

基於模糊群集決策樹應用於離子植入機故障偵測

洪士程
朝陽科技大學資訊工程系
e-mail : schong@cyut.edu.tw

蕭宇良
朝陽科技大學資訊工程系
e-mail : s9727633@cyut.edu.tw

摘要

在本文中提出一個模糊群集決策樹(FCDT)，應用於較多類別和連續性質屬性的資料分類問題上。FCDT 結合了分群演算法以及決策樹 C4.5。所提出的分群演算法以可分離矩陣和模糊規則為基礎得到分割屬性，利用這些分割屬性將資料分群至終端群集。在終端群集裡，包含了超過一個以上的類別，將進一步以 C4.5 進行分類。我們已成功地應用 FCDT 進行離子植入機中正在運作的晶圓片的分類測試，並和現有的商業軟體 See5 和 CART 來比較分類結果與計算時間。結果顯示 FCDT 比 See5 和 CART 都具有較短的訓練時間，同時具有更高的分類準確度。

關鍵詞：分群演算法，決策樹，C4.5，模糊規則，離子植入機。

Abstract

In this paper, we propose a fuzzy clustering decision tree (FCDT) for the classification problem with large number of classes and continuous attributes. The FCDT combines a proposed clustering algorithm with the decision tree C4.5. The proposed clustering algorithm split the data set into terminal clusters using splitting attributes based on a separation matrix and fuzzy rules. The terminal clusters consisting of the data of more than one class will be further classified using the C4.5. We have successfully applied the FCDT to the classification problem of the working wafers in an ion implanter, and compared the classification results and the computation time with the existing software See5 and CART. The comparison results show that the FCDT retains the classification accuracy of CART and See5 at the same time retains shorter training time.

Keywords: Clustering algorithm, decision tree, C4.5, fuzzy rule, ion implanter.

1. 前言

目前存在幾種分類技術可處理連續性質屬性的資料分類問題，例如類神經網路方法 [1]-[2]，最大值可能性方法 [3]-[4]，模糊集理論為基礎的方法 [5]-[6]，決策樹 [7]-[8]，等等。其中，類神經網路方法可用來處理無規則分佈的資料，並且在使用彈性上佔有優勢，但耗費計算成本以及初始權重值的隨機設定容易造成結果的差異。最大值可能性方法廣泛應用在只能獲得極少數感測資料的分類問題上，然而當目標類別無法以統計模型充分描述時，便會降低了此方法的性能。模糊集理論為基礎的方法雖已經成功地應用在特徵分類問題上，不過當類別的數量與屬性的數目變大時將提高計算的複雜性。

決策樹主要設計做為非連續數值資料類別的分類。目前最被廣泛使用的分類演算法是基於熵的決策樹演算法，例如 ID3 [9] 和 C4.5 [10] 決策樹分類演算法。分類與迴歸樹 (CART) [11]-[12] 和 Quest (SLIQ) [13] 分類算法是採用吉尼係數 (Gini index) 作為生成二元決策樹的分割依據。C4.5 對於連續性質屬性的資料是一個很可靠的分類技術，其優點是不須知道資料的分佈情形。然而當類別數量與資料的筆數變大時，計算複雜度的增加將成指數型增長。事實上，所得到初長成的完整決策樹，必須經過修剪，才能獲得最小成本的分類樹，也就是以最少層的樹來達到最有效的分類。因此，利用這個性質可以維持 C4.5 高的分類精確度且克服計算複雜度。為了達成這個目標，在本文中將提出一個以可分離矩陣為基礎的分群演算法，作為 C4.5 分類演算法的預先處理步驟。這個分群演算法先對所有資料產生一個群集分割樹，最後在群集分割樹的終端群集直接套用 C4.5 方法，因為在終端群集裡面的資料類別數量已比原來少得多，這將大大地減少計算的複雜度且仍可維持高的分類正確率。

我們以下面的方式來安排本篇論文。在第 2 節，介紹所提出的分類器-模糊群集決策樹 (Fuzzy Clustering Decision Tree, FCDT)。在第 3

節裡，我們將介紹基於分類的錯誤偵測和隔離策略，在第4節裡，將FCDT應用於半導體製程中晶圓片的配方分類，測試結果可獲得10重交疊驗證的分類精確度，並和現有的商業軟體 See5 和 CART 做比較。在第5節，將做一個結論。

2. 模糊群集決策樹(FCDT)

如同先前所提到的，FCDT 結合了分群演算法與 C4.5 方法。分群演算法會依據群集分割樹把所有資料分配到終端群集(Leaf Cluster, LC)。

2.1 分離矩陣為基礎的群集演算法

任何一種產品在製作過程期間控制設定是維持不變的，所以在每一個產品類別的資料樣式中，具有連續性質屬性的數值可能以特定的機率密度分佈在某一範圍。於是，兩個產品類別的資料間能夠透過屬性值的重疊程度來判斷是否可分離。為了決定兩個類別資料間的可分離性，首先定義 $D(C_i, C_j)_k$ 為類別 C_i 與 C_j 之間對於屬性 k 的可分離指標。

$$D(C_i, C_j)_k = \begin{cases} 0, & \text{若 } C_i \text{ 和 } C_j \text{ 對於屬性 } k \text{ 可分離,} \\ 1, & \text{其他} \end{cases} \quad (1)$$

