

屬性離散化粗集分類器預測企業獲利率

Forecasting business profitability using rough sets classifier based on Attribute-discretization

張陸賢

雲林科技大學資訊管理所研究生
g9523202@yuntech.edu.tw

鄭景俗

雲林科技大學資訊管理所教授
chcheng@mis.yuntech.edu.tw

摘要

本研究的目的有二：（1）提出屬性離散化粗集演算法的資料探勘技術；（2）以資源基礎角度，評估企業於獲利率之表現，並建立自動化的預測系統，將能有效減少成本與人力的投入。在實證方面，以U公司實際的交易資料庫為研究對象建立模式，並運用資料探勘技術進行資料分析。亦即首先運用決策樹對資料庫屬性作離散化來強化粗集分類器，可提高其對獲利率預測的正確率，並萃取出分類規則。本研究提出的新方法其正確率為85.73%，比其他方法的正確率高且規則數較少。

關鍵詞：資料探勘、資源基礎、粗集理論、離散化、獲利率

Abstract

This study has two objectives : (1) This study proposes attribute discretization of database attributes by decision trees, which enhances rough sets classifier. (2) This study basing on the resource-based view evaluates the profitable efficiency, and investigates how the resources of the firms influence the activities generating more profit. If an automated classification system can be implemented, both the input of cost and manpower required for the inventory can be reduced.

In practice, the data used for the experiment is one industry of electronics manufacturing in Taiwan. This study proposes attribute discretization of database attributes by decision trees, which enhances

rough sets classifier to increase accurate rate of profitability forecasting. From comparison of accurate rate, the proposed method is 85.73%, which is higher than the listing methods, and with less number of rules.

Keyword : data mining、resource-based、rough set theory、attribute discrete、business profitability

1.前言

本研究以資源基礎角度對電子製造業進行經營效率之評估，評估企業以所擁有的資源資產於獲利率之表現，進而擷取出能有效預測決策的資訊，為企業在瞬息萬變的環境中提供未來發展方向之參考。

資料探勘(Data Mining)已是很成熟的技術，但每種方法都不是最完美，都各有其優缺點，並存在著某些瓶頸[10]。例如決策樹，面臨的問題就是，若將不相關的屬性當成訓練資料來學習，則有可能造成結果的偏差，則準確性就有可能會不穩定[5]；例如類神經網路，是以替代方式更新鍵結值與閾值，計算量大，相當耗費電腦資源。類神經網路訓練的過程中無法得知需要多少神經元個數，太多或太少的神經元均會影響系統的準確性，因此往往需以試誤的方式得到適當的神經元個數[5]。例如粗集方法的數學基礎是集合論，難以直接處理連續的屬性，而現實資訊表中連續屬性是普遍存在的。因此連續屬性是粗集理論實用化的難點。[12]。

有鑑於此，本研究發展一套新的方法，能改善或解決其他技術以往所面臨過的問題。

2. 文獻探討

本節介紹資源基礎、獲利率、資料探勘的分類方法及粗集理論相關文獻。

2.1 資源基礎

Porter(1991)認為資源基礎(Resource-based)的探討重點在分析企業內部資源是否有效開發利用及如何取得、累積與培育資源。換言之，資源基礎乃在解釋企業所擁有的獨特競爭力來自企業內部的那些資源[16]。鄭健雄(1998)認為資源基礎強調以內部資源取向為主，重視企業內部資源優弱勢分析，以發掘及培養企業自身之長期競爭優勢。吳存翰(2001)綜合各學者之研究，對資源基礎理論之定義提出說明，資源基礎理論乃是企業經營者必須辨識組織本身所擁有的各種資源，並運用專長能力加以組合、管理，促使內部某特定資源發展成為長期且持續的獨特競爭能力。

2.2 獲利率

營業利益率(獲利率)為反映一家企業本業獲利能力的指標。由於營業利益的計算已將過程中所耗用的一切成本均列入考量，因此在關注本業的獲利能力時，營業利益率便為重要指標。與毛利率概念相同，當營業利益率發生變化時，通常代表著企業體質發生轉變，例如新產品效益出現、規模經濟量顯現或是管理能力的提升等等，這些企業體質轉佳的效益都會顯現在營業利益率數字中。

