



國立中山大學電機工程學系

碩士論文

利用基因演算法產生模糊分類系統

Generation of Fuzzy Classification Systems

using Genetic Algorithms

研究生：李政聰 撰

指導教授：李錫智 博士

中華民國九十四年一月

致謝

在碩士班生涯的兩年半中，非常感謝恩師 李錫智 教授於學業上及生活上對我的提攜與照顧。除學識淵博外，其和善待人及學者風範，更使我敬佩有加。特別是當我家務繁忙時，除表達極度的關切之意，還原諒了我的任性。最後我想以四個字，誠摯地表達我對老師的賞識之恩，正如同中山電機的第一屆學長們所言－師恩永懷。

亦感謝口試委員 吳志宏 教授及 黃宗傳 教授，除在百忙之中撥冗指教之外，精闢而獨到的見解也使得本論文能更趨完備。

另外，特別感謝高雄大學電機系 吳志宏 教授及學長 蔡賢亮 博士對我在程式撰寫能力上的啟發。尤其吳老師時常不辭辛勞又不計酬勞的從高雄大學遠道而來，實令我萬分感激。

感謝學長 歐陽振森 博士及博士班 李婉瑞 學姊，在研究上及生活上的扶持。以及碩士班同門同屆的好友 劉茂宗 同學，我們時時互相鼓勵，也互相砥礪，在我心情低落時引導我，在我疏忽怠惰時督促我，很幸運能擁有這麼一個一輩子的好朋友。

也非常感謝女友 王于禎 小姐這四年來，在我學業繁忙而不克返家時，替我打點了不少家中的瑣事。

最後，謹將本論文獻給我勞苦功高的雙親－李豐地 先生及 林瑞文 女士，雙親二十餘年來對我的養育及栽培之恩，此生已無以回報。而先慈未及見我完成學位、成家立業，忝為人子，實是畢生最大的遺憾。

李政聰 于 西灣

2006/2/19

摘要

為了解決樣本分類問題，本論文提出了一種改良式 fuzzy GBML (genetic-based machine learning) 演算法來設計一個 FRBCS (fuzzy rule-based classification system)。

過去的 hybrid fuzzy GBML 演算法使用了計算較為耗時的 SS 模糊模型，此外，為了加快匹茲堡式演算法的收斂速度，還結合了計算繁複的密西根式演算法，因而造成 fuzzy GBML 演算法的計算時間冗長。而我們所提出的改良式 fuzzy GBML 演算法使用了計算較為快速的 MW 模糊模型，且以一個計算簡單的 heuristic procedure 來取代密西根式演算法的功能。實驗結果亦顯示，我們的改良式 fuzzy GBML 演算法不但較 hybrid fuzzy GBML 演算法計算時間更短、收斂速度更快，且還提升了些微的分類正確率。

關鍵字：樣本分類，模糊分類系統，fuzzy GBML，模糊模型，基因演算法。

Abstract

In this thesis, we propose an improved fuzzy GBML (genetic-based machine learning) algorithm to construct a FRBCS (fuzzy rule-based classification system) for pattern classification problem.

Existing hybrid fuzzy GBML algorithm is consuming more computational time since it used the SS fuzzy model and combined with the Michigan-style algorithm for increasing the convergent rate of the Pittsburgh-style algorithm. By contrast, our improved fuzzy GBML algorithm is consuming less computational time since it used the MW fuzzy model and instead of the role of the Michigan-style algorithm by a heuristic procedure. Experimental results show that improved fuzzy GBML algorithm possesses the shorter computational time, the faster convergent rate, and the slightly better classification rate.

Keywords: pattern classification, FRBCS, fuzzy GBML, fuzzy model, genetic algorithm.

目錄

摘要.....	i
Abstract.....	ii
第一章 簡介.....	1
1.1 樣本分類問題.....	1
1.2 學習能力與推廣能力.....	1
1.3 編排方式.....	3
第二章 Fuzzy GBML 演算法.....	4
2.1 Fuzzy Rule-Based System.....	4
2.2 Fuzzy GBML 演算法.....	7
2.3 匹茲堡方法與密西根方法.....	8
第三章 Hybrid Fuzzy GBML 演算法.....	12
3.1 模糊分割的方式與 Fuzzy Rule 的形式.....	13
3.2 Fuzzy Rules 的前提部份與結論部份.....	15
3.3 Single Winner Rule 的模糊推論方法.....	18
3.4 Heuristic Procedures.....	20
3.4.1 利用訓練樣本產生初始的 Fuzzy Rules.....	20
3.4.2 利用被錯誤的分類或是拒絕分類的訓練樣本產生新的 Fuzzy Rules.....	22
3.5 基因運算方式.....	23
3.6 Hybrid Fuzzy GBML 演算法.....	25

第四章	我們的方法.....	27
4.1	動機.....	27
4.2	Fuzzy Rule 的形式及產生結論部份的方式.....	28
4.3	Weighted Voting 的模糊推論方法.....	30
4.4	過程及計算時間的比較.....	31
4.4.1	Fuzzy Rule 產生結論部份的過程.....	31
4.4.2	模糊推論過程.....	32
4.4.3	計算時間的比較.....	34
4.5	改變模糊模型.....	34
4.5.I	信用指派的問題.....	35
4.5.II	密西根式演算法存在的必要性.....	36
4.6	改良式 Fuzzy GBML 演算法.....	38
第五章	實驗結果與分析.....	42
5.1	產生結論部份及模糊推論的時間比較.....	44
5.2	改變模糊模型之後的因應方法.....	45
5.3	分類正確率與計算時間的比較.....	47
5.4	收斂速度的比較.....	48
第六章	結論.....	53
	參考文獻.....	54

圖表目錄

圖 1.1	平面樣本空間及其決策分界線.....	2
圖 2.1	五個語文變數及其 Fuzzy Set 的歸屬函數.....	5
圖 2.2	均勻模糊分割的樣本空間.....	6
圖 2.3	匹茲堡式演算法的編碼方式.....	9
圖 2.4	密西根式演算法的編碼方式.....	9
圖 2.5	匹茲堡式演算法的編碼實例.....	10
圖 2.6	密西根式演算法的編碼實例.....	10
圖 3.1	四種均勻模糊分割.....	13
圖 3.2	<i>don't care</i> 的歸屬函數.....	14
圖 3.3	五條 Fuzzy Rules 所連成的字串.....	16
圖 3.4	匹茲堡式演算法的族群.....	23
圖 3.5	密西根式演算法的族群.....	24
圖 4.1	兩種模糊模型分別產生的決策分界線.....	27
圖 4.2	Weighted Voting 的模糊推論方法.....	36
圖 4.3	[17]中的收斂速度比較.....	37
圖 5.1	Breast W Data 的收斂速度比較.....	49
圖 5.2	Diabetes Data 的收斂速度比較.....	50

圖 5.3	Glass Data 的收斂速度比較.....	50
圖 5.4	Heart C Data 的收斂速度比較.....	51
圖 5.5	Iris Data 的收斂速度比較.....	51
圖 5.6	Sonar Data 的收斂速度比較.....	52
圖 5.7	Wine Data 的收斂速度比較.....	52
表 5.1	實驗所使用的 Data Sets.....	42
表 5.2	實驗所設定的參數值.....	43
表 5.3	Data Sets 及其 $P_{don't\ care}$ 的預設值.....	43
表 5.4	兩種模糊模型的時間比較(單位:秒).....	44
表 5.5	[17] 移除密西根式演算法之後的實驗結果.....	45
表 5.6	改變模糊模型之後的因應方法之實驗結果.....	46
表 5.7	分類正確率及分類錯誤率的比較.....	47
表 5.8	計算時間的比較.....	48

第一章 簡介

1.1 樣本分類問題

樣本分類問題 (pattern classification problem) 的目標是設計一個分類器 (classifier)，當給定一個類別 (class) 未知的樣本 (pattern) 時，這個分類器可以告訴我們這個樣本是屬於什麼類別。例如，現在我們設計了一個“人類性別分類器”，只要輸入一個人的身高、體重和興趣，這個分類器就會告訴我們這個人是屬於男人還是女人。而這個問題的重點在於，我們要如何設計這個分類器，才能使得它的分類正確率能夠盡可能的高。一般來說，一個分類器的分類正確率可經由兩個部份來評估，一個是它的學習能力 (learning ability)，另一個為其推廣能力 (generalization ability)。我們將在下一節中說明。

1.2 學習能力與推廣能力

首先，我們先給定一些已經正確分類，且數目固定不變的樣本，我們稱這些樣本為訓練樣本 (training pattern)，我們希望我們所設計

的分類器，能夠透過給定的訓練樣本，在整個樣本空間 (pattern space) 上，劃出所有的決策分界線 (decision boundary)，將樣本空間中類別不同的訓練樣本全部劃分開來，或者錯誤率為最小。舉例來說，圖 1.1 的左圖是一個包含了 20 個訓練樣本的平面樣本空間，這 20 個訓練樣本可分成黑色跟白色兩類，我們希望我們所設計的分類器能夠在這個平面樣本空間上劃出一條決策分界線，將類別不同的訓練樣本全部劃分開來 (如圖 1.1 右圖)。

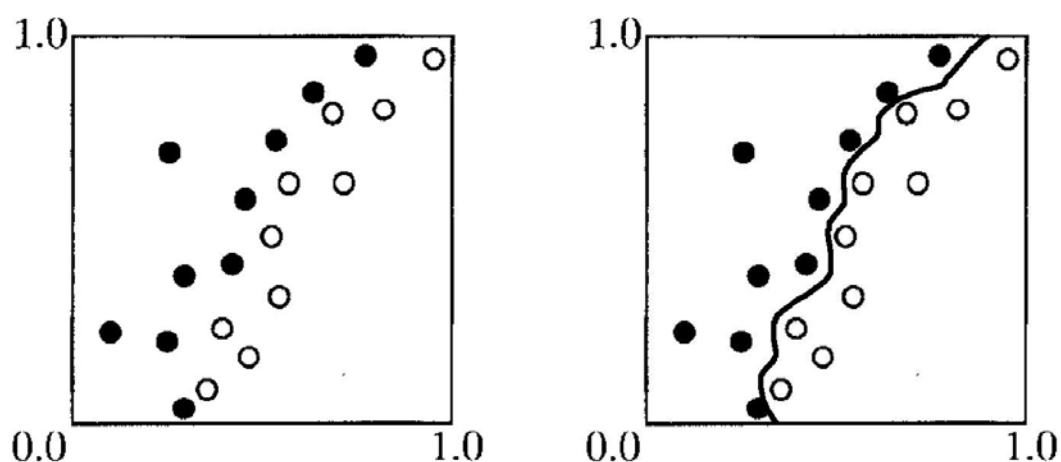


圖 1.1 平面樣本空間及其決策分界線

給定分類器不同的訓練樣本就能產生不同的決策分界線來劃分給定的訓練樣本，這個過程即稱為分類器的學習過程 (learning procedure)，而訓練樣本被正確劃分的正確率即稱為此分類器的學習能力。

分類器經過學習之後，找到了一組規則來分類給定的訓練樣本，

這時候我們再給定一些在學習過程中，未曾見過的樣本（即，和訓練樣本不完全相同的樣本）給分類器，看看它是否也能正確的分類這些未曾見過的樣本，若分類正確率越高，則代表這個分類器的推廣能力越高，前述的過程即稱為分類器的測試過程（testing procedure）。

值得一提的是，當一個分類器的學習能力太高的時候，很容易因為 overfitting 的問題而導致分類器的推廣能力下降，所以在設計一個分類器的過程中，我們應盡量避免這類情形的發生。

1.3 編排方式

本論文的編排方式概述如下。在第二章中，我們將簡介 fuzzy genetic-based machine learning (fuzzy GBML) 演算法的研究背景，即為本論文設計分類器的方法。而第三章將說明 Ishibuchi (石淵久生) 等人所提出的 fuzzy GBML 演算法[17]。第四章則敘述我們所提出的 fuzzy GBML 演算法，並對我們的架構及優點作詳細的說明。第五章是實驗模擬的結果與分析。最後在第六章對本篇論文作一個總結。

