

# 基於共同演化機制下區域性屬性加權的方法

李志文

元智大學資管系碩士生  
S966219@mail.yzu.edu.tw

李日崎

元智大學資管系碩士生  
S976208@mail.yzu.edu.tw

林志麟

元智大學資管系助理教授  
jun@saturn.yzu.edu.tw

## 摘要

資料中過多不重要的屬性常常會影響分類建模的準確度。針對此一問題，過去研究常利用屬性加權的方式來改善。然而，由於資料的多樣性，對資料全面採取相同的屬性加權仍常無法達到很好的分類效果。本研究以基因演算法透過共同演化(coevolution)的方式，產生出多組不同之區域屬性權重組合，以改善單一全面屬性權重組合無法有效提升分類效果的問題。在效能評估方面，本研究實驗結果顯示採取多組區域屬性加權所建立之分類器比未使用屬性加權之同類分類器有較好的分類準確度。

**關鍵詞：**共同演化、屬性加權、基因演算法、 $K$ 最近鄰分類法

## Abstract

Redundant attributes often make building an effective classifier difficult. Feature weighting is one way to resolve this problem. However, as the diversity of the data increases, all data using the same set of feature weighting may be inappropriate. This work proposes a local feature weighting approach, which uses a coevolution genetic algorithm to generate multiple sets of local feature weighting. Our experimental results show that the accuracy of a classifier can be improved with multiple sets of local feature weighting than without using any feature weighting.

**Keywords:** Coevolution, Feature weighting, Genetic algorithm,  $K$  nearest neighbor

## 1. 前言

在很多問題中，資料集相當複雜與多樣化，每筆資料都由很多屬性組合而成，但這些眾多的屬性卻並非全都是重要的屬性。文獻[6]指出若資料集有太多不重要或是多餘的屬性反而會降低分類的效果。因應此一情況，屬性篩選與屬性加權是兩種常用的解決方式。

屬性選擇(feature selection)的主要概念為只保留較重要的屬性作為後續分類建模之用。文獻[8]指出屬性選擇可達到以下三個目標。第一是可節省分類時計算的時間；第二是可提升分類的準確度；第三是可提高分類結果的可靠度。

屬性加權(feature weighting)的主要概念是將每個原始的屬性值皆乘上不同權重值，越重要的屬性給予越高的權重。吾人可將屬性選擇視為屬性加權的一個特例，因為未被選到的屬性就如同得到的權重為0一樣[18]。

屬性選擇和屬性加權皆屬於NP hard的問題，而啟發式演算法(heuristic algorithms)常用來解決這類問題，企圖以少量計算時間尋找一個適合的解。基因演算法(Genetic algorithms, GA)是目前最普遍用在屬性選擇及屬性加權的啟發式方法，文獻[1]、[10]、[13]、[15]、[18]、[19]皆是以GA為基礎的屬性選擇或屬性加權方法。

屬性選擇的用意是為了能夠降低屬性的個數，作法上針對每個屬性只有兩種可能情況發生，亦即保留該屬性或是剔除該屬性。因此，屬性選擇可視為二元權重值的作法(0代表剔除屬性，1代表保留屬性)。文獻[17]證明使用連續數值的權重值比二元數值的權重值更能提升分類結果。換言之，屬性加權相較於屬性選擇是更高層次的機制，因為屬性加權不單單只是選擇或不選某個屬性，而是更進一步依照屬性的重要性給予權重，因此可對屬性提供較豐富的解釋力。

雖然屬性加權的方式能有效提升分類效果，但文獻[7]證明，在做屬性加權時，若使用過多連續數值的權重值，很容易造成過度訓練(overfitting)的現象，因此提出只使用三種權重值(即zero weight, middle weight及full weight)的概念，稱為3FW。文獻[9]同樣採用3FW的概念，但更進一步以GA調整middle weight，提出adaptive-3FW，並提出modified crossover及modified mutation的基因操作方法，可有效解決過度訓練的現象，且比3FW有更好效果。