$D(C_i, C_j)_k$ 的值可利用 3 倍標準差[14]計算得到，詳細描述如後。以 X_i^k 代表屬性為 k 而類別為 C_i 的隨機變數，而 μ_i^k 與 σ_i^k 分別代表 X_i^k 的平均值與標準差。我們假定 $\mu_i^k < \mu_j^k$ ，如果 $\mu_j^k - 3\sigma_j^k$ 大於 $\mu_i^k + 3\sigma_i^k$ 則可以判定類別 C_i 和類別 C_j 在屬性為 k 時，其重疊的程度將是非常小，如圖 1 所示，類別 C_i 和類別 C_j 是可以分離的情況，而在圖二則表示為不可分離。因此，可以定義 $D(C_i, C_j)_k$ ，使得類別 C_i 與 C_j 的可分離指標可簡單判斷如下：

$$D(C_i, C_j)_k = \begin{cases} 0, & \text{若 } \mu_i^k + 3\sigma_i^k < \mu_j^k - 3\sigma_j^k, \\ 1, & \text{其他} \end{cases} \quad (2)$$

最後定義 $[D(C_i, C_j)_k]$ 為類別 C_i 與 C_j 之間對於屬性 k 的可分離矩陣，矩陣中的第 (i, j) 元素就是可分離指標 $D(C_i, C_j)_k$ 。

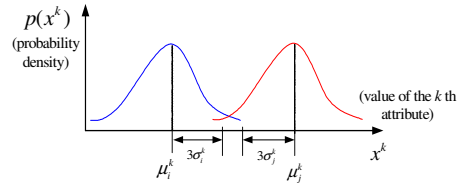


圖 1. 說明類別 C_i 與 C_j 為可分離

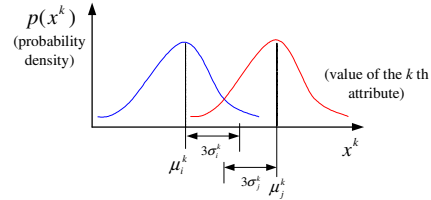


圖 2. 說明類別 C_i 與 C_j 為不可分離

2.2 利用可分離矩陣分割群集

根群集包含了所有類別的資料，以 C_{r_0} 表示。將 C_{r_0} 裡的每一個類別視為一個節點， $[D(C_i, C_j)_{k_1}]$ 可視為屬性 k_1 對 C_{r_0} 裡所有節點的可分離矩陣。也就是當 $D(C_i, C_j)_{k_1} = 1$ 時， C_i 與 C_j 的節點會有一條弧線相連結。依照可分離矩陣所建構出的圖形稱為可分離圖形，裡面包含著可以分離的子圖形。每一個連接的子圖形表示一個群集，而不連接的子圖形數目表示可使用屬性 k_1 從 C_{r_0} 分割產生群集的數目。

例如，圖 3(a) 的可分離矩陣建構出的可分離圖形如圖 3(b) 所示，包含了二個不連接的群集 A 與 B。群集 A 由 C_1 、 C_2 、 C_3 、 C_4 四個類別組成，另一個群集 B 由 C_5 、 C_6 、 C_7 、 C_8 四個類別組成。所得到的群集

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	1	1	0	0	0	0	0	0
C_2	1	1	1	0	0	0	0	0
C_3	0	1	1	1	0	0	0	0
C_4	0	0	1	1	0	0	0	0
C_5	0	0	0	0	1	1	1	0
C_6	0	0	0	0	1	1	1	0
C_7	0	0	0	0	0	1	1	0
C_8	0	0	0	0	0	0	0	1

圖 3(a). 屬性 k_1 的可分離矩陣 $[D(C_i, C_j)_{k_1}]$

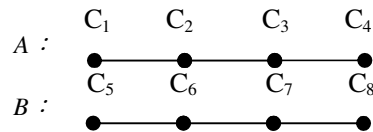


圖 3(b). 圖 3(a) 所對應之可分離圖形

	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈
C ₁	1	1	0	0	0	0	0	0
C ₂	1	1	0	0	0	0	0	0
C ₃	0	0	1	1	0	0	0	0
C ₄	0	0	1	1	0	0	0	0
C ₅	0	0	0	0	1	1	0	0
C ₆	0	0	0	0	1	1	0	0
C ₇	0	0	0	0	0	0	1	1
C ₈	0	0	0	0	0	0	1	1