第二章 Fuzzy GBML 演算法

2.1 Fuzzy Rule-Based System

自 1965 年柏克萊大學的 Zadeh 教授提出 Fuzzy Set Theory[1]至今，Fuzzy Rule-Based Systems (FRBSs) 已成功的解決了許多控制工程上的問題[2]。簡單來說，一個 FRBS 即是一個由許多 rules 所組成的推論 (reasoning) 系統，當我們給定這個推論系統一個輸入值之後，它會根據系統內所包含的 rules 以及預設的推論方法，以模糊邏輯 (fuzzy logic) 為工具，推論出這個輸入值所對應的輸出值。而 FRBSs 最大的好處在於，每條 fuzzy rule 的前提部份 (antecedent part) 及結論部份 (consequent part)，都以語文變數 (linguistic variable) 的形式來表示，如“小”或“大”，而每個語文變數則使用一個 fuzzy set 來代表它，最後再以一個歸屬函數 (membership function) 定義了這個 fuzzy set 在論域 (universe of discourse) 上的分佈 (distribution) 情形，經由語文變數字面上的意義，人類可以很輕易的了解這些 fuzzy rules 的涵義。例如，“IF 頭髮稀疏 THEN 是禿頭”，這條 fuzzy rule

的前提部份和結論部份各使用了一個語文變數（即，“頭髮稀疏”和“禿頭”），雖然我們不知道代表這兩個語文變數的 fuzzy set 如何定義其歸屬函數於論域上的分佈情形（即，頭髮要少到什麼程度才算是稀疏，以及頭髮要稀疏到什麼程度才算是禿頭），但透過語文變數字面上的意義，我們卻可以很輕易的了解這條 fuzzy rule 所代表的涵義。一個典型的例子如圖 2.1 所示。

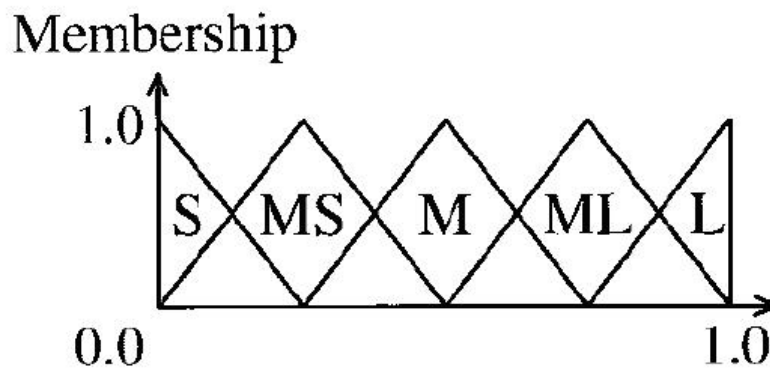


圖 2.1 五個語文變數及其 Fuzzy Set 的歸屬函數

圖 2.1 中包含了 5 個語文變數，即 S (Small)、MS (Medium Small)、M (Medium)、ML (Medium Large) 及 L (Large)，每個語文變數均以一個 fuzzy set 來代表它，最後再用 5 個三角形的歸屬函數來定義這 5 個 fuzzy sets 在論域上的分佈情形。

在大多數的模糊控制中，fuzzy rules 均來自於人類專家的經驗，但是經由人類專家而獲取 fuzzy rules 是一件非常不方便且不容易的事。所以在過去的十年間，許多文獻提出了直接從數值資料(numerical

data) 自動產生 fuzzy rules 的方法[3]，或者使用自我學習法來調整 fuzzy rules 中 fuzzy set(s)的歸屬函數[4]，希望藉由這些方法來減少模糊控制工程對人類專家的依賴。使用這些方法設計 FRBSs 時，若我們希望最後產生的每條 fuzzy rule 都可以很直觀的被人類所理解，那麼在一開始的時候，使用均勻的模糊分割（fuzzy partition）來作為 fuzzy sets 的歸屬函數是較為適當的。如圖 2.2 所示。

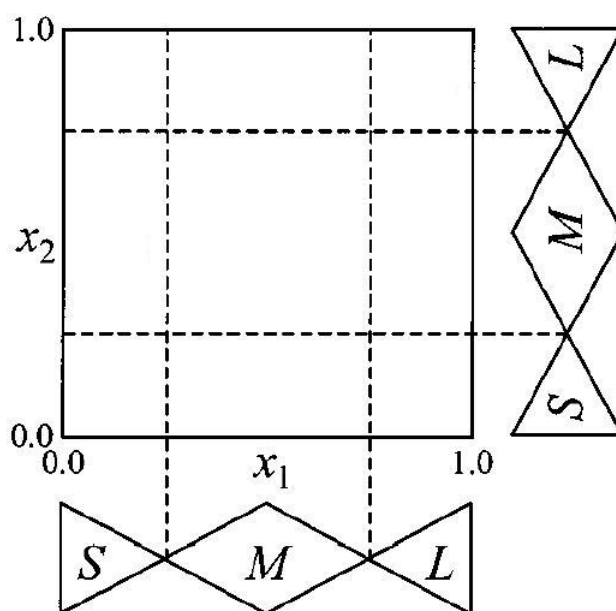


圖 2.2 均勻模糊分割的樣本空間

在一個平面的樣本空間上，存在著 3 個語文變數，S (Small)、M (Medium) 及 L (Large)，我們將每個維度均勻的分割成 3 個 fuzzy sets，用來表示這 3 個語文變數的意義。當然，若擁有人類專家對語文變數的經驗知識時，我們不會一定要將每個維度作均勻的分割。

應用 Fuzzy Set Theory 的模糊分類方法，可大略的分成四類[5]。

其中，基於 fuzzy relations 的方法即是使用 fuzzy rules 來說明輸入和輸出之間的關係，如 FRBSs。而在基於 fuzzy pattern matching 的方法中，若有 2 個訓練樣本其屬性值 (attribute value) 存在著某種程度以上的符合，我們便將這 2 個訓練樣本合併。基於 Fuzzy clustering 的方法則應用 clustering 的技術在樣本空間上劃出決策分界線，如 fuzzy c-means[6]。剩餘不屬於這三類的方法便歸為同一類。而當我們應用一個 FRBS 來解決分類問題時，我們可將此 FRBS 稱為一個 FRBCS (fuzzy rule-based classification system)，本論文所提出的 FRBCS 設計方法屬於 Fuzzy Rule-Based Method 的一種，即為基於 fuzzy relations 的方法。

2.2 Fuzzy GBML 演算法

基因演算法 (Genetic Algorithm, GA) [7][8]於 1975 年由密西根大學的 Holland 教授所提出。簡單來說，基因演算法即是一種用來解決組合最佳化 (combinatorial optimization) 問題的演算法。為了設計一個不需要人類專家提供經驗的 FRBS，許多文獻均提出將基因演算法應用於 FRBSs，我們統稱這類的方法為 fuzzy genetic-based machine learning (fuzzy GBML) 演算法。

例如，藉由基因演算法來產生 FRBS 的 fuzzy rules 且調整出最適

合每條 fuzzy rule 結論部份的 fuzzy set(s)[9]；或者用基因演算法同時調整樣本空間中模糊分割（即，fuzzy set）的個數以及每個 fuzzy set 的歸屬函數[10]；以及將模糊分類系統中的每一條 fuzzy rule 都編碼成族群（population）中的一條染色體（chromosome），然後藉由模糊分類系統來決定最適合每條 fuzzy rule 前提部份及結論部份的 fuzzy sets[11]等等。

雖然之前已經列舉了許多 fuzzy GBML 演算法，但其中絕大部份都是用來解決控制工程上的問題或是 function approximation 的問題，只有少許幾種可應用於樣本分類問題。然而，針對樣本分類問題而言，同樣也有許多 fuzzy GBML 演算法被提出，像是用基因演算法來最佳化模糊決策樹（decision tree）[12]；或者當屬性值為離散屬性時，用以產生模糊分類系統中的 fuzzy rules[13]；以及在多維度的樣本空間中，先用一個 hyperbox 來代表一條 fuzzy rule，然後再藉由 hyperboxes 來產生及調整 fuzzy rules[14]等等。

2.3 匹茲堡方法與密西根方法

一般來說，fuzzy GBML 演算法可依據應用基因演算法時的編碼方式，大致的分成匹茲堡方法（Pittsburgh approach）[15]及密西根方法（Michigan approach）[16]兩大類，我們將使用匹茲堡方法的 fuzzy

GBML 演算法簡稱為匹茲堡式 (Pittsburgh-style) 演算法，而使用密西根方法的 fuzzy GBML 演算法則簡稱為密西根式 (Michigan-style) 演算法。在匹茲堡式演算法中，我們將一個 rule set 編碼成族群中的一條染色體，所以整個族群是由許多的 rule set 所組成。如圖 2.3 所示。

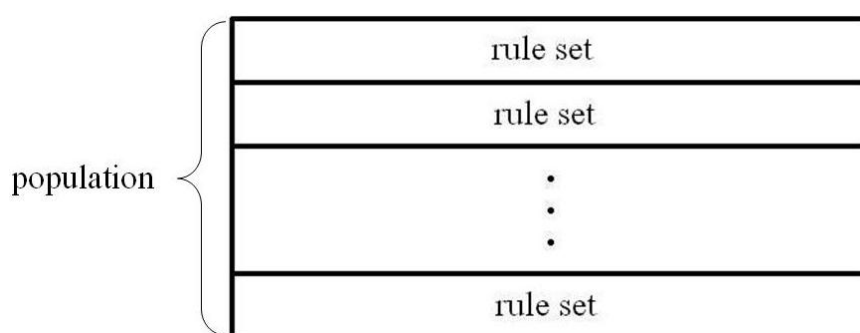


圖 2.3 匹茲堡式演算法的編碼方式

而密西根式演算法則將一條 fuzzy rule 編碼成族群中的一條染色體，再用整個族群來代表一個 rule set。如圖 2.4 所示。

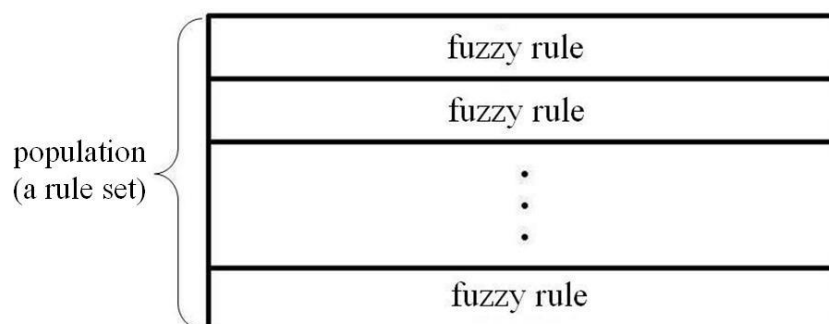


圖 2.4 密西根式演算法的編碼方式

舉例來說，若有一個由下列 3 條 fuzzy rules 所組成的 rule set。

IF X_1 is A_1 and X_2 is A_1 THEN Y is A ,

IF X_1 is A_1 and X_2 is A_2 THEN Y is B ,

IF X_1 is A_2 and X_2 is A_1 THEN Y is C ,

則在匹茲堡式演算法中，我們將此 rule set 編碼成族群中的一條染色體（如圖 2.5 所示）。

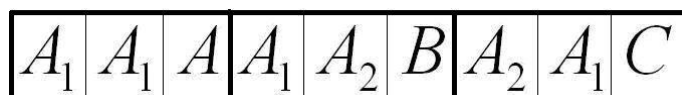


圖 2.5 匹茲堡式演算法的編碼實例

若在密西根式演算法中，我們則將一條 fuzzy rule 編碼成族群中的一條染色體，再以整個族群來表示這個 rule set（如圖 2.6 所示）。

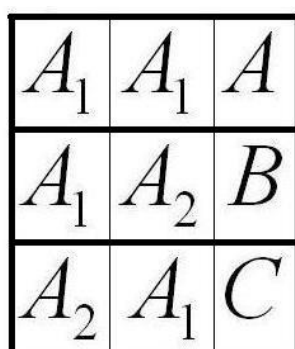


圖 2.6 密西根式演算法的編碼實例

雖然匹茲堡式演算法及密西根式演算法均以被此染色體正確分類的訓練樣本個數來作為此染色體的 fitness value，但在匹茲堡式演

算法中，一條染色體代表的是一個 rule set，而在密西根式演算法中，一條染色體所代表的卻是一條 fuzzy rule，所以匹茲堡式演算法可以在龐大的樣本空間中，找出一個好的 rule set，而密西根式演算法則能找到許多好的 fuzzy rules。另一方面，雖然匹茲堡式演算法最後可以找到一個好的 rule set，但因每一個 generation 所檢查 (examine) 的 fuzzy rules 數目較多，故收斂速度較慢，因而需較長的計算時間，而密西根式演算法雖然收斂速度較快，所需的計算時間較短，但最後所找到的族群卻不一定能構成一個好的 rule set。

