

A Decremental Rule Extraction Based on the Rough Set :

Case of Customer Satisfaction on Restaurant's Service

以約略集合為基礎的減量式規則擷取：

以餐廳客戶服務滿意度為例

高燕芬

國立暨南國際大學資訊管理學系
s98241020@ncnu.edu.tw

黃俊哲

國立暨南國際大學資訊管理學系
cchuang@ncnu.edu.tw

Ying-Ling Hsieh

國立暨南國際大學資訊管理學系
s98213544@ncnu.edu.tw

摘要

動態資料庫是一種常見的商業資料庫類別，資料刪除也是在管理資料庫中頻繁的操作。不幸的是，大多現有的資料探勘(DM)演算法假設資料庫是靜態的且更新資料庫需要重新計算所有規律以得到法則。在現有的資料探勘技術中，約略集合(RS)是一個知識發現的工具，可以用來找出隱藏在龐大數據中的邏輯規律。然而，傳統的約略集合(RS)並不能產生有優先順序的法則，且常常缺少重點，產生太多無法被保證可信的法則。此研究提出了 DAREA (Decremental Alternative Rule-Extraction Algorithm)解決從資料庫刪除物件的問題以及產生以偏好為基礎的法則，根據強度指數(strength index, SI)，特別是在所需的屬性集合(reduct)並不是唯一的情形。該演算法並不需要重新計算，在一開始就可以快速產生且形成法則。這個解決辦法被應用在餐廳各項服務和顧客基本資料對滿意度的影響。透過此案例討論驗證，該演算法優於傳統約略集合(RS)方法。

關鍵詞：動態資料庫、約略集合、減量式探勘、減量物件、法則歸納

壹、緒論(Introduction)

過去，研究者常常假設資料庫是靜態來簡化資料探勘的問題(Hong et al., 2011)，在現實的應用中，資料庫會因為資料的插入、刪除以及修改等管理資料庫中常見的操作隨著時間更動(S. Zhang et al., 2007)。根據這些結果，如果這種行為隨著時間的推移而改變，原本的系統可能會導致不可接受的結果以及決策(Crespo et al., 2005)。商業資料庫是動態的，在某種意義上(i)在資料庫的資料可能會隨著時間不斷更新，所以內容以及大小是可以改變的(Zhang et al., 2003; Sohel et al., 2004)。(ii)舊的資料必須從資料庫裡刪除(S. Zhang et al., 2007; Zhang et al., 2009)。(iii)分散式資料庫在一個固定的間隔時間更新一個資料區塊(Parthasarathy et al.)。

資料刪除是在許多商業資料庫中常用的操作之一(Sohel et al., 2004)，當一些資料庫紀錄被刪除或修改，資料庫的內容會更新且稱為更新資料庫(updated database)。更新資料庫(updated database)的探勘被稱作減量式探勘(Sohel et al., 2004)，這顯然是低效能且耗時的，必須反覆執行資料的探勘演算法去分析原始沒有遞減的資料(Zhang et al., 2009)，尤其是減量式規則的提取，隨著時間比較被更新的資料庫以及資料的刪除來提出法則，因為刪除舊的資料式資料庫管理常見的操作之一(Zhang et al., 2007)。不幸的是，大多現有的資料探勘演算法假設資料庫是靜態的且資料庫更新需要重新搜尋所有模式，通過掃描整個包含舊的與新的資料，以探測提出決策(Sohel et al., 2004)。

在資料探勘的領域，約略集合(RS)方法可以處理基於單獨物件模型的質性資料(Kusiak, 2001)，約略集合理論由 Pawlak 所提出(Pawlak, 1982)，作為一個在知識系統表達不確定性資料的方法。該理論已經被廣泛應用在制定決策中，特別是多準則中排序和分類的問題(Greco et al., 2001; Yuhua et al., 2010)。然而，到目前為止。很少的約略集合(RS)方法將減量式規則提取考慮進去，因此當資料從資料庫中刪除需要很長的時間(S. Zhang et al., 2007; Zhang et al., 2009)。

此外，相關的文獻中發現(Gorsevski and Jankowski, 2008; Pattaraintakorn and Cercone, 2008; Shyng et al., 2007)，使用約略集合(RS)歸納屬性時常產生太多不重要的規則，約略集合(RS)方法並不能保證決策表的分類是可信賴的(Tseng, 1999)。因此，曾等人(Tseng et al., 1999)提出替代規則提取演算法(Alternative Rule Extraction Algorithm)來解決這個問題。替代規則提取演算法(AREA)發現以偏好為基礎的規則，按照屬性集合(reducts)的強度指標(SI)，特別是當多個屬性集合(reducts)可以取得同個強度指標的值(SI)，在所需的屬性集合(reducts)不一定是唯一的情形下。使用替代規則提取演算法(AREA)，替代的規則是可以被定義的，與原來的決策具有相同的優先規則，但可能比原來的更具吸引力。

