

第一章 緒論

1.1 研究動機與目的

隨著現在科技的飛速發展，使機器不斷地精密化和系統化。機器的運行複雜程度越高，精密程度就越低，發生異常現象的程度也越高。在複雜的機器運作之下，往往伴隨著龐大的數據資料的產生。要在這些繁瑣龐大的資料中劫取出有價值的資訊，已成為研究上與實務上的重要議題。

我們以晶圓製程為例，在晶圓製程過程中有著數百或數千的製程步驟，且每一個步驟又可能會有數個測量項目，因而產生了龐大的晶圓數據資料，可能是某個步驟中金屬層的厚度，也可能在某個步驟中落在晶圓上的微粒的大小或塵埃的個數，可能為某處的電阻值，或是某個步驟中所處的溫度值或壓力值等等的各式各樣的測試數據資料，若將每一個項目都視為一個可能影響製程良率的變數，那麼每一塊晶圓經過一個完整的製程過程後，我們可能可以得到一個包含數百或數千個測試項目的數據資料。

單單一片晶圓極可能有個數百或數千變數的測試資料，當晶圓一多時可想而知資料量馬上暴增。由於製程過程複雜、變數眾多，工程師往往無法從龐大的資料中察覺出可能導致製成異常的原因或是歸納出造成產品異常的因素，而且工程師可能也無法在短時間內有效地縮小可能造成製程異常的原因範圍。

從晶圓經過機台加工而產生的數據資料中，如何找出可能導致製程異常的原因或造成產品異常的因素？如何降低製程變異，以達到品質的提昇？這種問題為診斷偵測的問題，也為不確定性的問題，這而這種問題就是本論文所要探討的。

1.2 研究方法

能夠有效地處理診斷偵測或不確定性的問題，最常見的方法有專家系統(expert

system)和貝氏網路(bayesian network)這兩個方法。

專家系統是一個訊息處理的系統[10]，它處理現實世界中提出的需要由專家來分析和判斷的複雜問題，如果這個系統所要解決的問題和專家解決的問題可以比較的話，它應該得出和專家相近的結論，也就是說一個專家系統的任務，就是要解決某個專門領域中需要專家才能解決的問題。因此，它必需力求收集足夠的專家知識。它完成任務的方法，是借助電腦模擬專家解決問題的方法來實現的，也就是說專家系統必須具有從專家那裡獲得的知識，且具有和專家同等的處理能力。

建立一個專家系統的處理，通常稱為知識工程(knowledge engineering)，在專家系統建立著之間，牽入一種特別的交互作用的組成，稱為知識工程師(knowledge engineer)，有一個或多個屬於某些問題的領域的人類專家，知識工程師從人類專家對問題的解決的程序、謀略和規則抽出，將這種知識注入所建立的專家系統裡，如圖1-1所示。

建立專家系統的目的，在於通過電腦實現某領域專家能力的工作。目前，專家系統已在許多領域得到應用，如醫學、化學、地質石油探勘、氣象和其他領域等等，承擔著不同的任務。如果不考慮領域的特性而按照專家系統本身求解的特點，可加專家系統分為：解釋型、診斷型、監視型、維護型、控制型、預測型、規劃型、設計型，如果將它們進一步歸納，大致上可以分為三類：診斷型、控制型、設計型，而其中的診斷型即可處理我們上述的診斷預測的問題。

貝氏網路由圖形理論(graph theory)和機率理論(probability theory) 這兩種不同數學領域的理論所組成的。藉由圖形理論得到圖形中節點跟邊線的結構，此圖形也可將變數和變數間的關係表現出來。我們也可以由機率理論中的各種理論得到我們所需要的機率，再以機率來解決許多的問題。

貝氏網路在處理不確定事物的問題上是一個方便又好用的工具，也可以處理上面所面臨的問題，其用途與專家系統有點接近，在後面將作詳細介紹。

貝氏網路與專家系統有一些的不同，其不同的地方如下：

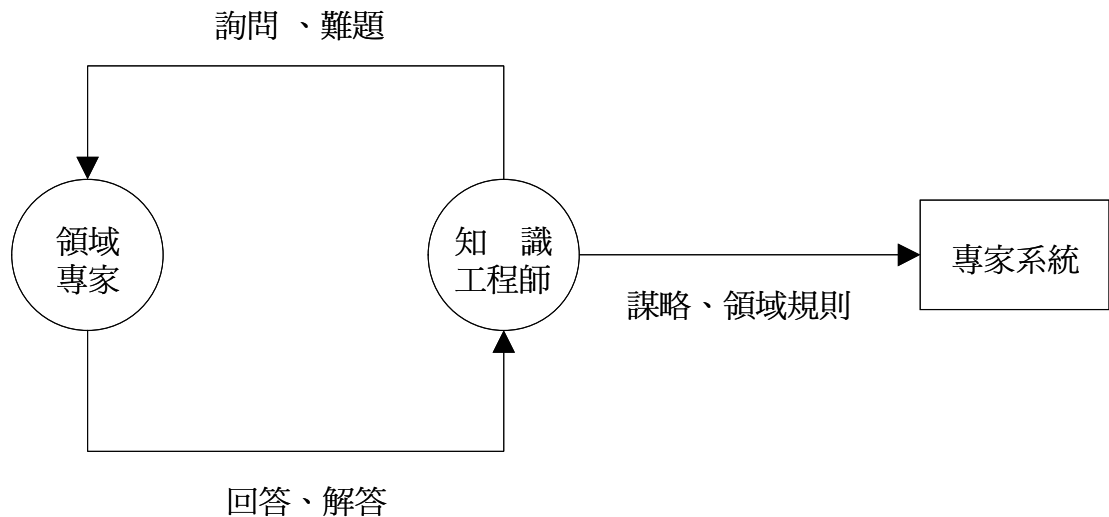


圖1-1 知識工程：專家知識轉化為電腦程序

- (1) 專家系統是由知識工程師與人類專家詢問探討中得到許多專家知識，因而建立規則基底(rule-base)，之後才能建立專家系統。
- (2) 貝氏網路是從數據資料中經由圖形理論得到貝氏網路的基本架構，再搭配機率理論建立貝氏網路。

也就是說：最主要的差別即在專家知識跟知識工程師上，專家系統著重於知識基底(knowledge-base)上，而貝氏網路著則重於資料基底(data-base)上，也因此我們選擇比較方便貝氏網路，即使沒有某個領域的知識，我們只需利用貝氏網路也可處理這些資料。

本論文主要是藉由[9]的方法將龐大的晶圓數據資料先經由特徵選取(feature selection)的處理以減少資料量，再藉由圖形化模型(graphical model, GM)的方法建立出無方向的圖形化模型，藉由[8]的方法來排序節點(ordering nodes)，無方向的圖形搭配排序節點後所得的排序建立方向性的圖形化模型，即得到貝氏網路的基本結構。若將經過特徵選取後數據經由k-means分群分析(k-means cluster analysis)將數據量化，根據貝氏網路的基本結構知道變數間的關係，利用這些關係將量化後的數據經由統計計算出各個條件機率(conditional probability)。基本結構跟條件機率組

合成一個貝氏網路，以提供我們遇到問題時能夠加以運用，診斷出可能導致製程異常的原因或造成產品異常的因素，進而降低製程變異，以預測出品質的提昇與否。

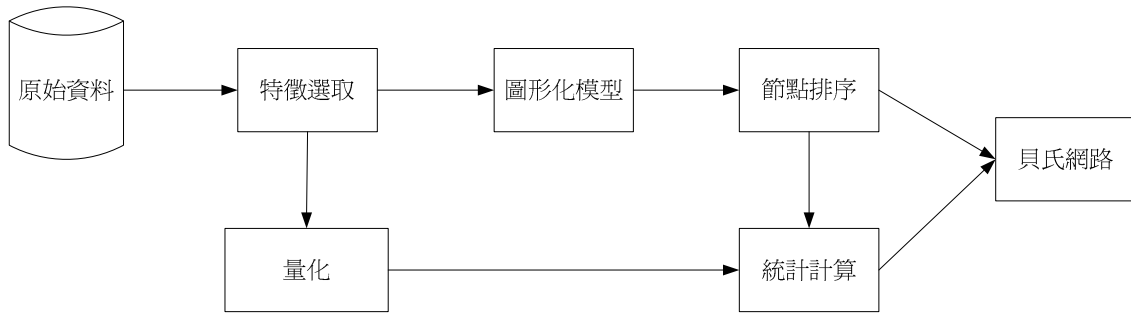


圖1-2 研究方法

圖1-2為本論文的研究方法，藉由[9]特徵選取的方法和建立圖形化模型的方法得到一個無方向的圖形。

節點排序是由變數Y(被解釋變數)跟其他未排序的變數 (X_1, X_2, \dots, X_n) (解釋變數)間的共同訊息量(mutual information entropy) $I(Y | X_i)$ 來做排序，藉由式子 (1.1) 我們可以得到Y跟其他變數間的共同訊息量，將得到的訊息量由大到小作排序可以得到 $(X_{(1)}, X_{(2)}, \dots, X_{(n)})$ ，其中 $X_{(1)}$ 是Y 跟節點 $X_{(1)}$ 間有著最大的共同訊息量。

$$I(Y | X) = H(Y) - H(Y | X) \quad (1.1)$$

其中 $H(Y) = -\sum_y P(y) \log P(y)$ ， $H(Y | X) = -\sum_y \sum_x P(x, y) \log P(y | x)$ 。

根據節點排序後的所得到的排序可以將無方向的圖形化模型（圖1-3 (a)）轉換成方向性非循環圖形模型（圖1-3 (b)），在方向性非循環圖形中節點一定是經過排序的，相連的節點間，排序比較前面的節點是排序比較後面的節點的父（母）節點，如圖1-3 (a) 經由節點排序得到 (B, D, A, C, E, F) ，A和B兩個相連的節點中B在節點排序中比A前面，所以節點B為節點A的父（母）節點。

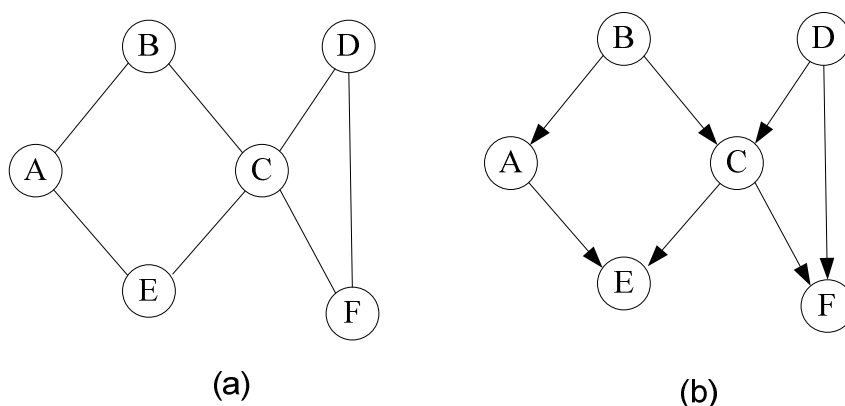


圖1-3 (a) 無方向的圖形
(b) 方向性非循環圖形

爲了將經由特徵選取過後的資料分類，可以藉由k-means分群理論來分類別。由類別值當作資料的數值以達到量化的目地，如表1-1。

- (1) 從資料中隨機選擇幾個數值當分群的平均值或中心。
- (2) 將比較接近某個分群平均值的資料納入此分群中。
- (3) 計算每一個分群的平均值。
- (4) 回第二步驟，直到收斂爲止。

樣本	數值
1	0.755725
2	0.914474
3	0.88488
4	0.93295
.	.
.	.
.	.
.	.
152	0.698225

K-means分群	
分群數量：3	

分群	中心
1	0.620338
2	0.786138
3	0.966858

樣本	類別
1	2
2	3
3	3
4	3
.	.
.	.
.	.
.	.
152	1

表 1-1 量化

1.3 論文架構

本論文共分爲五章。第二章爲貝氏網路簡介，會對貝氏網路作一些簡單的介紹。第三章爲聯合樹演算法，將說明如何由貝氏網路來建立聯合樹模型。第四章爲應用說明與討論實驗結果。第五章爲結論。



第二章 貝氏網路簡介

2.1 機率

2.1.1 貝氏原理

下面發生的獨立事件跟機率的乘法原則非常相似：

$$P(A, B) = P(A)P(B) \quad (2.1)$$

其中 $P(A, B)$ 為 A 和 B 同時發生的機率。實際上有些特別的例子是相依事件的乘法運算，其中 $P(A|B)$ 為得知 B 已經發生的情形之下 A 的機率：

$$P(A, B) = P(A)P(B|A) \quad (2.2)$$

$$P(A, B) = P(B)P(A|B) \quad (2.3)$$

因此我們可以經由 (2.2) 式和 (2.3) 式得到：

$$P(A)P(B|A) = P(B)P(A|B) \quad (2.4)$$

我們可以將 (2.4) 式作一些改變得到一個簡單的貝氏原理：

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (2.5)$$

2.1.2 貝氏連鎖原理

當我們在得知事件 I 發生的情形之下，想要計算 $P(A, B)$ 的機率，我們可以將式子寫成 $P(A, B|I)$ ：

$$P(A, B|I) = P(A|I)P(B|A, I) \quad (2.6)$$

$$P(A, B|I) = P(B|I)P(A|B, I) \quad (2.7)$$

所以經由 (2.6) 式和 (2.7) 式得到另一形式的貝氏原理：

$$P(A|B,I) = \frac{P(A|I)P(B|A,I)}{P(B|I)} \quad (2.8)$$

藉由使用乘法原則，我們可以將幾個機率連鎖在一起，例如：

(1) 當 E_1 ， E_2 和 I 都發生的情形之下， H 的機率為多少：

$$P(H|E_1, E_2, I) = \frac{P(H|I)P(E_1, E_2|H, I)}{P(E_1, E_2|I)} \quad (2.9)$$

(2) 當 E_1 ， E_2 ， E_3 和 I 都發生的情形之下， H 的機率為多少：

$$P(H|E_1, E_2, E_3, I) = \frac{P(H|I)P(E_1, E_2, E_3|H, I)}{P(E_1, E_2, E_3|I)} \quad (2.10)$$

其中值得我們注意的是

$$\begin{aligned} P(E_1, E_2, E_3|I) &= P(E_1|E_2, E_3, I)P(E_2, E_3|I) \\ &= P(E_1|E_2, E_3, I)P(E_2|E_3, I)P(E_3|I) \end{aligned} \quad (2.11)$$

，這個式子可以使用上面所講的式子 (2.6) 式跟 (2.7) 式計算出兩個機率值。



2.1.3 條件獨立

如果 A 沒有影響到 B ，且 B 也沒有影響到 A ，則可以說兩者 A 和 B 互相獨立，即有以下特性：

$$P(A, B) = P(A)P(B) \quad (2.12)$$

$$P(A|B) = \frac{P(A, B)}{P(B)} = P(A), A \perp B \quad (2.13)$$

$$P(B|A) = \frac{P(B, A)}{P(A)} = P(B), B \perp A \quad (2.14)$$

因為有了上面的特性，所以我們才會說 A 和 B 互相獨立。

我們擁有 A ， B 和 C 的變數，當我們說得知 C 的情況之下 A 和 B 為條件獨立時：

$$P(A|C, B) = P(A|C) \quad (2.15)$$

若式子 (2.9) 中得知 I 的情形之下 E_1 與 E_2 為條件獨立，則：

$$\begin{aligned}
P(H | E_1, E_2, I) &= \frac{P(H | I)P(E_1 | E_2, H, I)P(E_2 | I)}{P(E_1 | E_2, I)P(E_2 | I)} \\
&= \frac{P(H | I)P(E_1 | H, I)P(E_2 | I)}{P(E_1 | I)P(E_2 | I)} \quad (2.16)
\end{aligned}$$

