國立交通大學

管理學院（資訊管理學程）碩士班

碩士論文

螞蟻分類技術之研究

A Study of the Ant Colony System for the Discovery of Classification Rules

研究生：陳惠琪

指導教授：劉敦仁、林妙聰

中　華　民　國　九　十　五　年　六　月

螞蟻分類技術之研究

學生：陳惠琪　　指導教授：劉敦仁教授
林妙聰教授

國立交通大學管理學院（資訊管理學程）碩士班

摘　　要

資料探勘中的分類是知識管理中最基本也是必備的一環，唯有加以分類編碼才可能將知識成為資料庫並加速知識的擴散。螞蟻演算法是在 1991 年由 Colorni 等學者提出，為一新近發展的求近似解演算法。螞蟻演算法原本多運用於求解組合最佳化問題，例如旅行銷售員問題(traveling salesman problem)、二次分派問題(quadratic assignment problem)等。近幾年來，許多研究者發現螞蟻演算法對於資料探勘(data mining)方面亦有不錯的表現。因此，本論文即希望探討如何藉由螞蟻演算法的分類技術，以提升知識管理者的處理效率及一致性。

本研究的分類法著重於名詞性的分類而非數值型態之分類，運用常用的分類法與最近新興的螞蟻分類技術(Ant-Miner)來進行比較。本研究所使用進行比較的軟體工具共有兩套：其中一套為 Weka 的軟體，係是由 Java 所寫成，為根據各式的機器學習(machine learning)演算法所寫出來的資料探勘軟體；另一套 Ant-Miner 係由 Rrfael 等學者在 2002 年，將原本運用在各最佳化解題的螞蟻系統，運用在資料探勘方面所發展成的。

之前的針對螞蟻的相關分類技術文件，僅針對常用來做測試的 UCI Machine Learning Repository 的資料來進行分類。本研究中除了測試 UCI 的資料外，亦將實際於 2005 年經由問卷調查所搜集的資料，輸入至 Ant-Miner 中測試其效果，並將其結果與貝氏分類法及決策樹分類法進行比較。研究結果發現不管那一種分類法，其訓練資料量的大小所建立之分類模式會造成正確率的不同，而分類正確率與執行效率也會有一定的對比關係，此項比較與分析將可做為實務上採行之參考。

關鍵詞：資料探勘、知識分類、螞蟻演算法、貝氏分類、決策樹

A Study of the Ant Colony System for the Discovery of Classification Rules

Student：Hui-Chi Chen

Advisors： Dr. Duen-Ren Liu
Dr. B.M.T. Lin

Institute of Information Management
National Chiao Tung University

ABSTRACT

Ant Colony Optimization (ACO) was proposed by Colorni et al in 1991 from the collaborative behavior of ant colonies. It has been applied to such combinatorial optimization problems as traveling salesman problem, quadratic assignment problem, just to name a few. In the recent years, the ACO approach was deployed in the area of data mining, where algorithmic and statistical techniques are used to discover or extract useful information as well as knowledge from large volume of data. This thesis aims to study the efficiency and effectiveness of Ant-Miner, a well-known classifier that is developed using ACO.

The major function of Ant-Miner is to extract classification rules out of the examined data sets. The terms or conditions of a rule will be added or removed by ant colony through collaboration or pheromone sharing. The focus of this research is set on the performance comparison between Ant-Miner and Weka, which is a data mining tool incorporating machine learning mechanisms.

In this research, we have two data sets with nominal attributes. The first set is selected from the UCI Machine Learning Repository, and the second is a real data set collected in 2005 by a local research institute. We use the two data sets to compare Naivebayes and Decision Tree with Ant-Miner.

Experimental results and analysis show that different classification tools demonstrate different levels of efficiency and effectiveness. We also examine the performance of Ant-Miner resulted from different parameter settings, including such as colony size, evaporation rate and diversification level.

Keywords: Data Mining, Knowledge Management, ACO, NaiveBayes, Decision Tree

誌　　謝

重回校園真的是一個很巧的機緣，在職場多年，逐漸的覺得自己知識及能力的不足，一直就很想再進校園繼續念書，但是總是缺乏動力，而就在因緣際會下，我接觸到了交大專班的學長姐們，了解到重回校園是多麼新鮮而且有趣的學習，於是便下定決心展開了我重當學生的生涯。

首先我要感謝我的指導教授劉敦仁老師，劉老師總是細心的指導我們，帶著我們一點一點的進步，而且很有耐心的解釋該如何了解及定義自己論文的方向，同時也要感謝的是林妙聰老師，讓我接觸到有趣螞蟻演算法，進而讓我決定了自己的論文方向，並不厭其煩的跟我們說明指導，讓我受益匪淺。

還要感謝資管所的老師、學長姐、同學們以及實驗室的伙伴，在我覺得困惑及疑問時，總是仔細的指導及鼓勵。

最後要感謝的就是我的家人，由於有了家人的支持及鼓勵，讓我能夠順利的完成學業。

目錄

表目錄

圖目錄

第一章 緒論

1.1 研究背景與動機

資料探勘(Data Mining)是目前資料庫應用中相當熱門及實用的技術，由於企業不斷的追求效率，更使得資料探勘成為各項資訊運用的第一把交椅。

由於近幾年來許多企業紛紛推動知識管理，希望藉此強化企業的競爭力。因此不斷地要求各部門員工將個人的專業知識記錄下來，並將知識依資料欄位以文字或數字型態的方式一筆一筆的加以記錄，但這些知識在未經整理前僅是一筆一筆的資料而已，於是人們開始整理資料，以便找出有意義的資訊，也就是開始進行資料探勘，而資料探勘中最常使用的一種型態就是分類了，一開始資料的分類通常由一位專職人員來進行人工分類，但由於人員的變遷，而每位人員對於分類的定義及專業領域的認知不盡相同，長期下來使得資料分類變得混亂而複雜，造成使用者無法找到正確的資料，而由於資料量愈來愈大，分類的工作負擔也就會愈趨龐大。

資料的分類是知識管理中最基本也是必備的一環，唯有加以分類編碼才可能將知識成為資料庫供全企業員工有效率的查詢，以加速知識的擴散，但若資料分類管理不當，會導致知識管理系統在搜尋資料時的效能低落，甚至找不到適當的資料，進而造成知識無法有效利用或傳遞，而也會使員工使用率降低，進而降低了知識分享的效果。不僅如此，大量無效的知識資料，不但佔用了龐大的儲存空間，也減低了知識搜尋和運用的效率，而雜亂的資料記錄分類更造成知識流通不易。

而螞蟻演算法是在 1991 年由 Colorni 等學者[21]提出，為一新近發展的求近似解演算法，一開始為運用在求解組合之最佳化問題、旅行銷售員問題(traveling salesman problem)、二次分派問題(quadratic assignment problem)等，而近幾年來有許多研究者發現螞蟻演算法對於資料探勘(data mining)方面亦有不錯的表現，所以希望藉由螞蟻演算法的分類技術[32]提升知識管理者的處理效率及一致性。

1.2 研究目的

本研究希望能夠透過資料採勘中良好的分類機制，如運用螞蟻演算法所設計的Ant-Miner分類器來簡化資料的分類作業，不僅降低了人工作業負擔及分類的不一致性，良好的分類準確率也會提升資料的搜尋效率，高的搜尋效率會讓使用率同時提升，如此便可大大的改善知識管理系統的運作效益。

目前的技術文件的實驗資料來源大部份都是來自於網址為http://www.ics.uci.edu/~mlearn/MLRepository.html的資料，此網站稱為「UCI Machine Learning Repository」，是美國加州大學爾灣分校（University of California at Irvine）的資訊電腦學院（Donald Bren School of Information and Computer Science），這個網站收集了各式各樣的資料，並加以整理說明，以方便各個研究學者使用各種不同的方法，來對這些資料進行實驗，並比較所得的結果。本研究不僅針對其資料進行比較分析，亦將實際之問卷調查資料作為實驗資料，進行其比對及差異性分析。

1.3 研究架構

本論文之研究架構內容大致可分為六個章節：

1. 緒論：本章主要在說明本研究的背景與動機、研究目的以及研究之架構
2. 文獻探討：針對在本研究中所提到之相關文獻資料作一回顧探討。其內容包含資料探勘、知識管理、螞蟻演算法、貝氏分類法以及決策樹分類法。
3. 實驗架構：主要是針對此次實驗所使用的三種分類法，螞蟻分類法、貝氏分類法以及決策樹分類法詳細的技術說明。
4. 實驗設計：主要是針對本研究之實驗程序流程、實驗資料、使用之軟硬體以及所使用的演算法變數定義進行說明。
5. 實驗結果比較分析：主要是針對在實驗中所比較的兩組資料其分類正確率以及效率進行其結果分析，並針對螞蟻分類軟體(Ant-Miner)訂定不同的參數進行比較。
6. 結論與建議：針對一個觀察者的角度來看螞蟻分類技術是否有改善的空間以及未來值得研究的方向。

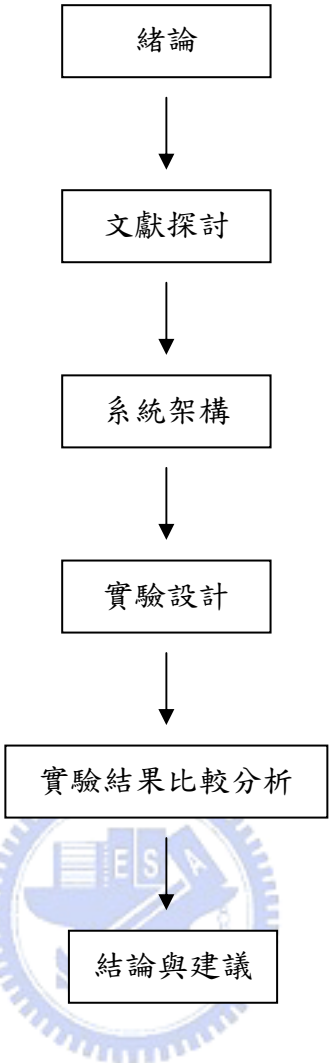


圖 1-1 研究架構流程圖

第二章 文獻探討

2.1 資料探勘篇

2.1.1 資料探勘的意義

資料探勘(Data Mining)是目前非常熱門及廣泛運用在各企業界的一項技術，在學術上的定義[29]是為藉由各種不同的演算法，透過各項技術自動化地或是半自動化地進行資料分析且能自資料中抽取出有價值之資訊的一種技術；也就是從大量資料中，藉由各種不同的資料分析方式來尋找或預測可能的規則或知識，其目的就是希望從資料中發掘有用的資訊或知識。

2.1.2 資料探勘的型態

資料探勘(Data Mining)依據不同的分析方式及產生的知識型態大致做了一些分類[17]，常見的資料探勘型態有下列幾項：關聯法則(Association Rules)、分類(Classification)、推估(Estimation)、預測(Prediction)、群集偵測(Cluster Detection)等等，其型態說明如下：

1. 關聯法則(Association Rules)

   主要是透過企業之交易資料庫(Transaction Database)，尋找資料庫中資料項目或屬性之間的關聯性，以分析及了解資料中潛在的意義，其中最著名的發現就是啤酒與尿布的關係，當人們購買啤酒時，同時也會購買尿布，這便是透過資料庫中的交易分析，了解了客戶的消費行為。因此透過分析後所找出的資料關聯性，便會了解，那些產品客戶會一起購買，或是客戶在買了某一樣產品之後，在多久之內會買另一樣產品等等，所以其最終目的便是希望能從資料與資料之間的分析中，找出未知的關係，作為決策時的參考。

2. 分類(Classification)

   分類分析是從已知類別的物件集合中，依據其屬性(可能影響物件類別的變數)建立一個分類模式(如決策樹、貝氏分類法等)來描述物件屬性與類別之關係，然後再根據這個分類模式對其他未經分類或是新的資料做預測。這些用來尋找特徵的已分類資料可能是來自現有的歷史性資料，或是企業蒐集保存的客戶基本資料(Profile)。換言之，分類分析是一種依資料屬性建立類別的過程，通常從資料中產生「若干」法則，例如可藉由病歷來預測病人是否具有肝病，或可藉由信用資料來預測客戶的信用風險。

3. 推估(Estimation)

   分類善於處理離散性數值，推估則善於處理連續性數值，可用來推估一些未知的連續性變數，例如信用卡申請者之教育程度、收入、職業等因素，推估信用卡之消費額度與適合使用哪一種促銷專案。

4. 預測(Prediction)

   預測與分類和推估的方法是相當接近的，不同的是，預測是推估未來的數值與趨勢。利用歷史資料，選用為已知變數值的訓練資料，可建立模型以檢視過去到現在的觀察值的變化，再利用最近的資料輸入所建立之模型中，即可獲得關於未來變化的預測值。

2.1.3 資料探勘的應用

1. 透過各種不同的演算法利用自動化或半自動化式，來探索或分析大量資料，以發覺隱藏期中且有意義的資料，如資料分類系統
2. 在龐大的的資料資料庫中尋找出有價值的內含意義，藉由統計及人工智慧等先進的科學技術，將資料深入分析，找出其中所隱含的知識，並根據企業的不同的問題建立相關的模型，以提供企業進行決策時依據，如專家系統。
3. 將顧客相關的資料蒐集起來並作分析，亦即將原始資料轉變為商機，進而成為企業的商業智慧，如客戶關係管理系統、推薦系統。

2.2 知識管理篇

2.2.1 知識管理的意義

知識管理（Knowledge Management，簡稱ＫＭ），是指企業建立一個包含了將資料、資訊技術整合成符合企業精神的組織流程，並透過知識分享(Knowledge Sharing) 彼此的互動及全體員工的創新力和創造力，促使整個企業及個人持續性的得以進步，進而產生更大的競爭力，為整個企業創造更多的價值。

知識管理常被看作一種廣泛的代表性名詞，包括公司內的腦力激盪、正式會議、非正式的討論、簡報等，各式各樣的資料蒐集和累積。而現今不論是學術上或實務上，對「知識管理」的定義均有其獨特的見解，並無統一的說法。而根據我們在實務上的經驗，及輔以理論的研究，我們認為，知識管理應該是一種具有完整規劃的系統，能夠持續地、普遍地蒐集人類的智慧；同時，知識管理也是一種互動的過程，不只是被動式地蒐集知識，而是進一步轉化企業文化，讓人們更注重「資訊的交換」；因為這種互動的、流動的資訊，才是知識最能創造價值的地方。

資策會 MIC 研究報告也指出：知識管理可分為 3 種層次[20]，第 1 層是知識的保留，將組織內的知識文件化；第 2 層是知識的分享，利用良好的機制幫助知識在組織內流通，並產生互動；第 3 層是知識管理的極致，員工吸收知後，發現新的問題進而創造新知識管理系統後。而根據 GartnerGroup 研究，目前採用知識管理系統後，70%的價值發揮在 Know-how 的分享，由此可知知識管理最立即可見的效益是能促進知識的分享。

2.2.2 知識管理的目的

針對知識管理的目的在彼得 杜拉克的書[4]中曾說明：

1. 增加組織整體知識的存量與價值。
2. 應用知識以提昇技術、產品、與服務創新的績效以及組織整體對外的競爭力。
3. 促進組織內部的知識流通，提昇成員獲取知識的效率
4. 指導組織知識創新的方向。

5. 協助組織發展核心技術能力。
6. 有效發揮組織內個體成員的知識能力與開發潛能。
7. 提昇組織個體與整體的知識學習能力。
8. 形成有利於知識創新的企業文化與價值觀。

2.2.3 知識管理的形成及種類

如下之圖表說明：

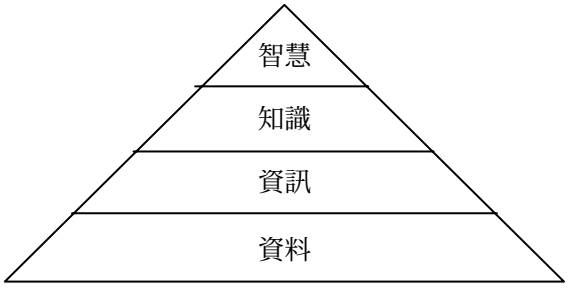


圖 2-1 知識的形成示意圖

表 2-1 內隱及外顯知識比較表

內隱知識	外顯知識
■ 主觀的經驗知識 ■ 具情境特殊性 ■ 類比性 ■ 不易直接具體表達 ■ 保存於人身上, 製程, 關係等型式中 ■ 難以透過文字, 程式或圖形具體調列規劃之形式向外傳遞	■ 客觀的理性知識 ■ 具條理順序性 ■ 數位性 ■ 可以文字和數字來表達 ■ 保存於產品, 程序, 手冊等之具體形態中 ■ 可以透過正形式及系統性語言傳遞的知識