為了解決與動態數據庫資料的刪除和提取可信和替代的規則等相關的問題，本研究提出了 DAREA (遞減替代規則提取演算法)。該演算法解決如何從資料中提取規則當刪除一些資料可以被刪除的問題。該演算法比較被更新資料庫以及將特別的資料從原本的資料庫移除以提出規則，而不是再次執行替代規則提取演算法(AREA)。根據建議的方法，當資料庫必須在一個特定的時間間隔和特定的資料從資料庫中刪除，規則提取的結果正確反映了目前的情形。當資料被刪除後，需要重新執行演算法去分析整個資料庫。

該演算法在實際生活中被應用在多個領域中，本研究將演算法應用在餐廳服務與顧客評價的關係中。因為餐廳的競爭激烈，因此如果能夠有一個法則能夠依循，就能夠提升餐廳的競爭力，透過了解顧客基本資料以及餐廳所提供的服務，得知顧客評價與特性的關聯，進而得到決策法則來做出促銷策略，留住顧客來店意願。

以下第二節將回顧過往相關的文獻介紹，第三節解釋所提出演算法的演算步驟，第四節將該演算法應用到餐廳顧客評價的案例中分析，第五節則對本文所述內容做出結論。

貳、文獻回顧

文獻回顧包括基於規則產生(rule generation)以及動態資料庫減量式技術中約略集合理論的近似觀念。

一、約略集合理論的近似值概念

基於近似的概念，約略集合的規則產生(rule generation)演算法能夠產生有意義的決策法則，有潛力實現廣大的知識和提供新的見解。Pawlak(Pawlak, 1982)定義約略集合普遍概念相當於拓撲的操作，內部以及封閉，稱為近似。

下近似值的集合是所有顆粒(granules)完全被包含在集合的聯集；上近似值的集合是所有非空集合顆粒(granules)的聯集；邊界的集合是上近似值與下近似值間的差異(Pawlak, 1982)。

二、約略集合基本規則的產生

為了獲得有意義的決策法則使用兩個步驟。首先，去掉條件屬性中多餘的屬性，產生屬性集合。其次，太多的屬性集合(reduct)在第一個步驟中被產生，法則歸納演算法要求產生具有最大強度指標(SI)的簡潔的決策規則，有可能實現淵博的知識並提供新的見解。

(一) 屬性簡化過程

屬性集合(reduct)被定義為一個屬性的最小足夠的集合，具有相同的能力去辨別當所有屬性的集合都被使用(Pawlak, 1991; Ziarko and Rijsbergen, 1994)。屬性集合表示條件屬性必須能夠做出決定，在簡化屬性的過程中，前置法則歸納演算法去除多餘的資訊或屬性，挑選在原始屬性集合中具有相同辨識的屬性集合(Fan et al.,2009)。

(二) 法則歸納過程

屬性集合有一個以上的屬性(reduct)，所以簡化決策法則不會產生單一的效果，因此決策法則可以根據使用者問題範圍假定標準所誘導。在一般情形下。決策法則是一些屬性值的結合，就像是同一類所匹配的集合。

為了規則的歸納，一個演算法發明來產生決策法則，這是有可能實現廣大知識和提供新的見解。這些決策法則對專家來說更有益處去分析且獲得對問題的掌握(Fan et al.,2009)。當屬性選擇可以有效提高分類的精準度，約略集合屬性選擇演算法使用其選擇屬性。

然而，在過去約略集合方法無法產生有優先順序的法則，也就是說，不能產生有意義的一般法則(Sun et al.)。約略集合基礎演算法往往產生太多法則沒有重點，且不能保證決策表的分類是可信的。Tseng et al. (2006)提出了新的 RST 法則歸納方法，稱為規則提取演算法(rule extraction algorithm)，根據屬性集合的強度指標(SI)產生以偏好為基礎的法則。強度指標(SI)是用來辨識有意義的屬性集合，直接使用屬性集合產生演算法提供的結果可能會導致有許多是沒有意義的條件屬性集合。具有較高強度指標值的屬性集合比較低強度指標值的更佳。但必須注意屬性集合的比較只限於在屬性集合中同一決策屬性以及一些選定的屬性。