雖然GA可演化出一組合適的屬性權重組

合，但對於有些資料集，所有的資料全面只使用同一組屬性權重組合並沒辦法有效提升分類結果。文獻[2]、[3]提出，在解決分群問題(clustering)時，每筆資料合適的屬性權重不一定要相同，允許每個群聚(cluster)使用各自適合的屬性權重組合，以產生較好的分群結果。為了透過GA同時演化出多組不同的屬性權重組合，文獻[12]描述了共同演化(coevolution)的架構，文獻[5]進一步以共同演化的概念，利用GA演化出多組不同屬性權重組合，並以K-means解決分群問題，證明使用多組屬性權重組合確實比只使用單一組屬性權重組合好。

本研究以基因演算法透過共同演化(coevolution)的方式，產生出多組不同之區域屬性權重組合，以改善單一全面屬性權重組合無法有效提升分類效果的問題。本文其餘章節規劃如下。第二節介紹相關研究，包括：共同演化的概念，adaptive-3FW的基因編碼方式及基因操作機制，及描述本研究將如何使用K-means和KNN；第三節介紹本研究的架構與步驟；第四節說明實驗數據和結果；第五節總結並討論未來的研究方向。

## 2. 相關研究探討

### 2.1 共同演化

在說明共同演化的概念前，先描述演化式演算法(evolutionary algorithms, EAs)，它以模擬自然界的演化機制解決複雜的問題。演化式演算法的隨機性及仿效自然界的機制有別於傳統的演算法，並有強大的解題能力可突破許多演算法設計上的障礙。

GA是演化式演算法中最廣為使用的。GA完整表現出自然界的演化機制，其演化的過程從隨機產生初始母體(initial population)開始進行，整個母體是由多條染色體(chromosome)組成，每條染色體又是由固定數目的基因(gene)所構成。每條染色體可經由適應函數(fitness function)評估其適應值(fitness value)，接著透過一系列基因操作，包括交配(crossover)、突變(mutation)及複製(reproduction)，保留適應值較高的染色體至下一代，即完成一次的迭代。當滿足迭代次數或達到終止條件便可獲得GA的最佳解。

一般而言，共同演化演算法可分為兩大類，一個為競爭式共同演化(competitive

coevolution)，另一個為協同式共同演化(cooperative coevolution)，本研究採用後者。共同演化是模擬自然界中各個群體彼此共生的概念，文獻[5]描述了協同式共同演化的架構，整個共同演化的模型中包含了數個母體，每個母體獨立執行各自的基因操作，產生各自的區域最佳解(local optimal solution)。每個母體經過一次的迭代後，會針對全部母體評估出一個全域最佳解(global optimal solution)。

### 2.2 Adaptive-3FW

過去有很多文獻都以GA的方式演化各屬性的權重值，但過多連續數值的權重值容易造成過度訓練的現象。文獻[9]提出採用adaptive-3FW的GA，限制整條染色體只能使用三種權重值 0(zero weight)，1(full weight)及M(middle weight)，以避免使用過多權重值所造成的過度訓練現象，其中zero weight和full weight分別代表剔除和保留某個屬性，使得adaptive-3FW可同時包含屬性選擇和屬性加權的機制。以下為說明adaptive-3FW的編碼方式並描述其modified crossover及modified mutation的操作方式。

#### 2.2.1 Encoding

在adaptive-3FW的GA中，每條染色體代表一種屬性權重組合，假設某個資料集有d個屬性，那麼每條染色體將會由(d+1)個基因組成，最後一個基因位置紀錄每條染色體的M值，是一個介於0~1的實數。剩下的d個基因只能使用三種值0、1及M，其中M是根據染色體最後一個基因位置紀錄的M值變動，如圖1所示：

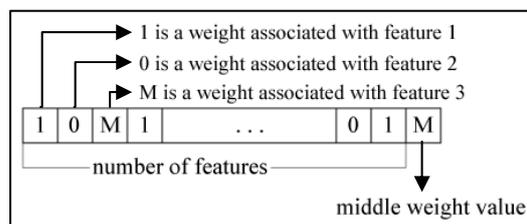


圖 1 adaptive-3FW encoding 方式

資料來源：文獻[9]

