

Meta-PSO 最佳化參數用於分類問題

楊正宏
國立高雄應用科技大學
電子工程系
chyang
@cc.kuas.edu.tw

吳國銓
國立高雄應用科技大學
電子工程系
a092105140
@cc.kuas.edu.tw

莊麗月
義守大學
化學工程學系
chuang
@isu.edu.tw

摘要

特徵選取方法是近年來較熱門的分類前處理方法，分類問題中 K 最近鄰居法參數 K 影響分類效果甚大，且粒子族群最佳化部份參數會影響演算法之效率，因此在本文中，我們利用 Meta-粒子族群最佳化做參數最佳化，並挑選最佳的特徵子集合，以 K 最近鄰居法利用留一交互驗證做為評估標準。本文擷取五個分類問題做為實驗樣本，並以 weka 及單純的粒子族群最佳化做為實驗結果之比較。實驗結果顯示，本文所提出的方法能獲得較佳的分類正確率。

關鍵詞：特徵選取、Meta、粒子族群最佳化、K 最近鄰居法、留一交互驗證。

Abstract

Recently, the feature selection is an pre-processing before solving classification problems. In classification problems, parameters of K-nearest neighbor and particle swarm optimization affect the effect of algorithm. Therefore, we utilized Meta-particle swarm optimization to optimal, and explore better feature subset, which are using K-NN with leave one out cross-validation. In this paper we analyzed five problems of UCI data and compared with weka and simply particle swarm optimization. Experimental results showed that the proposed method had a higher accuracy than the other tested methods for classification problems.

Keywords: feature selection, particle swarm operation, Meta, K-nearest neighbor, leave one out cross-validation.