三、 減量技術

在資料探勘，減量技術提供了一種方法，解決從一個動態資料庫中刪除物件的問題，不需要執行 DM 演算法，與增量技術相似(Liu et al.)。在現實中資料刪除是資料庫管理中常見的執行活動(Zhang et al., 2007)，例如，當交易發生很長一段時間，舊的交易必須從資料庫中刪除，使資料探勘能夠正確反映目前情形(Sohel and Rahman, 2004)。分散式資料庫在固定時間間隔被新資料區塊所更新(Otey et al.)，這些資料庫的變化隨著時間通過必要的資料庫刪除操作而發生(Zhang et al., 2007; Zhang et al., 2009)，當某些交易在資料庫中被刪除或是修改，資料庫的內容被更新，則此資料庫稱為被更新資料庫。更新資料庫的探勘被稱為遞減式探勘(Zhang et al., 2009)，顯然是低效能且費時去重新運行資料探勘演算法來分析資料庫所有包含增量資料和原始資料(Sohel and Rahman, 2004)，所以減量式規則提取演算法允許通過比較從資料來源提取的規則以及在更新資料庫被刪除的資料，資料庫發生刪除的更動為資料庫管理所必要的(Zhang et al., 2007)，可能因為資料儲存很長一段時間或是因為資料庫使用國內犯罪率來提取法則(Zhang et al., 2007; Sohel and Rahman, 2004)，如果是返回購置的秩序則應該被刪除。

然而，很少有研究專門討論資料探勘的減量式技術，特別是在約略集合領域。張(Zhang et al., 2009)等人提出的遞減式演算法，名為 DUA，當資料從資料庫刪除的發現模式。Tasoulis 和 Vrahatis(2005)提出了一種 k-windows 演算法可以有效且高度辨別模式結構的變化，以追蹤動態變化的資料庫集群模式的演變，不需要額外的計算成本。因此，有效解決在動態資料庫刪除資料庫相關的問題是有需要的。

歸納此研究主要的理由：

- 1 傳統的約略集合不能產生優先順序的法則，所以不能產生更有意義的一般規則(Sun et al.)。相反的，以前的約略集合描述不確定性的機制是不有效的。因此，由約略集合產生的法則往往是不穩定且分類精準度低的(Wang and Li, 2004)。
- 1 大多約略集合方法不考慮動態資料庫的問題(Fan et al., 2009; Ahn et al., 2000)，特別是資料刪除的部分(Zhang et al., 2009)。
- 1 傳統的演算法很顯然是低效能且耗時的(Sohel and Rahman, 2004)，因為必須重新執行資料探勘演算法分析整個資料庫，包含增量資料和原始資料。
- 1 為了解決減量的問題，提供分配重量的靈活性並降低開發複雜數學模組的負擔，一個有效的啟發式演算法是非常有益的。

一個基於約略集合理論的減量式替代規則提取演算法在第三節中提出。

參、方法

一、 被提出的以法則歸納為基礎減量式約略集合的結構

被提出的演算法是基於 Pawlak(Pawlak, 1991)提出的屬性集合產生以及曾等人 (Tseng et al.)提出的替代法則提取演算法,所提出的方法以修改部分原來的法則集合來更新法則集合。由於大多數資料庫物件常常在變化(例如物件往往被增加、刪除、更新),在實際可行的應用中必須為能夠應付物件變化的方法(Zhong et al.)。由於這種方法是減量的,有效的處理物件的變化。表一表示五個物件變化的情形:

表一 減量式演算法案例結構

		Generated new Rule	Replaced old Rule	Deleted old Rule
Case 1		---	---	---
Case 2		---	---	V
Case 3		---	V	---
Case 4		V	---	---
Case 5	Case 5.1	V	V	---
	Case 5.2	V	---	V
	Case 5.3	---	V	V
	Case 5.4	V	V	V

Case I 物件的減量不會跟原來的法則引起衝突,且原本的法則可以支配新的物件。

Case II 物件的減量不會跟原來的法則引起衝突,但原本的法則不能支配新的物件。

Case III 物件的減量會跟原來的法則產生衝突,但可以由其中的一個屬性集合/法則來支配。

Case IV 物件的減量會跟原來的法則產生衝突,且新物件無法被任何一個原始的屬性集合支配。

Case V 物件的減量會跟原來的法則產生衝突,且新物件的特性和其中一個原始物件相同,但有不同的輸出。

二、 減量式演算法的程序

演算法主要內容如下: 首先,所有參數都設定為空值,然後挑選被刪除的物件且將號碼設為 *numdel*。第二,如果任何 *numdel* 在列集合 D,該 *numdel* 被移除,如果任何列集合為空集合,那麼屬性集合使用 Pawlak (1991) 屬性集合產生程序來產生新的屬性集合 $D(X_i)A_j$,增加到屬性集合表 (R)。第三,鑑別且刪除可能被 R 的 *numdel* 影響的屬性集合。

新的 *temp* 值, *Tnew*, 然後與舊的 *temp* 值相比較,告知已決定是否法則應該再次被提取。如果 *Tnew* 與告知的不同,那麼法則根據 AREA 再次被提取。根據法則集合的減量式演算法程序說明如下:

(一) 符號:

U : 物件的有限集合
 A : 屬性集合
 d : 屬性決策集合
 i : 物件索引
 j : 屬性索引
 n : 屬性集合索引
 l : 新等級的值
 L : 原本等級的值
 q : 物件資料的數量
 r : 屬性號碼
 k : 在屬性集合表內屬性集合號碼
 S : 案例號碼，由決定性的屬性和 U_i 決定
 T_{new} : SI 排列號碼，根據在表中新的屬性集合的案例號碼 S
 T_{old} : SI 排列號碼，根據在表中舊的屬性集合的案例號碼 S
 $Temp$: 紀錄是否 SI 的順序是否改變且不同於原本的決定
 X_i : 物件號碼，例如：物件 $X_i \in 1 =$ 物件 1
 a_{ij} : 物件 X_i 的第 j 個屬性
 $numdel$: 刪除物件的號碼，例如：如果刪除的物件是物件 1，那麼 $numdel=1$
 $I()$: 表中原始資訊
 $I(X_i)A_j$: 在物件 X_i ， I 列中第 j 個屬性，例如 $I(X_i)A_j$ 是表中原始資訊裡 X_i 的 A_j 列集合
 $D()$: 在表中每個屬性的差集, A_j ，(每個物件的相同類別)，和屬性， d (每個物件的相同類別其對應的決策)
 $D(X_i)A_j$: 物件第 j 個屬性在差集表的列，例如 $D(X_i)A_2$ 是物件 1 的屬性 A 在差集表的列集合
 E : 延伸表
 R : 表中的屬性集合
 F : 表中的最終法則
 F_{old} : 從舊法則挑選出來的法則(屬性)集合
 F_{new} : 從新法則挑選出來的法則(屬性)集合
 $F_{generate}$: 從產生的決策挑選出來的法則(屬性)集合
 $F_{replace}$: 從更換的決策挑選出來的法則(屬性)集合
 F_{delete} : 從刪除的決策挑選出來的法則(屬性)集合
 OC : 物件基數列，例如 $I(numdel)oc$ 是 $numdel$ 的物件基數列在表中原始的資訊
 MO : 合併物件列的編號， e.g., R_{MO} 是在表中屬性集的合併物件列
 SO : 支援物件列的編號， e.g., F_{SO} 是在表中最終決策表的支援物件列

(二) 解決程序:

Input: The number of a removed object, $numdel$.

Output: The set of decision rules and alternative rules, F_{new} .

Step 0 Initialization

- (i). When an object is removed, set the object number to $numdel$.
- (ii). Set $S = 1$, $l=1$, $Fold =$ original rules set, $F_{new} = \emptyset$, $F_{generate} = \emptyset$,
 $F_{replace} = \emptyset$, $F_{delete} = \emptyset$.

Step 1 Check if there is any $numdel$ in sets of columns in D and remove $numdel$.

For $i = 1$ to q

For $j = 1$ to r

If $numdel \in D(X_i)A_j \ \&\& \ D(X_i)A_j - numdel == \text{empty}$

go to step 1.1.

Else go to step 2

End If

Endfor

Endfor

Step 1.1 Apply the reduct generation procedure of Pawlak [17], to generate a new reduct of $D(X_i)A_j$.

Step 1.2 Check whether the new reduct exists in the R .

For $i = 1$ to k

If new reduct $\in R_i$

go to step 1.2.1.

Else go to step 1.2.2.

End If

Endfor

Step 1.2.1 Add the new reduct to R .

Step 1.2.2 Merge the new reduct with the identified original reduct, into R . The new reduct of object number joins RMO and the cardinality is also added to ROC.

Step 2 Check whether the new reduct is better than the original.

For $l = 1$ to L

If a new reduct is not yet generated,

Go to step 2.1.

Else go to step 3.

End If

Endfor

Step 2.1 Check whether the intersection of A_j and $D(X_i)$ is empty.

If it is empty

Go to Step 1.1.

Else go to step 2.

End If

Step 3 Find any reduct that is possibly affected by *numdel*, in *R*:
 For $n = 1$ to k
 If $numdel \in RMO(n)$
 Go to step 3.1.
 End If
 Endfor
 Go to step 3.2.

Step 3.1 When the *numdel* is removed from RMO, subtract $I(numdel)OC$ from ROC.

Step 3.2 Re-compute the strength index, SI. Sort SI according to case number, *S*, and *Tnew* is stored, with the order, for each reduct.

Step 4 Check whether the new order has not changed, in *R*.
 $Temp=0$
 For $n = 1$ to k
 If $Tnew(n) \neq Told(n)$
 $Temp=1$
 End If
 Endfor

Step 5 Decide whether to re-extract the rules, according to the value of *Temp*.
 if $Temp = 1$, the rules must be re-extracted, so go To Step 5.1, else go To Step 6.

Step 5.1 Re-extract the rules, according to AREA
 Set *Fold*=the original rules , *Fgenerate*=the new rules ,
Freplace=the replaced rules , *Fdelete*=the deleted rules.

Step 6 Print $Fnew = \{Fold + Fgenerate + Freplace - Fdelete\}$.