#### 2.2.2 Modified crossover

傳統單點的 crossover 方式並不適用在adaptive-3FW的GA上，原因是傳統單點的crossover應用在二元編碼的染色體上，可能的情況為將某個基因位置的值從0變成1，或1變成0，但此種方式可能會造成屬性權重太劇

烈的變化。此外 adaptive-3FW 規定整條染色體的  $M$  值是相同的，故傳統單點的 crossover 方式並不適用。

文獻[9]提出 modified crossover，可應用在 adaptive-3FW 的 GA 上。假設兩條要做 crossover 的染色體，其  $M$  值分別是  $M_1$  和  $M_2$ ，根據公式(1)重新產生新的  $M$  值  $N_1$  和  $N_2$ 。

$$(1) \begin{cases} N_1 = \alpha M_1 + (1 - \alpha) M_2 \\ N_2 = \alpha M_2 + (1 - \alpha) M_1 \end{cases}$$

$\alpha$  是介於 0~1 的隨機值，接著定義  $A_1, A_2, A_3, A_4, B_1, B_2, B_3, B_4$  如下：

$$A_1 = \begin{cases} N_1, M_2 \geq 0.5 \\ 0, M_2 < 0.5 \end{cases}$$

$$A_2 = \begin{cases} N_1, M_1 \geq 0.5 \\ 0, M_1 < 0.5 \end{cases}$$

$$A_3 = \begin{cases} 1, M_1 \geq 0.5 \\ N_1, M_1 < 0.5 \end{cases}$$

$$A_4 = \begin{cases} 1, M_2 \geq 0.5 \\ N_1, M_2 < 0.5 \end{cases}$$

$$B_1 = \begin{cases} N_2, M_2 \geq 0.5 \\ 0, M_2 < 0.5 \end{cases}$$

$$B_2 = \begin{cases} N_2, M_1 \geq 0.5 \\ 0, M_1 < 0.5 \end{cases}$$

$$B_3 = \begin{cases} 1, M_1 \geq 0.5 \\ N_2, M_1 < 0.5 \end{cases}$$

$$B_4 = \begin{cases} 1, M_2 \geq 0.5 \\ N_2, M_2 < 0.5 \end{cases}$$

最後，母代(parent)基因根據下列規則產生子代(offspring)基因：

$$\begin{cases} \text{Parent gene1} = 0 \\ \text{Parent gene2} = 0 \end{cases} \rightarrow \begin{cases} \text{Offsprin gene1} = 0 \\ \text{Offsprin gene2} = 0 \end{cases}$$

$$\begin{cases} \text{Parent gene1} = 0 \\ \text{Parent gene2} = M_2 \end{cases} \rightarrow \begin{cases} \text{Offsprin gene1} = A_1 \\ \text{Offsprin gene2} = B_1 \end{cases}$$

$$\begin{cases} \text{Parent gene1} = 0 \\ \text{Parent gene2} = 1 \end{cases} \rightarrow \begin{cases} \text{Offsprin gene1} = N_1 \\ \text{Offsprin gene2} = N_2 \end{cases}$$

$$\begin{cases} \text{Parent gene1} = M_1 \\ \text{Parent gene2} = 0 \end{cases} \rightarrow \begin{cases} \text{Offsprin gene1} = A_2 \\ \text{Offsprin gene2} = B_2 \end{cases}$$

$$\begin{cases} \text{Parent gene1} = M_1 \\ \text{Parent gene2} = M_2 \end{cases} \rightarrow \begin{cases} \text{Offsprin gene1} = N_1 \\ \text{Offsprin gene2} = N_2 \end{cases}$$

$$\begin{cases} \text{Parent gene1} = M_1 \\ \text{Parent gene2} = 1 \end{cases} \rightarrow \begin{cases} \text{Offsprin gene1} = A_3 \\ \text{Offsprin gene2} = B_3 \end{cases}$$