1. 前言

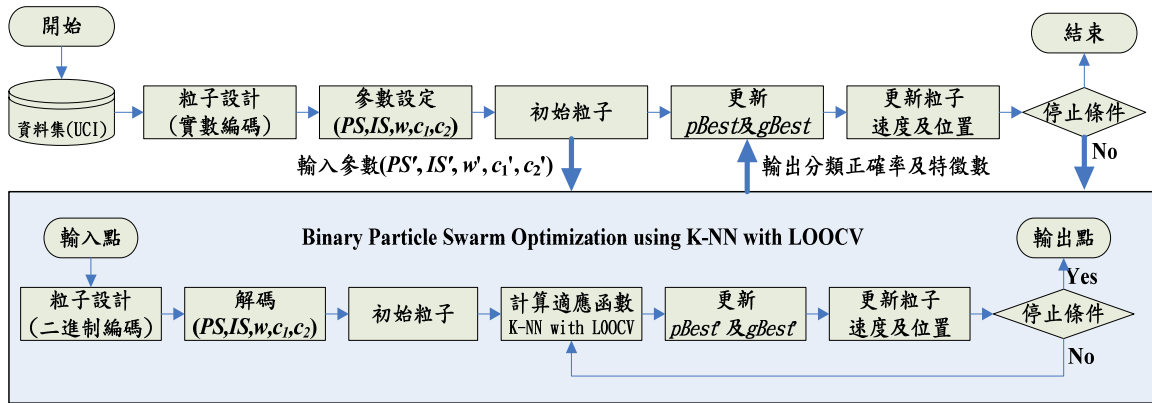
特徵選取方法是近年來較熱門的分類前處理方法。特徵選取的主要目的包括：1)在每筆

資料中，有些特徵維度會影響到分類的正確性或是多餘的，經特徵選取後反而可能會提升分類的正確性；2)於原始特徵集合 D 中挑選子集合 $d(D>d)$ 作分類辨識，在減少維度的情況下可節省運算時間，經過挑選的特徵作為機器學習、圖形辨識、調適性控制[14]等用途。而特徵選取方法大致上可分為兩種：1)Filter：主要是針對單一特徵進行重要性的測量，將所有重要性高的特徵組合成特徵子集合，較常用的方法有資訊增益、交互資訊、卡方檢定等方法[1, 9]。2)Wrapper：藉由持續的加入或刪除特徵，利用分類演算法中的目標函數評估特徵集合，再利用分類演算法的特性搜尋最佳的子集合。近年來，有許多不同類型的特徵選取方法被發展出來，包含窮舉演算法、分枝界限演算法[12, 17]、循序搜尋演算法[15]、基因演算法[14]等方法。3) Hybrid Filter & Wrapper：此方法先利用 Filter 挑選重要的特徵，經挑選後的特徵再進行 Wrapper 搜尋最佳的特徵集合，如文獻[13, 18, 22]。然而三種方法各有優缺點，詳細可參閱文獻中之說明。在本文中則使用 Wrapper 方法進行實驗。

粒子族群最佳化 (Particle Swarm Optimization, PSO) [7]近年來被廣泛應用在各領域，如旅行推銷員問題[20]、類神經網路[2]及貝氏分類器[4]等。然而為了使 PSO 能解離散問題，由原先實數編碼改為二進制編碼，稱二進制粒子族群最佳化(Binary Particle Swarm Optimization, BPSO) [6]。

本研究提出以 BPSO 做為特徵選取的方法，利用 K 最近鄰居法(K-Nearest Neighbor, K-NN)使用留一交互驗證法(Leave one out cross-validation, LOOCV)作為適應函數之評估標準。BPSO 有部份參數需要設定，包括慣性權重 w 、學習因子 c_1 及 c_2 ，文獻[10, 19]分別利用 PSO 最佳化 PSO 參數。再者適應函數 K-NN 中的參數 K 個鄰居極可能影響分類效果，因此在本研究提出以 Meta-PSO 做為參數最佳化。

圖 1 Meta-PSO 流程圖



2. 研究方法

2.1 粒子族群最佳化

PSO 最初概念源自鳥類和魚類族群之特性所發展出的一種最佳化演算法。在 PSO，每個粒子均為一個解(即鳥群裡的每隻鳥)，這些粒子和基因演算法中的染色體具有相同意義。PSO 和基因演算法最大差異則是在於演化方式，基因演算法主要是基於母代，藉由選擇、交配、突變進行世代替換，而產生較佳的子代。不同於基因演算法，每個粒子在各自搜尋解的經驗中，個體最佳的經驗稱之為 $pBest$ ，而在所有粒子中最佳的經驗稱之為 $gBest$ 。根據這兩種經驗來決定自己的飛行速度及移動方向，而決定所在位置。在每一次的迭代中，粒子都必須根據 $pBest$ 及 $gBest$ 進行目前的位置及速度的更新。假設 N_{dim} 為問題的空間維度(即搜尋空間 $\mathcal{R}^{N_{dim}}$)， N_p 為粒子 $P = \{P_1, P_2, \dots, P_{N_p}\}$ 的數量，而每個粒子 $P_i = (x_i, v_i, pBest_i)$ ，其中包含搜尋空間之當前位置 ($x_i \in \mathcal{R}^{N_{dim}}$)、速度 ($v_i \in \mathcal{R}^{N_{dim}}$) 及個體最佳經驗 ($pBest_i \in \mathcal{R}^{N_{dim}}$)。為了避免過於混亂的搜尋，一般會將 x_i 限定在 $[x_{min}, x_{max}]$ 而 v_i 則為 $[-v_{max}, +v_{max}]$ 。其中文獻[16]將更新公式引進慣性權重 w 使粒子在搜尋全域最佳解及區域最佳解間取得平衡，公式如下所示：

$$v_i^{new} = w \cdot v_i^{old} + c_1 \cdot rand_1 \cdot (pBest - x_i^{old}) + c_2 \cdot rand_2 \cdot (gBest - x_i^{old}) \quad (1)$$