為了上述的問題，Ishibuchi 等人在 2005 年提出了一個結合匹茲堡方法和密西根方法的 hybrid fuzzy GBML 演算法[17]，希望能同時結合這兩種方法的優點，快速的為樣本分類問題找出一個具高分類正確率的 FRBCS。我們將在下一章中說明這個方法。

第三章 Hybrid Fuzzy GBML 演算法

為了加快匹茲堡式演算法的收斂速度，提升密西根式演算法的分類正確率，Ishibuchi 等人在 2005 年提出了一個結合匹茲堡方法和密西根方法的 hybrid fuzzy GBML 演算法[17]。這個 hybrid fuzzy GBML 演算法首先以匹茲堡式演算法為主要的架構，先將每個 rule set 編碼成族群中的一條染色體，之後再於每條染色體（即，rule set）上應用密西根式演算法，有效的找到較好的 fuzzy rules，最後將密西根式演算法所找到的 fuzzy rules 替換掉染色體中較差的 fuzzy rules。簡單的說，即是以密西根式演算法作為匹茲堡式演算法中的 heuristic mutation，用以將每條染色體中較差的 fuzzy rules “突變 (mutation)” 為較好的 fuzzy rules。希望能藉著密西根式演算法來加快匹茲堡式演算法的收斂速度，使得這個 hybrid fuzzy GBML 演算法能夠快速的找到一個具高分類正確率的 rule set。

接下來先說明各個符號的定義。假設有一個 data set 給定了 m 個 n 維的訓練樣本 $\mathbf{x}_p = (x_{p1}, \dots, x_{pn})$ ， $p = 1, 2, \dots, m$ ，這 m 個樣本總共分成 M 個類別。意即，現在有一個樣本分類問題，要將 m 個 n 維的訓

練樣本分成 M 類。

3.1 模糊分割的方式與 Fuzzy Rule 的形式

因為一開始的時候，我們並不知道什麼樣的歸屬函數會比較適合這些訓練樣本每個維度的屬性值，所以 Ishibuchi 在每個維度上，同時使用了如圖 3.1 所示的四種均勻模糊分割，如此一來，每個維度上便產生了 14 個歸屬函數（即，fuzzy set），每個 fuzzy set 再各用一個語文變數來代表它。

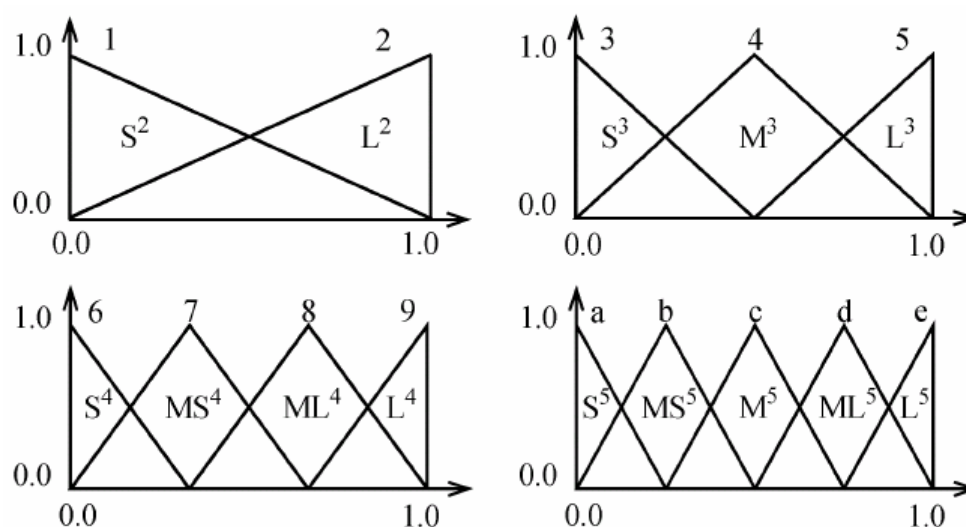


圖 3.1 四種均勻模糊分割

然而，若在給定的訓練樣本中，存在著某些對分類結果不具影響的維度（例如，全部訓練樣本的第三維屬性值均屬於語文變數 MS^5 ），那麼在 FRBCSs 的學習過程中，考慮這個維度的屬性值將造成計算上不

必要的浪費，所以 Ishibuchi 在每個維度上，再加入一個語文變數 *don't care*。而 *don't care* 的歸屬函數則如圖 3.2 所示。

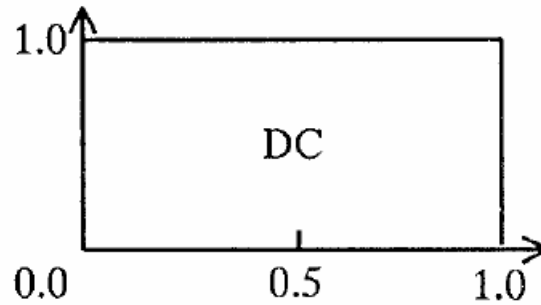


圖 3.2 *don't care* 的歸屬函數

加上 *don't care* 之後，每個維度上便產生了 15 個 fuzzy sets，所以當處理一個 n 維的分類問題時，便有 15^n 條可能的 fuzzy rules，Ishibuchi 希望他們所提出的 hybrid fuzzy GBML 演算法能夠快速的在 15^n 條 fuzzy rules 中，選出少數幾條具代表性的 fuzzy rules 來構成一個具高分類正確率的 FRBCS。

在[17]中，Ishibuchi 使用單一結論類別的 fuzzy rules (fuzzy rules with a single consequent class) 來分類樣本，其形式為

$$\begin{aligned} \text{Rule } R_q : & \text{ If } x_1 \text{ is } A_{q1} \text{ and } \dots \text{ and } x_n \text{ is } A_{qn} \\ & \text{ then Class } C_q \text{ with } CF_q, \quad q = 1, 2, \dots, N_{\text{rule}} \end{aligned}$$

其中， q 代表 fuzzy rule R 的編號，而 $\mathbf{x} = (x_1, \dots, x_n)$ 則是一個 n 維樣本的屬性值向量， A_{qi} 代表 fuzzy rule R_q 前提部份的第 i 個 fuzzy set，

C_q 代表 fuzzy rule R_q 的結論類別 (consequent class), CF_q 代表 fuzzy rule R_q 的 certainty factor, 即 fuzzy rule R_q 的可信程度, N_{rule} 代表一個 rule set 中 fuzzy rules 的總數, 即一個 rule set 是由 N_{rule} 條 fuzzy rules 所組成。

3.2 Fuzzy Rule 的前提部份與結論部份

因為可能的 fuzzy rules 總數將隨著問題維度的增加而成指數性的成長, 計算機 (computer) 的記憶體 (memory) 有時容不下這麼龐大的數字, 所以當應用基因演算法時, [17] 只將 fuzzy rules 的前提部份作編碼, 等基因演算法產生了目前 rule set 中的 fuzzy rules 之後, 再來計算這些 fuzzy rules 的結論部份, 如此一來, 便不需將所有可能的 fuzzy rules 均儲存於記憶體內。以下我們說明[17]在應用基因演算法時, fuzzy rules 前提部份的編碼方式。首先, Ishibuchi 用 15 個符號 (0~9,a,b,c,d,e) 來代表 *don't care* 和 14 個 fuzzy sets (如圖 3.1 及圖 3.2), 若訓練樣本的維度為 n , [17] 便以 n 個字元 (character) 來編碼一條 fuzzy rule 的前提部份, 所以當一個 rule set 中包含了 N_{rule} 條 fuzzy rules 時, [17] 便將這個 rule set 編碼為一條長度為 $n \times N_{rule}$ 的字串 (string), 其中每條長度為 n 的子字串即代表了一條 fuzzy rule 的前提部份。

舉例來說，“1000d0”即代表“If x_1 is S^2 and x_2 is *don't care* and x_3 is *don't care* and x_4 is *don't care* and x_5 is ML^5 and x_6 is *don't care*”，因為 x_2 、 x_3 、 x_4 及 x_6 所對應的 fuzzy set 均為 *don't care*（即，0），表示不論 x_2 、 x_3 、 x_4 及 x_6 這四個維度的屬性值為何，對這條 fuzzy rule 的結論部份均沒有任何影響，所以在表示這條 fuzzy rule 的前提部份時可將其省略，而簡化的表示為 “If x_1 is S^2 and x_5 is ML^5 ”。最後，[17]將一個 rule set 編碼成一條連鎖的字串。如圖 3.3 即代表了一個由 5 條 fuzzy rules 的前提部份所組成的 rule set。

1000d0	e00504	09dba3	230040	c06708
--------	--------	--------	--------	--------

圖 3.3 五條 Fuzzy Rules 所連成的字串

接著我們說明，當一條 fuzzy rule 的前提部份決定了之後，如何依據給定的 data set 計算出它的結論部份（ C_q 及 CF_q ）。

首先，Ishibuchi 以 $\mathbf{A}_q = (A_{q1}, \dots, A_{qn})$ 來代表一條 fuzzy rule R_q 的前提部份，而 \mathbf{A}_q 跟某一個訓練樣本 \mathbf{x}_p 的相容度（compatibility grade）為

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \cdot \mu_{A_{q2}}(x_{p2}) \cdot \dots \cdot \mu_{A_{qn}}(x_{pn})$$

其中， $\mu_{A_{qi}}(\cdot)$ 代表 fuzzy set A_{qi} 的歸屬函數。而當 R_q 的前提部份為 \mathbf{A}_q 時， R_q 的結論類別為類別 h ($h \in \{1, 2, \dots, M\}$) 的信心程度 (confidence) 為

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{\sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p)}{\sum_{p=1}^m \mu_{\mathbf{A}_q}(\mathbf{x}_p)}.$$

所以當一條 fuzzy rule R_q 的前提部份 \mathbf{A}_q 決定了之後，我們便計算

$$c(\mathbf{A}_q \Rightarrow \text{Class } h), \quad h = 1, 2, \dots, M.$$

接著從 M 個信心程度中，找出擁有最大信心程度的類別 h_ω ，作為 fuzzy rule R_q 的結論類別 C_q ，即

$$c(\mathbf{A}_q \Rightarrow C_q) = \max\{c(\mathbf{A}_q \Rightarrow \text{Class } h) \mid h = 1, 2, \dots, M\}.$$

求得 fuzzy rule R_q 的結論類別 C_q 之後，再將這 M 個信心程度中，除了類別 h_ω (即，結論類別 C_q) 之外的信心程度加總起來，然後以 C_q 的 complementary 信心程度 $c(\mathbf{A}_q \Rightarrow \bar{C}_q)$ 來代表這個值，即

$$c(\mathbf{A}_q \Rightarrow \bar{C}_q) = \sum_{\substack{h=1 \\ h \neq C_q}}^M c(\mathbf{A}_q \Rightarrow \text{Class } h).$$

最後以類別 C_q 的信心程度減去 C_q 的 complementary 信心程度，所得的值即為 fuzzy rule R_q 的 certainty factor CF_q ，意即

$$CF_q = c(\mathbf{A}_q \Rightarrow C_q) - c(\mathbf{A}_q \Rightarrow \bar{C}_q)$$

值得注意的是，在演算法的演化過程中，若 CF_q 值的計算結果為負 (<0)，[17]便為 fuzzy rule R_q 重新產生另一個新的前提部份 \mathbf{A}'_q ，直到其 CF_q 值的計算結果大於 0 或等於 0 為止。意即，在 rule set 中不會包含任何 CF_q 值為負的 fuzzy rule。

3.3 Single Winner Rule 的模糊推論方法

在這一節中，我們將說明 single winner rule 的模糊推論方法 (fuzzy reasoning method) 如何決定一個樣本的類別。

假設 S 是一個由 N_{rule} 條 fuzzy rules 所組成的 rule set，當一個類別未知的樣本 \mathbf{x}_p 進入 S 之後，我們計算

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q, \quad q = 1, 2, \dots, N_{\text{rule}}.$$

接著以

$$\mu \mathbf{A}_\omega(\mathbf{x}_p) \cdot CF_\omega = \max\{\mu \mathbf{A}_q(\mathbf{x}_p) \cdot CF_q \mid R_q \in S\}.$$

找出 S 中擁有最大乘積 $\mu \mathbf{A}_\omega(\mathbf{x}_p) \cdot CF_\omega$ 的 fuzzy rule(s) R_ω ，而擁有最大乘積的 fuzzy rule(s) R_ω 即稱為樣本 \mathbf{x}_p 的 winner rule，最後將樣本 \mathbf{x}_p 分類為 R_ω 的結論類別 C_ω 。