圖 4(a). 屬性 k_2 的可分離矩陣 $[D(C_i, C_j)_{k_2}]$

	C ₁	C ₂	C ₃	C ₄
C ₁	1	1	0	0
C ₂	1	1	0	0
C ₃	0	0	1	1
C ₄	0	0	1	1

圖 4(b). $[D(C_i, C_j)_{k_2}]$ 在圖 3(b) 群集 A 的子矩陣

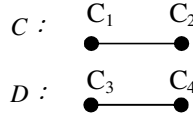


圖 4(c). 使用屬性 k_2 進行群集 A 的分割

可以由其他的屬性繼續進行分割。例如，在圖 3(b) 中的群集 A 可以藉由屬性 k_2 進一步分割，已知屬性 k_2 的可分離矩陣 $[D(C_i, C_j)_{k_2}]$ 如圖 4(a) 所示，針對群集 A 所屬的類別 C_1, C_2, C_3, C_4 收集 $[D(C_i, C_j)_{k_2}]$ 的前四列與前四行，可以得到一個如圖 4(b) 的子矩陣。重複上述使用 $[D(C_i, C_j)_{k_1}]$ 分割 Cr_0 的步驟，群集 A 可使用屬性 k_2 進一步分割成群集 C 與 D，如圖 4(c) 所示。

2.3 分割群集的屬性選擇

透過可分離矩陣便可指出所有類別屬性值的機率密度分佈情形，因此可以使用粗略的分割例如模糊分割來分離不相連的群集，而不是像 C4.5 一樣將每個連續的值視為一個離散的值做處理。通常對於一個已知屬性的數值範圍，且類別數量龐大的分類需要更詳盡的模糊分割。換句話說，給定模糊分割和屬性值的範圍，類別數量較少的群集所獲得的分類準確度會較高。任何一個不精確的群集分割，將會影響後續沿著此群集所在分類樹路徑的群集分割精確度，因此如何挑選一個適當的屬性來分割一個群集便顯得相當重要。分割一個群集所須使用的屬性挑選準則為，哪一個屬性所分割出來的子群集其類別數目平均值乘上變異數為最小，便是我們需要的分割屬性。

例如图 5(a) 與圖 5(b) 為兩個不同屬性 k_1 與 k_2 的可分離矩陣，若使用 k_1 當作分割群集的屬性，則可以得到三個子群集，一個只包含一個類別，另一個包含三個類別，最後一個包含四個類別。然而若使用 k_2 當作分割群集的屬性，則可以得到四個子群集，每個各

	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈
C ₁	1	0	0	0	0	0	0	0
C ₂	0	1	1	0	0	0	0	0
C ₃	0	1	1	1	0	0	0	0
C ₄	0	0	1	1	0	0	0	0
C ₅	0	0	0	0	1	1	0	0
C ₆	0	0	0	0	1	1	1	0
C ₇	0	0	0	0	0	0	1	1
C ₈	0	0	0	0	0	0	1	1

圖 5(a). 可分離矩陣 $[D(C_i, C_j)_{k_1}]$

	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇	C ₈
C ₁	1	1	0	0	0	0	0	0
C ₂	1	1	0	0	0	0	0	0
C ₃	0	0	1	1	0	0	0	0
C ₄	0	0	1	1	0	0	0	0
C ₅	0	0	0	0	1	1	0	0
C ₆	0	0	0	0	1	1	0	0
C ₇	0	0	0	0	0	0	1	1
C ₈	0	0	0	0	0	0	1	1

圖 5(b). 可分離矩陣 $[D(C_i, C_j)_{k_2}]$

包含二個類別。利用前面所提到的挑選準則，可挑選 k_2 作為分割群集 $Cr_0 = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8\}$ 的屬性。將這個準則以數學式表達，令 L_k 與 $n_{k,l}(Cr_j)$ 分別代表使用屬性 k 來分割群集 Cr_j 所得到的子群集的數目，與在第 l 個子群集中的類別數目。於是，挑選分割 Cr_j 的最適當屬性準則為

$$\min_k \alpha_k(Cr_j) = \frac{1}{L_k} \left[1 + \left(\sum_{l=1}^{L_k} (n_{k,l}(Cr_j) - \bar{n}_k(Cr_j))^2 / (L_k - 1) \right) \right] \quad (3)$$

大括弧內的第二項表示在子群集中的類別數目的變異數，其中 $\bar{n}_k(Cr_j) = \sum_{l=1}^{L_k} n_{k,l}(Cr_j) / L_k$ 。這個準則意味著所挑選的最適當屬性能夠分割出更多的子群集，同時在子群集中類別數目的變異數愈小愈好。

接下來將產生群集分割樹的演算法整理如下。

演算法 I：產生群集分割樹

步驟 1：給定原始資料集 Cr_0 以及所有屬性的可分離矩陣，設定尚未分割的群集 $YSC = \{Cr_0\}$ 。

步驟 2：對每一個 YSC 內的群集，依照準則(3) 得到相對應的分割屬性 k 。利用這個屬性去分割群集，並將分割後的子群集放入 YSC 中，同時去除掉已經被分割的群集與無法用任何屬性進行分割的群集。

