一種改良的啟發式方法以建構名目屬性之二元決策樹

葉榮懋
南台科技大學管理與資訊系

施武榮
南台科技大學管理與資訊系

徐芳玲

摘要

資訊科技的日新月異，資料的儲存與處理規模均與過去有相當大的差距。如何從龐大的資料量中擷取出有用的資訊以提供給決策者參考，一直是資料探勘領域裡所關注的重點。決策樹由於其運算容易，又能產生清楚的規則，使其成為資料探勘中最常用的分類技術之一。但是當處理的資料量龐大，且名目屬性的屬性值相當多的情況之下，若每一屬性值都形成一個分支，則決策樹的分支太多將會造成所萃取的規則過於複雜難以解讀，資料在處理上的效率也會大打折扣。

本論文發展一種簡化決策樹的方法，可將資料庫內的名目屬性做二元分割，把資料分成二支，以減少過多與不必要的決策樹分支。本研究採用主成分分析法中，可表示大部分變異的第一主成分，並利用該成分裡經過標準化成分分數的平均值，作為二元分割屬性值的基準，以消除過多的屬性值分支，使得決策樹的外顯知識容易解讀。最後，並以四個UCI資料庫內的資料集作為測試樣本，結果顯示本研究所提的方法，在決策樹的精簡與分類正確性上都有良好的表現。

關鍵字：決策樹、資料探勘、分類、啟發式方法、主成分分析

A Modified Heuristic Method to Construct the Binary Decision Tree of Nominal Attributes

Jong-Mau Yeh
Department of Management and Information Technology, Southern Taiwan University

Wurong Shih
Department of Management and Information Technology, Southern Taiwan University

Fang Ling Shyu

Abstract

The ability to extract useful information from a large-scale database to aid decision-making is critical in data mining. Classification is an important problem in data mining. It has been studied extensively as a possible solution to the knowledge acquisition. Decision tree has become one of the most commonly used techniques for classifying data because the algorithm for generating a decision tree can be easily implemented. However, when there are too many distinct values of the nominal attributes in each node of a tree, the branches of the tree become enormous and complicated. As a result, the effectiveness of data processing in a large data set may be compromised. This paper aims to propose a heuristic method to simplify the decision tree by splitting the nominal attributes into two branches. We adopt principal component analysis to present an algorithm for finding a good partition strategy in order to reduce unnecessary branches of a decision tree. Since the principal component can represent most of the variants, the first component scores of each attribute will be utilized as the thresholds for splitting examples. The decision tree can be simplified to a binary tree so that the explicit knowledge of a tree can be easily extracted. We also compare against other heuristic methods and give an analysis of experimental results on four UCI data sets.

Key words : Decision tree, Data mining, Classification, Heuristic method, Principal component analysis

壹、前言

電子商務的快速步調，使得企業須不斷地累積有關於顧客、供應商、產品與服務的龐大資料，針對這些龐大的資料，資料儲存量甚至以GIGA、TERA等單位來計量。雖然這些資料庫中包含了潛在有價值的資訊，但是卻難以分析隱藏在其中具有重要意義的資訊。資料探勘提供企業組織從這些可觀的儲存資料中找尋出趨勢、模式與關聯的工具，以指引他們完成公司或組織的重要策略性決策。

在這些資料庫的紀錄中，有一個被指定的屬性稱為『因變屬性』(Dependent attribute)，其他的被稱為『預測屬性』(Predictor attribute)。而一般分類方法的目標為：建立一個模式以『預測屬性』做為輸入，然後輸出一個對應『因變屬性』的值。當因變屬性為數值型態時，這類型的問題通常被稱為『迴歸問題』；其他非數值的型態則被歸類為『分類問題』。分類模式屬於資料探勘的研究領域之一，近年來學者們已經提出許多分類模式：類神經網路 (Neural networks)、遺傳演算法 (Genetic algorithms)、貝氏方法 (Bayesian methods) ，及決策樹(Decision trees)等方法(Han & Kamber 2001)。其中決策樹以直覺的表示方式，使得分類模式的結果容易被解釋與理解，此種形式相對於其他複雜的方法而言較具吸引力。此外，在建立決策樹時，不需要分析者輸入任何的參數，在利用決策樹萃取知識時，亦不須由相關領域的專家才能取得；這些優點使得決策樹成為實際應用上最頻繁的技術之一(Murthy 1998)。但由於大型資料庫使用上的普及，造成許多知識發掘技術的執行時間複雜度與分類結果的簡潔度都隨資料的數量而遽增。以決策樹方法而言，若所欲表現的概念趨於複雜或是枝節過多時，訓練得出的知識，也隨著法則數的增加而變得更加難以解讀。若以銷售地區為例，其屬性值很多，如果將所有縣市所代表之屬性值，皆毫無遺漏的自決策樹分支延伸出去，則單一個節點就會擁有高達數十個分支，每個分支後可再延伸出更形複雜的枝節；這種龐大的決策樹架構，往往造成決策樹外顯知識難以理解，而失去其原有之優勢。

本文專注在決策樹建構方法上，改良Coppersmith et al. (1999)的啟發式方法以縮減決策樹法則數，希望藉由屬性值二元化之方式，將多元分支縮減成二元分支方式表示。本文中所提出的方法為轉換資料集的空間，意圖以資料本身的統計量作為屬性值分割的依據，利用主成分分析將原有多屬性值與多分枝的決策樹，簡化成二元樹的型態，以期能減少最後所得到之決策樹的節點數目與法則規模。此種作法對於知識精簡與解讀應有相當之助益，而應用本方法到實例問題測試，在分類之正確性上亦有良好的表現。

貳、相關研究

建立最佳樹的某些技術層面是相當困難的(Laber & Nogueira 2004)。學者經由不同的衡量目標與條件下證明由資料集內建立最佳決策樹的難度為NP-complete(Hyafil & Rivest 1976; Cox et al. 1989; Murphy & McCraw 1991; Naumov 1991) 。Heath(1993) 證明若給予兩

個互斥的資料集以求得最小化分類錯誤數量的分割，屬於NP-complete。但是更一般的連續型或多變量決策樹的問題，則被證明為屬於NP-hard (Hoffgen et al. 1995)。由於多變量決策樹在運算上的複雜，使得許多學者專注研究在減少決策樹大小規模，或是減少資料向度，以減低運算的困難度。