若樣本 \mathbf{x}_p 使得 2 條以上的 fuzzy rules 同時擁有最大乘積，但這些 fuzzy rules 的結論類別卻不完全相同，則 rule set S 將拒絕 (reject) 分類這個樣本 \mathbf{x}_p 。

使用 single winner rule 的模糊推論方法，使得我們可以很輕易的計算出密西根式演算法的族群 (即，一個 rule set) 中，每條 fuzzy rule 的 fitness value，因為當一個樣本被其 winner rule R_ω 分類時，我們可以依據分類結果的正確與否 (即，此樣本被正確的分類或錯誤的分類)，輕易的增加或減少 R_ω 的 fitness value。Fitness value 的計算方式有很多種，Ishibuchi 選擇了[18]的計算方式，即

$$\begin{aligned} fitness(R_q) &= NCP(R_q) - NMP(R_q), \\ q &= 1, 2, \dots, N_{rule}. \end{aligned}$$

其中， $NCP(R_q)$ 代表被 R_q 正確分類的訓練樣本個數，而 $NMP(R_q)$ 則代表被 R_q 錯誤分類的訓練樣本個數。Ishibuchi 將密西根式演算法中 fitness value 的計算方式稱為信用指派 (credit assignment)。

3.4 Heuristic Procedures

在[19]中提出了數個 heuristic procedures 來提升密西根式演算法的收斂速度，[17]應用了其中的兩個。第一個是，利用 data set 給定的訓練樣本產生初始 rule set 中的 fuzzy rules。第二個則為，利用被 rule set 錯誤的分類或是拒絕分類的訓練樣本產生新的 fuzzy rules。以下我們將一一舉例說明。

3.4.1 利用訓練樣本產生初始的 Fuzzy Rules

首先假設一個 rule set 是由 N_{rule} 條 fuzzy rules 所組成，一開始我們先從 data set 給定的 m 個訓練樣本中隨機選出 N_{rule} 個，然後為這 N_{rule} 個訓練樣本各別產生一個較適合其屬性值的前提部份。產生的過程如下，假設 $\mathbf{x}_p = (x_{p1}, \dots, x_{pn})$ 為這 N_{rule} 個訓練樣本中的某一個，我們先計算維度上的 14 個 fuzzy sets (如圖 3.1) 各自和屬性值 x_{p1} 的相容度，作為各自被選出作為 x_{p1} 的前提 fuzzy set 的機率，故每個 fuzzy set A_k ($k = 1, 2, \dots, 14$) 被選到的機率為

$$P(A_k) = \frac{\mu A_k(x_{pi})}{\sum_{j=1}^{14} \mu A_j(x_{pi})}.$$

最後依每個 fuzzy set 各自的機率大小，從 14 個 fuzzy sets 中選出一個，作為 x_{p1} 的前提 fuzzy set。而 \mathbf{x}_p 剩餘的屬性值， x_{p2}, \dots, x_{pn} ，亦依上述方法產生一個較適合的前提 fuzzy set 之後，便產生了一個較適合 \mathbf{x}_p 其屬性值的前提部份。剩下的 $(N_{\text{rule}} - 1)$ 個訓練樣本亦依上述過程產生 $(N_{\text{rule}} - 1)$ 個前提部份。接下來以預設的突變機率 $P_{\text{don't care}}$ ，將前提部份的 fuzzy sets 突變為 *don't care*。最後計算這 N_{rule} 條 fuzzy rules 的結論部份，rule set 中初始的 N_{rule} 條 fuzzy rules 便產生完畢。

舉例來說，假設一個 rule set 是由 $N_{\text{rule}} = 5$ 條 fuzzy rules 所組成，而 data set 給定了 20 個 2 維的訓練樣本 $\mathbf{x}_p = (x_{p1}, x_{p2})$ ， $p = 1, 2, \dots, 20$ ，一開始我們先從這 20 個訓練樣本中隨機選出 5 個，假設選出的 5 個訓練樣本為 \mathbf{x}_3 、 \mathbf{x}_5 、 \mathbf{x}_9 、 \mathbf{x}_{15} 及 \mathbf{x}_{16} ，且其中的 $\mathbf{x}_3 = (0.2, 0.5)$ ，接著我們計算維度上的 14 個 fuzzy sets 各自和 $x_{31} = 0.2$ 的相容度。則第一個 fuzzy set S^2 被選到作為 $x_{31} = 0.2$ 的前提 fuzzy set 的機率為

$$P(A_1) = \frac{\mu_{A_1}(x_{31})}{4} = \frac{0.8}{4} = 0.2$$

，而第二個 fuzzy set L^2 被選到的機率為

$P(A_2) = 0.05$ ，依此類推，14 個 fuzzy sets 的機率均計算結束後，再依每個 fuzzy set 各自的機率大小，從 14 個 fuzzy sets 中選出一個，作為 $x_{31} = 0.2$ 的前提 fuzzy set。然後再依上述方法產生 $x_{32} = 0.5$ 的前提 fuzzy set 之後，便產生了一個較適合 \mathbf{x}_3 其屬性值的前提部份。再依上

述過程產生較適合 \mathbf{x}_5 、 \mathbf{x}_9 、 \mathbf{x}_{15} 及 \mathbf{x}_{16} 的前提部份之後，便產生了 5 個較適合訓練樣本的前提部份。接下來以預設的突變機率 $P_{don't\ care}$ ，將前提部份的 fuzzy set 突變為 *don't care*，假設 $P_{don't\ care} = 0.5$ ，則每個前提 fuzzy set 突變為 *don't care* 的機率即為 0.5。最後計算這 5 條 fuzzy rules 的結論部份，rule set 中初始的 5 條 fuzzy rules 便產生完畢。

以往的方法大部份均使用隨機挑選前提 fuzzy sets 的方式來產生初始 rule set 中的 fuzzy rules，但是當樣本空間的維度很高的時候，以隨機的方式很容易產生前提部份沒有包含任何訓練樣本的 fuzzy rules，造成密西根式演算法的收斂速度緩慢，使用這個 heuristic procedure 能保證產生的前提部份至少包含了一個以上的訓練樣本，可加快密西根式演算法的收斂速度。

3.4.2 利用被錯誤的分類或是拒絕分類的訓練樣本產生新的 Fuzzy Rules

這個 heuristic procedure 則利用被目前的 rule set 錯誤的分類或是拒絕分類的訓練樣本，先依 3.4.1 節所述的 heuristic procedure 產生較適合其屬性值的 fuzzy rules，然後將產生的 fuzzy rules 替換進目前的 rule set。希望能在下一個 generation 時，正確的分類這些在上一個 generation 時被錯誤的分類或是拒絕分類的訓練樣本，以加快密西根

式演算法的收斂速度。

3.5 基因運算方式

在這一節中，我們將說明 Ishibuchi 在[17]中所使用的基因運算方式，接著再說明 elitist strategy 的運算方式。首先假設整個族群是由 N_{pop} 條染色體所組成。

匹茲堡式演算法是將每個 rule set 編碼成族群中的一條染色體（如圖 2.3），而一條染色體（即，rule set）則由許多 fuzzy rules 所組成（如圖 3.4 所示）。

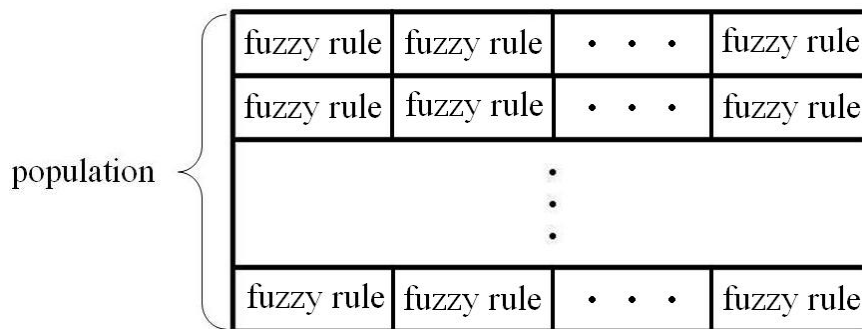


圖 3.4 匹茲堡式演算法的族群

而 Ishibuchi 在匹茲堡式演算法中所使用的選擇（selection）方式為 binary tournament selection，即每次均從族群中隨機挑選 2 條染色體，然後將 fitness value 較佳的染色體複製至交配池（matching pool）中，如此重複 N_{pop} 次之後，便挑選出 N_{pop} 條染色體至交配池中。而交配

(crossover) 方式為，從交配池中隨機選出 2 條染色體，然後以染色體中的 fuzzy rule 為單位，進行 rule-wise 的交配運算。突變方式則為，若某個 fuzzy set 已確定要進行突變運算，則此 fuzzy set 突變為 *don't care* 的機率為 $P_{don't\ care}$ ，而突變為 14 個語文變數中任一個的機率均為 $(1 - P_{don't\ care})/14$ 。

我們舉個例子來說明突變的過程，首先假設 $P_{don't\ care} = 0.5$ 。若一個 fuzzy set 已確定要進行突變運算，則此 fuzzy set 突變為 *don't care* 的機率為 0.5，而突變為 14 個語文變數中任一個的機率均為 $(1 - 0.5)/14 \cong 0.036$ 。

密西根式演算法則是將每條 fuzzy rule 編碼成族群中的一條染色體（如圖 2.5），而一條染色體（即，fuzzy rule）則由許多 fuzzy sets 所組成，最後再用整個族群來代表一個 rule set（如圖 3.5 所示）。

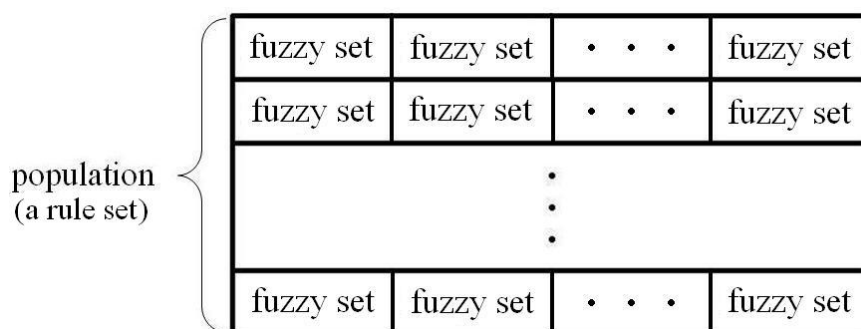


圖 3.5 密西根式演算法的族群

Ishibuchi 於密西根式演算法中所使用的基因運算方式，大致上和匹茲

堡式演算法相同。選擇方式亦使用 binary tournament selection。交配的方式則以 fuzzy rule 中的 fuzzy set 為單位，進行 bit-wise 的交配運算。而突變運算亦和匹茲堡式演算法相同。

而 elitist strategy 的運算方式為，我們將歷代 (generation) 族群中，擁有最佳 fitness value 的染色體保存下來，這條保存下來的染色體即稱為 elitist，若在這一個 generation 的族群中，產生了比 elitist 更好的染色體，則將 elitist 更新為這條更好的染色體，否則便從這一個 generation 的族群中，隨機挑一條染色體，然後以 elitist 取代它，以保證下一個 generation 的族群不致於比這一個 generation 的族群更差。

3.6 Hybrid Fuzzy GBML 演算法

最後我們說明 Ishibuchi 等人所提出的 hybrid fuzzy GBML 演算法 [17]。首先，假設整個族群是由 N_{pop} 個 rule sets 所組成，其中的每個 rule set 則由 N_{rule} 條 fuzzy rules 所組成。

Hybrid Fuzzy GBML Algorithm

Step 1) 以 3.4.1 節所述的 heuristic procedure，產生 N_{pop} 個由 N_{rule} 條 fuzzy rules 組成的 rule sets。

Step 2) 依正確分類的訓練樣本個數，計算族群中每個 rule set 的

fitness value。

Step 3) 以匹茲堡式演算法產生一個新的族群。

Step 4) 依預設的機率 ([17]中預設的機率為 0.5) 對族群中的 rule sets 個別執行單一回合 (iteration) 的密西根式演算法 (即, 產生及替換 fuzzy rules 的部份)。