$$x_i^{new} = x_i^{old} + v_i^{new} \quad (2)$$

其中 v_i^{new} 代表粒子更新後速度， v_i^{old} 代表粒子更新前速度。 x_i^{new} 代表粒子更新後位置，

x_i^{old} 則代表粒子更新前位置。 $pBest - x_i^{old}$ 是粒子本身最佳值的位置和粒子當前所在位置間的距離， $gBest - x_i^{old}$ 是至目前迭代為止之最佳值的位置和粒子當前所在位置間的距離 ($gBest_i \in \mathcal{R}^{N_{dim}}$)。慣性權重 w ，一般介於 0.4 到 0.9 之間。 c_1 和 c_2 為學習因子，其範圍為 0~4，一般均設定為 2。 $rand_1$ 及 $rand_2$ 均為介於 0 至 1 之間均勻分佈的隨機變數。

BPSO 之粒子設計為 $x_i \in \{0, 1\}$ ，而速度 v_i 之更新與方程式(1)相同。在此將速度 v_i 視為一個機率值介於 0 到 1 之間，利用速度 v_i 來判斷粒子的位置是否改變，而使 v_i 成為一個機率值可套入 Sigmoid 函數，如方程式(3)所示：

$$S(V_{id}^{new}) = \frac{1}{1 + e^{-V_{id}^{new}}} \quad (3)$$

其中若使 $v_{max} = 6$ 可使 v_i 介於 0.9975 至 0.0025 之間，再藉由隨機產生之亂數 $rand_i$ 來判斷粒子位置是否改變，位置改變如方程式(4)所示：

$$\text{if } (rand_{id} < S(V_{id}^{new})) \\ \text{then } X_{id}^{new} = 1 \quad \text{else } X_{id}^{new} = 0 \quad (4)$$

2.2 Meta-PSO

"Meta"一詞出現於文獻[10, 19]中改良式 PSO，其改良方式為 PSO 對 PSO 本身做參數最佳化，因而稱之。Meta-PSO 尚未廣泛應用於各領域，主要原因可能是 Meta-PSO 需要比一般 PSO 較長的運算時間，然而在文獻[10]中利用 Meta-particle 尋找 PSO 幾個重要參數：慣性權重、學習因子、 v_{max} 等，[19] 除了作參數最佳化，另外還針對粒子搜尋解的方式之最佳

化，如星形拓撲(star topology)、環形拓撲(ring topology)及動態鄰近法(dynamic neighborhood)等，而本文主要則針對分類器做最佳化。

2.3 K 最近鄰居法

K 最近鄰居法(K-Nearest Neighbor, K-NN)是由 Fix 和 Hodges 在 1951 年所提出的[5]。在 K-NN 的訓練資料中，每一個資料點都依照本身的特徵維度被定義在一個 D-維的空間中，而 K 所代表的是測試資料在 D-維空間裡所尋找的 K 個最近的鄰居，K-NN 的分類效果就是受到這 K 個鄰居的數量影響[5]，而鄰居的計算方式是根據歐幾里德距離(或是馬式距離)，利用此種方式針對測試資料與訓練資料進行相似性的測量，統計 K 個鄰居中各類別出現的頻率，將測試資料的類別定義為出現頻率最高的類別，藉此達到分類的目的。

2.4 留一交互驗證法

本研究利用 BPSO 找出特徵子集合，以 K-NN 做為評估標準，評估方式採用一交互驗證法以求得分類正確率。在每一次的評估過程中，在所有資料中挑選一筆資料作為 K-NN 測試資料，其餘則為 K-NN 的訓練資料，藉此訓練並獲得測試資料的類別，經過全部資料都測試之後，即可得知分類的正確率。

2.5 Meta-PSO with K-NN

藉由前文提及之文獻研究，將 BPSO 應用於分類問題，並利用 Meta-PSO 對 BPSO 做參數最佳化，其中因 K-NN 中 K 參數對於不同的資料型態會有不同的分類效果，因此本研究對於 Meta-PSO 之編碼特別對 K-NN 之參數 K 做參數最佳化。其中適應函數為經過一次 BPSO 搜尋後之最佳解，而 BPSO 則是依 Meta-PSO 之參數作為搜尋設定，以 K-NN 利用 LOOCV 做為分類正確率之評估，Meta-PSO 之基本流程如圖 1 所示，以下將詳細介紹本研究方法。