$$\begin{cases} \text{Parent gene1} = 1 \\ \text{Parent gene2} = 0 \end{cases} \rightarrow \begin{cases} \text{Offsprin gene1} = N_1 \\ \text{Offsprin gene2} = N_2 \end{cases}$$

$$\begin{cases} \text{Parent gene1} = 1 \\ \text{Parent gene2} = M_2 \end{cases} \rightarrow \begin{cases} \text{Offsprin gene1} = A_4 \\ \text{Offsprin gene2} = B_4 \end{cases}$$

$$\begin{cases} \text{Parent gene1} = 1 \\ \text{Parent gene2} = 1 \end{cases} \rightarrow \begin{cases} \text{Offsprin gene1} = 1 \\ \text{Offsprin gene2} = 1 \end{cases}$$

### 2.2.3 Modified mutation

文獻[9]也同樣提出 modified mutation，以應用在 adaptive-3FW 的 GA 上。此突變操作主要包含兩個步驟。第一個步驟為一條要做 mutation 的染色體其染色體中最後一個基因所記錄的  $M$  值，根據突變機率有機會突變成一個新的介於 0~1 的實數  $M'$ 。若突變成  $M'$ ，則整條染色體的  $M$  值也會跟著變為  $M'$ 。第二個步驟為此染色體的每個基因根據突變機率都有機會發生突變，若某個基因發生突變則會在三種權重值 0(zero weight)、1(full weight)及  $M$ (middle weight) 中隨機選一種當作此基因位置的值。

### 2.3 K-means 分群法

K-means 分群法(K-means clustering)是目前常用的分群方法，其目標是對資料的屬性值進行相似程度的計算，並將具有某些共同特性(即較接近)的資料放置到同一群集中。K-means 分群演算法步驟如下：

步驟一：根據預先給定的  $K$  值，隨機產生  $K$  個起始的群聚中心(Cluster centers)。

步驟二：假設某資料集共有  $n$  筆資料，每筆資料表示成  $\vec{X}_i = (X_i^1, X_i^2, \dots, X_i^d)$ ，其中  $X_i^1, X_i^2, \dots, X_i^d$  表示  $\vec{X}_i$  的  $d$  個屬性值，接著計算每筆資料與各群聚中心  $Z_l (1 \leq l \leq K)$  的歐幾里得距離(Euclidean distance)，公式如(2)所示

$$(2) \quad d(\vec{X}_i, Z_l) = \left( \sum_{j=1}^d |X_i^j - Z_l^j|^2 \right)$$

每筆資料以距離最近(即相似度最高)的群聚中心，將資料歸屬於此群聚。

步驟三：根據步驟二的分群結果，重新計算  $K$  個群聚內所有資料的平均值，找出新的群聚中心。

步驟四：重複步驟二及步驟三，直到每筆資料都不再變動其所屬群聚，即完成分群。

傳統K-means在使用上是將各個屬性的權重視為相同。文獻[2]提出每個群聚使用各自適合的屬性權重組合的概念。本研究也同樣採用種方式，在計算每筆資料和群聚中心的距離時，必須考慮個別群聚有其各自的屬性權重組合  $W_l (1 \leq l \leq K)$ ，因此將上述公式(2)改寫成下列公式(3)：

$$(3) \quad d(\bar{X}_l, Z_l) = \left( \sum_{j=1}^d W_l^d \times |X_l^j - Z_l^j|^2 \right)^{1/2}$$

### 2.4 K 最近鄰分類法

本研究的分類器採用K最近鄰分類法(K-nearest neighbor, KNN)，KNN是屬於案例學習(instance-based learning)中的一種方法。由於KNN使用上的簡易性使得它被廣泛應用在各個領域，文獻[1]、[4]、[11]、[14]、[16]在做屬性加權時，皆以KNN作為分類器。KNN的基本精神在於假設擁有相同特徵的資料，在其特徵所形成的特徵空間中會聚集在一起。在分類問題中，對於一筆未知類別的測試資料，可藉由距離它最近的K個的訓練資料進行投票，票數最高的類別即是此未知類別資料的分類結果。