為了因應決策樹應用在大型資料庫時所產生的效率與複雜性劇增的問題，目前已經發展出相當多簡化決策樹的方法。這些方法可分成五個主要種類(Aha & Breslow 1998)。

1. 控制樹的大小規模 (Controlling tree size)：此種做法由修剪演算法組成，主要是以控制樹的大小規模來達到簡化的目的。包括前置修剪(Pre - pruners) (Auer et al. 1995)與後置修剪(Post - pruners) (Esposito et al. 1997) 。
2. 修正測試空間 (Modifying the Space of Index Tests)：測試每一個節點某個特徵是否等於(或不等於)某個值，經由單變量(亦可多變量)的測試中將實例分群。包括假設導向(Hypothesis - driven) (Pagallo & Haussler 1990)與資料導向(Data – driven) (Brodley & Utgoff 1995; Zheng 1995)。
3. 修正測試搜索 (Modifying the search for index tests)：此方法包括幾種子類別組成：使用替代函數評估分割、分割連續型資料、或是自動化前向搜尋等。主要的有離散化(Discretization) (Quinlan 1996)與前向搜尋(Lookahead search) (Ragavan & Rendell 1993)。
4. 減少資料集的大小 (Reduce data set size)：此類別利用遺傳演算法等搜尋特徵子集合之空間以減少測試資料的方式，達到使決策樹大小規模縮減的目的。包括特徵選擇(Feature selection) (Cherkauer & Shavlik 1996; Terano & Ishino 1996; Sorensen & Janssens 2003; Carvalho & Freitas 2004)與個案選擇 (Case selection) (John 1995)。
5. 替換資料架構 (Alternative data structures)：此類別的演算法將決策樹轉換成其他替代的資料結構。替代的方式包括法則集合 (Rule sets) 與圖表法 (Graphs) (Quinlan 1993)。

雖然簡化決策樹有上述的方法，但一般在建構決策樹時，經常遇到分支決策樹的問題。因為許多決策樹在建立程序時，都依照一個共同的特徵：如果一個有k個值的名目屬性被選為節點中區分範例的鑑別者 (Discriminator)，那麼就會有 EMBED Equation.3 個對應於該屬性值的子樹產生，而當k大於2的時候，則常常會有不必要的樣本空間分割發生。此種k分支決策樹的主要缺陷為樣本空間的支離破碎，不僅導致歸納運作上產生執行效率的障礙，樹的支離破碎亦導致概念複雜化與不相關的屬性產生。再者，決策樹的建立方法係依據某些衡量值 (Goodness measure) 選擇最佳的屬性作為分支節點，其中一個常見的衡量值為資訊獲得量 (Information gain)(Quinlan 1986)。當屬性在預估分類值上準確度高時，這些屬性的衡量值也高。然而，在同一屬性中往往有某些屬性值的預測準確性高，而有些屬性值預測準確性卻很差。如果資料集內的屬性，應用在決策樹上的都是好的預測者，那麼樣本空間被分割成支離破碎的情況就不太會發生。但在真實世界的應用中，大部分的屬性都同時兼具有較佳與較差的預測者。以一個社會科學的資料集為例，想藉由資料中去預測收入的高低。在年齡這屬性上，直覺地我們可以了解在20歲以下或65歲以上的族群，較不可能賺取高收入，此時年齡是不錯的預測者；但是20歲以上到65歲之間的族群，其收入可能是低收入，亦有可能是高收入，這使得年齡成為一個較

弱的預測者。在此資料集內，如果將年齡分成五個區間：＜20，20－30，30－40，40－50與＞50。我們可以得知 ＜20的區間預測準確度佳，而其他區間的預測準確度較弱，因此並沒有必要將年齡在決策樹區分為多個分支。

針對此*k*分支決策樹的問題，Quinlan(1993)提出C4.5將連續的數值屬性二元分割後用來建立決策樹，Quinlan(1996)並以實證的結果來比較二元分支與 EMBED Equation.3 分支的績效。他們的研究結果建議：二元決策樹應用在實際狀況中，在預測分類值上相當有效率。此種二元決策樹所產生的分支較少，且它們的預測準確率與*k*分支決策樹近似。

上述連續的數值屬性雖可以二元分割，但名目屬性(nominal attributes)的資料卻不像連續屬性無法直接排序或比較大小。針對名目屬性的二元分割，Mehta et al. (1996)提出 SLIQ方法，此法在屬性值的數目較小時(*k*)，在所有可能二元分割的方式中(共有2^{n-1}-1種)，逐一搜尋最佳的組合；當屬性值較大時則使用貪婪(greedy)演算法，首先訂出二個集合，前者集合為空集合，後者集合包括所有的屬性值項目。在執行演算法時，反覆從後者集合挑選出一個項目移到前者，直到不能改善為止。

Coppersmith et al. (1999) 證明將資料轉軸，可在主成分軸上存在最佳的L分割之分割點，提出一啟發式二元分割的啟發式方法，並驗證彼等方法的執行績效比SLIQ法較佳。此方法係將資料轉換空間，求算其在第一主成分的成分分數。接著，將此主成分分數排序，而後採用類似Quinlan(1993) 連續屬性二元分割的方式，一一測試每一分數，尋找那一點為最佳的分割點，最後將此排序後的主成分分數分成兩堆。此種方法的缺點是(1)主成分分數需經排序(2)並無法在一次就能有效的找到分割點，若該名目屬性有n種屬性值，則有n-1種二元分割的方式，需逐一執行n-1次測試。如此n個主成分分數之排序與n-1次分割點的測試，將會耗費相當大的運算時間。針對上述的缺點，本研究主要在探討名目屬性是否可以適用於二元決策樹之建構方式，並提出改良的啟發式二元分割方式期能達到簡化決策樹，且又同樣保有相當之正確性。

參、研究方法

一、主成分分析

在科學研究中，我們常碰到必須處理一些彼此可能有相關的變項存在之情境。如何將這許多變項予以減少，並使其改變為較少個互相獨立的線性組合變項，是很迫切需要的工作。如果我們只利用少數的潛在變量或成分便能有效代表許多彼此有關的變項之結構，是非常經濟且有效率的。在這種情形下，我們可以使用主成分分析來達到這種目的；另一方面，利用主成分分析，我們也可以知道如何將少數幾個變項予以線性組合，使經由線性組合而得成分的變異數為最大，使資料在這些成分方面顯出最大的個別差異來。