2.1.4 聯合機率

我們可以用一些條件機率來表示一個聯合機率：

$$P(A, B, C, D) = P(A | B, C, D)P(B | C, D)P(C | D)P(D) \quad (2.17)$$

其一般形式為：

$$P(V_1, V_2, \dots, V_n) = \prod_{i=1}^n P(V_i | V_{i-1}, \dots, V_1) \quad (2.18)$$

聯合機率的優點即是當我們擁有一個聯合機率時，我們就可以問一些跟不確定事物有關的重要問題；缺點為不太可能創造出超過 10 或多一點變數的聯合機率，因為當我們要建立聯合機率表格時會有幾千或幾萬筆的數據，也因此它適用的範圍也被縮小了。



2.2 貝氏網路

貝氏網路由兩個部分所組成：一個是結構部分一個是數值部分。結構部分為一個方向性非循環圖形(directed acyclic graph, DAG)，它即代表著貝氏網路的結構，在圖形中每一個節點即表示著一個變數，每一條有方向的邊線即表示著兩個變數間的關係。數值部分即使用條件機率來表示兩個變數間關係的強度[4]。

貝氏網路可以以 (V, E, P) 來表示：

- V 為變數或頂點的集合，如： $V = \{a, b, c, d\}$ 。
- E 為邊的集合，如： $E = \{a \rightarrow b, a \rightarrow d, b \rightarrow c, d \rightarrow c\}$ ，若和 V 一起即為一個方向性非循環圖 $G = (V, E)$ 。
- $P = P(v | \pi_v)$ ： $v \in V$ ，其中 π_v 為 v 父(母)節點的集合，如： $P = P(c | b, d)$ 。

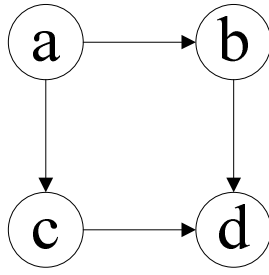


圖 2-1 簡單方向性非循環圖

將以上三項合起來即成爲一個貝氏網路。

貝氏網路藉由收集條件獨立來建立全體變數的聯合機率，例如圖 2-2 ：

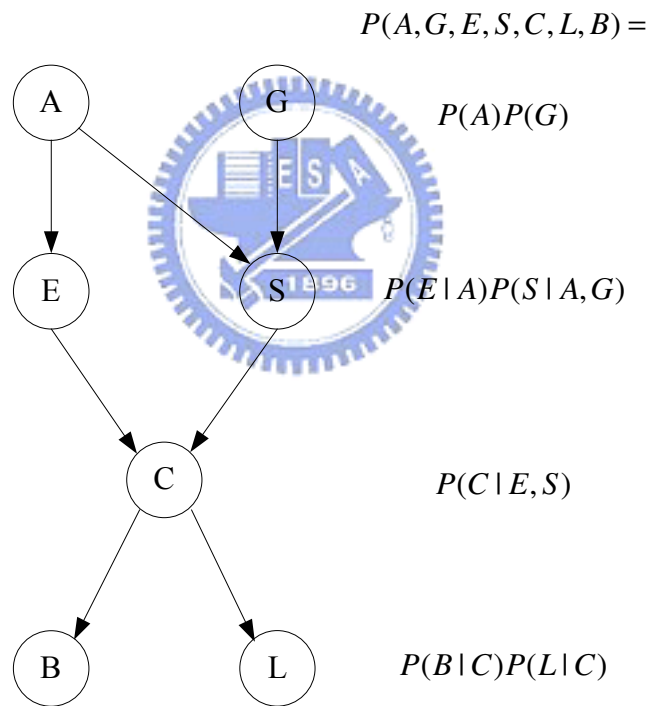


圖 2-2 全體變數聯合機率

因此貝氏網路的全體聯合機率表示式爲：

$$P(V_1, V_2, \dots, V_n) = \prod_{i=1}^n P(V_i | pa(V_i)) \quad (2.19)$$

其中 $pa(V_i)$ 為第 V_i 個節點的父（母）節點。當我們擁有聯合機率值時，我們就可以解決許多的問題了。

貝氏網路在許多領域都得到應用，如醫學、化學、地質石油探勘、氣象、晶圓製程和其他領域等等，在個個領域都承擔著不同的任務。如果不考慮領域的特性而按照貝氏網路本身求解的特點，最主要分為：診斷、預測、分類和決策四種功用。而本論文即使用其中兩個功能：診斷與預測。

比如說：當我們知道某個病人有病狀 E 的情形之下，我們想要診斷出他感染病症的原因 Q 的機率為多少。

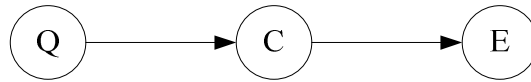


圖 2-3 簡單診斷例子

我們應用貝氏原理找出 $P(Q|E)$ ：

$$P(Q|E) = \frac{P(E|Q)P(Q)}{P(E)} \quad (2.20)$$

其中 $P(E|Q)$ 為：

$$P(E|Q) = \sum_C P(E|C)P(C|Q) \quad (2.21)$$

因此將 (2.21) 式帶入 (2.20) 式中，則將 (2.20) 式改成：

$$P(Q|E) = \frac{\sum_C P(E|C)P(C|Q)P(Q)}{P(E)} = \frac{\sum_C P(C,E,Q)}{P(E)} \quad (2.22)$$

我們在 (2.22) 式中，可以看到圖 2-3 的全體變數聯合機率，藉由幾個條件機率相乘所得的聯合機率來執行診斷的問題。

由於貝氏網路全體變數的聯合機率為 (2.19) 式，若當變數總和 n 一多的話，聯合機率的表格會變龐大且計算時間會變多。為了改善這些缺點，我們需要根據變數間的關係將變數建立分群(cluster)，當我們一個小分群的聯合機率即可解決的

問題，不需要再用全體變數的聯合機率來回答，若需要用到全體變數的聯合機率的話，也可由所有分群的聯合機率來組成，因此我們可以借用聯合樹演算法來快速計算聯合機率，下面一章我們將對聯合樹演算法作詳細的介紹。



第三章 聯合樹演算法

在得到一些其他變數或觀察到的證據(evidence)的情況之下，要計算所有變數的排斥的(marginal)機率是可行的。在經過多人的努力之下研究出很多演算法，其中有一個演算法的設計是直接操作在貝氏網路的結構之下。每個變數任何時間都在接受新的證據，然後再傳送證據給鄰近的變數，再之後由這鄰近的變數再將證據傳送出去，一個變數一個變數地傳送，過了一段時間之後希望貝氏網路能夠趨近穩定。現在這最普通最受大家歡迎的演算法就是聯合樹演算法[1,2,3,5,6,7]。

聯合樹演算法有兩個主要的步驟：轉換(transformation)和傳播(propagation)這兩個步驟。轉換步驟是從貝氏網路建立一個非方向性的聯合樹。第二個步驟，傳播是在聯合樹上傳播接受到的證據。圖3-1 聯合樹演算法流程，下面我們將一一介紹聯合樹演算法。



圖3-1 聯合樹演算法流程

3.1 轉換

轉換是由貝氏網路轉換成聯合樹的一個步驟，是靠著一連串圖形化轉換的過程。不要被轉換這個字眼所搞混，它意指一個”新的”圖形被創造出來，所以我們仍然還有一個未經改變的貝氏網路。從貝氏網路轉換成聯合樹有四個主要步驟：

演算法 1：轉換

1. 將貝氏網路中的方向性非循環圖形轉換成一個moral graph，此圖形沒有方向。
2. 在moral graph中增加一些邊線形成一個三角化的圖形。
3. 在三角化的圖形中建立分群圖形(cluster graph)。
4. 靠著連接分群創造出聯合樹。

3.1.1 Moral Graph

Moral Graph是將貝氏網路圖形中有相同子節點的父（母）節點，用沒方向性的線兩兩相連起來，之後再將有方向性的線改成沒方向性的線。我們通常都會說這些父(母)節點互相結婚。其中任一個父(母)節點都可以跟其他任何的節點結婚。於圖3-2中，左邊為原來的貝氏網路圖形，右邊為經過moralize之後的moral graph。因為a節點和d節點都是b節點的父（母）節點，所以增加a-d的邊線。增加三個邊線b-c、b-e和c-e是因為b節點、c節點和e節點都是f節點的父（母）節點。在圖3-2中增加的邊線都是黑點的線。

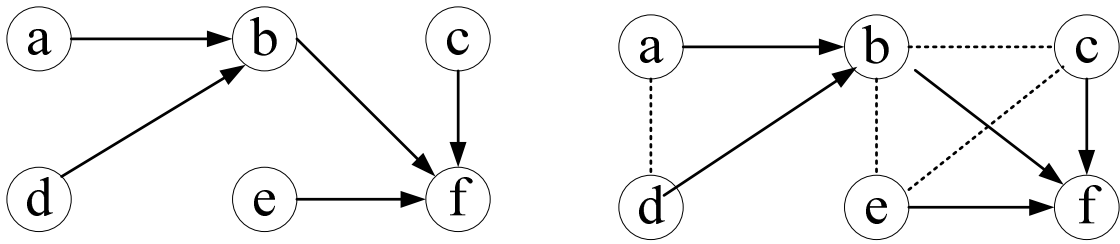


圖3-2 左：貝氏網路圖形；右：moralized graph，黑點線為增加的邊線

要從貝氏網路的圖形來建立moral grapg的演算法為：

演算法 2：moralizing貝氏網路圖形

1. 建立一個 G_M 的圖形，將貝氏網路的圖形複製到 G_M 圖形。
2. 若 G_M 圖形中有相同子節點的父（母）節點，在 G_M 圖形中加入沒方向性的將兩兩父（母）節點相連起來。
3. 將 G_M 圖形中所有有方向性的線改成沒方向性的線。

在此演算法的第一步驟中，我們建立一個貝氏網路的複製圖形，所以從這個步驟之後的操作都是操作在新的圖形上，因此我們就不用擔心破壞了原本的貝氏網路圖形。

3.1.2 三角化圖形

若在非方向性圖形中的有任何一循環(cycle)的邊線數量大於三的話，即為非三角化圖形，如圖3-3中 $\{A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow A\}$ 即為一個循環，且邊線數為五。當圖形為非三角化圖形時，下面的演算法 3可以在圖中產生填入的線(fill-in)使圖形三角化。我們擁有一個非方向性圖形 $G = (V, E)$ ，再經由下面演算法產生填入的邊線(fill-in) E' ，使圖形 $G' = (V, E \cup E')$ 成爲一個三角化圖形，下面我們將作一些演算法的說明：

演算法 3：最大基數搜尋(MCS)

1. 一開始沒有填入的線，所以 $E' = \emptyset$ 。
2. 令 $i = 1$ 並從節點集合 V 中選一個初始節點 $\alpha(1) = V_i$ ，使得 $\alpha(i) = V_i$ 。
3. 在尚未編號的節點中，從中找出跟已編過號碼的節點相連的邊線數比較多的節點編號為 $i + 1$ ，所以 $\alpha(i + 1) = V_k$ 。
4. 如果 $Nbr(V_k) \cap \{\alpha(1), \dots, \alpha(i)\}$ 中的節點沒有完整(complete)，就增加一條邊線到 E' 中使得 $Nbr(V_k) \cap \{\alpha(1), \dots, \alpha(i)\}$ 變成完整，然後再跳回步驟2.；否則就到步驟5.。其中 $Nbr(V_k)$ 即是與 V_k 相鄰節點的集合，完整即是說任兩個節點都要連接在一起。
5. 假如 $i = n$ 時，就停止；否則令 $i = i + 1$ ，然後回到步驟3.。其中 n 為節點總數。



圖3-3是一個非三角化的圖形。我們使用演算法 3來填入邊線使圖3-3變成三角化的圖形。在這個演算法中，我們選擇 C 節點來當初使節點。

- 步驟1.： $E' = \emptyset$ 。
- 步驟2.： 令 $i = 1$ 和 $\alpha(1) = C$ 。

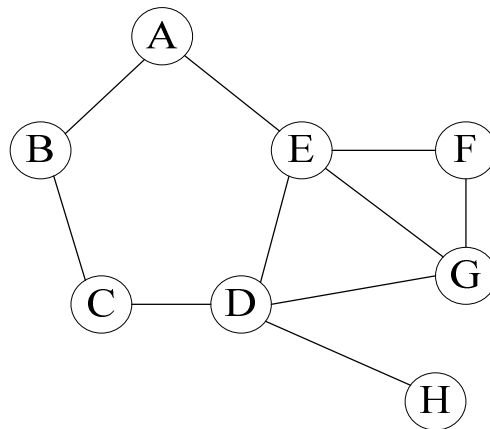


圖3-3 未三角化的非方向性圖形

- 步驟3.：節點 B 和節點 D 和已編過號碼的節點相連的邊線數都為一，其他為編號的節點與已編過號碼的節點相連的邊線數都為零。由於節點 B 和節點 D 都一樣，所以我們從中任選一個，我們選擇節點 D 並編為2號，即 $\alpha(2) = D$ 。
- 步驟4.：在這個情況下 $Nbr(D) \cap \{\alpha(1)\}$ 只有 C 節點，所以沒有完不完整的問題，因此不需要增加邊線。
- 步驟5.： $i = 1 \neq n$ ，我們將 i 加一，然後回到步驟3。
- 步驟3.~5.：同上面的步驟， B 節點和 E 節點編為3號和4號， $\alpha(3) = B, \alpha(4) = E$ 。
- 步驟3.~4.：和已編過號碼的節點相連的邊線數最多的節點是 A 節點，所以， $\alpha(5) = A$ 。如圖3-4(a)所示， $Nbr(A) \cap \{\alpha(1), \dots, \alpha(4)\} = \{B, E\}$ ，且 $\{B, E\}$ 的集合中節點並沒有兩兩相連，所以並不完整。因此，我們必須增加 $B-E$ 這條線到 E' (如圖3-4(b)所示)，並從步驟2.重新開始。且要注意現在 $E' = \{B-E\}$ 。
- 步驟2.~5.：節點 C 、 D 和 B 的編號為1、2和3。
- 步驟3.~4.：和已編過號碼的節點相連的邊線數最多的節點是 E 節點， $Nbr(E) \cap \{\alpha(1), \dots, \alpha(3)\} = \{B, D\}$ ，如圖3-4(c)所示， $\{B, D\}$ 這個集合並不完整。因此，我們必須將 $B-D$ 這條線加入 E' ，然後再從步驟2.重新開始。現在 $E' = \{B-E, B-D\}$ (如圖3-4(d))。
- 步驟2.~5.：將節點 C 、 D 、 B 、 E 、 A 、 G 、 F 和 H ，從1號到8號來編號。所得到的結果圖形為 $G' = (V, E \cup E')$ 是一個三角化圖形，其編號就如圖3-5所示。

