Neurocomputing 146 (2014) 95–103

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An improved differential evolution and its application to determining feature weights in similarity-based clustering

Chun-Ru Dong [a,b], Wing W.Y. Ng [a,*], Xi-Zhao Wang [b], Patrick P.K. Chan [a], Daniel S. Yeung [a]

[a] Machine Learning and Cybernetics Research Center, School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

[b] Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Computer Science, Hebei University, Baoding 071002, China

ARTICLE INFO

Article history:
Received 16 August 2013
Received in revised form
5 March 2014
Accepted 3 April 2014
Available online 15 July 2014

Keywords:
Similarity-based clustering
Feature weighting
Differential evolution
Dynamic differential evolution
Differential evolution strategy

ABSTRACT

In this work, we propose an optimization model to tune feature weights for improving performance of clustering via a minimization of uncertainty (fuzziness and non-specificity) of its similarity matrix among objects. To solve the proposed model efficiently, we propose an evolutionary search approach by integrating multiple strategies from both differential evolution and dynamic differential evolution. Then, the proposed method is applied to both weighted fuzzy c-means and weighted similarity-matrix-based transitive closure clustering. Experiments on 11 benchmarking databases show that the proposed method outperforms clustering methods without feature weighting and the feature weighting method based on gradient descent in terms of clustering performance evaluation indices and robustness.

1. Introduction

Cluster analysis is one of the important processes in many fields such as pattern recognition, data mining and image processing. Similarity-based clustering finds clusters of objects in a database by their pairwise similarities. In many cases, there is no clear cut of similarity and dissimilarity for two objects. Therefore, fuzzy clustering plays important role in cluster analysis [1,2]. Feature weighting and selection techniques are widely used in cluster analysis and unsupervised learning. According to the relationship between feature selection process and construction of learning model, feature selection techniques can be divided into three categories: filter, wrapper and embedded methods [3–5]. In contrast to selecting a subset of features, feature weighting methods assign a real-valued weight to each feature according its importance [2,6–12]. A proper assignment of feature weights could significantly improve the quality of clustering [11–12]. So, we propose a new optimization method of feature weighting for similarity-based clustering in this work.

In [2], Yeung et al. proposed a method to improve the performance of similarity-based clustering by feature weight learning. In their work, the traditional Euclidean distance used in similarity calculation was replaced by a weighted Euclidean distance. The weight is determined via a minimization of the fuzziness of the similarity matrix using the Gradient Descent (GD) technique. Later, the proposed feature weight learning scheme was applied to improve the clustering quality of fuzzy c-means clustering (FCM) [6]. Several other GD-based feature weighting methods are proposed in [7,12,13]. However, there exist some disadvantages of GD-based optimization algorithms. Firstly, for objective functions, it may converge to a local minimum with a great possibility. Secondly, the learning efficiency of GD is relatively low. Thirdly, the determination of parameters (e.g. learning rate and momentum) is difficult and significantly affects the final solution and the convergence efficiency [14].

The determination of feature weights can be formulated as an optimization problem in a bounded real space. The Differential Evolution (DE) has been proposed by Storn and Price in 1995 [15] and is extended to global optimization over continuous spaces in [16,17]. The DE is a population-based stochastic search approach which converges to the global optimal with a high probability. It requires few parameter tuning, is robust, easy to use and very suitable to parallel computation [17]. The DE attracts a lot of interests in applications and researches because it is suitable for optimization problems which are noisy, change over time and even discontinuous [18]. The jDE [19], the Opposition-based DE (ODE) [20], the SaDE [21], the DE with Global and Local

* Corresponding author.
E-mail addresses: dongcr@hbu.edu.cn (C.-R. Dong), wingng@ieee.org (W.W.Y. Ng).

http://dx.doi.org/10.1016/j.neucom.2014.04.065
0925-2312/

Neighborhoods (DEGLN) [22] and the JADE [23] are instances of DE variants with either adaptive parameters tuning or more efficient evolutionary strategies. A variant of DE called Multi-differential-strategy cooperating Evolution DE (MEDE) was proposed in our previous work [24]. During each loop of the MEDE, a DE strategy is sequentially selected from a pool of preselected DE strategy candidates with repetition to perform evolution operation using current target vector. This process will be executed iteratively until termination conditions are satisfied. However, the MEDE has not fully considered characteristics and bias of different DE strategies. The MEDE could be further improved by selecting DE strategies according to different phases of the DE and tuning parameters adaptively. This paper proposes a new DE-based feature weight learning scheme for similarity-based clustering.

In Section 2, we summarize feature weight learning schemes for similarity-based clustering. Classical DE and Dynamic DE algorithms are introduced in Section 3. The new multiple differential evolution strategies based feature weighting method is proposed in Section 4. We present and discuss experimental results in Section 5. Section 6 provides conclusion and discussion on future works.

2. Feature weight learning for similarity-based clustering

In this section, the similarity-based clustering with weighted features will be introduced briefly. Let $X=\{x_1,x_2,\ldots,x_N\}$ be the set of N objects for clustering and each object $x_i=(x_{i1},x_{i2},\ldots,x_{iM})\in R^M$. The objective of similarity-based clustering is to group these objects into several clusters based on a similarity matrix $S_{N\times N}$ which is composed of pairwise similarity measure of all objects (x_p,x_q):

$$s_{qp}=\delta(x_p,x_q) \tag{1}$$

In [2], the weighted similarity measure is defined as follows:

$$\delta^{(w)}(x_p,x_q)=\frac{1}{1+\beta d_{pq}^{(w)}} \tag{2}$$

where β is a positive parameter determined by solving the following equation:

$$\frac{2}{N(N-1)}\sum_{q>p}\delta_{pq}^{(1)}=0.5 \tag{3}$$

and $d_{pq}^{(w)}$ denotes the weighted Euclidean distance defined as follows:

$$d_{pq}^{(w)}=d_{pq}^{(w)}(x_p,x_q)=\left(\sum_{j=1}^{M}w_j^2(x_{pj}-x_{qj})^2\right)^{1/2} \tag{4}$$

where N, $w=(w_1,w_2,\ldots,w_M)$ and w_i denote the number of objects and the weight vector of features and the importance degree (weight) of the ith feature, respectively. Let $\delta_{pq}^{(1)}$ in Eq. (3) be the similarity between object p and object q with all $w_i=1$. The computation of β in this way will uniformly distribute all similarity values around 0.5 since no additional information is avaliable for estimating these similarity degrees [2].