2.3 螞蟻演算法篇

在自然界中螞蟻的溝通方式是透過釋放費洛蒙(pheromone)，以覓食為例，螞蟻在尋找食物時，沿路會釋放費洛蒙，但由於在找尋食物的螞蟻不止一隻，所以其路線也不止一條，但如果尋找食物的路線較短時，螞蟻往返的頻率較密集，因此會有濃度較高的費洛蒙，而後續出去尋找食物的螞蟻會根據濃度較高的費洛蒙路線來往返，而濃度不高的路線的費洛蒙則隨著時間的流逝而蒸發消失，因此找到最短路線的最佳解。

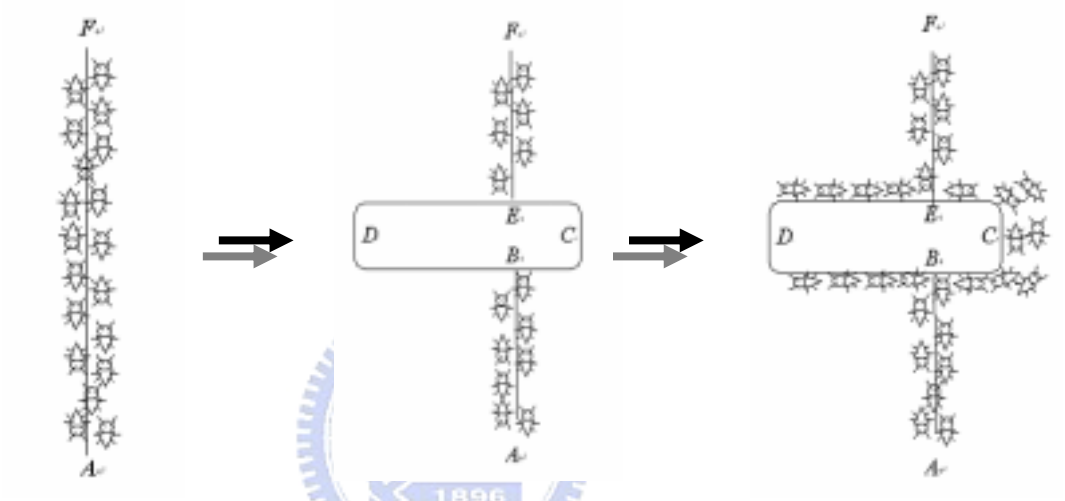


圖 2-2 真實螞蟻覓食示意圖
資料來源：林妙聰教授，2005

2.3.1 螞蟻演算法的演進過程

1. 在1991年Colorni，Dorigo及Maniezzo根據自然界中螞蟻的合作現象發表了一篇螞蟻族群最佳化(Ant Colony Optimization)來解決旅行銷售員問題(TSP)[28]。
2. 1994年提出螞蟻系統(Ant System)來解決零工式生產排程問題。
3. 1996年提出螞蟻族群系統(ant colony system, ACS)，為一個啟發式演算法，被廣泛應用於求解各種組合最佳化問題。[25]。
4. 1999年提出分離式最佳化螞蟻演算法(Ant Algorithms for Discrete Optimization)[27]。
5. 2002年提出利用螞蟻理論做分類規則搜尋(An Ant Colony Algorithm

for Classification Rule Discovery)[32]。

6. 2002年8月Rafael S. Parpinelli，Heitor S. Lopes，Member，IEEE，and Alex A. Freitas提出利用螞蟻理論做資料探勘(Data Mining With an Ant Colony Optimization Algorithm)並做出Ant-Miner (ant-colony-based data miner)[33]。
7. 2003年周仕雄提出以螞蟻為理論基礎所做集群分析技術Ant System-based Clustering Algorithm(ASCA)[6]。
8. 2004年P.S. Shelokar，V. K. Jayaraman，B. D. Kulkarni提出螞蟻族群分類系統(An ant colony classifier system：application to some process engineering problems)，其利用C++撰寫，透過啟發式的訊息(heuristic information)來建立規則和透過費洛蒙的間接溝通方式來改善規則[31]。

2.3.2 螞蟻演算法的特性

[一]、螞蟻系統的主要特性[25]

1. 正向回饋(positive feedback)
   迅速的發現好的解決辦法。
2. 分佈式計算(distributed computation)
   增加探索的可能性，避免過早收斂(premature convergence)侷限於區域最佳解。
3. 建設性的貪婪探索法(constructive greedy heuristic)
   在初期搜索的階段中獲得可接受的解，加速解題的進行。

[二]、螞蟻系統的不可或缺特性[25]

1. 它是多用途的(versatile)
   因為它能被用于相同的問題的相似的版本； 例如，有一簡單擴展從那裡巡迴售貨問題(TSP)到 非對稱的巡迴售貨問題(ATSP) 。
2. 它是健全的(robust)
   對於其他問題的最佳化可做變化性的使用，例如二次的分派問題(QAP)和作業安排調度問題(JSP) 。
3. 它是族群為基礎的方法(population based approach)
   因為它允許開發作為一個搜尋機制的正向回饋。

2.3.3 螞蟻演算法的運用範圍

目前螞蟻族群演算法所運用的範圍如下：[2][3][5][11][15][16]

1. 旅行銷售員問題：即一地區有若干城市，城市間有道路，道路之距離有遠近，求其經過每個城市但不重覆的前提下最短的距離。
2. 二次分派問題：將 M 個設備分給 N 個地點，設備間有流量需求，地點間距離不同，求距離乘流量的和的最低成本值。
3. 工作排程問題：如求最短的完成時間，最小的延遲時間…等。
4. 運輸繞路問題：根據各點的需求量、服務時間、貨車的載重量、各點只能去一次…等，求服務成本最小化。
5. 網路路由問題：將螞蟻視為封包，並修改路由表格(routing table)即修改費洛蒙的濃度以指引資料封包的流向。
6. 連續性順序問題：在一成本存於連線上且具方向性的圖形中，找出最小成本的漢米敦圖形，類似非對稱性的旅行銷售員問題(即需滿足某些點間的特定順序）。
7. 著色問題與頻道分配問題：即點圖上顏色時，相鄰的點顏色不能相同，所使用的顏色要最少。
8. 一般分配問題：將 M 份工作分給 N 個工作單位，且一個工作單位只能有一件工作，求總成本最小化。
9. 其他應用：如結合類神經網路與螞蟻演算法改進高速網路交通流量控制。

2.3.4 自然界螞蟻 Vs 人工螞蟻

在Rafael等學者的研究中[32]指出與真正的螞蟻有一些差別的比較：

1. 人造螞蟻有記憶。
2. 他們不完全盲。
3. 他們居住在時間分離的環境。

另一方面，人工螞蟻也採用真正的螞蟻的幾種特性：

1. 螞蟻偏愛費洛蒙較高的路徑。
2. 較短的路徑對於費洛蒙濃度的成長比率較高。
3. 螞蟻根據每條路徑費洛蒙的濃度來做間接的溝通。

因此設計人工螞蟻時需注意以下三項工作

1 制定狀態轉移規格：設計人工螞蟻需考慮開發與探索的比率,以及如何運

用費洛蒙及適當的啟發函數。

2　建立問題限制：如此才會符合問題的限制。

3　設計費洛蒙異動規則：

3.1 區域更新：避免一過於強勢的路徑，吸引所有螞蟻走上同一路徑，導致廣不足，而走上區域最佳解。

3.2 總體更新：

3.2.1　回合最佳：每回合所得解答中最好的路徑，進行增加費洛蒙的動作。

3.2.2　總體最佳：截至目前所獲得之解答中針對最好者的路徑，進行費洛蒙更新。

3.3 費洛蒙蒸發：設定過高的費洛蒙蒸發率將使得求解經驗無法累積。

為獲得較佳的解，Dorigo等人[25]曾建議，在ACS 中的螞蟻除了直接選擇已知的路徑外，也要會選擇未曾走過的路徑，此即「開發」(Exploitation)與「探索」(Exploration)兩種路徑選擇機制。「開發」(Exploitation)的功能主要能夠改善原有之解答，會常會被侷限於區域最佳解，若加上「探索」(Exploration)，尋求未知的路徑，則可加大搜尋的廣度，尋找更佳的解答。

這兩種路徑選擇機制被採用的機率是影響求解效能的關鍵。當「開發」被採用的機率很高時，可能會使得ACS的效能降低。而當這兩種機制被採用的機率差不多時，可提高ACS的效能。

2.3.5 螞蟻的分群技術

在近幾年來，有研究學者針對以螞蟻為理論基礎所提出的集群分析技術如周仕雄的 Ant System-based Clustering Algorithm(ASCA)[6]及周世章的螞蟻分群演算法（ant-based clustering algorithm, ACA）[7]等，其說明如下：

首先說明關於螞蟻系統的相關基礎理論及計算方程式[25]：

1. 歐幾里德距離

$$d_{ij} = \sqrt{(xi - xj)^2 + (yi - yj)^2} \qquad (1)$$

d_{ij} ：從城市 i 點至城市 j 點之歐幾里德距離

2. 節點轉換機率

假設 m 隻螞蟻隨機的走在 n 個城市中，當牠移動至下一個城市的考慮因素為費洛蒙強度及節線長度，其機率計算方程式如下（此機率稱為 transition probability）。

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}]^{\beta}} & if\ j \in allowed_k \\ 0 & otherwise \end{cases} \qquad (2)$$

$p_k(i,j)$ ：第 k 隻螞蟻在城市 i 選擇去城市 j 的機率。

τ ：費洛蒙。

η（visibility）：期望值，為城市 i 與城市 j 之間距離的倒數，$\eta = \dfrac{1}{d_{ij}}$

$allowed_k$ ：第 k 隻螞蟻沒有拜訪的城市。

α 、β ：控制費洛蒙足跡對 η 的相對重要性參數。

3. 費洛蒙更新(pheromone updating)

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad (3)$$

以上為各節點之更新方式，其中 $\Delta\tau_k(i,j)$ 為節線在兩個時間點的費洛蒙濃度

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{L_k} if\ k-th\ ant\ uses\ edge(i,j)\ in\ its\ tour\ (between\ time\ t\ and\ t+n) \\ 0 \qquad otherwise \end{cases} \qquad (4)$$

t：時間點。

Q：屬於常數。

L^k：第 K 隻螞蟻所完成的路徑總長度。

ρ：費洛蒙的衰退參數，$0 < \rho < 1$，$1-\rho$ 為兩時間點的蒸發值。

m：螞蟻的數量。

這種更新法則，稱之為 ant cycle 演算法。

4. 新的更新法則演算法

$$\Delta\tau_{ij}^{k} = \begin{cases} Q\,,\ if\ (i,j) \in tour\ done\ by\ ant\ k \\ 0\,,\ otherwise \end{cases} \quad \text{ant-density model} \qquad (5)$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{d_{ij}}\,,\ if\ (i,j) \in tour\ done\ by\ ant\ k \\ 0\,,\ otherwise \end{cases} \quad \text{ant-quantity model} \qquad (6)$$

其中，螞蟻都自 i 至 j，遺留下的足跡品質為 $\dfrac{Q}{d_{ij}}$

關於周仕雄的 ASCA 之程序，其中包含了 Divide, Agglomerate_obj, Agglomerate, Remove 四種程序，如下頁之流程圖。

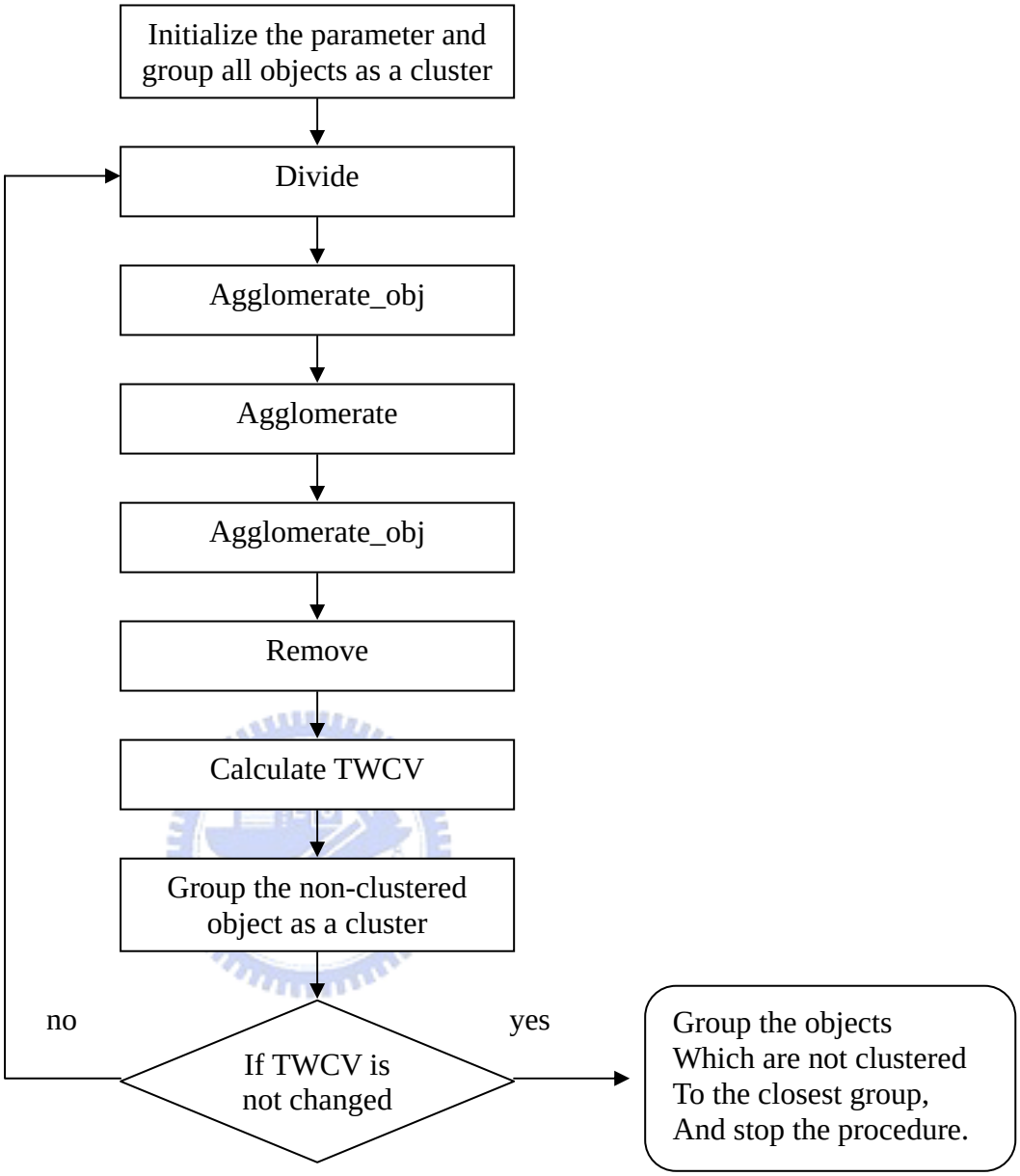


圖 2-3 ASCA 流程圖

資料來源：周仕雄，2003

ASCA 之集群技術程序說明內容如下：

1. Divides：藉由費洛蒙的濃度分群並逐一的演算。
2. Agglomerate_obj：將尚未分群的單元融合。
3. Agglomerate：將相似的兩個群聚融合在一起。
4. Remove：在每個群聚中，移走差異性較大的點，直到變異不再改變為止。