Step 1) Meta-PSO 粒子設計及參數設定

首先考慮 BPSO 需最佳化之參數：慣性權重 w 、學習因子 c_1 及 c_2 ，K-NN 之參數 K ($K \in \{1, 3, 5\}$) 共四個參數，其中將參數 K 分為三個維度，因此 Meta-PSO 之粒子設定為 $x_i = (w, c_1, c_2, k_1, k_3, k_5)$ ， $N_{dim} = 6$ 。其 Meta-PSO 參數設定如表 1 所示。

表 1 Meta-PSO 之參數設定

參數	值
粒子數 N_p	20
粒子維度 N_{dim}	6
迭代次數 N_{iter}	30
搜尋範圍 (x_{min}, x_{max})	(0.0, 40.0)
速度之範圍 v_{max}	6.0
慣性權重 w	0.8
學習因子 c_1, c_2	2.0, 2.0

Step 2) Meta-PSO 初始粒子

隨機產生 N_p 個 N_{dim} 維粒子，其中初始位置 $x_i \in [x_{min}, x_{max}]$ 、初始速度 $v_i \in [0, 1]$ 。

Step 3) 計算 Meta-PSO 適應函數

利用 Meta-PSO 產生之參數，進行 BPSO 於 K-NN 之分類過程。經演化後，得分類正確率及特徵子集合。

Step 3-1)解碼

首先將 Meta-PSO 之搜尋範圍 $[x_{min}, x_{max}]$ 轉換成 $[0, 1]$ ，如此即可對應成參數的搜尋範圍 $[p_{min}, p_{max}]$ 。解碼公式如方程式(5)所示。

$$\phi(p, p_{min}, p_{max} = p_{min} + p(p_{max} - p_{min})) \quad (5)$$

表 2 BPSO 之參數設定

參數	值
粒子數 N'_p	20
粒子維度 N'_{dim}	依資料而定
迭代次數 N'_{iter}	30
速度之範圍 v'_{max}	6.0
慣性權重 w'	$\phi(\frac{x_{i1}}{x_{max}}, 0.8, 1.2)$
學習因子 c'_1	$\phi(\frac{x_{i2}}{x_{max}}, 0.0, 4.0)$
學習因子 c'_2	$\phi(\frac{x_{i3}}{x_{max}}, 0.0, 4.0)$
K (K 最近鄰居)	$\begin{cases} 1, & x_{i5} = \max(x_{i5}, x_{i6}, x_{i7}) \\ 3, & x_{i6} = \max(x_{i5}, x_{i6}, x_{i7}) \\ 5, & x_{i7} = \max(x_{i5}, x_{i6}, x_{i7}) \end{cases}$

Step 3-2)BPSO 粒子設計及參數設定

基於特徵選取的概念，粒子之編碼為 $x'_i = (F_1, F_2, \dots, F_D)$ ， $F_i \in \{0, 1\}$ 、D 為分類問題之特徵維度。其中 F 代表一個特徵以二進制表示，1 代表該特徵被選取，反之則不選

取。另外 BPSO 代號以撇號(''), prime)與之區隔，參數設定如表 2 所示。

Step 3-3)BPSO 初始粒子

隨機產生 N_p 個 N_{dim} 維粒子，其中初始位置 $x_i' \in \{0, 1\}$ 、初始速度 $v_i' \in \{0, 1\}$ 。

Step 3-4)計算 BPSO 適應函數

在 BPSO 當中，以 K-NN 使用 LOOCV 的方式作為特徵集合之評估標準，依此作為 BPSO 之適應函數值。

Step 3-5)更新 $pBest'$ 及 $gBest'$

粒子當前位置之適應值，與本身及群體最佳值比較，若當前解比本身最佳解好，則當前解為 $pBest'$ ；若此 $pBest'$ 為所有群體最佳解，則當前解為 $gBest'$ 。