步驟 3：重複步驟 2 直到 $YSC = \phi$ 。

圖 6 顯示為一群集分割樹，利用圖 3(a)屬性 k_1 與圖 4(a)屬性 k_2 這兩個屬性的可分離矩陣，來分割根群集 $Cr_0 = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8\}$ 。演算法 I 使用了三次遞迴得到最後的群集分割樹，每個群集所使用的分割屬性以及 YSC 的變化過程也標示在圖中。

此外，定義在群集分割樹尾端的葉片群集為終端群集(Leaf Cluster, LC)。每個 LC 可能只包含一個類別或很多個類別，同時無法透過可分離矩陣的其他任何屬性再進行分割。

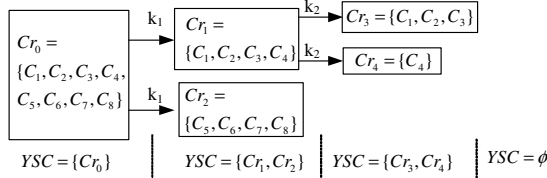


圖 6. 利用演算法 I 得到之群集分割樹

2.4 可分離矩陣為基礎的分群演算法

一旦在群集分割樹中用來分割群集的屬性挑選完畢，便可以針對單一屬性以模糊規則 [15]-[16] 為基礎來建構出每個群集的模糊分割規則。

2.4.1 分群演算法

分群演算法包含了兩個部分：訓練部分，用以產生模糊規則；分類部分，用以分類新的資料到 LC。在訓練部分，依據給定的資料集與已知的 LC 來產生模糊規則，然後建構出整個群集分割樹的完整模糊規則。在分類部分，依據訓練部分產生的模糊規則，透過模糊推論的方法來分類未知類別資料到適當的子群集，這兩部分的詳細做法描述如下。

2.4.2 分群演算法的模糊規則產生程序

在群集分割樹中非 LC 的群集，可以經由演算法 I 配合所挑選出的分割屬性，將一個群集分割為多個子群集。以 Cr_j 表示一個群集，而 k_i 表示演算法 I 所挑選出的分割屬性。假設屬性 k_i 有 g 筆資料以 $x_{k_i}^p$ ， $p=1, \dots, g$ 表示，且完成分割後有 M 個子群集， $SCr_{j1}, \dots, SCr_{jM}$ ，以這 g 筆資料構成分割 Cr_j 的訓練資料。群集分割產生的模糊規則型式如下：

規則 R_i ：若 $x_{k_i}^p$ 落在模糊區間 A_i^K 裡，則 $x_{k_i}^p$ 屬於子群集 SCr_{ji} 具有正確度 $CF = CF_i^K$ ， $i=1, \dots, K$ ，其中 K 代表樣式空間軸的模糊區間數， A_i^K 是第 i 個模糊小區間， SCr_{ji} 為結果子群集，必定為 M 個子群集中的一個，而 CF_i^K 是規則 R_i 的正確度。

使用相對應的分割屬性來分割不同的群集，得到模糊規則的步驟都是一樣的。為了說明方便，將以一個群集如何產生模糊規則的步驟做解說。

以 $m_i(\cdot)$ 表示 (\cdot) 對應於模糊區間 A_i^K 的成員函數。也就是說， $m_i(x_{k_i}^p)$ 可以視為 $x_{k_i}^p$ 對應於模糊區間 A_i^K 所具有的一致性程度。定義

$$\beta_{SCr_{ji}}(R_i) = \sum_{x_p \in SCr_{ji}} m_i(x_{k_i}^p) \quad (4)$$

上式代表對應於模糊區間 A_i^K ，子群集 SCr_{ji} 所具有一致性程度的總和。接下來，分割群集 Cr_j 所產生的模糊規則演算法可以描述如下：

演算法 II：產生單一個群集的模糊規則

步驟 1：給定分割屬性為 k_i ，以及屬於群集 Cr_j 的

的 g 筆訓練資料 $x_{k_i}^p$ ， $p=1, \dots, g$ 。 Cr_j 的

子群集為 SCr_{ji} ， $i=1, \dots, M$ 。設定 $i=1$ 。

步驟 2：利用(4)式計算每一個子群集 SCr_{ji} ，

$i=1, \dots, M$ 的一致性程度的總和

$$\beta_{SCr_{ji}}(R_i)。$$

步驟 3：找出一致性程度的總和為最大值的子群集 SCr_{jx} ，也就是

$$\beta_{SCr_{jx}}(R_i) = \max\{\beta_{SCr_{j1}}(R_i), \dots, \beta_{SCr_{jM}}(R_i)\} \quad (5)$$