三、 AREA 與 DAREA 的比較

下面表二和表三計算時間的折線圖，顯示 AREA 和 DAREA 以 CPU 計算時間的比較。|dbl|的值分別為 100，200，400，500，700 和 1000。

表二 使用 AREA 重新提取規則(當特性=5)

DB	1% DB	10% DB	40% DB	70% DB
100	286	246	171	134
200	721	595	361	185
400	2309	1815	959	400
500	3397	2685	1326	547
700	6462	5088	2501	895
1000	13257	10233	4635	1453

表三 使用 DAREA 重新提取規則(當特性=5)

DB	1% DB	10% DB	40% DB	70% DB
100	112	110	114	81
200	173	189	203	91
400	403	360	226	143
500	423	385	238	141
700	667	594	350	173
1000	1120	866	481	196

圖 1 所示，DAREA 的性能比 AREA 更好對每個資料刪除的比率。顯然的 CPU 對 AREA 計算時間劇烈的增加當物件刪除數量從 100 到 1000，但 CPU 對 DAREA 的計算時間為逐漸增加。

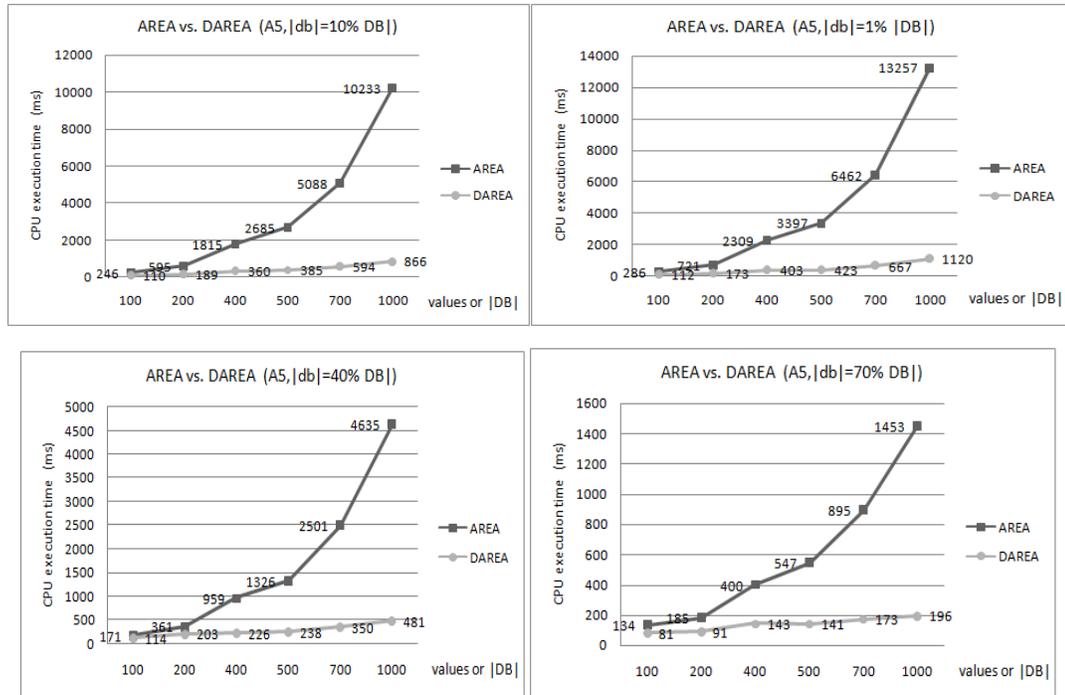


圖 1 CPU 對 AREA 和 DAREA 的計算時間

圖 2 顯示對 DAREA 和 AREA 計算時間的比率，CPU 對 AREA 的執行時間除以 CPU 對 DAREA 的執行時間，例如當 DB 70% 且物件數量為 1000，AREA 的時間為 1453、DAREA 的時間為 196，則其比率為 7.41，表示 DAREA 的效率優於 AREA 時間的 7.41 倍。結果顯示從資料庫刪除資料 DAREA 比 AREA 的比率更好，且增加計算的效率當物件的數量從 100 增加到 1000。

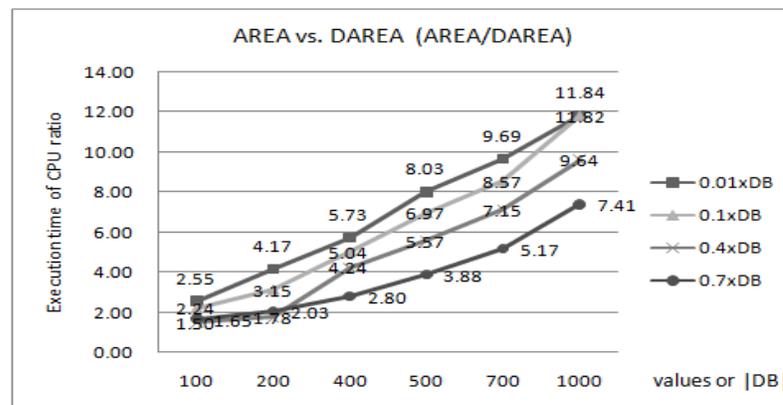


圖 2 AREA 對 DAREA CPU 執行時間比率的比較

如果 db 分別 = 1%、10%、40% 且 DB 70%，那麼圖中代表 AREA 對 DAREA 的 CPU 計算時間比率。圖 2 所示，當物件數量是 700，DAREA 效率比 AREA 優於 5 至 9 倍以上；當物件數量是 1000，DAREA 效率比 AREA 優於 7 至 11 倍以上。