在計算上，營業利益率 = (營業收入 - 營業成本 - 營業費用) ÷ 營業收入 × 100%，比率越高，表示企業的獲利能力越高[3]。

2.3 資料探勘分類方法

利用資料探勘技術，可以由龐大的資料庫中，得到其中所隱含的資訊、知識，因此資料探勘是目前在產學相當受重視且被廣泛利用的一門學科。

資料探勘的應用相當廣泛，如應收帳款之呆帳預警模式[5]、目標行銷之應用研究[6]、產險行銷之個案研究[2]、建構脂肪肝預測模式[4]。

2.3.1 資料探勘的定義

表 1 各學者對資料探勘所作的定義

人名及年代	對資料探勘所作的定義
Friedman(1997)	發現資料內的特徵 (pattern) 及關係，強調於大量觀察資料，包括資料庫管理、人工智慧、機器學習、特徵確認和資料可視化等幾個領域
Berry and Linoff (1997)	為了發現隱藏在資料中有意義的型樣或規則性，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。
Hand (1998)	其為一種輔助的分析程序，針對於資料庫中有趣的及有價值且未發現的關係
Berson et al. (2000)	自動偵測資料庫中的特徵
Kim et al. (2000)	提煉未知且可能有用的知識來幫助決策的一系列流程。

綜合以上學者的觀點，皆認為資料探勘是在龐大的資料庫中找尋出具有價值的隱藏訊息，並且將這些訊息加以分析[7,9]。

2.3.2 分類法

分類法(Classification)是利用訓練資料中的各個特徵屬性，找出其中的分類規則，建立起分類器(Classifier)，再利用與訓練資料具有相同屬性的資料，但內容不完全相同的資料，對分類器進行分類測試，若是分類結果的正確率可以達到使用者的要求標準，則此分類器便可用來對新的資料作為分類之用。下面介紹主要的相關分類法：

(A) ID3與C4.5演算法

ID3演算法被廣泛用來處理離散型資料。ID3演算法最重要的特色就是能兼顧高分類正確率以及降低決策樹的複雜度。因此，使用ID3 可以提升決策樹的可理解性。Quinlan 於1993年提出C4.5演算法，其基本理論是改良他在1979 年所提出的ID3演算法。C4.5演算法是先建構一棵完整的決策樹，再針對每一個內部節點依使用者定義的錯誤預估率(Predicted Error Rate)來作決策樹修剪的動作[1]。

(B) 類神經網路分類法

類神經網路(neural network)是人工智慧的一種，也是醫學、數學、資訊工程與電子工程研究的結合應用。類神經網路系統模擬大腦神經細胞的運作方式，由一些高度連結的處理單元(稱做節點或是神經元，neuron)組成動態的運算系統。類神經網路會透過不斷地自我調整，使得輸入的資訊在經過神經元的運算之後能得到預設的輸出結果。

類神經網路的運作分成兩個階段：

1. 訓練階段：調整網路內部各節點連結的權重值，使得輸入值經過網路計算之後能得到目標的輸出值。
2. 測試階段：驗證網路的準確度或是利用訓練完成的網路進行預測[11]。

2.4 粗集理論

粗集理論在1982年時由波蘭Zdzislaw Pawlak所提出。由於在應用粗集理論進行分析時，不須服從任何的假設，這使得粗集理論的應用彈性更大。而且因為粗集理論可以挖掘出資料屬性彼此之間的關係，這也使得粗集理論被應用在許多領域，例如決策分析(Decision Analysis)、資料庫知識發掘(Knowledge discovery from databases)、專家系統(Expert Systems)、決策支援系統(Decision Support Systems)中的模式識別(Pattern Recognition)等[16]。

執行粗集合理論的第一個步驟就是由所要分析的資料所組成決策表；接著找出屬性之間的不可區分關係；接著以屬性刪減及尋找核心找出所有可能的最小屬性集；然後再以上、下限近似集作為資料分類的依據；最後就是找出最有分類效率的決策規則。在完成物件資料定義之後，可實施資料特徵值化簡，因資料所包含之屬性不全然對實驗結果會產生重大影響，因此如能加以化簡則可有效降低複雜度[12,13,14]。