步驟4：計算正確度 CF_i

$$CF_i = \left(\beta_{SC_{R_i}}(R_i) - \beta(R_i) \right) / \sum_{j=1}^M \beta_{SC_{R_j}}(R_i) \quad (6)$$

其中 $\beta(R_i) = \sum_{SC_{R_j} \neq SC_{R_i}} \beta_{SC_{R_j}}(R_i) / (M-1)$ 表示

對於模糊區間 A_i^K ，除了 SC_{R_i} 以外的其他子群集一致性程度總和的平均值。

步驟5：若 $i = K$ ，則停止；否則，設定 $i = i+1$ ，回到步驟2。

接下來可以利用訓練資料，以演算法 III 產生群集分割樹的所有模糊規則，程序描述如下。

演算法 III：產生分群演算法的所有模糊規則

步驟1：給定已知類別的訓練資料，計算每個類別 C_i 其屬性為 k 的 μ_i^k 與 σ_i^k ，對每個屬性 k 依據(2)式計算 $[D(C_i, C_j)]_k$ 。

步驟2：套用演算法 I 得到群集分割樹中分割每個群集所需的屬性。

步驟3：套用演算法 II 產生群集分割樹中每個群集的模糊規則。

2.4.3 分群演算法的分類程序

只要產生群集分割樹中的模糊規則，便可以決定新資料在每個群集裡屬於哪個子群集。

以 x'_{k_j} 代表在群集 C_j 中分割屬性為 k_j 的新資料，定義 $\alpha_{SC_{R_j}}$ 為 x'_{k_j} 在子群集 SC_{R_j} 中的加權一致性程度，其值設定為對所有模糊規則 R_i ， x'_{k_j} 對 A_i^K 的一致性程度總和 $m(x'_{k_j})$ 乘以 R_i 正確度 CF_i^K ，亦即 $\alpha_{SC_{R_j}} = \sum_{R_i} m(x'_{k_j}) \cdot CF_i^K$ 。新資料的分類程序可以說明如下。

分類程序：對應於 $SC_{R_{j_l}}$ 這個子群集， x'_{k_j} 具有最大的加權一致性程度，則 $SC_{R_{j_l}}$ 就是 x'_{k_j} 分類的結果群集，也就是說，
 $SC_{R_{j_l}} = \arg(\max\{\alpha_{SC_{R_{j_1}}}, \dots, \alpha_{SC_{R_{j_M}}}\})$ 。

如此，分類新的資料到 LC 的分類演算法可以整理如下。

演算法 IV：分群演算法的所有分類程序

步驟1：給定新資料 $x' = (x'_1, \dots, x'_n)$ ，其中 n 代

表屬性數目，設定目前群集 $ACr = Cr_0$ 。

步驟2：使用 x_k ，下標 k 表示用以分割 ACr 的屬性，利用上述的分類程序來分類 x' 到子群集 $CACr$ 。若 $CACr$ 並不是 TC，設定 $ACr = CACr$ ，然後重複此步驟；否則便停止。

2.5 終端群集的分類程序

在使用可分離矩陣為基礎的分群演算法後，每個產生的 LC 也許只會包含一個類別的資料或仍舊包含了很多類別的資料。既然在 LC 裡的資料數量與類別數目會比 Cr_0 少很多，直接套用 C4.5 來分類 LC 將會使計算更為容易。因此，分群演算法可以幫助降低 C4.5 的計算複雜度。

C4.5 是一個發展良好的分類工具，這個分類工具詳細的資料可以很容易找到[10]。簡單來說，C4.5 分類規則建構在三個主軸上：(i) 分裂的規則，(ii) 分裂準則的好壞，(iii) 選擇一個最佳或最終樹作分析的準則。在第(i)項，C4.5 主要有三個分裂規則，我們所採用的是 *Information gain ratio* 判斷法則。這個法則建立樹的起始程序，透過一個很簡單的問題，如 $k_i \leq b$? 將 LC 分裂成二個節點，這裡的 k_i 是一個屬性而 b 為一個數值。在第(ii)項，C4.5 會對於每個屬性所有可能的分裂點尋找出最好的分裂點，也就是最能夠減少 *Information gain ratio* 雜質測量[10]的那個分裂點。C4.5 會在後續的階段遞迴地套用這個分裂規則來分割非終端節點。為了降低初始建構出的樹具有之複雜度，如第(iii)項所提到的，C4.5 會使用一個修剪程序來找出最佳的樹。但是一個樹的複雜度是由終端節點的數目來衡量，所以 C4.5 的計算複雜度主要浪費在第(ii)項，亦即徹底的搜尋最好的分裂點。這個計算複雜度隨著資料量的增加成指數型增長，這也是為何我們的分群演算法可以幫助降低計算複雜度的地方。