As shown in Eq. (2), similarity values among objects are greatly affected by feature weights which need to be determined before clustering. Owing to the existence of uncertainty in the judgment of two objects to be similar or disimilar, the similarity matrix is a fuzzy matrix. A larger fuzziness of the similarity matrix leads to a more difficulty in determining crisp clustering result. So, Yeung et al. proposes a feature weight learning scheme to reduce fuzziness of similarity by assigning different weights to features. The scheme minimizes the following fuzziness evaluation function [12]:

$$E(w)=\frac{2}{N(N-1)}\sum_{q<p}\frac{1}{2}(\delta_{pq}^{(w)}(1-\delta_{pq}^{(1)})+\delta_{pq}^{(1)}(1-\delta_{pq}^{(w)})) \tag{5}$$

According to (5), the fuzziness of features reaches its maximum when all weighted similarity degrees are equal to 0.5 and its minimum when all weights are equal to either 0 or 1. A larger $E(w)$ value indicates a larger fuzziness. The maximum fuzziness leads to the most ambiguous clustering while the minimum fuzziness leads to unambiguous clustering. The fuzzy evaluation function $E(w)$ is constructed based on a simple function: $f(x,y)=x(1-y)+y(1-x)$, $0\le x,y\le 1$, and its partial derivative $\partial f/\partial x=1-2y$ has the following properties

$$\frac{\partial f}{\partial x}>0 \text{ if } y<0.5,\quad \frac{\partial f}{\partial x}<0 \text{ if } y>0.5.$$

By minimizing Eq. (5), all $\delta_{pq}^{(w)}$ will approach to 0 (if $\delta_{pq}^{(1)}<0.5$) or 1 (if $\delta_{pq}^{(1)}<0.5$). Clustering with feature weights determined by minimizing Eq. (5) will generally yield a better decision on similar and dissimilar among objects and result in better clustering results in compare with clustering without feature weighting [2]. Moreover, similarity among objects in different clusters will decrease and the average similarity among objects in the same cluster will increase.

Once weights for features are determined, clustering could be performed using off-the-shelf clustering algorithms such as the clustering based on similarity matrix's transitive closure (SMTC-C) [25], fuzzy c-means (FCM) [26,27] and its variants, etc.

However, the GD-based optimization techniques used in [2,6] may be trapped by local minima and could result in a poor clustering results. Therefore, we propose a DE-based feature weight learning method to enhance its performance.

3. Differential evolution and dynamic differential evolution

3.1. Differential evolution

Let $f(x):R^M\to R$ $(LB\le x\le UB)$ be a real-valued function to be minimized, in which $LB=(x_1^{\min},\cdots,x_M^{\min})$ and $UB=(x_1^{\max},\cdots,x_M^{\max})$ denote the lower bound vector and the upper bound vector, respectively. The DE aims to minimize the multi-dimensional real-valued function by iteratively executing a series of evolutional operations over a set of candidate solutions (population).

In the DE, several parameters need to be pre-determined: maximal number of iterations MI, population size NP, differential scale weight F, and crossover probability CR. Then NP candidate solutions are randomly created using the following function to form the initial population P.

$$x_{ij}(0)=x_j^{\min}+rand(0,1)(x_j^{\max}-x_j^{\min}) \tag{6}$$

for $j=1,2,\cdots,M$, where M, $x_i(0)$ and $rand(0,1)$ denote the number of features of an object, the ith individual solution at the generation $G=0$ and a real value generated from a uniform distribution within the range [0, 1].

Then, the following evolutionary operations are iteratively executed over the population.

(1) Mutation: mutation is applied to create a vector $v_{i,G}$ with respect to each individual $x_{i,G}$ at the generation G. Some frequently used mutation strategies are listed as follows:

1. DE/rand/1:

$$v_{i,G}=x_{p_1,G}+F(x_{p_2,G}-x_{p_3,G}) \tag{7a}$$

2. DE/rand/2:

$$v_{i,G}=x_{p_1,G}+F(x_{p_2,G}-x_{p_3,G})+F(x_{p_4,G}-x_{p_5,G}) \tag{7b}$$

3. DE/best/1:

$$v_{i,G} = x_{best,G} + F(x_{p_1,G} - x_{p_2,G}) \tag{7c}$$

4. DE/best/2:

$$v_{i,G} = x_{best,G} + F(x_{p_1,G} - x_{p_2,G}) + F(x_{p_3,G} - x_{p_4,G}) \tag{7d}$$

5. DE/rand-to-best/1:

$$v_{i,G} = x_{i,G} + F(x_{best,G} - x_{i,G}) + F(x_{p_1,G} - x_{p_2,G}) \tag{7e}$$

In Eqs. (7a)–(7e), p_1, p_2, p_3 and p_4 are four mutually distinct integers randomly selected from $\{1, 2, \ldots, i-1, i+1, \ldots, NP\}$. Meanwhile, $x_{best,G}$ is the individual yielding the minimal fitness value at generation G.(2)Crossover: after the mutation step, crossover is applied to generate a trial vector $t_{i,G}$ based on the pair of target vector $x_{i,G}$ and the mutation vector $v_{i,G}$. There are mainly two types of crossover strategies, i.e., binomial crossover (bin) and exponential crossover (exp). A basic binomial crossover operation is as follows:

$$t_{i,G}^j = \begin{cases} v_{i,G}^j, & \text{if } (rand_j[0,1) < CR) \text{ or } j = j_{rand} \\ x_{i,G}^j, & \text{otherwise} \end{cases} \tag{8}$$

where j_{rand} is an integer randomly selected from $[1, M]$. The condition $j = j_{rand}$ is introduced to ensure that the trial vector will differ from its corresponding target vector by at least one feature value.(3)Selection: for each newly generated trial vector, elements of it exceeding the corresponding upper or lower bounds will be randomly re-initialized into the pre-specified range. Then selection operation will be used to select individuals which will survive in the next generation from the transitional population. That is

$$x_{i,G+1} = \begin{cases} t_{i,G}, & \text{if } f(t_{i,G}) < f(x_{i,G}) \\ x_{i,G}, & \text{otherwise} \end{cases} \tag{9}$$

In the DE, these three kinds of operations will be repeated until the maximum number of iterations is reached, or any predefined termination condition is satisfied. The individual yielding the minimum fitness value in the last generation will be returned as the final solution. A pseudocode of the DE is given in Fig. 1.

