

運用資料探勘技術於知識圖之建立

學生：鄭祐昇

指導教授：黃欽印 教授

東海大學工業工程與經營資訊研究所

摘要

知識圖，也就是貝氏網路，是一種圖形化知識表示方法。藉由節點、連結與條件機率表三部分，呈現出知識中的不確定性和因果關係。網路中，節點與節點之間的連結建立過程，稱之為貝氏網路架構學習，為本研究探討的重點。

貝氏網路架構學習可藉由專家知識來完成。專家靠著自己的經驗累積，可以較平常人對於領域知識更能有效的預測或控制。不過現在的環境越來越複雜，每個領域知識間會更有相互的影響。且專家礙於對問題主觀的認知，以及專家之間會產生意見衝突的情形。有賴資訊技術在資料傳輸、儲存和計算的長足進步，以及資料探勘技術的盛行。貝氏網路架構學習，除了倚賴專家的經驗之外，也可以利用資料探勘技術，從客觀的歷史資料分析來完成。

本研究針對歷史資料，利用資料探勘技術中的粗集理論，粹取隱藏在資料中的決策法則。經過一些法則篩選的程序後，產生最終的建構法則。再利用這些建構法則轉換成節點間的連結，完成貝氏網路架構學習。而此建構的結果，在架構與資料結構相似性比較上，會比 K2 演算法有更好的建構績效。

關鍵字詞：知識圖，貝氏網路，資料探勘，粗集理論

Applying Data Mining Technoque in the Development of Knowledge Map

Student: Yu-Sheng Cheng

Advisor: Prof. Chin-Yin Huang

Department of Industrial Engineering and Enterprise Information
Tunghai University

ABSTRACT

Knowledge map, also named Bayesian network, is a graphical model that encodes probabilistic relationships among variables of interest. A Bayesian network can be used to learn causal relationships, and hence can be used to gain understanding about a problem domain and to predict the consequences of intervention.

To develop the dependency structure of Bayesian network is referred to as structural learning. Human experts can use their domain knowledge to develop the structures. But different experts can give different and sometimes conflicting assessments due to the subjective nature of the process.

More recently, researchers have developed methods for learning Bayesian networks from data. Data mining is a popular data analytical method and can apply to develop the structures.

In this thesis, one of the data mining techniques, called rough sets theory, is applied to generate decision rules from data, and apply those rules to develop the Bayesian networks. The result indicates that the procedures developed in this thesis can deliver a better outcome in terms of causal structure and data similarity, compared with K2 algorithm.

Keywords: Knowledge map, Bayesian network, Data mining, Rough sets theory

目錄

中文摘要	i
英文摘要	ii
目錄.....	iii
圖目錄	v
表目錄	vii
第一章 緒論	1
1.1 研究背景與動機	1
1.2 研究目的與範圍	3
1.3 研究方法與步驟	4
1.4 論文架構	6
第二章 文獻探討	8
2.1 貝氏網路理論	8
2.2 貝氏網路架構學習	11
2.2.1 搜尋式演算法	14
2.2.2 限制式演算法	18
2.2.3 d-separation	20
2.3 資料探勘方法	22
第三章 粗集理論與建構貝氏網路	26
3.1 資料產生	27
3.2 資料轉換與法則產生	28
3.2.1 粗集理論基本概念	28
3.2.2 挖掘法則.....	37
3.3 法則篩選	40
3.3.1 法則支持度	40
3.3.2 法則長度.....	41
3.3.3 顯著性檢定	41
3.4 法則建構網路	42
3.5 參數學習	44
3.6 建構績效比較	44
3.6.1 架構相似性	45
3.6.2 資料結構相似性	45
3.6.3 關鍵節點相似性	47
第四章 實證研究	48
4.1 範例一：警鈴	48

4.1.1 粗集方法建構之警鈴貝氏網路	48
4.1.2 K2 演算法建構之警鈴貝氏網路	53
4.2 範例二：動物分類	56
4.2.1 粗集方法建構之動物分類貝氏網路	56
4.2.2 K2 演算法建構之動物分類貝氏網路	62
4.3 範例三：汽車故障診斷	69
4.3.1 粗集方法建構之汽車故障診斷貝氏網路	70
4.3.2 K2 演算法建構之汽車錯誤診斷貝氏網路	79
4.4 本章總結	84
第五章 結論與建議	85
5.1 結論.....	85
5.2 未來研究方向	85
參考文獻	87

圖目錄

圖 1.1 研究步驟示意圖	6
圖 1.2 論文架構示意圖	7
圖 2.1 貝氏網路與條件機率表	9
圖 2.2 K2 演算法流程	18
圖 2.3 SGS 演算流程	19
圖 2.4 D-SEPARATION 示意圖	21
圖 2.5 汽車電子引擎系統的 D-SEPARATION 範例	22
圖 3.1 本研究之貝氏網路建構步驟	27
圖 3.2 粗集近似空間	31
圖 3.3 法則長度 1 時可能之貝氏網路架構	42
圖 3.4 法則長度 2 時可能之貝氏網路架構	43
圖 3.5 完整圖形範例	43
圖 3.6 貝氏網路模型	45
圖 4.1 警鈴之貝氏網路	48
圖 4.2 產生 ATTRIBUTE5 (MARY) 建構法則	50
圖 4.3 本研究方法建構之警鈴貝氏網路雛型	50
圖 4.4 本研究方法架構之警鈴貝氏網路圖形	51
圖 4.5 本研究方法之警鈴貝氏網路魏克森符號檢定	52
圖 4.6 K2 演算法建構之警鈴貝氏網路雛型	53
圖 4.7 K2 演算法建構之警鈴貝氏網路	54
圖 4.8 K2 演算法之警鈴貝氏網路魏克森符號檢定	55
圖 4.9 動物分類之貝氏網路	56
圖 4.10 本研究方法建構之動物分類貝氏網路雛型	58
圖 4.11 計算關聯的 UNEXPLAINED INFORMATION 值	59
圖 4.12 去除多餘關聯後的動物分類貝氏網路	59
圖 4.13 本研究方法架構之動物分類貝氏網路圖形	60
圖 4.14 本研究方法建構之動物分類貝氏網路魏克森符號檢定	61
圖 4.15 K2 演算法建構之動物分類貝氏網路雛型	62
圖 4.16 K2 演算法建構之動物分類貝氏網路	63
圖 4.17 K2 演算法建構之動物分類貝氏網路魏克森符號檢定	64
圖 4.18 MOKEY 機率比較長條圖	66
圖 4.19 PENGUIN 機率比較長條圖	67
圖 4.20 PLATYPUS 機率比較長條圖	67
圖 4.21 ROBIN 機率比較長條圖	68
圖 4.22 TURTLE 機率比較長條圖	68

圖 4.23 汽車診斷之貝氏網路	70
圖 4.24 本研究方法建構之汽車診斷貝氏網路雛型	73
圖 4.25 計算汽車診斷貝氏網路 UNEXPLAINED INFORMATION 值	74
圖 4.26 去除多餘關聯後的汽車網路貝氏網路雛型	74
圖 4.27 本研究方法建構之汽車診斷貝氏網路	75
圖 4.28 範例汽車診斷貝氏網路關鍵節點	77
圖 4.29 本研究方法建構之汽車診斷貝氏網路關鍵節點	77
圖 4.30 本研究建構之汽車診斷貝氏網路魏克森符號檢定	78
圖 4.31 K2 演算法建構之汽車診斷貝氏網路雛型	79
圖 4.32 K2 演算法建構之汽車診斷貝氏網路	80
圖 4.33 K2 演算法建構之汽車診斷貝氏網路關鍵節點	82
圖 4.34 K2 演算法建構之汽車診斷貝氏網路魏克森符號檢定	83

表目錄

表 2.1 搜尋式建構演算法一覽表	12
表 2.2 限制式建構演算法一覽表	13
表 3.1 資訊系統	29
表 3.2 對等關係集合	30
表 3.3 決策表 A	33
表 3.4 決策表 B	34
表 3.5 可辨識矩陣	34
表 3.6 REDUCT {A, B} 決策表	35
表 3.7 REDUCT {A, B} 可辨識矩陣	36
表 3.8 最終 REDUCT {A, B} 決策表	37
表 3.9 決策表 C	39
表 3.10 REDUCT {A, D, E, I} 決策法則	39
表 3.11 REDUCT {A, I} 決策法則	40
表 4.1 警鈴之變數狀態值轉換	49
表 4.2 警鈴之建構法則	50
表 4.3 動物分類之變數狀態值轉換	57
表 4.4 動物分類之建構法則	58
表 4.5 機率分配比較	66
表 4.6 汽車診斷之變數狀態值轉換	71
表 4.7 汽車診斷之建構法則	72
表 4.8 架構相似性比較表	84
表 4.9 資料結構相似性比較表	84

第一章 緒論

1.1 研究背景與動機

假設任何的現象背後，都會遵循著一個規則在進行，倘若我們能挖掘出這種規則，則有助於預測或控制此現象的未來趨勢。例如市場變化，消費者行為模式，疾病和病徵的關聯，以及機器設備的錯誤診斷等。

專家靠著自己的經驗累積，可以較平常人對於領域知識（domain knowledge）更能有效的預測或控制。不過現在的環境越來越複雜，每個領域知識間會更有相互的影響。有賴資訊技術在資料傳輸、儲存和計算的長足進步，以及資料探勘（data mining）的盛行，專家除了倚賴自己的經驗之外，也可以由歷史資料來對事件做判斷。資料探勘就好比是專家的腦袋一樣，利用過去發生的事件，學習事件背後的規則，用以推論未來可能會發生的情形。

資料探勘兩個主要功能，分別是描述（description）和預測（prediction）。描述是將資料中所包含的特性，做概念性的描述。預測是透過現有的資料，推測未來的趨勢。資料探勘所含的方法，有些可以達到這兩種功能，有些只能盡其一。依據[18][21]，較常使用的資料探勘方法可分為四個部分，如下：

1. 概念描述（concept description）：將與分析目標相關的資料用簡明的方法，呈現出資料的普遍性特徵。常用的方法包括屬性導向歸納法（Attribute-Oriented Induction; AOI），線上分析技術（On-Line Analytical Processing; OLAP）等。
2. 關聯分析（association analysis）：找出資料中有相互關聯的屬性，當信賴度超過門檻值時，以法則的方式呈現出來。例如 apriori algorithm，FP-tree 等。
3. 分類（classification）與預測：屬於監督式學習（supervised learning），利用訓練資料產生的分類，預測未知資料的分類。例如決策樹

(Decision tree), 類神經網路 (Neural networks), 基因演算法 (Genetic algorithms), 貝氏網路 (Bayesian networks), 粗集理論 (Rough sets theory) 等。

4. 分群 (clustering): 屬於非監督式學習 (unsupervised learning), 經由相似性衡量, 將相似的資料歸為一群。例如 k-means algorithm, agglomerative 等。

知識圖 (Knowledge maps), 也就是貝氏網路, 是資料探勘常用的分類演算法之一, 為本論文要探討的對象。其他的名稱包括信賴網路 (Belief Networks) 因果機率網路 (Causal Probabilistic Networks; CPNs)。貝氏網路是一種圖形的知識表示方法, 藉由網路圖形表示出知識中的不確定性 (uncertainty) 和因果關係 (causal relationship), 因此貝氏網路可以兼顧定量和定性兩方面分析。

資料探勘有許多的分析方法, 而貝氏網路有四特點是與其他方法明顯不同, 如下[3][19]:

1. 貝氏網路可以處理不完整資料的情形。對於領域知識內的不完整資料, 貝氏網路可利用如 Monte-Carlo, Gaussian approximation, MAP and ML approximation 和 EM algorithm 等一些演算法來彌補因資料的不完整產生預測不準確情形。
2. 貝氏網路可學習變數間的因果關係。因果關係可讓領域知識易於了解, 以及看出當干擾出現時對網路產生的影響, 有助於決策擬定和意外預防。
3. 貝氏網路以貝氏統計結合領域知識和歷史資料。
4. 貝氏網路可避免資料過度配適 (over fit) 的問題。

貝氏網路建構可視為是一種網路學習。網路學習的問題有許多變化, 其中包含網路架構是已知或未知, 以及網路中的變數是確定或不確定。在不同的情況下, 有不同的學習重點。因此, 網路學習可分為四種情形[29]:

1. 架構已知, 變數確定: 稱為參數學習, 在於完成各變數的條件機率表

(conditional probability tables; CPTs)。

2. 架構未知，變數確定：稱為架構學習，在於從許多可能的網路架構中，搜尋出符合資料描述的架構。其後也要進行參數學習。
3. 架構已知，變數不確定：此時會有隱藏變數存在的問題，類似類神經網路學習。
4. 架構未知，變數不確定：此時學習問題變的極為困難，至今尚未有適當的方法來解決。

專家的知識對於建構貝氏網路有很大的幫助，但是當問題的變數過多、關係變的複雜的時候，就需要利用一些演算法，從資料面來分析出可能的架構。本研究的動機在於如何從資料面，得到建構網路所需要的資訊，進而有助於縮短建構貝氏網路的時程。

1.2 研究目的與範圍

本論文研究目的在於第二種網路學習的問題。亦即在不確定的網路架構下，尋找能符合資料描述的網路架構。研究目的詳述如下：

1. 假設專家所蒐集的變數可以充分表達領域知識，亦即不會出現有隱藏變數的情形。本研究可以從這些變數中，挖掘出可表達這些變數間關係的建構法則。利用這些法則，建構成貝氏網路。完成貝氏網路中建構學習的部分。
2. 貝氏網路除了是一個分類的方法，也是一個很好的知識表示方法。因此貝氏網路建構的目的，包括預測、知識解釋和探索等。當觀察者希望能藉由貝氏網路達到上述三種目標時，均能使用本研究所提出的方法來完成貝氏網路。

本研究所提出的建構方法—粗集理論，屬於資料探勘中的分類演算法。粗集理論可以只針對資料作分析，不需要額外的函數輔佐。粗集理論適合用於多變數分析，但因粗集理論本身的限制，本研究方法有下面的限制：

1. 粗集理論因只能針對資訊系統中的資料作分析。若要以本研究方法來建構貝氏網路，必須選擇可以表示成資訊系統的資料。
2. 建構資料必須是要有意義的，也就是資料內的變數間存在有關聯。若是收集一些極不相干的資料，即使最終能以本研究方法建構成貝氏網路，也不能符合貝氏網路建構目的。
3. 當資料出現有遺漏值（missing value）的情形，粗集理論就不適用。儘管有些統計的方法可以處理遺漏值問題，不過這會影響到粗集理論推論的精確性。因此，選用的資訊系統中的資料不能有遺漏值。
4. 本研究的目的是在貝氏網路架構學習，研究範圍不包括參數學習的部分。

1.3 研究方法與步驟

本研究的進行步驟可分為四階段，詳述如下（四階段關聯如圖 1.1）。

步驟 1. 相關文獻蒐集與探討

1. 貝氏網路相關文獻：貝氏網路基本理論介紹，包含貝氏網路的定義和特徵以及機率推論。
2. 建構學習相關文獻：貝氏網路發展至今，在學術上曾對於貝氏網路架構學習相關方法的整理。
3. 粗集理論相關文獻：粗集理論用於網路架構學習方面的研究。

步驟 2. 貝氏網路架構學習問題分析

1. 搜尋式演算法：至今最常用的網路架構演算法，經由品質衡量方法以及搜尋演算法，求出在可能的網路架構中，擁有最高的品質，即為最後的貝氏網路。
2. 限制式演算法：此類方法的網路架構理論基礎，認為網路上的變數，是因變數之間存在於較高的相依性，方能夠成變數之間的關聯。相依性高低的判定是利用透過變數間的條件獨立測試決定。
3. 條件獨立測試：網路架構演算法常利用的基本假設。條件獨立測

試可使網路機率推論更加有效率，且有助於簡化網路架構以建立變數間的關聯。

4. 圖論：貝氏網路為圖論中的一種，因此貝氏網路會符合某些圖論的特性。這些特性也有助於利用到貝氏網路架構的建立。包含變數間的關聯，條件獨立，d-separaton 等。

步驟 3. 粗集理論用於網路建構之探討

1. 粗集理論：粗集理論的基本介紹，包含 reduct、core、近似推論以及決策法則。
2. 顯著性檢定：傳統粗集理論所得到的法則不能全部採用，可利用統計的假設檢定，來檢定產生的法則是否為隨機性(casual)情形。
3. 不確定性衡量：粗集理論利用資訊理論中的熵函數，推導出的不確定性衡量公式。計算網路中傳遞的資訊量情形，以簡化網路架構。

步驟 4. 實證研究與說明

1. 貝氏網路應用軟體—Norsys Netica[24]，內建若干貝氏網路範例。本研究利用此軟體從範例網路產生資料，再以這些資料當作理論驗證的資料來源。同樣以此軟體完成貝氏網路參數學習的部分。
2. 挖掘建構法則，利用粗集理論分析軟體—Grobian[15]。
3. 以相同的資料，搭配貝氏網路應用軟體—Bayesware Discovery，當作實證研究實的績效比較。此軟體以 K2 演算法為基礎。
4. 績效比較分為三部分，架構相似性、資料結構相似性和關鍵節點相似性。
5. 實證研究採用三個範例，分別如下。
 - (1) 警鈴：網路包含 5 個節點，用以解釋警鈴可能發生的原因以及是否會做出錯誤判段的情形。
 - (2) 動物分類：網路包含 8 個節點，以 7 個分類指標對 5 種動物進行分類。
 - (3) 汽車錯誤診斷：網路包含 15 個節點，用 14 個觀測點的好壞情

形，決定汽車能否發動。

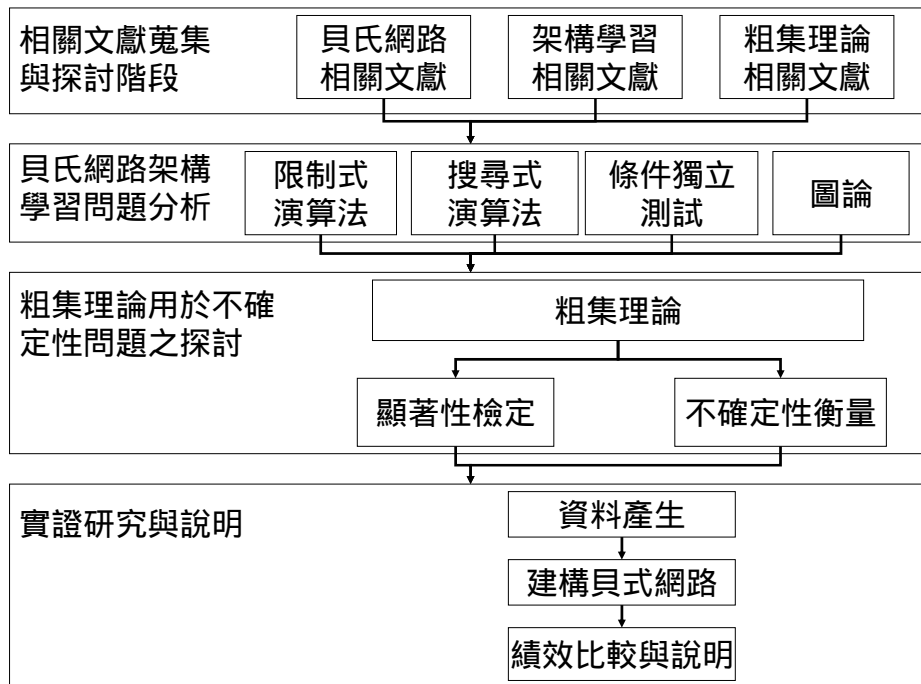


圖 1.1 研究步驟示意圖

1.4 論文架構

本論文可分為五個章節來介紹。各章節的內容概述如下（各章節關係如圖 1.2）。

1. 第一章『緒論』：說明本研究的研究背景與動機，進而提出研究目的與範圍，並概略說明研究進行的方法與步驟。
2. 第二章『文獻探討』：針對過去提出的網路架構學習之研究，進行文獻的蒐集、整理與探討。
3. 第三章『粗集理論與建構貝氏網路』：利用本研究提出的粗集理論，挖掘資料中存在的 uncertainty 法則，試圖利用這些法則來建構出貝氏網路的架構。
4. 第四章『實證研究』：以 Norsys Netica 中的貝氏網路範例產生出資料，再搭配第三章提出的方法，進行本研究的理論驗證。
5. 第五章『結論與建議』：說明本研究的結果與心得，並提出後續研究的方向與建議。

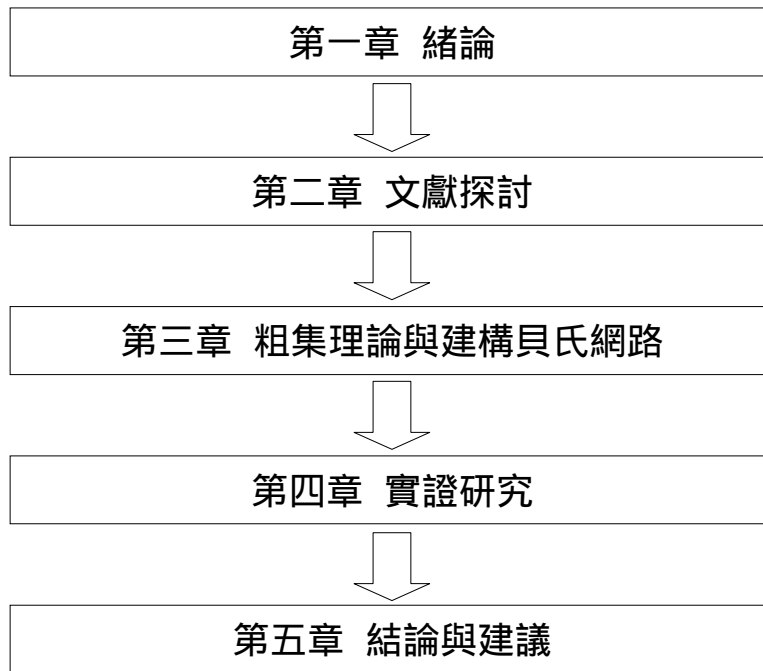


圖 1.2 論文架構示意圖

第二章 文獻探討

本章針對貝氏網路在網路建構上有關的研究，進行整理和探討。2.1 節先介紹貝氏網路基本理論；2.2 節針對網路建構問題進行相關方法的整理和探討；2.3 節介紹資料探勘方法在網路架構上的研究，並引導出本論文選用的架構方法。

2.1 貝氏網路理論

網路以非循環方向圖形 (directed acyclic graph; DAG) 方式呈現以有悠久的歷史[27]。最早應用的時候是在 1921 年 Sewall Wright。他發展出一套方法，稱為路徑分析 (Path Analysis)，之後成為用來表示經濟，社會或心裡學中因果關係的基本方式。推論圖形 (inference diagrams) 成為非循環方向圖形的另一種應用，用來作為決策分析，網路中包含事件節點和決策節點。

上述網路的主要應用，在於有效的描述變數機率函數。故當網路架構完成時，就進行一連串機率值的計算。此目的是要獲得從下而上，或是上而下的機率推論，都能得到一致性的結果。計算的核心是根據 Reverend Bayes 在 1763 年提出的貝氏定理，經由事前機率和概似函數計算事後機率。

貝氏網路並不注重在邏輯和認知模型的循環，但廣為應用在專家系統。過去 10 年，貝氏網路成為一多用途的工具，現在是最常用的機率知識表現方式。利用在疾病診斷輔助、機器故障系統、情節敘述、文件篩選、圖形解釋、不確定環境下的廠區規劃等方面[27]。

貝氏網路如圖 2.1，由下面四點主要特徵構成[4][9][19]：

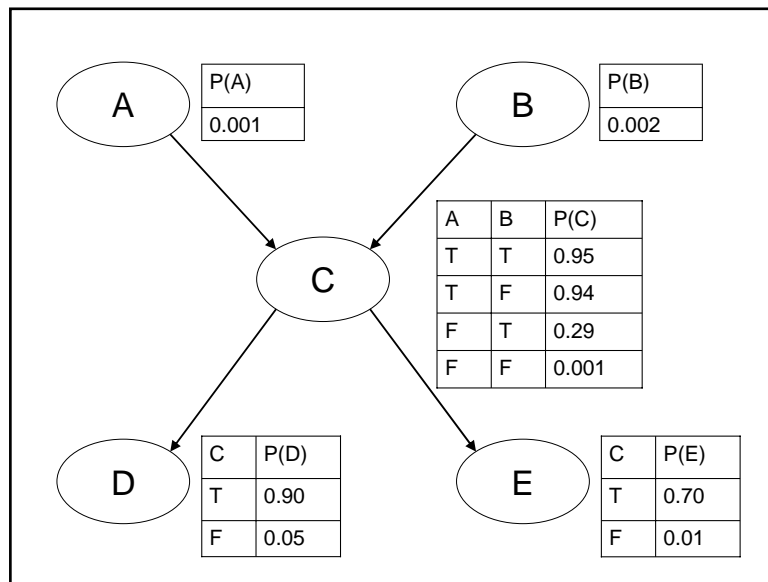


圖 2.1 貝氏網路與條件機率表

1. 有一組隨機變數 $X = \{X_1, \dots, X_n\}$ ，構成網路上的節點 (nodes)。如圖中的 A,B,C,D,E 均是。
2. 任兩變數間存在著相依關係，就會有一連結 (link)，此這連結必須是具方向性 (directed)。例如圖中，A 有連結指向 C，表示 A 的狀態會影響 C。稱 C 為 A 的子節點 (descendant node)，A 為 C 的父節點 (ancestor node)。若變數與其他變數都是獨立關係，則在圖中不會有任何連結，變數應該被消除。
3. 每個節點透過與其父節點的條件機率分配 (conditional probability distribution; CPD)，可計算出在父節點的影響下，節點狀態值之機率變化。此機率變化的情形可繪成條件機率表 (conditional probability table; CPT)。
4. 網路圖形是一非循環方向圖形。

經由上述的特徵，貝氏網路定義如下[9][19]。

{ 定義: 貝氏網路 } 貝氏網路 B 可表示成， $B = (D, P)$ 。D 是一非循環方向圖形 (DAG)。若網路有 n 個節點，P 是每個節點條件機率分配所構成的集合，則 $P = \{p(x_1 | \dots | x_{n-1}), p(x_n | \dots | x_{n-1})\}$ 。每個子節點與其父節點的條件機率分

配乘積，稱為此貝氏網路 B 的聯合機率分配 (joint probability distribution; JPD)，如(2.1)式：

$$p(x) = \prod_{i=1}^n p(x_i | \pi_i) \quad (2.1)$$

由(2.1)式，圖 2. 1 的貝氏網路可表示成如下式的聯合機率密度，或是各個變數的條件機率密度的乘積：

$$p(a,b,c,d,e) = p(a)p(b)p(c|a,b)p(d|c)p(e|c) \quad (2.2)$$

貝氏網路的機率推論基礎為貝氏定理。假設事件 A 的事前機率 (prior) 為 $p(A)$ ，事件 B 發生的情況下，變數 A 發生的條件機率為 $p(A|B)$ 。經由貝氏定理推算，可求得變數 B 的事後機率 (posterior) 為 $p(B|A)$ 。網路架構複雜時，可利用貝氏定理發展出的演算法，例如 exact propagation algorithms, approximate propagation algorithms 和 symbolic propagation algorithms 等 [9]，計算節點的父代與子代的機率。貝氏定理如下式：

$$p(B|A) = \frac{p(A|B)p(B)}{p(A)} = \frac{p(A|B)p(B)}{\sum_i p(A|B_i)p(B_i)} \quad (2.3)$$