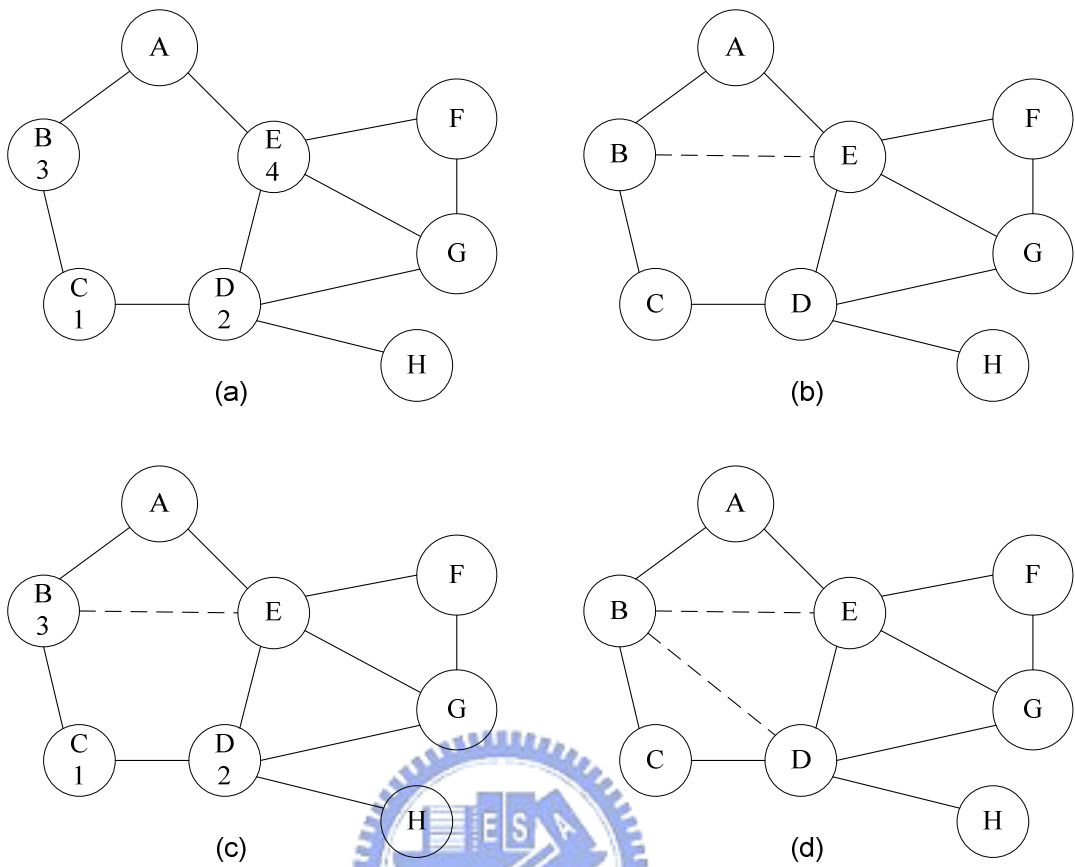


圖3-4 運用最大基數搜尋演算法將邊線填入

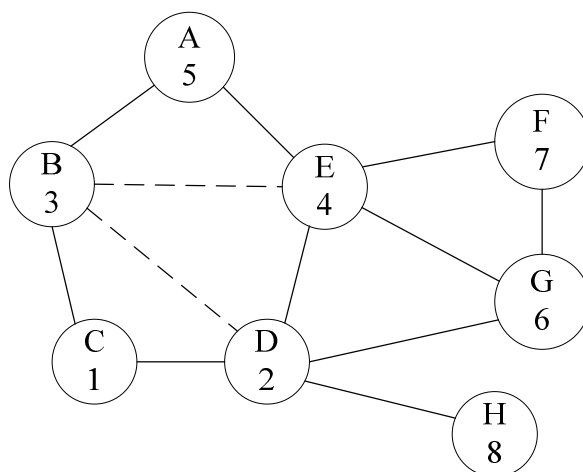


圖3-5 運用最大基數搜尋演算法將節點編號

選取不同的起始，相同的圖所得到的三角化圖形也會有不同的情形。例如在同一張圖上運用最大基數搜尋，所選取的起始點不同，就可能產生不同的編號，如圖3-6所示。

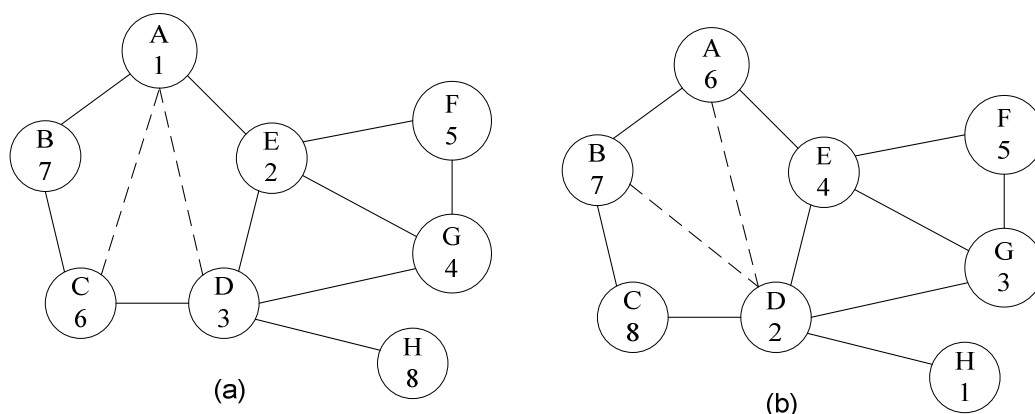


圖3-6 另外兩種運用最大基數搜尋演算法將節點編號



3.1.3 分群圖形

分群圖形是在三角化圖形中由一些擁有相同特性的節點聚集起來的圖形，而這個過程就當作是群集(clustering)，我們可以用簡單的結構來得到一個新的圖形，此新的圖形事實上仍然還保留的原本圖型的特性。首先我們要先將相同特性的點聚集來：

演算法 4：建立分群

1. 運用演算法 3 將三角化圖形上的節點編號，如： (V_1, \dots, V_n) 。
2. 建立 Π_{v_i} ，為每個節點的父（母）節點集合，節點編號數值較小有可能是數值大的父（母）節點，反之則不然。

3. 計算 π_i ，為計算 Π_{v_i} 集合裡面有幾個節點。
4. 建立路徑點 λ_i
 - 如果 $i = n$ 的話， λ_i 就等於路徑點。
 - 如果 $i < n$ 而且 $\pi_{i+1} < 1 + \pi_i$ 的話， λ_i 就等於路徑點。
 - 若不是以上兩種情況，即沒有 λ_i 。
5. 分群即是

$$C_j = \lambda_j \cup \Pi_{\lambda_j}$$

圖 3-7 是一個三角化的圖形，我們運用簡單的步驟來群聚節點以達到分群的效果：

- 步驟 1.：運用演算法 3 將三角化圖形上的節點編號，如圖 3-8 所示，將 A、B、C、E、D、F、G、H 和 I，由 1 號編到 9 號。
- 步驟 2.：找出每個節點的父（母）節點集合，節點 A 的父（母）節點為 $\Pi_{v_1=A} = \{\emptyset\}$ ，節點 D 的父（母）節點為 $\Pi_{v_5=D} = \{B, E\}$ ，如圖 3-8 所示。
- 步驟 3.：計算 π_i ， $\pi_1 = 0$ ， $\pi_2 = 1$ ， \dots ， $\pi_9 = 1$ 。

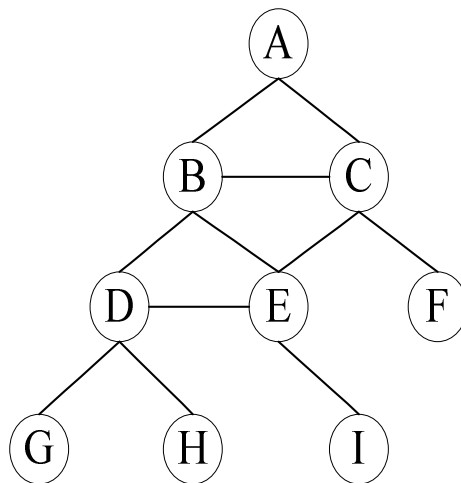


圖 3-7 三角化的圖形

- 步驟 4. : $i=1 < n$ 時, $\pi_{i+1} = 1$, $1 + \pi_i = 1$, 所以不屬於兩種情況裡的任一種, 所以沒有 λ_i 。當 $i=4$ 時, $\pi_{4+1} = 2$, $1 + \pi_4 = 3$, 屬於第二種情況, 所以 $\lambda_4 = E$ 。
- 步驟 5. : 重新編排所有的路徑點,

$$\lambda^{old} = \{\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = C, \lambda_4 = E, \lambda_5 = D, \lambda_6 = F, \lambda_7 = G, \lambda_8 = H, \lambda_9 = I\}$$

$$\lambda^{new} = \{\lambda_1 = C, \lambda_2 = E, \lambda_3 = D, \lambda_4 = F, \lambda_5 = G, \lambda_6 = H, \lambda_7 = I\}$$

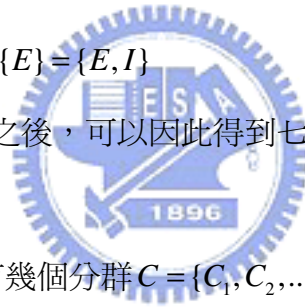
$$C_1 = \lambda_1 \cup \Pi_{\lambda_1} = C \cup \{A, B\} = \{A, B, C\}$$

$$C_2 = \lambda_2 \cup \Pi_{\lambda_2} = E \cup \{B, C\} = \{B, C, E\}$$

-
-
-

$$C_7 = \lambda_7 \cup \Pi_{\lambda_7} = I \cup \{E\} = \{E, I\}$$

經過以上這五個步驟之後, 可以因此得到七個分群 $\{C_1, C_2, \dots, C_7\}$, 如圖3-8所示。



當我們群集之後, 會有幾個分群 $C = \{C_1, C_2, \dots, C_n\}$, 我們可以用一個簡單的構造將這些分群建構出一個新的圖形, 就叫分群圖形(Cluster graph)。一個分群圖形 $G = (C, E)$, 其中 $C = \{C_1, C_2, \dots, C_n\}$, 且將每一個分群都當作節點, 當分群與分群之間有著共同的節點的時候就會有邊線E互相連接這些分群, 如圖3-9所示。

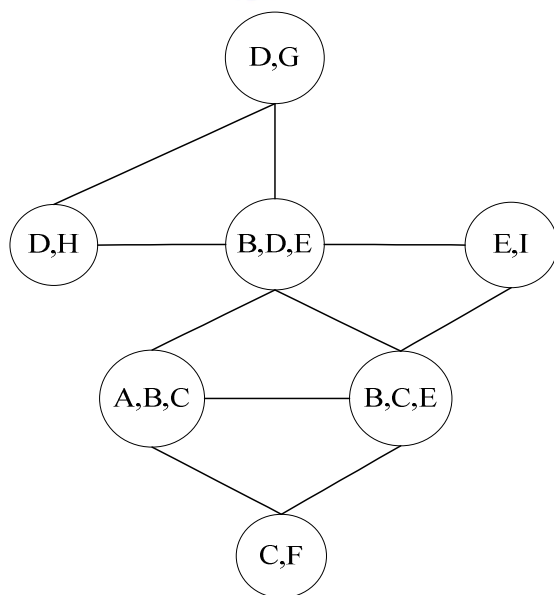
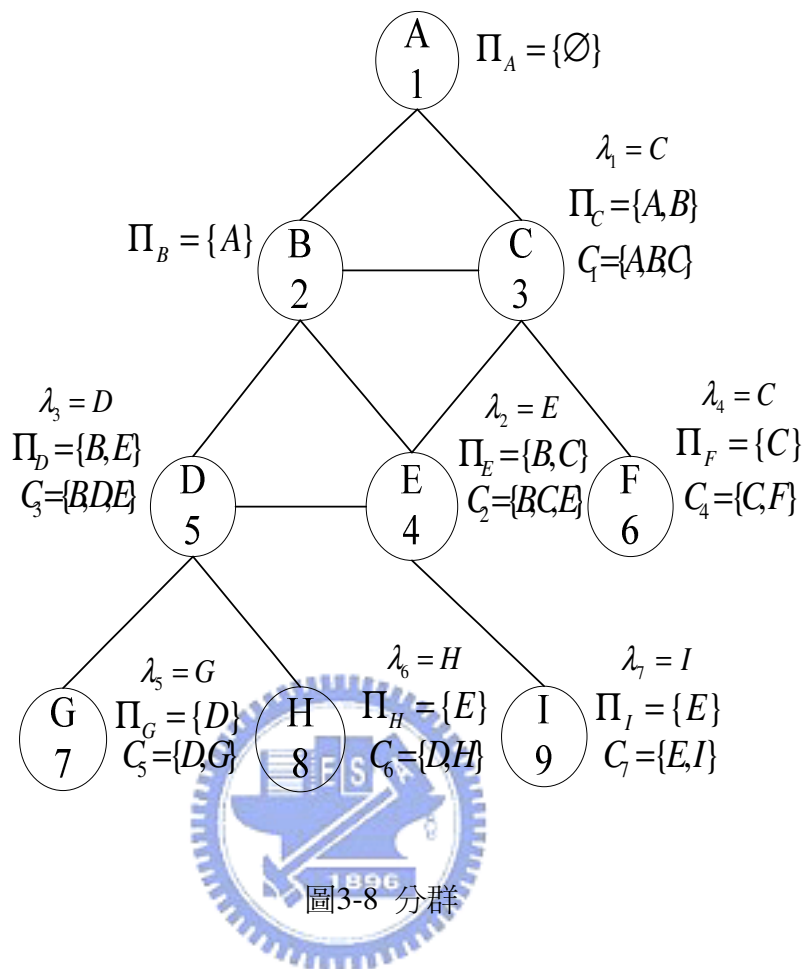


圖3-9 分群圖形

3.1.4 聯合樹

我們從 3.1.3 小節得到了一個分群集合 $C = \{C_1, C_2, \dots, C_n\}$ ，我們需要以符合聯合樹的結構來將這些分群連接起來。如果我們因此而建立好聯合樹的話，那麼我們就可以盡快的計算出機率值。下面的演算法是藉由組織群集三角化圖形 $G = (V, E)$ 而來的分群形成一個樹狀結構圖 $G' = (C, E')$ ：

演算法 5：建立聯合樹

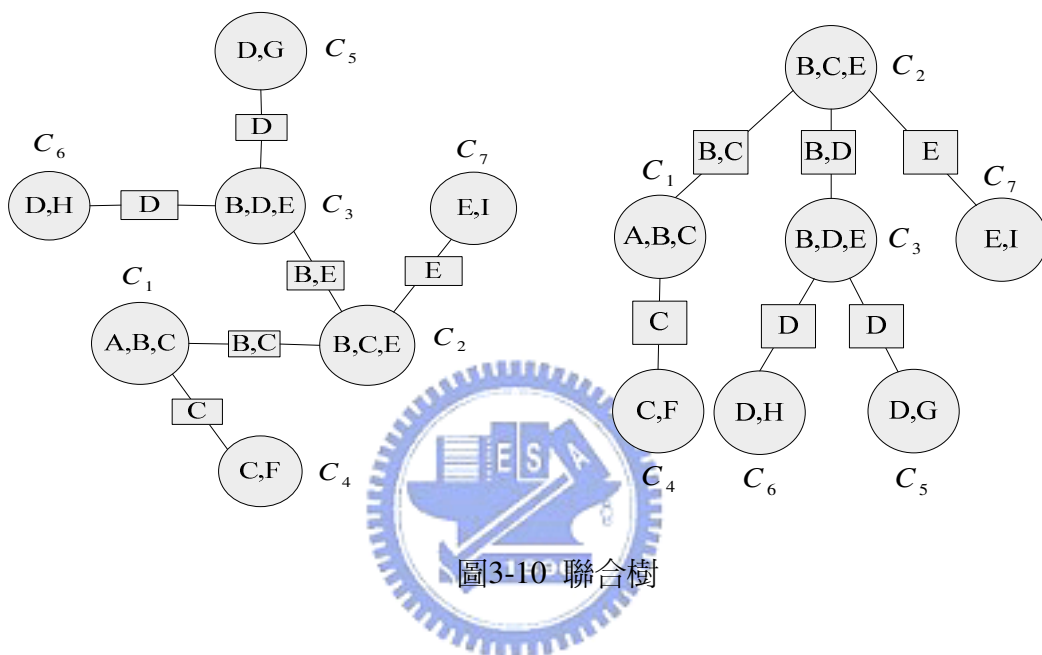
1. 使用演算法 4 群集三角化圖形 $G = (V, E)$ ，以得到分群集合 $C = \{C_1, C_2, \dots, C_n\}$ 。
2. 當任何一個分群 $C_i \in C$ 且 $i = 2, \dots, n$ ，與從 $\{C_1, \dots, C_{i-1}\}$ 之中選出一個在兩個分群之間有著最多共同節點的數量的分群 C_k ，並增加 $C_i - C_k$ 的邊線到 E' 裡面，一開始的時候 $E' = \{\emptyset\}$ 。
3. 將 C_i 和 C_k 的交集的節點 $\{C_i \cap C_k\}$ 記錄起來。