3.2. Dynamic differential evolution

In traditional DE, the population remains unchanged until it is replaced by a new population. This may reduce the rate of convergence of the DE [28]. So, the Dynamic DE (DDE) is proposed in [29] to dynamically update all individuals at the current generation when a new individual has been selected into the next generation. If the fitness value of the newly selected individual is smaller than that of the current optimal individual, the optimal individual $x_{best,G}$ will also be updated. Trial vectors of DDE are always generated using the newly updated population. The DDE variants of the five mutation strategies given in Eqs. (7a), (7b), (7c), (7d) and (7e) are named as DDE/rand/1, DDE/rand/2, DDE/best/1, DDE/best/2, and DDE/rand-to-best/1, respectively.

Generally speaking, compared with DEs, the DDEs are search algorithms with more greedy bias, better local search capability and faster convergence speed. Conversely, DEs usually have better exploration capability. So, there is a tradeoff between selecting DE and DDE for optimization. This is the major motivation of this work to combine them to make use of advantages from both DE and DDE.

For optimization problem: $\min f(x) : R^M \to R, \quad LB \le x \le UB$

Set parameters: population size ***NP****, maximal iteration number* ***MI****, crossover probability* ***CR****, and differential scaling factor* ***F***

Initialize a population with NP individuals, G=0;

Repeat
    For i=1:NP
        Generate a new trial vector $t_{i,G}$ *base on the mutation and crossover operations*
            If $f(t_{i,G}) < f(x_{i,G})$, $x_{i,G+1} = t_{i,G}$
        Else $x_{i,G+1} = x_{i,G}$
        Endif
    Endfor
    G:=G+1
Until stopping condition is satisfied

Return $x_{best,G}$, $f(x_{best,G})$

Fig. 1. The pseudocode of a basic differential evolution.

4. Feature weight learning based on an improved differential evolution

4.1. New self-adaptive multievolutionary strategy with hybrid differential evolution

DEs with different differential evolution strategies (DESs) yields better optimization results for some specific types of optimization problems. According to the well-known Occam's Razor Principle, there is no DE with a specific DES always outperforms DEs with other strategies. So it is necessary and significant to propose DEs with multiple DESs.

As an improvement to the multiple DES method (MEDE) in our previous work [24], we propose a new multiple DES method with hybrid DE (MEHDE). The good search ability and high efficiency of the MEHDE are guaranteed mainly by the following techniques:

1) Compared with the MEDE, more effective DESs are integrated into the MEHDE to enhance its search capability in terms of both accuracy and convergence ratio. Based on experiences and analysis in lots of previous published works, e.g. [28–32], we chose ten variants of DES as candidate of the evolutionary strategies pool in the MEHDE. Among these strategies, some are more efficient in exploration (e.g., rand/2/bin) for diversity of the population while others perform better in local search and yield faster convergence (e.g., best/1/bin).
2) DE and DDE are both used in MEHDE to provide better global and local searches. The DE-based DESs of the MEHDE will explore an adequate large portion of the solution space and then DDE-based DESs of the MEHDE will perform local search to speed up the convergence of optimization.
3) To guarantee the computational efficiency of the MEHDE, only one DES is executed in each turn for the current individual to create a trial vector, crossover and selection operation. Then the next adjacent DES will be executed for the next individual. This process will be executed in iteration until a stopping criterion is satisfied. The MEHDE improves search performance significantly while uses similar computational cost of traditional DE or DDE algorithms.

In addition, it is shown that DE based on binomial crossover generally outperforms exponential crossover [28]. So, binomial crossover operation is used in the MEHDE. Table 1 lists ten variants of DE/DDE in the candidate DES pool of the MEHDE. These strategies will be sequentially selected. In MEHDE, three termination conditions are adopted to judge whether the algorithm should be terminated. They are (1) the number of iterations G reaches a predefined maximum value MI; (2) the objective function value is smaller than a given threshold δ and (3) the algorithm converges on a value. The pseudocode of the MEHDE is given in Fig. 2.

Table 1
Differential Evolution strategies of the MEHDE.

DE-based DESs	DDE-based DESs
DE/rand/1/bin	DDE/rand/1/bin
DE/rand/2/bin	DDE/rand/2/bin
DE/best/1/bin	DDE/best/1/bin
DE/best/2/bin	DDE/best/2/bin
DE/rand-to-best/1/bin	DDE/rand-to-best/1/bin

For optimization problem: $\min f(x) : R^M \to R, \quad LB \le x \le UB$

Set the control parameters: population size ***NP****, maximal iteration number* ***MI****, differential scaling factor* ***F****, and crossover probability* ***CR***

Let G=0, IV=0
Initialize population pop(G) according to (6);
Let $x_{\mathrm{best},G} = \min_{x \in pop(G)} f(x)$ *optimal individual until G-th generation*

and $x_{\mathrm{best}} = x_{\mathrm{best},G}$ *optimal individual until now*

Repeat

pop_old = pop(G)
For i=1:NP
If $IV<10$, $IV = IV+1$, else $IV = 1$
Generate a new trial vector $t_{i,G}$ *by the IV-th differential evolution strategy from the candidate DES pool (DE-based DESs are based on pop_old(G) and* $x_{\mathrm{best},G}$*, while DDE-based DESs are based on pop(G) and* x_{best} *)*
{DE/rand/1/bin,
DE/rand/2/bin,
DE/best/1/bin,
DE/best/2/bin,
DE/rand-to-best/1/bin,
DDE/rand/1/bin
DDE/rand/2/bin,
DDE/best/1/bin,
DDE/best/2/bin,
DDE/rand-to-best/1/bin}

If $f(t_{i,G}) < f(x_{i,G})$*, then* $x_{i,G+1} = t_{i,G}$, $x_{i,G} = t_{i,G}$;
If $f(x_{\mathrm{best}}) < f(t_{i,G})$ *then* $x_{\mathrm{best}} = t_{i,G}$;
Endif
Else $x_{i,G+1} = x_{i,G}$ *Endif*
Endfor
G=G+1;
Update $x_{\mathrm{best},G} = \min_{x \in pop(G)} f(x)$
Until stopping condition is satisfied

Return $x_{\mathrm{best},G}$, $f(x_{\mathrm{best},G})$

Fig. 2. The pseudocode of our proposed MEHDE.