KNN 是一種以比較各資料間相似程度為基礎的方法，在此簡單描述 KNN 的流程及其常使用的相似程度計算方式：歐幾里得距離(Euclidean distance)。假設有一組訓練資料  $\{(\bar{X}_1, Y_1), \dots, (\bar{X}_n, Y_n)\}$  及一筆未知類別的待測資料  $\bar{X}$ ，其中  $\bar{X}_i = (X_i^1, X_i^2, \dots, X_i^d)$ ， $X_i^1, X_i^2, \dots, X_i^d$  表示  $\bar{X}_i$  的d個屬性值， $Y_i$  為其所屬類別。KNN 的第一步要先計算  $\bar{X}$  和每筆訓練資料的距離  $d(\bar{X}, \bar{X}_i)$ ，下列公式(4)為歐幾里得距離的計算公式：

$$(4) \quad d(\bar{X}, \bar{X}_i) = \left( \sum_{j=1}^d |X^j - X_i^j|^2 \right)^{1/2}$$

接著根據計算出的距離，找出與  $\bar{X}$  最接近的k個鄰居  $\bar{X}_{(1)}, \dots, \bar{X}_{(k)}$ ，並以k個鄰居相對應的類別  $Y_{(1)}, \dots, Y_{(k)}$  進行投票，以最多票的類別決定  $\bar{X}$

的類別。

目前在分類的研究中最常採用 k 次交叉驗證法(k-folds cross-validation)評估正確率。作法為將資料集切割成 k 個子資料集，1 個子資料集被保留作為驗證模型用，其他 k-1 個子資料集用來訓練。交叉驗證必須重複 k 次，每個子資料集皆驗證一次，最終的正確率為 k 次的平均結果。此評估方法的目的是重複運用隨機產生的子資料集進行訓練和驗證，可得到較客觀的正確率，本研究實驗部分採用 ten-folds cross-validation。

## 3. 研究方法

### 3.1 方法架構

本研究提出同時使用多組屬性權重組合的屬性加權方法。作法會先將訓練資料經過本研究提出的 Coevolution Multiple Feature Weighting(CMFW)方法，產生出多個屬性權重組合都不同的群聚，稱為 global feature weighting，每筆測試資料欲做分類時，會先計算和各群聚中心的距離，決定此筆測試資料屬於那個群聚，再以該組屬性權重組合進行KNN分類。詳細研究架構圖如圖 2所示：

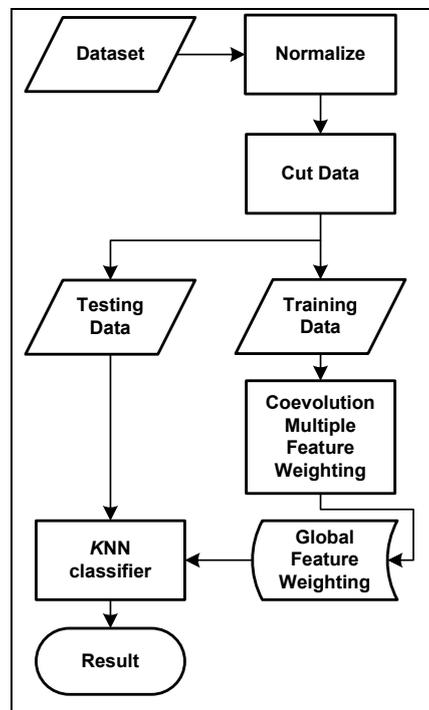


圖 2 研究架構圖

資料來源：本研究

### 3.2 CMFW

CMFW 方法結合了文獻[2]提出的 local attribute weighting 的 K-means 及文獻[5]提出的將多組屬性權重共同演化的概念。屬性權重的演化採用文獻[9]提出的 adaptive-3FW 的 GA，再以 KNN 作為此方法的分類器。接下來各小節描述各步驟，圖 3 為 CMFW 的流程圖：