Step 3-6)更新粒子速度及位置

以方程式(1)、(3)及(4)做為更新依據。

Step 3-7)停止條件

當迭代次數 N_{iter}' 達設定次數則停止，否則跳到 Step 3-3，直到符合停止條件。當符合停止條件時，BPSO 得一組正確率及特徵子集合。

Step 4) 更新 $pBest$ 及 $gBest$

其概念同 Step 3-4。

Step 5) 更新粒子速度及位置

以方程式(1)、(2)做為更新依據。

Step 6) 停止條件

當迭代次數 N_{iter} 達設定次數則停止，否則跳到 Step 3，直到符合停止條件。當符合停止條件時，Meta-PSO 得一組最佳 BPSO 參數、分類正確率及特徵子集合。

3. 結果與討論

3.1 資料格式

本文中的資料集取自 UCI Repository [11]，其基本格式如表 3 所示。依文獻[8]知，分類問題大小的分類為： $0 < D \leq 19$ 為小資料

量的分類問題、 $20 \leq D \leq 49$ 為中資料量的分類問題、 $50 \leq D$ 為大資料量的分類問題(其中 D 代表特徵數)。因此本研究使用的資料集包括 *Sonar* 為大資料量的分類問題，及 *wdbc*、*Ionosphere* 為中資料量的分類問題，其餘的 *vehicle*、*heart* 及 *glass* 為小資料量的資料集。

表 3 資料基本格式

Dataset	samples	classes	features
<i>glass</i>	213	7	10
<i>heart</i>	270	2	13
<i>wdbc</i>	569	2	30
<i>ionosphere</i>	351	2	34
<i>sonar</i>	208	2	60

3.2 實驗結果分析與比較

為瞭解參數對於分類結果之影響，以下分別針對 K-NN 參數 K 之設定間的差異，以及各特徵選取的方法比較等。以 GA 之實驗數據取自於 Weka[21]。

未經過特徵選取之實驗結果如表 4 所示，我們可以發現，不同的資料集在 K-NN 不同的設定下，均有不同的結果。如 1-NN 的分類下，*ionosphere* 及 *sonar* 有較佳的分類效果；3-NN 在 *glass*、*vehicle* 及 *wdbc* 有較佳的分類效果；5-NN 則是 *heart* 有較好的效果。結果顯示，K-NN 得依資料特性，而決定 K 的大小，其原因可能會因資料類與類間之分佈過散或集中，而使效果有所差異。在特徵選取的部份，表 5 之結果顯示，有效的特徵選取方法能獲得較佳的分類效果，其原因是經過特徵選取後，將多餘的或會影響分類效果的特徵刪除，使分類正確率提高。

表 4 K-NN 用於不同資料之結果

Dataset	All Features		
	1-NN	3-NN	5-NN
<i>glass</i>	91.08	<u>91.55</u>	88.26
<i>heart</i>	75.76	79.26	<u>80.00</u>
<i>wdbc</i>	95.25	<u>97.01</u>	96.66
<i>ionosphere</i>	<u>86.89</u>	86.04	85.47
<i>sonar</i>	<u>87.50</u>	83.65	82.21
Average	87.30	<u>87.50</u>	86.52

Meta-PSO 之實驗結果如表 5 所示,本文在 *wdbc* 資料集之分類結果不如 GA 好。由表 5 知,本研究方法找出 *wdbc* 之特徵子集合為 2 而 GA 為 15,由此可知本文方法在 *wdbc* 資料集,刪除了重要特徵而影響了分類效果,其原因可能是 Meta-PSO 在參數最佳化的過程中,找到的 c_2 為 3.4 左右而 c_1 為 0.3 左右,如此將使 BPSO 快速偏向 *gBest* 而使粒子快速收斂,因而陷入區域最佳解。而其餘的資料集均能找到比較好的結果,因此本研究方法能有效找到最佳特徵子集合於分類問題。