4.2. Feature weight learning based on MEHDE

In this subsection, the proposed feature weight learning method based on the improved differential evolution MEHDE will be introduced. As introduced in Section 2, by minimizing the fuzziness degree $E(w)$ of the similarity matrix among objects, the uncertainty (fuzziness and nonspecificity) of similarity-based clustering will be reduced. It yields a better clustering partition with higher intra-class similarity and lower inter-class similarity. Meanwhile, the nonspecificity for obtaining a crisp partition will also be decreased. So, in our proposed feature weight learning method, the fuzzy degree of similarity matrix will be selected as the objective function to be minimized. Then, the feature weight learning problem can be written as

$$\min E(w) \quad s.t. \quad w \in R^M, \quad lw \le w \le uw \tag{10}$$

where w, M, lw and uw denote the feature weight vector to be optimized, the number of features, the lower bound and the upper bound of feature weights, respectively. According to Eqs. (1)–(5), the optimization problem (10) can be rewritten as

$$\min \sum_{q<p} \left(\frac{1}{1+\beta d_{pq}^{(w)}} \left(1 - \frac{1}{1+\beta d_{pq}^{(1)}} \right) + \frac{1}{1+\beta d_{pq}^{(1)}} \left(1 - \frac{1}{1+\beta d_{pq}^{(w)}} \right) \right)$$

$$s.t. \quad w \in R^M, \quad lw \le w \le uw \tag{11}$$

where $d_{pq}^{(1)}$, $d_{pq}^{(w)}$ and β denote the Euclidean distance, the weighted Euclidean distance and a positive constant for calculating the similarity between two objects. The coefficient $1/N(N-1)$ is ignored because it is a constant and the solution of (11) does not dependent on it.

For a given dataset, the constant β and the Euclidean distance between objects x_p and x_q ($d_{qp}^{(1)}$) are fixed. Moreover, the values of feature weights indicate the relative importance degree of features. Our experiments show that, for most datasets, optimal values of feature weights are always close to zero, especially in cases that feature values are normalized to [0, 1]. Based on this observation, w is selected from a closed interval near zero $[lw, uw]$ to enhance the search efficiency of the MEHDE.

The feature weight learning problem becomes a minimization problem of an M-dimensional real-valued function as (11). Resulting weights could be used in weighted similarity-based clustering to enhance cluster partitioning capability. In most cases, the objective function, i.e. the fuzziness degree function (5), is usually multimodal and generally has multiple local minima. GD-based search methods usually have many disadvantages such as trapped by local minima, strong dependency to the learning rate and solution initialization, and low search efficiency, etc. Therefore, the proposed MEHDE is used to solve the optimization problem (11). By combining the feature weight optimization model in (11) and the DE-based MEHDE, a better performance of weighted similarity-based clustering is expected.

5. Experimental results and discussion

5.1. Experimental setting

The clustering performance of the proposed MEHDE-FWL will be compared with GD-based feature weight learning (GD-FWL) method and clustering without feature weighting. In order to evaluate the performance of the proposed MHEDE-based feature weight learning algorithm (MHEDE-FWL), 11 benchmarking datasets are used: Rice Taste Data from [38] and 10 datasets from the UCI machine learning data repository [39]. Class labels of those datasets are disregarded and clustering is performed instead of classification for these datasets. A summary of these datasets is given in Table 2.

All data is normalized into [0, 1] to avoid counterproductive influence from feature with an extremely large range. In our experiments, $lw = 0$ and $uw = 10$. Meanwhile, the constant β is determined for each dataset by solving Eq. (3). For the MEHDE-FWL, the differential scale factor F and crossover probability CR are selected within intervals [0.3, 0.9] and [0.1, 1], respectively, based on experiences and recommendations from [28,30–32]. For each dataset, values of F and CR are selected by a trial-and-error method. Tables 3 and 4 show parameters used by the MEHDE-FWL for experiments.

For the GD-FWL, we set the maximum number of iterations to be 6000 and the learning rate $\eta \in [0.1, 5]$. In iteration of GD, the learning rate will be dynamically updated by a Fibonacci search as in [2].

Two clustering algorithms are used in experiments: similarity matrix's transitive closure clustering (SMTC-C) [25] and fuzzy c-means clustering (FCM) [26,27]. They are directly applied as the baseline method: clustering without feature weighting. Their weighted versions will be referred as weighted SMTC-C (WSMTC-C) and weighted FCM (WFCM).

Table 2
A summary of selected datasets in our experiments.

Datasets	# Instance	# Feature
Rice taste data	105	5
Iris	150	4
Servo	167	4
Thyroid gland	215	5
BUPA	345	6
MPG	398	8
Boston housing	506	13
Pima	768	8
Image segmentation	2310	19
Libras movement	360	90
Gas sensor (batch1)	445	128

Table 3
Parameters of MEHDE-FWL.

Population size NP	10 * # feature
Maximal iteration number MI	1000
w	$0 \le w \le 10$

Table 4
Values of CR and F for each dataset.

Datasets	CR	F
Rice taste data	0.8	0.6
Iris	1.0	0.4
Servo	0.4	0.4
Thyroid gland	1.0	0.4
BUPA	1.0	0.4
MPG	0.4	0.4
Boston housing	0.6	0.4
Pima	0.8	0.4
Image segmentation	0.8	0.4
Libras movement	0.6	0.8
Gas sensor (batch 1)	0.6	1.0

5.2. Comparison between MEHDE-FWL and GD-FWL based on FCM

The WFCM with weights optimized by GD-FWL is proposed in [6]. Four clustering evaluation indices are used for evaluating the performance of fuzzy clustering [6]. In this subsection, we use the same set of indices to compare clustering results of classical FCM, WFCM with GD-FWL and WFCM with MEHDE-FWL.