另外周世章所提出的程序如下：

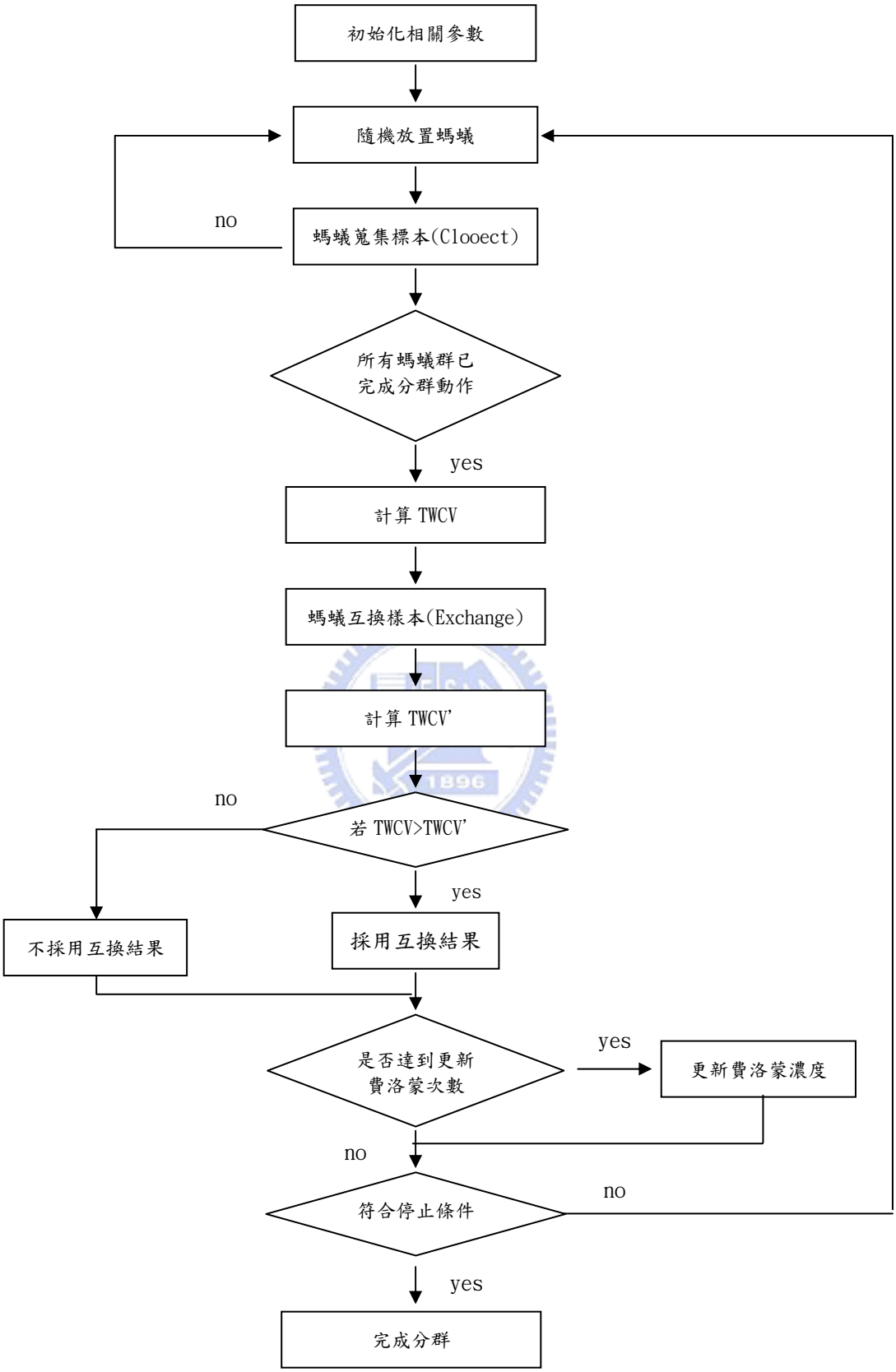


圖 2-4 ACA 流程圖

資料來源：周世章，2004

2.3.6 螞蟻的分類技術

在各螞蟻理論相關的技術文件中發現，從 2002 年開始便有研究學者[32][33]開始針對資料探勘方面的模式來進行研究，開始提出利用螞蟻理論做分類規則搜尋，讓螞蟻又多了一項可以應用的領域。之後也寫出相關程式 Ant-Miner 來進行測試，這就是我此篇研究的內容，而針對螞蟻的分類技術我將會在第三章詳細說明。

2.4 其他分類法篇

2.4.1 貝式分類法

貝式分類法（Bayes classifier）[12]乃是根據貝氏定理（Bayes' theorem）為基礎，(貝氏定理是由一位英國牧師 Reverend Thomas ,Bayes 所提出的)，用以計算未知類別的資料其屬於各分類類別的機率。整個貝式分類法的目標是希望能透過機率統計的分析，達到最小誤差的一種分類方式；即利用各類別已知的屬性〈attributes〉機率值及各類別之事前機率，計算新案例於各類別的機率，最後比較各類別的機率，機率最大者則該案例分於此類別。

目前貝氏分類法大致可分為兩類：單純貝氏分類 (Naive Bayesian Classifier)和貝氏信念網路(bayesian belief networks)。Microsoft Naive Bayes（貝葉斯演算法）能夠快速構建可用於分類和預測的資料採礦模型。如果知道可預測屬性的每種狀態，便可計算出輸入屬性每個可能狀態的概率。這種演算法只支援離散（不連續）屬性，它認為所有輸入屬性都是彼此獨立的（前提是知道可預測屬性）。因為貝葉斯演算法的計算速度非常快，因此在初始資料研究階段通常會選擇這種演算法進行分類和預測問題。

在理論上，貝氏分類法與其他分類法比較起來有最小的錯誤率，然而在實際上，屬性彼此的關係很少是獨立的，且資料分布也很難認定，所以在分類上還是很難達到完全的正確。但是目前有進一步的演算法，可以處理屬性彼此的關係，如：貝氏信任網路（Bayesian Belief Networks），而在處理連續屬性的問題上，也有一些分割的演算法可以將連續數值轉換成離散數值，如：ten-bins，entropy 等方式，可以使貝氏分類法的結果更具可靠性。

2.4.2 決策樹分類法

決策樹是一種歸納學習法，主要是透過訓練資料(training data)來研究資料分類的規則以及共通的特徵，然後根據這些規則或特徵來建立分類模式，透過此建立的分類模式對其他新資料或未經分類的資料作預測。而決策樹就是利用樹狀結構圖的方式來表達決策的流程，因而稱之為決策樹。

由於決策樹具有規則導向及易於理解的特性，為一種應用相當廣泛的資料探勘技術，早期多用於醫療方面的研究，但目前的研究範圍已經廣泛的運用到

各個不同的領域了。

每個決策樹皆是從根部開始發展的，稱之為根節點(root node)，每一個分支所延伸出來的節點稱之為樹枝節點 (internal node)，將會用來判斷決定每一筆資料該進入下一層那一個子節點，如此重覆的執行直到所有資料均到達葉節點(leaf)為止。簡略說起來，決策樹就像是一群布林函數 (Boolean function)的集合，樹狀結構圖中的每個節點(node)都包含著一組屬性(Attrubute)，在屬性中決定該類別之後的分類。

以下的一個簡易例子即說明利用決策樹來決定何種天氣適合出去活動，如果天氣看起來有陽光，而濕度低於 75 度的適合活動；如果天氣看起來陰陰的適合活動；如果天氣有下雨但是沒有風也是適合活動的，所以新的測試資料便可以依據此決策樹所設計出來的 rule 來分析是否適合出去活動。

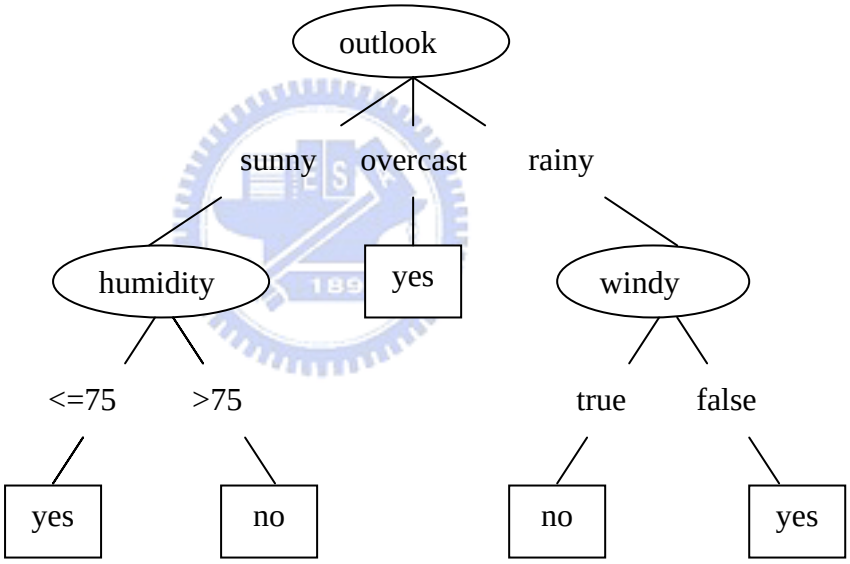


圖 2-5 決策樹簡易說明流程圖

其主要演算法有以下幾種：

1. CART(Classification And Regression Trees)：利用訓練資料建構一完整的決策樹後，運用整體錯誤率（Entire Error Rate）進行事後修剪的工作。
2. CHAID(CHi square Automatic Interaction Detector)：運用卡方檢定選擇能使資料產生統計上顯著差異的分類屬性來分割資料。主要適用於建立類別屬性（Categorical Attribute）的決策樹。
3. ID3：其主要核心在於其以遞迴的方式將訓練資料作切割。在每一次產生節點時，某些輸入的訓練子集將取出測試，主要特點是以最大的

資訊獲取來當作測試與最後被選取的節點。

4. C4.5：C4.5 是 ID3 的延伸方法，它更加具備有處理連續數值型屬性、雜訊的屬性選擇特性，另外也兼具修剪樹的能力。

第三章 實驗架構

3.1 螞蟻分類技術

3.1.1 Ant-miner 架構流程圖

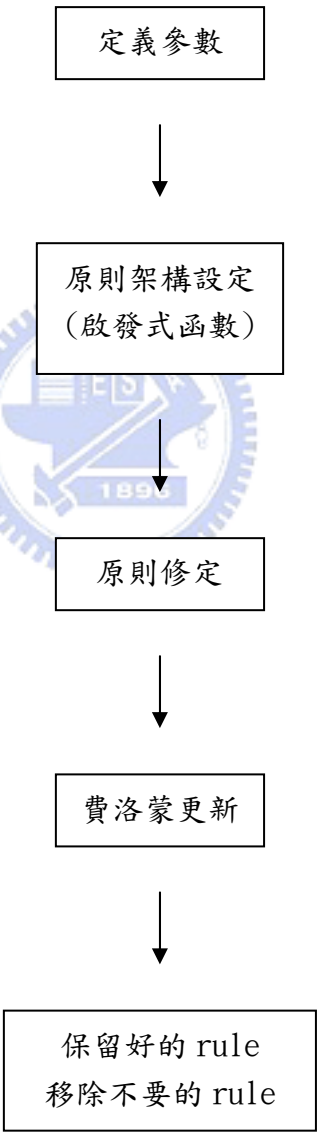


圖 3-2 Ant Miner 架構流程圖

```
Training Set = all training cases;
WHILE (NO. of uncovered cases in the Training Set > Max_Uncovered_cases)
 i = 0 ;
 REPEAT
  i = i + 1 ;
  Ant_i incrementally constructs a classification rule ;
  Prune the just-constructed rule ;
  Update the pheromone of the trail followed by Ant_i ;
 UNTIL (i ≥ No_of_Ants) OR
       (Ant_i constructed the same rule than the previous
        No_Rules_Converg – 1 Ants)
 Select the best rule among all constructed rules ;
 Remove the cases correctly covered by the selected rule from the Training Set,
END WHILE
```

圖 3-3 Ant-Miner 程序

資料來源：Rrfael et al., 2002

3.1.2 螞蟻分類技術說明

在 2002 年 Rrfael 等學者有了更突破性的發展，將原本運用在各最佳化解題的螞蟻系統往資料探勘方面發展，大家對於這項技術都相當看好[32]。

發展螞蟻分類技術(Ant-Miner)的原因如下：

1. ACS(Ant Colony System)有簡單的 agent，且與其他隻共同合作以完成工作。
2. 整體一致的行為。
3. 在廣大的搜尋空間中，生產出能夠為問題找出高品質解決方案的健全系統。
4. ACS 針對包含預測屬性的值、邏輯條件的組合能力去處理其有彈性及健全的搜尋。

每隻螞蟻都看做為了目的地問題，遞增地架構及修改解答的 Agent，其原則很簡單，如左之判斷式：IF <condition> Then <class>

說明如下：

Condition：包含預測屬性的邏輯組合

<condition>：=term1 AND term2….

term：包含三項內容<attribute，operator，value>

attribute：屬性

value：所屬領域屬性的值

operator：關係式

class：每筆 record 的預測類別

一開始 Rule 是不包含任何 term 的，之後再一筆一筆加入，慢慢的現有路徑便會延展開來，於是便包含所有的可能路徑。

其所選擇加入的 term 是由二個條件來判斷：

1. 啟發式函數問題的相依性
2. 費落蒙的總數

Ant 一次加一個條件至 rule 中直到無法延申為止，而無法延申的情形有二：

甲、 無論加入任何一個 term，加入時其 case 數小於 user-specified 的 threshold 即停止(稱之為 Min_cases_per_rule，也就是每個 rule 包含的最小 case 數)。

乙、 所有屬性皆已經被 Ant 使用，也就是說沒有任何屬性可以再加入 rule 中了。

以上情形只要一種條件成立即停止。原則上我們可根據所建立的 rule 來進行分類，但實際上，在處理程序中通常要修剪其所加入的 rule，也就是要移除不適當的 term，這些不適當的 term 可能是因為 a.term 在選擇程序時有一些可能性的變動，b.目光短淺；也就是可能一次只考慮到一個屬性而忽略了屬性之間的互相影響。

當一隻螞蟻完成了它的 rule 以及費洛蒙的更新後，接下來便換另一隻螞蟻開始架構它的 rule，並使用新的費洛蒙，這個程序會重覆執行，而這必須事先在系統中事先定義其參數(No_of_Ants)，即為螞蟻的數量，根據這些數量重覆執行。而當之後的螞蟻所產生出來的 rule 與前一隻相同(即收斂)時，便會提早中止這些執行，在系統定義其參數為(No_Rules_Converg)，即螞蟻收斂的 rule 數。

不斷重覆以上的動作直到 training set 中不包含的 case 數小於事先所定義的 threshold(稱之為 Max_uncovered_cases，也就是 training

set 不包含的最大 case 數)，當在 training set 離開的 case 數小於 Max_uncovered_cases 時，rule 的搜尋便停止，這些所搜尋到的 rule 會存放至 ordered rule 中，以使用於新 case 的分類。

以下針對相關的方程式作說明：

1. 原則架構(Rule Construction)

$$P_{ij}(t)=\frac{\tau_{ij}(t)\cdot\eta_{ij}}{\sum_{i}^{a}\sum_{j}^{b_i}\tau_{ij}(t)\cdot\eta_{ij},\forall i\in I} \qquad (7)$$

η_{ij} ：is the value of a problem-dependent heuristic function for termij;

$\tau_{ij}(t)$：is the amount of pheromone currently available (at time t) in the position i,j of the trail being followed by the ant;

a：is the total number of attributes;

b_i：is the total number of values on the domain of attribute i;

I： are the attributes i not yet used by the ant

η_{ij} 就是 heuristic function 中計算每個 term 的預測能力值，所以 η_{ij} 的值愈高其 term 就是最適當的分類，也就是會有較高的被選擇機會。但它有兩個限制是不能加入的，那就是 a. 此項 term 之前已經加入過了 b. 小於所設定的 rule 最小 case 數。

2. 啟發式函數(Heuristic Function)

在 Ant-Miner 中的啟發式函數是針對每一項 term 的品質的評估，也就是關於 rule 的改善預測準確率的能力，換言之就是 entropy 的測量(或資訊的數量)

$$\inf oT_{ij}=-\sum_{w=1}^{k}\left(\frac{freqT_{ij}^{\ w}}{\left|T_{ij}\right|}\right)*\log_2\left(\frac{freqT_{ij}^{\ w}}{\left|T_{ij}\right|}\right) \qquad (8)$$

k：類別(class)數

$\left|T_{ij}\right|$：is the total number of cases in partition T_{ij} (partition containing the cases where attribute A_i has value V_{ij});

$freqT_{ij}^{\ w}$：is the number of cases in partition T_{ij} that have class w.

為了進行正規化其 $\inf oT_{ij}^{ij}$ 值必需介於 0 與 $\log_2(k)$ 之間：

Range：$0 \le \inf oT_{ij} \le \log_2(k)$

並使用以下方程式進行正規化：

$$\eta_{ij} = \frac{\log_2(k) - \inf oT_{ij}}{\sum_{i}^{a}\sum_{j}^{b_i}\log_2(k) - \inf oT_{ij}} \qquad (9)$$

a：代表屬性的總和(the total number of attributes)

b_i：代表領域屬性 i 的價值總和數(the number of values in the domain of attribute i.)

為了要使用以上的啟發式函數，有兩個地方要注意：

a. 如果 partition T_{ij} 是空的，也就是屬性 A_i 的值 V_{ij} 在 training set 中並不會發生，便將 $\inf oT_{ij}$ 設成最大值，也就是 $\inf oT_{ij} = \log_2(k)$，相當於給 $Term_{ij}$ 最低可能預測能力。

b. 如果 partition T_{ij} 的所有 case 皆屬於相同的類別，則讓 $\inf oT_{ij} = 0$，相當於給 $Term_{ij}$ 最高可能預測能力。

3. 原則修訂(Rule Pruning)

原則修訂的主要目的就是要移除存在於 rule 中不適當的 term，如此，不但潛在的提升了預測的能力，也讓 rule 改善得更簡單化。在各項原則的歸納中，原則修訂是必要的，其作法就是移除某一項 term，一次只移除一個，再計算看看它的 quality 是否有改善，每一個 term 都要輪到，這個程序要執行一直到只剩一個 term 或移除的 term 不再會對原則的 quality 有任何的改善為止，其計算 quality 的方程式如下：

$$Q = \left(\frac{TruePos}{TruePos + FalseNeg}\right) \times \left(\frac{TrueNeg}{FalsePos + TrueNeg}\right) \qquad (10)$$

TruePos (true positives)：在 rule 中已預測類別中所涵蓋的 case 數

FalseNeg (false negatives)：在 rule 中已預測類別中所未涵蓋的 case 數

FalsePos (false positives)：在 rule 中不同於已預測類別中所涵蓋的 case 數

TrueNeg (true negatives)：在 rule 中不同於已預測類別中所未涵蓋

的 case 數

而 $0 \le Q \le 1$

4. 費洛蒙更新(Pheromone Updating)

使用以下方程式計算費洛蒙的初始值

$$\tau_{ij}(t=0) = \frac{1}{(\sum_{i=1}^{a} b_i)} \qquad (11)$$

k：class 數

a：屬性的總合(the total number of attributes)

b_i：代表領域屬性 i 的價值總和數(the number of values in the domain of attribute i.)