Step 5) 執行 elitist strategy。

Step 6) 若已達要求的分類正確率 (預設為 100%), 或者已達預設的 generation 數, 則結束演算法, 否則回到 Step 2。

第四章 我們的方法

4.1 動機

近年來越來越多的文獻指出，在模糊分割相同的情況下，改變 fuzzy rule 的形式、certainty factor 的計算方式或是模糊推論的方式，都會影響一個 FRBCS 的分類結果[20],[21]。在[20]中還舉了一個簡單的例子，我們若將一個 $[-2,2] \times [-2,2]$ 的平面樣本空間分割成 5×5 的均勻模糊分割，然後以單一結論類別的 fuzzy rules 與 single winner rule 的模糊推論方法來模擬方程式 $x_1 + x_2 = 0.05$ 的決策分界線，則模擬結果將如圖 4.1 的左圖所示。

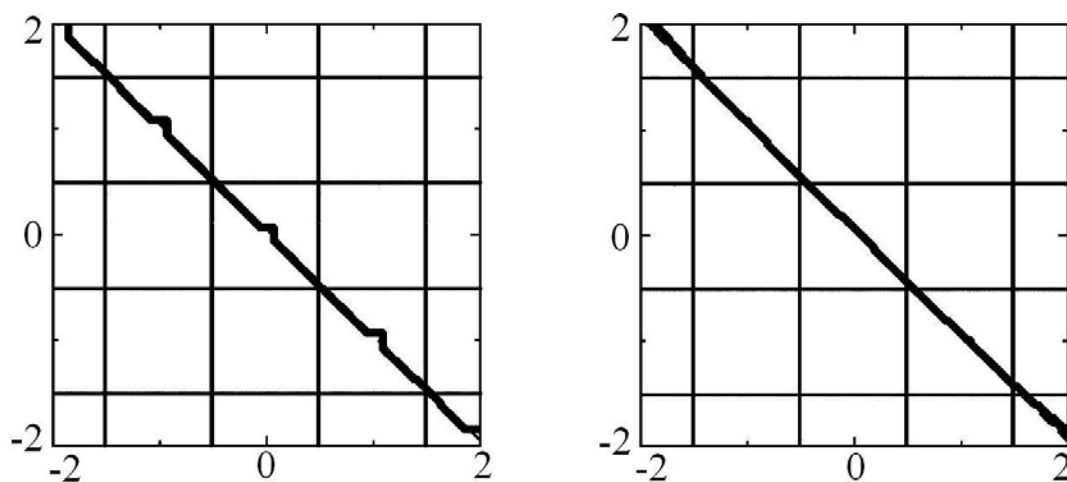


圖 4.1 兩種模糊模型分別產生的決策分界線

但若我們以多結論類別的 fuzzy rules (fuzzy rules with multiple consequent classes) 與 weighted voting 的模糊推論方法來模擬相同的方程式，則模擬結果將如圖 4.1 的右圖所示。從圖 4.1 中我們可以很明顯的看出，右圖的決策分界線所得到的分類結果是比較好的。所以接下來，我們首先說明多結論類別的 fuzzy rules 與 weighted voting 的模糊推論方法。

4.2 Fuzzy Rule 的形式及產生結論部份的方式

多結論類別的 fuzzy rules 其形式為

$$\begin{aligned} \text{Rule } R_q : & \text{ If } x_1 \text{ is } A_{q1} \text{ and } \dots \text{ and } x_n \text{ is } A_{qn} \\ & \text{ then Class 1 with } CF_{q1} \text{ and } \dots \text{ and Class } M \text{ with } CF_{qM}, \\ & q = 1, 2, \dots, N_{\text{rule}}. \end{aligned}$$

其中， q 代表 fuzzy rule R 的編號，而 $\mathbf{x} = (x_1, \dots, x_n)$ 則是一個 n 維樣本的屬性值向量， A_{qi} 代表 fuzzy rule R_q 前提部份的第 i 個 fuzzy set， CF_{qh} 代表 fuzzy rule R_q 的結論類別為類別 h 的 certainty factor， N_{rule} 代表一個 rule set 中 fuzzy rules 的總數，即一個 rule set 是由 N_{rule} 條 fuzzy rules 所組成。

如同單一結論類別的 fuzzy rules，當一條多結論類別的 fuzzy rule

R_q 其前提部份決定了之後，我們便能依據給定的 data set 計算出 R_q 的結論部份，且計算過程更為簡單快速，我們直接使用信心程度 $c(\mathbf{A}_q \Rightarrow \text{Class } h)$ 作為結論類別 h 的 CF_{qh} 值，即

$$CF_{qh} = c(\mathbf{A}_q \Rightarrow \text{Class } h), \quad h = 1, 2, \dots, M.$$

而在演算法的演化過程中，若所有的 CF_{qh} 值皆為 0，我們便為 fuzzy rule R_q 重新產生另一個新的前提部份 \mathbf{A}'_q ，直到存在著某個 CF_{qh} 值大於 0 為止。

當我們使用了多結論類別的 fuzzy rules 之後，如果模糊推論方法還是使用 single winner rule 的方法，所得到的決策分界線將會和使用單一結論類別的 fuzzy rules 所得到的決策分界線相同（即，分類結果相同）。若改變了 fuzzy rule 的形式之後，分類結果仍然相同，那改變 fuzzy rule 的形式就變的毫無意義了。所以當我們改變了 fuzzy rules 的形式之後，模糊推論方法勢必也要隨著改變。為了敘述上的方便，之後我們將使用單一結論類別的 fuzzy rules 與 single winner rule 的模糊推論方法之模糊模型（fuzzy model）簡稱為“SS 模糊模型”，而使用多結論類別的 fuzzy rules 與 weighted voting 的模糊推論方法之模糊模型則簡稱為“MW 模糊模型”。

4.3 Weighted Voting 的模糊推論方法

在這一節中，我們將說明且舉例 weighted voting 的模糊推論方法如何決定一個樣本的類別。

假設 S 是一個由 N_{rule} 條 fuzzy rules 所組成的 rule set，當一個類別未知的樣本 \mathbf{x}_p 進入 S 之後，我們計算

$$SC(h) = \sum_{q=1}^{N_{\text{rule}}} \mu \mathbf{A}_q(\mathbf{x}_p) \cdot CF_{qh}, \quad h=1,2,\dots,M.$$

計算結束之後將會得到 M 個 $SC(h)$ 值，分別代表樣本 \mathbf{x}_p 屬於各個類別 h 的機率，所以我們從這 M 個 $SC(h)$ 值中找出一個最大值 $SC(\omega)$ ，然後將樣本 \mathbf{x}_p 分類為類別 ω 。

若樣本 \mathbf{x}_p 使得 2 個以上的 $SC(h)$ 值同時擁有最大值，則 rule set S 將拒絕分類這個樣本 \mathbf{x}_p 。

舉例來說，假設 S 是一個由 $N_{\text{rule}} = 2$ 條擁有 3 個結論類別的 fuzzy rules 所組成的 rule set：

Rule R_1 : If x is S^2 then $CF_{11} = 0.3$ and $CF_{12} = 0.1$ and $CF_{13} = 0.6$

Rule R_2 : If x is L^2 then $CF_{21} = 0.1$ and $CF_{22} = 0.5$ and $CF_{23} = 0.4$

而一個類別未知的樣本 \mathbf{x}_p 和 fuzzy set S^2 的相容度為 0.8，和 fuzzy set

L^2 的相容度為 0.2，當此樣本 \mathbf{x}_p 進入 S 之後，我們計算

$$SC(1) = \sum_{q=1}^2 \mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_{q1} = 0.8 \cdot 0.3 + 0.2 \cdot 0.1 = 0.26$$

再依 $h = 2, 3$ 類推，得到

$$SC(2) = 0.8 \times 0.1 + 0.2 \times 0.5 = 0.18$$

$$SC(3) = 0.8 \times 0.6 + 0.2 \times 0.4 = 0.56$$

最後得到 3 個 $SC(h)$ 值，分別代表樣本 \mathbf{x}_p 屬於各個類別的機率，因其中的 $SC(3) = 0.56$ 擁有最大值，表示樣本 \mathbf{x}_p 屬於類別 3 的機率最大，故我們將樣本 \mathbf{x}_p 分類為類別 3，意即，推論結果為樣本 \mathbf{x}_p 屬於第 3 類。

4.4 過程及計算時間的比較

在這一節中，我們將比較單一結論類別的 fuzzy rules 與多結論類別的 fuzzy rules，在產生結論部份時的計算過程，以及 single winner rule 與 weighted voting 這兩種方法的模糊推論過程。最後再以兩個分別使用了 SS 模糊模型與 MW 模糊模型的匹茲堡式演算法來比較這些過程所需的計算時間。

4.4.1 Fuzzy Rule 產生結論部份的過程

當一條單一結論類別的 fuzzy rule R_q 其前提部份決定了之後，我們需計算出 R_q 的結論類別 C_q 及 certainty factor CF_q ，其計算過程為

Step 1) 計算 $c(\mathbf{A}_q \Rightarrow \text{Class } h)$, $h = 1, 2, \dots, M$.

Step 2) $c(\mathbf{A}_q \Rightarrow C_q) = \max\{c(\mathbf{A}_q \Rightarrow \text{Class } h) \mid h = 1, 2, \dots, M\}$.

Step 3) $c(\mathbf{A}_q \Rightarrow \overline{C}_q) = \sum_{\substack{h=1 \\ h \neq C_q}}^M c(\mathbf{A}_q \Rightarrow \text{Class } h)$.

Step 4) $CF_q = c(\mathbf{A}_q \Rightarrow C_q) - c(\mathbf{A}_q \Rightarrow \overline{C}_q)$.

而一條多結論類別的 fuzzy rule R_q 其前提部份決定了之後，我們只需計算出 R_q 每個類別的信心程度，然後直接使用信心程度 $c(\mathbf{A}_q \Rightarrow \text{Class } h)$ 作為結論類別 h 的 CF_{qh} 值，即

$$CF_{qh} = c(\mathbf{A}_q \Rightarrow \text{Class } h), \quad h = 1, 2, \dots, M.$$

由以上的比較可明顯的看出，多結論類別的 fuzzy rules 在產生結論部份時，要比單一結論類別的 fuzzy rules 簡單且快速了許多。

4.4.2 模糊推論過程

接下來我們再比較 single winner rule 與 weighted voting 這兩種方法的模糊推論過程。首先假設一個 rule set 是由 N_{rule} 條 fuzzy rules 所組成，而現在有一個類別未知的樣本 \mathbf{x}_p 進入這個 rule set。

Single Winner Rule Method

Step 1) 計算 $\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q$, $q = 1, 2, \dots, N_{\text{rule}}$ 。

Step 2) 找出 N_{rule} 個乘積中的最大乘積。

Step 3) 若擁有最大乘積的 fuzzy rule 數目唯一，則分類此樣本 \mathbf{x}_p 並結束演算法。

Step 4) 若擁有最大乘積的 fuzzy rules 其結論類別均相同，則分類此樣本 \mathbf{x}_p 並結束演算法。

Step 5) 拒絕分類此樣本 \mathbf{x}_p 。

在模糊推論過程中，若 Step 3 不成立，表示有兩條以上的 fuzzy rules 同時擁有最大乘積。若 Step 4 不成立，表示這兩條以上擁有最大乘積的 fuzzy rules 其結論類別不完全相同。

Weighted Voting Method

Step 1) 計算 $SC(h) = \sum_{q=1}^{N_{rule}} \mu \mathbf{A}_q(\mathbf{x}_p) \cdot CF_{qh}$ ， $h = 1, 2, \dots, M$ 。

Step 2) 找出 M 個 $SC(h)$ 值中的最大值。

Step 3) 若有兩個以上的 $SC(h)$ 同時擁有最大值，則拒絕分類此樣本 \mathbf{x}_p 並結束演算法。

Step 4) 分類樣本 \mathbf{x}_p 。

在模糊推論過程中，若 Step 3 不成立，表示擁有最大值的 $SC(h)$ 個數唯一。

因為 weighted voting 的模糊推論過程需將 rule set 中所有 fuzzy rules 的 $\mu_{A_q}(\mathbf{x}_p) \cdot CF_{qh}$ 加總起來，所以其計算時間將隨著 rule set 中 fuzzy rules 數 (N_{rule}) 的增加而增加，通常會比 single winner rule 的模糊推論過程來的耗時，不過值得慶幸的是，一個好的 rule set 中不會包含太多的 fuzzy rules。