Table 5 shows the four indices for evaluating clustering results. The partition coefficient V_{pc} [33] and the partition entropy V_{pe} [34] are based on the fuzzy partition of sample set, while the Fukuyama–Sugeno function V_{fs} [35] and the Xie–Beni function V_{xb} [36] are based on the geometric sample structure. Empirical studies in [37] show that a good interpretation of partitions over samples may be obtained by maximizing V_{pc} or minimizing V_{pe}. Meanwhile, a good clustering should yield compact clusters while far separated among different clusters which usually minimize V_{fs} and V_{xb} [6,36].

To illustrate the validity of the optimization model for feature weight learning, the clustering result of the Iris dataset by WFCM with MEHDE-FWL is shown in Fig. 3. The feature weight vector determined by the MEHDE-FWL is [0.0001, 0.0005, 3.5292, 0.0011]. The 1st and 2nd features are not shown in Fig. 3 because they yield very small feature weights. This result is consistent with results of other works, so the proposed method is able to weight features according to their significance.

Table 6 shows experimental results of the three clustering methods on 11 datasets. Results of clustering with feature weighting by both MEHDE-FWL and GD-FWL being shown in Table 6 are average over 20 repeated runs. Their standard deviation values are shown in brackets after their average values. The WFCM with MEHDE-FWL yields the best performance almost in all indices of all experiments. The WFCM without feature weighting perform the worst in most cases. Table 6 also shows that results yielded by the proposed method is more robust (smaller standard deviations) than the other two methods in comparisons.

Moreover, the proposed method yields larger improvement for complex datasets. This further verifies that the DE-based optimization algorithms yield better search capabilities than that of GD-based methods, especially for complex or multimodal problems. Meanwhile, experimental results show that all the four indices cannot be optimized simultaneously. So, optimal values in the largest number of indices could be a good choice for finding the optimal number of clusters.

5.3. Comparisons based on SMTC clustering

In this subsection, we will use the SMTC-C [25] as the clustering technique to compare clustering without feature weighting, with MEHDE-FWL and with GD-FWL. In the SMTC-C, clustering is performed by thresholding a transitive closure $TC(S) = (t_{ij})_{N \times N}$ of

Table 5
A brief summary of the four selected clustering evaluation indices for FCM.

Evaluation index	Functional description	Optimal partition
Partition coefficient	$V_{pc}(U) = (\sum_{j=1}^{n} \sum_{i=1}^{c} u_{ij}^2)/n$	$\max(V_{pc})$
Partition entropy	$V_{pe}(U) = -\frac{1}{n}[\sum_{j=1}^{n} \sum_{i=1}^{c} (u_{ij} \log u_{ij})]$	$\min(V_{pe})$
Fukuyama–Sugeno function	$V_{fs}(U, v_1, L, v_c; X) = \sum_{j=1}^{n} \sum_{i=1}^{c} u_{ij}^2 (\|\|X_j - v_i\|\|^2 - \|\|v_i - \overline{v}\|\|^2)$	$\min(V_{fs})$
Xie–Beni function	$V_{xb}(U, v_1, L, v_c; X) = \frac{\sum_{j=1}^{n} \sum_{i=1}^{c} u_{ij}^2 \|\|X_j - v_i\|\|^2}{n(\min_{i \neq k}\{\|\|v_i - v_k\|\|^2\})}$	$\min(V_{xb})$

similarity matrix $S_{N \times N}$. Objects x_i and x_j are categorized into the same cluster if $t_{ij} \geq \alpha$ where α is a given threshold.

We adopt the same setting as in Section 5.2, but with another set of evaluation indices. Detailed descriptions of these indices could be found in [2] and we provide a brief description here

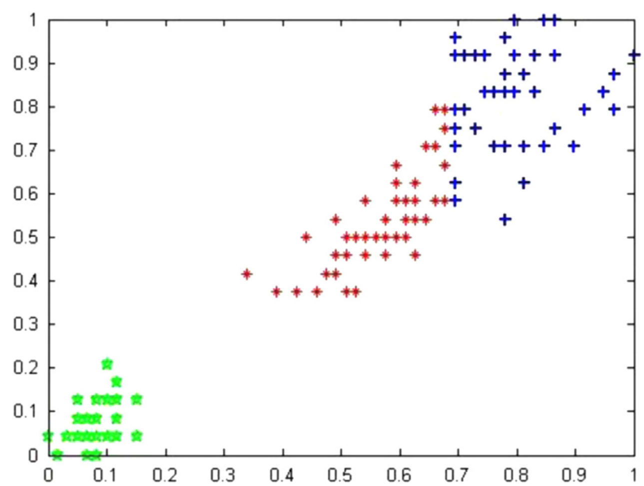


Fig. 3. Clustering of IRIS by FCM based on feature weights vector [0.0001, 0.0005, 3.5292, 0.0011].

(1) *Fuzziness of the similarity matrix.* Smaller fuzziness leads to more crisp clustering decision.
(2) *Intra-class similarity*. It is computed by the average similarity of all pairs of objects belonging to the same cluster. Larger intra-class similarity shows a better clustering.
(3) *Inter-class similarity.* It is computed by the average similarity of all pairs of objects belonging to different clusters. Smaller inter-class similarity is preferred for good clustering.
(4) *Ratio of intra-class to inter-class similarities*. This provides an overall evaluation of intra- and inter- class similarities. Larger ratio value indicates a better clustering result.
(5) *Nonspecificity.* This index evaluates the difficulty of selecting a partition from a clustering graph. A smaller nonspecificity value indicates a more crisp clustering.

After feature weights being fixed, clustering graph will be computed using the SMTC-C with different threshold values. Based on the computed clustering graph, aforementioned indices are computed to evaluate the clustering results. Similar to Section 5.2, average and standard deviations of indices for SMTC-C with both GD-FWL and MEHDE-FWL are computed over 20 independent runs.

Experimental results are shown in Table 7. Again, the SMTC-C with MEHDE-FWL outperforms other methods and the SMTC-C without feature weighting performs the worst, in most cases. In comparison to the SMTC-C with GD-FWL, the SMTC-C with MEHDE-FWL yields significantly improvements in the fuzziness, the nonspecificity and the ratio of intra-class to inter-class similarities for all datasets in experiments.