更新費洛蒙有兩個基本的概念，那就是：

1. 被使用的 term 要增加它的費洛蒙

quality 的計算方程式如下：

$$Q = (\frac{TruePos}{TruePos + FalseNeq}) \times (\frac{TrueNeg}{FalsePos + TrueNeg}) \qquad (10)\text{同上}$$

各項 $term_{ij}$ 的費洛蒙更新是根據以下的方程式：

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) \cdot Q, \forall\, i, j \in to\ the\ rule \qquad (12)$$

2. 未被用的 term 要減少它的費洛蒙，就像真的螞蟻一樣費洛蒙是會被蒸發的。即針對每一個費洛蒙 t_{ij} 的值做正規化，也就是說當費洛蒙的值根據方程式

3.1.2.4 執行後而並未增加費洛蒙時，這一項 term 就必須減少它的費洛蒙。

3.2 貝式分類法技術

3.2.1 貝式分類法架構流程圖

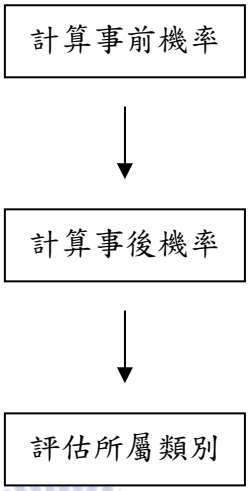


圖 3-4 貝式分類流程圖

3.2.2 貝式分類法技術說明

貝氏定理[13][18]：若事件 C1,C2,…Cn 為樣本空間之 n 類別資料集合，在給定一個觀察量 X=[x1,x2,…,xm]T 其含有 m 個特徵參數，希望使用貝氏分類法決定此一觀察量 X 所屬的分類 C_i，並且期望分類時錯誤的機率可以達到最小。由貝氏定理可以得知以下之公式。

$$P(C_i|X)=\frac{P(C_i)P(X|C_i)}{P(X)} \qquad (13)$$

其中 $P(C_i)$ 為事前機率，代表 C_i 這個類別發生的機率，$P(X)$ 為常數，$P(X|C_i)$ 為概似機率代表 C_i 類別中出現觀察量 X 的機率，$P(C_i|X)$ 為事後機率被使用來做為判斷觀察量 X 所屬於的分類 C_i 的依據，其判斷方式如以下之公式。

$$X \in C_i \quad if \quad P(C_i|X) > P(C_j|X) \quad for \quad i \le j \le n \quad i \ne j \quad (14)$$

當我們要判斷某特徵值 x 究竟屬於哪一個類別時，則我們僅需估算類別 Ci 與類別 Cj 之間的相似率（likelihood ratio）R：

$$R = \frac{P(C_i|X)}{P(C_j|X)} = \frac{P(C_i)P(X|C_i)}{P(C_j)P(X|C_j)} \quad (15)$$

假如 R > 1，表示 x 比較偏向類別 Ci；反之，假如 R < 1，表示 x 比較偏向類別 Cj。

3.3 決策樹分類法技術

3.3.1 決策樹分類法架構流程圖

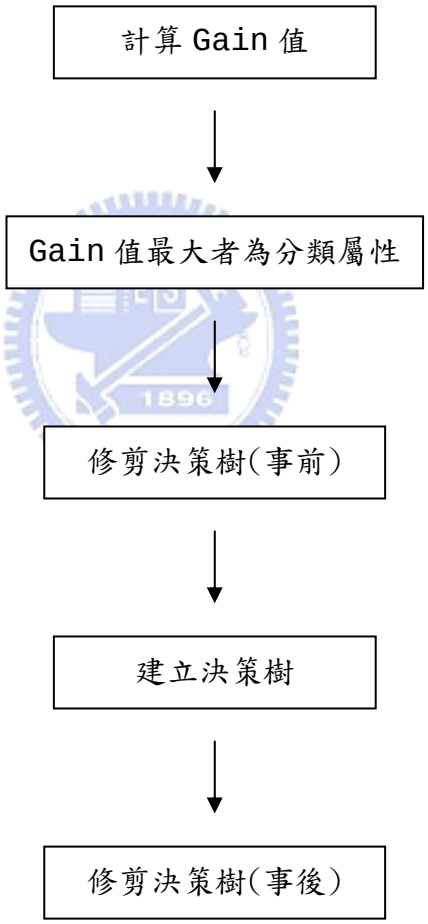


圖 3-5 決策樹分類流程圖

3.3.2 決策樹分類法技術說明

Information Gain 是一種資訊理論的屬性選擇法，由 Quinlan 於 1979 年提出並使用在 ID3 決策樹歸納演算法中。information gain 比較正式的定義為，I(X)是測試前的資訊，代表訓練集合被種類(classification)分割後的資訊；E(Ak, X)是測試後的資訊，代表訓練集合被屬性 Ak 測試後每個子集合內的資訊。

其方程式公式如下：

X：A finite set of examples.

$\{A_1,\cdots,A_p\}$：a set of attribute

$$Gain(A_k, X) = I(X) - E(A_K, X) \tag{16}$$

$$E(A_k, X) = \sum_{i=1}^{n} \frac{|X_i|}{|X|} I(X_i) \tag{17}$$

可針對以上之方程式所產生的決策樹陸續修剪或等到一顆完整的決策樹完成之後，再修剪成人們易懂的規則，

第四章 實驗設計

4.1 實驗程序流程

本實驗設計為一開始先蒐集兩組資料，將兩組資料建立成系統可接受的格式，也就是將原本的 Excel 檔轉換成 ARFF 格式的檔案，之後將資料針對三種分類法，也就是貝氏分類法、C4.5 決策樹分類法及 Ant-Miner 來進行比較，並將其結果進行分析。

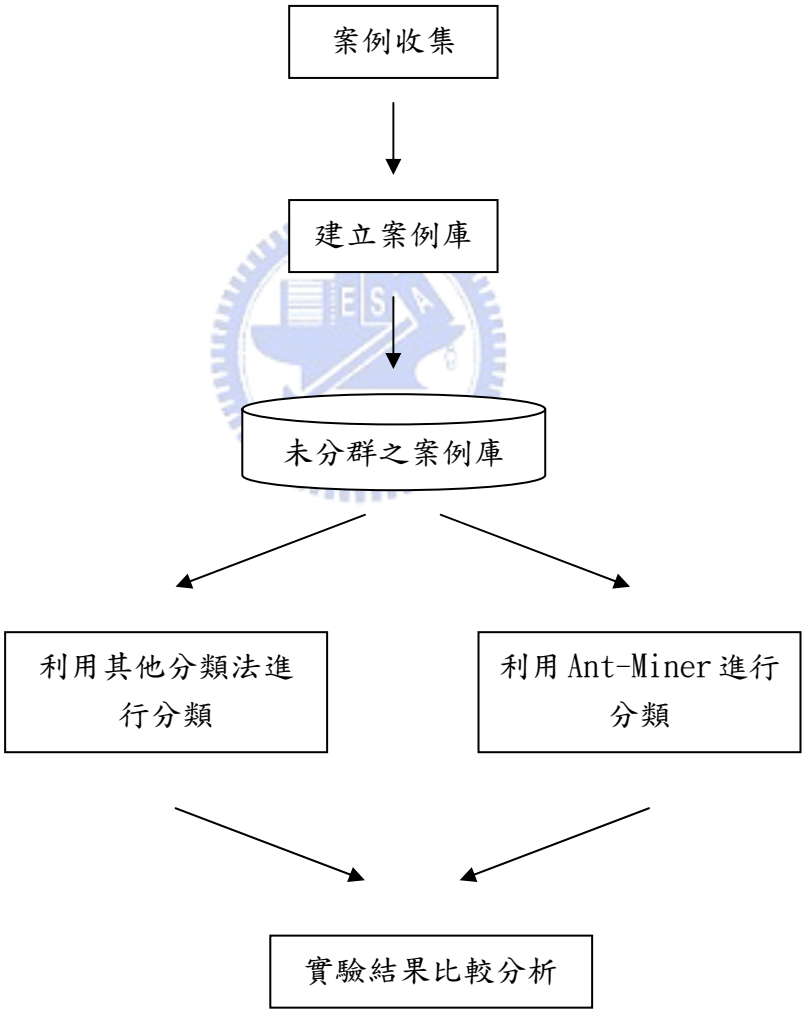


圖 4-1 實驗程序流程圖

4.2 實驗資料

實驗的資料共分兩組進行測試，第一組資料所採用的是 UCI 中所提供的隱形眼鏡相關實驗資料，其資料是根據年紀及淚液等相關因素來判斷適合配載那一種隱形眼鏡。第二組的資料則採用某食品研究所隨機所發出的問卷調查資料，資料為實際查訪並未做任何修改之資料，但因其項目非常繁鎖，所以僅用人工挑選幾個重要欄位來進行實驗，其資料是根據各年齡層、性別及工作性質等來判定他喜歡那一種酒類，因有人喜歡的酒類不止一種，本研究僅保留其最喜歡的酒類來進行研究，而由於其調查之資料量太大故表 4-2 僅呈現部份資料。

表 4-1 第一組實驗資料(UCI)

age	spectacle-prescrip	astigmatism	tear-prod-rate	contact-lenses
young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	yes	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	normal	hard
pre-presbyopic	myope	no	normal	soft
pre-presbyopic	myope	yes	reduced	none
pre-presbyopic	myope	yes	normal	hard
pre-presbopic	hypermetrope	no	reduced	none
pre-presbyopic	hypermetrope	no	normal	soft
pre-presbyopic	hypermetrope	yes	reduced	none
pre-presbyopic	hypermetrope	yes	normal	none
pre-presbyopic	myope	no	reduced	none
presbyopic	myope	no	normal	none
presbyopic	myope	yes	reduced	none
presbyopic	hypermetrope	no	reduced	none
presbyopic	hypermetrope	no	normal	soft
presbyopic	hypermetrope	yes	reduced	none
presbyopic	hypermetrope	yes	normal	none
presbyopic	myope	no	reduced	none
presbyopic	myope	yes	normal	hard

第一組資料共有 24 筆，因資料筆數不多故僅分五次來進行資料交叉驗證，隨機抽取 4 筆或 5 筆當 test data，其餘的則當做 training data，其分次明細如下表：

表 4-2 第一組實驗資料交叉驗證筆數分配

	第一次	第二次	第三次	第四次	第五次
Training Data	19	19	19	20	19
test data	5	5	5	4	5

表 4-3 第二組實驗資料(某食品研究所)

性別	年齡	教育程度	職業	婚姻狀況	宗教	血型	星座	是否喜歡喝酒	酒類
男	50-59 歲	國中或高中職	以勞力為主之員工	已婚	無	A	天秤座	喜歡	啤酒
男	40-49 歲	專科或大學	負責人或部門主管	已婚	無	A	獅子座	喜歡	啤酒
女	30-39 歲	專科或大學	以勞力為主之員工	已婚	道教	O	獅子座	喜歡	啤酒
男	20-29 歲	專科或大學	以勞力為主之員工	未婚	無	O	魔羯座	不喜歡	無
女	20-29 歲	專科或大學	以腦力為主之員工	未婚	無	AB	金牛座	喜歡	梅酒
女	15-19 歲	國中或高中職	學生	未婚	無	O	獅子座	不喜歡	無
男	20-29 歲	國中或高中職	負責人或部門主管	未婚	無	B	巨蟹座	喜歡	啤酒
女	40-49 歲	專科或大學	負責人或部門主管	已婚	佛教	O	水瓶座	不喜歡	無
女	20-29 歲	專科或大學	學生	未婚	道教	O	處女座	不喜歡	無
女	30-39 歲	國中或高中職	負責人或部門主管	已婚	無	A	巨蟹座	不喜歡	無
男	50-59 歲	國中或高中職	負責人或部門主管	已婚	佛教	AB	金牛座	不喜歡	無
女	50-59 歲	國中或高中職	以腦力為主之員工	已婚	無	A	金牛座	不喜歡	無
女	40-49 歲	國中或高中職	家庭主婦	已婚	道教	O	射手座	喜歡	葡萄酒
女	30-39 歲	國中或高中職	家庭主婦	已婚	佛教	B	處女座	喜歡	啤酒
男	20-29 歲	專科或大學	學生	未婚	佛教	B	天秤座	喜歡	葡萄酒
男	20-29 歲	國中或高中職	以腦力為主之員工	未婚	無	O	金牛座	喜歡	啤酒
男	30-39 歲	專科或大學	負責人或部門主管	已婚	無	B	水瓶座	喜歡	啤酒
女	30-39 歲	國中或高中職	負責人或部門主管	已婚	佛教	O	巨蟹座	不喜歡	無
男	50-59 歲	專科或大學	負責人或部門主管	已婚	佛教	B	雙魚座	不喜歡	無
男	20-29 歲	專科或大學	學生	未婚	無	O	雙子座	喜歡	啤酒

....

....

....

....

第二組資料共有 600 筆，因資料筆數較多，故分十次來進行資料交叉驗證，隨機抽取 60 筆當 test data，540 筆為 training data。

4.3 軟體說明

4.3.1 Weka 軟體

Weka 是一套由 Java 寫成，針對資料探勘(Data Mining)且集合各種機器學習(Machine Learning)演算法而成的免費軟體，其所輸入的資料格式可為 CSV 純文字格式及 ARFF(Attribute-Relation File Format)格式，它的下載網址為 http://www.cs.waikato.ac.nz/ml/weka/。

此套軟體所包含的功能有以下四項：

1. 資料的事前處理(data pre-processing)
2. 分類(classification, regression)
3. 分群(clustering)
4. 關聯規則(association rules)
5. 圖表呈現(visualization.)

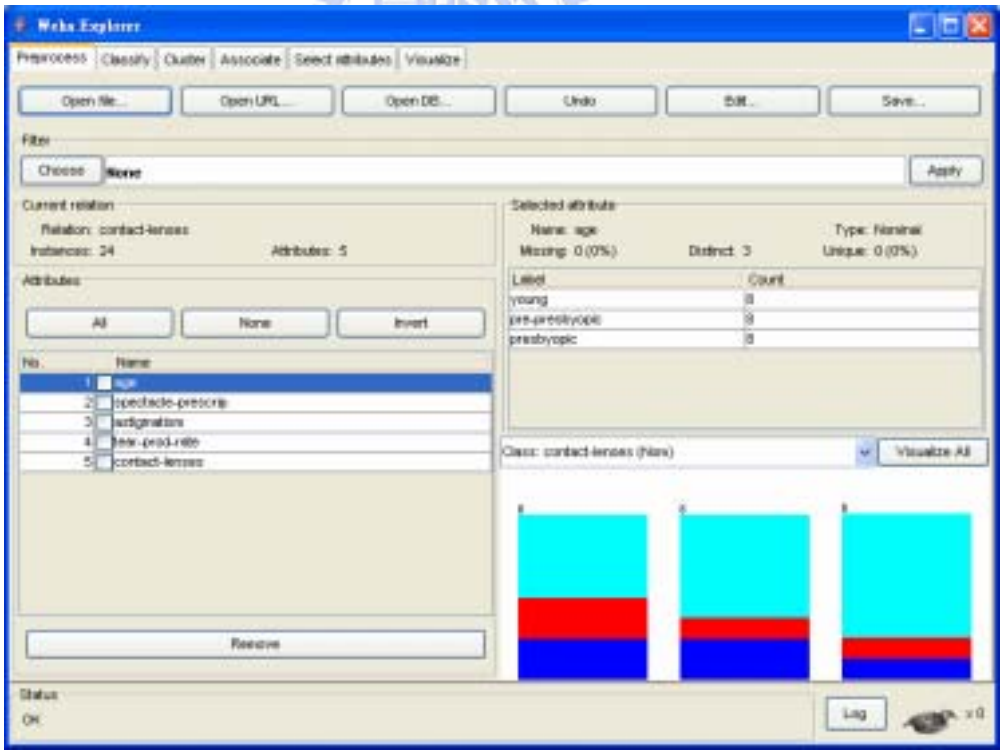


圖 4-2 Weka 軟體圖形介面

4.3.2 Ant_Miner 軟體

這一套稱為 Ant-Miner(Ant Colony-based Data Miner)的圖形介面工具軟體，是由 Java 所寫成的一套最新版的資料探勘演算法，它是由 Rrfael 等學者在 2002 年所提出的，它也有方便的使用者圖形介面，並以 ACO(Ant Colony Optimization)作為基本的概念，而其輸入資料格式與 Weka 軟體相同，是使用 ARFF 格式來進行輸入，它的下載網址為 http://iridia.ulb.ac.be/~mdorigo/ACO/aco-code/public-software.html