3.3 參數

在 PSO 參數的部份, [3, 16]等文獻均對所有參數有所實驗及討論。 w 、 c_1 及 c_2 分別影響當前速度,個體最佳經驗及群體最佳經驗,以模擬鳥類飛行來說明,當設定愈高時,鳥類飛行的速度就愈快,其搜尋解的速度就愈快(收斂快),但卻容易掉入區域最佳解,反之設定愈小時,其飛行的速度就愈慢,雖然比較不易掉進區域最佳解,但搜尋解的速度卻愈慢(收斂慢),相對要找到全域最佳解可能要花費更長的時間。然而,這些都得視搜尋空間的大小而定,所以不同的問題應設定不同的參數來提升尋找全域最佳解的機會。在 K-NN 中, K 愈大對於類別統計上會更有機會找到較佳的分類效果,但若是資料分佈過散,則會造成類別統計上的誤差,而造成反效果。所以得依資料特性來決定 K 的大小,一般 K 設為 1、3、5。綜合上述,這些參數在不同的資料集中,會因不同的特性有不同的分類效果,因此參數最佳化是件必要的工作。

3.4 討論

PSO 及 K-NN 依不同的資料做參數最佳化是必要的,因此使用 Meta-PSO 做為搜尋方法,然而 Meta-PSO 較為人詬病的部份是時間複雜度。雖然如此,在一般的情況下,這些參數在實驗過程中,需不斷的嘗試及更改,在設定的過程中,非但浪費時間且在設定參數時往往依據以前文獻資料,然而本研究實驗結果顯示,在不同的資料裡利用不同的參數,才能比較容易找到問題的最佳解。因此,Meta-PSO 之時間複雜度仍可容忍的範圍。在 Meta-PSO 參數設定的部份,由於粒子設計為實數,因此我們利用文獻所建議的參數做為搜尋的設定。

4. 結論

本研究提出以 Meta-PSO 針對參數設定做智慧型搜尋,應用於分類問題。經測試五筆資料集,進行 GA、BPSO 及 Meta-PSO 分析比較,結果顯示本研究方法能有效找出較佳的特徵集合及分類正確率。我們也發現 K-NN 之參數 K 影響分類效果甚大,另一方面, BPSO 之參數設定則關係到收斂速度及落入區域解。

在未來的研究中,期望能夠將本研究方法應用於各種分類領域中。在特徵選取或參數最佳化方面,能將演算法效能提升、加以改進,獲得更佳的特徵子集合,以供往後研究者之研究與應用。

參考文獻

- [1] Battiti, R., "Using mutual information for selecting features in supervised neural net learning," *Neural Networks, Transactions on IEEE*, vol. 5, pp. 537-550, 1994.
- [2] Chatterjee, A., K. Pulasinghe, K. Watanabe, and K. Izumi, "A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems," *Industrial Electronics, Transactions on IEEE*, vol. 52, pp. 1478-1489, 2005.
- [3] Eberhart, R. C., and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," *Congress on Evolutionary Computation, 2000. Proceedings of the 2000*, 2000, pp. 84-88 vol.1.
- [4] Elom, S. C., A. F. Alex, and G. J. Colin, "A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set," *Proceedings of the 8th annual conference on Genetic and evolutionary computation Seattle, Washington, USA: ACM Press*, 2006.
- [5] Ghosh, A. K., P. Chaudhuri, and C. A. Murthy, "On visualization and aggregation of nearest neighbor classifiers," *Transactions on IEEE Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1592-1602, 2005.
- [6] Kennedy, J., and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *Systems, Man, and Cybernetics*, 1997. *Conference on 'Computational Cybernetics and Simulation'*, 1997 IEEE International, 1997, pp. 4104-4108 vol.5.
- [7] Kennedy, J., and R. Eberhart, "Particle