Table 6
Experimental results of FCM, WFCM with GD-FWL and WFCM with MEHDE.

Data sets	Methods	V_{pc}	V_{pe}	V_{fs}	V_{xb}
Iris	FCM	0.86	0.25	−17.59	0.08
	WFCM-GD	0.93 (0.000)	0.14 (0.001)	−19.85 (0.130)	0.08 (0.000)
	WFCM-MEHDE	0.93 (0.000)	0.13 (0.000)	−20.01 (0.000)	0.08 (0.000)
Pima	FCM	0.57	0.62	68.72	1.17
	WFCM-GD	0.60 (0.050)	0.59 (0.050)	67.72 (6.960)	1.57 (1.390)
	WFCM-MEHDE	0.81 (0.004)	0.31 (0.007)	64.19 (0.320)	0.62 (0.004)
Rice	FCM	0.75	0.41	0.09	0.19
	WFCM-GD	0.77 (0.027)	0.38 (0.041)	0.21 (0.317)	0.19 (0.013)
	WFCM-MEHDE	0.82 (0.000)	0.30 (0.005)	−0.14 (0.035)	0.20 (0.009)
Servo	FCM	0.62	0.56	26.41	0.50
	WFCM-GD	0.68 (0.107)	0.48 (0.144)	25.54 (7.577)	0.69 (0.232)
	WFCM-MEHDE	0.99 (0.035)	0.03 (0.082)	10.21 (1.026)	2.10 (0.095)
Thyroid	FCM	0.7	0.47	8.34	0.56
	WFCM-GD	0.82 (0.019)	0.32 (0.055)	1.43 (3.800)	0.25 (0.124)
	WFCM-MEHDE	0.87 (0.022)	0.24 (0.053)	1.73 (3.172)	0.25 (0.092)
Bupa	FCM	0.6	0.58	16.81	0.93
	WFCM-GD	0.64 (0.055)	0.54 (0.067)	16.00 (1.881)	0.84 (0.277)
	WFCM-MEHDE	0.77 (0.004)	0.38 (0.005)	13.21 (0.042)	0.47 (0.000)
MPG	FCM	0.73	0.43	2.53	0.19
	WFCM-GD	0.83 (0.135)	0.28 (0.210)	−50.59 (80.10)	0.31 (0.151)
	WFCM-MEHDE	0.96 (0.008)	0.09 (0.019)	−103.21 (41.38)	0.46 (0.011)
Boston	FCM	0.74	0.42	33.46	0.23
	WFCM-GD	0.73 (0.030)	0.40 (0.390)	47.72 (17.409)	0.25 (0.019)
	WFCM-MEHDE	0.97 (0.009)	0.09 (0.179)	-54.24 (1.229)	0.33 (0.004)
Image	FCM	0.40	1.25	−464	0.58
	WFCM-GD	0.39 (0.016)	1.27 (0.029)	−417.88 (25.016)	0.90 (0.133)
	WFCM-MEHDE	0.64 (0.014)	0.75 (0.027)	−787 (18.213)	0.65 (0.043)
Libras	FCM	0.07	2.71	103.61	2.74E+08
	WFCM-GD	0.07 (0.125)	2.70 (0.024)	103.61 (12.500)	4.59E+07 (1.02E+07)
	WFCM-MEHDE	0.07 (0.012)	2.69 (0.0001)	101.52 (3.176)	2.42E+05 (1.03E+04)
Gas	FCM	0.41	1.52	−703.20	0.36
	WFCM-GD	0.42 (0.086)	1.50 (0.180)	−717.05 (121.70)	0.34 (0.073)
	WFCM-MEHDE	0.42 (0.001)	1.49 (0.012)	−726.15 (27.61)	0.39 (0.002)

Overall, experimental results show that the proposed MEHDE algorithm yields a better optimization capability and more robust output in comparison to GD-based approaches. Moreover, clustering with feature weights computed by the proposed MEHDE based method outperforms the clustering with GD-based feature weight learning method.

5.4. Efficiency analysis of GD, DE, DDE and MEHDE based search techniques

The efficiencies of GD-based methods strongly depend on both initialization and learning rate. GD could converge very slowly or even diverge if improper choice is made for either initialization or learning rate. In contrast, DE-based methods demonstrate strong search abilities and relatively robust performances [15–24,41].

On the other hand, compared with DEs and DDEs, the proposed MEHDE usually yields a better search ability and robust performance without additional computational cost. These are achieved mainly by two improvements: (1) integration of multiple variants of DESs which are efficient either in local or global searches; (2) adopting of a sequential selection method for candidate DESs, so only one DES is used to update the current individual in population. So, the computational complexity of the MEHDE is basically identical to that of DE/rand/1/bin. Owing to the fact that mutation and crossover operations are performed at the component level for each DE vector, the amount of fundamental operations in DE/rand/1/bin is proportional to the total number of loops conducted in the algorithm [40]. Meanwhile, in each generation of DE, a loop over NP is conducted which contains of a loop over M(the dimensionality of the vector to be optimized). Thus, the computational complexity of MEHDE is $O(NP \cdot M \cdot G_*)$. Furthermore, in our method the runtime complexity of evaluating the fitness function, i.e. the fuzziness degree of the similarity matrix, is $O(N^2)$. So, the computational complexity of the MEHDE-based feature weight learning is $O(NP \cdot M \cdot G_* \cdot N^2)$.

The trial-and-error based tuning of the scaling factor F and the crossover probability CR in the proposed method will cost considerable computation resources. So, one of the important future works of us is to add direct assignment based on experience or self-adaptive parameter techniques to the MEHDE to overcome this disadvantage.

6. Conclusion and future works

This paper is devoted to develop an effective and robust feature weighting technique for similarity-based clustering. The objective is to improve the performance of clustering in terms of some selected clustering evaluation indices. So, the MEHDE integrating multiple DESs and hybrid DE/DDE scheme is proposed. Then, the MEHDE is used to determine feature weights in similarity-based clustering by minimizing the uncertainty (fuzziness and nonspecificity) existing in the similarity matrix. This reduces the difficulty of making a crisp decision based on similarity matrix. The

Table 7
Experimental results of SMTC-C, W-SMTC-C with GD-FWL, and W-SMTC wih MEHDE-FWL.