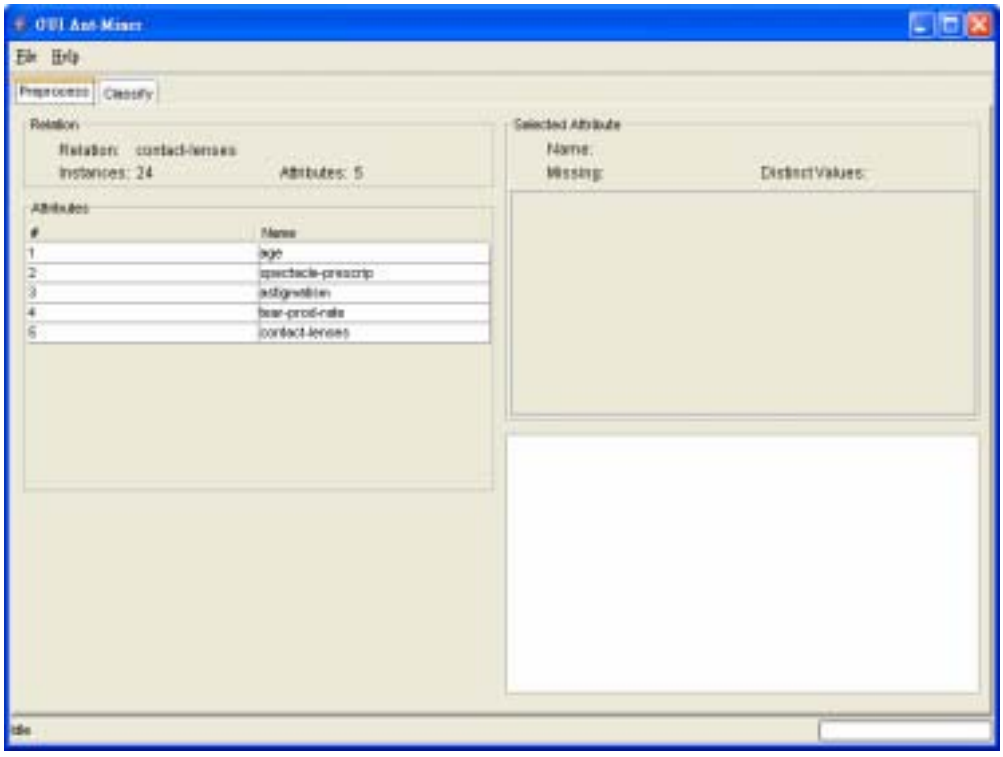


圖 4-3 Ant-Miner 軟體圖形介面

4.4 演算法變數定義

4.4.1 NaiveBayes 變數定義

1. debug：若設定在 True 則會顯示額外的資訊，反之則無。

2. useKernelEstimator：若設定 True 則使用數值型態的屬性評估，而不是用常態的分配。
3. useSupervisedDiscretization – 若設定 True 則將數值型態屬性轉換成名詞性屬性，反之則無。

4.4.2 C4.5 變數定義

1. binarySplits：建立決策樹時，是否針對使用 binary 的方式切開名詞性的屬性。
2. confidenceFactor：設定修剪時 confidence factor 的值(愈小的值得到愈多的修剪)。
3. debug：若設為 true, 則會顯示額外的資訊。
4. minNumObj：每個葉節點的最小資料數。
5. numFolds：使用 reduced-error pruning 時，決定資料的總數 One fold is used for pruning, the rest for growing the tree。
6. reducedErrorPruning：是否使用 reduced-error pruning 取代 C.4.5 pruning.
7. saveInstanceData：是否儲存訓練資料。
8. seed：使用降低錯誤修剪時，seed 使用於隨機的資料。
9. subtreeRaising：當修剪時，是否考慮分枝增加。
10. unpruned：修剪是否執行。
11. useLaplace：是否根據 Laplace 計算葉節點。

4.4.3 Ant-Miner 變數定義

1. 驗證群數(Number of cross-validation folds)：將 data set 分割成所設定之群數，以進行交叉驗證。
2. 螞蟻數 Number of ants：此為族群的總數，只有最好的螞蟻的路徑才能更新費洛蒙，如果不是則將其設為 1。
3. 每個 rule 的最少 case 數(Minimum number of cases per rule)：每條 rule 在必須包含的最少 case 數，以避免 rule 的 overfitting。
4. 未包含的最大 case 數(Maximum number of uncovered cases in the training set)：搜尋 rule 的程序會一直執行，直到訓練資料小於這個欄位所設定的值。
5. 螞蟻收斂的 rule 數(Number of rules used to test the convergence of the ants)：當螞蟻找出的 rule 與前一隻相同時，系統會將其合併成一個 rule。

6. 執行次數(Number of iterations)：螞蟻族群所可執行的最多次數，當螞蟻收斂或執行次數達到時才會停止並啟動另一個執行

第五章 實驗結果比較分析

5.1 參數值設定

5.1.1 NaiveBayes 參數值設定

第一組(UCI)

1. debug = False
2. useKernelEstimator = False
3. useSupervisedDiscretization = False

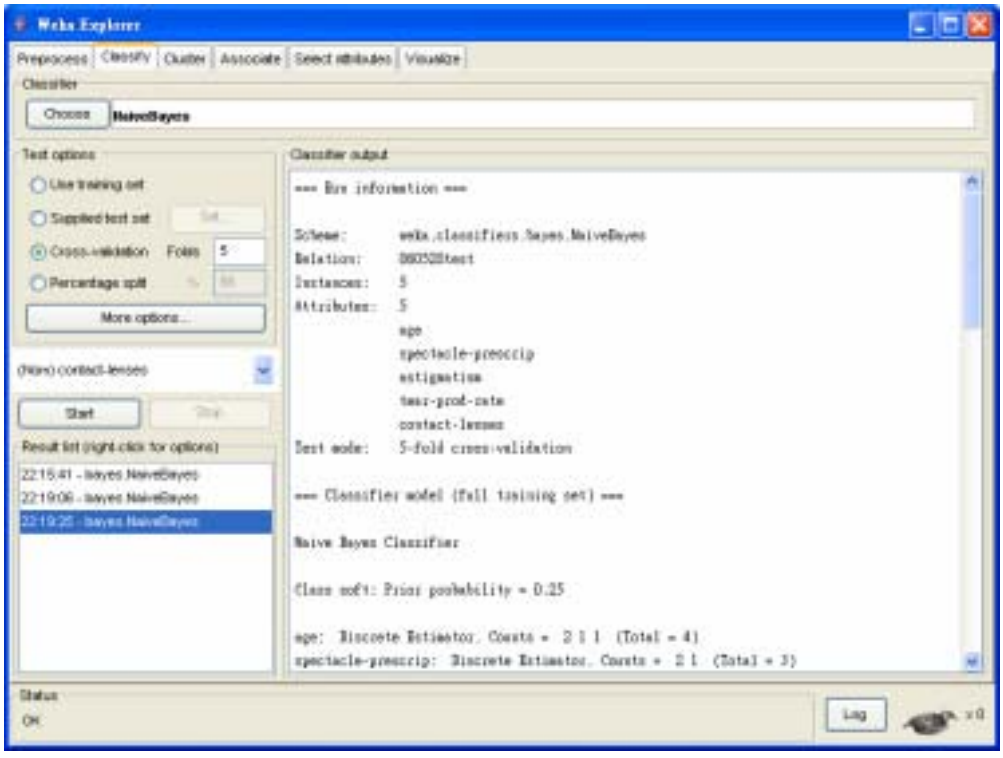


圖 5-1 Weka 軟體第一組分類執行輸出畫面

第二組(某食品研究所)

1. debug = False
2. useKernelEstimator = False
3. useSupervisedDiscretization = False

5.1.2 C4.5 決策樹參數值設定

第一組資料及第二組資料皆設定相同的參數值

1. binarySplits = False
2. confidenceFactor = 0.25
3. debug = False
4. minNumObj = 2
5. numFolds = 3
6. reducedErrorPruning = False
7. saveInstanceData = False
8. seed = 1
9. subtreeRaising = True
10. unpruned = False
11. useLaplace = False

5.1.3 Ant-Miner 變數值設定

第一組

1. cross-validation folds = 5
2. Number of Ants (No_of_ants) = 24;
3. Minimum number of cases per rule (Min_cases_per_rule) = 5
4. Maximum number of uncovered cases in the training set (Max_uncovered_cases) = 10
5. Number of rules used to test convergence of the ants (No_Rules_Converg) = 10

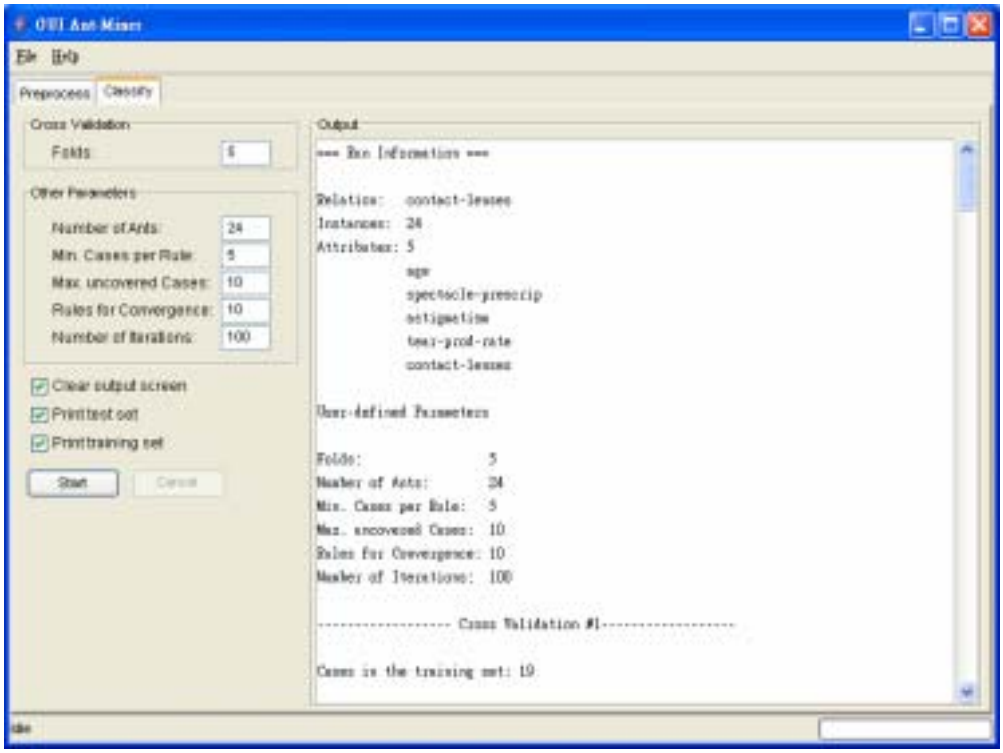


圖 5-3 Ant-Miner 軟體第一組分類執行輸出畫面

第二組

1. cross-validation folds = 10
2. Number of Ants (No_of_ants) = 10;
3. Minimum number of cases per rule (Min_cases_per_rule) = 10
4. Maximum number of uncovered cases in the training set (Max_uncovered_cases) = 10
5. Number of rules used to test convergence of the ants (No_Rules_Converg) = 10

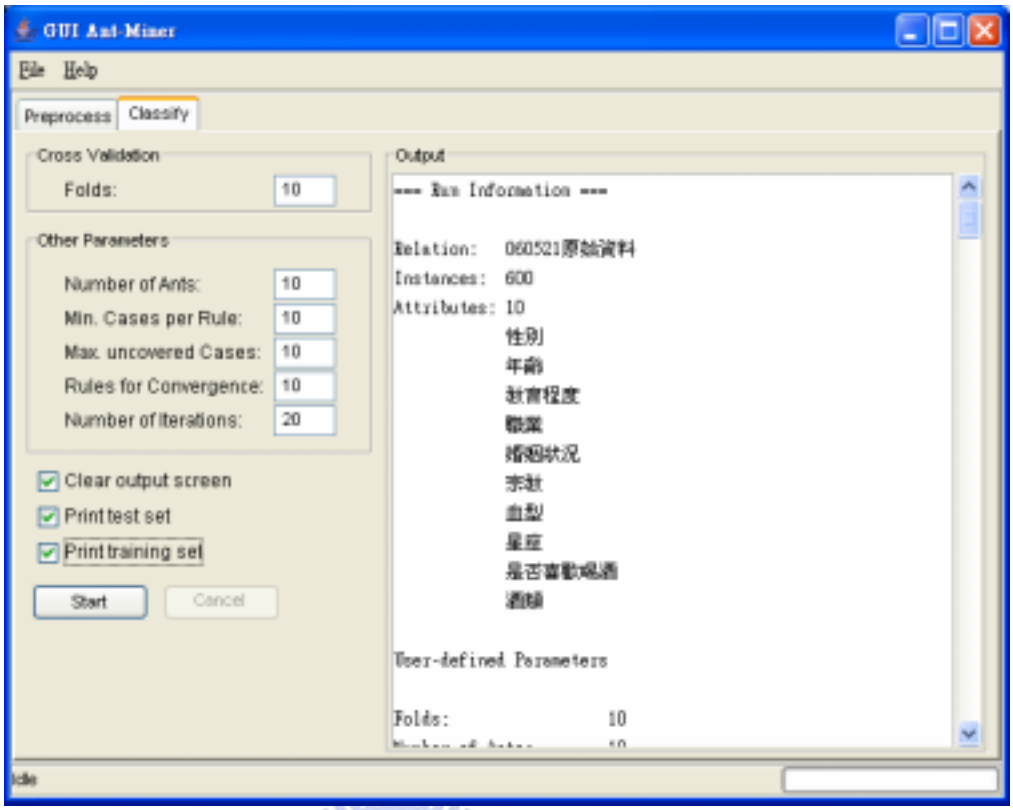


圖 5-4 Ant-Miner 軟體第二組分類執行輸出畫面

5.2 執行結果

5.2.1 貝氏分類法執行結果

從表 5-2 到表 5-3 呈現的分別是透過 NaiveBayes 執行實驗所得到的相關數據。

表 5-2 NaiveBayes 執行第一組測試資料實驗數據

Prior probability	第一次	第二次	第三次	第四次	第五次
Class soft	0.25	-	0.25	0.5	0.29
Class hard	0.38	0.29	0.25	-	-
Class none	0.38	0.71	0.5	0.5	0.71

表 5-3 NaiveBayes 執行第二組測試資料實驗數據

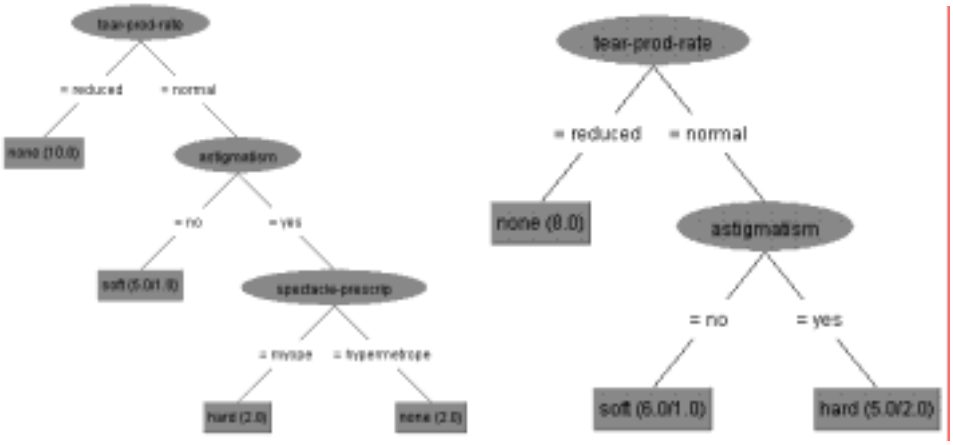
Prior probability	第一次	第二次	第三次	第四次	第五次
Class 啤酒	0.17	0.3	0.12	0.31	0.23
Class 葡萄酒	0.08	0.07	0.14	0.03	0.08
Class 威士忌	0.03	0.06	0.03	0.04	-
Class 梅酒	0.03	0.03	0.03	0.03	0.03
Class 紹興酒	-	0.03	-	-	-
Class 米酒	-	0.03	-	0.03	-
Class 高粱酒	-	-	0.05	0.03	0.03
Class 藥酒	-	-	-	0.03	0.03
Class 無	0.69	0.48	0.64	0.5	0.61

Prior probability	第六次	第七次	第八次	第九次	第十次
Class 啤酒	0.28	0.26	0.34	0.21	0.3
Class 葡萄酒	0.09	0.09	0.06	0.04	0.1
Class 威士忌	0.03	-	0.03	0.04	-
Class 梅酒	-	-	-	0.04	-
Class 紹興酒	0.03	-	0.04	0.03	-
Class 米酒	0.04	0.05	0.03	0.04	-
Class 高粱酒	0.03	0.03	0.04	0.03	-
Class 藥酒	0.03	0.03	-	-	-
Class 白蘭地	0.03	-	-	-	-
Class 無	0.45	0.55	0.45	0.56	0.6