- swarm optimization," in Neural Networks, 1995. Proceedings., **Conference on IEEE International**, 1995, pp. 1942-1948 vol.4.
- [8] Kudo, M., and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," **Pattern Recognition**, vol. 33, pp. 25-41, 2000.
- [9] Li, Q., J.-H. Li, G.-S. Liu, and S.-H. Li, "A rough set-based hybrid feature selection method for topic-specific text filtering," **Conference on Machine Learning and Cybernetics, 2004. Proceedings of 2004 International**, 2004, pp. 1464-1468 vol.3.
- [10] Meissner, M., M. Schmuker, and G. Schneider, "Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training." **BMC Bioinformatics** vol. 7, 2006, p. 125.
- [11] Murphy, P. M., and D. W. Aha, "UCI Repository of Machine Learning Databases," **Technical Report, Department of Information and Computer Science, University of California, Irvine, Calif.**, 1994.
- [12] Narendra, P. M., and K. Fukunaga, "A Branch and Bound Algorithm for Feature Subset Selection," **Transactions on Computers**, vol. C-26, pp. 917-922, 1977.
- [13] Ni, B., and J. Liu, "A hybrid filter/wrapper gene selection method for microarray classification," **Conference on Machine Learning and Cybernetics, 2004. Proceedings of 2004 International**, 2004, pp. 2537-2542 vol.4.
- [14] Oh, I.-S., J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," **Transactions on IEEE Pattern Analysis and Machine Intelligence**, vol. 26, pp. 1424-1437, 2004.
- [15] Pudil, P., J. Novovicova, and J. Kittler, "Floating search methods in feature selection," **Pattern Recognition Letters**, vol. 15, pp. 1119-1125, 1994.
- [16] Shi, Y., and R. Eberhart, "A modified particle swarm optimizer," in Evolutionary Computation Proceedings, 1998. **Conference on IEEE World Congress on Computational Intelligence., The 1998 IEEE International**, 1998, pp. 69-73.
- [17] Somol, P., P. Pudil, and J. Kittler, "Fast branch & bound algorithms for optimal feature selection," **Transactions on Pattern Analysis and Machine Intelligence**, vol. 26, pp. 900-912, 2004.
- [18] Uncu, O., and I. B. Turksen, "A novel feature selection approach: Combining feature wrappers and filters," **Information Sciences**, vol. 177, pp. 449-466, 2007.
- [19] Veenhuis, C., "Advanced Meta-PSO," **Conference on Hybrid Intelligent Systems**, 2006. HIS '06. Sixth International, 2006, pp. 54-54.
- [20] Wang, K.-P., L. Huang, C.-G. Zhou, and W. Pang, "Particle swarm optimization for traveling salesman problem," **Conference on Machine Learning and Cybernetics, 2003 International**, 2003, pp. 1583-1585 Vol.3.
- [21] Witten, I. H., and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, **Morgan Kaufmann**, 2005.
- [22] Zhu, Z., Y.-S. Ong, and M. Dash, "Wrapper-Filter Feature Selection Algorithm Using a Memetic Framework," **Transactions on IEEE Systems, Man, and Cybernetics, Part B**, vol. 37, pp. 70-76, 2007.

表 5 Meta-PSO 分類結果

Name	D	GA		BPSO		Meta-PSO	
		d	1-NN	d	1NN	d	K-NN
glass	10	3	<u>98.59</u>	3	<u>98.59</u>	3	<u>98.59</u>
heart	13	7	81.30	8	81.48	6	<u>85.56</u>
wdbc	30	15	97.36	14	97.72	8	<u>98.42</u>
ionosphere	34	10	94.02	14	93.45	8	<u>95.73</u>
sonar	60	32	93.27	26	91.83	36	<u>97.60</u>
Average		92.91		92.61		<u>95.18</u>	