3.研究方法

本研究採取模式驗證方法，使用所提出強化粗集技術演算法的資訊探勘方法，針對個案公司之實際資源基礎相關資料為範例進行驗證，來預測企業的獲利率。

3.1 研究架構

本研究依提出的演算法，可分為五個部分來進行，如圖1。

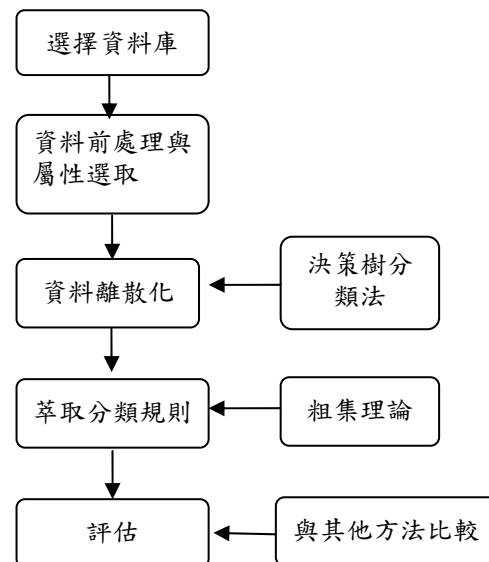


圖1 研究架構

3.2 演算步驟

為方便計算與進行程式化，將提出的模式給予步驟化如下：

步驟1 選擇資料庫

選定以個案 U 公司的資源基礎部份作為主要探討對象，對其進行經營效率之評估，評估企業以所擁有的資源資產於獲利率之優劣表現。

步驟2 資料處理

1. 從資料庫中選定員工數、採購額、投資額、銷售額與銷售量等作為條件屬性，並轉登錄於 EXCEL 電子檔。
2. 進行資料換算處理，並自行定義出資料比例。
3. 將資料庫中獲利率原本的數值型資料，依切割值為 0 及 2.65，切割為三個區塊，將所屬的區塊以符號取代。(以 Pool、Fair 及 Good 三個等級取代)。

步驟3 資料離散化

利用決策樹演算法，對資料作離散化。

1. 利用決策樹分類，找出資料分割值(如圖 2

所示)。

2. 利用找出的切割值，將連續性的數值資料切割成數個區塊。
3. 將資料庫中原本的數值型資料，依所屬的區塊以符號取代。

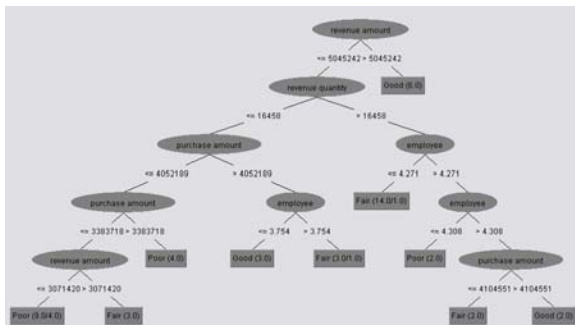


圖 2 利用決策樹演算法對資料表做分類所產生的決策樹圖

步驟 4 萃取分類規則

利用粗集理論對已離散化的資料，萃取出分類規則。將已明確區分的資料屬性，利用 LEM2(Learning From Examples, Module 2) [16]演算法產生分類規則。

步驟 5 驗證與比較

再以資料庫中轉出的尚未符號化之原始數據，以 Decision Tree、Artificial Neural Networks 及 Rough Set 的方法進行實驗，再比較探討彼此之間的差異與原因，最後提出適當之建議。

4.個案驗證與分析

在此章節裡，首先簡單介紹個案公司背景與資料前處理的過程；後半段則是進行粗集理論的實際驗證及其他資料探勘技術的實做，並就所有各項技術的實驗數據結予以分析比較。

4.1 個案公司簡介

個案 U 公司為全球設計製造服務(DMS)領導廠商，提供完整 DMS 解決方案，主要涵蓋資訊產品(Computing)、通訊產品(Communication)、消費電子(Consumer Electronics)與汽車電子(Car Electronics)等四大服務領域。除此之外 U 公司