3. 錯誤偵測與隔離策略

離子植入機的正常運作對於生產量是一個極為重要關鍵的，主要是因為離子植入機一旦發生故障，晶圓會造成損壞且無法重作。因此，即時的錯誤偵測可防止更多晶圓的損壞且錯誤的隔離可以縮小離子植入機的停機時間，因此對於離子植入機來說是一個重要的開

發，且也會影響到半導體製造流程的製程良率和生產量。

3.1 錯誤警報的產生

圖 7 為我們所提出的錯誤偵測與隔離策略的方法架構圖，首先我們會對離子植入機中正在運作的晶圓進行訊號量測，接著進行分類，判斷是否和所下的配方符合，如果符合則繼續檢查下一個晶圓，但如果不符合，首先判斷所下的配方是否正確，如果不正確則需通知作業人員，反之則進行判斷是否達到錯誤警報的標準，如果未達到標準則繼續檢查下一個晶圓，反之則產生錯誤警報並且進行錯誤隔離。

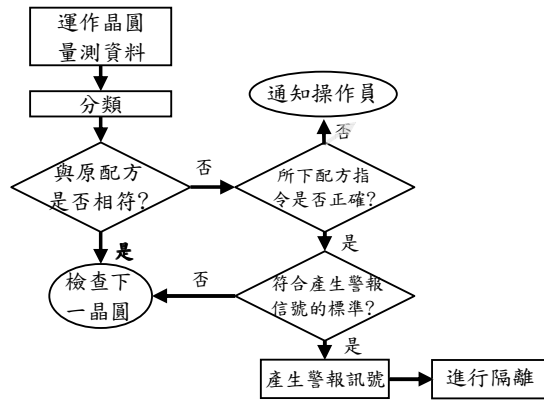


圖 7. 錯誤偵測與隔離策略方法架構圖

一般來說，離子植入機一旦發生故障，會造成連續性的晶圓損壞，由於這個特性我們可以用來判斷離子植入機是否真的發生故障，或只是因為機械及工廠環境所造成的電氣訊號突波，主要是因為電氣訊號突波通常只會發生在一些少數的晶圓上。因此我們可以制定訊號產生的標準，詳細說明如後。定義 q_i 為第 i 類配分配錯誤的機率， ε 為晶圓連續發生錯誤的最小可能性，制定為 10^{-15} ，當第 $l-1$ 個晶圓分類正確，而第 l 個， $l+1$ 個， \dots ， $k+n$ 個晶圓分類錯誤時的機率，如(7)式所示。定義 n 為連續發生晶圓錯誤的數量， n_1 代表會受電氣訊號突波影響的連續晶圓最大數量，制定為4，如(8)式所示。同時滿足下列兩個條件，即可判斷達成錯誤警報產生的標準，則警報信號將在第 $k+n$ 個晶圓時產生：

$$q_i(l) \times q_{i+1}(l+1) \times \dots \times q_{i+n}(l+n) \leq \varepsilon \quad (7)$$

且

$$n \geq n_1 \quad (8)$$

3.2 隔離策略

為了排除離子植入機的故障，我們必須提出隔離策略，以進行機台的修復。主要目標是針對離子植入機的四個子系統進行隔離，分別為離子源、萃取器、質量分析器及加速器，如圖 8 所示。我們分別以 10 個屬性對這四個子系統進行量測，所對應的製程量測參數如表 1 所示：

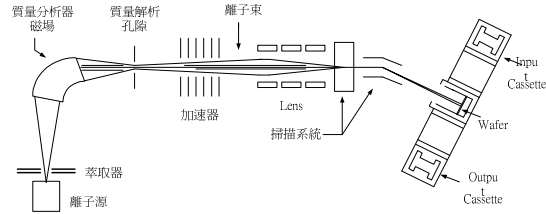


圖 8. 離子植入機之設備系統

表 1. 製程量測參數

	屬性	單位	量測位置
k_1	燈絲電壓	Volts	離子源
k_2	燈絲電流	Amps	離子源
k_3	萃取離子電壓	KV	萃取器
k_4	萃取離子電流	mA	萃取器
k_5	離子加速/減速電壓	KV	萃取器
k_6	磁場強度	KGauss	質量分析器
k_7	高電壓電源供應器電流	μA	質量分析器
k_8	離子束電流	mA	質量分析器
k_9	離子束壓力	Torr/e6	加速器
k_{10}	反應腔室壓力	Torr/e6	加速器

在第 2 節提到，終端群集(LC)會直接套用 C4.5 進行分類，圖 9 為針對 LC₇ 利用 C4.5 進行分類的結果，圖中粗體實線為分類正確的路徑，粗體虛線為分類錯誤的路徑，指的是將第 5 類配方錯誤分類到第 4 類。在第 5 類配方其分配錯誤的機率為 0.22%，由表 2 可得知，分配錯誤的筆數分別為 2 筆和 4 筆，直接套用(7)式即可計算出小於門檻值 ε 的結果，而連續發生 6 片晶圓損壞也大於門檻值 n_1 ，同時滿足上列兩個條件，因此達成錯誤警報產生的標準。在圖 9 可得知兩個錯誤分類的路徑都因為屬性 k_8 所造成，對照表 1 可得知屬性 k_8 為量測質量分析器的屬性參數，因此可以判定為質量分析器損壞，需進行故障隔離及修復。