4.4.3 計算時間的比較

我們以兩個分別使用了 SS 模糊模型與 MW 模糊模型的匹茲堡式演算法來比較其計算時間。實驗結果（說明於第五章中）顯示，雖然 weighted voting 的模糊推論過程較 single winner rule 的模糊推論過程來的耗時，但若連同 fuzzy rules 產生結論部份時所需的計算時間一併計算，MW 模糊模型還是較 SS 模糊模型（[17]所使用的模糊模型）更為省時（在 $N_{rule} = 10$ 的情況下）。

4.5 改變模糊模型

首先，我們先於 hybrid fuzzy GBML 演算法的架構之下，將 SS 模糊模型變更為 MW 模糊模型，但變更之後卻面臨了一些隨之而來的問題。在這一節中，我們將提出這些問題並同時說明我們的因應方法。

I. 信用指派的問題

[17]使用 single winner rule 的模糊推論方法來分類樣本，使得我們在 rule set 應用密西根式演算法時，可依 3.3 節所述的信用指派方式，輕易的計算出 rule set 中每條 fuzzy rule 的 fitness value。但是當我們使用了 weighted voting 的模糊推論方法之後，並不會產生所謂的 winner rule，為此我們提出了另一種信用指派的方式來計算密西根式演算法的族群中，每條 fuzzy rule 的 fitness value，我們將密西根式演算法族群中的每條染色體均視為一個由一條 fuzzy rule 所組成的 rule set，故 fitness value 的計算方式為

$$\begin{aligned} \text{fitness}(R_q) &= NP(\omega) - NP(\bar{\omega}), \\ q &= 1, 2, \dots, N_{\text{rule}}. \end{aligned}$$

其中， ω 代表 fuzzy rule R_q 的多個結論類別中，信心程度最大的類別，而 $\bar{\omega}$ 則代表除了類別 ω 之外的所有類別， $NP(\omega)$ 代表類別 ω 的訓練樣本總數， $NP(\bar{\omega})$ 則代表類別 $\bar{\omega}$ 的訓練樣本總數。因為使用了多結論類別的 fuzzy rule 與 weighted voting 的模糊推論方法，加上我們所使用的 fitness value 計算方式較[17]所使用的方式簡單，使得此時的演算法在計算時間上較[17]快速許多，但分類正確率卻比[17]略低(實驗結果列於第五章中)。於是我們再提出另一個想法。

II. 密西根式演算法存在的必要性

在[17]中已清楚說明，密西根式演算法在 hybrid fuzzy GBML 演算法中的功能為，找到數條較好的 fuzzy rules 替換掉 rule set 中較差的數條 fuzzy rules，以加快匹茲堡式演算法的收斂速度，若我們將密西根式演算法從 hybrid fuzzy GBML 演算法中移除，將使得匹茲堡式演算法的收斂速度減緩，而最後的分類正確率較 hybrid fuzzy GBML 演算法略低（因收斂速度較慢），但計算時間則約減少為原本的 58%（實驗結果列於第五章中）。

而 weighted voting 的模糊推論方法是將 rule set 中，每一條 fuzzy rule 的 $\mu_{A_q}(x_p) \cdot CF_{qh}$ 加總起來，整個推論的過程與 rule set 中的每一條 fuzzy rule 均息息相關，並沒有所謂的單一較好的 fuzzy rule。如圖 4.2 所示。

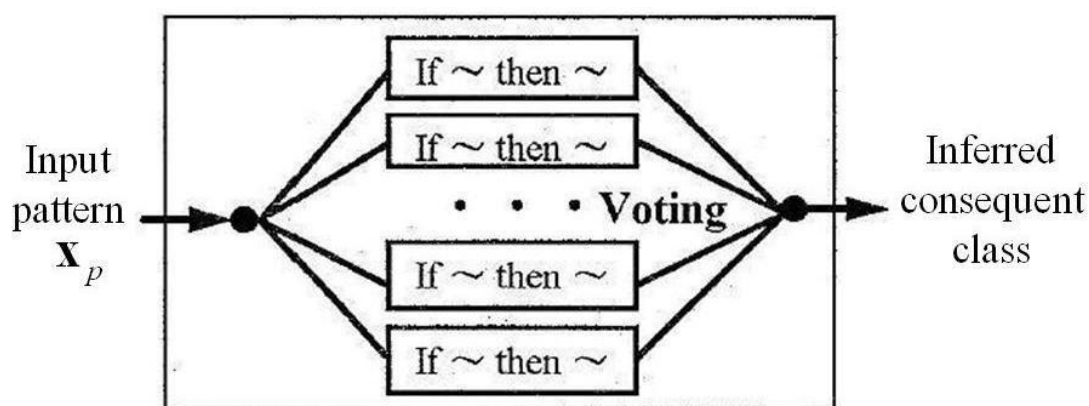


圖 4.2 Weighted Voting 的模糊推論方法

簡單的說，當使用 weighted voting 的模糊推論方法時，不適合使用密西根式演算法。因此在此節的演算法架構中，我們將密西根式演算法從整個演算法中移除。實驗的結果（列於第五章中）證實了我們的想法，我們得到了一個比 4.5-I 節的架構略高的分類正確率及更快的計算速度（因移除了整個密西根式演算法），但是這個架構的分類正確率還是比[17]略低。

但在[17]中，Ishibuchi 使用 generation 的數目來比較方法之間的收斂速度（如圖 4.3 所示）。

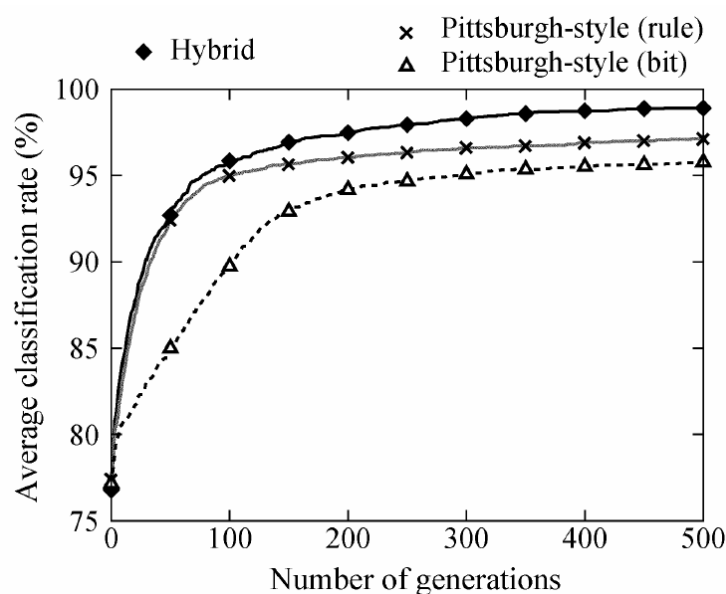


圖 4.3 [17]中的收斂速度比較

我們認為這不是非常合理的比較方式，因為 hybrid fuzzy GBML 演算法的收斂速度雖然較匹茲堡式演算法為快，但其每個 generation 所需

的計算時間亦相對的增加，就如同我們以另一個需數倍計算時間的方法來些微加快原本演算法的收斂速度一般，所以在本論文中，我們以計算所需的時間來比較方法之間的收斂速度。最後，我們提出了一個改良式 fuzzy GBML 演算法，藉著減少計算時間來加快 fuzzy GBML 演算法的收斂速度，我們將在下一節中詳述並舉例說明我們所提出的方法。

4.6 改良式 Fuzzy GBML 演算法

我們所提出的改良式 fuzzy GBML 演算法，首先以 MW 模糊模型取代了[17]中所使用的 SS 模糊模型，接著再以 3.4.2 節所述的 heuristic procedure 來取代密西根式演算法。因此，在我們的改良式 fuzzy GBML 演算法中，並沒有使用任何的密西根式演算法。以下將說明我們所提出的改良式 fuzzy GBML 演算法。首先，假設整個族群是由 N_{pop} 個 rule sets 所組成，其中的每個 rule set 則由 N_{rule} 條 fuzzy rules 所組成，而在執行 Step 6 時，欲替換的 fuzzy rules 數為 $N_{replace}$ 條。

Our Improved Fuzzy GBML Algorithm

Step 1) 以 3.4.1 節所述的 heuristic procedure，產生 N_{pop} 個由 N_{rule} 條 fuzzy rules 組成的 rule sets。

Step 2) 依正確分類的訓練樣本個數，計算族群中每個 rule set 的 fitness value。

Step 3) 以匹茲堡式演算法產生一個新的族群。

Step 4) 執行 elitist strategy。

Step 5) 依正確分類的訓練樣本個數，計算族群中每個 rule set 的 fitness value。

Step 6) 對族群中的每個 rule set，任選 N_{replace} 個被這個 rule set 錯誤的分類或是拒絕分類的訓練樣本，接著以 3.4.2 節所述的 heuristic procedure，利用這 N_{replace} 個訓練樣本產生 N_{replace} 條新的 fuzzy rules，最後隨機地替換掉 rule set 中的 N_{replace} 條 fuzzy rules。

Step 7) 若已達要求的分類正確率，或者已達預設的 generation 數，則結束演算法，否則回到 Step 2。

接下來，我們以一個簡單的例子來說明我們所提出的改良式 fuzzy GBML 演算法。首先，假設給定的訓練樣本數為 150 個，而整個族群是由 $N_{\text{pop}} = 10$ 個 rule sets 所組成，其中的每個 rule set 則由 $N_{\text{rule}} = 20$ 條 fuzzy rules 所組成，而在執行 Step 6 時，欲替換的 fuzzy rules 數為 $N_{\text{replace}} = 2$ 條。

Step 1) 從 150 個訓練樣本中隨機選出 20 個，接著以 3.4.1 節所述的

heuristic procedure，產生第一個 rule set 中，初始的 20 條 fuzzy rules。剩餘的 9 個 rule sets 也以相同的方法來產生 rule set 中，初始的 20 條 fuzzy rules。

Step 2) 依正確分類的訓練樣本個數，計算這 10 個 rule sets 的 fitness value。假設第一個 rule set 正確的分類了 120 個訓練樣本，則此 rule set 的 fitness value 便為 120。

Step 3) 以匹茲堡式演算法產生一個新的族群。

Step 4) 從這 10 個 rule sets 中任意選出一個，以 elitist 來取代它。

Step 5) 依正確分類的訓練樣本個數，計算這 10 個 rule sets 的 fitness value。

Step 6) 假設此時第一個 rule set 的 fitness value 仍為 120，則從被此 rule set 錯誤的分類或是拒絕分類的 30 個訓練樣本中，任意選出 2 個，接著以 3.4.2 節所述的 heuristic procedure，利用這 2 個訓練樣本產生 2 條新的 fuzzy rules，最後隨機地替換掉第一個 rule set 中的 2 條 fuzzy rules。剩餘的 9 個 rule set 亦依上述方法產生並替換掉 rule set 中的 2 條 fuzzy rules。

Step 7) 若已達要求的分類正確率，或者已達預設的 generation 數，則結束演算法，否則回到 Step 2 繼續下一個回合的運算。

改良式 fuzzy GBML 演算法於每個 generation 中均計算每個 rule

set 的 fitness value 兩次，因為當演算法執行了 Step 3 或 Step 6 之後，均會產生一個和之前不同的族群，所以我們立即在此時計算每個 rule set 的 fitness value，以保持 elitist 隨時為最佳。

我們的改良式 fuzzy GBML 演算法首先以計算較為快速的 MW 模糊模型來取代原本 hybrid fuzzy GBML 演算法所使用的 SS 模糊模型，接著再以計算步驟較精簡的 heuristic procedure 來取代原本計算繁複的密西根式演算法。種種實驗亦證明了，我們的方法不但計算速度更快、收斂速度更快，且還提升了些微的分類正確率。

第五章 實驗結果與分析

在這一章中，我們實作了 Ishibuchi 等人於[17]中所提出的 hybrid fuzzy GBML 演算法，以及我們於第四章中所提出的改良式 fuzzy GBML 演算法，並選用了 7 個非常具代表性的 data sets (如表 5.1) 來比較兩者之間的分類正確率、計算時間及收斂速度。而實驗平台 (PC) 所安裝的 CPU 為 Athlon XP 2500+。

DATA SETS USED IN OUR COMPUTER SIMULATIONS

Data set	Number of attributes	Number of patterns	Number of classes
Breast W	9	683*	2
Diabetes	8	768	2
Glass	9	214	6
Heart C	13	297*	5
Iris	4	150	3
Sonar	60	208	2
Wine	13	178	3

* Incomplete patterns with missing values are not included.