更結合先進微電子構裝技術上的專業與經驗，提供從模組、主機板至系統之產品與服務，具備短、小、輕、薄及省電特性之設計製造能力。

自 1976 年成立至今，基於多年與世界一流客戶成功合作與共同成長的世界級經營管理經驗，以尖端產品研發及製程能力，提供從設計到配銷的完整 D-to-D 服務 (Design to Delivery)，以其彈性製造服務與全球產品後勤運送的優勢，致力於服務大型、中型以及多樣客製化營運模式之品牌客戶。同時，藉由先進自動化設備、製程技術、工程研發能力及優良品質，搭配完善 IT 系統的整體性供應鏈，成為客戶設計與營運的延伸團隊及重要合作夥伴。

4.2 資料前處理

在資料前處理的作業上，首先使用 SQL/Plus 資料庫軟體工具，撰寫包含條件篩選及運算轉換的簡單處理語法，直接從個案資料庫擷取最原始的資料並轉存成純文字檔。之後再把該檔案匯入 Excel 套裝軟體中，針對屬性進行挑選過濾，將資料簡化只保留下必要有意義的欄位。

經過前處理後，整理出資源基礎彙總表(表 2)，計有 72 筆記錄，將做為後續獲利率預測、粗集演算法、決策樹的直接或間接的資料來源。

表 2 2001~2006 年度資源基礎彙總表

No	員工數	投資額	採購額	銷售額	銷售量	獲利率
1	3.479	6133534	2846506	2449858	6538	Pool
2	3.386	6133534	3283013	2825540	17453	Pool
3	3.331	6133534	4090518	6520522	3242	Pool
4	3.261	6133534	2647861	2437517	5907	Pool
5	3.261	6133534	2647861	2437517	5907	Pool
6	3.261	6133534	2647861	2437517	5907	Pool
7	3.261	6133534	2647861	2437517	5907	Pool
8	3.261	6133534	2647861	2437517	5907	Pool
9	3.261	6133534	2647861	2437517	5907	Pool
10	3.261	6133534	2647861	2437517	5907	Pool
11	3.261	6133534	2647861	2437517	5907	Pool
12	3.261	6133534	2647861	2437517	5907	Pool
13	3.261	6133534	2647861	2437517	5907	Pool
14	3.261	6133534	2647861	2437517	5907	Pool
15	3.261	6133534	2647861	2437517	5907	Pool
16	3.261	6133534	2647861	2437517	5907	Pool
17	3.261	6133534	2647861	2437517	5907	Pool
18	3.261	6133534	2647861	2437517	5907	Pool
19	3.261	6133534	2647861	2437517	5907	Pool
20	3.261	6133534	2647861	2437517	5907	Pool
21	3.261	6133534	2647861	2437517	5907	Pool
22	3.261	6133534	2647861	2437517	5907	Pool
23	3.261	6133534	2647861	2437517	5907	Pool
24	3.261	6133534	2647861	2437517	5907	Pool
25	3.261	6133534	2647861	2437517	5907	Pool
26	3.261	6133534	2647861	2437517	5907	Pool
27	3.261	6133534	2647861	2437517	5907	Pool
28	3.261	6133534	2647861	2437517	5907	Pool
29	3.261	6133534	2647861	2437517	5907	Pool
30	3.261	6133534	2647861	2437517	5907	Pool
31	3.261	6133534	2647861	2437517	5907	Pool
32	3.261	6133534	2647861	2437517	5907	Pool
33	3.261	6133534	2647861	2437517	5907	Pool
34	3.261	6133534	2647861	2437517	5907	Pool
35	3.261	6133534	2647861	2437517	5907	Pool
36	3.261	6133534	2647861	2437517	5907	Pool
37	3.261	6133534	2647861	2437517	5907	Pool
38	3.261	6133534	2647861	2437517	5907	Pool
39	3.261	6133534	2647861	2437517	5907	Pool
40	3.261	6133534	2647861	2437517	5907	Pool
41	3.261	6133534	2647861	2437517	5907	Pool
42	3.261	6133534	2647861	2437517	5907	Pool
43	3.261	6133534	2647861	2437517	5907	Pool
44	3.261	6133534	2647861	2437517	5907	Pool
45	3.261	6133534	2647861	2437517	5907	Pool
46	3.261	6133534	2647861	2437517	5907	Pool
47	3.261	6133534	2647861	2437517	5907	Pool
48	3.261	6133534	2647861	2437517	5907	Pool
49	3.261	6133534	2647861	2437517	5907	Pool
50	3.261	6133534	2647861	2437517	5907	Pool
51	3.261	6133534	2647861	2437517	5907	Pool
52	3.261	6133534	2647861	2437517	5907	Pool
53	3.261	6133534	2647861	2437517	5907	Pool
54	3.261	6133534	2647861	2437517	5907	Pool
55	3.261	6133534	2647861	2437517	5907	Pool
56	3.261	6133534	2647861	2437517	5907	Pool
57	3.261	6133534	2647861	2437517	5907	Pool
58	3.261	6133534	2647861	2437517	5907	Pool
59	3.261	6133534	2647861	2437517	5907	Pool
60	3.261	6133534	2647861	2437517	5907	Pool
61	3.261	6133534	2647861	2437517	5907	Pool
62	3.261	6133534	2647861	2437517	5907	Pool
63	3.261	6133534	2647861	2437517	5907	Pool
64	3.261	6133534	2647861	2437517	5907	Pool
65	3.261	6133534	2647861	2437517	5907	Pool
66	3.261	6133534	2647861	2437517	5907	Pool
67	3.261	6133534	2647861	2437517	5907	Pool
68	3.261	6133534	2647861	2437517	5907	Pool
69	4.102	7960755	4866727	4853030	22980	Good
70	4.038	7960755	4338160	4325950	20018	Good
71	3.963	7960755	4937133	4923238	24018	Good
72	3.843	7960755	4989305	4975264	28649	Good