貝氏網路中的機率是指對事件會發生的相信 (belief) 程度。相信程度的給定會因人而異，也會因為個人經驗的累積而改變。例如，X 表示丟一公正銅板會是正面的事件。有人相信此事件會發生 ($p(X)=1$)，也有人相信此事件不會發生 ($p(X)=0$)。如果 9 次當中出現了 5 次反面，則第 10 次會出現正面的相信程度，會大於不會出現的相信程度。此種相信程度須符合三大機率公設才有意義[20]，如下：

1. 對於全集 S 中的事件 A_i ， $0 \leq p(A_i) \leq 1$ 。
2. $p(S)=1$ 。
3. 若 A_i, A_j 為 S 中的互斥事件， $A_i \cap A_j = \emptyset$ ， $p(A_i \cup A_j)=p(A_i)+p(A_j)$ 。

貝氏網路建構過程稱為貝氏網路學習。貝氏網路學習包含兩部分，架構學習和參數學習。架構學習是在不確定的變數關係中，決定出變數的關

聯。參數學習是在確定的網路架構下，決定網路節點的機率[9][19]。架構學習為本研究重點，2.2 節將會有進一步的文獻探討。

2.2 貝氏網路架構學習

貝氏網路架構學習，可視為是一種因果關係探索（causal discovery），其目的是在領域知識中，不確定的變數與變數間的關係建構成網路[8]。當變數之間的關係是相依時，就會存在著一條連結。變數間關係是獨立時，將不會有任何連結。

變數關係建構，可由領域知識中的專家來完成，或者是從領域知識蒐集的統計資料分析而得[3]。但是，專家的知識經驗並不容易取得，加上專家都會有自己主觀上的認知，容易造成建構時的衝突，而經由資料面分析建構成網路，是一種較為客觀的方法[9]。

由資料面分析建構貝式網路的方法，[31]區分成三類，包括線性與常態假設、獨立關係測試與貝氏方法。本研究採用[10][11][22]所提出的分類，將常用的建構方法分成兩大類，分別如下：

1. 搜尋式演算法（search-&-scoring-based methods），或稱為貝氏方法（Bayesian approach）。此類演算法可參考表 2. 1。
2. 限制式演算法（constraint-based methods），或稱為非貝氏方法（non-Bayesian approach）。此類演算法可參考表 2. 2。

表 2.1 搜尋式建構演算法一覽表[10]

演算法	最終模型	節點排序	品質衡量	主要特徵
Chow-Liu	Trees(馬可夫網路的一種)	不需要	熵	需要 $O(N^2)$ 次成對的相依性計算
Rebane-Pearl	Polytrees (貝氏網路的一種)	不需要	熵	需要 $O(N^2)$ 次成對的相依性計算, 能提供連結方向
K2	貝氏網路	需要	貝氏方法	啟發式搜尋, 有效率
HGC	貝氏網路	不需要, 但需要事前網路 (prior net)	貝氏方法	利用事前網路作為領域知識
Kutato	貝氏網路	需要	熵	利用條件獨立測試加速熵函數計算
Wong-Xiang	馬可夫網路	不需要	熵	結果均是 I-maps
BENEDICT	貝氏網路	需要	熵	啟發式搜尋, 利用 d-separation
CB	貝氏網路	不需要	貝氏方法	結合 PC 和 K2 演算法, 能提供連結方向
Suzuki	貝氏網路	需要	MDL	能獲得最佳結構, 但無效率
Lam-Bacchus	貝氏網路	不需要	MDL	不結合其他方法, 能提供連結方向
Friedman-Goldszmidt	貝氏網路	不需要	MDL 或貝氏方法	不結合其他方法, 能提供連結方向

表 2.2 限制式建構演算法一覽表[10]

演算法	最終模型	節點排序	CI-tests 次數	主要特徵
Wermuth-Lauritzen	貝氏網路	需要	$O(N^2)$	雖僅需要執行 $O(N^2)$ 次條件獨立測試，但不敷使用
Boundary DAG	貝氏網路	需要	隨變數數量呈指數成長	一種簡單的演算法
SRA	貝氏網路	部分需要	隨變數數量呈指數成長	Boundary DAG 的延伸，只需要部分節點排序，啟發式搜尋
Constructor	馬可夫網路	不需要	隨變數數量呈指數成長	利用交叉驗證的方式避免過度配適
SGS	貝氏網路	不需要	隨變數數量呈指數成長	可以提供連結方向
Verma-Pearl	貝氏網路	不需要	隨變數數量呈指數成長	SGS 另一種形式，可提供連結方向並能察覺方向是否有衝突
PC	貝氏網路	不需要	$O(N^{k+2})$	SGS 加強型演算法，可提供連結方向，有效率

搜尋式演算法，假設最佳的網路模型會最滿足資料。先將貝氏網路轉變成聯合機率密度，再以 Bayesian, MDL (minimum description length) , Kullback-Leibler (KL) entropy function 等品質衡量方法，搜尋出能獲得最大品質衡量係數的網路，作為最終貝氏網路。此類演算法將在 2.2.1 節會有進一步的探討。

限制式演算法，假設網路內的節點會因為相依性而產生連結。這類的限制通常是條件獨立敘述 (conditional independent statements; CI-statements)。條件獨立敘述是利用條件獨立測試 (conditional

independence test; CI-tests) 來判別變數間的獨立或相依關係後的敘述。此類演算法將在 2.2.2 節會有進一步的探討。

Cowell[12]證明出在每一個搜尋式演算法，都可以找到一個對等的限制式演算法，反之亦然。所以這兩類演算法的差異不是在於所使用的衡量方法(品質衡量係數或是 CI-tests)，而是在於能否有效利用 d-separation 的觀念來建構網路模型[10]。有關 d-separation 圖論觀念，在 2.2.3 節會有進一步的探討。

2.2.1 搜尋式演算法

假設收集到的資料可以充分表達領域知識，此類演算法進行步驟如下[9]：

1. 選擇品質衡量方法和搜尋演算法。
2. 利用搜尋演算法選擇有高品質的可能貝氏網路架構當作候選網路。此時需要估計選擇可能網路的參數，然後計算所有可能貝氏網路架構的品質係數。
3. 選出在所有候選的貝氏網路架構中，擁有最高品質衡量係數的貝氏網路。

品質衡量 (quality measure) 方法與搜尋演算法成為此類建構方法的重點。品質衡量是評估選出的網路模型是否符合資料描述的一種評估方法。搜尋演算法是以一套規則在所有可能的網路模型中，搜尋出有高品質的貝氏網路。此兩部份分述如下[9]：

1. 品質衡量方法

品質衡量方法，是計算每個可能網路架構的品質衡量係數。品質衡量方法有貝氏品質衡量方法，MDL 品質衡量方法，資訊衡量方法等。

一般品質衡量方法可看成由三個函數構成，分別是事前資訊 (prior information)、可用資料 (available data) 和網路複雜度 (network complexity)，各部分介紹如下。

- (1) 事前資訊：網路與事前資訊越相似，會得到越高的函數值。資料缺乏時，事前資訊變的相當重要。但資料充足時，事前資訊是可以忽略的。此函數一般常用的表示法為 $\log p(B)$ ，其中 $p(B)$ 表示網路 B 的事前機率， $p(B)=p(D, \theta)$ ，為 $P(JPD)$ 的參數。若無事前資訊，此函數為 0， $p(B)$ 為一均勻分配。
- (2) 可用資料：此函數是一種適合度檢定，檢定網路 D 和資料 S 的適合度。當網路和資料越是一致，此函數值會越高。網路中連結的增加，會讓此函數值變高，因此要增加參數或自由度，讓結果更符合資料。此函數表示為 $\log p(S/D, \theta)$ ，利用網路和參數 (θ) 對資料的條件機率，檢定適合度。
- (3) 網路複雜度：此函數是懲罰網路複雜度過高的模型，所以網路的連結越多或參數越多，此函數值會越低。

品質衡量方法中，最常使用的是貝氏品質衡量方法，因此搜尋式建構方法又可稱為貝氏方法。在此以貝氏衡量方法為例，作為品質衡量方法的介紹[9]。

貝氏品質衡量方法，是經由計算每個可能網路的事後機率密度求得。假設建構資料表示為 S，可能網路架構為 D，網路參數為 θ 。則可能網路的事後機率密度如下式：

$$\begin{aligned}
 p(D, \theta | S) &= \frac{p(D, \theta, S)}{p(S)} = \frac{p(D, \theta, S)}{\sum_{D, \theta} p(D, \theta, S)} \\
 &= \frac{p(D, \theta) p(S | D, \theta)}{\sum_{D, \theta} p(D, \theta) p(S | D, \theta)} = \frac{p(S, D, \theta)}{\sum_{D, \theta} p(D, \theta) p(S | D, \theta)} \quad (2.4)
 \end{aligned}$$

(2.4)式的分母為一常數項，所以可簡化成(2.5)：

$$\begin{aligned}
 p(D, \theta | S) &\propto p(S, D, \theta) = p(D) p(\theta | D) p(S | D, \theta) \\
 &= p(D) p(\theta | S, D) p(S | D) \propto p(D) p(\theta | S, D) \quad (2.5)
 \end{aligned}$$

經由上述的轉換，每個可能網路的事後機率密度 $p(D, \theta | S)$ ，改由 $p(S, D)$ 或者 $p(D)p(\theta | S, D)$ 來計算。也就是說，品質衡量係數可由下面各因子 $p(D)$ ， $p(\theta | D)$ ， $p(S | D, \theta)$ ， $p(\theta | S, D)$ 計算。此四因子的意思如下：

- (1) $p(D)$ ：每個節點在網路 D 的事前機率密度，由專家提供。
- (2) $p(\theta | D)$ ：每個節點在網路 D 時的參數事前機率密度，由專家提供。
- (3) $p(S | D, \theta)$ ：在網路 D 與參數 θ 時，對資料 S 的概似估計。
- (4) $p(\theta | S, D)$ ：參數 θ 在網路 D 以及資料 S 時的事後機率密度。

利用這些基本因子，可以構成貝氏品質衡量係數。貝氏品質衡量係數如下三式：

$$\begin{aligned} Q_{GH}(D, S) &= \log p(D) + \log \int p(S | D, \theta) p(\theta | D) d\theta \\ &= \log p(D) + \sum_{i=1}^n \left[\sum_{k=1}^{s_i} \left[\log \frac{\Gamma(\eta_{ik})}{\Gamma(\eta_{ik} + N_{ik})} + \sum_{j=0}^{r_i} \log \frac{\Gamma(\eta_{ijk} + N_{ijk})}{\Gamma(\eta_{ijk})} \right] \right] \end{aligned} \quad (2.6)$$

(2.6)式為 Geiger 和 Heckerman 提出，其中 $\Gamma(\cdot)$ 為一 gamma 函數，此衡量方法不包含懲罰函數。

$$Q_{CH}(D, S) = \log p(D) + \sum_{i=1}^n \left[\sum_{k=1}^{s_i} \left[\log \frac{\Gamma(r_i)}{\Gamma(N_{ik} + r_i)} + \sum_{j=0}^{r_i} \log \Gamma(N_{ijk} + 1) \right] \right] \quad (2.7)$$

(2.7)式為 Cooper 和 Herskovits 提出，如同(2.6)式一樣，不包含懲罰函數。

$$Q_{SB}(D, S) = \log p(D) + \sum_{i=1}^n \sum_{j=0}^{r_i} \sum_{k=1}^{s_i} N_{ijk} \log \frac{N_{ijk}}{N_{ik}} - \frac{1}{2} \text{Dim}(B) \log N \quad (2.8)$$

(2.8)式為標準的貝氏品質衡量係數，其中 $\frac{1}{2} \text{Dim}(B) \log N$ 為懲罰函數。

2. 搜尋演算法

搜尋演算法，利用一套有效率的方法在可能架構空間中，找到一個品質衡量係數最高的貝氏網路。此類演算法可參考，在此介紹 K2 演算法。

Cooper 和 Herskovits 提出 K2 演算法[3][31]。此方法需在四點假設下進行。

- (1) 變數為離散型數值變數。
- (2) 案例間互為獨立。
- (3) 每個變數均有對應值。
- (4) 觀察資料庫之前，貝氏網路架構的機率不重要。

搜尋方法是假設有一組排序後的節點，且開始時網路沒有任何連結。增加每一個節點的父代集合，然後測量其品質衡量係數，品質衡量係數高的成為此節點的父代。當品質係數不再增加時，可得到一個最終的貝氏網路。詳細過程如下。

一組 n 個離散變數集合 Z 。每個變數為 X_i ， $X_i \in Z$ ，有 r_i 個可能的對應變數值 $(v_{i1}, \dots, v_{ir_i})$ 。假設 D 有 m 個案例的資料庫，每一個案例均包含一個對應於 Z 中每個變數的變數值。若 B_s 表示一個貝氏網路可能架構，包含 Z 中的變數。設 w_{ij} 代表 X_i 關於 D 的第 j 個單一對應，而 X_i 有 q_i 個單一對應。假設 N_{ijk} 為案例個數，其中案例是指 D 中 X_i 對應 v_{ik} ， X_j 對應 w_{ij} 的案例。

設 $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ ，則下式為尋找最有可能的網路架構。

$$P(B_s, D) = P(B_s) \prod_{i=1}^n g(i, \pi_i) \quad (2.9)$$

Cooper 和 Herskovits 提出一個貪婪演算法，搜尋給每個變數，使 $g(i, \pi_i)$ 最大的父代集合，為最終的網路架構。此貪婪演算法如下：

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (2.10)$$

K2 演算法需要一組排序後的節點才能建構網路，網路架構的品質對節點的排序相當敏感，此成為 K2 演算法的隱憂。若是排序跟現實問題差異太大，會造成網路間條件獨立無法呈現，造成網路結構相當複雜。簡單架構的網路，會比複雜架構提供更多的資訊。合適的節點排序有助於 K2 演算法架構貝氏網路。K2 演算法的詳細流程如圖 2.2。

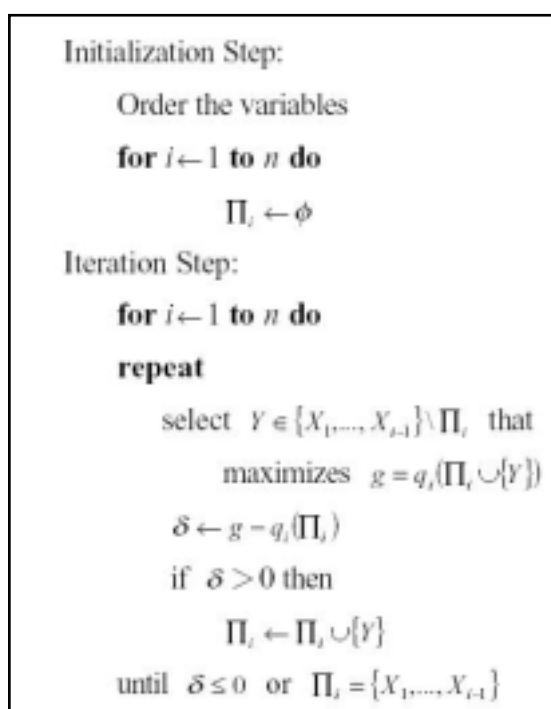


圖 2.2 K2 演算法流程[31]

2.2.2 限制式演算法

條件獨立測試是此類演算法的基礎，藉由歷史資料的統計測試獲得變數之間相依或獨立的關係。要利用條件獨立測試的結果來建構貝氏網路，需要搭配三個假設，分別是 Causal Sufficiency, Causal Markov, Faithfulness，簡述如下[22]：

1. Causal Sufficiency：在領域知識內，不存在沒被觀察到的變數成為已觀察到的變數的父代節點。沒被觀察到的變數，包括隱藏（hidden）和潛在（latent）變數。
2. Causal Markov：任一變數與其所有非子代節點在其父代節點條件下為獨立關係。

3. Faithfulness : 若一個貝氏網路圖形與其資料的機率分配相互精確可靠 (faithful), 條件成立在機率分配中的所有獨立關係, 都可在貝氏網路圖形滿足 Causal Markov 假設下獲得證實。

SGS 為此類演算法中最簡單易懂的, 在本節以 SGS 作此類演算法的介紹。SGS 演算法流程見圖 2. 3。

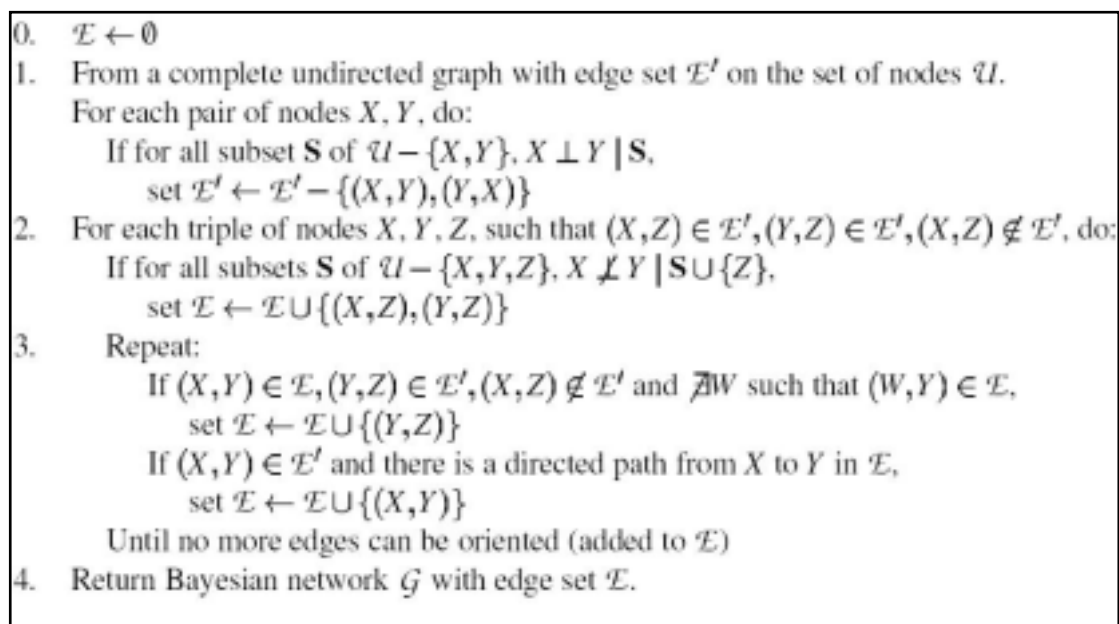


圖 2. 3 SGS 演算流程[22]

SGS 假設兩變數 X, Y 之間存在一連結 (edge), 要經過數次 CI-tests 才能決定。所有連結所成的集合為 E , 最開始的時候節點間均不存在具方向性的連結集合, $E = \emptyset$ 。

步驟一, 所有節點均存在連結, 成為一個完整無方向性圖形 (complete undirected graph)。所有連結構成集合 E' , 所有節點構成集合 U 。取其中一對節點 X, Y 與其他不包含 X, Y 節點所構成的集合 S 進行條件獨立測試。若 X, Y 符合 Faithfulness 假設, 則 X, Y 存在一連結。若 X, Y 之間不存在連結 ($E' - \{(X, Y), (Y, X)\}$), 則 X, Y 會與其他節點的子集合形成 d-separation, 此子集合會以 X, Y 其中之一為子代節點。SGS 測試完所有 $U - \{X, Y\}$ 的 d-separation 可能性, 以及測試其他像 X, Y 這樣的成對節點是否存在連結後, 結束步驟一。

步驟二，決定在步驟一進行後存在連結的方向性。任取三個節點，以 X, Y, Z 為例， X, Z 以及 Y, Z 均存在連結， X, Y 不存在連結。令 $S' \subseteq U - \{X, Y, Z\}$ ， $S = \{Z\} \cup S'$ ，若不存在 S 可以 d-separate X, Y ，則 X, Y, Z 的方向性為 $X \rightarrow Z$ ， $Y \rightarrow Z$ 。此時 $E = E \cup \{(X, Z), (Y, Z)\}$ 。測試完所有三個節點的組合後，結束步驟二。

步驟三，保持圖形為非循環的架構。

所有利用條件獨立測試的限制式演算法，無法必然地給每個連結方向性。例如， $X \rightarrow Y \rightarrow Z$ ， $X \rightarrow Y \leftarrow Z$ ， $X \leftarrow Y \rightarrow Z$ 都是發生在 X 與 Z 在 Y 條件下獨立的情形。

Pearl 和 Verma[28]提出 IC (inductive causation) 演算法，也是利用條件獨立為基礎的限制式演算法。Cheng et al.[10]則是利用相互資訊理論作為條件獨立測試的限制式演算法。此演算法可利用排序好的節點，節省條件獨立測試次數，加快建構時間。

此類演算法有兩個缺點。其一，因為節點沒有經過排序，所以條件獨立測試的先後順序會影響最終建構出來的圖形。以 SGS 演算法為例，若是網路中存在 $X \rightarrow Z \rightarrow Y$ 架構， X 與 Y 在 Z 條件下為相依的情形。但在步驟一時，沒有 $X-Y$ 連結，以後的步驟二或三也不會出現 $Y \rightarrow Z$ 的連結。

其二，條件獨立測試次數會與變數的增加呈指數性成長。例如，SGS 的步驟一，要測試所有 $2^{|S|}$ 組子集合在 S 包含 $n-2$ 個變數。SGS 以及 IC 演算法均有此缺點。TPDA (three-phase dependency analysis) 以及 PC 演算法利用節點排序節省條件獨立測試的次數。

2.2.3 d-separation

網路中的節點是否滿足條件獨立，可由上一節介紹的 CI-tests 來決定，也可以經由 d-separation (direction-dependent separation) 的觀念，直接從網

路中看出滿足條件獨立的節點[29]。

一組節點 E 成為 X 與 Y 這兩組節點的 d -separation，是要所有通過 X 與 Y 的路徑均被 E 所阻礙 (block)。所謂被 E 所阻礙，是要一個節點 Z 滿足下面三個網路情形的前兩種 (如圖 2.4)：

1. Z 屬於 E ，且 Z 是線型 (linear) 網路的中間點。
2. Z 屬於 E ，且 Z 是發散型網路的中間點。
3. Z 或是 Z 的子代節點不屬於 E ，且是收斂型網路的中間點。此時不屬於 d -separation 的情形。

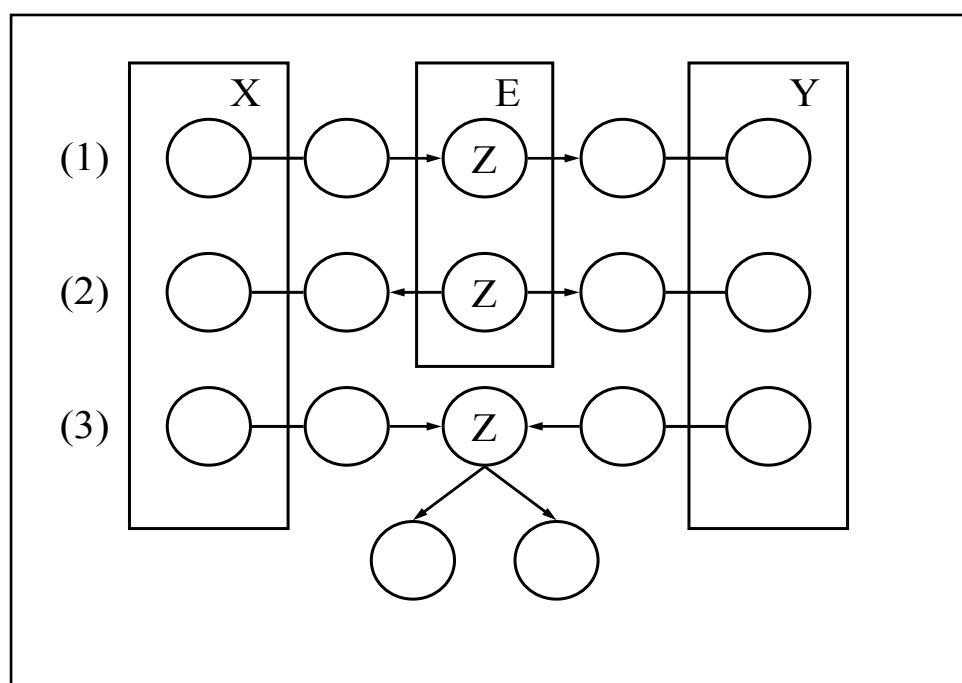


圖 2.4 d -separation 示意圖[29]

以汽車的電力與引擎系統為例，介紹上述三種 d -separation 的情形，如圖 2.5。

1. 知道點火 (節點 Ignition) 是否有問題後，汽車是否有汽油 (節點 Gas) 與音響 (節點 Radio) 是否能開就是獨立的情形。滿足 d -separation 的第一種情形。
2. 知道電池 (節點 Battery) 是否有電後，汽車是否有汽油 (節點 Gas) 與

音響（節點 Radio）是否能開就是獨立的情形。滿足 d-separation 的第二種情形。

3. 其他的節點不再使汽油與音響成為獨立的情形。但當我們知道汽車不能啟動（節點 Starts）時，若發現音響（節點 Radio）能撥音樂，此時會增加沒有汽油（節點 Gas）的可能性。因為此時音響與汽油的關係會因為啟動而變成相依，同樣的情形可用在啟動的子節點移動（節點 Moves）。滿足第三種情形，不構成 d-separation。

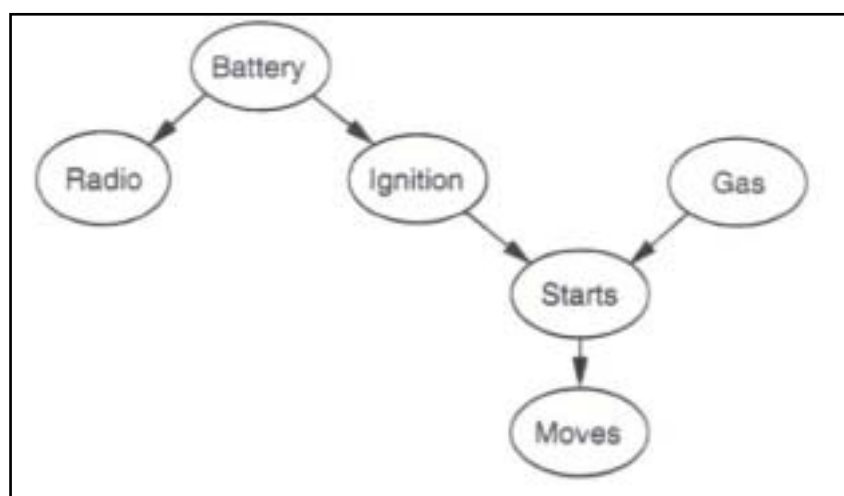


圖 2.5 汽車電子引擎系統的 d-separation 範例[29]

2.3 資料探勘方法

[3]提出以資料探勘演算法中的線性代數方法，從資料中獲得變數的間的相依關係，進一步建構成貝氏網路。

假設資料的變數型態為二元資料，將變數轉換成矩陣 M 。原矩陣透過矩陣相似性轉換，得到一相似矩陣。資料轉換中設計的矩陣與相似矩陣之間具有相同的關係，保留矩陣之列到矩陣之行的關係。相似矩陣也保留原矩陣的特徵值，所以一個矩陣的相似矩陣足以替代原矩陣。