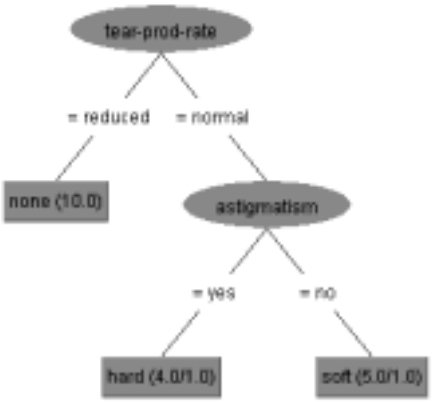
5.2.2 決策樹分類法執行過程

從圖 5-5 呈現的分別是透過決策樹執行實驗所得到的相關數據。

圖 5-5 C4.5 決策樹執行第一組訓練資料產生之決策樹



第一次　　第二次

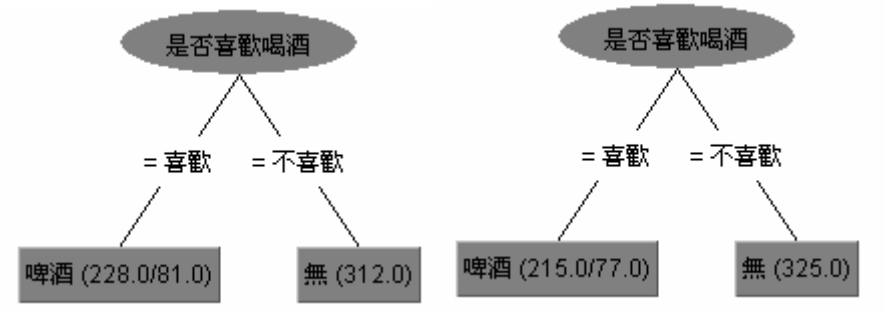


第三次

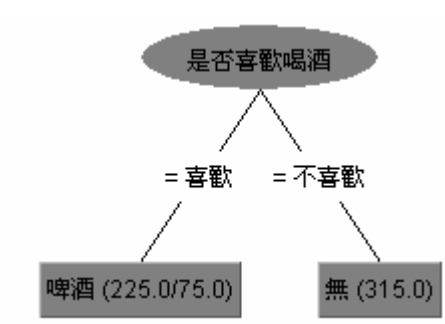
第四次

第五次

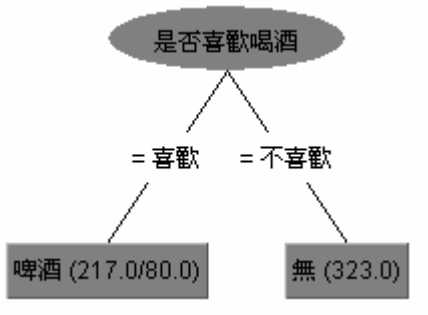
圖 5-6 C4.5 決策樹執行第二組訓練資料產生之決策樹



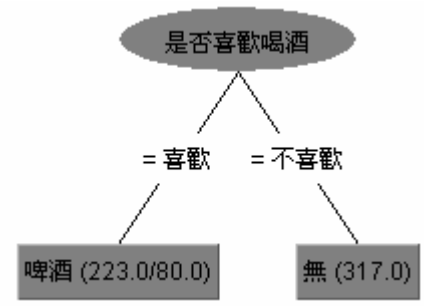
第一次　　第二次



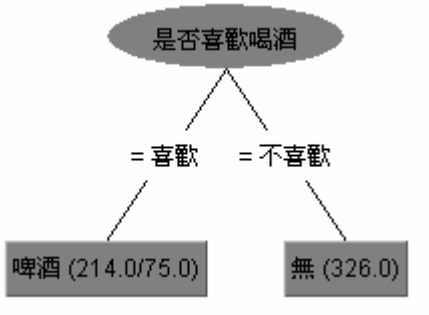
第三次



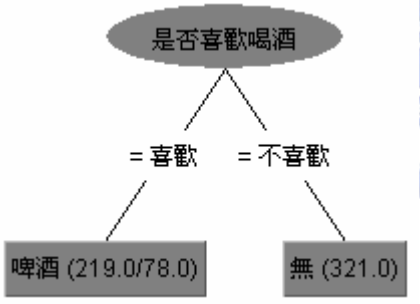
第四次



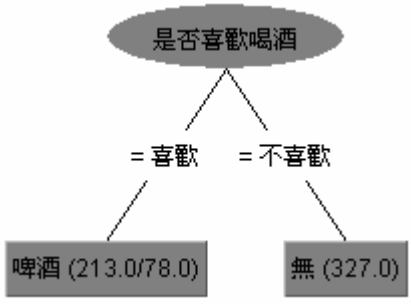
第五次



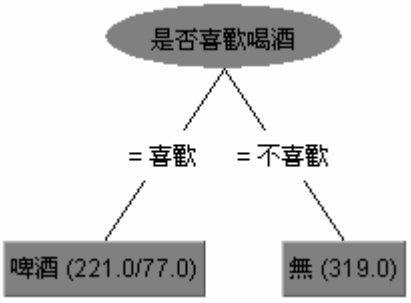
第六次



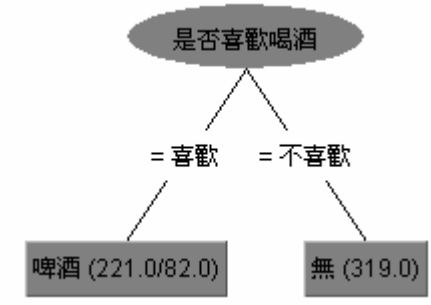
第七次



第八次



第九次



第十次

5.2.3 螞蟻分類法執行結果

表 5-4 Ant-miner 執行第一組資料所找出的分類 rule

次數	分類的 rule
第一次	IF astigmatism = 'no' AND tear-prod-rate = 'normal' THEN 'soft' IF tear-prod-rate = 'reduced' THEN 'none'
第二次	IF astigmatism = 'yes' AND tear-prod-rate = 'normal' THEN 'hard' IF tear-prod-rate = 'reduced' THEN 'none'
第三次	IF astigmatism = 'no' AND tear-prod-rate = 'normal' THEN 'soft' IF tear-prod-rate = 'reduced' THEN 'none'
第四次	IF astigmatism = 'yes' AND tear-prod-rate = 'normal' THEN 'hard' IF tear-prod-rate = 'reduced' THEN 'none'
第五次	IF tear-prod-rate = 'reduced' THEN 'none'

表 5-5 Ant-miner 執行第二組資料所找出的分類 rule

次數	分類的 rule
第一次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒' IF 職業 = '以勞力為主之員工' THEN '啤酒' IF 教育程度 = '專科或大學' THEN '啤酒' IF 性別 = '男' AND 職業 = '負責人或部門主管' THEN '啤酒' IF 教育程度 = '國中或高中職' AND 血型 = 'O' THEN '啤酒' IF 性別 = '男' AND 血型 = 'B' THEN '啤酒' IF 教育程度 = '國中或高中職' THEN '啤酒'
第二次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'

	IF 宗教 = '無' THEN '啤酒' IF 性別 = '男' AND 宗教 = '道教' THEN '啤酒' IF 婚姻狀況 = '未婚' AND 血型 = 'B' THEN '葡萄酒' IF 教育程度 = '專科或大學' THEN '啤酒' IF 職業 = '以勞力為主之員工' AND 宗教 = '佛教' THEN '啤酒' IF 性別 = '男' AND 教育程度 = '國中或高中職' AND 宗教 = '佛教' THEN '啤酒
第三次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒' IF 宗教 = '無' THEN '啤酒' IF 職業 = '以勞力為主之員工' THEN '啤酒' IF 血型 = 'O' THEN '啤酒' IF 性別 = '男' AND 職業 = '負責人或部門主管' THEN '啤酒' IF 性別 = '男' AND 職業 = '以腦力為主之員工' THEN '啤酒' IF 血型 = 'B' THEN '葡萄酒'
第四次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒' IF 宗教 = '無' THEN '啤酒' IF 職業 = '以勞力為主之員工' THEN '啤酒' IF 教育程度 = '專科或大學' THEN '啤酒' IF 教育程度 = '國中或高中職' AND 宗教 = '道教' THEN '啤酒' IF 性別 = '男' AND 教育程度 = '國中或高中職' THEN '啤酒'
第五次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒' IF 宗教 = '無' THEN '啤酒' IF 職業 = '以勞力為主之員工' THEN '啤酒' IF 教育程度 = '專科或大學' THEN '啤酒' IF 教育程度 = '國中或高中職' AND 宗教 = '道教' THEN '啤酒' IF 性別 = '男' AND 教育程度 = '國中或高中職' THEN '啤酒'
第六次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'

	IF 宗教 = '無' THEN '啤酒' IF 職業 = '以勞力為主之員工' THEN '啤酒' IF 教育程度 = '專科或大學' THEN '啤酒' IF 教育程度 = '國中或高中職' AND 宗教 = '道教' THEN '啤酒' IF 性別 = '男' AND 教育程度 = '國中或高中職' THEN '啤酒'
第七次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '男' THEN '啤酒' IF 婚姻狀況 = '未婚' THEN '啤酒' IF 教育程度 = '國中或高中職' AND 婚姻狀況 = '已婚' THEN '啤酒' IF 婚姻狀況 = '已婚' THEN '葡萄酒'
第八次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒' IF 教育程度 = '專科或大學' THEN '啤酒' IF 性別 = '男' AND 年齡 = '15-19 歲' THEN '葡萄酒' IF 職業 = '以勞力為主之員工' THEN '啤酒' IF 性別 = '男' AND 職業 = '負責人或部門主管' THEN '啤酒' IF 職業 = '以腦力為主之員工' THEN '啤酒' IF 教育程度 = '國中或高中職' THEN '啤酒'
第九次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '男' AND 婚姻狀況 = '已婚' THEN '啤酒' IF 婚姻狀況 = '未婚' THEN '啤酒' IF 婚姻狀況 = '已婚' AND 宗教 = '佛教' THEN '葡萄酒' IF 血型 = 'A' THEN '啤酒' IF 宗教 = '道教' THEN '葡萄酒'
第十次	IF 是否喜歡喝酒 = '不喜歡' THEN '無' IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒' IF 宗教 = '無' THEN '啤酒' IF 職業 = '以勞力為主之員工' THEN '啤酒' IF 教育程度 = '專科或大學' THEN '啤酒' IF 教育程度 = '國中或高中職' AND 宗教 = '道教' THEN '啤酒' IF 性別 = '男' AND 教育程度 = '國中或高中職' THEN '

	啤酒'

5.3 分類正確率

5.3.1 第一組分類正確率比較

表 5-6 呈現的是第一組資料執行五次交叉驗證的結果，由此可以很明確的看到 Ant-Miner 的分類正確率優於 NaiveBayes 及 Decision Tree。

表 5-6 第一組分類正確率

	第一次	第二次	第三次	第四次	第五次	平均
NaiveBayes	60%	60%	20%	75%	40%	51%
D Tree	0%	80%	20%	0%	80%	36%
Ant-Miner	100%	100%	80%	75%	60%	83%

5.3.2 第二組分類正確率比較

表 5-7 呈現的是第二組資料針對 NaiveBayes 及 Ant-Miner 兩種分類法執行十次交叉驗證的結果，我們可以發現資料量較大時，也就是可訓練的資料增加時 NaiveBayes 及 Decision Tree 的分類正確率也會提升，其分類正確率優先順序為 Decisrion Tree、Ant-Miner 及 NaiveBayes。

表 5-7 第二組分類正確率

	第一次	第二次	第三次	第四次	第五次
NaiveBayes	90%	80%	90%	86.67%	81.67%
D Tree	90%	83.33%	85%	88.33%	88.33%
Ant-Miner	83.33%	85%	85%	80%	88.33%

	第六次	第七次	第八次	第九次	第十次	平均
NaiveBayes	70%	75%	78.33%	78.33%	95%	82.5%
D Tree	80%	85%	85%	83.33%	91.67%	85.99%
Ant-Miner	81.67%	80%	93.33%	81.67%	93.33%	85.17%

5.4 分類效率

5.4.1 第一組分類效率比較

在表 5-8 上為第一組之分類效率其優先順序依序是 Decision Tree 優於 Ant-Miner，而 Ant-Miner 優於 NaiveBayes。

表 5-8 第一組分類效率

單位：秒

	第一次	第二次	第三次	第四次	第五次	平均
NaiveBayes	0.05	0.04	0.03	0.04	0.03	0.038
D Tree	0.01	0.01	0.01	0.01	0.01	0.01
Ant-Miner	0.078	0.109	0.063	0.297	0.062	0.1218

5.4.2 第二組分類效率比較

而在表 5-9 上為第二組之分類效率其優先順序依序是 Decision Tree 優於 NaiveBayes，而 NaiveBayes 優於 Ant-Miner。

表 5-9 第二組分類效率

單位：秒

	第一次	第二次	第三次	第四次	第五次
NaiveBayes	2.12	1.86	1.95	2.25	1.52
D Tree	0.02	0.03	0.02	0.02	0.01
Ant-Miner	5.266	5.047	5.093	4.875	5.907

	第六次	第七次	第八次	第九次	第十次	平均
NaiveBayes	1.85	2.55	1.75	1.92	2.15	1.992
D Tree	0.02	0.01	0.01	0.02	0.01	0.016
Ant-Miner	5.156	3.109	5.141	3.906	4.984	4.848

5.5 針對 Ant-Miner 參數比較

在之前的技術文件(Rafael et al., 2002)[32]曾提到關於螞蟻數(No_of_ants)設定之問題，實際上並不需要設定成與 case 數一樣多，所以針對此問題，我將針對第二組資料之螞蟻數(No_of_ants)設定的不同做了一些比較。

表 5-10 Ant-Miner 不同螞蟻數分類正確率比較

螞蟻數	第一次	第二次	第三次	第四次	第五次
10	81.67%	80%	93.33%	81.67%	93.33%
100	81.67%	81.67%	88.33%	88.33%	88.33%
200	81.67%	86.67%	85%	83.33%	73.33%
300	86.67%	78.33%	85%	90%	85%
400	81.67%	86.67%	86.67%	85%	86.67%
500	80%	86.67%	80%	80%	85%
600	73.33%	78.33%	93.33%	88.33	83.33%

螞蟻數	第六次	第七次	第八次	第九次	第十次	平均
10	81.67%	80%	93.33%	81.67%	93.33%	85.17%
100	80%	91.67%	86.67%	83.33%	75%	84.5%
200	80%	90%	86.67%	85%	88.33%	82.5%
300	78.33%	83.33%	88.33%	86.67%	88.33%	85%
400	85%	80%	86.67%	88.33%	80%	84.67%
500	88.33%	86.67%	81.67%	88.33%	93.33%	84.97%
600	83.33%	85%	88.33%	85%	83.33%	84.16%

表 5-11 Ant-Miner 不同螞蟻數分類效率比較

單位：秒

螞蟻數	第一次	第二次	第三次	第四次	第五次
10	5.266	5.047	5.093	4.875	5.907
100	39.109	34.031	38.594	42.203	38.11
200	82.296	66.016	86.25	83.181	77.547
300	121.391	128.594	121.546	134.063	120.062
400	143.687	174.266	164.047	162.953	161.266
500	198.844	202.047	202.578	209.797	200.922
600	249.391	263.797	261.828	255.734	191.938

螞蟻數	第六次	第七次	第八次	第九次	第十次	平均
10	5.156	3.109	5.141	3.906	4.984	4.848
100	39	39.687	33.656	39.266	40.484	38.414
200	80.25	90.5	75.797	79.406	69.11	79.035
300	133.891	125.5	121.047	120.453	121.422	104.797
400	166.75	161.594	182.812	163.875	137.688	161.694
500	206.281	195.032	207.812	219.391	202.843	204.555
600	238.734	240.844	268.984	210.344	177.391	235.899

由以上的實驗中我們可以發現螞蟻數(No_of_ants)的增加並不會提升很大的分類正確率，但由於螞蟻數的增加卻會嚴重的影響到執行效率。

第六章 結論與建議

在經過了一些實驗測試後，我們發現，不管是第一組為了機器學習所整理過的資料或第二組經由問卷調查出來，未做任何修正的資料，經過亂數的抽取 test data 後，其 Ant-Miner 分類的正確率雖不是最好的，但不論資料的多寡，它皆能維持在一定的水準之上，所以此項分類技術是值得推薦的。而之所以造成決策樹分類法之分類正確率高於螞蟻分類法的原因在於第二組資料屬於同一類別之數量偏高。