4.3 決策樹演算法分類

利用決策樹演算法對資源基礎資料庫作分類，所產生的決策表如表3所示。

表 3 獲利率分類決策表

```

revenue amount <= 5045242
|   revenue quantity <= 16458
|   |   purchase amount <= 4052189
|   |   |   purchase amount <= 3383718
|   |   |   |   revenue amount <= 3071420: Poor
|   |   |   |   revenue amount > 3071420: Fair
|   |   |   |   purchase amount > 3383718: Poor
|   |   |   purchase amount > 4052189
|   |   |   employee <= 3.754: Good
|   |   |   employee > 3.754: Fair
|   |   revenue quantity > 16458
|   |   employee <= 4.271: Fair
|   |   employee > 4.271
|   |   |   employee <= 4.308: Poor
|   |   |   employee > 4.308
|   |   |   |   purchase amount <= 4104551: Fair
|   |   |   |   purchase amount > 4104551: Good
revenue amount > 5045242: Good
  
```

利用分類決策表找出銷售額(RevenueAmount)的切割值為3071420及5045242，員工數(Employee)的切割值為3.754、4.271及4.308，採購額(PurchaseAmount)的切割值為3383718、4052189及4104551，銷售量(RevenueQuantity)的切割值為164582。

4.4 預測獲利率之規則萃取

依提出的演算法萃取或利率規則計算如下：

- 1.離散化連續性的數值資料，將連續性的數值資料切割成數個區塊：利用決策樹找出的切割值，將銷售額(RevenueAmount)切割成3個區塊，將員工數(Employee)切割成4個區塊，將採購額(PurchaseAmount)切割成4個區塊，將銷售量(RevenueQuantity)切割成2個區塊。

將資料庫中原本的數值型資料，依所屬的區塊以符號取代如銷售額(RevenueAmount)以RA1~RA3取代，員工數(Employee)以EM1~EM4取代，採購額(PurchaseAmount)以PA1~PA4取代，銷售量(RevenueQuantity)以RQ1~RQ2取代。

- 2.利用粗集理論以粗集分類器對已離散化的資料，萃取出分類規則：將已切割成數個區塊並以文字取代的資料，分割為訓練集及測試集，其中訓練集佔原資料之66%，測試集佔原資料之34%，再利用LEM2 (Learning From Examples, Module 2) [16]演算法產生分類規則。重複抽樣執行10次其結果如表4所示。其平均正確率為85.73%，規則數為10.3條，部份規則如下：