Jordan 標準型[3]是由一個矩陣之特徵值及每個特徵值重複的個數所形成的一個標準型，近似三角矩陣，保留原矩陣的本質。若該矩陣的特徵向

量線性獨立，則 Jordan 標準型為一對角矩陣。Jordan 標準型具有最簡化及唯一性的性質。

由於 Jordan 標準型是由特徵值所構成的一種矩陣型式，若 Jordan 標準型中有特徵值 0 出現，則節點間的關聯可能不存在或是非常微小。以 Jordan 標準型為此方法的轉換矩陣。

進行相似轉換之前，先找出相似轉換矩陣 M 的 Jordan 標準型 J 。求得 Jordan 標準型作為相似性轉換時所需的轉換矩陣，使得 $M=JTJ^{-1}$ ，得到矩陣 M 的相似型。最後，以 $J^{-1}MJ$ 求得相似於矩陣 M 的相似矩陣 T 。由相似矩陣 T 中取得變數間的關聯，並架構出貝氏網路的雛型。

此方法雖然在某些理論驗證時，架構相似度不比利用 K2 演算法來的理想，無法充分呈現領域知識的架構。但是在資料相似度卻與 K2 演算法相同，由此證明變數機率分佈與 K2 演算法相同。因此，資料探勘方法同樣適用於建構貝氏網路。

Ślęzak 利用粗集理論中的 reduct 建構成 reduct network[32][33]。此方法的優點是將決策法則與 information reduct 結合起來，以圖形的方式呈現出決策法則以最佳的方向串聯的結果。

一資訊系統可表示成 $A=(U, A)$ ， U 表示所有無缺值且有限物件(objects) 所成的集合， A 表示所有無缺值且有限屬性所成的集合。對於每一個屬性 $a \in A$ ，都會有一組 V_a ，表示 a 的屬性值。

若有一組屬性的子集合 $B \subseteq A$ ， B 所包含的元素稱為 *B-information function*，表示成下式：

$$\begin{aligned} Inf_B : U &\rightarrow P\left(B \times \bigcup_{a \in B} V_a\right) \\ x &\mapsto \{(a, a(x)) : a \in B\} \end{aligned} \quad (2.11)$$

B 集合內的元素，若是有屬性值均相同的物件，稱作不可辨識關係 (B -indiscernibility relation)，表示成下式：

$$IND(B) = \{(x, y) \in U \times U : Inf_B(x) = Inf_B(y)\} \quad (2.12)$$

粗集理論中的 reduct 是指在選取的屬性集合中，擁有最高且最短的表示成立所構成的集合。information reduct 是指在資訊系統中，最短且足以表達所有屬性所表達的屬性集合。若 B 表示 $A=(U, A)$ 的 information reduct，則滿足下列條件：

$$IND(B) \equiv IND(A) \quad (2.13)$$

information reduct 為資料中最重要的變數集合，在 reduct network 中的必要節點。節點間的關係確定，需靠決策表中的 decision reduct 決定。一決策表可以表示成 $A=(U, A, \{d\})$ ，與資訊系統不一樣的是，多了一樣決策變數 d 。同樣，從決策表中亦可以得到 reduct，稱之為 decision reduct，滿足下式條件：

$$IND(B) \equiv IND(B \cup \{d\}) \quad (2.14)$$

Ślęzak 將 reduct network 定義如下：

(定義：reduct network) 若有一非循環方向圖形 $D=(A, \vec{E})$ 稱作 reduct network，則對於每一屬於 A 的屬性 a 在決策表中得到一 reduct 集合 (a) ， (a) 如下式：

$$\pi(a) = \{b \in A : \text{there is arrow from } b \text{ to } a \text{ in } D\} \quad (2.15)$$

則決策表可表示成： $a=\{U, (a), \{a\}\}$ ，其中 (a) 為下式：

$$\Pi(a) = \{b \in A : \text{there is direct path from } b \text{ to } a \text{ in } D\} \quad (2.16)$$

利用決策表中的 decision reduct 產生的變數關聯，構成一個 reduct network。此結果與貝氏網路最大的差別在於沒有包含機率密度，若是此網路在建構初期即進行變數的因果排序，最終的網路架構會呈現出資料中的因果關係。

本研究方法延續此觀念，配合一些法則篩選的條件，進行貝氏網路的架構。詳細的理論，以及如何利用此理論來進行貝氏網路架構學習，將在第三章中介紹。

第三章 粗集理論與建構貝氏網路

貝氏網路一般建構步驟，可依據 Heckerman 在[19]提出的方法。包含下面四點步驟：

1. 訂定貝氏網路建構目的。
2. 確定所有與問題可能相關的觀測資料。
3. 決定在這些觀測資料下對網路模型有價值的變數。
4. 組織這些變數構成貝氏網路。

本論文假設觀察者已經完成了前三個步驟，而正要利用本研究方法——粗集理論，進行第四個步驟。除此之外，尚包含下面四點假設：

1. 資訊系統中的物件彼此獨立。
2. 每一物件對應到每一變數都有其狀態值。
3. 建構變數間的關聯前，變數間的機率分佈不重要。
4. 有一組依照因果關係排序的變數。

本研究的建構步驟如以下各節所述。先由貝氏網路應用軟體中，挑選出已知的貝氏網路模型作為範例，並產生資料，詳細的介紹在 3.1 節。選取上述資料的適當部分，經過特定的步驟轉換成粗集理論可推論的資料型態。接以利用粗集資料分析，粹取出資料中隱藏的法則，粗集理論與其產生法則將在 3.2 節詳述。

產生法則不是都可以轉換成網路，其中存在著隨機性問題，以及過長的法則會不適合轉換成網路。法則篩選的依據，將在 3.3 節詳述。經過篩選過後的法則，可以轉換成網路，轉換方式在 3.4 節詳述。

建構好的網路雛型，利用剩下未參與建構的資料，透過貝氏網路應用軟體中匯入資料的功能，完成網路最後的參數學習部分，將在 3.5 節詳述此步驟。最後以本研究建構出的網路，與原先的範例網路，在架構與資料

結構的相似性以及關鍵節點上進行績效比較。此外，並利用 K2 演算法為基礎的貝氏網路建構軟體—Bayesware Discovery，一同進行建構績效比較，詳述在 3.6 節。本研究提出的建構進行步驟如圖 3.1。

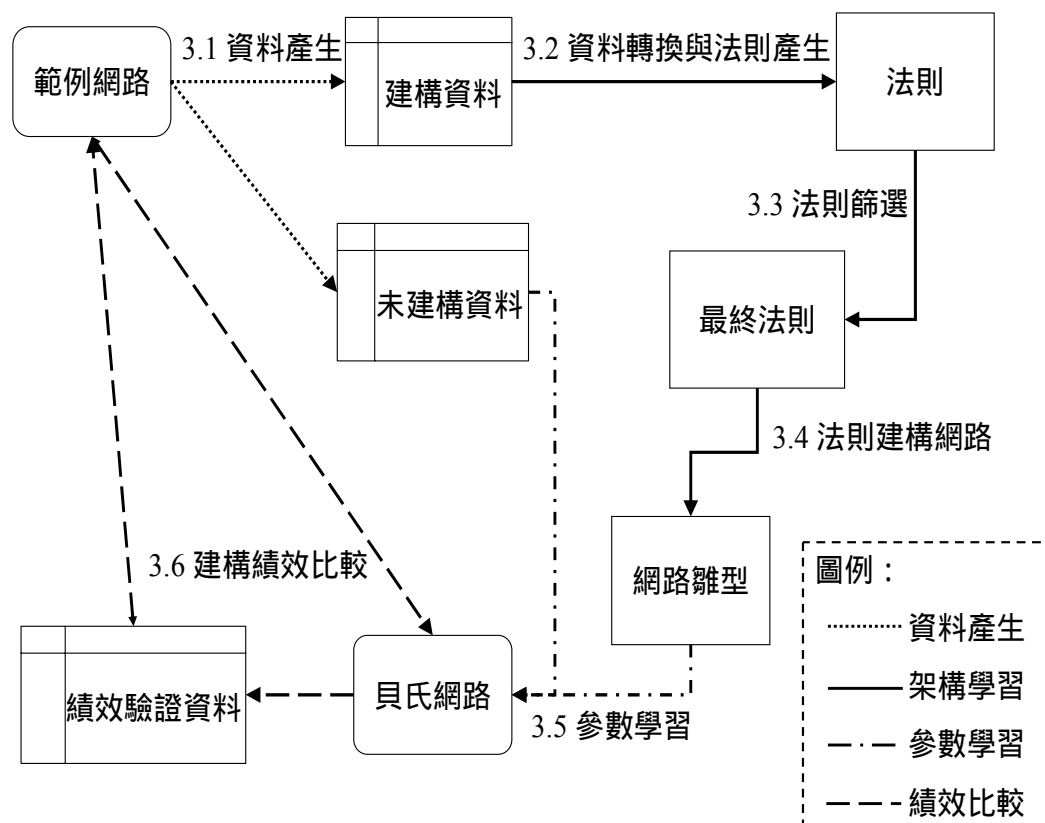


圖 3.1 本研究之貝氏網路建構步驟

3.1 資料產生

雖然貝氏網路可以避免資料過度配適的問題，但是粗集理論對於資料量的多寡會明顯影響到挖掘的品質。

粗集理論使用在小樣本的情形時，可挖掘出非隨機性且具高法則支持度與低法則長度的決策法則。當樣本數量過大時，隨機性 (casual) 的法則就會出現，此類的法則，無助於建構貝氏網路[5][7][14]。關於隨機性、法則支持度和法則長度等法則篩選衡量指標，將在 3.2 與 3.3 節詳述。

本研究對於每個範例網路模型，產生 400 筆資料。而利用其中一半的

資料建構貝氏網路，另外一半進行參數學習。

3.2 資料轉換與法則產生

粗集理論適用的變數型態，包括定性變數和定量變數兩種[34]。定性變數為非數值型資料型態，包含名目變數（nominal variables）或順序變數（ordinal variables）等。定量變數為數值型資料型態，包括連續型數值和離散型數值兩種。

名目變數的狀態值過多，會造成粗集理論推論時的困難。例如，某問題在分析時包含顏色這變數，而顏色具有七個狀態值，紅、橙、黃、綠、藍、靛、紫。在進行推論之前，應考慮此變數是否會因狀態值過多而失去其在問題中的代表性。若是，則可將顏色進行進一步的區分。例如分成暖色或寒色（此時顏色變為兩個狀態值），或者其他的方法讓狀態值的數目減少。而在進行轉換時，必須考慮轉換過程對問題的適切性，減少因為轉換而造成變數對問題表達有失真的情形。

連續型數值變數會因為數值分佈過廣，必須經過切割才能進行推論。切割連續型數值變數方法眾多，可以等距切割、等量切割、群集或者 3-4-5 法則等[18]。有些連續型數值變數，與問題間會有關鍵範圍的存在。例如，研究血壓對人的健康影響時，可以依據高血壓（收縮壓高於 140 毫米汞柱，舒張壓高於 90 毫米汞柱，低血壓（收縮壓低於 100 毫米汞柱，舒張壓低於 60 毫米汞柱），與正常血壓（兩值之間），將血壓分為三個範圍。

經過適當處理過後的資料，方可利用粗集理論進行推論。下面兩小節，先介紹粗集理論基本概念和模型。在理論基本的認識後，接著描述粗集理論如何經由資料分析，產生我們用來建構貝氏網路所需要的法則。

3.2.1 粗集理論基本概念

Pawlak 在 1985 提出粗集理論（Rough Set Theory）[25][26][34]。與其他人工智慧演算法最大的差異，粗集理論只需從資料面便可以進行分類，

不需要其他函數輔助。

粗集理論進行資料分析時，需仰賴資訊系統提供資料，如表 3.1。一資訊系統(information system)表示成 $I=(U, \mathcal{Q}, V_q, f_q)$ ，各元素意義如下[16]：

1. U 表示一組有限物件。
2. \mathcal{Q} 表示一組有限的屬性 (或稱為特徵或變數)。
3. 對於在屬性集合 \mathcal{Q} 中的每個屬性 q 。
 - (1) V_q 表示屬性 q 的一組屬性值。
 - (2) 一資訊函數 $f_q : U \rightarrow V_q$ 。資訊系統中的每一物件，在每個屬性都有其屬性值。

表 3.1 資訊系統

object	A	B	C	D
1	1	1	1	1
2	2	1	2	2
3	2	1	2	1
4	1	2	2	1
5	1	1	1	1

粗集理論基本觀念，是將資訊系統中的物件視為一個個集合。若是在單一個或是一組屬性下，不同物件擁有相同的屬性值，則會被歸於同一個集合。此時集合中的物件具有不可辨識關係 (indiscernibility relation) 或對等關係 (equivalence relation)。

假設 Q 表示屬性所成的集合， $Q \in \Omega$ ， θ_Q 表示不同物件在 Q 中的對等關係，定義成下式。

$$x \equiv_{\theta_Q} y \stackrel{\text{def}}{\Leftrightarrow} (\forall q \in Q) f_q(x) = f_q(y) \quad (3.1)$$

例如，表 3.1 中對等關係集合如表 3.2。

表 3.2 對等關係集合

屬性(Q)	對等關係(θ_Q)
A	{1,4,5}, {2,3}
B	{1,2,3,5}, {4}
C	{1,5}, {2,3,4}
D	{1,3,4,5}, {2}

粗集理論進行近似分類時有兩個基本概念，近似下界（lower approximation）與近似上界（upper approximation）。分別代表的意義如下 [13][25][34]：

1. 近似下界：在全集 U 中選取部份集合 Y ， $Y \subseteq U$ 。若是 Y 中物件的對等關係集合，均能被 Y 包含，則稱這些物件為近似下界。近似下界中物件均能明確表達集合 Y 的特性。此區域也稱為 positive region，符號表示為 $POS_X(Y)$ 。近似下界表示如下式。

$$\underline{Y}^{\theta} = \{x \in U : \theta_Q x \subseteq Y\} \quad (3.2)$$

2. 近似上界：在全集 U 中選取部份集合 Y ， $Y \subseteq U$ 。若是 Y 中物件的對等關係集合，與 Y 交集不為空集合，則稱這些物件為近似上界。近似上界中只有部分物件可能表達集合 Y 的特性。此區域也稱為 possible region。近似上界表示如下式。

$$\overline{Y}^{\theta} = \{x \in U : \theta_Q x \cap Y \neq \emptyset\} \quad (3.3)$$

如圖 3.2。近似下界包含最內圍淡灰色部分，近似上界則包含所有灰色部分。近似上界扣掉近似下界的部分，也就是深灰色的部分，為 Y 的不確定區域（uncertainty region），稱為近似界限（approximation boundary），符號為 $BN_Y(X)$ 。

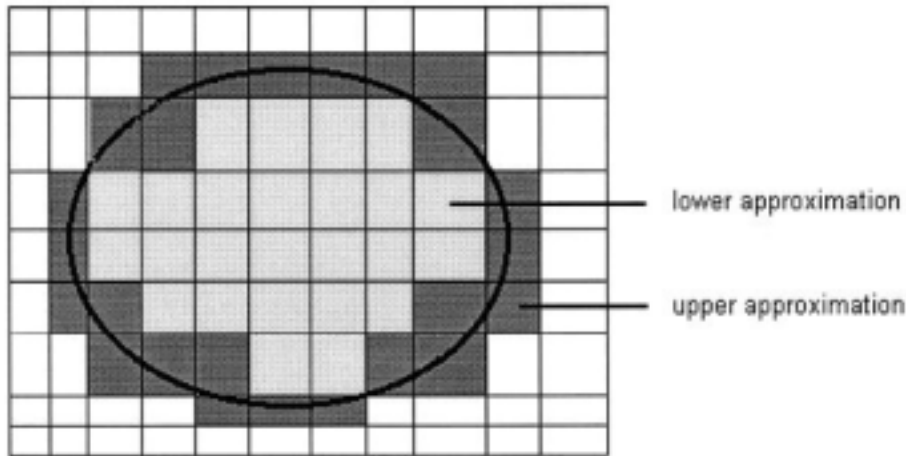


圖 3.2 粗集近似空間[34]

例如對等關係集合 $QX_1=\{1,4,8\}$, $QX_2=\{2,5,7\}$, $QX_3=\{3\}$, $QX_4=\{6\}$ 。選取 $Y=\{3,6,8\}$, 則近似上界與近似下界分別如下。

$$\underline{Y}^o = \{3,6\}$$

$$\bar{Y}^o = \{1,3,4,6,8\}$$

$$BN_Y(X) = \{1,4,8\}$$

粗集理論在進行近似推論時，有兩個近似衡量函數，分別是係數與係數[25][34]。簡述如下：

$$\gamma_Y(X) = \frac{|X|}{|U|} \quad (3.4)$$

函數，近似分類的品質。表示可以對知識 X 正確分類的比率。X Y，表示屬性 X 對屬性 Y 的相依性（dependency），或者是一致性（consistency）程度。當 =1，表示 X Y 是一致性。當 1，表示 X Y 是非一致性，也是不確定性（uncertainty）。

$$\alpha_Y(X) = \frac{|X|}{|X|} \quad (3.5)$$

函數，近似分類的準確性。表示知識 X 的完整性。當 =1，表示 X 對 Y 而言是確定的（deterministic）。當 1，表示 X 對 Y 而言是不確定

的。

當一資訊系統中的屬性，區分為條件屬性以及決策屬性時，資訊系統可稱為決策表 (decision table)。

假設 P, Q 分別決策表中的條件屬性與決策屬性, R 為 P 中的一組集合, 若 R 滿足下式, 則稱為在 P 中為 Q-indispensable[25]。

$$POS_{P-\{R\}}(Q) \neq POS_P(Q) \quad (3.6)$$

若在 P 中的每個 R 集合都是 Q-indispensable, 則稱 P 是 Q-independent。 $S \subseteq P$, 若稱 S 為 P 的一組 Q-reduct, 則 P 必須是 Q-independent, 且 S 滿足下式:

$$POS_S(Q) = POS_P(Q) \quad (3.7)$$

所有 Q-reduct 的交集, 稱為 Q-core。 Q-core 代表與 Q 關係最強烈的屬性。

粗集理論藉由 reduct 求得在決策表中, 能與決策屬性保持一致性的條件屬性。這些條件屬性與決策屬性構成確定性法則 (deterministic rules)。符號記為 $P \xrightarrow{\text{det}} Q$, 或 $P \Rightarrow Q$ 。

例如, 一決策表如表 3.3。 $R=\{A,B,C\}$ 。依據上述推論方法, 可獲得確定性法則。過程如下。

表 3.3 決策表 A

objects	A	B	C	Q
1	1	1	1	1
2	2	2	2	3
3	1	1	3	2
4	1	1	3	2
5	1	1	1	1
6	1	2	1	1
7	1	2	2	3
8	2	2	2	4

1. $POS_{(R-\{A\})}(Q) = \{1,3,4,5,6\} \neq POS_R(Q)$, A 是 Q-indispensable。
2. $POS_{(R-\{B\})}(Q) = \{1,3,4,5,6,7\} = POS_R(Q)$, B 不是 Q-indispensable。
3. $POS_{(R-\{C\})}(Q) = \emptyset \neq POS_R(Q)$, C 是 Q-indispensable。
4. 因為 $R-\{B\}=\{A,C\}$, 所以 $POS_{(R-\{B\})}(Q) = POS_{\{A,C\}}(Q) = POS_R(Q)$ 。
5. 因此 $\{A,C\}$ 為 Q-reduct , 也是 Q-core。產生下面法則。

$$\begin{aligned}
 A_1 \wedge C_1 &\Rightarrow Q_1 \\
 A_2 \wedge C_2 &\Rightarrow Q_3 \\
 A_1 \wedge C_3 &\Rightarrow Q_2 \\
 A_1 \wedge C_2 &\Rightarrow Q_3 \\
 A_2 \wedge C_2 &\Rightarrow Q_4
 \end{aligned}$$

一般粗集理論應用軟體，不是依照上述推論的方式來產生法則。而是利用可辨識矩陣(discernibility matrix)來計算決策表中的 reduct 以及 core，並粹取法則[34]。

可辨識矩陣是決策表中物件與物件間形成的矩陣，若物件數為 n，可辨識矩陣為 $n \times n$ 矩陣。矩陣內元素 (C_{ij})，利用比較每個物件之間狀態的異同來決定。若兩物件具有相同的決策屬性，則在可辨識矩陣中為空集合。若兩物件不具有相同的決策屬性，則在可辨識矩陣中表示出不同狀態值的條件屬性。如下式[25]。

$$c_{ij} = \begin{cases} \{a \in \Omega : a(x_i) \neq a(x_j)\} & i, j = 1, 2, \dots, n \\ \emptyset, & \text{otherwise} \end{cases} \quad (3.8)$$

例如，一決策表如表 3.4。a, b, c 為條件屬性，d 為決策屬性。

表 3.4 決策表 B

object	a	b	c	d
1	2	1	3	1
2	3	2	1	2
3	2	1	3	1
4	2	2	3	2
5	1	1	4	3
6	1	1	2	3
7	3	2	1	2
8	1	1	4	3
9	2	1	3	1
10	3	2	1	2

可辨識矩陣為一三角矩陣，只需採用上三角矩陣或下三角矩陣來進行計算。先比較物件與物件間的決策屬性的狀態值，接著比較條件屬性的狀態值。例如 1, 3 兩物件有相同的決策屬性狀態值，所以在可辨識矩陣內的元素為空集。1, 2 兩物件決策屬性狀態值不相同，條件屬性 a, b, c 也不相同。所以在可變數矩陣中的元素為 a, b, c。比較此 10 個物件的屬性狀態值，獲得表 3.5 的 10×10 可辨識矩陣。

表 3.5 可辨識矩陣

	1	2	3	4	5	6	7	8	9	10
1	-									
2	a,b,c	-								
3	-	a,b,c	-							
4	b	-	b	-						
5	a,c	a,b,c	a,c	a,b,c	-					
6	a,c	a,b,c	a,c	a,b,c	-	-				
7	a,b,c	-	a,b,c	-	a,b,c	a,b,c	-			
8	a,c	a,b,c	a,c	a,b,c	-	-	a,b,c	-		
9	-	a,b,c	-	b	a,c	a,c	a,b,c	a,c	-	
10	a,b,c	-	a,b,c	-	a,b,c	a,b,c	-	a,b,c	a,b,c	-

可辨識矩陣中的元素內狀態值為聯集，元數間的狀態值為交集。忽略空集的元素，可辨識矩陣轉變為可辨識函數。如下。

$$\begin{aligned}
 f(D) &= (a+b+c)b(a+c)(a+c)(a+b+c)(a+c)(a+b+c)(a+b+c)(a+b+c) \\
 &\quad \times (a+b+c)(a+b+c)(a+b+c)b(a+c)(a+c)(a+b+c)(a+c)(a+b+c) \\
 &\quad \times (a+b+c)(a+b+c)(a+b+c)b(a+c)(a+b+c)(a+c)(a+b+c)(a+b+c) \\
 &\quad \times (a+c)(a+b+c)(a+b+c)(a+b+c)(a+c)(a+b+c)(a+b+c) \\
 &= b(a+c) = ab+bc
 \end{aligned}$$

得到兩個 D-reduct，{a, b}，{b, c}，和一個 D-core，{b}。取 reduct {a, b}，表 3.5 簡化成表 3.6。

表 3.6 reduct {a, b} 決策表

object	a	b	d
1	2	1	1
2	3	2	2
3	2	1	1
4	2	2	2
5	1	1	3
6	1	1	3
7	3	2	2
8	1	1	3
9	2	1	1
10	3	2	2

表 3.6 中，繼續進行可辨識矩陣計算，去除不必要的條件屬性狀態值，如表 3.7。此時的可辨識矩陣不是一個三角矩陣。

表 3. 7 reduct {a, b} 可辨識矩陣

	1	2	3	4	5	6	7	8	9	10
1	-	a,b	-	b	a	a	a,b	a	-	a,b
2	a,b	-	a,b	-	a,b	a,b	-	a,b	a,b	-
3	-	a,b	-	b	a	a	a,b	a	-	a,b
4	b	-	b	-	a,b	a,b	-	a,b	b	-
5	a	a,b	a	a,b	-	-	a,b	-	a	a,b
6	a	a,b	a	a,b	-	-	a,b	-	a	a,b
7	a,b	-	a,b	-	a,b	a,b	-	a,b	a,b	-
8	a	a,b	a	a,b	-	-	a,b	-	a	a,b
9	-	a,b	-	b	a	a	a,b	a	-	a,b
10	a,b	-	a,b	-	a,b	a,b	-	a,b	a,b	-

計算每一行元素或列元素的可辨識函數，如下。

$$f_1(d) = (a+b)baa(a+b)a(a+b) = ab$$

$$f_2(d) = (a+b)(a+b)(a+b)(a+b)(a+b)(a+b) = a+b$$

$$f_3(d) = (a+b)baa(a+b)a(a+b) = ab$$

$$f_4(d) = bb(a+b)(a+b)(a+b)b = b$$

$$f_5(d) = a(a+b)a(a+b)(a+b)a(a+b) = a$$

$$f_6(d) = a(a+b)a(a+b)(a+b)a(a+b) = a$$

$$f_7(d) = (a+b)(a+b)(a+b)(a+b)(a+b)(a+b) = a+b$$

$$f_8(d) = a(a+b)a(a+b)(a+b)a(a+b) = a$$

$$f_9(d) = (a+b)baa(a+b)a(a+b) = ab$$

$$f_{10}(d) = (a+b)(a+b)(a+b)(a+b)(a+b)(a+b) = a+b$$

可辨識函數的結果中，聯集的部分取 core 屬性來表示，也就是說 a+b 取 b，忽略 a。得到最終的 reduct {a, b} 的決策表。如表 3. 8。

表 3. 8 最終 reduct {a, b} 決策表

object	a	b	d
1	2	1	1
2	*	2	2
3	2	1	1
4	*	2	2
5	1	*	3
6	1	*	3
7	*	2	2
8	1	*	3
9	2	1	1
10	*	2	2

註：*表示忽略

表 3. 8 中表示的就是最終的分類法則，如下。

$$a_2, b_1 \Rightarrow d_1$$

$$b_2 \Rightarrow d_2$$

$$a_1 \Rightarrow d_3$$

從上述可知，當決策表中的物件增加 n 個時，可辨識矩陣大小增加 n^2 。所以物件個數越多，計算速度就會越久。而物件數量，也會受到粗集理論應用軟體中可辨識矩陣的大小所限制。

3.2.2 挖掘法則

本研究要建構貝氏網路的法則必須要法則長度短且支持度高的法則，才能得到結構簡單且可信的貝氏網路。要獲得此種條件的法則，必須要改變係數和訂定容錯值[6][23]。

係數為 1 時，得到的 reduct 是一致的。係數不為 1 時，得到的 reduct 是非一致性的。reduct 非一致性的狀況下，長度會比一致性的狀況下來的短。

近似界線中的物件是不確定性，倘若在近似界線中定一個容錯值，則滿足此容錯值的物件便可以產生不確定性法則。此時的不確定性法則會擁

有比確定性法則更高的法則支持度。

訂定容錯值的方法，利用關聯法則中的兩個衡量指標之一，法則信賴度 (confidence)。法則信賴度為 1 時，表示為確定性法則。法則信賴度不為 1 時，表示為不確定法則。法則信賴度如下式。

$$Confidence(A \Rightarrow B) = \frac{\#_tuples_containing_A_and_B}{\#_tuples_containing_A} \quad (3.9)$$