假使我們就*N*個受試者的*p*個隨機變項$X_1,X_2,\ldots,X_P$加以觀察，得到下列矩陣*X*所示的*N*×*p*個觀察分數：

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \dots & \dots & & \dots \\ x_{N1} & x_{N2} & \dots & x_{Np} \end{bmatrix}$$

主成分分析所需的資料是樣本的變異數－共變數矩陣(variance-covariance matrix)S，或是相關係數矩陣(correlation matrix)R。以變異數－共變數矩陣S為例，將第一主成分表示為觀察分數的線性組合，

$$Y_1 = k_{11}X_1 + k_{21}X_2 + \dots + k_{p1}X_p \quad (1)$$

此線性組合Y1之變異數為：

$$s_{Y_1}^2 = k_1' S k_1 \quad (2)$$

此處k_1'為k_1的轉置向量，欲使得該變異數為最大，需先求得其特徵向量k_1，與對應之最大的特徵根λ_1。當原有之各 X 變項，在乘上特徵向量 k1 的元素之後，所得的分數之變異為最大，我們稱此特徵向量 k_1的元素為第一主成分的主成分係數。

在本論文中，我們採用主成分分析應用於決策樹屬性值之二元分割，目的在於利用其可取用較少之主成分即可解釋較多個向度的空間，且該主成分之成分分數間有最大的變異數，因此可由解釋變異最多之主成分來評估該主成分分數與最終決策點之關係，我們利用此性質作為二元分割之依據。

二、改良的啟發式二元分割方法

當名目屬性之屬性值很多時，若以多分支分割，因為分支數太多勢必造成樹的規模過大。為了減少樹的規模，可採用二元分割。例如：屬性A有n個屬性值，則共有2^{n-1}-1種二元分割的組合方式，Coppersmith et al. (1999)所提出的啟發式方式，將資料轉換空間，在主成分軸上以主成分分數的高低排序後，再一一測試那一點為最佳的分割點，因此二元分割的方式已從2^{n-1}-1種減少為n-1種。此方法是近年來針對名目屬性之二元分割，較佳的方法之一。但是此法無法有效判定二元分割的方式，需要將所有的點一一測試，造成處理績效不佳。本研究所提的啟發式方式亦採用主成分分析之方法，希望能減少其分割點測試的次數，在我們的方法中主成分分數不需排序，直接以主成分分數的平均值，做為二元分割的依據，如此二元分割的方式只剩一種，可大幅縮減二元決策樹的建構時間。

在Coppersmith et al. (1999) 所提的啟發式方法中，為了調整資料出現頻度之不一，先利用每個屬性的頻度矩陣(contingency matrix)，求出個別屬性值為準的類別機率矩陣

(class probability matrix)。再以出現的頻度為修正求出加權共變異矩陣(weighted covariance matrix)，接著利用此共變異矩陣，求算其在第一主成分的成分分數。本研究不求出加權共變異矩陣，而是先將資料標準化(standardization)後，再以相關矩陣作主成分分析。由於標準化的資料，其主成分分數的平均值為0，故以成分分數0做為二元分割的依據， 如此分割點的決定只有一次，其二元分割點的設定的方法如下所示。

在訓練資料集中，假設我們所觀測到的某個屬性的值分別對應於類別之出現頻度矩陣表示如下：

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \cdots & \cdots & & \cdots \\ x_{N1} & x_{N2} & \cdots & x_{Np} \end{bmatrix}$$

其中此矩陣中共有N個屬性值與p個類別， x_{ij}為屬性值i對應於類別j之出現頻度。先將原矩陣之x值轉化為z分數，以得到標準化之矩陣Z。

$$Z=\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1p} \\ z_{21} & z_{22} & \cdots & z_{2p} \\ \cdots & \cdots & & \cdots \\ z_{N1} & z_{N2} & \cdots & z_{Np} \end{bmatrix}$$

$$\text{其中 } z_{ij}=(x_{ij}-\overline{X_j})/S_j$$

$\overline{X_j}$：類別 j 的平均值

S_j：類別 j 的標準差

再以矩陣X的相關矩陣R，求出解釋變異最大的特徵根λ_1，與對應的特徵向量k1。然後以第一主成分的特徵向量k_1，求出第一主成分所對應的成分分數向量Y_1。

$$Y_1=Z\cdot k_1=\begin{bmatrix} \sum_{i=1}^{p} z_{1i}k_{i1} \\ \sum_{i=1}^{p} z_{2i}k_{i1} \\ \cdots \\ \sum_{i=1}^{p} z_{Ni}k_{i1} \end{bmatrix} \tag{3}$$

計算經過標準化處理後之主成分分數向量Y_1後，我們以此主成分分數的平均數（即為0）做為分割之門檻。主要的涵義在於表示該分數在該主成分中，表現在平均以上，或在平均以下，以此將屬性值分割成兩個子集合，主成分分數比0小的屬性值為一個分枝，比0大的屬性值為另一個分枝。

三、計算例

某資料集Lymphography中有148筆資料，可分為四個類別：normal find(N), metastases(M), malign lymph(L), fibrosis(F)；若以屬性Lymphatics 為例，此屬性共有四種屬性值：arched(A), normal(N), deformed(D), displaced(S)。首先計算屬性值對應於類別之頻度矩陣為：

$$X = \begin{array}{c} \\ A \\ N \\ D \\ S \end{array} \begin{array}{c} \begin{array}{cccc} N & M & L & F \end{array} \\ \begin{bmatrix} 0 & 39 & 28 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 26 & 16 & 4 \\ 0 & 16 & 17 & 0 \end{bmatrix} \end{array}$$

接著再將矩陣X化成標準化矩陣Z。

$\overline{X}$ = [0.50 20.25 15.25 1.00]，S = [1.00 16.46 11.53 2.00]

$$Z = \begin{bmatrix} -0.5 & 1.14 & 1.11 & -0.5 \\ 1.5 & -1.23 & -1.32 & -0.5 \\ -0.5 & 0.35 & 0.07 & 1.5 \\ -0.5 & -0.25 & 0.15 & -0.5 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & -0.82 & -0.88 & -0.33 \\ -0.82 & 1 & 0.96 & 0.23 \\ -0.88 & 0.96 & 1 & 0.04 \\ -0.33 & 0.23 & 0.04 & 1 \end{bmatrix}$$