肆、案例

A 餐廳想要訂定對不同族群的決策來提高來客率，透過了解顧客的資料以及評價了解顧客偏好且應用在不同的決策中，找出核心特性。

在此案例中，共有 13 個特性，列於表四中。特性 1 為顧客的性別；特性 2 為顧客不同類別年齡；特性 3 為顧客職業，包含上班族、勞工、學生；特性 4 表示顧客的經濟能力；特性 5 為顧客在點餐時是否感到有效率且愉悅的；特性 6 為內部裝潢，如桌椅是否舒適、內部環境是否乾淨；特性 7 為用餐時的服務，服務員是否能夠提供符合顧客的服務，用餐時是否感到舒適方便；特性 8 為點餐後到上餐的時間間隔速度；特性 9 為顧客對餐廳餐點價格的感覺；特性 10 為過去消費過的顧客對該餐廳的評價；特性 11 為餐廳所使用的食材新鮮程度；特性 12 為顧客食用餐點時是否感到滿意；特性 13 為當有突發狀況，如餐點中有異物等情形，餐廳的處理態度。產出為顧客對該餐廳的評價。表五為 10 名顧客的特性資料。

表四 餐廳顧客評價特性表

	屬性名稱	0	1	2	3
F1	性別	男	女	-----	-----
F2	年齡	20 歲以下	21 歲-30 歲	31 歲以上	-----
F3	職業	上班族	勞工	學生	其他
F4	收入	小於 30,000	30,000-70,000	70,000 以上	-----
F5	點餐服務	Low	Medium	High	-----
F6	內部裝潢	Low	Medium	High	-----
F7	用餐服務	Low	Medium	High	-----
F8	上菜速度	Low	Medium	High	-----
F9	價格	Low	Medium	High	-----
F10	過往評價	Low	Medium	High	-----
F11	食材新鮮度	Low	Medium	High	-----
F12	餐點評價	Low	Medium	High	-----
F13	問題處理態度	Low	Medium	High	-----
O	顧客滿意度	Low	Medium	High	-----

表五 10名顧客資料表

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	O
1	0	1	0	1	1	1	1	2	1	2	1	1	1	2
2	0	1	3	0	1	2	1	2	2	2	2	0	1	2
3	1	0	1	2	1	1	1	0	1	2	0	2	1	1
4	0	0	1	1	1	1	2	1	2	1	1	1	0	1
5	1	2	0	1	2	2	2	1	1	2	1	1	2	2
6	0	2	1	1	1	2	2	1	1	2	2	1	1	1
7	0	0	2	0	1	2	1	0	1	1	1	0	0	1
8	1	2	0	2	1	1	2	2	2	2	2	2	1	2
9	1	2	3	1	2	1	2	2	2	2	1	1	1	2
10	1	1	0	1	1	1	2	1	2	1	2	1	1	1
Normalized weight	0.63	0.68	0.78	0.79	0.79	0.78	0.79	0.63	0.79	1	0.95	0.95	0.63	

四、 計算結果

歷史的資料根據比例被分為兩個資料集合:0.632 訓練集合以及 0.368 測試集合 (Kusiak, 2001)。在這個案例中根據專家的專業知識決定最終屬性集合 7%的門檻值，如表六。

表六 原始資料經演算法計算的屬性集合表

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	O
1, 8	x	x	x	x	x	x	x	2	x	x	x	x	x	2
2	x	x	1	x	x	x	x	x	x	x	x	x	x	1
5, 9, 11, 12	x	x	x	x	x	x	x	x	x	1	x	x	x	1
3, 7	x	x	x	x	2	X	x	x	x	x	x	x	x	2
4, 15	x	x	x	x	x	X	x	x	x	x	x	x	2	2
14, 16	x	x	x	x	x	X	x	x	x	x	1	x	1	2
6	x	x	1	x	x	X	x	x	x	1	1	x	x	2

IF (上菜速度= 2) THEN (顧客滿意度= “2”)

IF (職業= 1) THEN (顧客滿意度= “1”)

IF (過往評價= 1) THEN (顧客滿意度= “1”)

IF (點餐服務= 2) THEN (顧客滿意度= “2”)

IF (問題處理態度= 2) THEN (顧客滿意度= “2”)

IF (食材新鮮度= 1) AND (問題處理態度= 1) THEN (顧客滿意度= “2”)