我們運用上面演算法 5 使得三角化圖形（圖3-7）轉換成聯合樹。我們在之前有運用過演算法 4 得到分群的集合 C ， $C_1 = \{A, B, C\}$ ， $C_2 = \{B, C, E\}$ ， $C_3 = \{B, D, E\}$ ， $C_4 = \{C, F\}$ ， $C_5 = \{D, G\}$ ， $C_6 = \{D, H\}$ ， $C_7 = \{E, I\}$ 。我們可以經由增加一些必須要的邊線到 E' 裡面，而產生聯合樹：

- 當分群 C_4 在 $\{C_1, C_2, C_3\}$ 中，有兩個分群 C_1 和 C_2 都與 C_4 之間有著最多共同節點的數量，我們選擇 C_1 並將 $C_1 - C_4$ 加入 E' ，且 $\{C_1 \cap C_4\} = \{C\}$ 。
- 當分群 C_5 在 $\{C_1, C_2, C_3, C_4\}$ 中，分群 C_3 與 C_5 之間有著最多共同節點的數量，因此我們將 $C_3 - C_5$ 加入 E' ，且 $\{C_3 \cap C_5\} = \{D\}$ 。
- 當分群 C_6 在 $\{C_1, C_2, C_3, C_4, C_5\}$ 中，有兩個分群 C_3 和 C_5 都與 C_6 之間有著最多共同節點的數量，我們選擇 C_3 並將 $C_3 - C_6$ 加入 E' ，且 $\{C_3 \cap C_6\} = \{D\}$ 。

- 照著上面相同的作法，我們將增加 $C_1 - C_2$ ， $C_2 - C_3$ 和 $C_2 - C_7$ 到 E' 裡。
- 最後 $E' = \{C_1 - C_2, C_1 - C_4, C_2 - C_3, C_2 - C_7, C_3 - C_5, C_3 - C_6\}$ 。

我們因此而得到樹狀結構 $G' = (C, E')$ 的聯合樹，如圖3-10一樣。在下面一節中，我們可以看到一些證據在樹狀結構聯合樹上運用傳播的方法更容易計算機率值跟更新機率值。



3.2 傳播

我們已經由前一節得到樹狀結構聯合樹，但我們沒有因此而破壞了原本的貝氏網路圖型的結構。現在我們要作的就是將前一節轉換過所得的樹狀結構用數量來表示，就是將機率直加入樹狀結構聯合樹裡面，這也就是我們的這一階段的目標。傳播又有兩個部分：為初始化部分和更新部分。下面我們將作介紹：

3.2.1 位能初始化

本小節為初始化的部分，我們必須先由貝氏網路中的條件機率在聯合樹上建立初始化位能(potential)，此位能有點類似機率，但並不是十分的正確，所以我們還必須經過傳播這些位能而得到準確的機率值：

演算法 6：初始化

1. 在聯合樹上建立初始化位能。
2. 傳播位能。

3.2.1.1 建立初始化位能

在貝氏網路圖形中，條件機率的形式為 $P(V_u | V_{pa(u)})$ ，若要計算所有變數節點的聯合機率，就如下式所示：

$$P(V) = \prod_{u \in n} P(V_u | V_{pa(u)}) \quad (3.1)$$

其中 V 節點的集合 $V = \{V_1, V_2, \dots, V_n\}$ ， u 為 V 裡面的其中一個節點。若我們要計算圖 3-11 所有變數的聯合機率的話，可得下面的式子：

$$P(V) = P(A)P(B|A)P(C|A)P(D|B)P(E|C)P(F|D,E)P(G|C)P(H|E,G)$$

其中 $V = \{A, B, C, D, E, F, G, H\}$ 。那我們就可以介紹下面的演算法來初始化位能：



演算法 7：初始化聯合樹

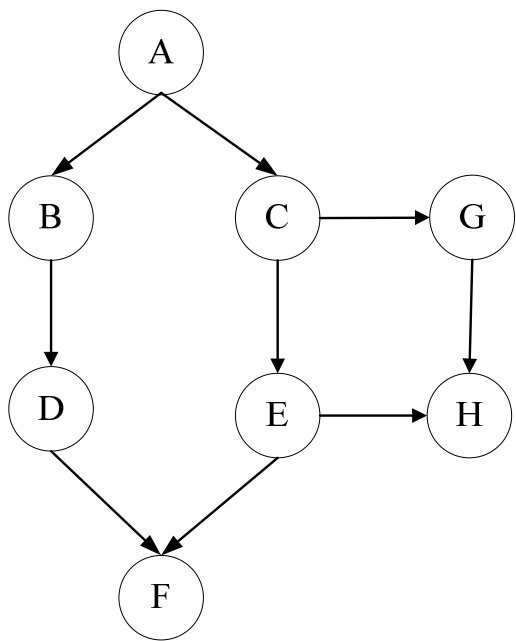
1. 將每一個分群 C 和每一個交集 S 的位能設為 1：

$$\phi_C \leftarrow 1, \phi_S \leftarrow 1$$

2. 每一個在貝氏網路的節點 u ，假如分群 C 包含節點 u 且節點 u 的父（母）點也必須包含在分群 C 裡面 $pa(u) \subseteq C$ ，然後將條件機率 $P(V_u | V_{pa(u)})$ （或沒有父（母）點的 $P(V_u)$ ）算入分群位能 ϕ_C 中：

$$\phi_C \leftarrow \phi_C P(V_u | V_{pa(u)})$$

我們運用前一小節 3.1 節轉換，將圖 3-11 轉換成樹狀結構聯合樹的圖形，如圖 3-12 所示。



A	
1	0.5
0	0.5

P(B A)	B=1	B=0
A=1	0.5	0.5
A=0	0.4	0.6

P(C A)	C=1	C=0
A=1	0.7	0.3
A=0	0.2	0.8

P(D B)	D=1	D=0
B=1	0.9	0.1
B=0	0.5	0.5

P(G C)	G=1	G=0
C=1	0.8	0.2
C=0	0.1	0.9

P(E C)	E=1	E=0
C=1	0.3	0.7
C=0	0.6	0.4

P(F D,E)		F=1	F=0
D=1	E=1	0.01	0.99
D=1	E=0	0.01	0.99
D=0	E=1	0.01	0.99
D=0	E=0	0.99	0.01

P(H E,G)		H=1	H=0
E=1	G=1	0.05	0.95
E=1	G=0	0.95	0.05
E=0	G=1	0.95	0.05
E=0	G=0	0.95	0.05

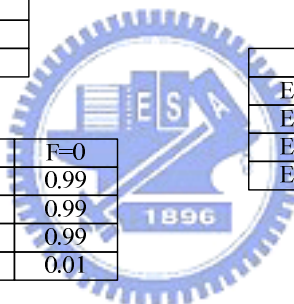


圖3-11 貝氏網路

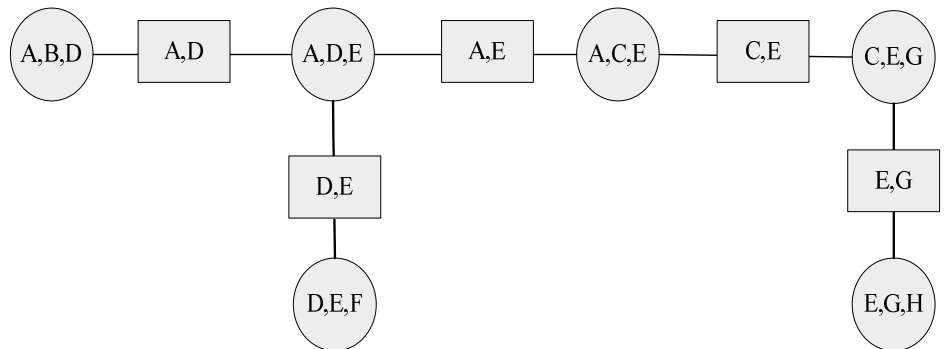


圖3-12 將圖3-11轉換成樹狀結構聯合樹的圖形

圖3-13解釋著從圖3-12中，將初始化分群 $\{A, C, E\}$ 跟交集 $\{C, E\}$ 。在這個例子中節點C跟節點E的父（母）點都包含在分群 $\{A, C, E\}$ 中，而節點A沒有父母點，所以經過初始化後 $\phi_{ACE} = 1 \times P(C|A) \times P(E|C)$ 和 $\phi_{CE} = 1$ 。

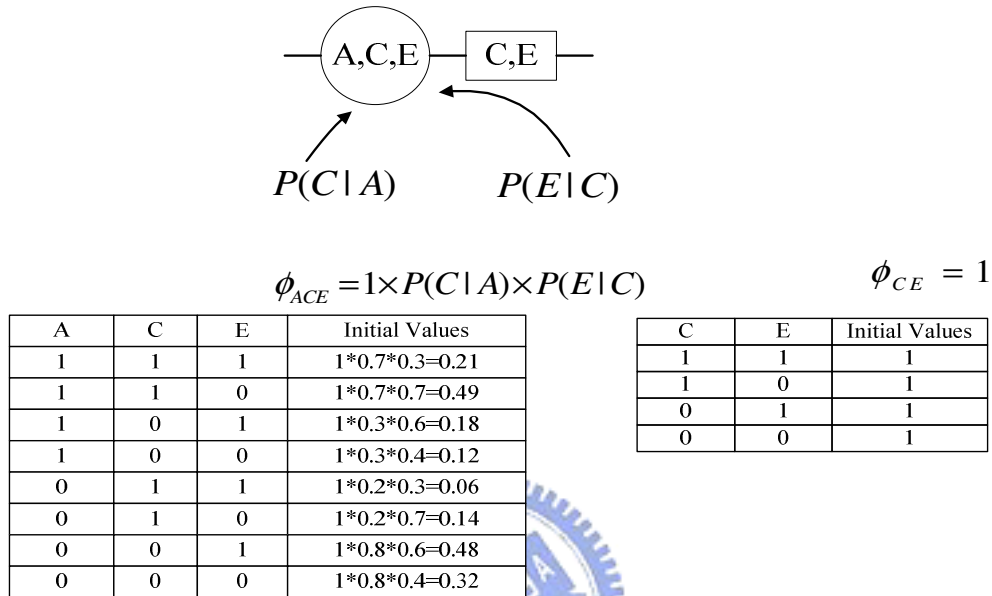


圖3-13 初始化分群 $\{A, C, E\}$ 跟交集 $\{C, E\}$

3.2.2 傳播位能

在傳播位能之前，必須先將排斥(marginalization)的操作方法解釋一下。比如說我們有兩個分群 $A = \{a, d\}$ 和 $B = \{a, b, d\}$ 而且 $A \subseteq B$ ，將運用排斥計算分群A的位能為，如表3-1和(3.2)式所示：

$$\phi_A = \sum_{\{a,b,d\} \setminus \{a,d\}} \phi_B = \sum_{\{b\}} \phi_B \quad (3.2)$$

在聯合樹經過初始化後，並不代表所有的機率值都是正確的。舉個例子來說：若現在有兩個分群A跟B，兩個分群接連在一起且有共同的交集 S_{AB} ，兩分群都有著變數 a 。我們經由上面剛講過的排斥位能可以算出 a 位能，可能會有不相同的結

a	b	d	ϕ_{abd}
1	1	1	0.12
1	1	0	0.05
1	0	1	0.27
1	0	0	0.3
0	1	1	0.16
0	1	0	0.01
0	0	1	0.03
0	0	0	0.06

a	d	ϕ_{ad}
1	1	0.39
1	0	0.35
0	1	0.19
0	0	0.07

表3-1 分群{a,d}的排斥位能

果。可能的原因是在聯合樹上個別的分群還沒受到其他分群的訊息所影響。全體的傳播指的就是在聯合樹上一連串的局部操作，此局部操作指的就是傳遞訊息 (message passing)，訊息傳遞可以從一個分群將更新的訊息傳遞到跟他相連的交集上，再從交集傳遞更新的訊息到另一個相連的分群上。比較好的傳播就是所有的分群傳遞一個訊息到所有分群的鄰近分群上。



3.2.2.1 傳遞訊息

單一個在兩個分群A和B之間的訊息傳遞由兩個步驟組成，第一個步驟叫做投影(Projection)：訊息由分群A傳遞到交集 S_{AB} 上；第二個步驟叫做吸收(Absorption)：訊息由交集 S_{AB} 傳遞到分群B上，如表3-2所示。

演算法 8：訊息傳遞(A,B)

1. 投影：更新交集 S_{AB} 上的位能：

$$\phi_{S_{AB}}^{Old} \leftarrow \phi_{S_{AB}}$$

$$\phi_{S_{AB}}^{New} \leftarrow \sum_{A \cap S_{AB}} \phi_A$$

2. 吸收：吸收訊息更新分群B的位能 ϕ_B ：

$$\phi_B \leftarrow \phi_B^{Old} \frac{\phi_{S_{AB}}^{New}}{\phi_{S_{AB}}^{Old}}$$

如果 $\phi_{S_{AB}}^{Old} = 0$ 則 $\phi_B \leftarrow 0$ 。

一個簡單的訊息傳遞，由分群 $A = \{a, b, d\}$ 傳遞訊息到分群 $B = \{a, d, e\}$ ，所有的變數都是二元的。交集 S_{AB} 的位能 ϕ_S 是藉由分群A投影更新的，然後分群B的位能 ϕ_B 就從交集 S_{AB} 吸收新的資訊更新，如表3-2所示。

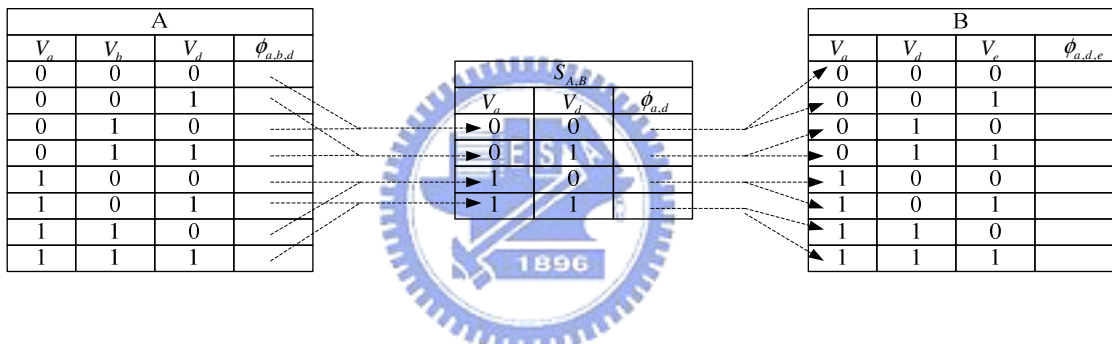


表3-2 一個簡單的訊息傳遞

3.2.2.2 全體傳播

在 n 個分群的聯合樹中，全體傳播由隨便選擇一個分群X開始，有 $2(n-1)$ 個訊息傳遞，又分為兩項：一為收集證據(collect-evidence)；一為散佈證據(distribute-evidence)。在收集證據期間，從距離分群X最遠的分群開始，每一個分群將傳遞訊息給它鄰近的分群，直到傳遞到分群X為止。在散佈證據期間，由本身分群X開始，將訊息傳遞給分群X鄰近的分群，直到傳遞到最遠的分群為止。收集證據這一項有 $n-1$ 個訊息要傳遞；而散佈證據也有另外 $n-1$ 個訊息要傳遞。