Data sets	Methods	Intra-similarity	Inter-similarity	Ratio	Maximal ratio	Fuzziness	Non-specificity
Iris	SMTC-C	0.88	0.44	1.99	2.16	0.32	1.78
	SMTC-C-GD-FWL	0.85 (0.00)	0.32 (0.00)	2.75 (0.03)	3.28 (0.04)	0.27 (0.00)	1.25 (0.04)
	SMTC-C-MEHDE –FWL	0.85 (0.00)	0.31 (0.00)	2.78 (0.00)	3.30 (0.00)	0.27 (0.00)	1.17 (0.00)
Pima	SMTC-C	0.94	0.42	2.25	2.7	0.34	1.83
	SMTC-C-GD-FWL	0.96 (0.00)	0.24 (0.00)	4.78 (0.62)	6.42 (0.74)	0.21 (0.00)	1.74 (0.02)
	SMTC-C-MEHDE –FWL	0.93 (0.00)	0.22 (0.00)	4.79 (0.00)	6.13 (0.01)	0.31 (0.00)	1.25 (0.00)
Rice	SMTC-C	0.92	0.40	2.38	2.79	0.33	1.85
	SMTC-C-GD-FWL	0.91 (0.00)	0.17 (0.00)	4.00 (2.13)	6.40 (3.56)	0.22 (0.00)	1.72 (0.03)
	SMTC-C-MEHDE –FWL	0.89 (0.00)	0.30 (0.00)	3.48 (0.07)	5.68 (0.22)	0.30 (0.00)	1.54 (0.00)
Servo	SMTC-C	0.73	0.47	1.54	1.99	0.34	0.87
	SMTC-C-GD-FWL	0.72 (0.02)	0.18 (0.00)	4.28 (1.21)	5.80 (2.25)	0.22 (0.00)	0.93 (0.15)
	SMTC-C-MEHDE –FWL	0.99 (0.00)	0.17 (0.00)	6.15 (0.70)	6.25 (0.29)	0.16 (0.00)	0.07 (0.05)
Thyroid	SMTC-C	0.92	0.34	2.83	3.49	0.32	1.75
	SMTC-C-GD-FWL	0.91 (0.00)	0.26 (0.00)	4.18 (0.47)	7.44 (2.93)	0.29 (0.00)	1.57 (0.05)
	SMTC-C-MEHDE –FWL	0.91 (0.00)	0.24 (0.00)	4.57 (0.00)	7.65 (0.00)	0.28 (0.00)	1.19 (0.00)
Bupa	SMTC-C	0.94	0.39	2.47	2.91	0.33	1.90
	SMTC-C-GD-FWL	0.97 (0.00)	0.14 (0.00)	4.22 (0.58)	7.49 (0.95)	0.31 (0.00)	1.69 (0.01)
	SMTC-C-MEHDE –FWL	0.95 (0.00)	0.30 (0.00)	3.39 (0.00)	6.55 (0.01)	0.30 (0.00)	1.37 (0.00)
MPG	SMTC-C	0.85	0.48	1.76	2.00	0.33	1.69
	SMTC-C-GD-FWL	0.89 (0.00)	0.17 (0.00)	5.26 (0.55)	6.03 (0.37)	0.20 (0.00)	1.65 (0.58)
	SMTC-C-MEHDE –FWL	0.94 (0.00)	0.19 (0.00)	5.46 (0.78)	6.56 (0.15)	0.17 (0.00)	0.92 (0.02)
Boston	SMTC-C	0.86	0.45	1.92	2.14	0.33	1.97
	SMTC-C-GD-FWL	0.88 (0.00)	0.16 (0.00)	5.64 (0.93)	6.32 (1.19)	0.22 (0.00)	2.29 (0.01)
	SMTC-C-MEHDE –FWL	0.94 (0.00)	0.26 (0.00)	4.14 (0.02)	6.73 (0.02)	0.23 (0.00)	1.03 (0.00)
Image	SMTC-C	0.91	0.44	2.08	2.32	0.33	1.95
	SMTC-C-GD-FWL	0.89 (0.00)	0.23 (0.00)	4.03 (0.40)	4.93 (0.23)	0.27 (0.00)	2.02 (0.05)
	SMTC-C-MEHDE –FWL	0.91 (0.00)	0.16 (0.00)	5.70 (0.01)	6.15 (0.09)	0.22 (0.00)	2.39 (0.01)
Libras	SMTC-C	0.90	0.48	1.91	1.97	0.34	2.00
	SMTC-C-GD-FWL	0.91 (0.00)	0.20 (0.00)	4.64 (0.18)	4.87 (0.21)	0.25 (0.01)	1.95 (0.02)
	SMTC-C-MEHDE –FWL	0.93 (0.00)	0.15 (0.00)	6.10 (0.07)	6.39 (0.05)	0.21 (0.00)	1.88 (0.00)
Gas	SMTC-C	0.94	0.39	2.45	2.68	0.32	2.06
	SMTC-C-GD-FWL	0.94 (0.00)	0.16 (0.02)	6.13 (0.63)	8.43 (0.42)	0.24 (0.06)	1.73 (0.02)
	SMTC-C-MEHDE -FWL	0.94 (0.00)	0.13 (0.00)	7.25 (0.15)	10.30 (0.02)	0.20 (0.00)	1.70 (0.00)

contributions of this paper mainly include the proposal of a multiple differential strategies with hybrid DE/DDE algorithm, and its adoption into feature weight learning for similarity-based clustering. Experimental results on 11 datasets show that the FCM and SMTC-C based on our method usually outperform that without feature weighting and that based on GD-based feature weight learning in terms of better clustering evaluation indices and robust performance.