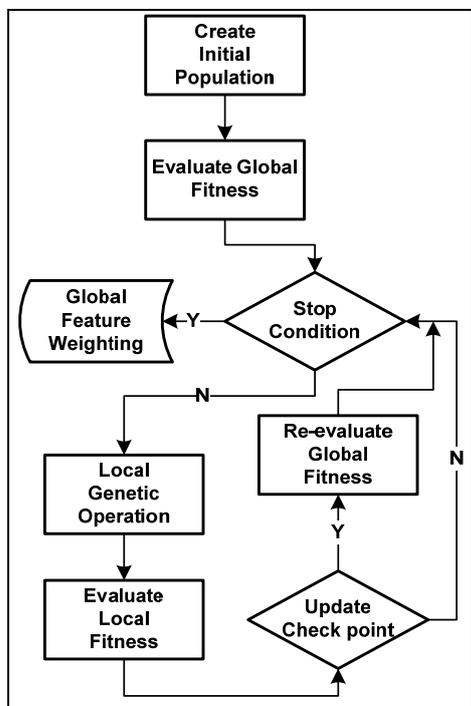


圖 3 CMFW 流程圖

資料來源：本研究

### 3.2.1 Initial population 的產生

CMFW 採用和文獻[5]相同的協同式共同演化的架構，整個共同演化的模型中包含了數個母體，每個母體獨立進行演化。每一個母體代表一個群聚，母體的數目  $K$  是由使用者自訂，即是使用者自行決定要演化出多少個不同屬性權重組合的群聚。

因為每個母體執行各自獨立且相同的基因操作，在此先介紹單一母體的染色體結構。假設每個母體皆有  $P$  條染色體，染色體  $(ind_{i,k})$  表示如下：

$$ind_{i,k} = [(w_1^{i,k}, \dots, w_m^{i,k}, M^{i,k}), (c_1^{i,k}, \dots, c_m^{i,k})]$$

每條染色體是由兩個部分構成，屬性權重部分  $(w_1^{i,k}, \dots, w_m^{i,k}, M^{i,k})$  及群聚起始中心部分  $(c_1^{i,k}, \dots, c_m^{i,k})$ 。其中  $k$  表示某個群聚， $m$  代表屬性個數， $M$  代表 adaptive-3FW 的  $M$  值。

### 3.2.2 Global fitness 計算

在評估 global fitness 時首先要從各母體中各取一條染色體，合併的成一個具有多組屬性權重組合的解，稱為 global solution  $(S_g)$ ，其結構如下所示：

$$S_g = [(ind_{i,1}), (ind_{i,2}), \dots, (ind_{i,K})]$$

將  $K$  個母體中的各條染色體，分別組合成  $P^K$  組擁有  $K$  個不同屬性權重組合及起始群聚中心的  $S_g$ 。

接著開始評估 global fitness，作法為將訓練資料分成 10-folds，其中 9-folds 根據某個  $S_g$  中  $K$  個不同屬性權重組合及起始群聚中心，以 local attribute weighting 的 K-means 將資料分成  $K$  個群聚。最後將剩下的 1-fold 當成驗證資料 (validation data)，每筆資料以距離最近的群聚中心，將資料歸屬於此群聚。並以該群聚的屬性權重組合進行 KNN，每個 fold 輪流當一次驗證資料，將 10 次結果取平均當作此  $S_g$  的適應值。取  $P^K$  組  $S_g$  中最好者作為目前的最佳解  $B_g$ 。

由於共有  $P^K$  組  $S_g$  需要評估，計算上相當耗時，因此除了初始時需要評估一次 global fitness 外，接下來每進行  $q$  次的迭代才又再一次評估 global fitness，並更新目前最佳解  $B_g$ ，此步驟即是 CMFW 流程圖中 re-evaluate global fitness。