由上述矩陣X的相關矩陣R，可求得4個特徵根如下：2.8401，0.9848，0.1752, 0.0000。因此我們採取解釋變異最大的特徵根2.8401（解釋變異部分佔總變異百分比為71%），與其所對應的特徵向量k1以進行下一步驟。

$$k_1 = \begin{bmatrix} -0.5702 \\ 0.5612 \\ -0.5699 \\ -0.1873 \end{bmatrix}$$

接著再求經過標準化處理後之第一主成分分數

$$Y_1 = Z \cdot k_1 = \begin{bmatrix} -1.4668 \\ 2.3909 \\ -0.7978 \\ -0.1262 \end{bmatrix}$$

由前一步驟的Y_1向量中，可知主成分分數比0大的屬性值有{N}，而比0小的屬性值有{A, D, S} ，故可將屬性Lymphatics分為 {N} 與 {A, D, S} 兩個分枝，以繼續往後之步驟。

四、不純度評估函數

當用在分類或是歸納學習上時，決策樹實質上像是一個機率評估機制。此評估準則為啟發式方法，其目標為由測試資料中產生可靠的不純度 (Impurity) 評估值，較常用的不純度評估準則有兩類(Takimoto & Maruoka 2003)：熵函數 (Shannon’s entropy)(Quinlan 1986; Quinlan 1993; Ruggieri 2002)用在ID3及C4.5上與吉尼係數(Gini index)(Breiman et al. 1984)用在CART。

1. 由資訊理論衍生的準則：熵函數 (Shannon’s entropy)
   此種方法評估的性質具有對稱性、擴展性、決定性、可加性與遞迴性。熵函數 EMBED Equation.3 的數學式表示如下：

$$h(S) = -\sum_{i=1}^{k} \frac{freq(C_i, S)}{|S|} \cdot \log_2 \frac{freq(C_i, S)}{|S|} \tag{4}$$

   此處$freq(C_i,S)$：在S集合內出現C_i類別的次數
   |S|：在S集合內的樣本數
   k：類別總個數

2. 由距離衡量衍生的準則：吉尼係數 (Gini index)
   距離指的是類別機率分配間的距離。此種方法評估的性質具有類別間的分隔性、離散性與區別性。吉尼係數$g(S)$的數學式表示如下：

$$g(S) = 1 - \sum_{i=1}^{k} \frac{freq(C_i, S)^2}{|S|^2} \tag{5}$$

若在集合T內，以屬性A的屬性值分割成n個分枝，而形成子集合$\{T_1, T_2, ..., T_n\}$，則

熵函數的加權總和為：$$H_A(T) = -\sum_{j=1}^{n} \frac{|T_j|}{|T|} h(T_j) \tag{6}$$

吉尼係數的加權總和為：$$G_A(T) = -\sum_{j=1}^{n} \frac{|T_j|}{|T|} g(T_j) \tag{7}$$

Quinlan(1986)在ID3 演算法中，採用熵函數作為分枝的根據，將不純度減少量定義成資訊獲得量，分枝的方法則是找出能獲得最大資訊之屬性作為節點之分枝標準。Quinlan(1993)又提出C4.5修改了ID3歸納學習法裡的屬性選擇法，對資訊獲得量再以屬性的資訊量做正規化(normalization)的處理，稱為獲得量比值(Gain Ratio)的屬性選擇法。通常屬性值較多的屬性（亦即分支數較多），資訊獲得量較高；但相同地，其屬性本身的資訊量也比較高，所以正規化的動作可以減少ID3在這方面的缺點。本研究採用上述一般

常用的熵函數與吉尼係數兩種不純度評估函數，以最大資訊獲得量之屬性作為節點之分枝標準。由於本研究採二元分割的方式，每個屬性的分支數目相同，故做正規化的影響較小。

五、決策樹建立步驟與流程

一般決策樹的演算法由兩個階段組成：樹的建立 (building) 與修剪 (pruning)。在建立樹的階段中，有四種方式：

1. 由下而上建立法 (Bottom - Up approaches)：主要是衡量類別間的距離，每一步驟皆將距離較小之類別合併，並形成另一個新的群組，同時求出該組之平均向量及共變異矩陣，過程一直重複至根節點為止。
2. 由上而下建立法 (Top - Down approaches)：此方法的主要工作有選定節點分枝規則、判斷分枝工作何時停止、指派每一最終節點枝類別。
3. 混合法 (Hybrid approaches)：合併上面兩種方法。
4. 成長-修剪法 (Growing - Pruning approaches)：樹的成長部分類似由上而下的建立方法，盡量將樹分枝為同質或近似同質分枝，之後再決定修剪規則以修剪成長完成的樹。

(Breiman et al. 1984; Quinlan 1986; Esposito et al. 1997)

本文採用由上到下建立的決策樹演算法，由根節點開始，經由使用不純度評估函數，計算最佳分割點；然後重複此方式將資料庫分割，直到滿足停止條件為止，演算法如下所示：

步驟1. 找出未處理過的屬性，計算其屬性值對應於類別所形成之頻度矩陣。
隨意選取一個未處理的屬性，將屬性中的各個屬性值對應於類別種類（或決策）的出現次數表示，並形成一頻度矩陣。

步驟2. 求出頻度矩陣之相關矩陣的特徵根及特徵向量。
求得特徵根與特徵向量之後，解釋變異最大的特徵根所對應的特徵向量即為第一主成分之成分係數。

步驟3. 求得第一主成分所對應的成分分數。
原頻度矩陣經過標準化處理後（減去平均數，再除以標準差），再乘上成分係數，即可求得其對應的成分分數。

步驟4. 該成分分數以0為分割點將屬性值分成兩個分枝。
將每個名目屬性值依照其所對應之成分分數，分割成兩群。

步驟5. 比較每個屬性二元分割的不純度，以資訊獲得量最大的屬性作為下一個分支的節點。

步驟6. 如果所有屬性都已分割或已達近似同質分枝則結束；否則回到步驟1.進行下一屬性的選取。

六、運算過程的比較

本文所提的演算法係Coppersmith et al. (1999)方法的修正，今利用彼等論文中的例

子，說明演算過程的差異。

在507筆的水果資料集，可分成Pick-and-Ship(PS), No-Pick(NP), 及Pick-and-Ripen(PR)三個類別；在顏色的屬性中，共有6種顏色的屬性值，分別為 {Green(G), Blue(B), Purple(P), Orange(O), Yellow(Y), Red(R)}，其屬性值對應於類別之頻度矩陣為：

$$X = \begin{array}{c} \\ G \\ B \\ P \\ O \\ Y \\ R \end{array} \begin{array}{c} \begin{array}{ccc} PS & NP & PR \end{array} \\ \begin{bmatrix} 20 & 3 & 1 \\ 19 & 4 & 2 \\ 25 & 4 & 7 \\ 80 & 5 & 25 \\ 100 & 2 & 3 \\ 200 & 3 & 4 \end{bmatrix} \end{array}$$