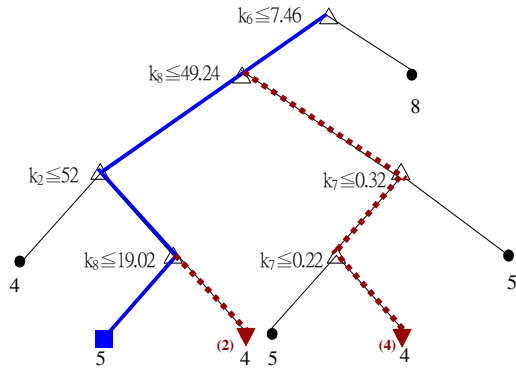


圖 9. 利用 C4.5 對 LC₇ 建立的決策樹

4. 測試結果與比較

在半導體製造過程中離子植入機為一瓶頸機台，主要是因為其價格非常昂貴，並且離子植入機的正常運作對於生產量是一個極為重要的關鍵[17]。我們以半導體製程中的離子植入機當作測試機器，而離子植入晶圓的各種配方當作測試產品。一般來講，相當多的屬性可以從離子植入機中量測得到，然而並不是每個屬性在分類上都是有用的。從工作晶圓透過半導體通訊標準網路 SECS-II，所得到的測量資料是由 10 個主要屬性， k_1 、 k_2 、 \dots 和 k_{10} 所組成，分別代表燈絲電壓、燈絲電流、萃取離子電壓、萃取離子電流、離子加速/減速電壓、磁場強度、離子束電流、高壓電源供應器電流、離子束壓力和反應腔室壓力。

一個由半導體晶圓廠提供具有 36 種不同的產品的配方，每個配方含有數百到一萬個不等的晶圓片資料集，用來測試所提出的分類器 FCDT 的分類準確性，其中配方的總類會因為訓練資料的增多而配方的類別相對的增加，在分類的時候該所屬的產品因該會分類到所屬的類別，如果分類錯誤則可往回追溯判斷是哪個屬性所造成。在所有蒐集到的資料集裡，隨機的切分成十等份。以其中九等份當作訓練資料，另外一等份作為測試資料。在演算法 II 中 $m_i(\cdot)$ 採用三角成員函數。將訓練資料套用演算法 III，所得到的群集分割樹與分割屬性如圖 10 所示，對於每一個分割群集所對應的模糊規則也一併得到。圖 10 中共有八個 LC，除了 LC₃、LC₅ 為單獨一個配方(類別 18、1)外，其餘六個 LC 都包含超過一個以上的配方。群集分割樹中每一個群集所使用的分割屬性標示在箭頭上方。每一個群集所包含的配方標示在括弧內。然後利用 C4.5 對六個 LC 進行分類，

建構出每一個 LC 的分類樹以及分割規則。最後利用測試資料來測試建構好的 FCDT 準確度。將十等份的資料循環排列重新組合成訓練資料與測試資料，重複上述步驟十次即獲得 10 重交疊驗證結果。表 2 為所得到的 10 重交疊驗證的分類錯誤率，同時列出以軟體 See5[18] 與 CART[11]所得到的分類錯誤率。

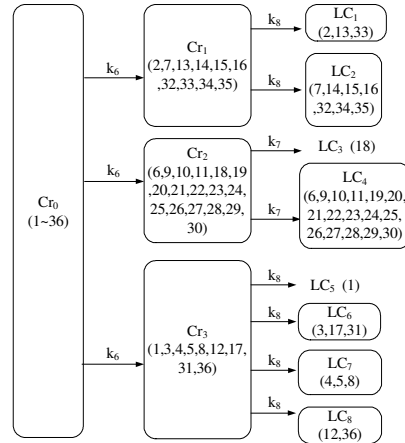


圖 10. 36 個配方的群集分割樹

表 2. 36 個配方的分類錯誤率

Class	Classification error rate (%)			Class	Classification error rate (%)		
	FCDT	CART	See5		FCDT	CART	See5
1	0	0	0	19	0	0.75	0.67
2	0	0	0	20	0	0.67	0
3	0	0.1	0	21	0	0	0.5
4	0.1	0.1	0.2	22	0	0	0
5	0.22	0.22	0.22	23	0	0	0
6	0	0	0.2	24	0	0.5	0
7	0	0	0	25	0	0	0
8	0.33	0.4	0	26	0	0	0
9	0.33	0.2	0.33	27	0	0	0
10	0	0	0	28	0	0	0
11	0	0	0	29	0	0	0
12	0	1.0	0	30	0	0	0
13	0	0	0	31	0	0	0
14	0.13	0	0.13	32	0	1.67	0
15	0.1	0	0.3	33	0	0	0
16	0	0.67	0	34	0	0	0.18
17	0	0.13	0	35	0	0.18	0
18	0	0.2	0	36	0.29	0	0.029