例如，a, b 為決策表中的條件屬性，d 為決策屬性。在決策表中滿足 a=1, b=1 的物件共有 20 個，而滿足法則 $a_1, b_1 \rightarrow d_1$ 有 18 個物件，滿足 $a_1, b_1 \rightarrow d_2$ 有 2 個物件。則法則信賴度為 $confidence(a_1, b_1 \rightarrow d_1) = 18/20 = 0.9$ 。

假設 P 為決策表中一組條件屬性，Q 為決策屬性，則容錯值如下式[23]，

$$error = \begin{cases} 1 - \frac{\#((\bigcap_{p \in P} \theta_p) \cap \theta_Q)}{\#(\bigcap_{p \in P} \theta_p)}, & \#(\bigcap_{p \in P} \theta_p) > 0 \\ 0, & \#(\bigcap_{p \in P} \theta_p) = 0 \end{cases} \quad (3.10)$$

其中， θ_Q 表示決策屬性 Q 的對等關係集合， θ_p 表示條件屬性 P 的對等關係集合。

例如，一決策表如表 3.9。a, b, i 為條件屬性，Q 為決策屬性。

表 3.9 決策表 C

object	a	b	c	d	e	f	g	h	i	Q
1	0	6	1	1	1	2	2	0	1	1
2	0	6	0	1	1	1	2	1	1	1
3	0	6	0	1	1	2	2	1	1	1
4	0	4	1	1	1	2	2	1	0	2
5	0	6	0	1	1	1	1	1	1	1
6	0	6	0	2	1	1	1	0	2	0
7	0	6	0	1	1	1	2	1	2	0
8	1	4	0	2	0	2	0	1	0	2
9	0	4	0	2	0	2	0	1	1	1
10	0	4	0	2	0	2	1	0	1	1
11	1	4	0	1	0	2	0	1	0	2
12	1	4	0	1	1	1	1	1	1	2
13	0	4	0	2	1	1	1	1	1	1
14	1	4	1	1	0	2	2	1	1	2
15	1	4	0	2	0	1	0	1	1	2
16	0	4	1	1	1	1	2	1	1	1
17	0	6	0	1	1	1	2	0	1	1
18	0	4	0	1	1	1	2	0	1	1
19	1	4	0	1	0	2	1	1	1	2
20	0	4	0	1	0	2	1	1	1	2
21	0	4	0	2	0	2	1	1	1	1

若訂定 係數為 1，則得到一組最短的 reduct {a, d, e, i}。產生的決策法則如表 3.10。

表 3.10 reduct {a, d, e, i} 決策法則

rules	a	d	e	i	Q	support
rule1	*	*	*	2	0	2
rule2	0	*	1	1	1	8
rule3	0	2	*	1	1	4
rule4	*	*	*	0	2	3
rule5	1	*	*	*	2	6
rule6	*	1	0	*	2	4

註：*表示忽略

若訂定最低的 係數為 0.4，則得到一組最短的 reduct {a, i}。若容錯值為 0.2，則產生決策法則如表 3.11。

表 3. 11 reduct{a, i} 決策法則

rules	a	i	Q	support
rule1	*	2	0	2
rule2	0	1	1	11
rule3	*	0	2	3
rule4	1	*	2	6

註：*表示忽略

法則 $a_0, i_1 \rightarrow Q_1$ 的信賴度 $\text{confidence}(a_0, i_1 \rightarrow Q_1) = 11/12 = 0.9167$ 。雖然這是不確定性法則，但其法則支持度和法則長度都要比表 3. 10 中的確定性法則來的高。因此，挖掘法則時，可由此方法，獲得適合於建構貝氏網路的決策法則。

本步驟要在一組由因果排序的變數下才能進行。排列順序是由因向果的方向排列。由此排序的變數，從最後一個變數當作粗集理論中的決策屬性，其餘變數當作條件變數，進行決策法則的挖掘。當法則粹取完成後，此變數便不再進行往後的挖掘法則。之後，由最後一個變數往前遞推，依序當作決策屬性。直到所有變數均完成挖掘法則，此步驟才結束。

3.3 法則篩選

經由上述步驟後，會得到相當多的法則。這些法則不是都適合建構成網路，必須要進一步的篩選，方能得到最適合的建構法則。進一步篩選的依據包括三點，法則支持度、法則長度以及顯著性檢定。

3.3.1 法則支持度

法則支持度，是關聯法則兩個衡量指標之一。粗集理論的決策法則，也能視為是一種關聯法則，所以也可以適用此指標。此指標的意義，是說明這一關聯法則出現在所有案例中的比例。當法則支持度越高，表示越有可能相信在這資料中隱藏有此現象。

法則支持度定義如下式[18]：

$$Support(A \Rightarrow B) = \frac{\# \text{ tuples containing both } A \text{ and } B}{total \# \text{ of tuples}} \quad (3.8)$$

例如，某店的銷售紀錄中，產生買電腦就會買應用軟體的法則。若同時買電腦和應用軟體在銷售紀錄中有 80 筆紀錄，全部的銷售紀錄為 100 筆。此法則支持度為 $Support(\text{電腦} \Rightarrow \text{應用軟體}) = 80/100 = 0.8$ 。

建構貝氏網路時，訂定最低法則支持度為 0.1。也就是說 200 筆建構資料中，法則要在 20 比以上的資料中出現。

3.3.2 法則長度

法則長度，是指前面條件屬性的個數。若法則如 $A_1 \wedge B_1 \rightarrow C_1$ ，則法則長度為 2。

法則長度與建構網路時的因果關聯有關。條件屬性可以轉換成網路中的因，也就是父代節點。決策屬性可以轉換成網路中的果，也就是子代節點。

如果法則長度過長，則因果關係會變得複雜，轉換成貝式網路時的關聯可能性也會變多。因為在某一法則中，同樣為父代節點的屬性，也有可能是在別的法則中成為父代和子代的因果關係。

建構貝氏網路時，為使網路架構能簡單，所以法則長度訂為以 2 為上限值。

3.3.3 顯著性檢定

隨機性法則，是指此法則的變數組合只會隨機性出現，也就是說此法則出現機率在所有變數組合中的機率會低於顯著水準。

倘若要檢驗法則 $Q \rightarrow d$ 是否為隨機性法則，假設所有物件的排列集合為

，對於每一個 $\sigma \in \Sigma$ ，則可訂定一特徵向量 $\overline{x_\sigma}^\Omega$ ，如下式[14][17]：

$$x_\sigma^r \stackrel{def}{=} \begin{cases} \sigma(x)^d, & \text{if } r = d \\ x^r, & \text{otherwise} \end{cases} \quad (3.9)$$

依照排列 x^d 的值，產生所有法則排列 $Q(d)$ 。所有 $Q(d)$ 法則的係數分佈為 $\{Q(d) : \sigma \in \Sigma\}$ 。要檢定法則 $Q(d)$ ，假設 H_0 ：此法則為隨機性， H_1 ：此法則不為隨機性。統計量如下式。

$$p(\gamma(Q \rightarrow d) | H_0) := \frac{|\{\sigma \in \Sigma : \gamma(Q \rightarrow \sigma(d)) \geq \gamma(Q \rightarrow d)\}|}{|U|!} \quad (3.10)$$

假設顯著水準為 0.05，單尾檢定。若得到的 $p(Q(d) | H_0)$ 低於顯著水準，則表示否定 H_0 ，此法則是隨機性。反之，則相性此法則不是隨機性，可以用來建構貝氏網路。

3.4 法則建構網路

將這些由法則轉換成的關聯，依照節點的因果先後順序，建構成網路。建構法則有兩種情形，法則長度為 1 和法則長度為 2 兩種情形。

法則長度為 1 時，條件屬性為關聯中的父代節點，決策屬性為關聯中的子代節點。如圖 3.3。

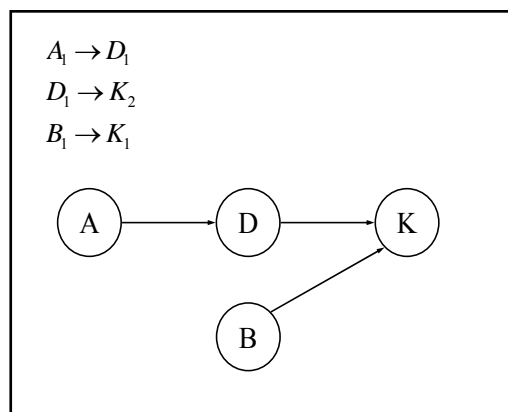


圖 3.3 法則長度 1 時可能之貝氏網路架構

法則長度為 2 時，條件屬性為關聯中的父代節點，決策屬性為關聯中的子代節點。若條件屬性中的兩屬性，也有法則存在時，則構成練狀結構。若條件屬性中的兩屬性，沒有法則存在時，則構成收斂結構。如圖 3.4，左圖為練狀架構，右圖為收斂架構。

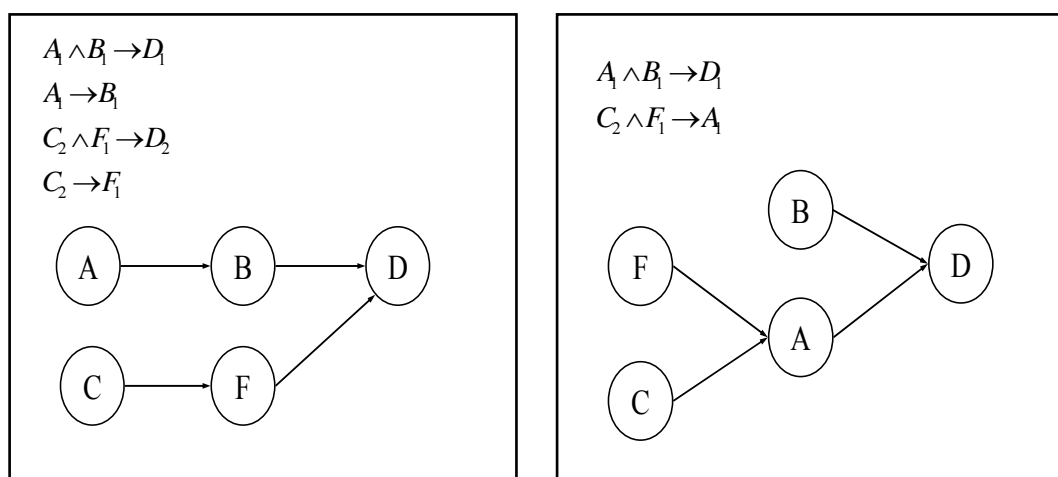


圖 3.4 法則長度 2 時可能之貝氏網路架構

當法則建構成網路時，若任三個節點出現完整圖形時，如圖 3.5。則會存在條件獨立的情形，此時計算子節點與父節點之間 conditional entropy 值（或稱為 unexplained information 值）的大小，去除 conditional entropy 值較低的關聯，以簡化貝氏網路。

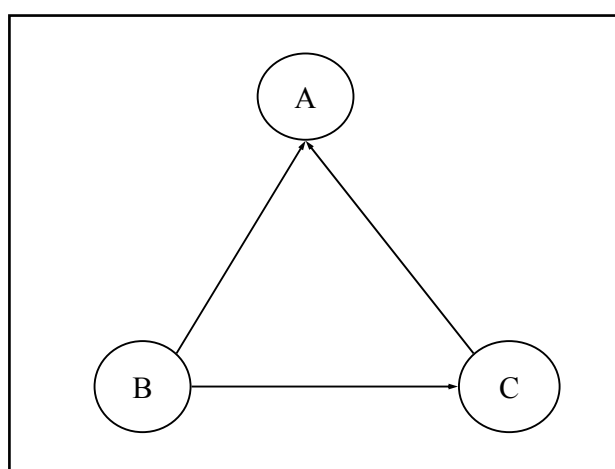


圖 3.5 完整圖形範例

若 X_i 表示條件屬性 Q 的對等關係集合 Q 內的一元素， Y_j 表示決策屬性 d 的對等關係集合 d 內的一元素。則 r_i 表示 X_i 元素內的物件個數， $r_i = |X_i|$ 。向量 $\hat{\pi}_i$ 表示 r_i 在所有物件中的比例， $\hat{\pi}_i = r_i / n$ 。向量 $\hat{\eta}_{i,j}$ 為 X_i 與 Y_j 交集的物件在 X_i 中的比例，表示為 $\hat{\eta}_{i,j} = |X_i \cap Y_j| / r_i$ 。則 conditional entropy 函數如下[13]：

$$H(d|D) \stackrel{def}{=} \sum_{i=c+1}^t \hat{\pi}_i \sum_{j \leq s} \hat{\eta}_{i,j} \log_2 \left(\frac{1}{\hat{\eta}_{i,j}} \right) \quad (3.11)$$

標準化後的函數如下：

$$S(d|Q) = 1 - \frac{H(d|Q)}{\log_2(n)} \quad (3.12)$$

$S(d|Q)$ 值越接近 1，表示從 Q 傳遞到 d 資訊量越多，兩者間的關聯越強。反之， $S(d|Q)$ 值越接近 0，表示從 Q 傳遞到 d 資訊量越少，兩者之間的關聯越弱。關聯弱的連結因該被去除。

3.5 參數學習

貝氏網路中節點間的關聯和機率分配，使得貝氏網路對問題的分析上，同時具有定量與定性的能力。

上述的步驟完成後，只完成貝氏網路的架構部分，也只能提供觀察者定性分析。雖然本研究不包含貝氏網路參數學習，但仍可藉由軟體來完成參數學習，使得最後貝氏網路具有完整的分析能力。

本研究利用在最初資料產生時，另外一半未參與建構的資料，作為參數學習之用。

3.6 建構績效比較

此步驟是為了比較本研究所架構出的貝氏網路，與範例貝氏網路間的相似情形。以三部分進行比較，分別是架構相似性、資料結構相似性與關鍵節點相似性。

3.6.1 架構相似性

架構相似性，是針對貝氏網路模型上節點之間的關聯來做比較。本研究採用 Jaccard 係數作為衡量方法[3][30]。

兩貝氏網路模型 A, B，分別包含關聯所成的集合 a, b。a, b 集合中，關聯的個數為 x, y，具有共同關聯的個數為 m。則 Jaccard 係數為 $m/(x+y-m)$ 。計算範例如下。

如圖 3.6, 貝氏網路模型 A, 包含關聯集合 $a=\{(P, Q), (P, R), (P, S), (P, T), (Q, S), (Q, T), (R, T), (S, T)\}$, $x=8$ 貝氏網路模型 B, 包含關聯集合 $b=\{(P, R), (P, S), (P, T), (Q, S), (Q, T), (R, T), (S, T)\}$, $y=7$ 。A, B 共同關聯集合 $\{(P, R), (P, S), (P, T), (Q, S), (Q, T), (R, T), (S, T)\}$, $m=7$ 。因此, 貝氏網路模型 B 的 Jaccard 係數為 $7/(8+7-7)=0.875=87.5\%$ 。

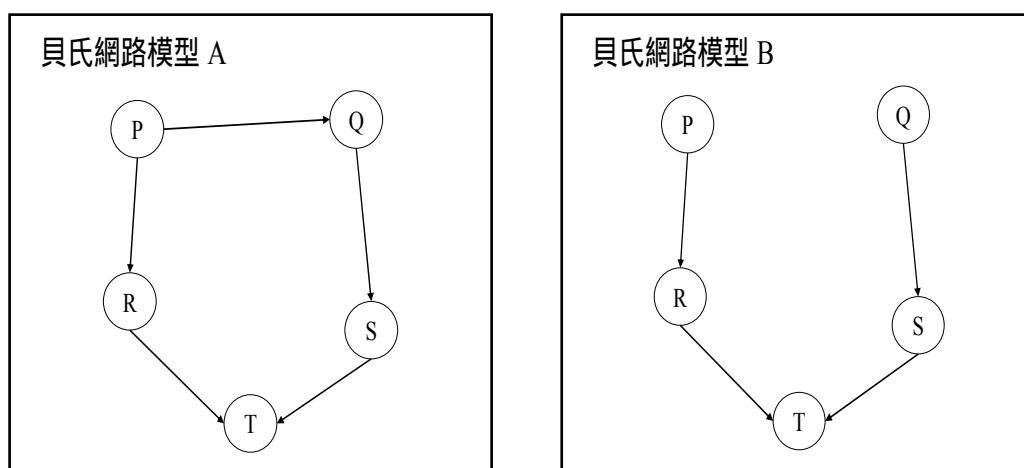


圖 3.6 貝氏網路模型

3.6.2 資料結構相似性

[3]資料結構相似性，將原始資料與最終建構成的貝氏網路產生的新資

料進行比對。假設這兩個資料都是來自同一母體，樣本是成對的。採用魏克森符號等級檢定 (Wilcoxon sign-rank test) 執行資料相似性檢定。

[1]魏克森符號等級檢定，先計算成對樣本觀察值的差異。若兩母體的觀察值分別是 X_1, X_2 ，則觀察值差異為 $D = X_1 - X_2$ 。去除無差異的樣本，然後取絕對值，由小到大排列並賦予等級。若 D 值相同，以平均值為其等級。以 T^+ 表示為正的 D 值等級集合， T^- 表示為負的 D 值等級集合。若兩母體分配相同，則 T^+ 與 T^- 應相同。假設檢定如下：

$$H_0 : T^+ - T^- = 0$$

$$H_1 : T^+ - T^- \neq 0$$

選定 T^+ 或 T^- 值的作為檢定統計量。當樣本大於 30 時， T 值趨近於常態分配。檢定統計量如下：

1. 平均值。

$$E(T) = \frac{n(n+1)}{4} \quad (3.13)$$

2. 標準差。

$$\sigma_T = \sqrt{\frac{n(n+1)(2n+1)}{24}} \quad (3.14)$$

3. 標準化。

$$z = \frac{T - E(T)}{\sigma_{T^+}} \quad (3.15)$$

以顯著水準 0.05，雙尾檢定。若 p 值小於 0.025 則否定 H_0 ，表示新資料與原始資料來自不同母體。反之，不否定 H_0 ，表示新資料與原始資料來自同一母體。

3.6.3 關鍵節點相似性

關鍵節點，適用在機器設備故障診斷。利用架構好的貝氏網路，鎖定最終決策節點為故障的情形時，追溯上面父代節點的機率變化。若父代節點也出現故障狀態機率高於正常狀態機率，則此父代節點稱為關鍵節點。

除了能用在機器設備診斷外，也能用在其他問題的貝氏網路。貝氏網路有一種特點，最終的子代節點，會有某一狀態機率值偏高的情況。當鎖定機率較低的狀態為出現時，則可推論出貝氏網路中的關鍵節點。

第四章 實證研究

本研究的實證部分，採用三個範例。依序是 4.1 節的警鈴、4.2 節的動物分類和 4.3 節的汽車故障診斷。將相同的資料，再以 K2 演算法進行貝氏網路建構，作為本研究的績效比較對象。4.4 節將對本實證研究的結果做一總結。

4.1 範例一：警鈴

此範例為 Judea Pearl 所設計[24]。網路包含五個二元狀態值節點，Burglary, Earthquakes, Alarm, John, Mary。發生竊盜或是地震都會促動警鈴。警鈴響時，John 和 Mary 都會通知你。但 John 可能會將警鈴聲聽成電話聲，Mary 可能因音樂太大聲而聽不到警鈴響，而沒有通知你。貝氏網路架構如圖 4.1。

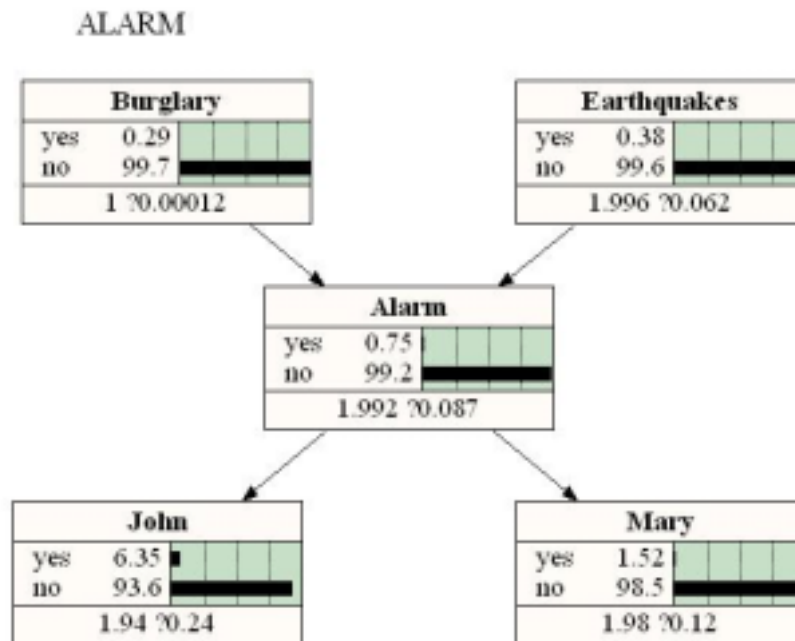


圖 4.1 警鈴之貝氏網路

4.1.1 粗集方法建構之警鈴貝氏網路

依據第三章的步驟，建構成貝氏網路，詳細的進行步驟如下：

1. 利用 Norsys Netica 產生 400 筆資料[24]。並用其中 200 筆資料進行網路建構，剩下的 200 筆資料進行參數學習。
2. 轉換資料資料格式為 Grobian 所限定的格式，狀態值均轉為數字。變數名稱改由數字代替。如表 4. 1。

表 4. 1 警鈴之變數狀態值轉換

貝氏網路變數		Grobian 格式	
Burglary	yes	Attribute1(a1)	1
	no		2
Earthquakes	yes	Attribute2(a2)	1
	no		2
Alarm	yes	Attribute3(a3)	1
	no		2
John	yes	Attribute4(a4)	1
	no		2
Mary	yes	Attribute5(a5)	1
	no		2

3. 給予變數因果排序，依序是 Burglary, Earthquakes, Alarm, John, Mary。
4. 匯入資料至 Grobian，進行法則粹取。先從最後的變數 Attribute5(Mary) 開始當作決策屬性，其餘變數當作條件屬性。得到結果如圖 4. 2。再依照 a4, a3, a2, a1 的順序，完成所有變數的建構法則。
5. 篩選符合條件的法則（支持度大於 20 筆資料，法則長度不超過 2，通過顯著性檢定），如表 4. 2。
6. 經由表 4. 2，看出下列變數間存在關聯。
 - (1) a3 (Alarm) a5 (Mary)
 - (2) a3 (Alarm) a4 (John)
 - (3) a1 (Burglary) a2 (Earthquakes) a3 (Alarm)

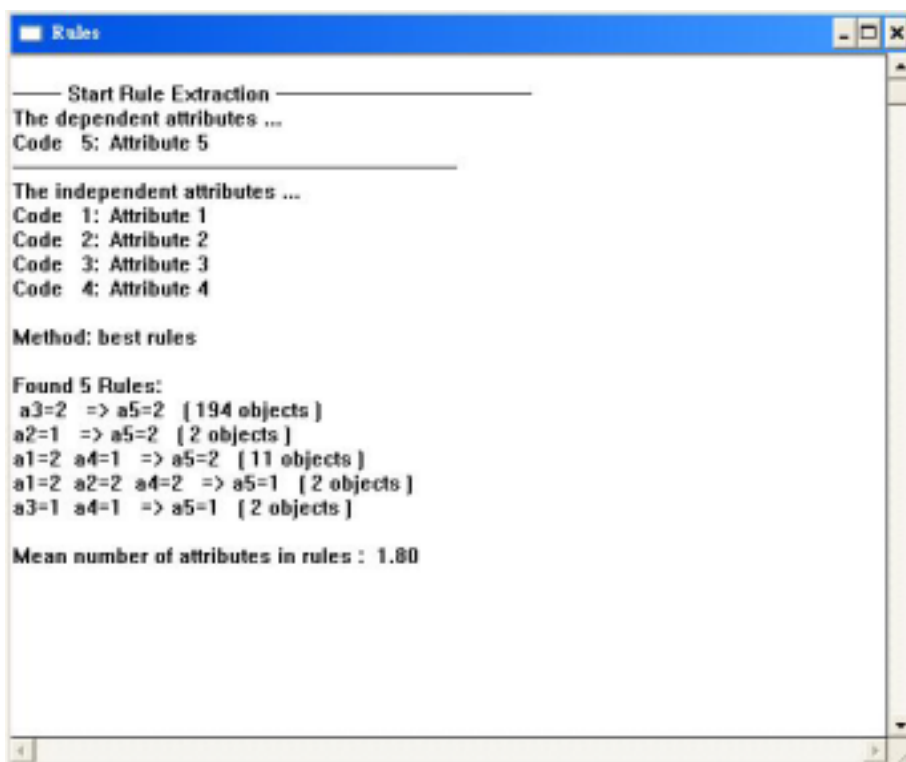


圖 4.2 產生 Attribute5 (Mary) 建構法則

表 4.2 警鈴之建構法則

Rules	Support	Length
a3=2 => a5=2	194 objects	1
a3=2 => a4=2	185 objects	1
a1=2 a2=2 => a3=2	193 objects	2

7. 依據步驟 6 變數間的關聯，建構出貝氏網路雛型。如圖 4.3。

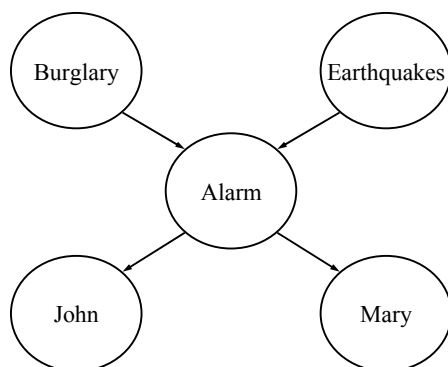


圖 4.3 本研究方法建構之警鈴貝氏網路雛型

8. 檢查網路中是否有多餘的連結。此貝氏網路雛型沒有多餘的連結。
9. 利用 Norsys Netica 進行參數學習。得到最後的貝氏網路如圖 4. 4。

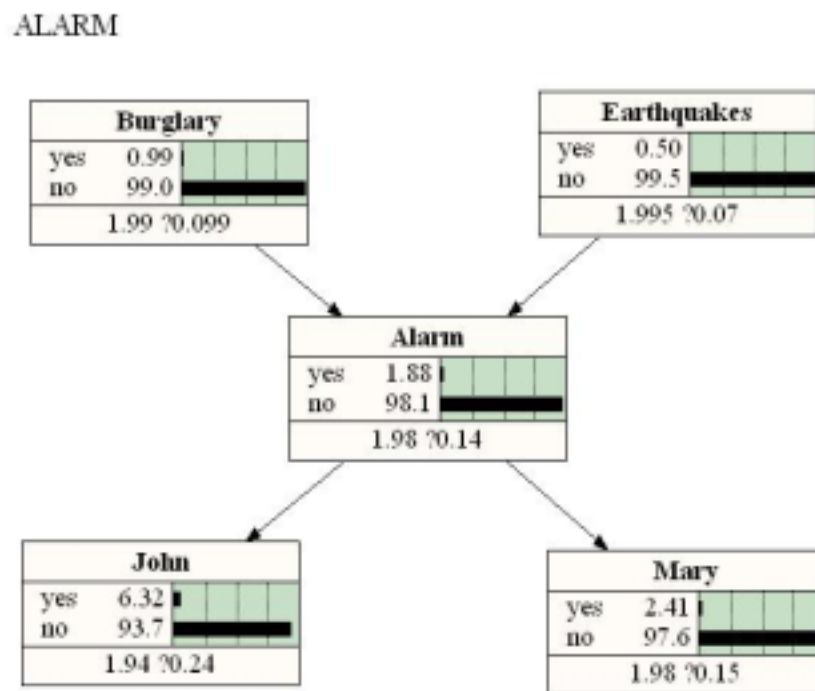


圖 4. 4 本研究方法架構之警鈴貝氏網路圖形

10. 進行架構相似性分析。

- (1) 警鈴範例貝氏網路圖形中的關聯集合， $a = \{(Burglary, Alarm), (Burglary, John), (Burglary, Mary), (Earthquakes, Alarm), (Earthquakes, John), (Earthquakes, Mary), (Alarm, John), (Alarm, Mary)\}$ ， $X=8$ 。
- (2) 本研究方法建構之警鈴貝氏網路中的關聯集合， $b = \{(Burglary, Alarm), (Burglary, John), (Burglary, Mary), (Earthquakes, Alarm), (Earthquakes, John), (Earthquakes, Mary), (Alarm, John), (Alarm, Mary)\}$ ， $Y=8$ 。
- (3) 共同的關聯集合 $\{(Burglary, Alarm), (Burglary, John), (Burglary, Mary), (Earthquakes, Alarm), (Earthquakes, John), (Earthquakes, Mary), (Alarm, John), (Alarm, Mary)\}$ ， $m=8$ 。
- (4) Jaccard 係數為 $8/(8+8-8)=1=100\%$ 。兩個貝氏網路的架構相似性為 100%。