另外，在本實驗中當中所提出的 Ant-Miner，其主要的功能就是將分類的 rule 抽取出來，在第二組十次交叉驗證的執行結果中不難發現以下現象：

1. IF 性別 = '男' AND 婚姻狀況 = '已婚' THEN '啤酒'
2. IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'
3. IF 職業 = '以勞力為主之員工' THEN '啤酒'
4. IF 教育程度 = '國中或高中職' AND 宗教 = '道教' THEN '啤酒'

由以上資料發現：在交叉驗證結果中重覆出現所被抽取出來的 rule，我們可以了解這是一般社會中所普遍存在的現象，因而我們可以在實驗中獲得一些有意義的資訊，所以本研究提出這些技術不僅是要讓大家了解目前最新的分類技術，也希望透過這些技術產生企業的 Know-how，可以更加廣泛的運用到各個企業領域，不但讓學者們所提出的理論得以實踐，也符合企業界對於知識充分的利用與管理的提升。

在使用 Ant-Miner 測試資料時發現輸入之六項參數中，最有趣的就是設定 Number of Ants，在之前的研究報告中，他們所設定的 Number of Ants 的值等於測試資料的 case 數，一開始我也使用同樣原則的設定，但在經過不斷的測試之後發現，Number of Ants 並不需要設定成測試資料的 case 數，因為如此只會拉長執行的時間，但效果並沒有提升，最好的設定值便是經由執行後可能產生的 rule 數，而該如何訂定可能產生的 rule 數，我想除了針對螞蟻演算法中的 heuristic function 及費洛蒙更新的原則外，這也是一個值得未來研究的方向。

另外針對分群的技術中研究者周仕雄曾提出 Ant K-means(AK)演算法(周仕雄，2003)[24]，此種演算法是結合了兩種演算法的優點以提升分群的效率，而在分類技術中，結合螞蟻理論與其它相關演算法的特性，發展出改良式的分類法更是一個值得研究的方向。

參考文獻

[1]. 林志賢，「建立以主題地圖為基礎的知識管理系統」大同大學資訊工程研究所，碩士論文，2004。

[2]. 林典翰，楊烽正，「優加劣減螞蟻擇段系統應用於零工式生產排程問題」，第一屆臺灣作業研究學會學術研討會暨 2004 年科技與管理學術研討會，國立臺灣大學工業工程研究所，2004。

[3]. 朱文正，「考量旅行時間可靠度之車輛途程問題—螞蟻族群演算法之應用」，國立交通大學交通運輸研究所，碩士論文，2003。

[4]. 杜拉克等著，張玉文譯，哈佛商業評論－知識管理，天下文化書坊，2000。

[5]. 呂家源，「螞蟻族群演算法針對排程問題之新解題模式」，暨南大學，碩士論文，2004。

[6]. 周仕雄，「Application of Ant System on Clustering Analysis in Data Ming」，國立臺北科技大學生產系統工程與管理研究所，碩士論文，2003。

[7]. 周世章，「應用螞蟻族群系統構建群落分析演算法」，逢甲大學交通工程與管理學系碩士班，碩士論文，2004。

[8]. 葉怡成，郭耀煌，專家系統方法應用與實作，全欣資訊，1992。

[9]. 陳生祥，「運用資料探勘技術建構企業財務危機預警模式-結合財務與非財務資料」，中原大學資訊管理學系，碩士論文，2005。

[10]. 張金華，「適用於資料挖掘的屬性挑選與快速 k-means 組群化演算法」逢甲大學資訊工程研究所，碩士論文，2000。

[11]. 張炳騰，鍾承志，洪國禎，洪龍廷，「多目標零工式平行機台排程之研究-應用蟻群最佳化演算法」，第一屆臺灣作業研究學會學術研討會暨 2004 年科技與管理學術研討會，東海大學工業工程與經營資訊所，2004。

[12]. 張智星，「資群聚與樣式辨認」，網路線上課程，可由作者之網頁 http://neural.cs.nthu.edu.tw/jang/books/dcpr/連結。

[13]. 黃代鈞，「以遺傳演算法結合貝氏分類法快速篩選與遺傳疾病相關的基因」，長庚大學資訊管理研究所，碩士論文，2004。

[14]. 詹金凌，「整合螞蟻理論與案例式推理於知識管理之應用」，國立臺北科技大學生產系統工程與管理研究所，碩士論文，2003。

[15]. 熊鴻鈞，「螞蟻族群演算法於生產排程之應用」，暨南大學，碩士論文，2003。

[16]. 蕭宗勝，「螞蟻族群演算法應用在組合問題之研究」，銘傳大學，碩士論文，2002。

[17].羅閔隆，「以經驗法則應用在關聯法則門檻值制定之研究」，大葉大學資訊管理學系，碩士論文，2004。

[18].藍中賢，「結合模糊集合理論與貝氏分類法之資料探勘技術-應用於健保局醫療費用審查作業」，元智大學資訊研究所，碩士論文，2000。

[19].張淑珍，「利用一次性的 SQL 改良決策樹建立信用卡審核之信用評等」，東吳大學資訊科學系，碩士論文，2005

[20].Thomas H. Davenport & Laurence Prusak, 胡瑋珊譯，「知識管理：企業組織如何有效運用知識」，中國生產力中心，1999。.

[21].A. Colorni, M. Dorigo, and V. Maniezzo. “The ant system: an autocatatlytic process.” *Technical Report No. 91-016*, Politecnico di Milano, Italy, 1991

[22].A. A. Freitas, “A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery”, *Advances in Evolutionary Computation*,Springer-Verlag,2002.

[23].D. Corne, M. Dorigo and F. Glover, New Ideas in Optimization, McGraw-Hill, 1999.

[24].Fayyad,U.M.,“Data Mining and knowledge Discovery:Making Sense Out of data, ” *IEEE Expert*,Volume 11, Issue 5,pp. 20-25,1996.

[25].M. Dorigo, V. Maniezzo and A. Colorni, “The Ant System: Optimization by a colony of cooperating agents”, *IEEE Transactions on Systems, Man, and Cybernetics*: *Part B*, Vol.26,No.1,pp.1-13,1996.

[26].M Dorigo and Gambardella, L.M., “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem,” *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp.53-66, 1997.

[27].M Dorigo, Gianni Di Caro,Luca M. Gambardella,”Ant Algorithms for Discrete Optimization”,*Artificial Life*,Vol.5,No.3,pp. 137-172,1999.

[28].M Dorigo, “Ant Colony Optimization”, http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html

[29].Michael,J.A. and Linoff G., “Data Mining Technique:for Marketing,Sales and Customer Support”, *Wiley Computer Publishing*, New York, 1997.

[30].P. Bosc, O. Pivert and L. Ughetto, “Database mining for the discovery of extended functional dependencies” *in Proc. NAFIPS 99*, New York, pp. 580–584, June 1999.

[31].P.S. Shelokar,V. K. Jayaraman,B. D. Kulkarni*, ”An ant colony classifier system:application to some process engineering problems”, *Computers and Chemical Engineering 28*, pp.1577-1584, 2004.

[32].Rafael S. Parpinelli,Heitor S. Lopes,Alex A. Freitas, ”An Ant Colony Algorithm for Classification Rule Discovery”, *Parpinelli, Lopes and Freitas*, pp.190-208, 2002

[33]. Rafael S. Parpinelli,Heitor S. Lopes,Member,IEEE,and Alex A. Freitas, ”Data Mining With an Ant Colony Optimization Algorithm”, IEEE Transactions on Evolutionary Computing, Vol 6, No.4, August 2002.

[34]. Richard Jensen*,Qiang Shen, ”Fuzzy-rough data reduction with ant colony optimization” ,*Fuzzy Sets and Systems 149*, pp.5-20, 2005.

[35]. T. M. Mitchell, “Machine Learning”., New York, McGraw-Hill, 1997.

[36]. T. M. Mitchell, “Machine learning and data mining”, *Commun. ACM*, vol. 42, no. 11, 1999.

附錄

A. 第一組資料執行結果

i. Weka 軟體執行結果內容

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes
Relation: 060520test
Instances: 5
Attributes: 5
age
spectacle-prescrip
astigmatism
tear-prod-rate
contact-lenses
Test mode: 5-fold cross-validation

---------- Cross Validation #1----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class soft: Prior probability = 0.25
Class hard: Prior probability = 0.38
Class none: Prior probability = 0.38

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances 3 60 %

Incorrectly Classified Instances	2	40	%
Kappa statistic	0.375		
Mean absolute error	0.349		
Root mean squared error	0.4187		
Relative absolute error	70.4647 %		
Root relative squared error	79.0964 %		
Total Number of Instances	5		

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0.25	0	0	0	soft
0.5	0.333	0.5	0.5	0.5	hard
1	0	1	1	1	none

=== Confusion Matrix ===

a	b	c	<-- classified as
0	1	0	a = soft
1	1	0	b = hard
0	0	2	c = none

---------- Cross Validation #2----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class none: Prior probability = 0.71
Class hard: Prior probability = 0.29

Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	3	60	%
Incorrectly Classified Instances	2	40	%
Kappa statistic	-0.25		

Mean absolute error 0.3143
Root mean squared error 0.3828
Relative absolute error 72.5242 %
Root relative squared error 80.2033 %
Total Number of Instances 5

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.75	1	0.75	0.75	0.75	none
0	0.25	0	0	0	hard

=== Confusion Matrix ===

a	b	<-- classified as
3	1	a = none
1	0	b = hard

---------- Cross Validation #3----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class none: Prior probability = 0.5
Class soft: Prior probability = 0.25
Class hard: Prior probability = 0.25

age: Discrete Estimator. Counts = 2 1 (Total = 3)
spectacle-prescrip: Discrete Estimator. Counts = 2 1 (Total = 3)
astigmatism: Discrete Estimator. Counts = 1 2 (Total = 3)
tear-prod-rate: Discrete Estimator. Counts = 1 2 (Total = 3)

Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances 1 20 %

Incorrectly Classified Instances	4	80	%
Kappa statistic	-0.1111		
Mean absolute error	0.4777		
Root mean squared error	0.5373		
Relative absolute error	104.4951 %		
Root relative squared error	107.3863 %		
Total Number of Instances	5		

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.333	0	1	0.333	0.5	none
0	0.5	0	0	0	soft
0	0.5	0	0	0	hard

=== Confusion Matrix ===

a	b	c	<-- classified as
1	1	1	a = none
0	0	1	b = soft
0	1	0	c = hard

---------- Cross Validation #4----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class soft: Prior probability = 0.5
Class none: Prior probability = 0.5

Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	3	75	%
Incorrectly Classified Instances	1	25	%
Kappa statistic	0.5		

Mean absolute error	0.3701
Root mean squared error	0.4214
Relative absolute error	61.6891 %
Root relative squared error	70.2376 %
Total Number of Instances	4

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.5	0	1	0.5	0.667	soft
1	0.5	0.667	1	0.8	none

=== Confusion Matrix ===

a	b	<-- classified as
1	1	a = soft
0	2	b = none

---------- Cross Validation #5----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class soft: Prior probability = 0.29
Class none: Prior probability = 0.71

Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	2	40 %
Incorrectly Classified Instances	3	60 %
Kappa statistic	-0.3636	
Mean absolute error	0.4423	
Root mean squared error	0.4915	
Relative absolute error	102.0743 %	
Root relative squared error	102.9878 %	

Total Number of Instances 5

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0.5	0	0	0	soft
0.5	1	0.667	0.5	0.571	none

=== Confusion Matrix ===

a	b	<-- classified as
0	1	a = soft
2	2	b = none

ii. Ant-Miner 軟體執行結果內容

=== Run Information ===

Relation: contact-lenses
Instances: 24
Attributes: 5
age
spectacle-prescrip
astigmatism
tear-prod-rate
contact-lenses

User-defined Parameters

Folds: 5
Number of Ants: 24
Min. Cases per Rule: 5
Max. uncovered Cases: 10
Rules for Convergence: 10
Number of Iterations: 100

---------- Cross Validation #1----------

Cases in the training set: 19

Cases in the test set: 5

Rules: 3

IF astigmatism = 'no' AND tear-prod-rate = 'normal' THEN 'soft'
IF tear-prod-rate = 'reduced' THEN 'none'
Default rule: hard

Accuracy rate on the training set: 84.21052631578947 %
Accuracy rate on the test set: 100.0 %

Time taken: 0.078 s.

---------- Cross Validation #2----------

Cases in the training set: 19

Cases in the test set: 5

Rules: 3

IF astigmatism = 'yes' AND tear-prod-rate = 'normal' THEN 'hard'
IF tear-prod-rate = 'reduced' THEN 'none'
Default rule: soft

Accuracy rate on the training set: 84.21052631578947 %
Accuracy rate on the test set: 100.0 %

Time taken: 0.109 s.

---------- Cross Validation #3----------

Cases in the training set: 19

Cases in the test set: 5

Rules: 3

IF astigmatism = 'no' AND tear-prod-rate = 'normal' THEN 'soft'
IF tear-prod-rate = 'reduced' THEN 'none'
Default rule: hard

Accuracy rate on the training set: 89.47368421052632 %
Accuracy rate on the test set: 80.0 %

Time taken: 0.063 s.

---------- Cross Validation #4----------

Cases in the training set: 20

Cases in the test set: 4

Rules: 3

IF astigmatism = 'yes' AND tear-prod-rate = 'normal' THEN 'hard'
IF tear-prod-rate = 'reduced' THEN 'none'
Default rule: soft

Accuracy rate on the training set: 90.0 %
Accuracy rate on the test set: 75.0 %

Time taken: 0.297 s.

---------- Cross Validation #5----------

Cases in the training set: 19

Cases in the test set: 5

Rules: 2

IF tear-prod-rate = 'reduced' THEN 'none'
Default rule: hard

Accuracy rate on the training set: 68.42105263157895 %

Accuracy rate on the test set: 60.0 %

Time taken: 0.062 s.

5-Fold Cross Validation Results

Accuracy Rate on Test Set	Rules Number	Conditions Number
83% +/- 7.68%	2.8 +/- 0.2	2.6 +/- 0.4

Total elapsed time: 0 s.

B. 第二組資料執行結果

i. Weka 軟體執行結果內容

=== Run information ===

Scheme: weka.classifiers.bayes.NaiveBayes
Relation: 060521test-1
Instances: 60
Attributes: 10
性別
年齡
教育程度
職業
婚姻狀況
宗教
血型
星座
是否喜歡喝酒
酒類
Test mode: 10-fold cross-validation

---------- Cross Validation #1----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 啤酒: Prior probability = 0.17
Class 無: Prior probability = 0.69
Class 葡萄酒: Prior probability = 0.08
Class 威士忌: Prior probability = 0.03
Class 梅酒: Prior probability = 0.03

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	54	90 %
Incorrectly Classified Instances	6	10 %
Kappa statistic	0.7662	
Mean absolute error	0.084	
Root mean squared error	0.1889	
Relative absolute error	45.3916 %	
Root relative squared error	63.9773 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.8	0.06	0.727	0.8	0.762	啤酒
0.977	0.063	0.977	0.977	0.977	無
0.75	0	1	0.75	0.857	葡萄酒
0	0.017	0	0	0	威士忌
0	0.017	0	0	0	梅酒

=== Confusion Matrix ===

a	b	c	d	e	<-- classified as
8	1	0	0	1	a = 啤酒
0	43	0	1	0	b = 無

```
  1   0   3   0   0 |   c =  葡萄酒
  1   0   0   0   0 |   d =  威士忌
  1   0   0   0   0 |   e =  梅酒
```

---------- Cross Validation #2----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 啤酒: Prior probability = 0.3
Class 無: Prior probability = 0.48
Class 梅酒: Prior probability = 0.03
Class 葡萄酒: Prior probability = 0.07
Class 威士忌: Prior probability = 0.06
Class 紹興酒: Prior probability = 0.03
Class 米酒: Prior probability = 0.03

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	48	80 %
Incorrectly Classified Instances	12	20 %
Kappa statistic	0.668	
Mean absolute error	0.0998	
Root mean squared error	0.2195	
Relative absolute error	53.4181 %	
Root relative squared error	72.9936 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.947	0.146	0.75	0.947	0.837	啤酒
0.968	0.034	0.968	0.968	0.968	無
0	0.017	0	0	0	梅酒

0	0.018	0	0	0	葡萄酒
0	0.053	0	0	0	威士忌
0	0	0	0	0	紹興酒
0	0	0	0	0	米酒

=== Confusion Matrix ===

a	b	c	d	e	f	g	<-- classified as
18	0	0	0	1	0	0	a = 啤酒
1	30	0	0	0	0	0	b = 無
0	0	0	0	1	0	0	c = 梅酒
2	1	1	0	0	0	0	d = 葡萄酒
2	0	0	1	0	0	0	e = 威士忌
0	0	0	0	1	0	0	f = 紹興酒
1	0	0	0	0	0	0	g = 米酒

---------- Cross Validation #3----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 無: Prior probability = 0.64
Class 葡萄酒: Prior probability = 0.14
Class 啤酒: Prior probability = 0.12
Class 高粱酒: Prior probability = 0.05
Class 梅酒: Prior probability = 0.03
Class 威士忌: Prior probability = 0.03