- If purchase amount = PA3 & revenue amount = RA2 & Employee = EM4 **than** net margin rate = Good (獲利率：高)
- If revenue quantity = RQ2 & revenue amount = RA4 & employee = EM2 **Than** net margin rate = Fair(獲利率：中)
- If revenue amount = RA2 & purchase amount = PA2 & revenue quantity = RQ1 **Than** net margin rate = Poor (獲利率：低)

4.5 研究結果與其他分類演算法比較

將原始資料採用決策樹(Decision Tree)，類神經網路(Neural Network)及粗集理論(Rough Set)等演算法與本研究所提的方法(離散化後以LEM2 (Learning From Examples, Module 2) [16]萃取規則)所作之比較，其重覆測試10次的結果如表4。

表4 本研究與其他演算法所產生結果比較表

演算法	規則數量 (10次平均值)	正確率 (10次平均值)
決策樹	23	65.18%
類神經網路	N/A	63.75%
粗集理論	27.8	61.47%
本研究	10.3	85.73%

註：N/A代表無法以規則呈現

5.結論

本研究提出以屬性離散的粗集分類器方法應用於獲利率資料庫上其分類的結果與決策樹、類神經網路及粗集理論等其他演算法相比，在規則數及分類正確率上皆呈現較佳的效果。其主要原因為採用決策樹的離散化方法。

本研究所提的方法應用在獲利率分類有上具有良好的分類效果，未來可將本研究方法應用於其它模式之預測，或是應用於商業資料如顧客關係管理。而研究方法上的改良，也可以結合最小熵原理離散化方法來強化粗集的分類正確率。

參考文獻

- [1] 丁一賢、陳牧言。2005。資料探勘。滄海書局。台中市。
- [2] 吳長洲。2003。運用資料探勘技術在產險行銷之個案研究。中原大學碩士論文，中壢市。
- [3] 林聰敏。2003。公營事業民營化前後經營績效比較之研究。中原大學碩士論文，中壢市。
- [4] 姚志成。2005。運用資料探勘技術建構脂肪肝預測模式。中原大學碩士論文，中壢市。
- [5] 郭一聰。2005。應用決策樹與類神經網路於應收帳款之呆帳預警模式研究。中原大學碩士論文，中壢市。
- [6] 陳金山。2003。資料探勘於目標行銷上之應用研究。雲林科技大學碩士論文，雲林縣。
- [7] 曾憲雄、蔡秀滿、蘇東興、曾秋蓉、王慶堯。2005。資料探勘。旗標出版社。台北市。
- [8] 蔡爵名、鄭景俗。區間離散化強化粗集及其在植物分類的應用。第十八屆2007國際資訊管理學術研討會論文集。銘傳大學，桃園縣。
- [9] 劉祥熹、黃日鈺。2005。銀行業借貸風險評估之實證分析－資料探勘與統計模型之比較。華人前瞻研究，第一卷第一期，48-71。
- [10] HAN, Jiawei and Micheline Kamber, Data Mining: Concepts and Techniques, Academic Press, pp279-327, 2001.
- [11] Michie, D. et.al.. Machine Learning, Neural, and Statistical Classification. Ellis Horwood, New York., 1994.
- [12] Pawlak Z, "Rough Sets.", Information Journal of Information and Computer Sciences, 11:341-356, 1982.
- [13] Pawlak, Z., Rough Sets : Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht , Boston., 1991.
- [14] Pawlak, Z., J. Grzymala-Busse, R. Slowinski, "Rough Sets", Commun ACM, 38(11), 89-95, 1995.
- [15] Pawlak, Z., J. Rough Sets and Intelligent Data Analysis, Information sciences, Vol.147, pp.1-12, 2000.
- [16] Porter, Michael E., America's Green Strategy, Scientific American 264(4), 168, 1991.
- [17] Predki, B., R. Slowinski, J. Stefanowski, R. Susmaga, and Sz. Wilk, "ROSE -Software Implementation of the Rough Set Theory", In: L.Polkowski, A.Skowron, eds. Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence, Vol. 1424,. Springer-Verlag, Berlin., pp605-608, 1998.