11. 利用 Norsys Netica 產生新的 400 筆資料，進行資料結構相似性分析。

(1) $H_0 : T^+ - T^- = 0$

$H_1 : T^+ - T^- \neq 0$

(2) 利用統計分析軟體 SPSS[2]，進行魏克森符號檢定。雙尾檢定，顯著水準 0.025。得到 $p=0.570 > 0.025$ ，不否定 H_0 。如圖 4.5。

(3) 兩個網路的資料結構是相似的。



圖 4.5 本研究方法之警鈴貝氏網路魏克森符號檢定

本方法建構出的警鈴貝氏網路，與原本範例貝氏網路。在架構相似性程度為 100%，本研究方法可以表達範例變數 100% 關聯。經參數學習後，產生的新資料與原本的資料可以通過魏克森符號檢定。顯示出本方法建構的警鈴貝氏網路足以描述原本的警鈴貝氏網路。

4.1.2 K2 演算法建構之警鈴貝氏網路

本節將以相同建構資料，利用 K2 演算法建構警鈴貝氏網路，之後進行架構相似性、資料結構相似性分析。進行步驟如下：

1. 給定一組相同的節點排序，Burglary, Earthquakes, Alarm, John, Mary。
2. 匯入建構資料至貝氏網路應用軟體—Bayesware Discovery，得到貝氏網路雛型如圖 4. 6。

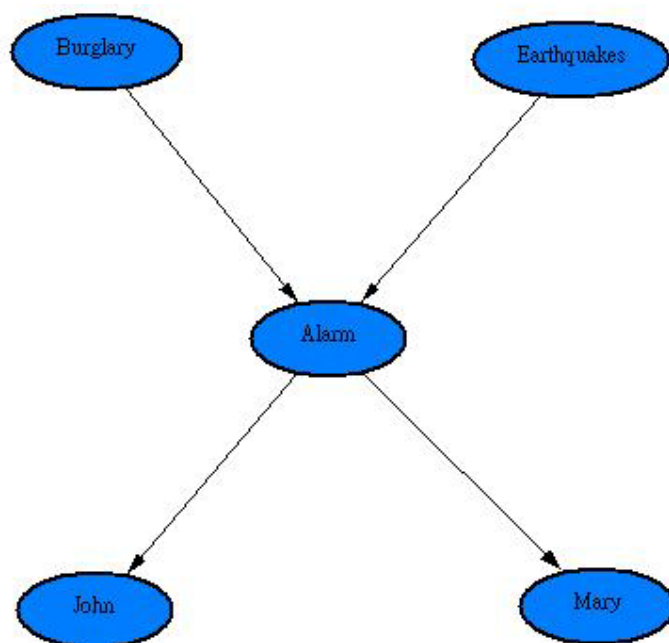


圖 4. 6 K2 演算法建構之警鈴貝氏網路雛型

3. 步驟 2 得到的貝氏網路雛型，搭配未參與建構資料，利用 Norsys Netica 完成參數學習。得到最終警鈴之貝氏網路如圖 4. 7。
4. 架構相似性。
 - (1) 警鈴範例貝氏網路圖形中的關聯集合， $a = \{(Burglary, Alarm), (Burglary, John), (Burglary, Mary), (Earthquakes, Alarm), (Earthquakes, John), (Earthquakes, Mary), (Alarm, John), (Alarm, Mary)\}$ ， $X=8$ 。
 - (2) K2 演算法建構之警鈴貝氏網路中的關聯集合， $b = \{(Burglary, Alarm), (Burglary, John), (Burglary, Mary), (Earthquakes, Alarm), (Earthquakes, John), (Earthquakes, Mary), (Alarm, John), (Alarm, Mary)\}$ ， $Y=8$ 。

- (3) 共同的關聯集合 $\{(Burglary, Alarm), (Burglary, John), (Burglary, Mary), (Earthquakes, Alarm), (Earthquakes, John), (Earthquakes, Mary), (Alarm, John), (Alarm, Mary)\}$, $m=8$ 。
- (4) Jaccard 係數為 $8/(8+8-8)=1=100\%$ 。兩個貝氏網路的架構相似性為 100%。

ALARM

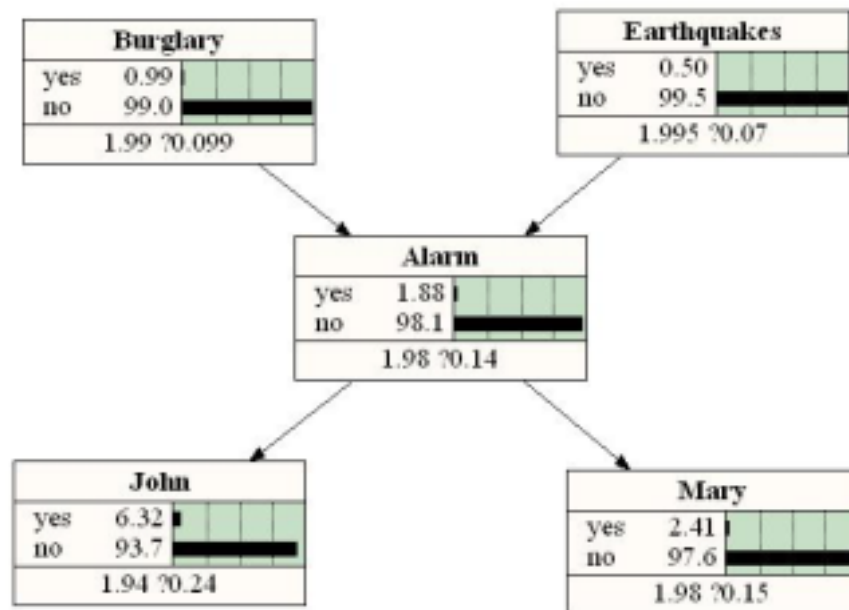


圖 4.7 K2 演算法建構之警鈴貝氏網路

5. 利用 Norsys Netica 產生新的 400 筆資料，進行資料結構相似性分析。
- (1) $H_0 : T^+ - T^- = 0$
 $H_1 : T^+ - T^- \neq 0$
 - (2) 利用統計分析軟體 SPSS，進行魏克森符號檢定。雙尾檢定，顯著水準 0.025。得到 $p=0.570 > 0.025$ ，不否定 H_0 。如圖 4.8。
 - (3) 兩個網路的資料結構是相似的。

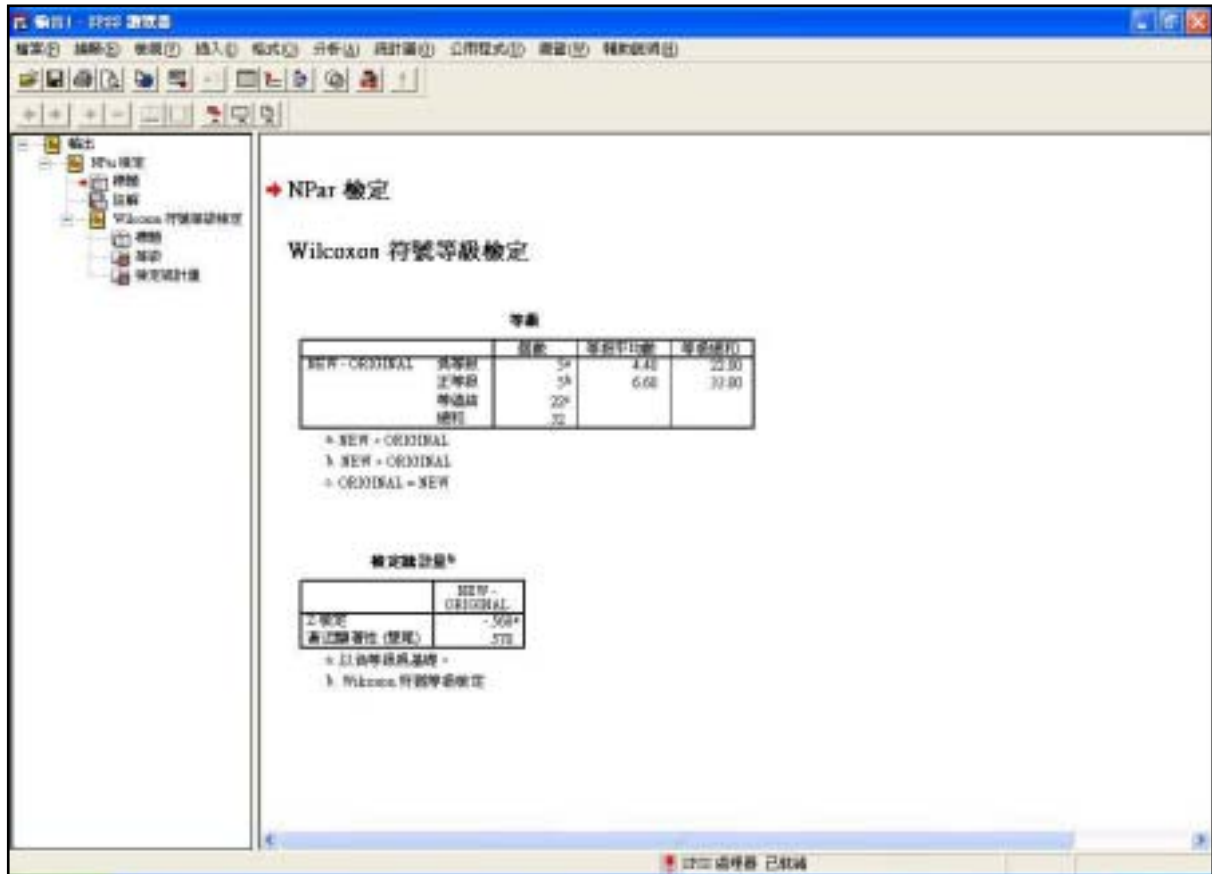


圖 4.8 K2 演算法之警鈴貝氏網路魏克森符號檢定

K2 演算法建構出的警鈴貝氏網路，與原本範例貝氏網路。在架構相似性程度為 100%，K2 演算法可以表達範例變數 100% 關聯。經參數學習後，產生的新資料與原本的資料可以通過魏克森符號檢定。顯示出 K2 演算法建構的警鈴貝氏網路足以描述原本的警鈴貝氏網路。

4.2 範例二：動物分類

這是一個簡單的貝氏網路分類範例[24]，將五種動物，包括猴子、企鵝、鴨嘴獸、知更鳥和烏龜，依照牠們的下述六種特徵分類。

1. 外殼（有，無）。
2. 撫育方式（胎生、卵生）。
3. 棲息環境（空中、陸地、水中）。
4. 綱（鳥類、哺乳類、爬蟲類）。
5. 恆溫（是、否）。
6. 外皮（皮毛、羽毛、甲殼）。

動物分類之範例貝氏網路架構圖，如圖 4.9。

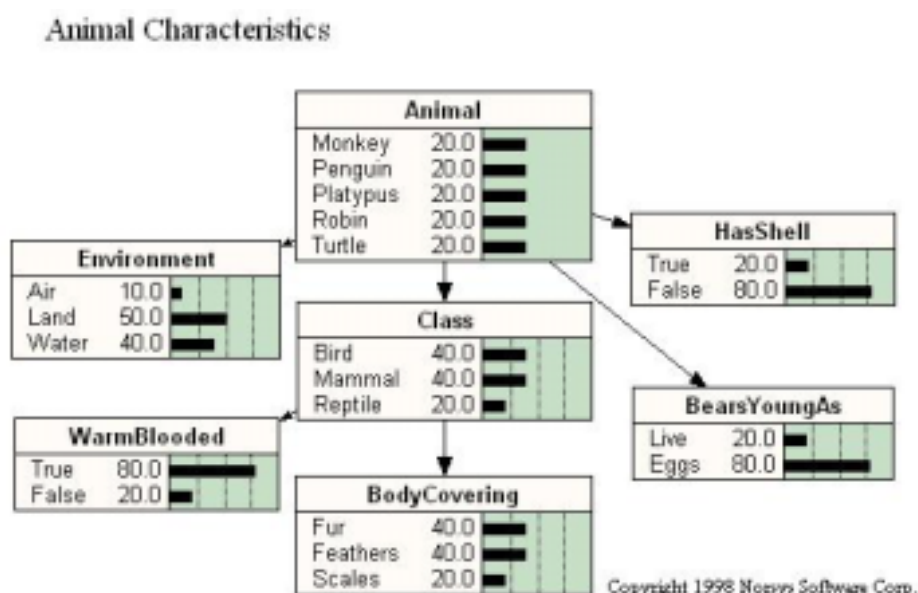


圖 4.9 動物分類之貝氏網路

4.2.1 粗集方法建構之動物分類貝氏網路

依據第三章的步驟，建構成貝氏網路，詳細的進行步驟如下：

1. 利用 Norsys Netica 產生 400 筆資料。並用其中 200 筆資料進行網路建構，剩下的 200 筆資料進行參數學習。
2. 轉換資料格式為 Grobian 所限定的格式，狀態值均轉為數字。變數名稱改由數字代替。如表 4.3。

3. 給予變數因果排序，依序是 Animal, Class, BearsYoungAs, HasShell, Environment, BodyCovering, WarmBlooded。
4. 匯入資料至 Grobian，進行法則粹取。先從最後的變數 Attribute7 (WarmBlooded) 開始當作決策屬性，其餘變數當作條件屬性。再依照 a6, a5, a4, a3, a2, a1 的順序，完成所有變數的建構法則。

表 4.3 動物分類之變數狀態值轉換

貝氏網路變數		Grobian 格式	
Animal	Monkey	Attribute1(a1)	1
	Penguin		2
	Platypus		3
	Robin		4
	Turtle		5
Class	Bird	Attribute2(a2)	1
	Mammal		2
	Reptile		3
BearsYoungAs	Live	Attribute3(a3)	1
	Eggs		2
HasShell	True	Attribute4(a4)	1
	False		2
Environment	Air	Attribute5(a5)	1
	Land		2
	Water		3
BodyCovering	Fur	Attribute6(a6)	1
	Feathers		2
	Scales		3
WarmBlooded	True	Attribute7(a7)	1
	False		2

5. 篩選符合條件的法則 (支持度大於 20 筆資料，法則長度不超過 2，通過顯著性檢定)，如表 4.4。
6. 經由表 4.4，看出下列變數間存在關聯。
 - (1) a4 (HasShell) a7 (WarmBlooded)
 - (2) a6 (BodyCovering) a7 (WarmBlooded)
 - (3) a2 (Class) a6 (BodyCovering)
 - (4) a4 (HasShell) a6 (BodyCovering)

- (5) a1 (Animal) a5 (Environment)
- (6) a3 (BearsYoungAs) a5 (Environment)
- (7) a2 (Class) a4 (HasShell)
- (8) a2 (Class) a3 (BearsYoungAs)
- (9) a1 (Animal) a3 (BearsYoungAs)
- (10) a1 (Animal) a2 (Class)

表 4.4 動物分類之建構法則

Rules	Support	Length
a4=2 => a7=1	159 objects	1
a6=3 => a7=2	41 objects	1
a2=1 => a6=2	84 objects	1
a4=1 => a6=3	41 objects	1
a1=3 => a5=3	42 objects	1
a3=1 => a5=2	33 objects	1
a2=1 => a4=2	84 objects	1
a2=1 => a3=2	84 objects	1
a1=3 => a3=2	42 objects	1
a1=2 => a2=1	42 objects	1

11. 依據步驟 6 變數間的關聯，建構出貝氏網路雛型。如圖 4.10。

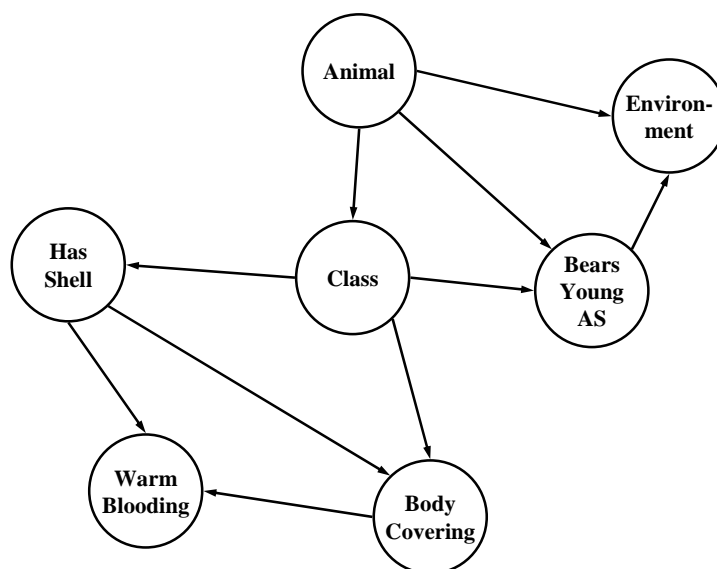


圖 4.10 本研究方法建構之動物分類貝氏網路雛型

12. 檢查網路中是否有多餘的連結。此貝氏網路雛型有四組成三角形的關聯。包括 $A=\{(Animal), (BearsYoungAs), (HasShell)\}$, $B=\{(Animal), (Class), (BearsYoungAs)\}$, $C=\{(Class), (Environment), (BodyCovering)\}$, $D=\{(Environment), (BodyCovering), (WarmBlooded)\}$ 。計算這些關聯的 Unexplained Information 值，如圖 4. 11。消除 Unexplained Information 值較低的關聯。得到新的動物分類貝氏網路雛型如圖 4. 12。

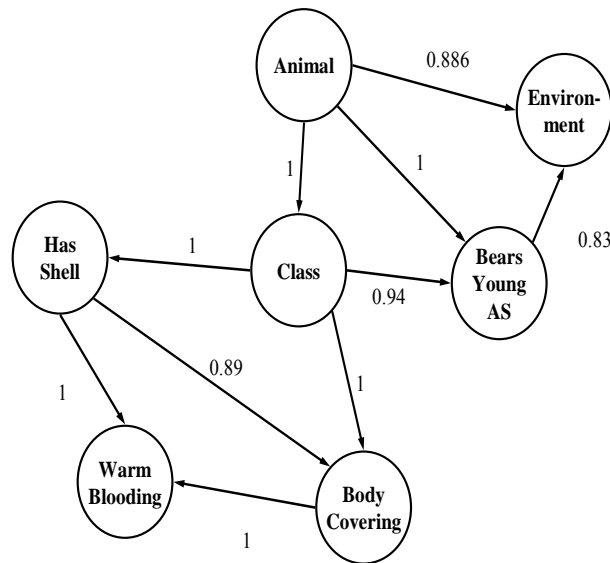


圖 4. 11 計算關聯的 Unexplained Information 值

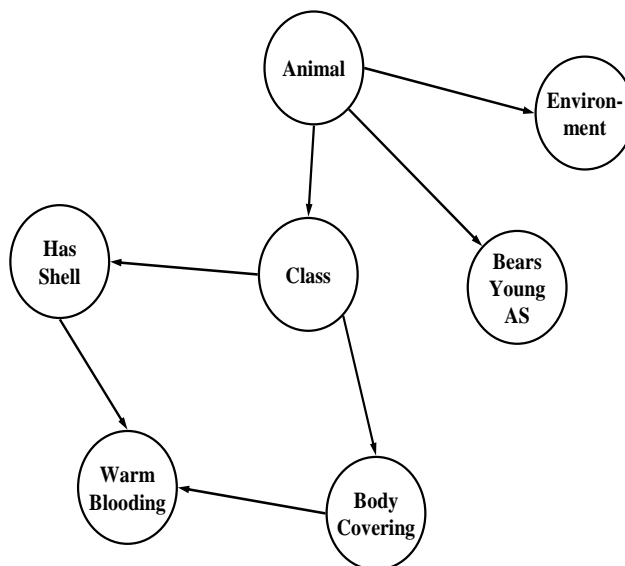


圖 4. 12 去除多餘關聯後的動物分類貝氏網路

13. 利用 Norsys Netica 進行參數學習。得到最後的貝氏網路如圖 4. 13。

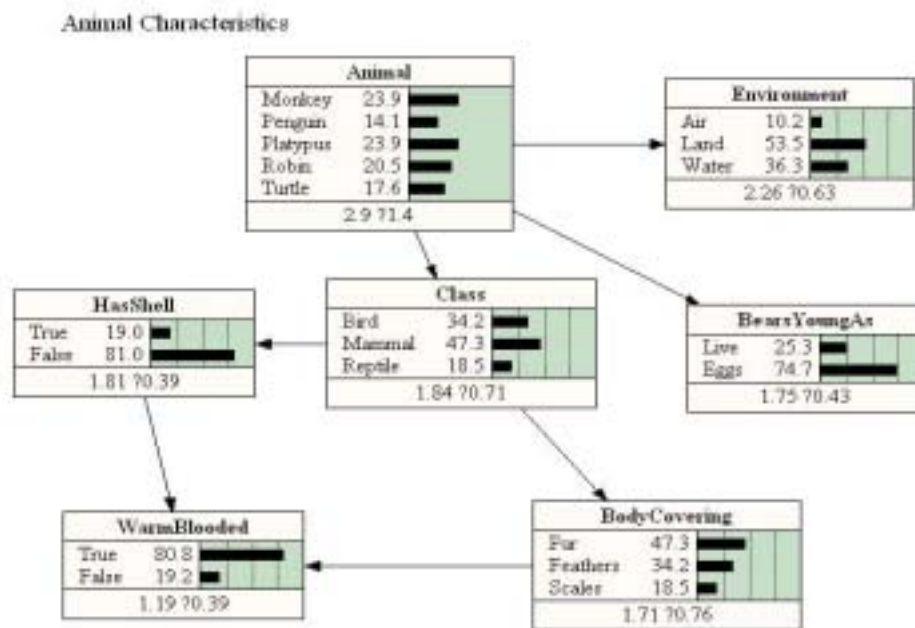


圖 4. 13 本研究方法架構之動物分類貝氏網路圖形

14. 進行架構相似性分析。

- (1) 動物分類範例貝氏網路圖形中的關聯集合， $a = \{(Animal, Class), (Animal, BearsYoungAs), (Animal, HasShell), (Animal, Environment), (Animal, BodyCovering), (Animal, WarmBlooded), (Class, BodyCovering), (Class, WarmBlooded)\}$ ， $X=8$ 。
- (2) 本研究方法建構之動物分類貝氏網路中的關聯集合， $b = \{(Animal, Class), (Animal, BearsYoungAs), (Animal, HasShell), (Animal, Environment), (Animal, BodyCovering), (Animal, WarmBlooded), (Class, HasShell), (Class, BodyCovering), (Class, WarmBlooded), (HasShell, WarmBlooded), (BodyCovering, WarmBlooded)\}$ ， $Y=11$ 。
- (3) 共同的關聯集合 $\{(Animal, Class), (Animal, BearsYoungAs), (Animal, HasShell), (Animal, Environment), (Animal, BodyCovering), (Animal, WarmBlooded), (Class, BodyCovering), (Class, WarmBlooded)\}$ ， $m=8$ 。
- (4) Jaccard 係數為 $8/(8+11-8)=0.7272=72.72\%$ 。兩個貝氏網路的架構相似性為 72.72%。

15. 利用 Norsys Netica 產生新的 400 筆資料，進行資料結構相似性分析。

(1) $H_0 : T^+ - T^- = 0$

$H_1 : T^+ - T^- > 0$

(2) 利用統計分析軟體 SPSS，進行魏克森符號檢定。雙尾檢定，顯著水準 0.025。得到 $p=0 < 0.025$ ，否定 H_0 。如圖 4. 14。

(3) 兩個網路的資料結構不是相似的。

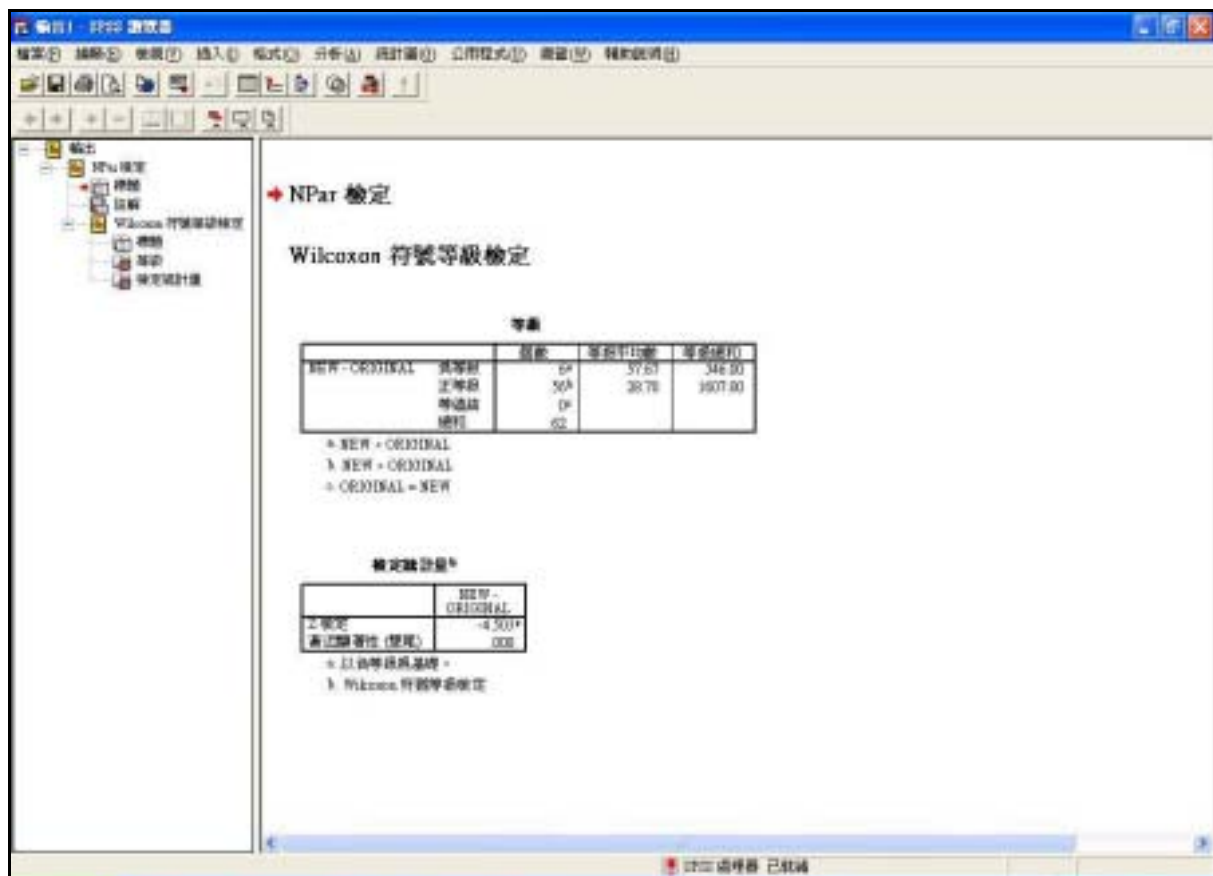


圖 4. 14 本研究方法建構之動物分類貝氏網路魏克森符號檢定

本研究方法建構之動物分類貝氏網路，與動物分類範例貝氏網路。在架構相似程度為 78.78%，本研究方法可表達出範例變數間 78.78%的關係。但經參數學習後，產生的新資料與原本的資料無法通過魏克森符號檢定。因此，本研究方法雖可描述出此動物分類的變數關聯，但無法提供明確的機率分佈。