其加權共變異矩陣為：

$$\Sigma = \begin{bmatrix} 6.296 & -1.633 & -4.663 \\ -1.633 & 0.897 & 0.735 \\ -4.663 & 0.735 & 3.928 \end{bmatrix}$$

利用此加權共變異矩陣Σ，求得特徵根為 7.486, 0.867, 0 。今以最大特徵根 7.846 所對應的特徵向量為 [0.771, -0.183, -0.597]。因此，若以原始6種顏色的順序 {G, B, P, O, Y, R}，可求得第一主成分分數為 0.603, 0.516, 0.406, 0.424, 0.723, 0.740，今再以主成分分數由小至大排序，對應顏色的結果為 {P, O, B, G, Y, R}。接著在此顏色的排序中，於不同顏色的交界處 (P與O), (O與B), (B與G), (G與Y), (Y與R)，一一計算不純度，選取最佳的二元分割點。經過此步驟，可得知若以吉尼係數為不純度評估指標，則{P, O, B}為一分支，{G, Y, R}為另一分支；若以熵函數為評估指標，則{P, O, B, G}為一分支，而{Y, R}為另一分支。可看出以不同的不純度評估指標，可能得到不同的結果。

若以本文所提的改良方法，將矩陣X化成標準化矩陣Z

$$Z = \begin{bmatrix} -0.765 & -0.477 & -0.663 \\ -0.779 & 0.477 & -0.552 \\ -0.694 & 0.477 & 0.000 \\ 0.085 & 1.430 & 1.988 \\ 0.368 & -1.430 & -0.442 \\ 1.784 & 0.477 & -0.331 \end{bmatrix}$$

對應於原始6種顏色的順序 {G, B, P, O, Y, R}，可求得第一主成分分數為 0.034, -0.052, -0.162, -0.144, 0.155, 0.172，今以 0為門檻值直接做二元分割，可得到比 0小的屬

性值分支有{B, P, O}，比0大的屬性值分支有{G, Y, R}，此結果與Coppersmith et al. (1999)利用吉尼係數為不純度評估指標的結果相同。因此本文所提出改良的啟發式方法，只使用一簡單的門檻值，不必一一搜尋最佳的二元分割點，方法很簡便。

再者，利用吉尼係數或熵函數等不同的不純度評估指標，可能造成不同的二元分割點，雖然分割後的集合很接近，但最後產生的整體二元樹會有所不同。Coppersmith等的方法中，在每個節點需一一搜尋最佳的分割點，然而每個節點不純度的最佳化，並不能意味著全體樹的最佳化，況且實務應用上評估二元樹品質的指標很多，為了驗證本修正演算法的績效，在下一部分說明本文與Coppersmith等的方法執行績效的比較。

肆、演算法之驗證

為了驗證本研究提出之名目屬性二元分割（簡稱NML法）之模式效果，本研究比較ID3 (Quinlan 1986)之多元分割方式，及Coppersmith et al. (1999)之二元分割方式（簡稱CS法），分析所獲致之結果。

一、實驗資料

本研究採用的資料集主要來源為UCI（University of California Irvine，專門提供測試學習效能的資料庫）。在UCI眾多資料庫中，我們為了因應實驗需要所挑選出來的資料集必須滿足下列特性：

採用資料類型為以名目屬性為主，或將數值屬性預先分割成數個區間之資料集。

單一屬性中含括較多的屬性值。採用二元分割應用之主因是為了簡化過多屬性值所造成的決策樹分枝繁雜與法則過多，使訓練結果難以解釋，因此測試資料集中應包含較多屬性值。

大型資料庫。由於此種二元分割方式主要用於改善大型資料庫中過多分枝與法則的狀況，因此本研究中亦需挑選至少一個以上資料筆數較多之資料庫作為實際執行之案例。

缺失值比例小。若缺失值在整理資料庫中所佔比例過高，不但造成資料前置處理上之困難，訓練所得之結果，也受到缺失值影響極大，故缺失值比例不宜過高。

經由上述原則，選出UCI中四個不同領域的資料集來進行模式的驗證，表1為本實驗的資料集總表。

表1：資料庫總表

資料庫名稱	領域	資料數	類別數	屬性數
Dermatology Database	醫學	366	6	34
Lymphography Database	醫學	148	4	18
Mushrooms Database	生物	8124	2	21
The Company Insurance Benchmark	行銷	9822	2	86

（一）皮膚醫學資料庫 (Dermatology Database)

資料庫為皮膚醫學上erythemato-squamous疾病診斷的差異真實案例。由於erythemato-squamous為一疾病群之統稱，該疾病群的診斷必須藉由病理切片觀察，且在疾病初期有某些特徵相近，使該疾病群在診斷上有其困難度。該資料庫欲藉由臨床診斷與病理切片的34項特徵來區別該疾病群之種類。該資料庫包含：6個類別，366個例示。共有34個屬性，屬性值除了家族病例{0, 1}與年齡{0-10, 10-20, …, 90-100}外，值域皆為{0, 1, 2, 3}的評量值。

（二）霍奇金氏病資料庫(Lymphography Database)

資料庫之目標在將資料集分成四個類別。該資料庫包含：4個類別，148個例示。共18個屬性，有2個屬性之值為{1, 2, 3, …, 8}，有4個屬性的屬性值域為{1, 2, 3, 4}，有1個屬性其值為{1, 2, 3}，其餘屬性皆為{0, 1}。

（三）蕈類資料庫(Mushrooms Database)

資料主要為蕈類的外表特徵類型的描述，藉以將其區分為有毒或可食兩種類別，所有屬性皆為名目屬性。該資料庫為大型資料庫，包含8124個例示，21個屬性，有超過11個屬性的名目屬性個數大於5個。

（四）保險公司標竿資料庫 (The Company Insurance Benchmark)