演算法 9：全體傳播

1. 隨便選擇一個分群標號為X。
2. 取消其他所有分群的標號，收集證據(X)。
3. 取消其他所有分群的標號，散佈證據(X)。

在傳遞訊息的機制裡面，一個分群想要傳遞訊息給它鄰近的分群，必須等到此分群接收他其他所有分群的訊息之後，才能夠將訊息傳遞出去。圖3-14說明著在聯合樹裡的機率傳播步驟，我們隨便選擇一個分群當起始點，選擇分群{A,C,E}當起始分群。在收集證據階段訊息的傳遞方向是往分群{A,C,E}傳遞，從{A,B,D}、{D,E,F}跟{E,G,H}這三個最遠的分群開始往分群{A,C,E}傳遞訊息，由實線箭頭來表示收集證據的傳遞步驟；在散佈證據的階段，訊息由分群{A,C,E}開始，以傳離分群為方向傳遞出去，由破折號箭頭來表示散佈證據的步驟，數字表示著訊息傳遞的順序。

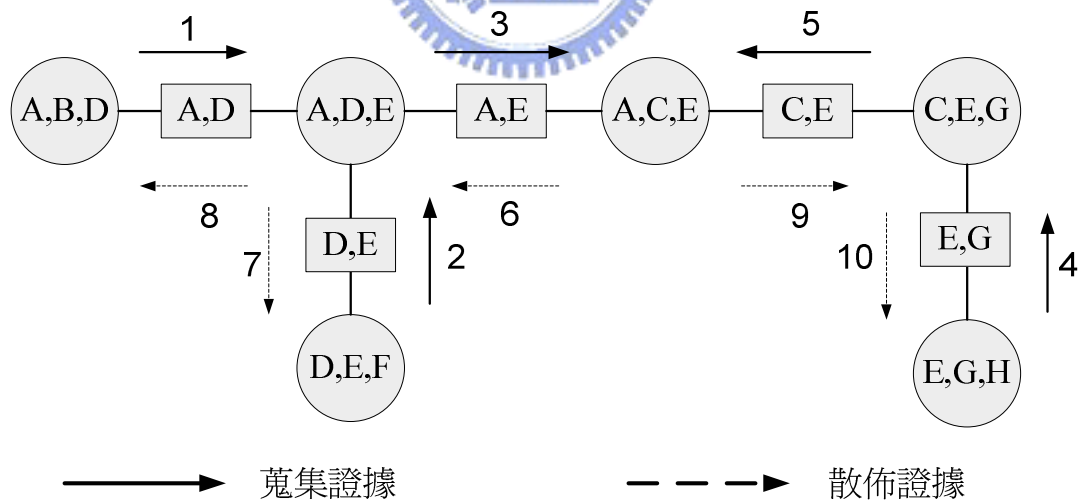


圖3-14 全體傳播

一般貝氏網路每一次處理問題往往都需要計算全體變數的聯合機率，但聯合樹就可以不需要計算全體變數的聯合機率，把跟問題相關的變數運用下面的式子計算出聯合機率，來處理問題：

$$P(V_x) = \frac{\prod_{cluster C} \phi(V_C)}{\prod_{separator S} \phi(V_S)} \quad (3.3)$$

其中C為分群，S為分群與分群的交集集合， V_x 為幾個分群的聯集集合。如果我們的問題可能跟圖3-11中的A和H有關，那麼我們就需要用上面的式子計算初跟A和G有關的聯和機率：

$$P(A, C, E, G, H) = \frac{\phi_{A,C,E} \phi_{C,E,G} \phi_{E,G,H}}{\phi_{C,E} \phi_{E,G}} \quad (3.4)$$

3.2.2.3 排斥

經過全體傳播之後，我們可以得到擁有準確機率值的聯合樹，且我們可以對我們感興趣的變數 V 計算機率值 $P(V)$ ，如下：

1. 選擇一個包含變數 V 的分群或交集 X 。
2. 使用排斥來計算機率值 $P(V)$ ：



$$P(V) = \sum_{X \in \mathcal{X}(V)} \phi_X \quad (3.5)$$

表3-3中的分群位能 ϕ_{ABD} 是由圖3-11的聯合樹而來的，將分群位能 ϕ_{ABD} 經過排斥可以計算出 $P(A)$ ，也可以經過排斥再計算出 $P(D)$ 。

3.2.3 更新機率

現在我們已經能夠計算任一變數 V 的機率值 $P(V)$ 了，但前提是在沒有新證據加入的情形之下。在這一節中，我們必須在新證據 $V_u = v_u$ 加入之下，略微的修改原本的聯合樹演算法，才能計算出更新後的機率值，圖3-15為有觀察資料的聯合樹演算法流程。

3.2.3.1 將證據編碼成一個近似函數

我們將變數 V_u 用數學方法將它編碼成一個近似函數(likelihood function)，用 $\lambda_{v_u}(v_u)$ 來表示。假如我們觀察得到一個變數，則近似函數為：

$$\lambda_{v_u}(v_u) = \begin{cases} 1, & \text{當 } V_u \text{ 的狀態為 } v_u \text{ 的時候} \\ 0, & \text{其他狀態} \end{cases}$$

當 V_u 為 v_u 時，近似函數為 $\lambda_{v_u}(v_u) \leftarrow 1$ ，如 V_u 有 (0,1,2,3) 四種狀態，當我們觀察到 $v_u = 2$ 時，近似函數為 $\lambda_{v_u}(v_u) = (0,0,1,0)$ 。

$\phi_{ABD} =$	A	B	C	ϕ_{ABD}
	1	1	1	0.225
	1	1	0	0.025
	1	0	1	0.125
	1	0	0	0.125
	0	1	1	0.180
	0	1	0	0.020
	0	0	1	0.150
	0	0	0	0.150

$P(A) = \sum_{B,D} \phi_{ABD} =$	A	P(A)
	1	0.225+0.025+0.125+0.125=0.500
	0	0.180+0.020+0.150+0.150=0.500

$P(D) = \sum_{A,B} \phi_{ABD} =$	D	P(D)
	1	0.225+0.125+0.180+0.150=0.680
	0	0.025+0.125+0.020+0.150=0.320

表3-3 排斥的例子

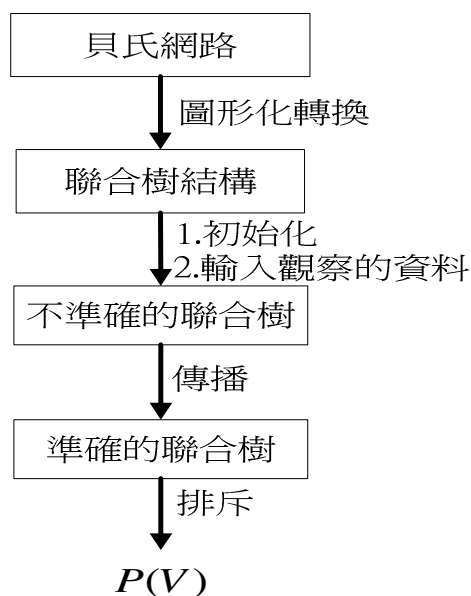


圖3-15 有觀察資料的聯合樹演算法流程

3.2.3.2 擁有觀察資料之位能初始化

我們將對演算法7作一些改變以包括建立的近似函數。

演算法 10：初始化聯合樹（經過修改的）

1. 將每一個分群C和每一個交集S的位能設為1：

$$\phi_C \leftarrow 1, \phi_S \leftarrow 1$$

2. 對任何節點 u ：

(a) 每一個在貝氏網路的節點 u ，假如分群C包含節點 u 且節點 u 的父(母)

點也必須包含在分群C裡面 $pa(u) \subseteq C$ ，然後將條件機率 $P(V_u | V_{pa(u)})$

(或沒有父(母)點的 $P(V_u)$) 算入分群位能 ϕ_C 中：

$$\phi_C \leftarrow \phi_C P(V_u | V_{pa(u)})$$

(b) 設每一個狀態的近似函數為1：


$$\lambda_{v_u}(v_u) \leftarrow 1$$

3.2.2.3 輸入觀察資料於網路中

當接收到觀察到的資料時，我們可以改變變數的近似函數。假如我們觀察的變數為 V_a 和 V_b ，進而更新近似函數 $\lambda_{v_a}(v_a)$ 和 $\lambda_{v_b}(v_b)$ ，之後再使用全體傳播的演算法更新所有的位能。

演算法 11：輸入觀察到的資料

1. 對每一個觀察到的資料 $V_u = v_u$ ：
 - (a) 將觀察到的資料編碼成一個近似函數 $\lambda_{v_u}^{New}$ 。
 - (b) 選擇包含 V_u 的一個分群 A ，更新 ϕ_A 和 λ_{v_u} ：


$$\phi_A \leftarrow \phi_A \lambda_{v_u}^{New}$$
$$\lambda_{v_u} \leftarrow \lambda_{v_u}^{New}$$

將觀察資料輸入到聯合樹裡面後，只有局部分群的位能是正確的，所以我們必須使用全體傳播的方法使我們得到準確的分群位能。

第四章 實驗與結果

4.1 實驗資料描述

實驗所使用的數據資料是由1個被解釋變數也就是良率變數以及30個解釋變數所組成，這些變數名稱爲Yield、Var2、Var3、…、Var31，這些數據是由晶圓自動化製程動作中所收集的實際數據，且數據中有152個樣本，因此數據爲一個152*31的矩陣數據。

本論文的主要目的是用上述資料來建立聯合樹模型，然後藉由聯合樹模型來進行錯誤診斷跟良率預測。

實驗部分由一個例子的模型發展程序開始，將原始資料先經由特徵選取、圖形化模型、節點排序建構出貝氏網路的架構，再將實際數據量化分類、統計計算將貝氏網路的機率部分計算出來，其做法如同圖1-2一樣，然後再使用第三章的聯合樹演算法建立聯合樹模型。接著是模型的應用和影響模型因素之探討。實驗中的部分方法會由 Statistica 軟體來執行，聯合樹模型的主要部分使用 Borland C++ Builder 撰寫程式建立模型。

4.2 模型發展程序

4.2.1 特徵選取

本論文參考[9]的方法，首先需要處理的問題包括歧異值的問題和缺少資料項的問題，再進行特徵選取的過程將解釋變數根據變數跟良率相關程度的高低做出排名，表4-1是跟良率相關程度比較高的前11個變數。

	1	2	3	4	5	6	7	8	9	10	11
變數	Var7	Var21	Var22	Var4	Var10	Var13	Var9	Var5	Var12	Var14	Var2

表 4-1 特徵選取（對良率變數比較有關係的變數 11 個）

4.2.2 圖形化模型

若從表4-1挑選出前七個變數和Yield這八個變數的集合 $V=\{\text{Var4},\text{Var7},\text{Var9},\text{Var10},\text{Var13},\text{Var21},\text{Var22},\text{Yield}\}$ ，藉由[9]中建立圖形化模型方法建立出圖形化模型，如圖4-1所示。

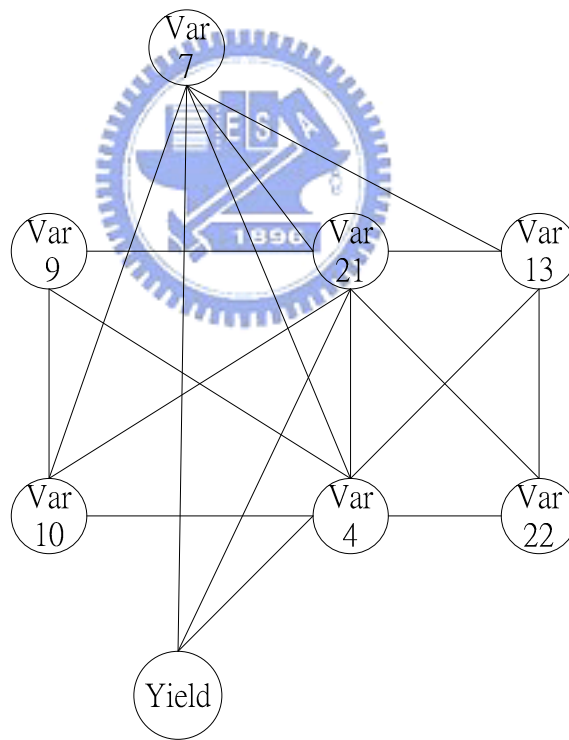


圖4-1 圖形化模型

4.2.3 節點排序

使用1.2小節所提到的節點排序，根據式子（1.1）計算Yield跟未經排序的節點{Var4,Var7,Var9,Var10,Var13,Var21,Var22}間的共同訊息量，然後根據得到的訊息量的由大到小做排序，如表4-2所示。而為了方便起見可以將經由節點排序之後的節點表示為 $\{V_1, V_2, \dots, V_7\}$ 而Yield表示為 V_8 ，其對應關係如表4-3所示。

	Var4	Var7	Var9	Var10	Var13	Var21	Var22
共同訊息量	0.0189	0.3441	0.1714	0.0258	0.0328	0.0644	0.0187
排序	6	1	2	5	4	3	7

表4-2 節點排序

簡化變數名稱	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
實際變數名稱	Var7	Var9	Var21	Var13	Var10	Var4	Var22	Yield

表4-3 對應變數名稱

利用經過排序的節點 $\{V_1, V_2, \dots, V_8\}$ 和Yield可以將圖4-1圖形化模型建構成方向性非循環圖形成為貝氏網路的基本架構，如圖4-2所示。

4.2.4 量化

由前一節得到排序後的節點為 $V = \{V_1, V_2, \dots, V_7, V_8\}$ ，將這些變數的實際數據經由1.2小節所提過的 k-means 分群理論將數據量化為L(Low)、M(Middle)和H(High)。如 V_8 經過k-means 分群理論將數據分為三類，實際樣本數值接近0.620338的為第一類樣本數值由L取代、實際樣本數值接近0.786138的為第二類樣本數值由M取代、實際樣本數值接近0.966858的為第三類樣本數值由H取代，如圖4-3所示。