### 3.2.3 Local genetic operation

協同式共同演化的架構中，每個母體是彼此獨立的，分別執行各自的基因操作。CMFW 中每條染色體是由兩個部分構成，屬性權重部分及群聚起始中心部分。屬性權重部分採用[9]提出的 adaptive-3FW 的 GA 進行演化。群聚起始中心部分採用傳統的單點交配及單點突變。

### 3.2.4 Local fitness 計算

評估 local fitness 作法類似評估 global fitness，以  $B_g$  為基礎，每條染色體根據所屬的母體取代  $B_g$  中相對應的群聚屬性權重組合及起始群聚中心，計算和  $B_g$  適應值的差距作為每條染色體的適應值。

## 4. 實驗設計

### 4.1.1 實驗設計與資料集

本實驗的主要目的是為了證明本研究提出的使用多組屬性權重組合的方法，會比未使用

屬性加權有較好的結果。作法上是以 UCI Cleveland Heart Disease Data set[20]進行實驗，將資料隨機分割成 10-folds，1-fold 被保留作為驗證分類結果的測試資料，其他 9-folds 為訓練資料。每個 fold 輪流當測試資料，最終的正確率為 10 次的平均結果。因為資料分割成 10-fold 是以隨機的方式，為了增加分類結果的公平性，本實驗重複上述步驟 10 次，每次分割資料的方式皆不同，稱為 ten-times ten-folds cross-validation。

本研究採用的分類器為 KNN，為了實驗在各種  $K$  值下本研究提出的方法皆有較好的表現，故分別實驗  $K = 1、3、5、7、9$  時的結果，並和原始未加屬性權重的 KNN 比較。實驗所使用的參數設定如下：

- I.  $K$ -means :  $K = 3$
- II. GA 迭代數 : 40
- III. GA 每個母體的染色體個數 : 5
- IV. GA 的交配機率 : 0.9
- V. GA 的突變機率 : 0.1

UCI Cleveland Heart Disease 資料集為兩類的分類問題，共有 296 筆資料，每筆資料有 13 個屬性。

#### 4.1.2 實驗結果

表 1 比較本研究提出的 CMFW 與傳統 KNN 在不同  $K$  值的分類準確度。結果發現在不同的  $K$  值下 CMFW 都有較好的分類結果。由於每一個  $K$  值都進行 ten-times ten-folds CV，表 1 中的 avg. 列為 ten-times ten-folds CV 的準確率，best 列為 10 次 ten-folds CV 最好的一次之準確率。

表 1 分類準確率  
資料來源：本研究

		CMFW	Original KNN
K=1	best	79.40%	77.66%
	avg.	78.19%	75.95%
K=3	best	86.47%	83.46%
	avg.	84.08%	81.33%
K=5	best	84.43%	82.42%
	avg.	83.53%	81.15%
K=7	best	84.83%	82.46%
	avg.	83.59%	81.67%
K=9	best	84.14%	81.47%
	avg.	82.97%	80.74%

## 5. 結論

本研究提出同時使用多組屬性權重組合的屬性加權方法(CMFW)，在共同演化的架構下結合 local attribute weighting 的  $K$ -means 及 adaptive-3FW 的 GA，再以 KNN 作為此方法的分類器。針對 UCI Cleveland Heart Disease Data set 進行實驗，實驗結果可看出使用多組屬性加權比未使用屬性加權有較好的結果。由於本方法在評估 global fitness 時，共有  $P^K$  組  $S_g$  需要評估，計算上相當耗時，連帶使得 GA 的迭代數和每個母體的染色體數目無法給定太大，可能會限制搜尋空間。因此未來研究方向可朝向解決評估 global fitness 較耗時的問題，增加搜尋空間，以進一步改善分類效果。