資料來自荷蘭資料探勘公司，係實際的世界性商務問題。該資料集之目的主要在於從顧客的相關資料，來預測顧客是否對商務保險證書有興趣。顧客的資訊有86個屬性包括產品使用資料與社會統計資料等。此資料庫分成訓練集與測試集，訓練集內有5822筆顧客的描述，顯示他們是否已經擁有商務保險證書，而測試集內有4000個顧客的資料。表2列出這86個屬性中其屬性值數的分配表，由表中可看出其中有一個屬性，其屬性值高達41個，在訓練決策樹時，此種屬性值極多的狀況，會造成決策樹之分枝與法則過多與繁雜。

表2：資料集內屬性值個數列表

屬性值數	2	3	4	5	6	7	8	9	10	13	41
屬性	6	5	2	3	8	8	7	2	40	1	1

（五）資料前置處理

在訓練與測試時，針對缺失值 (Missing Value)與數值型態資料 (Numeric Data)需預先考量與處理。

1. 一般針對缺失值的處理方式：

(1)忽略含缺失值之資料。

(2)以平均數或眾數取代。

(3)其他更進一步之方法。

本研究採用第(1)種方式處理。由於本研究之主要目的不在於探討缺失值之處理，故

僅採用最簡單之方式，並在挑選資料庫時已選擇缺失值比例較少之資料集，以節省前置處理所花費之時間。

2. 一般針對數值型態資料的處理方式：

(1)直接把數值資料當作名目型態使用。

(2)自行將數值分區，編成順序型資料。

(3)忽略數值型態資料。

本研究中採用第(2)種方法，預先將所得到之數值資料區分成數個區間，以名目屬性方式處理。以年齡來說，以0到100分成十等分，然後將連續屬性分別歸入相對應的區間後使用。

（六）資料的抽樣方法

本研究採用K份交叉驗證法(K-folds Cross validation)，將資料集分成k等分，以其中一等分作為測試資料集，其餘k-1份作為訓練資料集，將此實驗重複k次。

二、評估決策樹品質的指標

本文採用分類複雜度、分類正確性與分類之效率等三個部分(Bohanec & Bratko 1994; Murthy 1998; Lim et al. 2000; Osei-Bryson 2004)來檢驗決策樹的品質。

（一）分類複雜度

1. 樹的規模：樹自根節點所延伸出去的節點數量。
2. 產生的法則數：決策樹所對應的法則數。

（二）分類正確性

1. 訓練錯誤率：應用在訓練範例的錯誤率。
2. 測試錯誤率：應用在測試資料集的錯誤率。

（三）分類之效率

1. 最大深度：從根節點到最遠節點的距離。
2. 平均深度：從根節點到其他節點的平均距離。

三、實驗結果

本實驗採用 Matlab 完成所有演算法的程式，程式執行平台所使用之中央處理器為 Intel Pentium 4 2.40 GHz。為了比較ID3法、NML法與CS法三種演算法的執行績效，上述四個資料集，設定以同質率99%為分枝終止條件，經由5份交叉驗證法(5-folds cross validation)所獲致的五組實驗數據的平均值如下所示。

（一）皮膚醫學資料庫 (Dermatology Database)

從表3與表4可得知，若僅就本研究的NML法與CS法此二種二元分割的方法而言，當不純度評估為熵函數時，以CS法整體表現較佳；當不純度評估為吉尼係數時，則以本研

究之NML法之整體表現較佳。若以本研究之NML法與ID3法相比較，可節省之節點數：在熵函數時為38.5%，在吉尼係數下可節省43.8%。NML法與CS法之二元分割決策樹之正確性表現比ID3法較高，理由或在於實際資料庫內含有雜訊(noisy)資料，此雜訊在二元分割遞迴的過程中，由於數量相對較小，因此可以減少雜訊對結果的干擾。在決策樹的建構時間上，若就NML法與CS法此二種二元分割的方法而言，不論不純度評估為熵函數或吉尼係數時，本研究之NML法比CS法建構速度較快，可節省44%與41%的建構時間。至於ID3法係建構k分支的決策樹並非建構二元決策樹，故其執行速度較快。

表3：Dermatology Database 訓練結果（熵函數）

方法	節點數	法則數	訓練錯誤率	測試錯誤率	最大深度	平均深度	建構時間(秒)
ID3	102.3	77.6	0.00%	21.23%	6.2	2.39	1.54
NML	63	32	0.00%	11.18%	11.6	4.07	3.90
CS	57.4	29.2	0.00%	10.05%	10.6	3.76	7.05

表4： Dermatology Database 訓練結果（吉尼係數）

方法	節點數	法則數	訓練錯誤率	測試錯誤率	最大深度	平均深度	建構時間(秒)
ID3	96.4	73.2	0.00%	21.21%	6.2	2.34	1.32
NML	54.2	27.6	0.00%	9.51%	10.2	3.65	4.03
CS	61.4	31.2	0.00%	9.49%	11.6	3.93	6.91

（二）霍奇金氏病資料庫 (Lymphography Database)

從表5與表6顯示，若僅就本研究的NML法與CS法此二種二元分割的方法而言，以CS法為分割點之方法表現較佳，但與本研究之NML法相差不多。若以本研究之NML法與ID3法相比較，可節省之節點數：在熵函數時為16.9%，在吉尼係數下可節省13.3%。NML法與CS法之二元分割決策樹之正確性表現亦比ID3法較高。在決策樹的建構時間上，若就NML法與CS法此二種二元分割的方法而言，當以熵函數為不純度評估時，可節省33%的建構時間；以吉尼係數為不純度評估時，可節省35%的建構時間。

表5：Lymphography Database 訓練結果（熵函數）

方法	節點數	法則數	訓練錯誤率	測試錯誤率	最大深度	平均深度	建構時間(秒)
ID3	73.4	45.8	0.00%	29.01%	5.8	1.84	0.37
NML	61	31	0.00%	27.03%	9.4	3.31	1.26
CS	59	30	0.00%	22.30%	8.4	3.09	1.87

表6：Lymphography Database 訓練結果（吉尼係數）

方法	節點數	法則數	訓練錯誤率	測試錯誤率	最大深度	平均深度	建構時間(秒)
ID3	73.6	45.6	0.00%	28.32%	5.8	1.92	0.35
NML	6.83	32.4	0.00%	25.03%	9.8	3.37	1.24
CS	61	31	0.00%	26.32%	9	3.14	1.93

（三）蕈類資料庫 (Mushrooms Database)