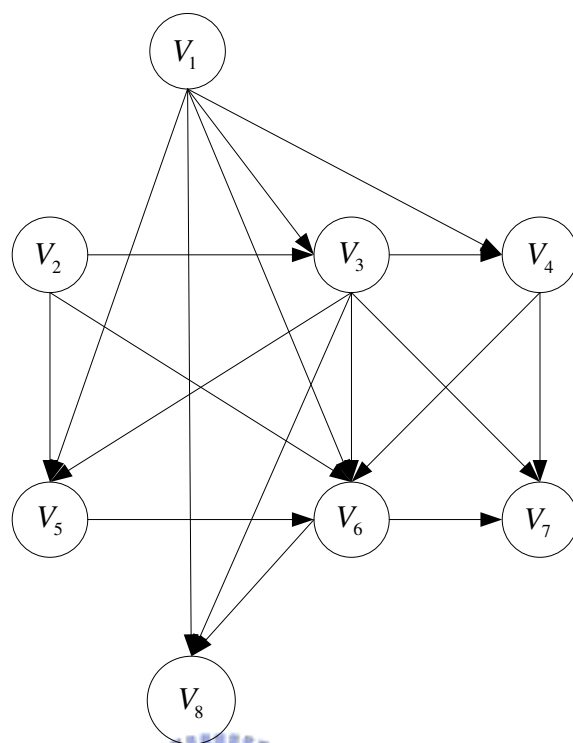


圖4-2 貝氏網路



樣本	數值
1	0.755725
2	0.914474
3	0.88488
4	0.93295
.	.
.	.
.	.
.	.
152	0.698225

K-means分群
分群數量：3

分群	代碼	中心
1	L	0.620338
2	M	0.786138
3	H	0.966858

樣本	類別
1	M
2	H
3	H
4	H
.	.
.	.
.	.
.	.
152	L

圖4-3 Yield量化結果

	L	M	H
V_1	-11.614	-11.018	-9.959
V_2	74.116	131.595	247.281
V_3	36.718	37.555	38.829
V_4	0.032	0.046	0.057
V_5	83.688	221.663	265.205
V_6	-10.298	-9.693	-9.669
V_7	36.532	37.345	38.249
V_8	0.620	0.786	0.966

表4-4 8個變數經k-means分三類，每各類別的中心數值

V_1	V_2	V_3	$P(V_3 V_1, V_2)$	V_1	V_2	V_3	$P(V_3 V_1, V_2)$
L	L	L	0	M	M	H	0
L	L	M	0.794	M	H	L	0
L	L	H	0.205	M	H	M	0
L	M	L	1	M	H	H	0
L	M	M	0	H	L	L	0
L	M	H	0	H	L	M	0
L	H	L	0	H	L	H	0
L	H	M	0	H	M	L	0
L	H	H	0	H	M	M	0
M	L	L	0.045	H	M	H	0
M	L	M	0.954	H	H	L	1
M	L	H	0	H	H	M	0
M	M	L	0.913	H	H	H	0
M	M	M	0.086				

表4-5 條件機率 $P(V_3 | V_1, V_2)$

4.2.5 統計計算

根據貝氏網路（圖4-2）的架構知道各節點間的關係，再利用量化後的結果經由統計計算把各個條件機率計算出來，因此就具備了貝氏網路的兩大部分（貝氏網路結構部分和數值部分（條件機率）），條件機率有： $P(V_1)$ 、 $P(V_2)$ 、 $P(V_3 | V_1, V_2)$ 、 $P(V_4 | V_1, V_3)$ 、 $P(V_5 | V_1, V_2, V_3)$ 、 $P(V_6 | V_1, V_2, V_3, V_4, V_5)$ 、 $P(V_7 | V_3, V_4, V_5)$ 、 $P(V_8 | V_1, V_3, V_6)$ ，其中 $P(V_3 | V_1, V_2)$ 的條件機率如表4-5所示。

4.2.6 聯合樹

我們使用先使用3.1節聯合樹演算法中的轉換經由演算法2~5將貝氏網路（圖4-2）轉換成聯合樹（圖4-5），在經過演算法2可以獲得moralized graph（圖4-4），經過演算法3可以獲得三角化圖形，由於圖4-4經過演算法3並未再填入邊線，所以圖4-4 moralized graph也是三角化圖形，根據演算法4可以從圖4-4中建立起三個分群，如： $C_1 = \{V_1, V_2, V_3, V_4, V_5, V_6\}$ 、 $C_2 = \{V_3, V_4, V_6, V_7\}$ 和 $C_3 = \{V_1, V_3, V_6, V_8\}$ 這三個分群，再運用演算法5建立聯合樹，如圖4-5所示。再由各個條件機率使用3.2節傳播裡的演算法6~9將三個分群的機率值計算出來。

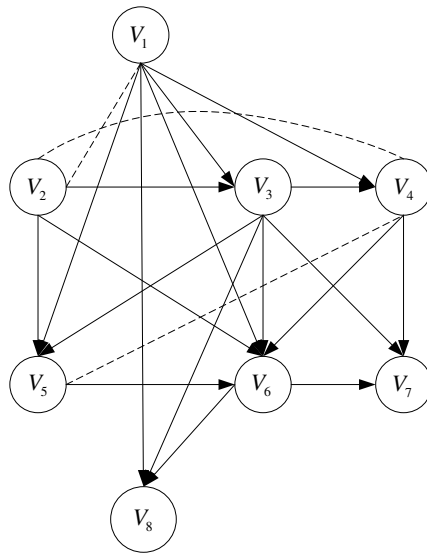


圖4-4 moralized graph（也是三角化圖形）

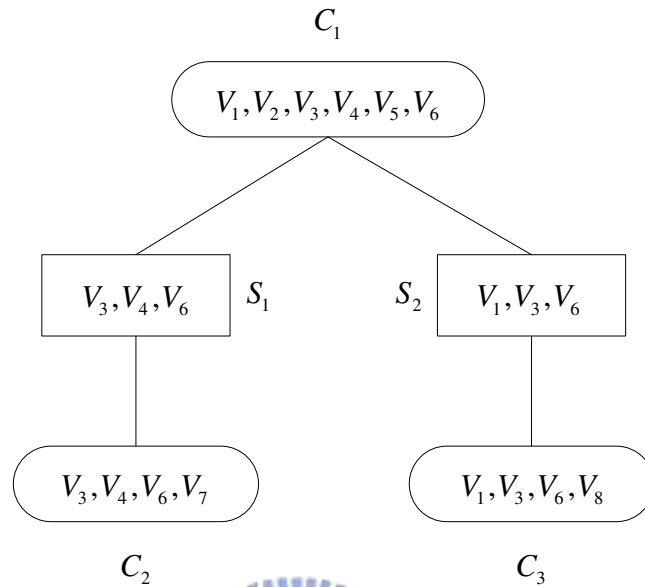


圖4-5 聯合樹

4.3 模型的應用

我們想利用聯合樹模型將跟低良率最有關係的變數狀態找出來，即是找出造成低良率的可能異常狀態；之後將這些異常狀態一個一個去除，預測每一個變數去除掉異常狀態後良率機率值是多少，再按照每一個變數去除異常狀態所得的良率機率值來做良率期望值的排名來當作每一個變數的影響力排名，其中根據期望值公式計算出良率期望值，如 (4.1) 式

$$E\{Yield\} = \sum_c P(c)R(c) \quad (4.1)$$

其中Yield為良率變數， $P(c)$ 為良率機率值而 $R(c)$ 為k-means 分類後的結果。下面為（對良率的）異常狀態搜尋與變數影響力排名的演算法：

演算法 12：(對良率的)異常狀態搜尋與變數影響力排名

N個變數的貝氏網路 $V = \{V_1, V_2, \dots, V_N\}$ ，其節點及合理的節點都是經過節點排序過後的節點且 V_N 為良率變數。

1. 運用第三章聯合樹的演算法計算聯合樹所有分群的機率值。
2. 記錄聯合樹中良率變數與每一個變數間有關的兩個分群(或一個分群)和兩分群間的分群跟交集，然後再運用式子(3.3)計算出包含良率變數跟每一個變數的聯合機率。
3. 將步驟2.所得到的聯合機率運用排斥做法將良率變數跟每一個變數的聯合機率計算出來，再除以良率的機率值，獲得在良率跟每個變數的條件機率(已知良率的情狀)。
4. 根據步驟3得知的條件機率，找出低良率發生的情形之下每一個變數的哪種狀態發生的機率最大，記錄於 $S(V_n)$ ， $1 \leq n \leq N-1$ 。
5. 第一次 $i=1$ ，從步驟4可以得知的 $S(V_i)$ 狀態，將演算法11中的步驟1.(a)的 $\lambda_{V_i}^{new}$ 裡的狀態與 $S(V_i)$ 的狀態相同的設為零(即狀態不發生)，然後再進行完演算法11之後，即可更新跟 V_i 有關的分群機率。
6. 執行聯合樹的全部訊息傳播後，將包含良率變數的分群機率透過排斥計算出良率機率值，再經由期望值公式(4.1)式計算出良率期望值，記錄在 $E'(V_i)$ ， $1 \leq i \leq N-1$ 。
7. 假如 $i=N-1$ 時，即停止；否則令 $i=i+1$ ，然後回步驟5。
8. 當 $i=N-1$ 結束時，按照 $E'(V_n)$ 的大小從大到小由1編號到N-1。

承接4.2小節的結果已經建立出一個八個變數的聯合樹模型圖4-5，將運用演算法12找出造成低良率的異常狀態和變數影響力排名：

- 步驟1.：於4.2小節已經將聯合樹的所有分群的機率值都計算出來，其中分群

為 $C_1 = \{V_1, V_2, V_3, V_4, V_5, V_6\}$ 、 $C_2 = \{V_3, V_4, V_6, V_7\}$ 和 $C_3 = \{V_1, V_3, V_6, V_8\}$ 。另外我們將良率變數的原始數據經由k-means分類量化成三類，如表4-6所示：

c	R(c)	P(c)	$E\{V_8\}$
L	0.620	0.220	0.830
M	0.786	0.384	
H	0.966	0.394	

表4-6 良率變數量化結果

表4-6中c為類別(class)，R(c)為量化後分類中心結果，P(c)良率機率， $E\{V_8\}$ 為原始期望值。

- 步驟2.：先找 V_1 跟 V_8 （良率變數）有關的分群：我們可以在分群 C_3 找到 V_1 跟 V_8 ，因此我們即找到包含 V_1 跟 V_8 的聯合機率：

$$P(V_1, V_3, V_6, V_8) = \frac{\phi_{C_3}}{1} = \phi_{C_3} \quad (4.2)$$

再找跟 V_2 和 V_8 有關的分群：分群 C_3 包含 V_8 分群 C_1 包含 V_2 ，兩分群間的分群跟交集： S_2 ，因此將兩分群 C_1 和 C_3 跟交集 S_2 使用 (3.3) 式計算出包含 V_2 跟 V_8 的聯合機率：

$$P(V_1, V_2, V_3, V_4, V_5, V_6, V_8) = \frac{\phi_{C_1} \phi_{C_3}}{\phi_{S_2}} \quad (4.3)$$

同理可以計算包含其他變數跟良率的變數的聯合機率。

- 步驟3.：將步驟2.得到的聯合機率運用排斥將良率變數跟每一個變數的聯合機率計算出來，將 (4.2) 式經過排斥可以得到 $P(V_1, V_8)$ ：

$$P(V_1, V_8) = \sum_{V_3, V_6} P(V_1, V_3, V_6, V_8) \quad (4.4)$$

將 (4.3) 式經過排斥可以得到 $P(V_2, V_8)$ ：

$$P(V_2, V_8) = \sum_{V_1, V_3, V_4, V_5, V_6} P(V_1, V_2, V_3, V_4, V_5, V_6, V_8) \quad (4.5)$$

同理我們可以計算出良率變數跟其他每一個變數的聯合機率，再將良率變數跟每一個變數的聯合機率都除以良率的機率值，得到良率變數和每一個變數間的條件機率，如表4-7和表4-8的條件機率 $P(V_1|V_8)$ 和 $P(V_2|V_8)$ 。

- 步驟4.：從步驟3.可以得知良率變數和每一個變數間的條件機率，如表4-7和表4-8就是其中兩個條件機率，於表4-7中，將 $V_8=L$ 的條件機率都做比較，找出機率值較大的將 V_1 的狀態記錄在 $S(V_1)$ 上所以 $S(V_1) = L$ ，再從表4-8得知 $S(V_2) = L$ ，同理我們可以得知 $V_8=L$ 跟每一個變數的哪個狀態最有關係（ $S(V_1) \sim S(V_7)$ 即可表示），如表4-9所示。我們也可從資料的分佈情況來判斷低良率的狀態大都落在變數的哪種狀態就可以知道低良率跟變數的哪個狀態最有關係，圖4-6(a)、圖4-6(b)和圖4-6(c)為六個變數的資料分佈情況，我們從圖4-6(a)可以發現圖的左下角都是低良率的狀態，所以我們可以從此圖中認定 $V_1=L$ 和 $V_2=L$ 的狀態都跟低良率很有關係，其他圖形也都可以判斷的出來且跟表4-9一致。



V_1	V_8	$P(V_1 V_8)$
L	L	0.739
L	M	0.460
L	H	0.287
M	L	0.109
M	M	0.375
M	H	0.573
H	L	0.150
H	M	0.163
H	H	0.139

表4-7 條件機率 $P(V_1|V_8)$

V_2	V_8	$P(V_2 V_8)$
L	L	0.773
L	M	0.756
L	H	0.414
M	L	0.040
M	M	0.101
M	H	0.445
H	L	0.185
H	M	0.142
H	H	0.140

表4-8 條件機率 $P(V_2|V_8)$

	$S(V_1)$	$S(V_2)$	$S(V_3)$	$S(V_4)$	$S(V_5)$	$S(V_6)$	$S(V_7)$
狀態	L	L	M	L	L	L	H
條件機率	0.739	0.773	0.529	0.402	0.775	0.781	0.516