表 5.1 實驗所使用的 Data Sets

在實驗開始之前，我們先作了一些準備工作。首先，為了將相同的模糊分割 (如圖 3.1 及圖 3.2) 應用於樣本空間中的每個維度，我們將這 7 個 data sets 每個維度的屬性值均 normalized 到[0,1]，每個維

度的最小屬性值為 0，而最大屬性值則為 1。接著，我們將 Breast W data 及 Heart C data 中，屬性值遺失 (missing value) 或是不完整 (incomplete) 的樣本從 data set 裡移除。

我們的實驗所設定的參數值為

Number of fuzzy rules	10
Number of rule sets	10
Crossover probability	0.9
Mutation probability	$1/n$
Number of replaced rules	1
Total number of generations	1000

表 5.2 實驗所設定的參數值

而突變為 *don't care* 的機率 $P_{don't\ care}$ 則依每個 data set 各自的維度高低來改變其預設值。若 data set 的維度越高，則 $P_{don't\ care}$ 的預設值就越大。

在我們的實驗中，每個 data set 其 $P_{don't\ care}$ 的預設值為

Data Set	Number of Attributes	Probability of Don't Care
Breast W	9	0.7
Diabetes	8	0.7
Glass	9	0.7
Heart C	13	0.8
Iris	4	0.5
Sonar	60	0.95
Wine	13	0.8

表 5.3 Data Sets 及其 $P_{don't\ care}$ 的預設值

5.1 產生結論部份及模糊推論的時間比較

在這一節中，我們以兩個分別使用了 SS 模糊模型與 MW 模糊模型的匹茲堡式演算法來比較其計算時間。實驗結果如表 5.4 所示。

Data Set	Mode	Consequent	Reasoning	Total
Breast W	SS	317	42	359
	MW	160	80	240
Diabetes	SS	326	43	369
	MW	164	81	245
Glass	SS	305	13	318
	MW	149	74	223
Heart C	SS	455	24	479
	MW	230	113	343
Iris	SS	57	5	62
	MW	29	14	43
Sonar	SS	493	63	556
	MW	247	122	369
Wine	SS	156	14	170
	MW	64	32	96

表 5.4 兩種模糊模型的時間比較（單位：秒）

其中，Consequent 代表 fuzzy rules 產生結論部份所需的計算時間，而 Reasoning 則代表模糊推論所需的計算時間。在這一章中，所有的實驗數據皆為 50 次實驗之平均值。

實驗結果顯示，weighted voting 的模糊推論過程較 single winner rule 的模糊推論過程來的耗時，且當 data 的類別數 (M) 較多時(如，

Glass data 及 Heart C data)，相對於 weighted voting 的模糊推論方法，single winner rule 的模糊推論方法將更為省時。但若連同 fuzzy rules 產生結論部份時所需的計算時間一併計算，MW 模糊模型還是較 SS 模糊模型來的省時（在 $N_{rule} = 10$ 的情況下）。若我們將匹茲堡式演算法的模糊模型由 SS 模糊模型改變為 MW 模糊模型，將可省下約 33.26%（7 個 data sets 的平均值）的計算時間。

5.2 改變模糊模型之後的因應方法

在這一節中，我們實作了 4.5 節中所提及的種種假設。首先，我們將密西根式演算法從 hybrid fuzzy GBML 演算法中移除（即成為一般的匹茲堡式演算法）。

Data Set	Correct (%)		Time (s)	
	Hybrid	Pitts.	Hybrid	Pitts.
Breast W	98.08	98.02	828	360
Diabetes	78.25	78.84	856	372
Glass	72.84	72.09	860	343
Heart C	60.78	60.65	1118	503
Iris	98.46	98.47	167	64
Sonar	89.71	88.89	1085	558
Wine	99.78	99.21	227	173

表 5.5 [17] 移除密西根式演算法之後的實驗結果

實驗結果(表 5.5)顯示，匹茲堡式演算法的計算時間約為 hybrid fuzzy

GBML 演算法的 48.25% (7 個 data sets 的平均值), 但因收斂速度較 hybrid fuzzy GBML 演算法緩慢, 造成分類正確率略為下降 ([17] 中已說明)。

接著, 我們實作了 4.5-I 節及 4.5-II 節中所提出的兩種改變模糊模型之後的因應方法。實驗結果則如表 5.6 所示。

Data Set	Correct (%)		Time (s)	
	4.5-I	4.5-II	4.5-I	4.5-II
Breast W	97.97	98.00	314	240
Diabetes	78.29	78.49	320	246
Glass	68.70	69.75	295	223
Heart C	65.43	66.11	450	344
Iris	98.52	98.56	56	43
Sonar	86.40	87.39	489	369
Wine	99.01	99.49	156	96

表 5.6 改變模糊模型之後的因應方法之實驗結果

在 4.5-I 節所提出的方法中, 我們先於 hybrid fuzzy GBML 演算法的架構之下, 將 SS 模糊模型變更為 MW 模糊模型, 接著再以我們所提出的信用指派方式來取代原本的信用指派方式, 以解決 weighted voting 的模糊推論方法無法產生 winner rule 的問題。而在 4.5-II 節所提出的方法中, 我們亦先於 hybrid fuzzy GBML 演算法的架構之下, 將 SS 模糊模型變更為 MW 模糊模型, 接著再將整個密西根式演算法從 hybrid fuzzy GBML 演算法中移除 (即成為一個使用 MW 模糊模型的

匹茲堡式演算法)。

實驗結果顯示，4.5-II 節所提出的演算法因為不需要執行密西根式演算法，所以計算速度較 4.5-I 節所提出演算法更快，且分類正確率更略為提高。由此得知，當匹茲堡式演算法使用 weighted voting 的模糊推論方法時，的確不適合再使用密西根式演算法。

5.3 分類正確率與計算時間的比較

在這一節中，我們比較了 Ishibuchi 等人於[17]中所提出的 hybrid fuzzy GBML 演算法，與我們於 4.6 節中所提出的改良式 fuzzy GBML 演算法，兩者之間的分類正確率、分類錯誤率及計算時間。

Data Set	Correct (%)		Error (%)	
	Hybrid	Our	Hybrid	Our
Breast W	98.08	98.13	1.91	1.87
Diabetes	78.25	78.80	21.68	21.16
Glass	72.84	69.58	26.14	30.36
Heart C	60.78	66.40	38.37	33.20
Iris	98.46	98.89	1.54	1.11
Sonar	89.71	88.85	10.28	11.15
Wine	99.78	99.80	0.22	0.20

表 5.7 分類正確率及分類錯誤率的比較

實驗結果 (表 5.7) 顯示，某些 data set 會比較適合特定的模糊推論方法及模糊模型。例如，在 Glass data、Heart C data 及 Sonar data

中，兩種 fuzzy GBML 演算法的分類正確率分別相差了 3.26%、5.62% 及 0.86%。發生這種情形時，我們還是希望能夠從中挑選出一個較適合此 data set 的模糊模型。而當模糊推論方法對 data set 的分類正確率影響較小時（例如，Breast W data、Diabetes data、Iris data 及 Wine data），我們的改良式 fuzzy GBML 演算法通常擁有較 hybrid fuzzy GBML 演算法略佳的分類正確率。

Data Set	Time (s)	
	Hybrid	Our
Breast W	828	412
Diabetes	856	431
Glass	860	388
Heart C	1118	596
Iris	167	73
Sonar	1085	636
Wine	227	101

表 5.8 計算時間的比較

而在計算時間方面，雖然不同效能的實驗平台（PC）將有不同的計算時間，但若依計算時間的比例來看，改良式 fuzzy GBML 演算法所需的計算時間約只有 hybrid fuzzy GBML 演算法的 49.34%（7 個 data sets 的平均值）。

5.4 收斂速度的比較

在這一節中，我們比較了 hybrid fuzzy GBML 演算法與改良式 fuzzy GBML 演算法的收斂速度。在本論文中，我們以計算時間來比較兩者之間的收斂速度。

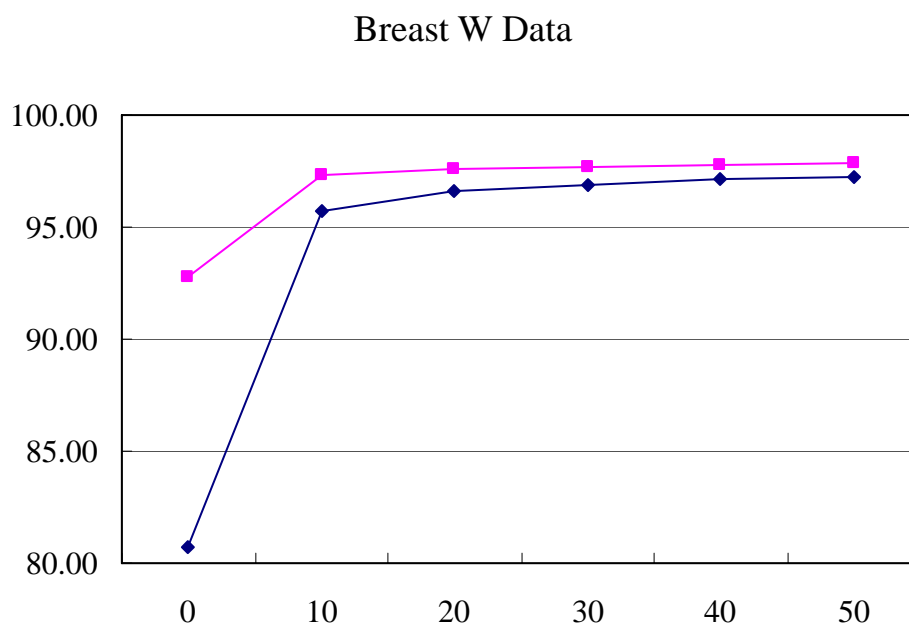


圖 5.1 Breast W Data 的收斂速度比較

圖中的水平軸代表計算時間，單位為秒 (second)，垂直軸代表分類正確率，單位為百分比 (%)。菱形所連成的曲線代表 hybrid fuzzy GBML 演算法的收斂曲線，而正方形所連成的曲線則代表改良式 fuzzy GBML 演算法的收斂曲線。

實驗結果顯示，改良式 fuzzy GBML 演算法不但收斂速度較快，初始 rule set 的分類正確率亦較高。以下為剩餘的六個 data sets 之收斂速度比較圖。