從表7與表8顯示，若僅就本研究的NML法與CS法此二種二元分割的方法而言，以本研究之NML法整體表現較佳。若以本研究之NML法與ID3法相比較，可節省之節點數：在熵函數時為51.5%，在吉尼係數下可節省49.0%。本研究之NML法在決策樹之正確性表現亦非常良好。在決策樹的建構時間上，若就NML法與CS法此二種二元分割的方法而言，當以熵函數為不純度評估時，可節省49%的建構時間；以吉尼係數為不純度評估時，可節省40%的建構時間。

表7：Mushrooms Database 訓練結果（熵函數）

方法	節點數	法則數	訓練錯誤率	測試錯誤率	最大深度	平均深度	建構時間(秒)
ID3	31	24	0.00%	0.00%	4	1.57	18.50
NML	13	7	0.00%	0.00%	6	2.36	46.70
CS	15	8	0.10%	0.10%	5.2	2.11	92.96

表8：Mushrooms Database 訓練結果（吉尼係數）

方法	節點數	法則數	訓練錯誤率	測試錯誤率	最大深度	平均深度	建構時間(秒)
ID3	31	24	0.00%	0.00%	4	1.57	18.63
NML	15.8	8.4	0.00%	0.00%	6.2	2.42	52.72
CS	18.2	9.6	0.29%	0.52%	6.8	2.47	88.69

（四）保險公司標竿資料庫 (The Company Insurance Benchmark, COIL 2000)

從表9與表10顯示，若僅就本研究的NML法與CS法此二種二元分割的方法而言結果均顯示NML法表現較佳。若以本研究之NML法與ID3法相比較，可節省之節點數：在熵函數時為51.6%，在吉尼係數下可節省47.4%。NML法與CS法之二元分割決策樹之正確性表現亦比ID3法較高。在決策樹的建構時間上，若就NML法與CS法此二種二元分割的方法而言，當以熵函數為不純度評估時，可節省39%的建構時間；以吉尼係數為不純度評估時，可節省38%的建構時間。

表9：The Company Benchmark Database訓練結果（熵函數）

方法	節點數	法則數	訓練錯誤率	測試錯誤率	最大深度	平均深度	建構時間(秒)
ID3	1888	1033	1.06%	14.75%	26	14.17	1348
NML	913	436	0.69%	13.53%	45	17.56	2007
CS	1014	489	0.72%	13.65%	46	17.81	3300

表10：The Company Benchmark Database訓練結果（吉尼係數）

方法	節點數	法則數	訓練錯誤率	測試錯誤率	最大深度	平均深度	建構時間(秒)
ID3	2233	1301	0.53%	17.93%	26	14.47	1488
NML	1175	565	0.03%	15.75%	45	17.85	2098
CS	1423	688	0.05%	16.40%	45	18.49	3392

（五）訓練所得之決策樹

由於蕈類資料庫 (Mushrooms Database)訓練所得之決策樹節點較少，故將其訓練所得之決策樹列出做為參考。圖1、圖2、與圖3為採用熵函數不純度評估，以同質率99%為分枝終止條件，在某一次執行訓練所得到的結果，其中決策樹的表示方式為：

[屬性] 分枝→類別（資料筆數）

由結果顯示，本研究的NML法與Coppersmith之CS法所產生的二元決策樹較ID3的決策樹含有較少之節點與法則數，但NML法與CS法之二元決策樹在各種深度（最大與平均深度）顯示皆較ID3決策樹更深。此外，從圖2與圖3亦可看出， NML法與CS法之二元分割的分割點並不相同，故所產生的二元決策樹亦不同。但是從樹的規模來看，NML法的節點數包含樹葉節點(leaf node)有13個，所產生的法則數有7條；CS法的節點數有15個，所產生的法則數有8條。

```
[ 5]1 -> 1 (400)
[ 5]2 -> 1 (400)
[ 5]3 -> 2 (192)
[ 5]4 -> 2 (576)
[ 5]5 -> 2 (2160)
[ 5]6 -> 2 (36)
[ 5]7 [19]1 -> 1 (1296)
      [19]2 -> 1 (1344)
      [19]3 -> 1 (48)
      [19]4 -> 1 (48)
      [19]5 -> 2 (72)
      [19]6 -> 1 (48)
      [19]7 [21]1 -> 1 (288)
            [21]2 [ 3]1 -> 1 (24)
                  [ 3]2 -> 1 (24)
                  [ 3]3 -> 2 (8)
                  [ 3]4 -> 2 (8)
            [21]3 -> 1 (40)
            [21]4 -> 1 (192)
            [21]5 [ 8] 1 -> 1 (8)
                  [ 8] 2 -> 2 (32)
      [19]8 -> 1 (48)
[ 5]8 -> 2 (256)
[ 5]9 -> 2 (576)
```

圖1：ID3之決策樹之例示

```
[ 5]1 -> 2 (3760)
[ 5]2 [19]1 [20]1 [21]1 [ 3]1 -> 1 (400)
                        [ 3]2 -> 2 (16)
                  [21]2 [ 8]1 -> 2 (140)
                        [ 8]2 -> 1 (72)
            [20]2 -> 1 (344)
      [19]2 -> 1 (3392)
```

圖2：NML法之二元決策樹之例示

```
[ 5]1 -> 2 (3796)
[ 5]2 [19]1 [17]1 -> 2 (72)
            [17]2 -> 1 (240)
      [19]2 [14]1 -> 1 (3768)
            [14]2 [21]1 [ 8]1 -> 2 (32)
                        [ 8]2 -> 1 (16)
                  [21]2 [ 3]1 -> 1 (192)
                        [ 3]2 -> 2 (8)
```

圖3：CS法之二元決策樹之例示

伍、結論

目前已有許多知識發掘的技術，這些技術執行的時間複雜度，與歸納結果的簡潔度都與資料量的多寡有相當敏感的關係。由於今日大型資料庫的普及，造成知識發掘技術所得到的結果過於複雜、不易解讀。本研究以決策樹為研究對象，利用主成分分析法，由解釋變異最多之主成分來評估該主成分分數與最終決策點之關係。利用此構想，將名目屬性值作二元分割，以求能增進決策樹應用在大型資料庫上之效能。本文的研究結果可得知：