表4-9 低良率跟變數最有關係的狀態

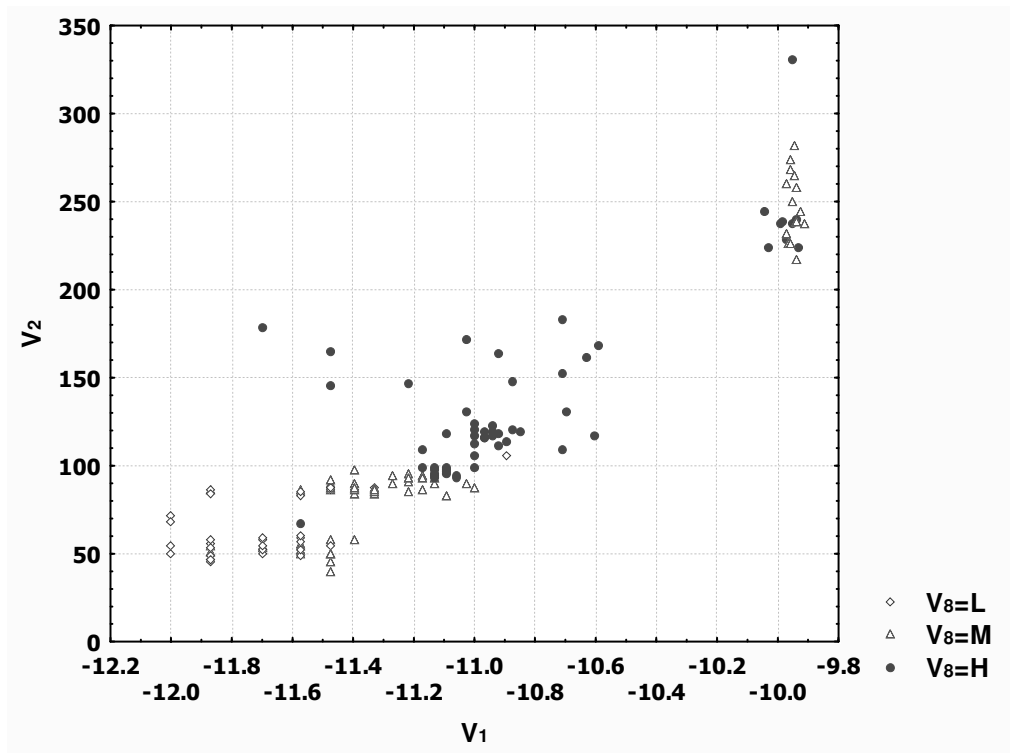


圖4-6(a) V_1 和 V_2 的資料分佈情況

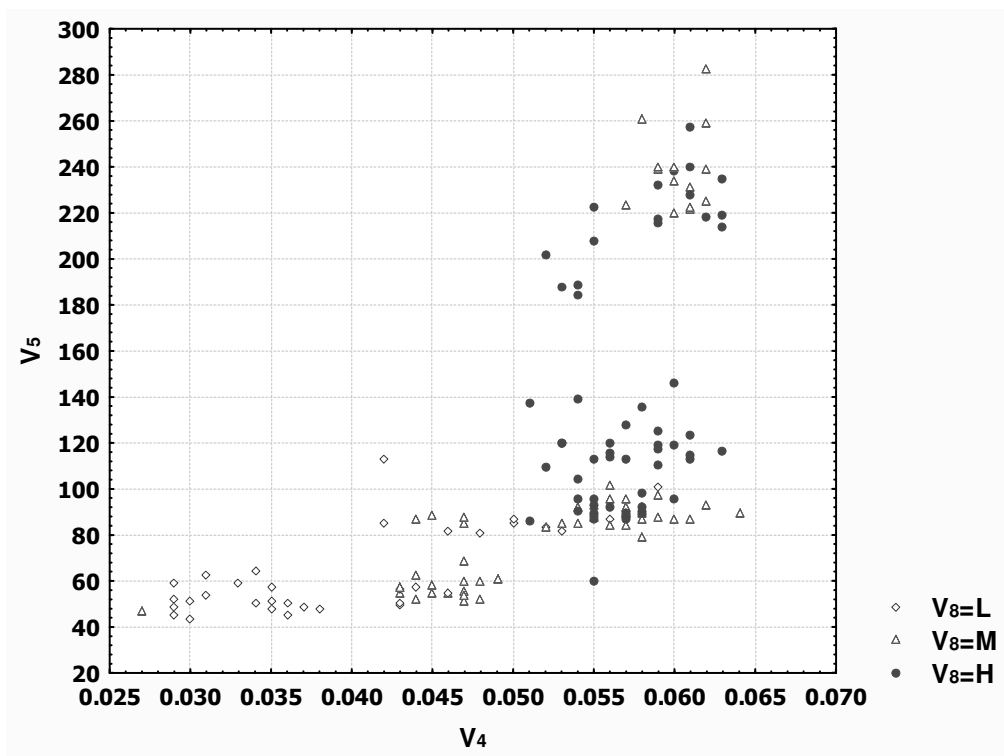


圖4-6(b) V_4 和 V_5 的資料分佈情況

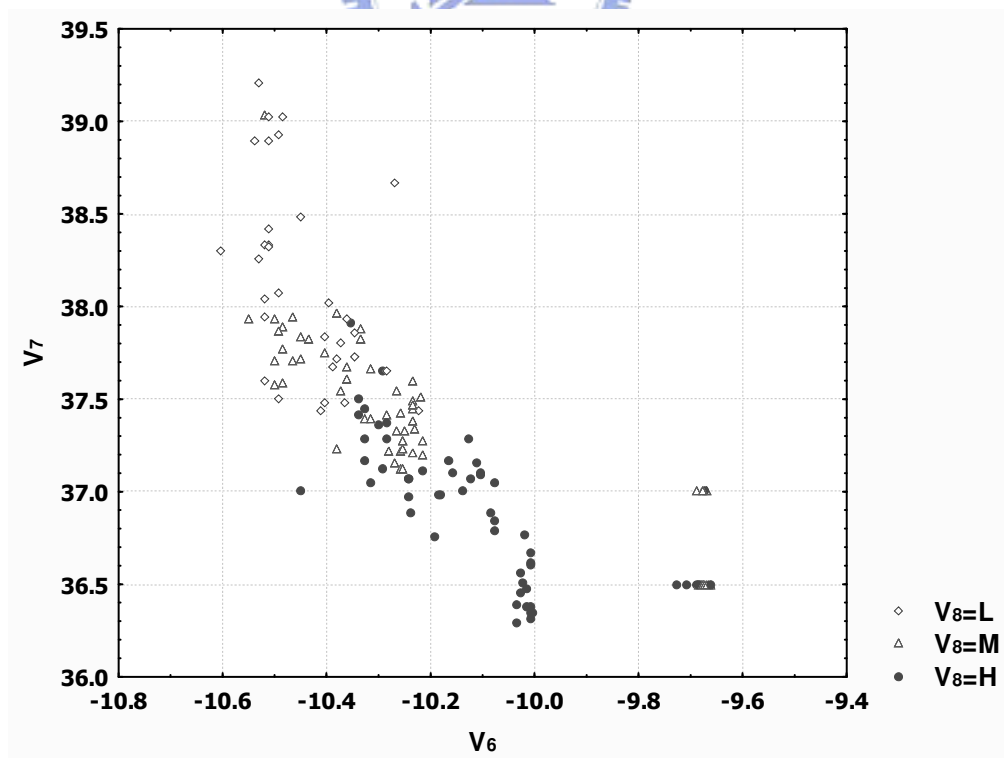


圖4-6(c) V_6 和 V_7 的資料分佈情況

- 步驟5.: $i=1$ 時，從表4-9得知 $S(V_1) = L$ 的狀態，所以將演算法11中的步驟 1. (a) 的 $\lambda_{V_1}^{new}$ 裡的三種狀態中的L狀態設為零，如 $\lambda_{V_1}^{new} = (0,1,1)$ ，然後進行完演算法11之後，即可更新跟 V_1 有關的分群機率。
- 步驟6.: 從步驟5. 可以知道有部分分群的機率已經更新過了，經過聯合樹的全體傳播過後，可以將包含良率變數的分群經過排斥將良率機率計算出來：

$$\begin{aligned}
 P(V_8) &= \sum_{C_3 \setminus \{V_8\}} \phi_{C_3} \\
 &= \sum_{V_1, V_3, V_6} P(V_1, V_3, V_6, V_8) \\
 &= (0.105, 0.379, 0.515)
 \end{aligned} \tag{4.6}$$

V_1 在未更新前的事前機率為 $P(V_1) = (0.453, 0.394, 0.151)$ ，而經過步驟5. 跟步驟6. 之後 V_1 的事前機率為 $P(V_1) = (0, 0.722, 0.277)$ ，由 V_1 更新過後的事情機率可以知道對低良率有關的L狀態已經去除了。因此我們可以將更新 $P(V_1)$ 後所得的良率經過 (4.1) 式計算出良率期望值：

$$\begin{aligned}
 E\{V_8\} &= \sum_c P(c)R(c) \\
 &= 0.105 * 0.620 + 0.379 * 0.786 + 0.515 * 0.966 \\
 &= 0.861
 \end{aligned} \tag{4.7}$$

然後將 (4.7) 式所得的期望值記錄在 $E'(V_1)$ 上，所以 $E'(V_1) = 0.861$ 。

- 步驟7.: 因為 $i \neq N-1$ ，所以 $i = i+1 = 2$ 然後回到步驟5.。
- 重複上面相同的作法：當 $i = 2$ 時，我們根據表4-8所示 $S(V_2) = L$ ，我們令 $\lambda_{V_2}^{new} = (0,1,1)$ ，然後經由演算法11更新跟 V_2 有關的分群機率，再經由聯合樹的全體傳播之後，可以得到更新過後 V_2 的事前機率 $P(V_2)$ ，如表4-10(a)所示，跟更新 $P(V_2)$ 之後獲得的良率期望值，如表4-10(b)，再將表4-10(b)運用 (4.1) 式計算出良率期望值記錄在 $E'(V_2)$ 上，再跟未更新前的良率期望值做比較看看到底提昇或下降多少比例，如表4-10(c)所示。我們可以根據圖4-2

的網路結構將解釋變數 $V_1 \sim V_7$ 分為三類：可控變數 $\{V_1, V_2\}$ 、中間變數 $\{V_3, V_4, V_5, V_6\}$ 和無關係變數 $\{V_7\}$ 。無關係變數 $\{V_7\}$ 對 V_8 完全沒有影響，所以我們不考慮無關係變數 $\{V_7\}$ 對 V_8 的影響，因此討論 $i=1$ 到 $i=6$ 對 V_8 的影響，我們即可獲得表4-10。

- 步驟8.：將表4-10(c)良率期望值按照大小從大到小由1編號到6，也表示變數對良率的影響力排名，如表4-11所示。

i	$P(V_i)$
1	(0,0.722,0.277)
2	(0,0.596,0.403)
3	(0.685,0,0.314)
4	(0,0.269,0.730)
5	(0,0.586,0.413)
6	(0,0.482,0.517)

(a)

i	$P(V_8)$
1	(0.105,0.379,0.515)
2	(0.133,0.250,0.616)
3	(0.223,0.220,0.555)
4	(0.153,0.407,0.439)
5	(0.228,0.351,0.420)
6	(0.245,0.397,0.356)

(b)

i	$E'(V_i)$	改良率
1	0.861	3.7%
2	0.875	5.4%
3	0.849	1.9%
4	0.840	1.2%
5	0.824	-0.7%
6	0.809	-2.5%

(c)

表4-10 (a)更新過後的變數的事前機率

(b)更新過後的良率機率值

(c)良率期望值及其改良率

	V_1	V_2	V_3	V_4	V_5	V_6
排名	2	1	3	4	5	6

表4-11 變數對良率的影響力排名

我們可以在表4-10(c)中竟然發現更新 V_5 和 V_6 的改良率是負值，由於我們使用聯合樹去除變數的異常狀態來更新機率值，但由於更新機率值是更新整個分群的機率，如更新 V_6 的話，三個分群都有包含 V_6 ，而為了符合聯合樹的理論所有變數的機率值都會改變，包含 V_1 和 V_2 這兩個可控變數的機率值，接著可以發現改變 V_6 之後在根據條件機率將變數 $V_1 \sim V_5$ 的異常狀態列出來，如表4-12所示，由於改變 V_6 讓異常狀態產生大變動，而且最主要的是使 V_1 和 V_2 這兩個可控變數的異常狀態都改變了，所以最後會對良率造成不好的影響，結果改良率才會是負值。

	V_1	V_2	V_3	V_4	V_5
異常狀態	H	H	M	H	L
條件機率	0.394	0.592	0.394	0.447	0.341

表4-12 $V_1 \sim V_5$ 的異常狀態

就物理觀念來說更新中間變數 $\{V_3, V_4, V_5, V_6\}$ 並不能影響 V_1 和 V_2 這兩個可控變數的機率，若我們強制更新中間變數的話，就有可能會對良率造成不好的影響，所以我們最好只更新 V_1 或 V_2 這兩個可控變數。我們將 V_1 或 V_2 的變數各個狀態對良率做一些探討，兩個變數主要有四種組合： V_1 和 V_2 各去除一種狀態、 V_1 去除一種狀態和 V_2 去除兩種狀態、 V_1 去除兩種狀態和 V_2 去除一種狀態、 V_1 和 V_2 各去除兩種狀態，下表就是各個狀態組合的良率值、期望值和改良率。 V_1 和 V_2 的三種狀態對低良率狀態的影響力分別為L狀態第一、H狀態次之和最後的M狀態，我們可以從表中發現若 V_1 和 V_2 同時將L狀態和M狀態去除掉的話(只剩下最沒影響力的M狀態)就會良率值會得到最好的改良，其改良率達到15%；若 V_1 和 V_2 同時將H狀態和M狀態去除掉的話(只剩下最有影響力的L狀態)就會良率值會得到最差的改良，其改良率達到-14.1%，其他狀態就如下表所示。

去除變數異常狀態		$P(V_8)$	$E\{V_8\}$	改良率
V_1	V_2			
L	L	(0.120,0.252,0.627)	0.879	5.9
L	M	(0.114,0.445,0.440)	0.846	2.0
L	H	(0.089,0.376,0.534)	0.867	4.5
M	L	(0.149,0.304,0.545)	0.859	3.6
M	M	(0.399,0.440,0.159)	0.748	-9.7
M	H	(0.333,0.398,0.268)	0.779	-6.0
H	L	(0.129,0.210,0.660)	0.884	6.5
H	M	(0.283,0.448,0.267)	0.787	-5.1
H	H	(0.203,0.389,0.407)	0.826	-0.4
L,M	L	(0.154,0.473,0.371)	0.827	-0.2
L,M	M	(0.208,0.426,0.364)	0.817	-1.5
L,M	H	(0.259,0.382,0.358)	0.807	-2.6
L,H	L	(0.107,0.167,0.725)	0.899	8.3
L,H	M	(0.077,0.452,0.469)	0.858	3.3
L,H	H	(0.024,0.373,0.601)	0.890	7.3
M,H	L	(0.148,0.248,0.603)	0.870	4.8
M,H	M	(0.463,0.445,0.091)	0.725	-12.5
M,H	H	(0.358,0.403,0.238)	0.769	-7.2

(a)

去除變數異常狀態		$P(V_8)$	$E\{V_8\}$	改良率
V_1	V_2			
L	L,M	(0.190,0.400,0.409)	0.828	-0.1
L	L,H	(0.073,0.151,0.774)	0.913	10.1
L	M,H	(0.095,0.456,0.447)	0.851	2.5
M	L,M	(0.275,0.391,0.332)	0.800	-3.5
M	L,H	(0.064,0.245,0.689)	0.900	8.4
M	M,H	(0.429,0.452,0.117)	0.736	-11.3
H	L,M	(0.319,0.319,0.361)	0.798	-3.7
H	L,H	(0.001,0.136,0.862)	0.941	13.4
H	M,H	(0.275,0.480,0.244)	0.784	-5.4
L,M	L,M	(0,0.608,0.391)	0.856	3.2
L,M	L,H	(0.259,0.382,0.358)	0.807	-2.6
L,M	M,H	(0.259,0.382,0.358)	0.807	-2.6
L,H	L,M	(0.263,0.320,0.416)	0.817	-1.4
L,H	L,H	(0.002,0.06,0.934)	0.954	15.0
L,H	M,H	(0.033,0.484,0.482)	0.867	4.5
M,H	L,M	(0.367,0.319,0.313)	0.781	-5.8
M,H	L,H	(0,0.2,0.8)	0.930	12.1
M,H	M,H	(0.486,0.476,0.037)	0.712	-14.1

(b)

表4-13 去除 V_1 和 V_2 不同組合的狀態所得的良率、良率期望值和改良率

4.4 影響應用因素之探討

4.4.1 樣本數分析

我們想探討樣本數量的多寡是否會對找出低良率跟每一個變數最有關係的狀態和變數對良率的影響力排名造成影響，所以我們承接 4.2 小節所建立的貝氏網路模型圖 4-2，對樣本數量作分析，分別為：50 個樣本、100 個樣本和 152 個樣本。8 個經過節點排序的變數為： $\{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8\}$ ，其相對應的名稱如表 4-3。把三個樣本數量不同的貝氏網路以 k-means 的方法將資料分 3 類，表 4-14 為三個樣本數量不同的良率變數在 k-means 中的平均值。

Class	Number of cases		
	50	100	152
S	0.757	0.737	0.620
M	0.824	0.826	0.786
H	0.923	0.961	0.966

表 4-14 50、100 和 152 個樣本數的良率變數於 k-means 三個類別的平均值（或中心點）

如同前面 4.3 節作法一樣，可以因此得到表 4-15 三個樣本數量不同的低良率跟每一個變數最有關係的狀態和表 4-16 三個樣本數量不同的變數對良率的影響排名。觀察表 4-15 裡的三個表格可以發現 7 個變數中有 2 個變數的狀態會有不同，但由於某些變數的樣本數太少而且有用訊息也太少，使得低良率跟每一個變數最有關係的狀態和影響力的排名會有些影響，我們相信樣本數越多時，我們所得的結果也會越準確，不然我們只能根據我們手上僅有數據來看數據回答問題了。