IF (過往評價= 2) AND (食材新鮮度= 1) THEN (顧客滿意度= “2”)

舉例而言，第一個法則(I)顯示如果上菜速度快，那麼顧客評高。

根據時間推移，某些顧客的資料可能因為長期未光顧造成無法提供參考價值

而必須被刪除。由於目前的預算限制和約束，這個演算法是非常高效的重新執行方式，以所有資料得到最終的決策法則。DAREA 演算法應用在下列處理物件刪除的問題中。表七顯示第一個物件刪除後得到新的法則

表七 第一個物件刪除

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	O
0	2	1	1	1	2	2	1	1	2	2	1	1	1

在表七，物件[0 2 1 1 1 2 2 1 1 2 2 1 1, 1] 的刪除並沒有和現有的法則衝突，且不能被任何原來的法則支配，因此這個案件可以被列為案件二。這種情形下，必須重新計算物件的屬性集合，產生新的法則 IF (用餐服務=2) AND (上菜速度=2) THEN (顧客滿意度=2)

五、 討論

在這個個案中，研究目的是要依據題取得法則來說明顧客的個人資料以及餐廳所提供的服務與顧客評價之間的關係。最終的結果集中於使用法則將背景和滿意度相對於顧客的喜好分類，因此研究結果有助於餐廳建立一個有競爭力的促銷活動。

這個解決方法還提供餐廳案例研究中在未來的資料需要使用和產生新的解決方案以建立新的促銷策略，透過案例發現上菜速度以及點餐服務對顧客的滿意度有相當大的關連，因此餐廳可以在點餐時加派人手或是將點餐流程電腦自動化，加快點餐速度，也可以多加訓練點餐櫃檯人員；而在問題處理態度方面，不僅僅是必須訓練員工的處理問題能力，也可以透過一套 SOP 的流程來因應所以可能的問題。第一個刪除的物件不與任何舊法則造成衝突但也不能被舊法則所支配，因此新物件不能對舊法則造成任何改變且對結果建立一個新的法則，意味著市場的性質沒有改變甚至收集以及分析新的重大物件。

伍、 結論

此研究以傳統的約略集合方法和減量式技術，提出了前人研究的弊端。曾等人(Tseng et al.)在 AREA 的基礎上提出了減量式的替代規則提取演算法以解決上述物件被刪除時必須重新計算的弊端。案例顯示了餐廳的服務影響了顧客的評價，餐廳透過此分析可以了解所需加強的部份以及訂定出促銷策略以吸引顧客，提升餐廳的競爭力，這對餐廳的營運者來說是非常有益處的。以餐廳的例子說明以表示如何找到解決方法，最後證明該方法優於傳統的方法。總結如下：

1. 在動態資料庫中，資料刪除在資料庫的管理動作中是經常的操作，從動態資料庫中刪除一個物件時，不必重新運作 AREA 來提取法則，該方法更新屬性集合使用差集因而減少了計算的時間。
2. 被建議的減量式替代規則提取演算法是在曾等人(Tseng et al.)所提出的 AREA 演算法基礎上，所以該演算法的 o 充分利用了曾的方法((Tseng, 1999)，根據屬性集合的強度指標確定以偏好為基礎的法則，並保證決策表

的分類是可信的。

3. AREA 可以選擇超過一個 SI 的最大值。因此，此研究處理了不完整規則的問題。

參考文獻

1. T.-P. Hong, C.-Y. Wang, and S.-S. Tseng, "An incremental mining algorithm for maintaining sequential patterns using pre-large sequences," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7051-7058, 2011.
2. S. Zhang, J. Zhang, and C. Zhang, "EDUA: An efficient algorithm for dynamic database mining," *Information Sciences*, vol. 177, no. 13, pp. 2756-2767, 2007.
3. F. Crespo, and R. Weber, "A methodology for dynamic data mining based on fuzzy clustering," *Fuzzy Sets and Systems*, vol. 150, no. 2, pp. 267-284, 2005.
4. S. Zhang, and L. Liu, "Mining Dynamic Databases by Weighting," *Acta Cybernetica*, vol. 16, no. 1, pp. 179-205, 2003.
5. S. Zhang, J. Zhang, and Z. Jin, "A decremental algorithm of frequent itemset maintenance for mining updated databases," *Expert Syst. Appl.*, vol. 36, no. 8, pp. 10890-10895, 2009.
6. [6] F. A. Sohel, and C. M. Rahman, "Association Rule Mining in Dynamic Database using the Concept of Border Sets," *Asian Journal of Information Technology*, vol. 3, pp. 508-515, 2004.
7. M. E. Otey, C. Wang, S. Parthasarathy et al., "Mining Frequent Itemsets in Distributed and Dynamic Databases." pp. 617-620.
8. A. Kusiak, "Feature transformation methods in data mining," *Electronics Packaging Manufacturing, IEEE Transactions on*, vol. 24, no. 3, pp. 214-221, 2001.
9. Z. Pawlak, "Rough Sets," *International Journal of Information and Computer Sciences*, vol. 11, no. 5, pp. 341-356, 1982.
10. S. Greco, B. Matarazzo, and R. Slowinski, "Rough sets theory for multicriteria decision analysis," *European Journal of Operational Research*, vol. 129, no. 1, pp. 1-47, 2001.
11. Q. Yuhua, L. Jiye, and D. Chuangyin, "Incomplete Multigranulation Rough Set," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 2, pp. 420-431, 2010.
12. P. V. Gorsevski, and P. Jankowski, "Discerning landslide susceptibility using rough sets," *Computers, Environment and Urban Systems*, vol. 32, no. 1, pp. 53-65, 2008.
13. P. Pattaraintakorn, and N. Cercone, "Integrating rough set theory and medical applications," *Applied Mathematics Letters*, vol. 21, no. 4, pp. 400-403, 2008.
14. J.-Y. Shyng, F.-K. Wang, G.-H. Tzeng et al., "Rough set theory in analyzing the attributes of combination values for the insurance market," *Expert Systems with Applications*, vol. 32, no. 1, pp. 56-64, 2007.
15. T.-L. B. Tseng, "Quantitative Approaches for Information Modeling," University of Iowa, Iowa city, 1999.
16. T.-L. B. Tseng, C.-C. Huang, and J. C. Ho, "Autonomous Decision Making in Customer Relationship Management: A Data Mining Approach."