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	54	90 %
Incorrectly Classified Instances	6	10 %
Kappa statistic	0.7987	
Mean absolute error	0.0831	
Root mean squared error	0.1888	

Relative absolute error	46.542	%
Root relative squared error	64.895	%
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	0	1	1	1	無
0.75	0.058	0.667	0.75	0.706	葡萄酒
0.857	0.038	0.75	0.857	0.8	啤酒
0.5	0	1	0.5	0.667	高粱酒
0	0.017	0	0	0	梅酒
0	0	0	0	0	威士忌

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
41	0	0	0	0	0	a = 無
0	6	2	0	0	0	b = 葡萄酒
0	0	6	0	1	0	c = 啤酒
0	1	0	1	0	0	d = 高粱酒
0	1	0	0	0	0	e = 梅酒
0	1	0	0	0	0	f = 威士忌

---------- Cross Validation #4----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 啤酒: Prior probability = 0.31
Class 無: Prior probability = 0.5
Class 藥酒: Prior probability = 0.03
Class 威士忌: Prior probability = 0.04
Class 梅酒: Prior probability = 0.03
Class 葡萄酒: Prior probability = 0.03
Class 米酒: Prior probability = 0.03
Class 高粱酒: Prior probability = 0.03

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	52	86.6667 %
Incorrectly Classified Instances	8	13.3333 %
Kappa statistic	0.7572	
Mean absolute error	0.0832	
Root mean squared error	0.1981	
Relative absolute error	53.3818 %	
Root relative squared error	72.7215 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.95	0.15	0.76	0.95	0.844	啤酒
1	0.037	0.971	1	0.985	無
0	0	0	0	0	藥酒
0	0	0	0	0	威士忌
0	0.017	0	0	0	梅酒
0	0	0	0	0	葡萄酒
0	0	0	0	0	米酒
0	0	0	0	0	高粱酒

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	<-- classified as
19	1	0	0	0	0	0	0	a = 啤酒
0	33	0	0	0	0	0	0	b = 無
1	0	0	0	0	0	0	0	c = 藥酒
1	0	0	0	1	0	0	0	d = 威士忌
1	0	0	0	0	0	0	0	e = 梅酒
1	0	0	0	0	0	0	0	f = 葡萄酒
1	0	0	0	0	0	0	0	g = 米酒
1	0	0	0	0	0	0	0	h = 高粱酒

---------- Cross Validation #5----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 梅酒: Prior probability = 0.03
Class 啤酒: Prior probability = 0.23
Class 葡萄酒: Prior probability = 0.08
Class 無: Prior probability = 0.61
Class 高粱酒: Prior probability = 0.03
Class 藥酒: Prior probability = 0.03

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	49	81.6667 %
Incorrectly Classified Instances	11	18.3333 %
Kappa statistic	0.6386	
Mean absolute error	0.096	
Root mean squared error	0.2271	
Relative absolute error	52.2055 %	
Root relative squared error	76.7202 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0.034	0	0	0	梅酒
0.714	0.087	0.714	0.714	0.714	啤酒
0	0.071	0	0	0	葡萄酒
1	0.048	0.975	1	0.987	無
0	0	0	0	0	高粱酒
0	0	0	0	0	藥酒

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as

```
 0   0   1   0   0   0 |   a =  梅酒
 0 10   3   1   0   0 |   b =  啤酒
 2   2   0   0   0   0 |   c =  葡萄酒
 0   0   0 39   0   0 |   d =  無
 0   1   0   0   0   0 |   e =  高粱酒
 0   1   0   0   0   0 |   f =  藥酒
```

---------- Cross Validation #6----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 啤酒: Prior probability = 0.28
Class 無: Prior probability = 0.45
Class 葡萄酒: Prior probability = 0.09
Class 白蘭地: Prior probability = 0.03
Class 紹興酒: Prior probability = 0.03
Class 米酒: Prior probability = 0.04
Class 藥酒: Prior probability = 0.03
Class 高粱酒: Prior probability = 0.03
Class 威士忌: Prior probability = 0.03

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	42	70 %
Incorrectly Classified Instances	18	30 %
Kappa statistic	0.5244	
Mean absolute error	0.0909	
Root mean squared error	0.2131	
Relative absolute error	59.5081 %	
Root relative squared error	78.6337 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.833	0.31	0.536	0.833	0.652	啤酒
0.867	0.033	0.963	0.867	0.912	無
0.2	0.018	0.5	0.2	0.286	葡萄酒
0	0	0	0	0	白蘭地
0	0	0	0	0	紹興酒
0	0.034	0	0	0	米酒
0	0	0	0	0	藥酒
0	0	0	0	0	高粱酒
0	0.017	0	0	0	威士忌

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	<-- classified as
15	1	1	0	0	1	0	0	0 |	a = 啤酒
3	26	0	0	0	1	0	0	0 |	b = 無
4	0	1	0	0	0	0	0	0 |	c = 葡萄酒
1	0	0	0	0	0	0	0	0 |	d = 白蘭地
1	0	0	0	0	0	0	0	0 |	e = 紹興酒
2	0	0	0	0	0	0	0	0 |	f = 米酒
1	0	0	0	0	0	0	0	0 |	g = 藥酒
0	0	0	0	0	0	0	0	1 |	h = 高粱酒
1	0	0	0	0	0	0	0	0 |	i = 威士忌

---------- Cross Validation #7----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 無: Prior probability = 0.55
Class 啤酒: Prior probability = 0.26
Class 米酒: Prior probability = 0.05
Class 葡萄酒: Prior probability = 0.09
Class 高粱酒: Prior probability = 0.03
Class 藥酒: Prior probability = 0.03

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	45	75	%
Incorrectly Classified Instances	15	25	%
Kappa statistic	0.5667		
Mean absolute error	0.1112		
Root mean squared error	0.243		
Relative absolute error	54.8195 %		
Root relative squared error	77.5582 %		
Total Number of Instances	60		

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.914	0.08	0.941	0.914	0.928	無
0.75	0.159	0.632	0.75	0.686	啤酒
0	0.034	0	0	0	米酒
0.2	0.073	0.2	0.2	0.2	葡萄酒
0	0	0	0	0	高粱酒
0	0	0	0	0	藥酒

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
32	1	1	1	0	0	a = 無
1	12	1	2	0	0	b = 啤酒
0	2	0	0	0	0	c = 米酒
1	3	0	1	0	0	d = 葡萄酒
0	1	0	0	0	0	e = 高粱酒
0	0	0	1	0	0	f = 藥酒

---------- Cross Validation #8----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 無: Prior probability = 0.45
Class 啤酒: Prior probability = 0.34
Class 威士忌: Prior probability = 0.03
Class 紹興酒: Prior probability = 0.04
Class 高粱酒: Prior probability = 0.04
Class 米酒: Prior probability = 0.03
Class 葡萄酒: Prior probability = 0.06

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	47	78.3333 %
Incorrectly Classified Instances	13	21.6667 %
Kappa statistic	0.6523	
Mean absolute error	0.1019	
Root mean squared error	0.2201	
Relative absolute error	54.4099 %	
Root relative squared error	73.05 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.931	0.032	0.964	0.931	0.947	無
0.864	0.132	0.792	0.864	0.826	啤酒
0	0.017	0	0	0	威士忌
0	0.069	0	0	0	紹興酒
0	0.017	0	0	0	高粱酒
0	0	0	0	0	米酒
0.333	0.018	0.5	0.333	0.4	葡萄酒

=== Confusion Matrix ===

a	b	c	d	e	f	g	<-- classified as
27	1	0	1	0	0	0	a = 無
0	19	1	1	0	0	1	b = 啤酒

0	1	0	0	0	0	0	c = 威士忌
0	1	0	0	1	0	0	d = 紹興酒
0	1	0	1	0	0	0	e = 高粱酒
0	0	0	1	0	0	0	f = 米酒
1	1	0	0	0	0	1	g = 葡萄酒

---------- Cross Validation #9----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 無: Prior probability = 0.56
Class 啤酒: Prior probability = 0.21
Class 威士忌: Prior probability = 0.04
Class 梅酒: Prior probability = 0.04
Class 紹興酒: Prior probability = 0.03
Class 葡萄酒: Prior probability = 0.04
Class 米酒: Prior probability = 0.04
Class 高粱酒: Prior probability = 0.03

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	47	78.3333 %
Incorrectly Classified Instances	13	21.6667 %
Kappa statistic	0.6096	
Mean absolute error	0.0862	
Root mean squared error	0.2074	
Relative absolute error	56.4671 %	
Root relative squared error	77.0234 %	
Total Number of Instances	60	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class

0.973	0.043	0.973	0.973	0.973	無
0.846	0.128	0.647	0.846	0.733	啤酒
0	0	0	0	0	威士忌
0	0.034	0	0	0	梅酒
0	0	0	0	0	紹興酒
0	0.017	0	0	0	葡萄酒
0	0.052	0	0	0	米酒
0	0	0	0	0	高粱酒

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	<-- classified as
36	0	0	0	0	0	1	0	a = 無
1	11	0	0	0	0	1	0	b = 啤酒
0	2	0	0	0	0	0	0	c = 威士忌
0	0	0	0	0	1	1	0	d = 梅酒
0	1	0	0	0	0	0	0	e = 紹興酒
0	1	0	1	0	0	0	0	f = 葡萄酒
0	1	0	1	0	0	0	0	g = 米酒
0	1	0	0	0	0	0	0	h = 高粱酒

---------- Cross Validation #10----------

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class 啤酒: Prior probability = 0.3
Class 葡萄酒: Prior probability = 0.1
Class 無: Prior probability = 0.6

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	57	95	%
Incorrectly Classified Instances	3	5	%
Kappa statistic	0.904		

Mean absolute error	0.0819
Root mean squared error	0.1751
Relative absolute error	23.0721 %
Root relative squared error	41.7773 %
Total Number of Instances	60

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	0.071	0.857	1	0.923	啤酒
0.6	0	1	0.6	0.75	葡萄酒
0.973	0	1	0.973	0.986	無

=== Confusion Matrix ===

a	b	c	<-- classified as
18	0	0	a = 啤酒
2	3	0	b = 葡萄酒
1	0	36	c = 無

ii. Ant-Miner 軟體執行結果內容

=== Run Information ===

Relation: 060521 原始資料
Instances: 600
Attributes: 10
性別
年齡
教育程度
職業
婚姻狀況
宗教
血型
星座
是否喜歡喝酒
酒類

User-defined Parameters

Folds: 10
Number of Ants: 10
Min. Cases per Rule: 10
Max. uncovered Cases: 10
Rules for Convergence: 10
Number of Iterations: 20

---------- Cross Validation #1----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 9

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'
IF 職業 = '以勞力為主之員工' THEN '啤酒'
IF 教育程度 = '專科或大學' THEN '啤酒'
IF 性別 = '男' AND 職業 = '負責人或部門主管' THEN '啤酒'
IF 教育程度 = '國中或高中職' AND 血型 = 'O' THEN '啤酒'
IF 性別 = '男' AND 血型 = 'B' THEN '啤酒'
IF 教育程度 = '國中或高中職' THEN '啤酒'
Default rule: 高粱酒

Accuracy rate on the training set: 86.29629629629629 %
Accuracy rate on the test set: 83.33333333333334 %

Time taken: 5.266 s.

---------- Cross Validation #2----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 9

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'
IF 宗教 = '無' THEN '啤酒'
IF 性別 = '男' AND 宗教 = '道教' THEN '啤酒'
IF 婚姻狀況 = '未婚' AND 血型 = 'B' THEN '葡萄酒'
IF 教育程度 = '專科或大學' THEN '啤酒'
IF 職業 = '以勞力為主之員工' AND 宗教 = '佛教' THEN '啤酒'
IF 性別 = '男' AND 教育程度 = '國中或高中職' AND 宗教 = '佛教' THEN '啤酒'
Default rule: 啤酒

Accuracy rate on the training set: 85.92592592592592 %
Accuracy rate on the test set: 85.0 %

Time taken: 5.047 s.

---------- Cross Validation #3----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 9

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'
IF 宗教 = '無' THEN '啤酒'
IF 職業 = '以勞力為主之員工' THEN '啤酒'
IF 血型 = 'O' THEN '啤酒'
IF 性別 = '男' AND 職業 = '負責人或部門主管' THEN '啤酒'
IF 性別 = '男' AND 職業 = '以腦力為主之員工' THEN '啤酒'
IF 血型 = 'B' THEN '葡萄酒'
Default rule: 高粱酒

Accuracy rate on the training set: 86.11111111111111 %
Accuracy rate on the test set: 85.0 %

Time taken: 5.093 s.

---------- Cross Validation #4----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 8

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'
IF 宗教 = '無' THEN '啤酒'
IF 職業 = '以勞力為主之員工' THEN '啤酒'
IF 教育程度 = '專科或大學' THEN '啤酒'
IF 教育程度 = '國中或高中職' AND 宗教 = '道教' THEN '啤酒'
IF 性別 = '男' AND 教育程度 = '國中或高中職' THEN '啤酒'
Default rule: 高粱酒

Accuracy rate on the training set: 87.03703703703704 %
Accuracy rate on the test set: 80.0 %

Time taken: 4.875 s.

---------- Cross Validation #5----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 8

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'
IF 宗教 = '無' THEN '啤酒'
IF 職業 = '以勞力為主之員工' THEN '啤酒'
IF 教育程度 = '專科或大學' THEN '啤酒'
IF 教育程度 = '國中或高中職' AND 宗教 = '道教' THEN '啤酒'
IF 性別 = '男' AND 教育程度 = '國中或高中職' THEN '啤酒'
Default rule: 高粱酒

Accuracy rate on the training set: 87.03703703703704 %
Accuracy rate on the test set: 80.0 %

Time taken: 5.907 s.

---------- Cross Validation #6----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 8

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'
IF 宗教 = '無' THEN '啤酒'
IF 職業 = '以勞力為主之員工' THEN '啤酒'
IF 教育程度 = '專科或大學' THEN '啤酒'
IF 教育程度 = '國中或高中職' AND 宗教 = '道教' THEN '啤酒'
IF 性別 = '男' AND 教育程度 = '國中或高中職' THEN '啤酒'
Default rule: 高粱酒

Accuracy rate on the training set: 86.11111111111111 %
Accuracy rate on the test set: 88.33333333333333 %

Time taken: 5.156 s.

---------- Cross Validation #7----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 6

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '男' THEN '啤酒'
IF 婚姻狀況 = '未婚' THEN '啤酒'

IF 教育程度 = '國中或高中職' AND 婚姻狀況 = '已婚' THEN '啤酒'
IF 婚姻狀況 = '已婚' THEN '葡萄酒'
Default rule: 高粱酒

Accuracy rate on the training set: 86.29629629629629 %
Accuracy rate on the test set: 81.66666666666667 %

Time taken: 3.109 s.

---------- Cross Validation #8----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 9

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'
IF 教育程度 = '專科或大學' THEN '啤酒'
IF 性別 = '男' AND 年齡 = '15-19 歲' THEN '葡萄酒'
IF 職業 = '以勞力為主之員工' THEN '啤酒'
IF 性別 = '男' AND 職業 = '負責人或部門主管' THEN '啤酒'
IF 職業 = '以腦力為主之員工' THEN '啤酒'
IF 教育程度 = '國中或高中職' THEN '啤酒'
Default rule: 高粱酒

Accuracy rate on the training set: 85.37037037037038 %
Accuracy rate on the test set: 93.33333333333333 %

Time taken: 5.141 s.

---------- Cross Validation #9----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 7

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '男' AND 婚姻狀況 = '已婚' THEN '啤酒'
IF 婚姻狀況 = '未婚' THEN '啤酒'
IF 婚姻狀況 = '已婚' AND 宗教 = '佛教' THEN '葡萄酒'
IF 血型 = 'A' THEN '啤酒'
IF 宗教 = '道教' THEN '葡萄酒'
Default rule: 啤酒

Accuracy rate on the training set: 87.03703703703704 %
Accuracy rate on the test set: 81.66666666666667 %

Time taken: 3.906 s.

---------- Cross Validation #10----------

Cases in the training set: 540

Cases in the test set: 60

Rules: 8

IF 是否喜歡喝酒 = '不喜歡' THEN '無'
IF 性別 = '女' AND 婚姻狀況 = '已婚' THEN '葡萄酒'
IF 宗教 = '無' THEN '啤酒'
IF 職業 = '以勞力為主之員工' THEN '啤酒'
IF 教育程度 = '專科或大學' THEN '啤酒'
IF 教育程度 = '國中或高中職' AND 宗教 = '道教' THEN '啤酒'
IF 性別 = '男' AND 教育程度 = '國中或高中職' THEN '啤酒'
Default rule: 高粱酒

Accuracy rate on the training set: 85.55555555555556 %
Accuracy rate on the test set: 93.33333333333333 %

Time taken: 4.984 s.

10-Fold Cross Validation Results

Accuracy Rate on Test Set	Rules Number	Conditions Number
85.17% +/- 1.58%	8.1 +/- 0.31	10.2 +/- 0.68

Total elapsed time: 48 s.