	$S(V_1)$	$S(V_2)$	$S(V_3)$	$S(V_4)$	$S(V_5)$	$S(V_6)$
狀態	L	L	H	L	L	L
條件機率	0.668	0.608	0.524	0.548	0.563	0.598

(a)50 個樣本

	$S(V_1)$	$S(V_2)$	$S(V_3)$	$S(V_4)$	$S(V_5)$	$S(V_6)$
狀態	L	L	H	M	L	L
條件機率	0.521	0.741	0.706	0.408	0.750	0.755

(b)100 個樣本

	$S(V_1)$	$S(V_2)$	$S(V_3)$	$S(V_4)$	$S(V_5)$	$S(V_6)$
狀態	L	L	M	L	L	L
條件機率	0.739	0.773	0.529	0.402	0.775	0.781

(c)152 個樣本

表4-15 三個樣本數量不同的低良率跟變數最有關係的狀態

	V_1	V_2	V_3	V_4	V_5	V_6
排名	2	5	1	6	4	3
良率期望值	0.846	0.842	0.848	0.838	0.843	0.844

(a)50 個樣本

	V_1	V_2	V_3	V_4	V_5	V_6
排名	3	2	1	6	4	5
良率期望值	0.869	0.873	0.886	0.826	0.862	0.845

(b) 100 個樣本

	V_1	V_2	V_3	V_4	V_5	V_6
排名	2	1	3	4	5	6
良率期望值	0.861	0.875	0.849	0.840	0.824	0.809

(c) 152個樣本

表4-16 三個樣本數量不同的變數對良率的影響排名

4.4.2 變數個數分析

於本小節我們想探討變數個數的多寡是否會對找出低良率跟每一個變數最有關係的狀態和影響良率的變數排名造成影響，從 3 個變數(從表 4-1 選出前 2 個變數再加上良率變數)分析到 6 個變數(從表 4-1 選出前 5 個變數再加上良率變數)的貝氏網路，來看看變數的多寡會不會造成影響。且所有的貝氏網路皆為 152 個樣本而且每一個變數資料都經由 k-means 分成三類。

	Number of variables			
	3	4	5	6
var 7	L	L	L	L
var 21	M	M	M	M
var 22		H	H	H
var 4			L	L
var 10				L

	Number of variables			
	3	4	5	6
var 7	0.868	0.817	0.897	0.865
var 21	0.863	0.863	0.863	0.896
var 22		0.812	0.847	0.864
var 4			0.856	0.816
var 10				0.834

(a)

(b)

	Number of variables			
	3	4	5	6
var 7	1	2	1	2
var 21	2	1	2	1
var 22		3	4	3
var 4			3	5
var 10				4

(c)

表 4-17 (a)：低良率與每一個變數最有關係的狀態（3~6 個變數）

(b)：（調整過後）良率期望值（3~6 個變數）

(c)：影響良率的變數排名（3~6 個變數）

	Number of variables									
	3	4	5	6	7	8	9	10	11	12
time(sec)	0.157	0.219	0.251	0.563	1.218	1.844	59.945	114.093	1784.171	17792.609

表 4-18 分析時間

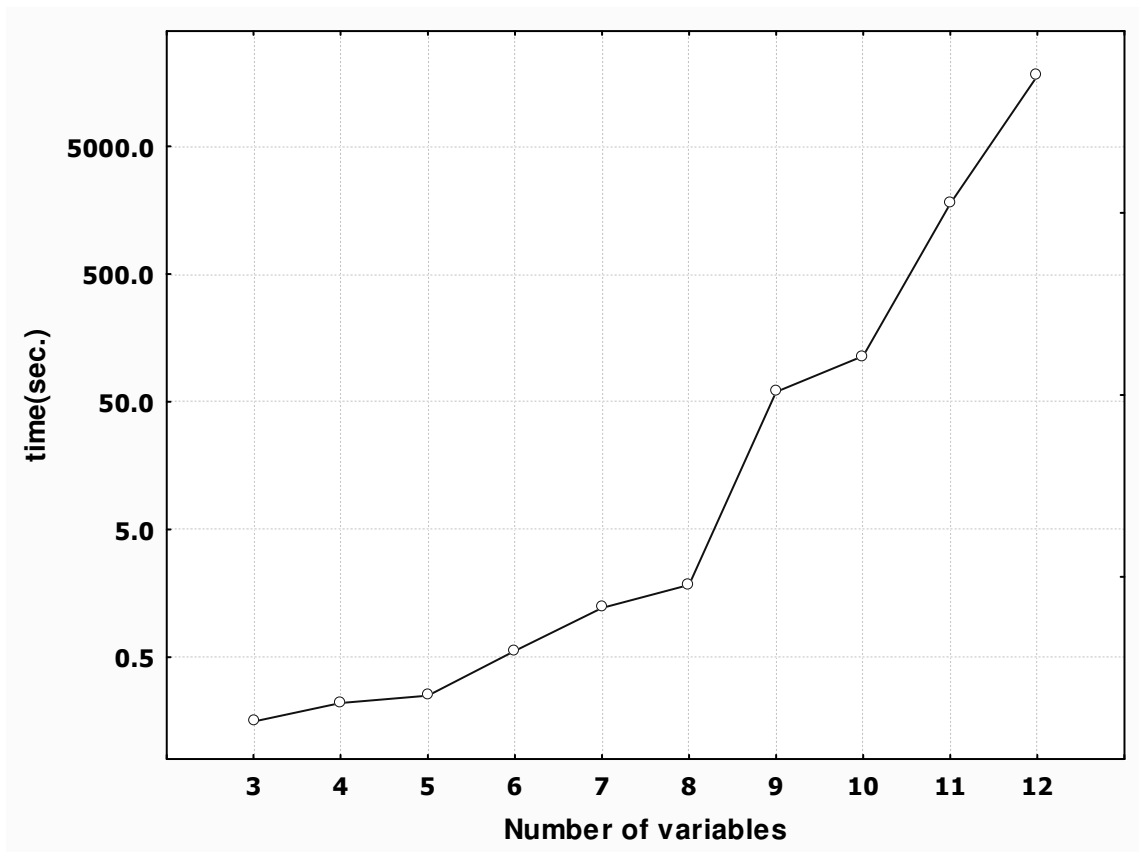


圖 4-7 分析時間

根據 4.3 節的做法，找出低良率跟每一個變數最有關係的狀態和變數對良率的影響力排名，如表 4-17 所示，我們從觀察表 4-17(a) 可以發現增加的變數不會對已經存在變數造成影響，但從表 4-17(c) 影響良率的變數排名可以發現增加的變數的排名大都是排在後面且會些微的影響前面的排名。

隨著變數的增加，所需要的資料量、計算量和分析時間也會跟著增加，我們隨著變數的增加記錄分析的時間，如表 4-18 所示，若將表 4-18 由圖形表示出來的話，如圖 4-7 (Y 座標為指數形式) 所示，我們可以發現圖形略呈指數關係，所以每增加一個變數分析時間就會呈指數形式成長。於 13 個變數時，需要處理大量的數據，但由於程式本身無法負荷，所以只能有效地分析到 12 個變數。

4.4.3 良率變數量化分析

我們將良率變數的數據分成 2 類和 3 類。我們將對這兩種狀態 (2 類和 3 類) 的良率變數作個實驗，看看不同的量化分類對找出低良率跟每一個變數最有關係的狀態和變數對良率的影響力排名是否有影響。我們選擇 8 個經過節點排序的變數為： $\{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8\}$ (相對應的名稱如表 4-3) 和 152 個樣本數的貝氏網路 (如：圖 4-2) 來作實驗，我們將良率變數分成 3 類和 2 類剩下的 7 個變數都分為 3 類，其 k-means 分類結果和事前機率和良率期望值，如表 4-19 所示。

C	2 類			3 類		
	R(C)	P(var 1)	E(var 1)	R(C)	P(var 1)	E(var 1)
L	0.718	0.547	0.824	0.620	0.220	0.820
M				0.786	0.384	
H	0.9542	0.452		0.966	0.394	

表 4-19 k-means 分類結果和事前機率和良率期望值

我們可以應用 4.3 節的做法找出(1)兩種良率分類中低良率跟其他變數中最有關係的狀態，如表 4-20 所示；(2)兩種良率分類經過更新的良率期望值，如表 4-21 所示、(3)：兩種良率分類的變數對良率的影響力排名，如表 4-22 所示。從表 4-20 中可以發現 7 個中有 2 個狀態是不同的，這應該是由於 k-means 分類之後，在良率分兩類的情形之下低良率的比率變高了，所以比較多的樣本被認定為低良率，所以會造成這樣的結果，但整體的差異不會太大，所以良率分類上的應該沒有太大的問題。

良率量化	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆
2類	L	L	M	H	L	L
3類	L	L	M	L	L	L

表 4-20 兩種良率分類中良率變數中的 L 狀態跟其他變數中最有關係的狀態

良率量化	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆
2類	0.857	0.879	0.863	0.772	0.848	0.840
3類	0.861	0.875	0.849	0.840	0.824	0.809

表 4-21 良率期望值

良率量化	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆
2類	3	1	2	6	4	5
3類	2	1	3	4	5	6

表 4-22 兩種良率分類的變數對良率的影響力排名

4.5 模型驗證

當我們於 4.2 小節將聯合樹的模型建構完成後，可以從演算法 12 的步驟 3 獲得我們所需要的條件機率，如 $P(\text{var}7 | \text{Yield} = L)$ 、 $P(\text{var}9 | \text{Yield} = L)$ 等等的條件機率，而根據這些條件機率我們於演算法 12 的步驟 4 可以找出跟低良率比較有關係的變數狀態。但在一個一百個樣本的貝氏網路中，經由聯合樹找出跟低良率比較有關係的變數狀態，比如說：var7 的 L 狀態就比 var7 的其他狀態跟低良率更有關係，但若還有後續的樣本低良率是否還是跟 L 狀態有關係呢？因此我們需要做個實驗看看，我們選擇一個 8 個經過節點排序的變數為： $\{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8\}$ （相對應的名稱如表 4-3）和 100 個樣本的貝氏網路來做聯合樹的模型，且每個變數皆分為 3 類，之後再拿 52 個樣本來作驗證。

如同前面的做法一樣，我們將 100 個樣本根據 4.2 小節建立起聯合樹的模型，再應用演算法 12 中步驟 1 到步驟 4 將跟低良率比較有關係的變數狀態找出來，如表 4-23 所示。再將 52 個測試樣本拿出來測試看看，是否在低良率發生的情形之下，跟低良率比較有關係的其他變數狀態是否會跟表 4-23 一樣呢？經過計算，當 52 個測試樣本中低良率發生的情形之下其他 7 個變數的所有狀態，如表 4-24 所示，52 個樣本中低良率的狀態有 16 個，其他變數的狀態皆記錄在表 4-24 中。可以根據表 4-24 找出七個變數跟低良率比較有關係的狀態，如表 4-25 所示。可以發現表 4-25 跟表 4-23 一模一樣，所以我們驗證的結果是正確的。

	V_1	V_2	V_3	V_4	V_5	V_6
狀態	L	L	M	L	L	L
條件機率	0.658	0.738	0.510	0.380	0.738	0.742

表 4-23 跟低良率比較有關係的變數狀態

	L	M	H
V_1	16	0	0
V_2	16	0	0
V_3	0	10	6
V_4	9	6	1
V_5	16	0	0
V_6	16	0	0

表 4-24 $V_8 = L$ 的情形之下，其他變數的狀態

(測試樣本， $V_8 = L$ 有 16 個)

	V_1	V_2	V_3	V_4	V_5	V_6
狀態	L	L	M	L	L	L

表 4-25 跟低良率比較有關係的變數狀態

(測試樣本)

第五章 結論

本論文主要研究對像是晶圓製程資料，根據 4.2 節所述根據數據資料來排序節點，搭配圖形化模型形成一個貝氏網路的基本架構，再由量化後的數據根據貝氏網路的架構將數值部分條件機率統計計算出來，因此形成一個完整的貝氏網路。依據第三章聯合樹的演算法可以將貝氏網路轉換成聯合樹的模型，聯合樹的應用如 4.3 節的應用一樣，可以找出造成低良率的每一個變數的異常狀態，更進一步去除變數的異常狀態提昇良率。這種診斷的方法也可以應用在很多地方上，如醫學、化學、氣象等等都可以解決他們的問題。

在實驗中從建立聯合樹模型開始，然後再應用聯合樹模型來診斷出造成低良率的每一個變數的異常狀態，更進一步去除變數的異常狀態提昇良率，接著討論一些影響應用的因素，討論不同樣本數對於本論文應用的影響，跟探討不同變數量對於本論文應用的影響，接著藉由不同的良率分類看看對於本論文應用的影響，最後要驗證診斷出來的異常狀態是否正確。

從實驗的結果我們可以得到以下幾點：

- (1) 樣本量的多寡對於診斷出造成低良率的每一個變數的異常狀態有些微的影響，樣本數太少所得到的結果可能有一點點不正確，因此希望能在足夠的樣本數量下建立聯合樹模型。
- (2) 從不同變數數量的結果來看，變數量不同並不會對診斷出造成低良率的每一個變數的異常狀態有些微的影響，只會對排名造成一點點影響而已，最大的影響是當變數量多時，處裡的資料量也會跟著多了起來，最後程式會因此而無法負荷。
- (3) 從不同的良率分類的結果來看，不同的良率分類對診斷出造成低良率的每一個變數的異常狀態只有些微的影響。

談到最後本論文的改進希望能從圖形化模型著手，當變數量越多時，若有分群包含的變數量佔所有變數量的 80%~90%的話，那麼可能計算聯合機率時需要計算到全體變數的聯合機率，最後可能會造成程式無法負荷而無法計算下去。所以當分群包含的變數量為所有變數量的 80%~90%時，重新建立更精簡、更少邊線的圖形化模型，使分群所包含的變數量減少，增加可以分析的變數量。



參考文獻

- [1] Enrique Castillo, José Manuel Gutiérrez, Ali S. Hadi, Expert Systems and Probabilistic Network Models, Springer-Verlag, Inc., New York, 1997.
- [2] Rober G. Cowell, et al., Probabilistic Networks and Expert Systems, Springer-Verlag, Inc., New York, 1999.
- [3] Finn V. Jensen, Bayesian networks and decision graphs, Springer-Verlag, Inc., New York, 2001.
- [4] Finn V. Jensen, An introduction to bayesian networks, Springer-Verlag, Inc., New York, 1996.
- [5] Leslie Tung, A Clique Tree Algorithm Exploiting Context Specific Independence, University of British Columbia, Master's Thesis, 2000.
- [6] Henrik Bengtsson, Bayesian Networks - A Self-Contained Introduction with Implementation Remarks, University of California, Master's Thesis, 1997.
- [7] Cecil Huang, "Inference in Belief Networks: A Procedural Guide", International Journal of Approximate Reasoning, pp. 111-158, 1994.
- [8] Nicandro Cruz-Ramirez, Algorithms Based on Information Theory that Build Bayesian Networks from Data, University of Sheffield, Master's Thesis, 2000.
- [9] 郭宇豪，「圖形化高斯模型應用於自動化生產之資料」，國立交通大學，碩士論文，民國93年。
- [10] 李家祺，人工智慧與專家系統，松岡，台北，民國78年。