## 參考文獻

- [1] Beddoe, G. R., & Petrovic, S, "Selecting and weighting features using a genetic algorithm in a case-based reasoning approach to personnel rostering," *European Journal of Operational Research*, vol. 175, no. 2, pp. 649-671, 2006.
- [2] Chan, E. Y., Ching, W. K., Ng, M. K., & Huang, J. Z, "An optimization algorithm for clustering using weighted dissimilarity measures," *Pattern Recognition*, vol. 37, no. 5, pp. 943-952, 2004.
- [3] Domeniconi, C., Gunopulos, D., Ma, S., Yan, B., Al-Razgan, M., & Papadopoulos, D, "Locally adaptive metrics for clustering high dimensional data," *Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. 63-97, 2007.
- [4] Fernández, F., & Isasi, P, "Local feature weighting in nearest prototype classification," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 40-53, 2008.
- [5] Gançarski, P., & Blansché, A, "Darwinian, Lamarckian, and Baldwinian (co) evolutionary approaches for feature weighting in  $K$ -means-based algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 5, pp. 617-629, 2008.
- [6] Hong, J. -, & Cho, S. -, "Efficient huge-scale feature selection with speciated genetic algorithm," *Pattern Recognition Letters*, vol. 27, no. 2, pp. 143-150, 2006.
- [7] Kohavi, R., Langley, P., & Yung, Y, "The utility of feature weighting in

- nearest-neighbor algorithms,” *Proceedings of the Ninth European Conference of Machine Learning, Prague*, 1997.
- [8] Kudo, M., & Sklansky, J, “Comparison of algorithms that select features for pattern classifiers,” *Pattern Recognition*, vol. 33, no. 1, pp. 25-41, 2000.
- [9] Lee, H., Kim, E., & Park, M, “A genetic feature weighting scheme for pattern recognition,” *Integrated Computer-Aided Engineering*, vol. 14, no. 2, pp. 161-171, 2007.
- [10] Oh, I. -, Lee, J. -, & Moon, B. -, “Hybrid genetic algorithms for feature selection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1424-1437, 2004.
- [11] Paredes, R., & Vidal, E, “A class-dependent weighted dissimilarity measure for nearest neighbor classification problems,” *Pattern Recognition Letters*, vol. 21, no. 12, pp. 1027-1036, 2000.
- [12] Potter, M. A., & De Jong, K. A, “Cooperative coevolution: An architecture for evolving coadapted subcomponents,” *Evolutionary Computation*, vol. 8, no. 1, pp. 1-29, 2000.
- [13] Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L. A., & Jain, A. K, “Dimensionality reduction using genetic algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 164-171, 2000.
- [14] Tahir, M. A., Bouridane, A., & Kurugollu, F, “Simultaneous feature selection and feature weighting using hybrid tabu Search/K-nearest neighbor classifier,” *Pattern Recognition Letters*, vol. 28, no. 4, pp. 438-446, 2007.
- [15] Tan, T., Fu, X., Zhang, Y., & Bourgeois, A. G, “A genetic algorithm-based method for feature subset selection,” *Soft Computing*, vol. 12, no. 2, pp. 111-120, 2008.
- [16] Vivencio, D. P., Hruschka Jr., E. R., Do Carmo Nicoletti, M., Dos Santos, E. B., & Galvão, S. D. C. O, “Feature-weighted k-nearest neighbor classifier,” *IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pp. 481-485, 2007.
- [17] Wettschereck, D., Aha, D. W., & Mohri, T, “A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms,” *Artificial Intelligence Review*, vol. 11, no. 1-5, pp. 273-314, 1997.
- [18] Yang, J., & Honavar, V, “Feature subset selection using genetic algorithm,” *IEEE Intelligent Systems and their Applications*, vol. 13, no. 2, pp. 44-48, 1998.
- [19] Özşen, S., & Güneş, S, “Attribute weighting via genetic algorithms for attribute weighted artificial immune system (AWAIS) and its application to heart disease and liver disorders problems,” *Expert Systems with Applications*, vol. 36, no. 1, pp. 386-392, 2009.
- [20] UCI Repository of Machine Learning Databases, Dept. Information and Computer Sciences, Univ. California, Irvine, 1998. Available online at <http://www.ics.uci.edu/~mlern/MLRepositor y.html>