Diabetes Data

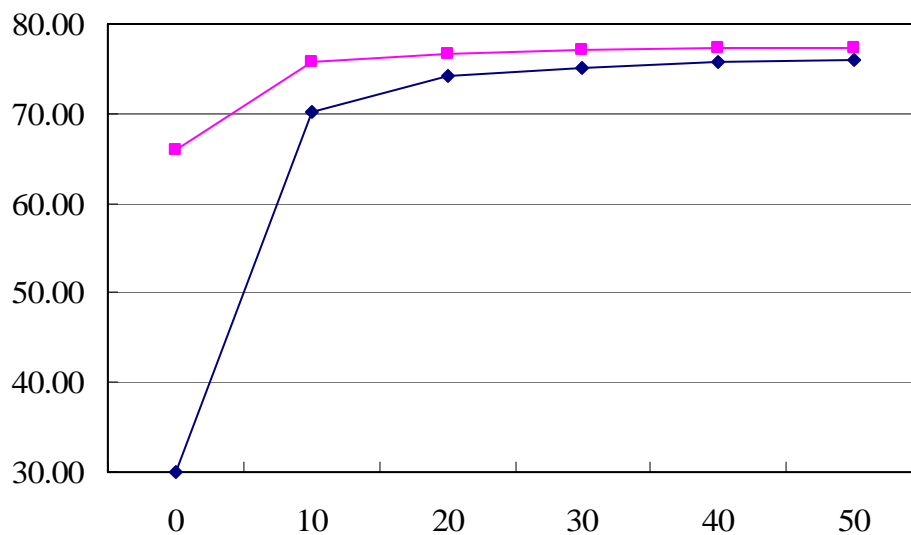


圖 5.2 Diabetes Data 的收斂速度比較

Glass Data

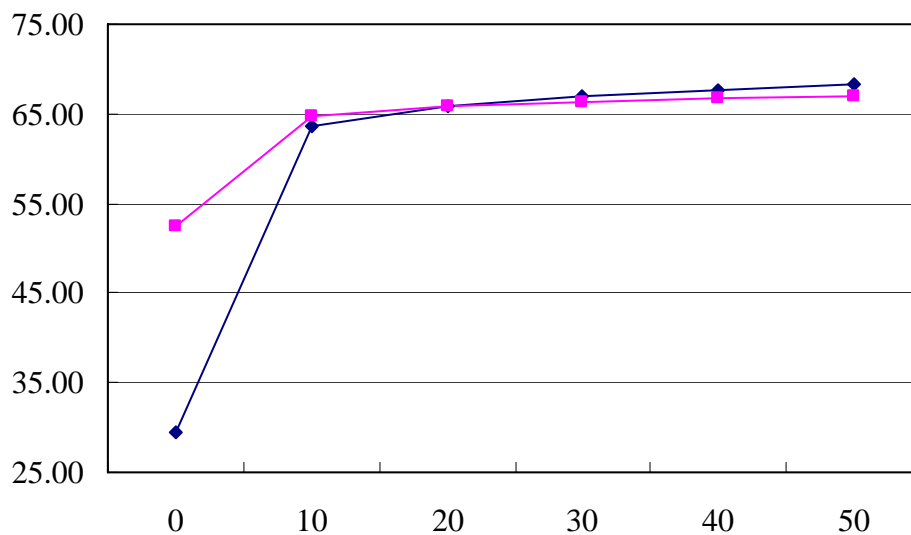


圖 5.3 Glass Data 的收斂速度比較

Heart C Data

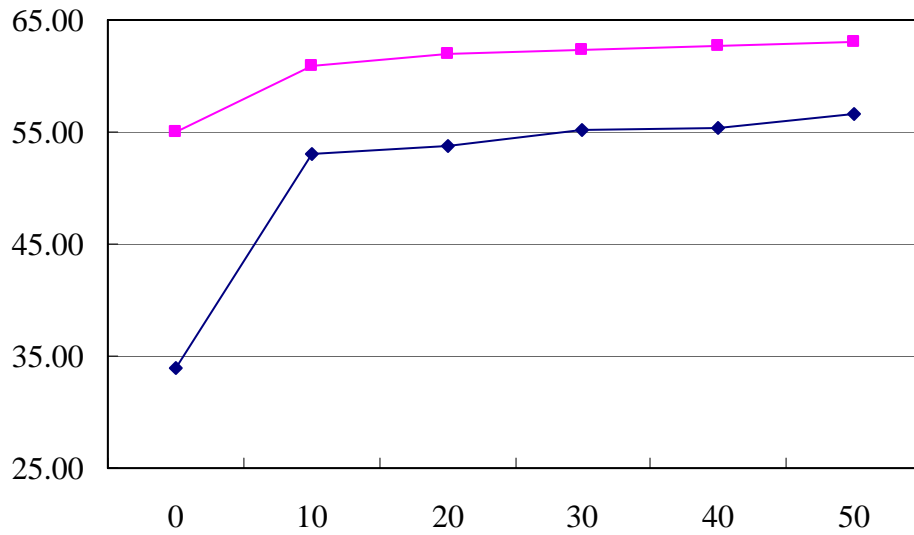


圖 5.4 Heart C Data 的收斂速度比較

Iris Data

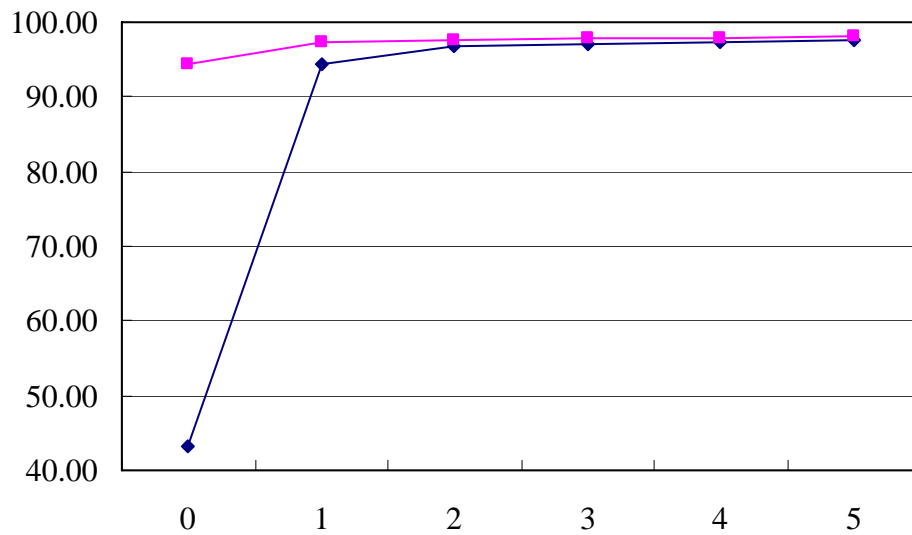


圖 5.5 Iris Data 的收斂速度比較

Sonar Data

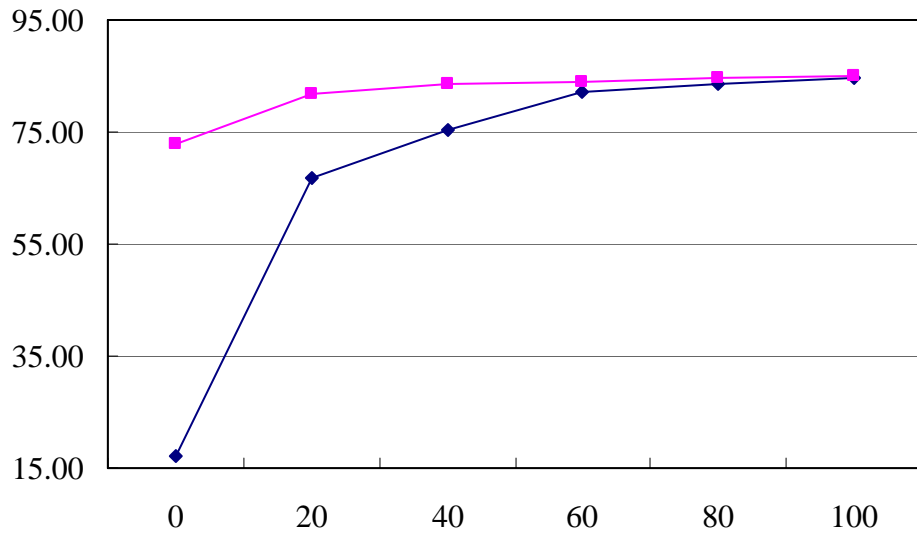


圖 5.6 Sonar Data 的收斂速度比較

Wine Data

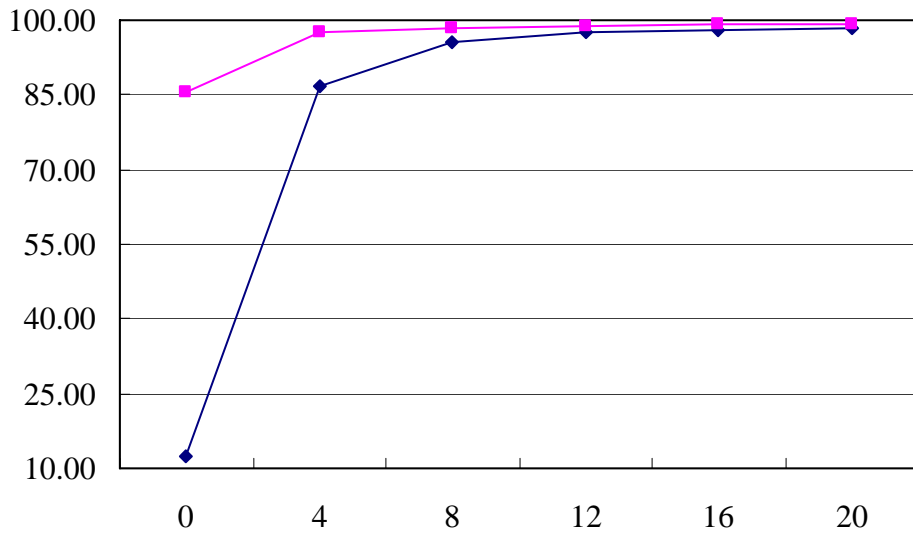


圖 5.7 Wine Data 的收斂速度比較

第六章 結論

本論文提出了一種改良式 fuzzy GBML 演算法來減少過去的 hybrid fuzzy GBML 演算法所需的計算時間、加快其收斂速度，且還提升了些微的分類正確率。

本論文首先說明了，當匹茲堡式演算法使用多結論類別的 fuzzy rules 與 weighted voting 的模糊推論方法（MW 模糊模型）時，計算速度將較其使用單一結論類別的 fuzzy rules 與 single winner rule 的模糊推論方法（SS 模糊模型）來的快速。接著說明了，當匹茲堡式演算法使用了 MW 模糊模型之後，不適合再使用密西根式演算法。最後，我們提出了一種改良式 fuzzy GBML 演算法，先以 MW 模糊模型取代了原本 hybrid fuzzy GBML 演算法所使用的 SS 模糊模型，再以一個 heuristic procedure 來取代原本的密西根式演算法。

在本論文及[17]中，均使用了許多 heuristic 的參數值。例如， P_c 、 P_m 、 N_{replace} 以及 $P_{\text{don't care}}$ 等等，如何依不同的 data set 找出最適合的參數值，使得改良式 fuzzy GBML 演算法所建構的 FRBCS 對每個 data set 的正確分類率均為最佳，我們認為這是未來另一個可以研究的方向。

參考文獻

- [1]. L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol.8, pp.338-353, 1965.
- [2]. L. X. Wang, *a Course in Fuzzy Systems and Control*. Prentice Hall, 1997.
- [3]. L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man, and Cybernetics*, vol.22, no.6, pp.1414-1427, 1992.
- [4]. J. S. R. Jang, "Self-learning fuzzy controllers based on temporal back propagation," *IEEE Transactions on Neural Networks*, vol.3, no.5, pp.714-723, 1992.
- [5]. M. Grabisch and F. Dispot, "A comparison of some methods of fuzzy classification on real data," in *Proc. 2nd Int. Conf. Fuzzy Logic Neural Networks*, Iizuka, Japan, July 1992, pp.659-662.
- [6]. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
- [7]. J. H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [8]. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.
- [9]. P. Thrift, "Fuzzy logic synthesis with genetic algorithms," in *Proc. 4th Int. Conf. Genetic Algorithms*, University of California, San Diego, July 13-16, 1991, pp.509-513.
- [10]. H. Nomura, I. Hayashi, and N. Wakami, "A self-tuning method of fuzzy reasoning by genetic algorithm," in *Proc. of the Int. Fuzzy Systems Intelligent Control Conf.*, Louisville, KY, Mar. 16-18, 1992, pp.236-245.

- [11].M. Valenzuela-Rendon, "The fuzzy classifier system: a classifier system for continuously varying variables," in *Proc. 4th Int. Conf. Genetic Algorithms*, University of California, San Diego, July 13-16, 1991, pp.346-353.
- [12].C. Z. Janikow, "A genetic algorithm for optimizing fuzzy decision trees," in *Proc. 6th Int. Conf. Genetic Algorithms*, University Pittsburgh, Pittsburgh, PA, July 15-19, 1995, pp.421-428.
- [13].Y. Yuan and H. Zhuang, "A genetic algorithm for generating fuzzy classification rules," *Fuzzy Sets System*, vol.84, no.1, pp.1-19, Nov. 1996.
- [14].S. Abe and M. -S. Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification," *IEEE Transactions on Fuzzy Systems*, vol.3, no.1, pp.18-28, Feb. 1995.
- [15].S. F. Smith, "A learning system based on genetic algorithms," Ph.D. dissertation, University Pittsburgh, Pittsburgh, PA, 1980.
- [16].L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artificial Intelligence*, vol.40, pp.235-282, Sep. 1989.
- [17].H. Ishibuchi, T. Yamamoto, and T. Nakashima, "Hybridization of fuzzy GBML approaches for pattern classification problems," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol.35, no.2, pp.359-365, Apr. 2005.
- [18].H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol.29, no.5, pp.601-618, Oct. 1999.
- [19].H. Ishibuchi and T. Nakashima, "Improving the performance of fuzzy classifier systems for pattern classification problems with

continuous attributes,” *IEEE Transactions on Industrial Electronics*, vol.46, no.6, pp.1057-1068, Dec. 1999.

[20].H. Ishibuchi and T. Yamamoto, “Rule weight specification in fuzzy rule-based classification systems,” *IEEE Transactions on Fuzzy Systems*, vol.13, no.4, pp.428-435, Aug. 2005.

[21].O. Cordon, F. Herrera, F. Hoffman, and L. Magdalena, *Genetic Fuzzy Systems*. Singapore: World Scientific, 2001.