4.2.2 K2 演算法建構之動物分類貝氏網路

本節將以相同建構資料，利用 K2 演算法建構警鈴貝氏網路，之後進行架構相似性、資料結構相似性分析。進行步驟如下：

1. 給定一組相同的節點排序，Animal, Class, BearsYoungAs, HasShell, Environment, BodyCovering, WarmBlooded。
2. 匯入建構資料至貝氏網路應用軟體—Bayesware Discovery，得到貝氏網路雛型如。

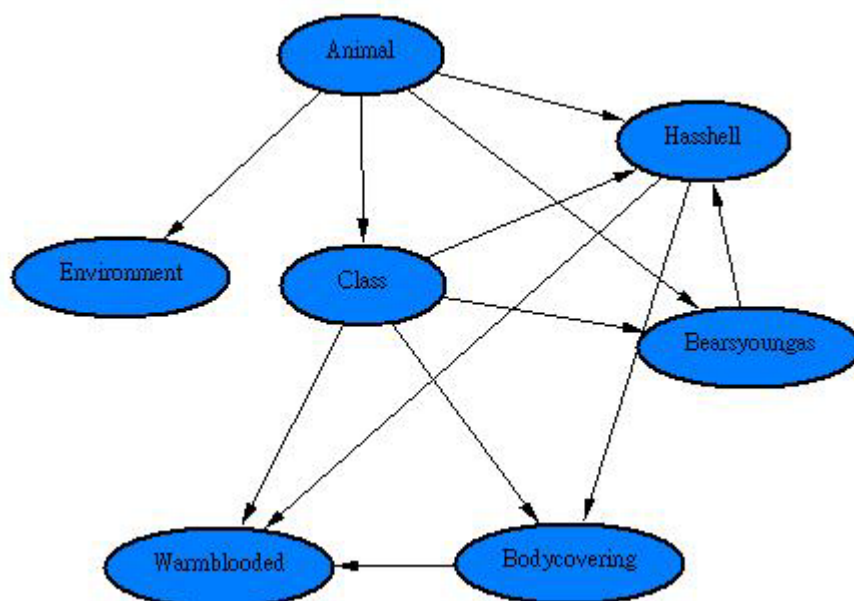


圖 4.15 K2 演算法建構之動物分類貝氏網路雛型

3. 步驟 2 得到的貝氏網路雛型，搭配未參與建構資料，利用 Norsys Netica 完成參數學習。得到最終警鈴之貝氏網路如。
4. 架構相似性。
 - (1) 動物分類範例貝氏網路圖形中的關聯集合， $a = \{(Animal, Class), (Animal, BearsYoungAs), (Animal, HasShell), (Animal, Environment), (Animal, BodyCovering), (Animal, WarmBlooded), (Class, BodyCovering), (Class, WarmBlooded)\}$ ， $X=8$ 。
 - (2) K2 演算法建構之動物分類貝氏網路， $b = \{(Animal, Class), (Animal, BearsYoungAs), (Animal, HasShell), (Animal, Environment), (Animal, BodyCovering), (Animal, WarmBlooded), (Class, BodyCovering),$

(Class, WarmBlooded), (Class, HasShell), (Class, BearsYoungAs), (BearsYoungAs, BodyCovering), (BearsYoungAs, HasShell), (BearsYoungAs, WarmBlooded), (HasShell, BodyCovering), (HasShell, WarmBlooded), (BodyCovering, WarmBlooded) } , Y=16。

- (3) 共同的關聯集合 {(Animal, Class), (Animal, BearsYoungAs), (Animal, HasShell), (Animal, Environment), (Animal, BodyCovering), (Animal, WarmBlooded), (Class, BodyCovering), (Class, WarmBlooded)} , m=8
- (4) Jaccard 係數為 $8/(8+16-8)=0.5=50\%$ 。兩個貝氏網路的架構相似性為 50%。

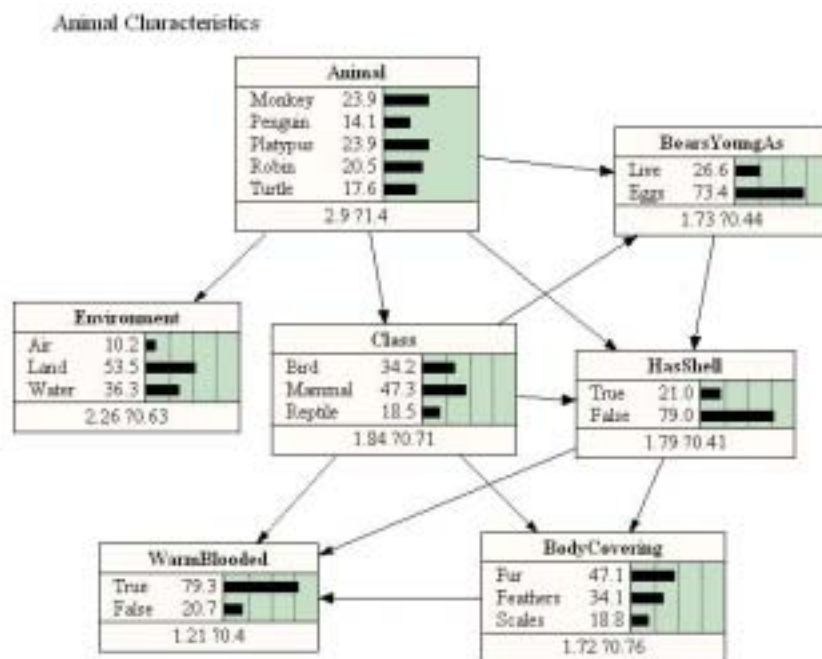


圖 4.16 K2 演算法建構之動物分類貝氏網路

5. 利用 Norsys Netica 產生新的 400 筆資料，進行資料結構相似性分析。

(1) H_0 : 新資料來自母體 N_1 =原始資料來自母體 N_2

H_1 : 新資料來自母體 N_1 原始資料來自母體 N_2

(2) 利用統計分析軟體 SPSS，進行魏克森符號檢定。雙尾檢定，顯著水準 0.025。得到 $p=0 < 0.025$ ，不否定 H_0 。如圖 4.17。

(3) 兩個網路的資料結構不是相似的。

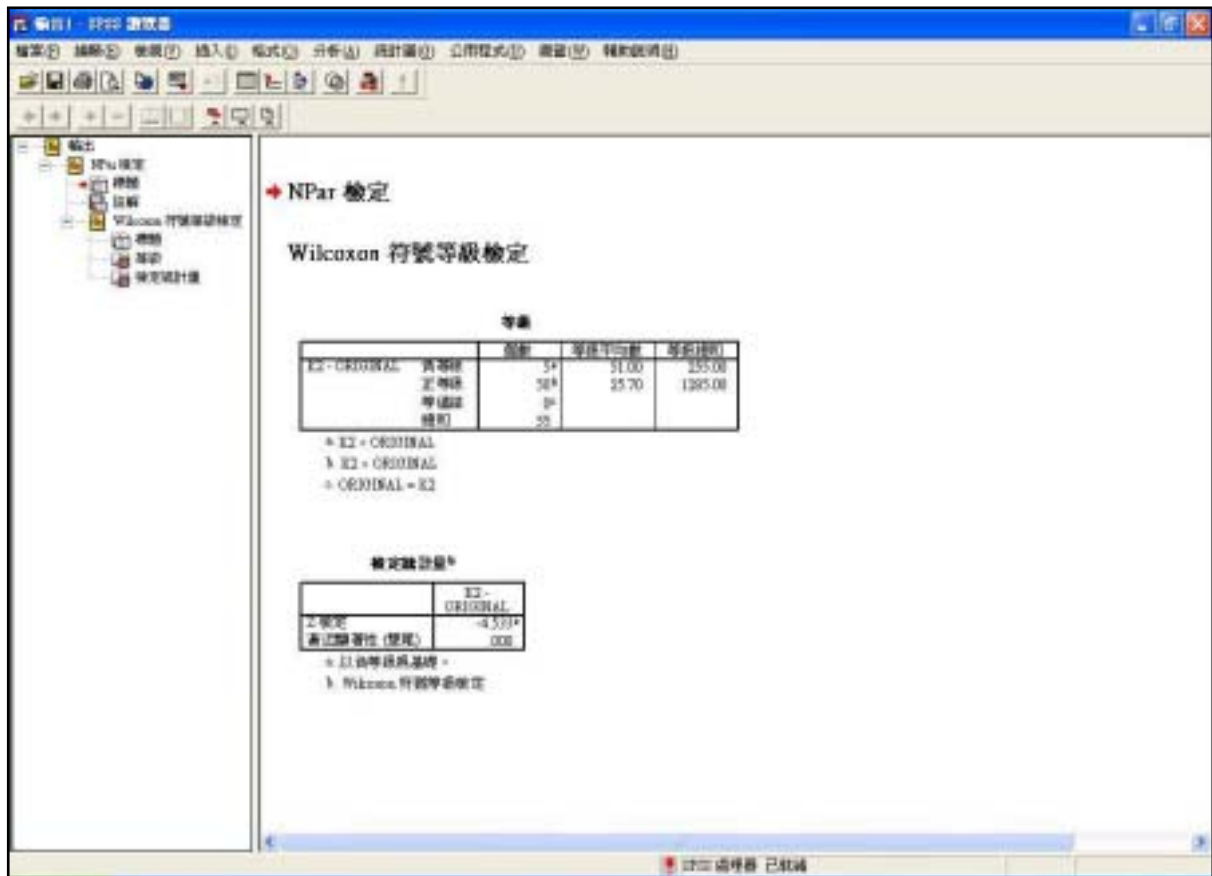


圖 4. 17 K2 演算法建構之動物分類貝氏網路魏克森符號檢定

K2 演算法建構之動物分類貝氏網路，與原本範例動物分類貝氏網路。在結構相似程度為 50%，K2 演算法可表達原本變數間 50% 關聯。但經參數學習後，產生的新資料與原本的資料無法通過魏克森符號檢定。

K2 演算法在架構相似性比較上，略差於本研究方法。原因為網路間存在過多的連結，增加了網路結構複雜程度。資料結構相似性分析上，與本研究方法相同，無法通過魏克森符號檢定。對於明確性的分類問題，K2 演算法也有使用上的限制。

關於本研究方法和 K2 演算法無法通過魏克森符號檢定原因，詳述如下：

本研究重點在於貝氏網路的架構學習。但架構確定之後，若無經過參數學習的步驟，還不是一個完整的貝氏網路。

本研究藉由貝氏網路應用軟體—Norsys Netica[24]，來完成參數學習的部分。但此軟體對於動物分類貝氏網路參數學習時出現下面的問題：

1. 參數學習的資料是由範例網路產生。因此，例如猴子對於環境分類的結果，必定是陸地 100%。
2. 本研究方法與 K2 演算法完成貝氏網路架構後，動物對於環境的關係為相依性。因此，猴子對於環境的機率分配因該是，陸地 100%。
3. 但是經由 Norsys Netica 匯入資料進行參數學習後，出現的機率分配為，陸地 96.1%、空中 1.96%、水中 1.96%。
4. 其他的節點也發生上述情形。因此，本研究方法與 K2 演算法在得到貝氏網路後，產生新資料要進行資料結構相似性分析時，會因為機率分配而產生許多新案例。
5. 與原始資料比較，本研究方法多出 55 個新案例，K2 演算法多出 48 個新案例。由於新案例過多，導致無法通過魏克森符號等級檢定，資料結構相似性分析也呈現不一致的情形。

本研究方法與 K2 演算法都無法通過魏克森符號等級檢定。因此，比較兩個方法在動物分類貝氏網路中，機率分配的情形。

範例貝氏網路中，六個動物分類特徵將五種動物分類，動物分類的機率會集中在某一特徵上。例如上述猴子的情形。接下來針對本研究方法和 K2 演算法在動物分類特徵上機率的集中情形做一比較。發現在撫育方式 (BearYoungAs)、外殼 (HasShell)、外皮 (BodyCovering)、恆溫 (WarmBlooded) 這四個特徵有不一樣的機率分配情形。如下表 4. 5。

表 4.5 機率分配比較

		BearYoungAs(%)	HasShell(%)	BodyCovering(%)	WarmBlooded(%)
Monkey	範例	100(Live)	100(False)	100(Fur)	100(True)
	本研究	98	97.1	94.2	96.2
	K2	96.1	95.2	93	94.8
Penguin	範例	100(Eggs)	100(False)	100(Feathers)	100(True)
	本研究	96.7	95.5	91.1	94.3
	K2	93.7	92.2	89.2	92.1
Platypus	範例	100(Eggs)	100(False)	100(Fur)	100(True)
	本研究	98	97.1	94.2	96.2
	K2	96.1	95.2	93	94.8
Robin	範例	100(Eggs)	100(False)	100(Feathers)	100(True)
	本研究	97.7	96.4	92.9	95.2
	K2	95.5	94.4	91.5	93.6
Turtle	範例	100(Eggs)	100(True)	100(Scales)	100(False)
	本研究	97.3	92.2	89.8	88.7
	K2	94.8	93.6	88.3	89.5

上表以動物為主將結果繪製成長條圖如下圖 4.18，圖 4.19，圖 4.20，圖 4.21，圖 4.22。圖中，左邊長條為範例網路的機率，中間長條為本研究方法建構貝氏網路的機率，右邊長條為 K2 演算法建構貝氏網路的機率。

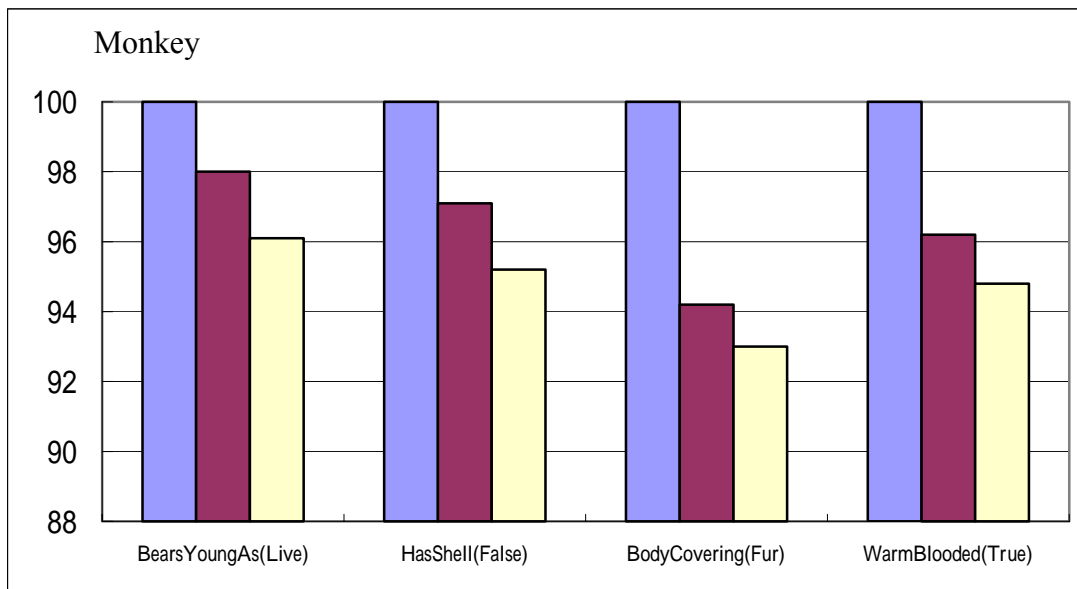


圖 4.18 Mokey 機率比較長條圖

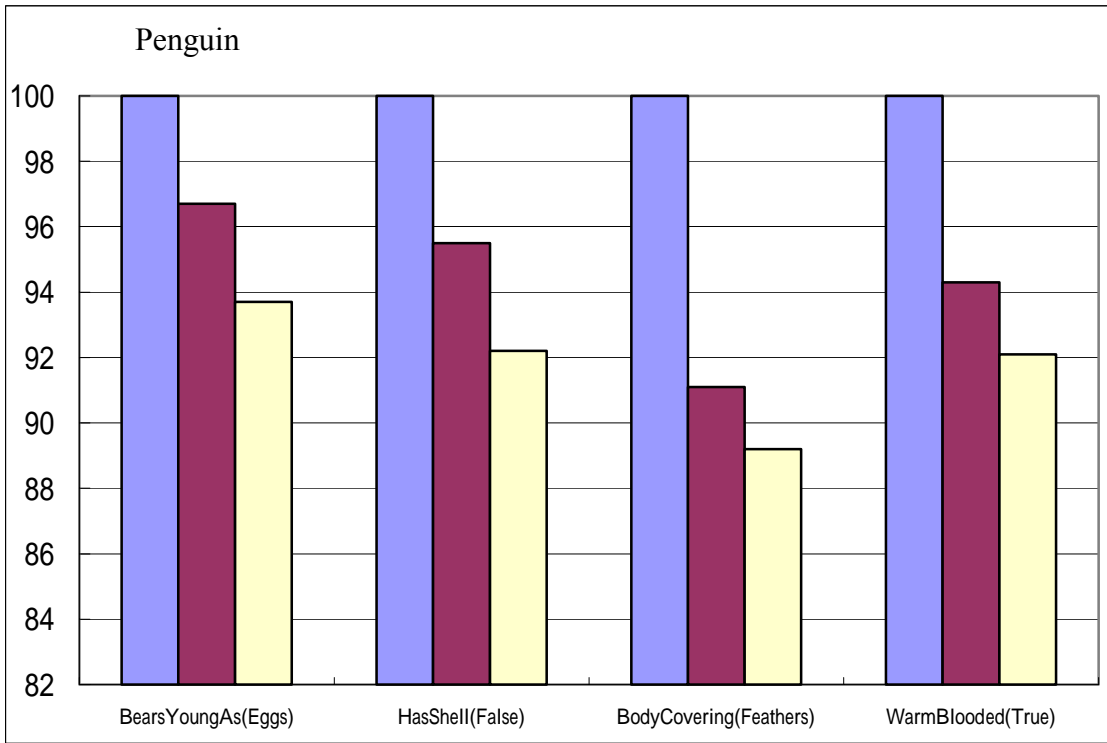


圖 4. 19Penguin 機率比較長條圖

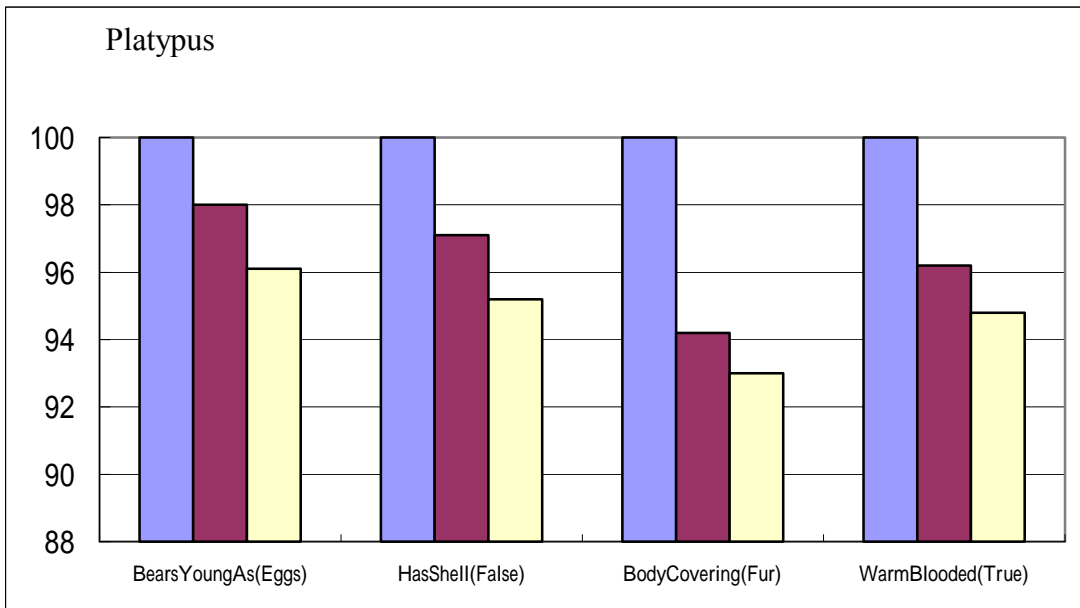


圖 4. 20 Platypus 機率比較長條圖

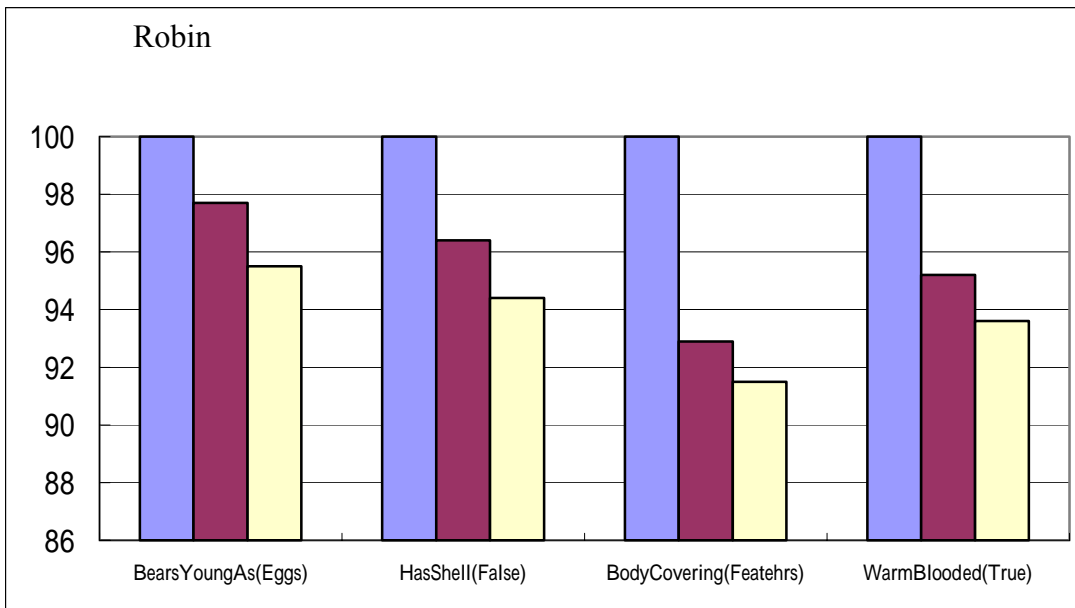


圖 4. 21 Robin 機率比較長條圖

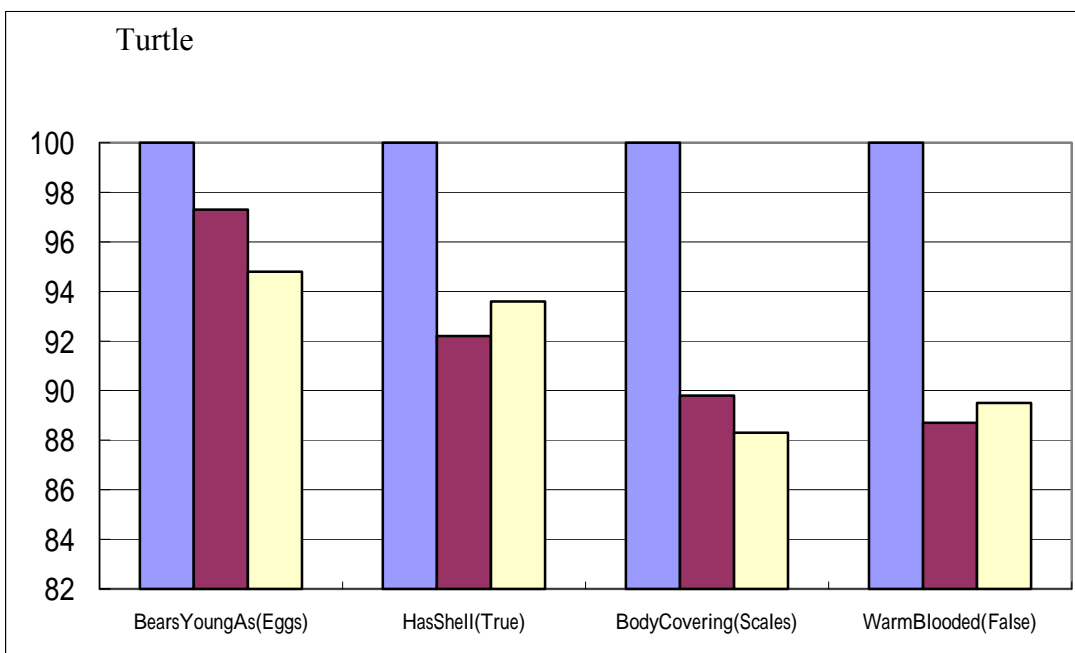


圖 4. 22 Turtle 機率比較長條圖

由上圖表看出,本研究方法比 K2 演算法在機率分配上更接近範例網路的機率。雖然兩種方法與範例網路在資料結構相似性上不能有一致的情形,但本研究方法建構的貝氏網路雛型經過參數學習後,有較接近於原始的機率。

4.3 範例三：汽車故障診斷

貝氏網路現今大量使用在診斷系統上，包括醫療診斷系統或是機器設備故障診斷系統。此範例是以汽車為例的一個貝氏網路診斷系統，診斷的決策點是汽車能否發動[24]。診斷的節點包括下述的 14 個節點。

1. 火星塞品質（好、差、極差）。
2. 汽車轉軸（能動、不能動）。
3. 啟動器（好、故障）。
4. 前燈（亮、暗、不亮）。
5. 火星塞電壓（強、弱、無）。
6. 電池電壓（強、弱、無）。
7. 點火正時（好、差、極差）。
8. 充電系統（好、故障）。
9. 火星塞（好、鬆弛、污損）。
10. 分配器（好、故障）。
11. 主保險絲（好、斷裂）。
12. 啟動馬達（好、故障）。
13. 電池壽命（新、舊、很舊）。
14. 交流發電機（好、故障）。

