7∩P2	1	25	P1	0.27
20	D2∩P1	0.67	35	D1∩P2	0.5	25	D1∩P1	0.33
20	C5∩D2∩P1	1	35	C7∩D1∩P2	1	26	P2	0.27
21	D2∩P2	0.25	37	C7∩D2	0.5	26	D1∩P2	0.25
21	C5∩D2∩P2	1	37	D2∩P2	0.5	26	C6∩P2	1
23	C5∩D3	0.5	37	C7∩D2∩P2	1	26	C6∩D1∩P2	1
23	D3∩P1	0.5	39	P1	0.09	27	D1∩P3	1
23	C5∩D3∩P1	1	39	C7∩D3	1	27	C6∩D1	1
34	C7	0.14	39	C7∩P1	0.5	27	C6∩P3	1
			39	D3∩P1	0.25	27	C6∩D1∩P3	1
			39	C7∩D3∩P1	1	28	D2∩P1	0.33
			40	C7∩D3∩P2	1	28	C6∩D2	1
						28	C6∩P1	1
						28	C6∩D2∩P1	1
						29	D2∩P2	1
						29	C6∩D2∩P2	1

30	C6∩D2∩P3	1
31	D3∩P1	0.25
31	C6∩D3	1
31	C6∩D3∩P1	1
32	D3∩P2	1
32	C6∩D3	1
32	C6∩D3∩P2	1
33	D3∩P3	1
33	C6∩D3∩P3	1
36	C7	0.29
36	C7∩D1	0.33
36	C7∩P3	1
36	D1∩P3	1
36	C7∩D1∩P3	1
38	C7∩D2	0.5
38	C7∩D2∩P3	1

表九、所有 Rules **中，**Compression Space ≧ 30bytes**，並且由大排到小排序**

Rule ID	DEC	CECs	CF	N(CECs)	N(C)	S(C)
R1	D1	D1	1	11	11	88
R2	D2	D2	1	10	10	80
R3	D3	D3	1	9	9	72
R4	D1	A3	0.36	14	5	50
R5	A3	D1	0.45	11	5	50
R6	A3	D2	0.5	10	5	50
R7	D2	A3	0.36	14	5	50
R8	A3	D2∩P2	1	4	4	48
R9	D3	A3	0.29	14	4	40
R10	A3	D3	0.44	9	4	40
R11	P2	D2	0.4	10	4	40
R12	D2	P2	0.36	11	4	40
R13	A1	D1	0.36	11	4	40
R14	D1	P1	0.36	11	4	40
R15	D1	P2	0.36	11	4	40

五、研究結論與未來相關研究的方向

物件導向資料庫具有反覆使用，穩固性高，可靠、快速設計、整合、容易解讀、可標示多媒體型態資料等優異的特性，使得物件導向資料系統日益受到重視，並 且對進階資料庫

應用極具相當的影響。在原本傳統關聯資料庫知識發覺的研究已相當成熟，因此將關聯式資料庫系統中學習知識發展的技術延伸應用於物件導向式資料庫系統中更是刻不容緩的事情。隨著資料量的迅速膨脹與資料關係複雜化，資料儲存的需求日益增加,資料壓縮技術因此也日趨重要。尤其在大量而複雜的物件資料中極需要經濟且有效率資料壓縮技術裨始能減少儲存空間、縮短傳輸速度、降低資料的傳輸成本並兼具加速查詢的效益。本研究提出一套基於 CIT 結構所具有的繼承特性以廣義化關聯資料探勘方法設計物件導向資料庫壓縮技術。從實例的結果中可發現，當資料庫內的資料具有高度重覆性時，此壓縮技術可達極佳之壓縮效益。此外，應用關聯資料探勘技術所衍生的壓縮規則同時可廣泛應用於分析導向的資料庫如資料倉儲，對於時常需要反覆存取與處理資料庫而言，本壓縮方法不僅可達壓縮的效果，更可達到快速資料檢索的效益。

因為本壓縮技術在應用關聯探勘方法時，所找出的關聯規則具有一定的重覆度，這些重疊的關聯規則會導致壓縮規則數量的增加與訂定壓縮規則的複雜性。未來本研究將針對如何選擇最佳的關聯規則以訂定資料壓縮規則，在提高壓縮比與避免存取效率的降低之間取得平衡做進一步的研究。

六、參考文獻

1. R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," Proc. 1993 ACM-SIGMOD Intl. Conf. Management of Data, Washington, DC, pp. 207-216, (May 1993).
2. R. Agrawal and R. Srikant," Fast algorithms for mining association rules," Proc. 20th VLDB Conference, pp. 487-499, 1994.
3. Changchien, S. W. and Lu, T. C., "A new efficient association rules mining method using class inheritance tree (CIT)," **第十二屆國際資訊管理，台灣大學，**May 18-19, 2001.
4. G.V., Cormach, "Data compression on a database System," Communications of the ACM, Vol. 28, No. 12, pp. 1336-1342, 1985.
5. S. Deogun, V. Raghavan, A. Sarkar and H. Sever, “Rough sets and data mining – analysis of imprecise data,” Kluwer Academic publishers, 1997.
6. C. Goh, M. Tsukamoto and S. Nishio, "On database compression with knowledge discovery algorithm," Proc. JSAI SIG-KBS-9602, pp.1-6, 1996.
7. M. Houtsma, and A. Swami, "Set-oriented data mining in relational databases," Data and Knowledge Engineering, Vol. 17, pp. 245-262, 1995.
8. W. Kim, “Introduction to object-oriented databases”, MIT Press, 1990.
9. S. Nishio, H. Kawano, and J. Han, "Knowledge discovery in object-oriented databases : rhe first step," Proc. AAAI-93 Workshop on Knowledge Discovery in Databases, Washington, DC, pp. 186-198, July 1993.
10. B. Y., Ryabko, "A locally adaptive data compression scheme," Communications of the ACM, Vol. 16, No. 2, 1987.
11. Michael Stonebraker and Greg Kemnitz, "The POSTGRES next generation database management system," CACM, pp. 78-92, Oct 1991.
12. T. A., Welch, "A rechnique for high-performance data compression," IEEE Transactions on Computers, Vol. C17, No. 6, pp. 8-19, 1984.
13. J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," IEEE Trans. on Information Theory, Vol. IT-23, No. 3, pp. 337-349, 1977.