從表 2 可看出，FCDT 的最大分類錯誤率為 0.33%，另外所得到的 10 重交疊驗證的分類錯誤率總和，以及所花費的訓練時間如表 3 所示。從表中可以發現 FCDT 的平均分類錯誤率總和為 1.5%，而 See5 和 CART 的平均分類錯誤率總和分別為 6.79% 與 3.02%。此外，FCDT 所需的訓練時間比 See5 與 CART 還要少，標示在表 3 的第二列中。這顯示 FCDT 不但在分類錯誤率總和上比 See5 與 CART 還低，且花費更少的訓練時間。

表 3. 訓練時間與 10 重交疊驗證分類錯誤率

	FCDT	CART	See5
訓練時間 (sec)	10.965	34.375	20.512
10 重交疊驗證分類錯誤率總和(%)	1.5	6.79	3.02

5. 結論

在本文中提出的模糊群集決策樹(FCDT)可用於解決較多類別和連續性質屬性的資料分類問題上。這個技術可以有效地降低終端群集分類的計算複雜度，因此非常適合做為 C4.5 的預先處理步驟。並且我們也將此技術實際套用在離子植入機的錯誤偵測上，測試結果顯示所提出的 FCDT 比 CART 和 See5 具有更高的分類正確性，同時節省了更多的訓練時間。

致謝

本論文承蒙行政院國家科學委員會補助研究經費，計畫編號 NSC 98-2221-E-324-039。特此致謝。

參考文獻

- [1] T. Denoeux, "A neural network classifier based on Dempster-Shafer theory," *IEEE Transactions on Systems, Man and Cybernetics*, Part A, vol. 30, no. 2, pp. 131-150, March 2000.
- [2] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man and Cybernetics*, Part C, vol. 30, no. 4, pp. 451-462, Nov. 2000.
- [3] J. Ediriwickrema, and S. Khorram, "Hierarchical maximum-likelihood classification for improved accuracies," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 4, pp. 810-816, July 1997.
- [4] L. Bruzzone, and D. F. Prieto, "Unsupervised retraining of a maximum likelihood classifier for the analysis of *multitemporal* remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 456-460, Feb. 2001.
- [5] B. Chandra, and P. P. Varghese, "Fuzzifying Gini Index based decision trees," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8549-8559, May 2009.
- [6] X. Chang, and J. H. Lilly, "Evolutionary design of a fuzzy classifier from data," *IEEE Transactions on Systems, Man and Cybernetics*, Part B, vol. 34, no. 2, pp. 1031-1044, April 2004.
- [7] D. M. Hawkins, and G. V. Kass, "Automatic Interaction Detection," *Topics in Applied Multivariate Analysis*, Cambridge Univ Press: Cambridge, pp. 267-302, April 1982.
- [8] J. R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo, Calif.: Morgan Kaufmann, Jan. 1993.
- [9] J. R. Quinlan, "Introduction of decision tree," *Mach. Learn.*, vol. 1, pp. 81-106, 1986.
- [10] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *J. Artif. Intell. Res.*, vol. 4, pp. 77-90, 1996.
- [11] L. Breiman, J. H. Friedman, J. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, London : Chapman & Hall, Jan. 1984.
- [12] R. L. Lawrence, and A. Wright, "Rule-Based Classification Systems Using Classification and Regression Tree (C4.5) Analysis," *Photogrammetric Engineering and Remote Sensing*, vol. 67, no. 10, pp. 1137-1142, Oct. 2001.
- [13] B. Chandra, and P. P. Varghese, "Fuzzy SLIQ Decision Tree Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, vol. 38, no. 5, pp. 1294-1301, Oct. 2008.
- [14] D. C. Montgomery, *Introduction to statistical quality control*. 6th edition, New York: John Wiley and Sons, May 2008.
- [15] O. Cordon, M. Jose del Jesus, and F. Herrera, "A proposal on reasoning methods in fuzzy rule-based classification systems," *International Journal of Approximate Reasoning*, Vol. 20, No.1, pp. 21-45, Jan. 1999.
- [16] H. Ishibuchi, and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Transaction on Fuzzy Systems*, vol. 9, no. 4, pp. 506-515, Aug. 2001.
- [17] S. Y. Lin, and S. C. Horng, "A Classification Based Fault Detection and Isolation Scheme for the Ion Implanter," *IEEE Transactions on Semiconductor Manufacturing*, vol. 19, no. 4, pp. 411-424, Nov. 2006.
- [18] J. R. Quinlan, (2008) Data mining tools See5 and C5.0, version 2.06.[Online]. Available:<http://www.rulequest.com/see5-info.html>