17. Z. Pawlak, "Rough Sets: Theoretical Aspects of Reasoning about Data," *Rough Sets: Theoretical Aspects of Reasoning about Data*, Boston.: Kluwer Academic Publishers, 1991.
18. W. P. Ziarko, and C. J. V. Rijsbergen, *Rough Sets, Fuzzy Sets and Knowledge Discovery*, New York: Springer-Verlag, 1994.
19. Y.-N. Fan, T.-L. Tseng, C.-C. Chern et al., "Rule induction based on an incremental rough set," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11439-11450, 2009.
20. C.-M. Sun, D.-Y. Liu, S.-Y. Sun et al., "Containing order rough set methodology." pp. 1722-1727.
21. T. L. Tseng, C. C. Huang, F. Jiang et al., "Applying a hybrid data-mining approach to prediction problems: a case of preferred suppliers prediction," *International Journal of Production Research*, vol. 44, no. 14, pp. 2935-2954, 2006/07/15, 2006.
22. Y. Liu, C. Xu, and Y. Pan, "An Incremental Rule Extracting Algorithm Based on Pawlak Reduction." pp. 5964-5968.
23. D. K. Tasoulis, and M. N. Vrahatis, "Unsupervised clustering on dynamic databases," *Pattern Recogn. Lett.*, vol. 26, no. 13, pp. 2116-2127, 2005.
24. Q. H. Wang, and J. R. Li, "A rough set-based fault ranking prototype system for fault diagnosis," *Eng. Appl. Artif. Intell.*, vol. 17, no. 8, pp. 909-917, 2004.
25. B. S. Ahn, S. S. Cho, and C. Y. Kim, "The integrated methodology of rough set theory and artificial neural network for business failure prediction," *Expert Systems with Applications*, vol. 18, no. 2, pp. 65-74, 2000.
26. N. Zhong, J.-Z. Dong, S. Ohsuga et al., "An incremental, probabilistic rough set approach to rule discovery." pp. 933-938.

Yen-Fen Kao

Department of Information Management, National Chi Nan University
s98241020@ncnu.edu.tw

Chun-Che Huang

Department of Information Management, National Chi Nan University
cchuang@ncnu.edu.tw

Ying-Ling Hsieh

Department of Information Management, National Chi Nan University
s98213544@ncnu.edu.tw

ABSTRACT

A dynamic database is a common type of business databases. Data deletion operations are also a frequent feature of database management activities. Unfortunately, most existing data mining (DM) algorithms assume that the database is static and that updating a database requires re-computation of all the patterns, to enable rule extraction. Of the DM techniques available, the rough set (RS) approach is a knowledge discovery tool that can be used to help identify logical patterns hidden in massive data. However, traditional RS approaches cannot produce rules with preferential order and often lacks focus, generates too many rules and cannot guarantee that the decision table is credible. This study proposes a DAREA (Decremental Alternative Rule-Extraction Algorithm), to address the issue of objects removed from the database and to generate preference-based rules, according to a strength index, specifically for the case wherein the desired reducts are not necessarily unique. The algorithm does not need to re-compute rule sets that can quickly generate and complete rules, from the very beginning. This solution approach applies to restaurant's service and customer basic information on Customer Satisfaction. This case study is devised, to validate that the proposed approach is superior to the traditional RS approach.

Keywords: Dynamic databases, Rough set approach, Decremental mining, Decremental object, Rule induction