汽車診斷之範例貝氏網路如圖 4.23。

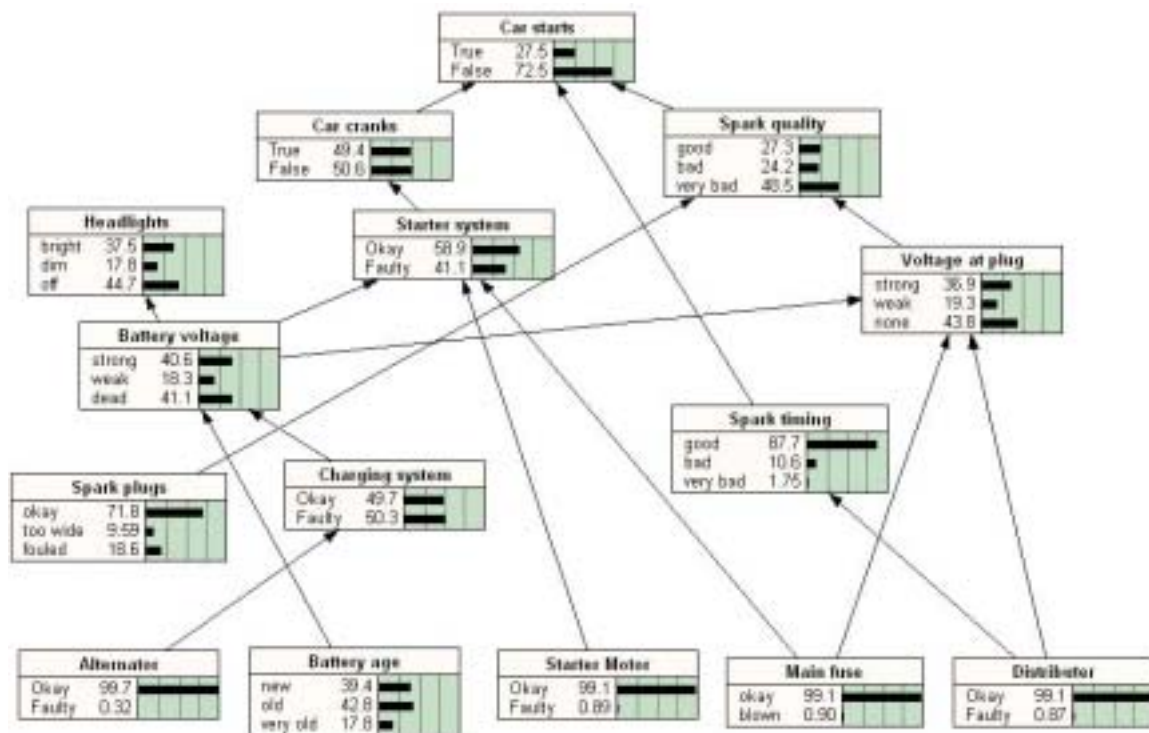


圖 4. 23 汽車診斷之貝氏網路

4.3.1 粗集方法建構之汽車故障診斷貝氏網路

依據第三章的步驟，建構成貝氏網路，詳細的進行步驟如下：

1. 利用 Norsys Netica 產生 400 筆資料。並用其中 200 筆資料進行網路建構，剩下的 200 筆資料進行參數學習。
2. 轉換資料資料格式為 Grobian 所限定的格式，狀態值均轉為數字。變數名稱改由數字代替。如表 4. 6。
3. 給予變數因果排序，依序是 Alternator, Battery age, Starter Motor, Main fuse, Distributer, Spark plugs, Charging system, Spark timing, Battery voltage, Voltage at plugs, Headlights, Starter system, Car cranks, Spark Quality, Car starts。
4. 匯入資料至 Grobian，進行法則粹取。先從最後的變數 Attribute15 (Car starts)開始當作決策屬性，其餘變數當作條件屬性。再依照 a14, a13, a12, a11, a10, a9, a8, a7, a6, a5, a4, a3, a2, a1 的順序，完成所有變數的建構法則。

表 4.6 汽車診斷之變數狀態值轉換

貝氏網路變數		Grobian 格式	
Alternator	Okay	Attribute1(a1)	1
	Faulty		2
Battery age	new	Attribute2(a2)	1
	old		2
	very old		3
Starter Motor	Okay	Attribute3(a3)	1
	Faulty		2
Main fuse	Okay	Attribute4(a4)	1
	blown		2
Distributor	Okay	Attribute5(a5)	1
	Faulty		2
Spark plugs	Okay	Attribute6(a6)	1
	too wide		2
	fouled		3
Charging system	Okay	Attribute7(a7)	1
	faulty		2
Spark timing	good	Attribute8(a8)	1
	bad		2
	very bad		3
Battery voltage	strong	Attribute9(a9)	1
	weak		2
	dead		3
Voltage at plugs	strong	Attribute10(a10)	1
	weak		2
	none		3
Headlights	bright	Attribute11(a11)	1
	dim		2
	off		3
Starter system	Okay	Attribute12(a12)	1
	faulty		2
Car cranks	True	Attribute13(a13)	1
	False		2
Spark Quality	good	Attribute14(a14)	1
	bad		2
	very bad		3
Car starts	True	Attribute15(a15)	1
	False		2

5. 篩選符合條件的法則（支持度大於 20 筆資料，法則長度不超過 2，通過顯著性檢定），如表 4.7。

表 4.7 汽車診斷之建構法則

Rules	Support	Length
a2=2 a7=2 => a15=2	44 objects	2
a14=3 => a15=2	100 objects	1
a13=2 => a15=2	97 objects	1
a6=1 a10=2 => a14=2	31 objects	2
a10=3 => a14=3	89 objects	1
a10=3 a12=1 => a13=1	20 objects	2
a12=2 => a13=2	72 objects	1
a9=1 => a12=1	81 objects	1
a2=2 a7=1 => a12=1	34 objects	2
a9=2 => a11=2	38 objects	1
a8=1 a9=3 => a11=3	67 objects	2
a2=1 a10=1 => a11=1	30 objects	2
a7=2 a9=2 => a10=2	21 objects	2
a5=1 a9=3 => a10=3	78 objects	2
a2=1 a7=1 => a9=1	40 objects	2

6. 經由表 4.8，看出下列變數間存在關聯。

- (1) a2 (Battery age) a7 (Charging system) a15 (Car Starts)
- (2) a14 (Spark Quality) a15 (Car Starts)
- (3) a13 (Car cranks) a15 (Car Starts)
- (4) a6 (Spark plugs) a10 (Voltage at plugs) a14 (Spark Quality)
- (5) a10 (Voltage at plugs) a14 (Spark Quality)
- (6) a10 (Voltage at plugs) a12 (Starter system) a13 (Car cranks)
- (7) a12 (Starter system) a13 (Car cranks)
- (8) a9 (Battery voltage) a12 (Starter system)
- (9) a2 (Battery age) a7 (Charging system) a12 (Starter system)
- (10) a9 (Battery voltage) a11 (Headlights)
- (11) a8 (Spark timing) a9 (Battery voltage) a11 (Headlights)
- (12) a2 (Battery age) a10 (Voltage at plugs) a11 (Headlights)

(13)a7 (Charging system) a9 (Battery voltage) a10 (Voltage at plugs)

(14)a5 (Distributer) a9 (Battery voltage) a10 (Voltage at plugs)

(15)a2 (Battery age) a7 (Charging system) a9 (Battery voltage)

7. 依據步驟 6 變數間的關聯，建構出貝氏網路雛型。如圖 4. 24。

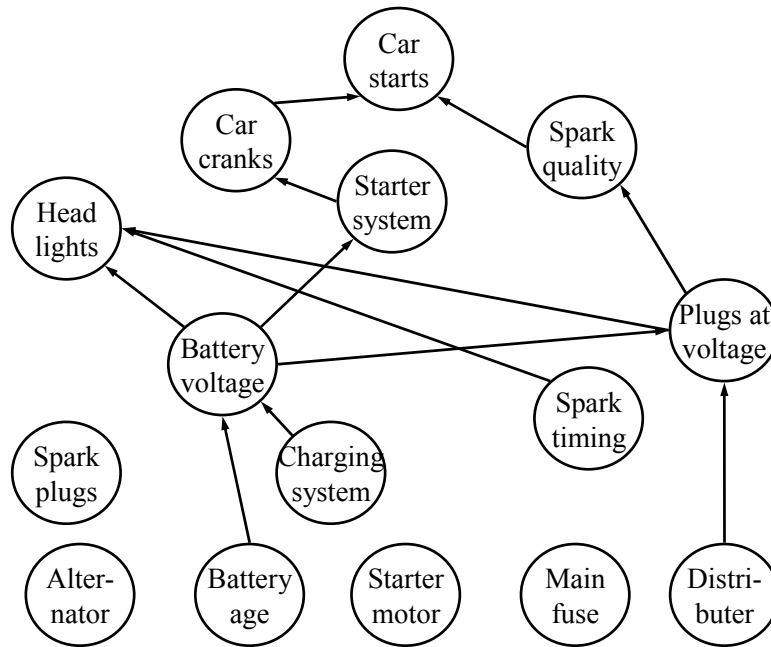


圖 4. 24 本研究方法建構之汽車診斷貝氏網路雛型

8. 檢查網路中是否有多餘的連結。此貝氏網路雛型有一組成三角形的關聯， $A = \{(\text{Battery voltage}), (\text{Plugs at voltage}), (\text{Headlights})\}$ 。計算這些關聯的 Unexplained Information 值，如圖 4. 25。消除 Unexplained Information 值較低的關聯 得到新的汽車診斷貝氏網路雛型如圖 4. 26。

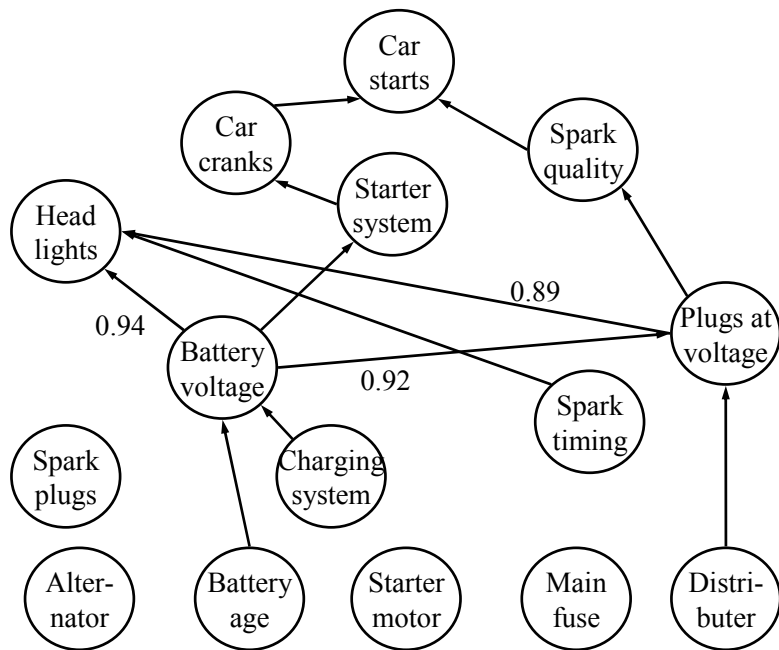


圖 4.25 計算汽車診斷貝氏網路 Unexplained Information 值

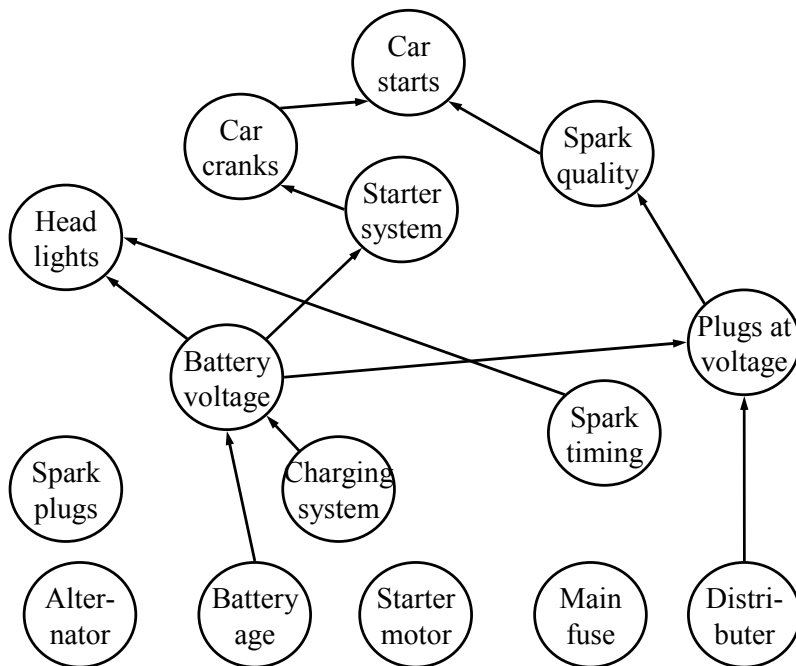


圖 4.26 去除多餘關聯後的汽車網路貝氏網路雛型

9. 利用 Norsys Netica 進行參數學習。得到最後的貝氏網路如圖 4.27。

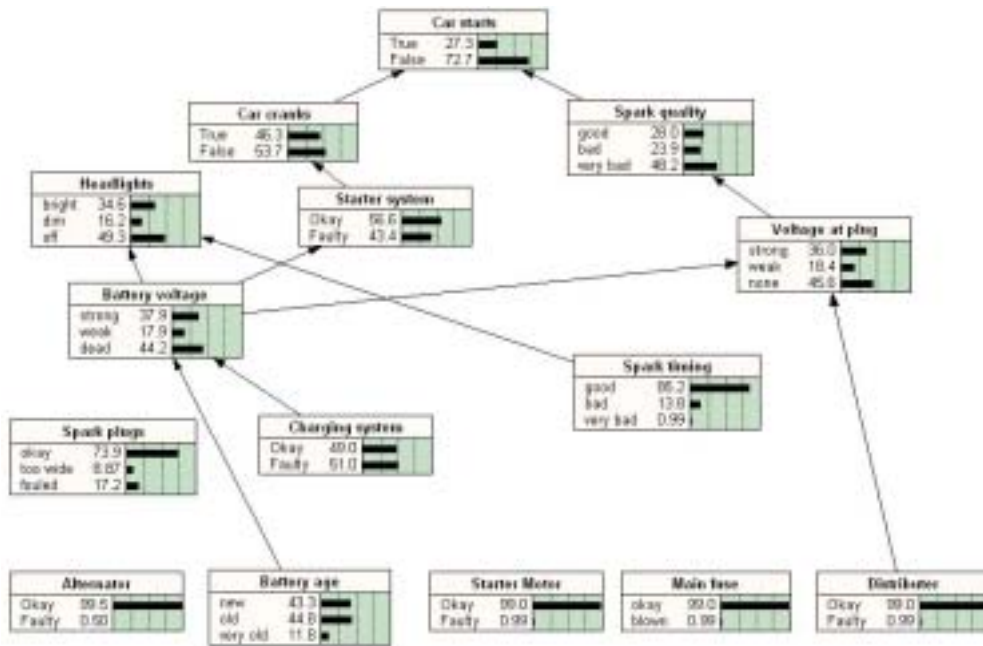


圖 4. 27 本研究方法建構之汽車診斷貝氏網路

10. 進行架構相似性分析。

- (1) 汽車診斷範例貝氏網路圖形中的關聯集合， $a = \{(Car\ Starts, Spark\ quality), (Car\ Starts, Car\ cranks), (Car\ Starts, Starter\ system), (Car\ Starts, Voltage\ at\ plug), (Car\ Starts, Battery\ voltage), (Car\ Starts, Spark\ timing), (Car\ Starts, Charging\ system), (Car\ Starts, Spark\ plugs), (Car\ Starts, Distributer), (Car\ Starts, Main\ fuse), (Car\ Starts, Starter\ Motor), (Car\ Starts, Battery\ age), (Car\ Starts, Alternator), (Spark\ quality, Voltage\ at\ plugs), (Spark\ quality, Battery\ voltage), (Spark\ quality, Charging\ system), (Spark\ quality, Spark\ plugs), (Spark\ quality, Distributer), (Spark\ quality, Main\ fuse), (Spark\ quality, Battery\ age), (Spark\ quality, Alternator), (Car\ cranks, Starter\ system), (Car\ cranks, Battery\ voltage), (Car\ cranks, Charging\ system), (Car\ cranks, Main\ fuse), (Car\ cranks, Starter\ Motor), (Car\ cranks, Battery\ age), (Car\ cranks, Alternator), (Starter\ system, Battery\ voltage), (Starter\ system, Charging\ system), (Starter\ system, Main\ fuse), (Starter\ system, Starter\ Motor), (Starter\ system, Battery\ age), (Starter\ system, Alternator), (Headlights, Battery\ voltage), (Headlights, Charging\ system), (Headlights, Battery\ age), (Headlights, Alternator), (Voltage\ at\ plug, Battery\ voltage), (Voltage\ at\ plug, Charging\ system), (Voltage\ at\ plug, Distributer), (Voltage\ at\ plug, Main\ fuse), (Voltage\ at\ plug, Battery\ age),$

(Voltage at plug, Alternator), (Battery voltage, Charging system), (Battery voltage, Battery age), (Battery voltage, Alternator), (Spark timing, Distributer)} , X=48。

(2) 本研究方法建構之汽車診斷貝氏網路中的關聯集合, $b = \{(Car\ Starts, Spark\ quality), (Car\ Starts, Car\ cranks), (Car\ Starts, Starter\ system), (Car\ Starts, Voltage\ at\ plug), (Car\ Starts, Battery\ voltage), (Car\ Starts, Charging\ system), (Car\ Starts, Battery\ age), (Spark\ quality, Voltage\ at\ plugs), (Spark\ quality, Battery\ voltage), (Spark\ quality, Charging\ system), (Spark\ quality, Distributer), (Spark\ quality, Battery\ age), (Car\ cranks, Starter\ system), (Car\ cranks, Battery\ voltage), (Car\ cranks, Charging\ system), (Car\ cranks, Battery\ age), (Starter\ system, Battery\ voltage), (Starter\ system, Charging\ system), (Starter\ system, Battery\ age), (Headlights, Battery\ voltage), (Headlights, Spark\ timing), (Headlights, Charging\ system), (Headlights, Battery\ age), (Voltage\ at\ plug, Battery\ voltage), (Voltage\ at\ plug, Charging\ system), (Voltage\ at\ plug, Distributer), (Voltage\ at\ plug, Battery\ age), (Battery\ voltage, Charging\ system), (Battery\ voltage, Battery\ age)\}$, Y=28。

(3) 共同關聯集合 $\{(Car\ Starts, Spark\ quality), (Car\ Starts, Car\ cranks), (Car\ Starts, Starter\ system), (Car\ Starts, Voltage\ at\ plug), (Car\ Starts, Battery\ voltage), (Car\ Starts, Charging\ system), (Car\ Starts, Battery\ age), (Spark\ quality, Voltage\ at\ plugs), (Spark\ quality, Battery\ voltage), (Spark\ quality, Charging\ system), (Spark\ quality, Distributer), (Spark\ quality, Battery\ age), (Car\ cranks, Starter\ system), (Car\ cranks, Battery\ voltage), (Car\ cranks, Charging\ system), (Car\ cranks, Battery\ age), (Starter\ system, Battery\ voltage), (Starter\ system, Charging\ system), (Starter\ system, Battery\ age), (Headlights, Battery\ voltage), (Headlights, Charging\ system), (Headlights, Battery\ age), (Voltage\ at\ plug, Battery\ voltage), (Voltage\ at\ plug, Charging\ system), (Voltage\ at\ plug, Distributer), (Voltage\ at\ plug, Battery\ age), (Battery\ voltage, Charging\ system), (Battery\ voltage, Battery\ age)\}$, m=27。

(4) Jaccard 係數為 $28/(48+28-27)=0.5714=57.14\%$ 。兩個貝氏網路的架構相似性為 57.14%。

11. 本研究方法建構之汽車診斷貝氏網路, 與原本範例汽車診斷貝氏網路。在架構相似度分析上, 遺漏了許多關聯。只能描述範例網路變數 57.14

%關聯。因此要進行關鍵節點分析，測試網路中存在的節點，能否具備足夠的代表性。

- (1) 汽車診斷的決策點為 Car starts，狀態值 False 機率高於狀態值 True 的機率，因此鎖定 Car starts 狀態值在 True，觀察其他節點的機率變化。
- (2) 汽車診斷範例貝氏網路，在 Car starts 狀態值為 True 的圖形如圖 4.28。
- (3) 本研究方法建構之汽車診斷貝氏網路，在 Car starts 狀態值為 True 的圖形如圖 4.29。

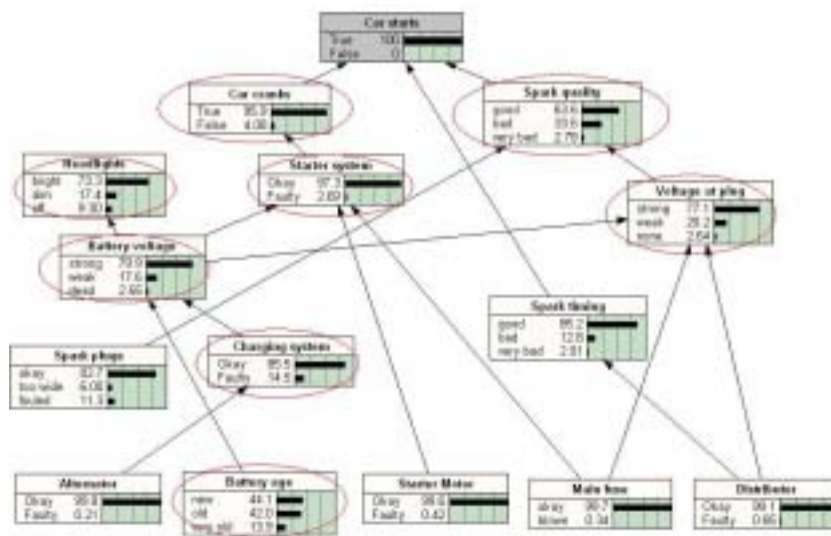


圖 4.28 範例汽車診斷貝氏網路關鍵節點

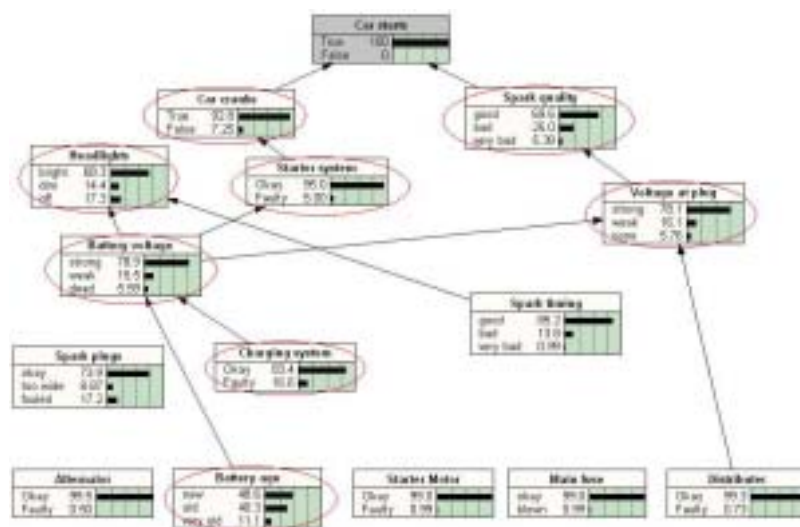


圖 4.29 本研究方法建構之汽車診斷貝氏網路關鍵節點

(4) 從兩圖中看出，Spark quality, Car cranks, Starter system, Headlights, Voltage at plug, Battery voltage, Charging system, Battery age，這 8 個節點在 Car starts 為 True 的情形時，節點狀態值的機率分佈有明顯的改變。顯示出在汽車狀態為正常時，這 8 個變數也是為正常。當汽車狀態為異常時，這 8 個變數異常機率明顯變高。因此，本研究方法建構出的貝氏網路，與原本的範例貝氏網路。均有相同的關鍵節點。

12. 利用 Norsys Netica 產生新的 400 筆資料，進行資料結構相似性分析。

(1) $H_0 : T^+ - T^- = 0$

$H_1 : T^+ - T^- > 0$

(2) 利用統計分析軟體 SPSS，進行魏克森符號檢定。雙尾檢定，顯著水準 0.025。得到 $p=0.952 > 0.025$ 。如圖 4.30。

(3) 兩個網路的資料結構是相似的。

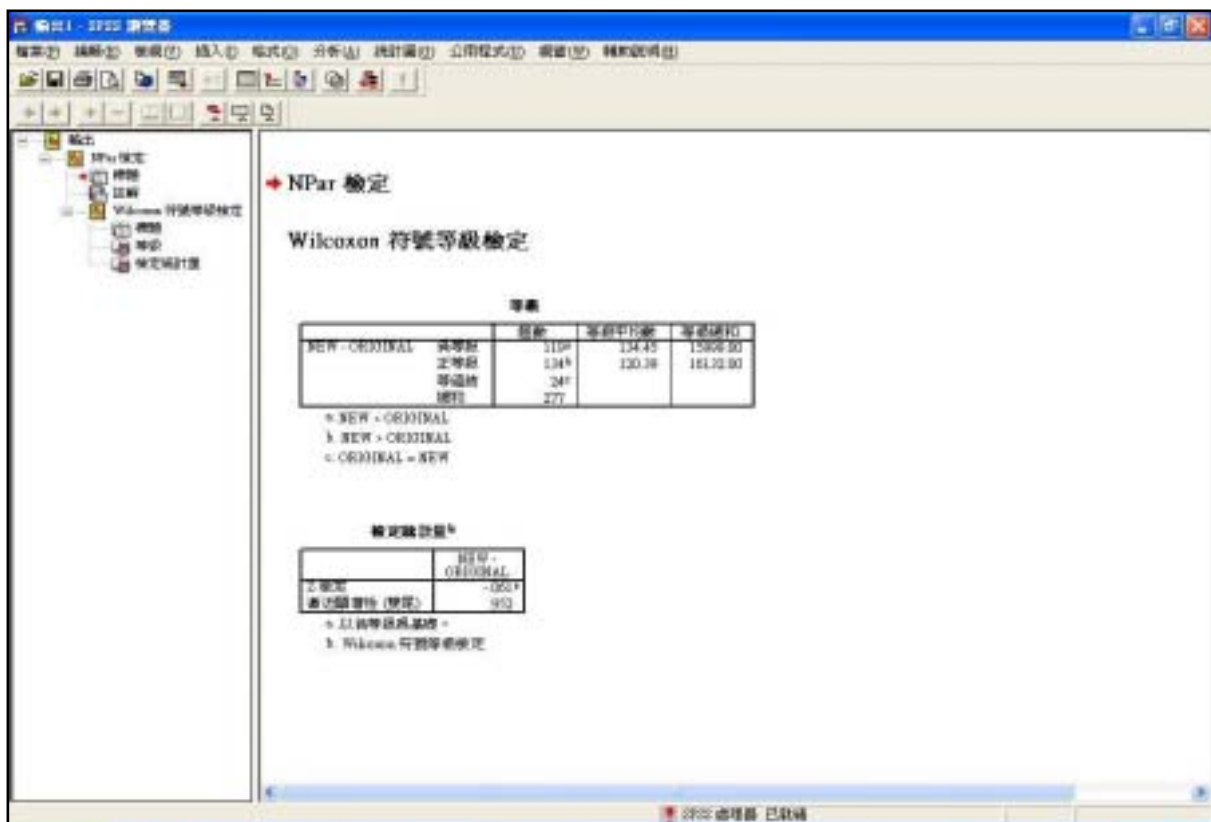


圖 4.30 本研究建構之汽車診斷貝氏網路魏克森符號檢定

本研究方法建構之汽車診斷貝氏網路，與原本範例汽車診斷貝氏網

路。在結構相似程度上，本研究方法可表達原本變數間 57.14% 關聯。但經關鍵節點相似性分析後，本研究方法建構之汽車診斷貝氏網路，能得到與範例網路相同的 8 個關鍵節點 Spark quality, Car cranks, Starter system, Headlights, Voltage at plug, Battery voltage, Charging system, Battery age。故在架構相似性上仍具其代表性。