二元決策樹在搜尋空間、搜尋效率與記憶體利用上，由於使用較少之節點數，會較多元決策樹更有效率。當資料集內之屬性，其屬性值越多時，二元分割後之決策樹可節省的節點越多。由於二元分割方式可以合併沒有必要另外獨立分枝的屬性值，因此二元決策樹擁有較少枝節點數，且屬性值越多的屬性，可以合併之屬性值亦越多。二元分割後所得之決策樹，正確性仍維持一定的水準。本文中所測試的四個資料庫所獲致的正確性評估指標，二元分割方法由於可將資料雜訊之影響減少，反而可得較佳之正確性。

名目屬性二元決策樹在選擇分割點的時間複雜度的比較上，以n個屬性值的名目屬性而言，若一一搜尋測試最佳分割點，則有2^{n-1}-1種的二元分割的組合方式；以Coppersmith et al. (1999)方法則需n-1次的比較測試；本論文係針Coppersmith等的方法提出改良，每次分割僅需一次判定即可，能節省更多計算之步驟與時間，可提升名目屬性二元決策樹的建構效率。經實驗證明，本文所提出的方法確實可大幅縮減二元決策樹的建構時間；雖然本文所提出啟發式的方法並非全域最佳解，但本文與Coppersmith等的方法在分類正確性與決策樹的精簡表現上並不較差，顯示本文所提的方法是可行的方法。

致謝

本論文承國科會經會補助（計畫編號：NSC 92－2416－H－218－012），特此誌謝。

參考文獻

1. Aha, D.W., and Breslow, L.A. "Comparing simplification procedures for decision trees," *Artificial Intelligence and Statistics* (5), 1998, pp.199-206.
2. Auer, P., Holte, R. C., and Maass, W. "Theory and applications of agnostic PAC-learning with small decision trees," *Proceedings of the twelfth international conference on machine learning*, 1995, pp.21-29.
3. Bohanec, M., and Bratko, I. "Trading accuracy for simplicity in decision trees," *Machine Learning* (15), 1994, pp.223-250.
4. Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. *Classification and Regression Trees*, Wadsworth International Group, Monterey, 1984.
5. Brodley, C. E., and Utgoff, P. E. "Multivariate decision trees," *Machine Learning* (19), 1995, pp.45-77.
6. Carvalho, D. R., and Freitas, A. A. "A hybrid decision tree/genetic algorithm method for data mining," *Information Science* (163:1-3), 2004, pp.13-35.
7. Cherkauer, K. J., and Shavlik, J. W. "Growing simpler decision trees to facilitate knowledge discovery," *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 315-318.
8. Coppersmith, D., Hong, S. J., and Hosking, J. R. M. "Partitioning nominal attributes in decision trees," *Data Mining and Knowledge Discovery* (3), 1999, pp. 197-217.
9. Cox, L. A., Qiu, Y., and Kuehner, W. "Heuristic least-cost computation of discrete classification functions with uncertain argument values," *Annals of Operations Research* (21:1), 1989, pp. 1-30.
10. Esposito, F., Malerba, D., and Semeraro, G. "A Comparative Analysis of Methods for Pruning Decision Trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (19), 1997, pp. 476-491.
11. Han, J., and Kamber, M. *Data mining: Concepts and techniques*, Morgan Kaufmann, San Francisco, 2001.
12. Heath, D.G. *A geometric framework for machine learning*, PhD thesis, Johns Hopkins, 1993.
13. Hoffgen, K. U., Simon, H. U., and Horn, K. S. "Robust trainability of single neurons," *Journal of Computer system Sciences* (50:1), 1995, pp. 114-125.
14. Hyafil, L., and Rivest, R. L. "Constructing optimal binary decision tree is NP-complete," *Information Processing Letters* (5:1), 1976, pp. 15-17.
15. John, G. H. "Robust Decision Trees: Removing Outliers in Databases," *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, 1995, pp. 174-179.

16. Laber, E. S., and Nogueira, L. T. "On the hardness of the minimum height decision tree problem," *Discrete Applied Mathematics* (144:1-2), 2004, pp. 209-212.
17. Lim, T. S., Loh, W. Y., and Shih, Y. S. "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Machine Learning* (40), 2000, pp. 203-229.
18. Mehta, M., Agrawal, R., and Rissanen, J. "SLIQ: A fast scalable classifier for data mining," *Proceedings of the Fifth International Conference on Extending Database Technology*, Avignon, France, 1996.
19. Murphy, O. J., and McCraw, R. L. "Designing storage efficient decision trees," *IEEE Transactions on Computers* (40:3), 1991, pp. 315-319.
20. Murthy, S. K. "Automatic construction of decision trees from data: a multi-disciplinary survey," *Data Mining and Knowledge Discovery* (2:4), 1998, pp. 345-389.
21. Naumov, G. E. "NP-completeness of problems of construction of optimal decision trees," *Soviet Physics Doklady* (36:4), 1991, pp. 270-271.
22. Osei-Bryson, K. "Evaluation of decision trees: a multi-criteria approach," *Computers and Operations Research* (31:11), 2004, pp. 1933-1945.
23. Pagallo, G., and Haussler, D. "Boolean feature discovery in empirical learning," *Machine Learning* (5), 1990, pp. 71-100.
24. Quinlan, J. R. "Induction of decision tree," *Machine Learning* (1), 1986, pp. 81-106.
25. Quinlan, J. R. *C4.5: Programs for machine learning*, Morgan Kaufmann, San Mateo, 1993.
26. Quinlan, J. R. "Improved use of continuous attributes in C4.5," *Journal of Artificial Intelligence Research* (4), 1996, pp. 77-90.
27. Ragavan, H., and Rendell, L.A. "Lookahead feature construction for learning hard concepts," *Proceedings of the Tenth International Conference on Machine Learning*, 1993, pp. 252-259.
28. Ruggieri, S. "Efficient C4.5," *IEEE Transactions on Knowledge and Data Engineering* (14:2), 2002, pp. 438-444.
29. Sorensen, K., and Janssens, G. K. "Data mining with genetic algorithms on binary trees," *European Journal of Operational Research* (151:2), 2003, pp. 253-264.
30. Takimoto, E., and Maruoka, A. "Top-down decision tree learning as information based boosting," *Theoretical Computer Science* (292:2), 2003, pp. 447-464.
31. Terano, T., and Ishino, Y. "Knowledge acquisition from questionnaire data using simulated breeding and inductive learning methods," *Expert Systems with Applications* (11:4), 1996, pp. 507-518.
32. Zheng, Z. "Constructing nominal X-of-N attributes," *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (2), 1995, pp. 1064-1070.