在參數學習後，產生的新資料與原始資料能通過魏克森符號檢定，故在資料結構上，本研究所建構之汽車診斷貝氏網路足以描述原本的貝氏網路。

4.3.2 K2 演算法建構之汽車錯誤診斷貝氏網路

本節將以相同建構資料，利用 K2 演算法建構警鈴貝氏網路，之後進行架構相似性、資料結構相似性以及關鍵節點相似性分析。進行步驟如下：

1. 給定一組相同的節點排序，Alternator, Battery age, Starter Motor, Main fuse, Distributer, Spark plugs, Charging system, Spark timing, Battery voltage, Voltage at plugs, Headlights, Starter system, Car cranks, Spark Quality, Car starts。
2. 匯入建構資料至貝氏網路應用軟體—Bayesware Discovery，得到貝氏網路雛型如圖 4. 31。

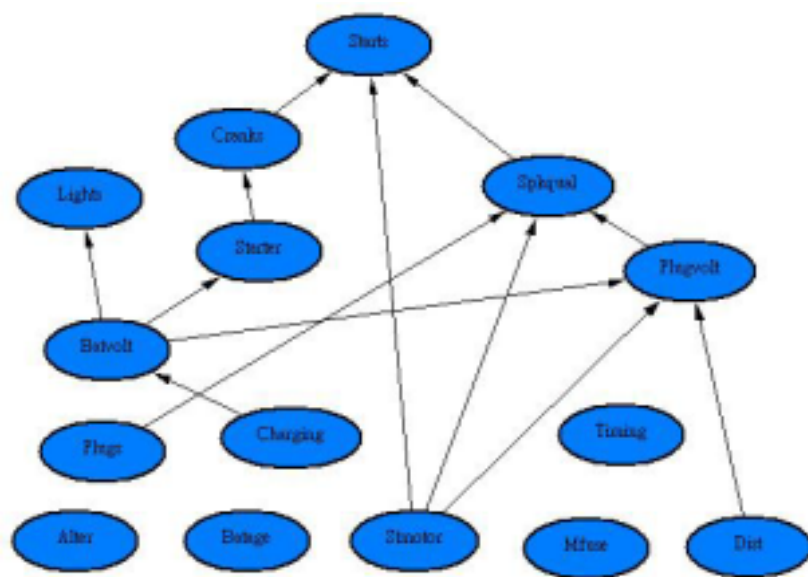


圖 4. 31 K2 演算法建構之汽車診斷貝氏網路雛型

3. 步驟 2 得到的貝氏網路雛型，搭配未參與建構資料，利用 Norsys Netica 完成參數學習。得到最終警鈴之貝氏網路如圖 4. 32。

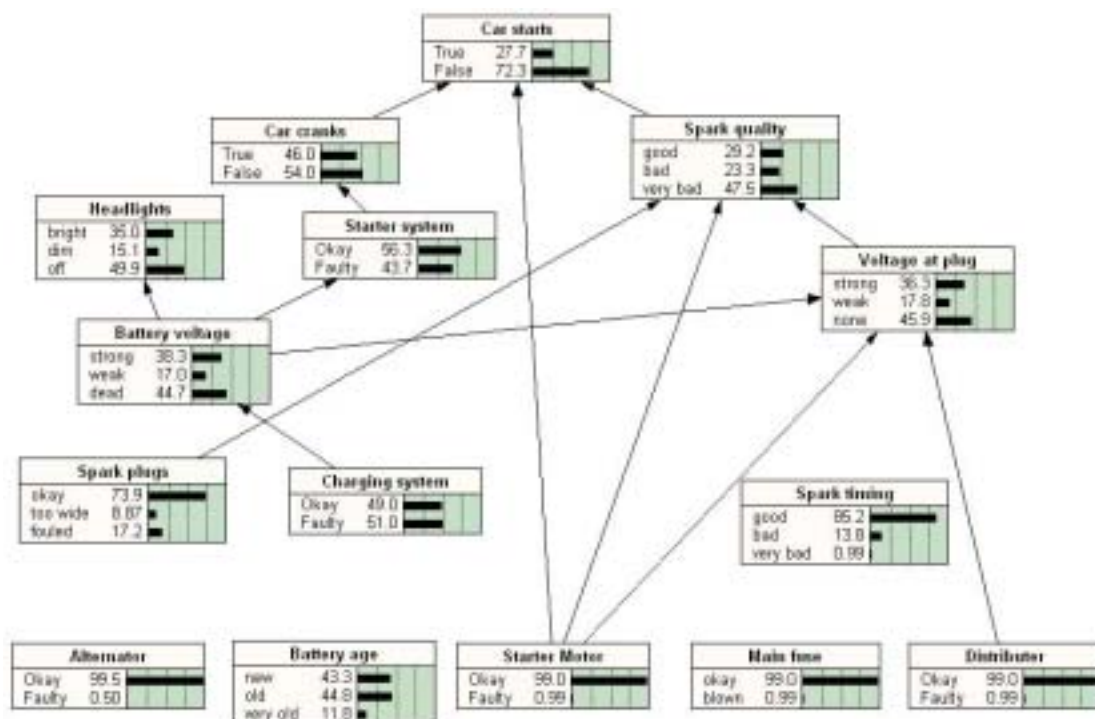


圖 4. 32 K2 演算法建構之汽車診斷貝氏網路

4. 架構相似性。

- (1) 汽車診斷範例貝氏網路圖形中的關聯集合， $a = \{(Car\ Starts, Spark\ quality), (Car\ Starts, Car\ cranks), (Car\ Starts, Starter\ system), (Car\ Starts, Voltage\ at\ plug), (Car\ Starts, Battery\ voltage), (Car\ Starts, Spark\ timing), (Car\ Starts, Charging\ system), (Car\ Starts, Spark\ plugs), (Car\ Starts, Distributer), (Car\ Starts, Main\ fuse), (Car\ Starts, Starter\ Motor), (Car\ Starts, Battery\ age), (Car\ Starts, Alternator), (Spark\ quality, Voltage\ at\ plugs), (Spark\ quality, Battery\ voltage), (Spark\ quality, Charging\ system), (Spark\ quality, Spark\ plugs), (Spark\ quality, Distributer), (Spark\ quality, Main\ fuse), (Spark\ quality, Battery\ age), (Spark\ quality, Alternator), (Car\ cranks, Starter\ system), (Car\ cranks, Battery\ voltage), (Car\ cranks, Charging\ system), (Car\ cranks, Main\ fuse), (Car\ cranks, Starter\ Motor), (Car\ cranks, Battery\ age), (Car\ cranks, Alternator), (Starter\ system, Battery\ voltage), (Starter\ system, Charging\ system), (Starter\ system, Main\ fuse), (Starter\ system, Starter$

Motor), (Starter system, Battery age), (Starter system, Alternator), (Headlights, Battery voltage), (Headlights, Charging system), (Headlights, Battery age), (Headlights, Alternator), (Voltage at plug, Battery voltage), (Voltage at plug, Charging system), (Voltage at plug, Distributer), (Voltage at plug, Main fuse), (Voltage at plug, Battery age), (Voltage at plug, Alternator), (Battery voltage, Charging system), (Battery voltage, Battery age), (Battery voltage, Alternator), (Spark timing, Distributer)} , X=48。

(2) K2 演算法建構之汽車診斷貝氏網路 , $b=\{(Car\ Starts, Spark\ quality), (Car\ Starts, Car\ cranks), (Car\ Starts, Starter\ system), (Car\ Starts, Voltage\ at\ plug), (Car\ Starts, Battery\ voltage), (Car\ Starts, Charging\ system), (Car\ Starts, Spark\ plugs), (Car\ Starts, Distributer), (Car\ Starts, Starter\ Motor), (Spark\ quality, Voltage\ at\ plugs), (Spark\ quality, Battery\ voltage), (Spark\ quality, Charging\ system), (Spark\ quality, Spark\ plugs), (Spark\ quality, Distributer), (Spark\ quality, Starter\ Motor), (Car\ cranks, Starter\ system), (Car\ cranks, Battery\ voltage), (Car\ cranks, Charging\ system), (Starter\ system, Battery\ voltage), (Starter\ system, Charging\ system), (Headlights, Battery\ voltage), (Headlights, Charging\ system), (Voltage\ at\ plug, Battery\ voltage), (Voltage\ at\ plug, Charging\ system), (Voltage\ at\ plug, Distributer), (Voltage\ at\ plug, Starter\ motor), (Battery\ voltage, Charging\ system)\}$, Y=27。

(3) 共同的關聯集合 $\{(Car\ Starts, Spark\ quality), (Car\ Starts, Car\ cranks), (Car\ Starts, Starter\ system), (Car\ Starts, Voltage\ at\ plug), (Car\ Starts, Battery\ voltage), (Car\ Starts, Charging\ system), (Car\ Starts, Spark\ plugs), (Car\ Starts, Distributer), (Car\ Starts, Starter\ Motor), (Spark\ quality, Voltage\ at\ plugs), (Spark\ quality, Battery\ voltage), (Spark\ quality, Charging\ system), (Spark\ quality, Spark\ plugs), (Spark\ quality, Distributer), (Car\ cranks, Starter\ system), (Car\ cranks, Battery\ voltage), (Car\ cranks, Charging\ system), (Starter\ system, Battery\ voltage), (Starter\ system, Charging\ system), (Headlights, Battery\ voltage), (Headlights, Charging\ system), (Voltage\ at\ plug, Battery\ voltage), (Voltage\ at\ plug, Charging\ system), (Voltage\ at\ plug, Distributer), (Battery\ voltage, Charging\ system)\}$, m=25。

(4) Jaccard 係數為 $25/(48+27-25)=0.5=50\%$ 。兩個貝氏網路的架構相似性為 50%。

5. K2 演算法建構之汽車診斷貝氏網路，與原本範例汽車診斷貝氏網路。在架構相似度分析上，遺漏了許多關聯。只能描述範例網路變數 57.14 % 關聯。因此要進行關鍵節點分析，測試網路中存在的節點，能否具備足夠的代表性。

- (1) 汽車診斷的決策點為 Car starts，狀態值 False 機率高於狀態值 True 的機率，因此鎖定 Car starts 狀態值在 True，觀察其他節點的機率變化。
- (2) 汽車診斷範例貝氏網路，在 Car starts 狀態值為 True 的圖形如圖 4.28。
- (3) 本研究方法建構之汽車診斷貝氏網路，在 Car starts 狀態值為 True 的圖形如圖 4.33。

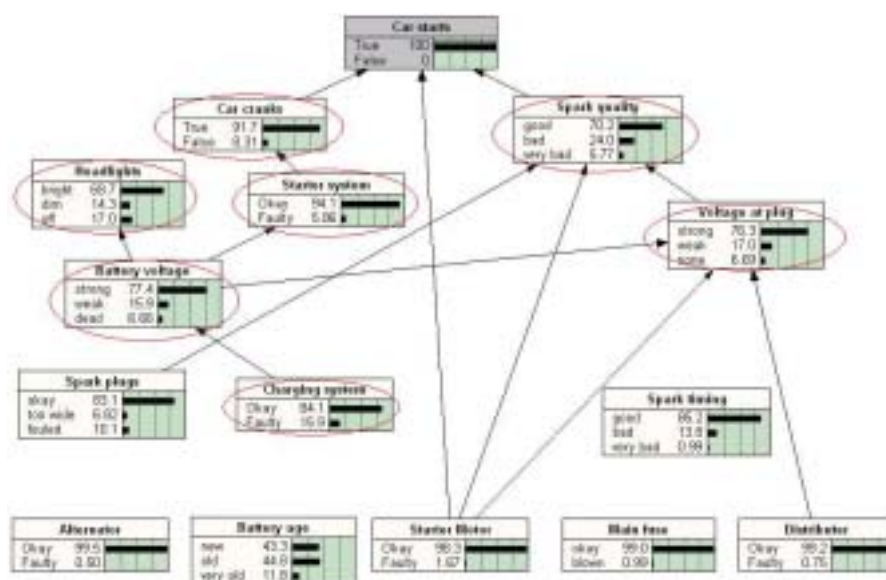


圖 4.33 K2 演算法建構之汽車診斷貝氏網路關鍵節點

- (4) 從兩圖中看出，K2 演算法建構之汽車診斷貝氏網路，在 Car starts 為 True 時，只出現了 7 個關鍵節點，少了 Battery age。因此 K2 演算法無法將範例網路中的關鍵節點描述出來。
6. 利用 Norsys Netica 產生新的 400 筆資料，進行資料結構相似性分析。
- (1) $H_0 : T^+ - T^- = 0$
 $H_1 : T^+ - T^- > 0$
 - (2) 利用統計分析軟體 SPSS，進行魏克森符號檢定。雙尾檢定，顯著

水準 0.025。得到 $p=0.661>0.025$ 。如圖 4. 34。

(3) 兩個網路的資料結構是相似的。

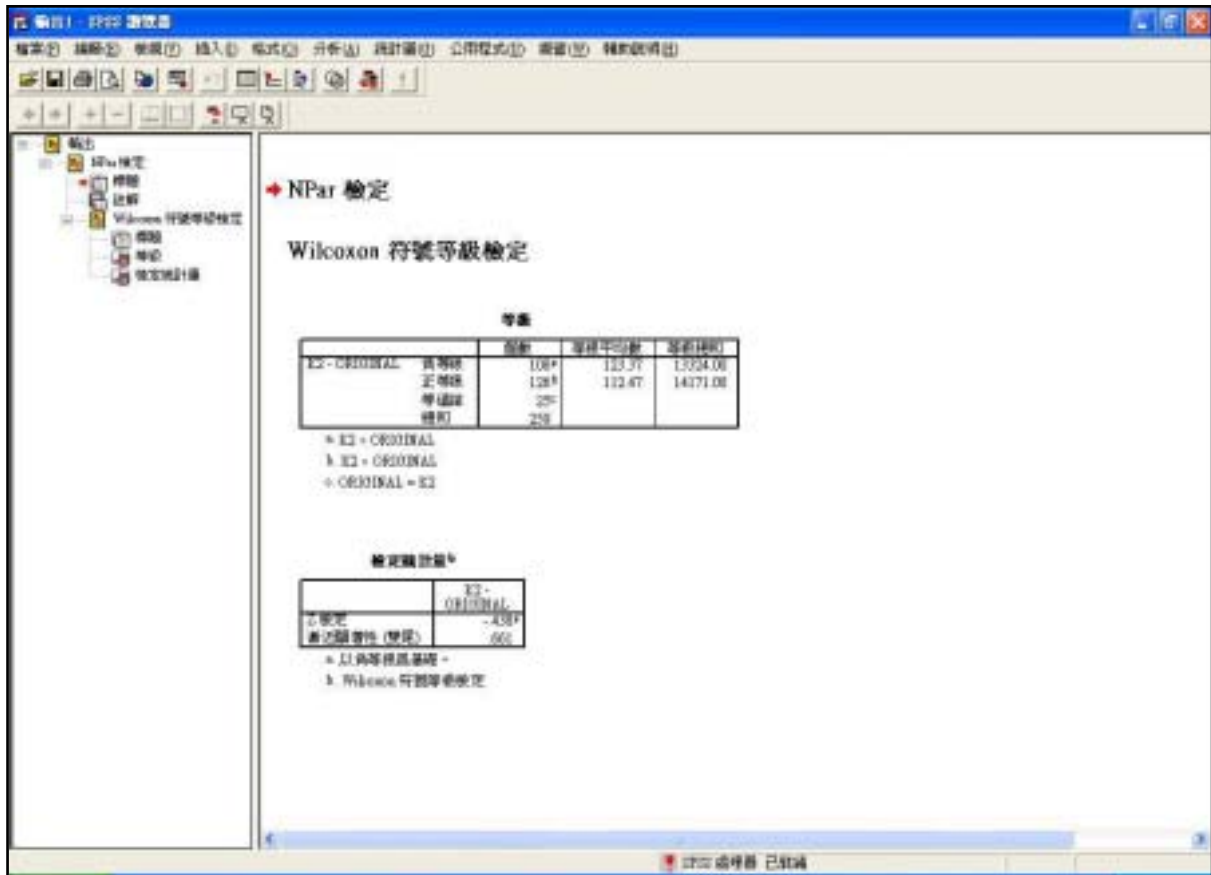


圖 4. 34 K2 演算法建構之汽車診斷貝氏網路魏克森符號檢定

K2 演算法建構之汽車診斷貝氏網路，與原本範例汽車診斷貝氏網路。在結構相似程度上，K2 演算法可表達原本變數間 50% 關聯。而經關鍵節點相似性分析後，K2 演算法建構之汽車診斷貝氏網路，只能得到與範例網路 7 個相同的關鍵節點 Spark quality, Car cranks, Starter system, Headlights, Voltage at plug, Battery voltage, Charging system，少了 Battery age。故在架構相似性上不具其代表性。

在參數學習後，產生的新資料與原始資料能通過魏克森符號檢定，故在資料結構上，K2 演算法所建構之汽車診斷貝氏網路足以描述原本的貝氏網路。

4.4 本章總結

本研究方法及 K2 演算法，對於三個範例貝氏網路的建構結果如下述。建構績效比較如表 4.8，表 4.9。

1. 警鈴貝氏網路：本研究方法及 K2 演算法，在此範例均有相同的表現。在網路架構方面，都能 100% 表示範例中變數的關聯。在資料結構方面，新資料都能通過與原始資料的魏克森符號檢定。
2. 動物分類貝氏網路：本研究方法在網路架構方面，比 K2 演算法更能表示範例中變數的關聯。在資料結構方面，兩者皆因範例中變數間的關係較為確定性，導致新資料與原始資料不能通過魏克森符號檢定。
3. 汽車診斷貝氏網路：本研究方法在網路架構方面，比 K2 演算法更能表示範例中變數的關聯。在資料結構方面，新資料都能通過與原始資料的魏克森符號檢定。

表 4.8 架構相似性比較表

建構方法	範例貝氏網路		
	警鈴	動物分類	汽車診斷
本研究方法	100%	72.72%	57.14%
k2 演算法	100%	50%	50%

表 4.9 資料結構相似性比較表

建構方法	範例貝氏網路 (p 值)		
	警鈴	動物分類	汽車診斷
本研究方法	0.570	0	0.952
K2 演算法	0.570	0	0.661

第五章 結論與建議

5.1 結論

本研究應用粗集理論，從資料中挖掘出決策法則，進一步建構成貝氏網路。因此與搜尋式和限制式演算法在建構的方法上都不盡相同。

在架構原理上，本研究方法與限制式演算法較相近，與搜尋式演算法較大差異。本研究方法與限制式相同，是利用變數間的關聯建構成貝氏網路架構。而非像搜尋式演算法，需從眾多的可能架構中選擇品質係數最好的貝氏網路架構。但本研究方法不能算是限制式演算法，因為本研究不是利用條件獨立測試建構貝氏網路架構。

根據第四章實證研究的結果，本研究在架構相似性上分別是警鈴範例 100%，動物分類範例 72.72%，汽車診斷範例 57.14%，可以達到一半以上描述問題中變數的關聯。在資料結構分析上，只有動物分類無法通過魏克森符號等級檢定，在變數機率分佈上稍嫌不足。因此，本研究方法應用在貝氏網路架構建立上，歸納出以下的重點。

1. 本研究對於變數少，架構簡單的問題，可以建構出能充分表現出問題變數關係及機率的貝氏網路，例如警鈴範例。
2. 本研究方法建構的貝氏網路架構時，會隨著變數數量變多，變數間的關聯變數雜，而變的績效不佳的情形。雖然如此，對於貝氏網路中關鍵的節點，本研究方法還是可以在貝氏網路中呈現出來。
3. 由動物分類的驗證範例中，本研究方法對於確定性分類，在參數學習之後，會衍生出許多原本不存在的事件。因此，本研究在資料結構相似性分析上無法通過魏克森符號等級檢定。此部分成為本研究方法應用上的限制。

5.2 未來研究方向

未來仍可以以本研究方法為基礎，但可以朝向下面本研究未進行的方法研究。

1. 變數排序是建構貝氏網路架構的一個重點。有些演算法需要變數排序才能建構貝氏網路架構，例如 K2 演算法和本研究方法均是。未來可以朝向從資料面取得變數排序，或者可不仰賴變數排序仍可利用本研究方法建構貝氏網路架構的方向研究。
2. 粗集理論粹取法則的過程，依賴可辨識矩陣的計算，所以資料量的多寡會受到可辨識矩陣大小而限制。對於變數多而需要大量資料的貝氏網路，可辨識矩陣變成了一種限制。未來研究方向，可朝解決粗集理論此一缺點進行。
3. 本研究方法是在靜態資料中進行貝氏網路雛型架構，未來可針對動態資料進行貝氏網路架構學習。或者可在新的證據下，可進行架構和參數的修正。

參考文獻

- [1] 林惠玲, 陳正倉, 「應用統計學」, 雙葉書廊有限公司, 民國 88 年。
- [2] 林傑斌, 「SPSS 10.0 與統計模式建構」, 文魁資訊股份有限公司, 民國 90 年。
- [3] 黃意策, 「應用資料採礦技術於貝氏架構之研究—線性代數與圖論之結合」, 碩士論文, 東海大學工業工程與經營資訊研究所, 民國 91 年 7 月。
- [4] 溫智傑, 「製造執行系統內錯誤診斷處理機制之研究」, 碩士論文, 東海大學工業工程與經營資訊研究所, 民國 92 年 1 月。
- [5] Bazan, J. G., “A Comparison of Dynamic and Non-dynamic Rough Set Methods for Extracting Laws from Decision Tables,” *Rough Sets in Knowledge Discovery*, Vol. 1, Heidelberg: Physical-Verlag, 1998.
- [6] Beynon, M., “Reducts within the Variable Precision Rough Sets Model: A Further Investigation,” *European Journal of Operational Research*, 2001.
- [7] Browne, C., Duntsch, I. and Gunther, G., “IRIS revisited: A Comparison of Discriminant and Enhanced Rough Set Data Analysis,” *Rough Sets in Knowledge Discovery*, Vol. 2, Heidelberg: Physical-Verlag, 1998.
- [8] Buntine, W., “A Guide to the Literature on Learning Probabilistic Networks From Data,” *IEEE Trans. Knowledge Data Engineering*, Vol.8, 83-90, 1996.
- [9] Castillo, E., Jose, G. M. and Ali, M. S., *Expert Systems and Probabilistic Network Models*, Springer-Verlag, New York, 1997.
- [10] Cheng, J., Russell, G., Jonathan K. and David B., “Learning Bayesian networks from data: An Information-theory Based Approach,” *Artificial Intelligence*, Vol. 137, 43-90, 2002.
- [11] Chrisman, L., “A Roadmap to Research on Bayesian Networks and other Decomposable Probabilistic Models,” Technical Report, School of Computer Science, CMU, Pittsburgh, PA, 1996.
- [12] Cowell, R.G, “When Learning Bayesian Networks from Data, Using Conditional Independence Tests is Equivalent to A Local Scoring Metric,” 17th International Conference on University in Artificial Intelligence, 2001.
- [13] Duntsch, I., Gunther, G., “Uncertainty Measures of Rough Set Prediction,” *Artificial Intelligence*, Vol. 106, 1998.
- [14] Duntsch, I., Gunther, G., “Statistical Evaluation of Rough Set Dependency Analysis,” *International Journal Human-Computer Studies*, 1997.
- [15] Duntsch, I., Gunther, G., “The Rough Set Engine GROBIAN,” <http://www.cosc.brocku.ca/~duentsch/grobian/grobian.html>, 1993.
- [16] Duntsch, I., Gunther, G., “Rough Set Data Analysis,” *Encyclopedia of Computer Science and Technology*, 2000.
- [17] Duntsch, I., Gunther, G., “Statistical Techniques for Rough Set Data Analysis,” *Rough*

- Sets: New Developments, Heidelberg: Physical Verlag, 2000.
- [18] Han, J. and Micheline K., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2001.
 - [19] Heckerman, D., "A Tutorial on Learning with Bayesian Networks," *Technical Reporter MSR-TR-95-06*, Microsoft Research, 1995.
 - [20] Krause, P. J., "Learning Probabilistic Networks," Technical Report, Philips Research Laboratories, UK, 1996.
 - [21] Lin P. D., "Survey of Algorithms used in Data Mining and Knowledge Discovery in Databases," <http://www.lasierra.edu/~dlin/classes/cpsc531/cpsc531.doc>, 1999.
 - [22] Margaritis, D., "Learning Bayesian Network Model Structure from Data," *submitted in partial fulfillment requirements for the degree of Doctor of Philosophy*, Carnegie Mellon University, Pittsburgh, 2003.
 - [23] Nakayama, H., Yuichi, H. and Renichi, I., "Rule Extraction Based on Rough Set Theory and its Application to Medical Data Analysis," *IEEE Systems, Man, and Cybernetics Conference Proceedings*, 1999.
 - [24] Norsys Software Corp., *Netica Application User's Guide Version 1.05*, <http://www.norsys.com/index.html>, 1997.
 - [25] Pawlak, Z., *Rough Sets. Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publisher, Dordrecht, Netherlands, 1991.
 - [26] Pawlak, Z., "Rough Set Approach to Knowledge-based Decision Support," *European Journal of Operational Research*, 1997.
 - [27] Pearl, J., "Graphical Models for Probabilistic and Causal Reasoning," *The Computer Science and Engineering Handbook*, 697-714, 1997.
 - [28] Pearl, J. and T.S. Verma, "An Algorithm for Deciding If A Set of Observed Independencies Has A Causal Explanation," 8th International Conference on Uncertainty in Artificial Intelligence, Stanford, CA, 1992.
 - [29] Russell, S.J. and Peter N., *Artificial Intelligence: a modern approach*, Prentice-Hall, 1995.
 - [30] Rousseau R., "Jaccard Similarity Leads to the Marczwskisteinaus Topology for Information Retrieval," *Information Processing and Management*, Vol. 34, 87-94, 1998.
 - [31] Singh, M. and Macro, V., "Construction of Bayesian Network Structures from Data: A Brief Survey and an Efficient Algorithm," *International Journal of Approximate Reasoning*, Vol. 12, 111-131, 1995.
 - [32] Ślęzak, D., "Rough Set Reduct Networks," submitted to RSSC'97, 1997.
 - [33] Ślęzak, D., Szczuka, M. and Tsumoto, S., "An Application of Reduct Networks to Medicine," submitted to RSSC'97, 1997.
 - [34] Walczak, B. and D.L. Massart, "Rough sets theory," *Chemometrics and Intelligent Laboratory Systems*, Vol. 47, 1999.