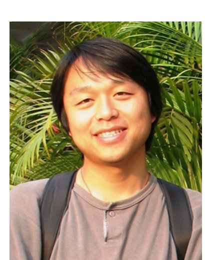
For datasets with more features, as indicated in [41], the performance of DE-based optimization methods may be deteriorated, especially when the dimensionality of the search space larger than 500. So, one of the future works of us is to improve the proposed method for large dataset with large number of features. One possibility is to incorporate latest evolutionary strategies which are specifically proposed for large-scale optimization, e.g. the fittest individual refinement (FIR) based method [42] and the stochastic properties of chaotic system based approach [43]. As stated in Section 5.4, we will also focus on the improvement of the MEHDE algorithm based on some self-adaptive parameter tuning techniques.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61003171, 61272201, 61170040), by the Natural Science Foundation of Hebei Province (F2013201110 and F2013201220), by the Key Scientific Research Foundation of Education Department of Hebei Province (ZD2010139), and by the Development of Science and Technology Mentoring Program of Baoding (10ZG007).

References

[1] A. Baraldi, P. Blonda, A survey of fuzzy clustering algorithms for pattern recognition, IEEE Trans. SMC-B 29 (6) (1999) 778–785.
[2] D.S. Yeung, X.Z. Wang, Improving performance of similarity-based clustering by feature weight learning, IEEE Trans. PAMI 24 (4) (2002) 556–561.
[3] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[4] D.Q. Zhang, S.C Chen, Z.H. Zhou, Constraint score: a new filter method for feature selection with pairwise constraints, Pattern Recognit. 41 (5) (2008) 1440–1451.
[5] Cosmin Lazar, Jonatan Taminau, Stijn Meganck, et al., A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (4) (2012) 1106–1119.
[6] X.Z. Wang, Y.D. Wang, L.J. Wang, Improving fuzzy c-means clustering based on feature-weight learning, Pattern Recognit. Lett. 25 (10) (2004) 1123–1132.
[7] Y.F. Yue, D.Y. Zeng, L. Hong, Improving fuzzy c-means clustering by a novel feature-weight learning, Pacific–Asia Workshop Comput. Intell. Ind. Appl. 2 (2008) 173–177.
[8] N. Ishii, Y. Wang, Learning feature weights for similarity using genetic algorithms, in: Proceedings of IEEE International Joint Symposia on Intelligence and Systems, 1998, pp. 27–33.
[9] F.R. Meng, M. Zhu, An improved weighted feature abstracting algorithm, in: Proceedings of the ICCASE 2009, 2009, pp. 96–99.
[10] E.Y. Chan, W.K. Ching, M.K. Ng, et al., An optimization algorithm for clustering using weighted dissimilarity measures, Pattern Recognit. 37 (5) (2004) 943–952.
[11] D.S. Modha, W.S. Spangler, Feature weighting in k-means clustering, Mach. Learn. 52 (2003) 217–237.
[12] J. Basak, R.K. De, S.K. Pal, Unsupervised feature selection using a neuro-fuzzy approach, Pattern Recognit. Lett. 19 (1998) 997–1006.
[13] S.K. Pal, R.K. De, J. Basak, Unsupervised feature evaluation: A neuro-fuzzy approach, IEEE Trans. Neural Netw. 11 (2) (2000) 366–376.
[14] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[15] R. Storn, K.V. Price, Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces, Techical Report TR-95-012, ICSI, USA, 1995.
[16] R. Storn, On the usage of differential evolution for function optimization, in: Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), 1996, pp. 519–523.
[17] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous space, J. Glob. Optimiz. 11 (4) (1997) 341–359.
[18] P. Rocca, G. Oliveri, A. Massa, Differential evolution as applied to electromagnetics, IEEE Antennas Propag. Mag. 53 (1) (2011) 38–49.
[19] J. Brest, S. Greiner, B. Bŏskovi´c, et al., Selfadapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.
[20] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Opposition based differential evolution, IEEE Trans. Evol. Comput. 12 (1) (2008) 64–79.
[21] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.
[22] S. Das, A. Abraham, U.K. Chakraborty, et al., Differential evolution using a neighborhood based mutation operator, IEEE Trans. Evol. Comput. 13 (3) (2009) 526–553.
[23] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.
[24] Y.C. He, X.Z. Wang, K.Q. Liu, et al., Convergent analysis and algorithmic improvement of differential evolution, J. Softw. 21 (5) (2010) 875–885.
[25] D. Dubois, Fuzzy Sets and Systems: Theory and Applications, Academic Press, New York, Boston, 1980.
[26] J.C. Dunn, Some recent investigations of a new fuzzy partition algorithm and its application to pattern classification problems, J. Cybern. 4 (1974) 1–15.
[27] J.C. Bezdek, Pattern recognition with fuzzy objective, Function Algorithms, Plenum, New York, 1981.
[28] G. Jeyakumar, C. Shunmuga Velayutham, A comparative performance analysis of differential evolution and dynamic differential evolution variants, World Congr. Nat. Biol. Inspired Comput. (2009) 463–468.
[29] A. Qing, Dynamic differential evolution strategy and applications in electromagnetic inverse scattering problems, IEEE Trans. Geosci. Remote Sens. 44 (1) (2006) 116–125.
[30] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417.
[31] J. Vesterstrom, R. Thomsen, A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, Congr. Evol. Comput. 2 (2004) 1980–1987.
[32] M.M. Efren, V.R. Jesus, A. Coello Coello Carios, A comparative study on differential evolution variants for global optimization, in: Proceedings of the Gennetic and Evolutionary Computation Conferrence, GECCO'06, 2006, pp. 485–492.
[33] J.C. Bezdek, Cluster validity with fuzzy sets, J. Cybern. 3 (3) (1974) 58–73.
[34] J.C. Bezdek, Mathematical models for systematic and taxonomy, in: Proceedings of 8th International Conference on Numerical Taxonomy, San Francisco, 1975, pp. 143–166.
[35] Y. Fukuyama, M. Sugeno, A new method of choosing the number of clusters for the fuzzy c-means method, in: Proceedings of 5th Fuzzy System Symposium, 1989, pp. 247–250.
[36] X.L. Xie, G.A. Beni, Validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 3 (8) (1991) 841–846.
[37] J.C. Dunn, Indices of partition fuzziness and the detection of clusters in large data sets, in: M.M. Gupta (Ed.), Fuzzy Automata and Decision Process, Elsevier, New York, 1976.
[38] K. Nozaki, H. Ishibuchi, H. Tanaka, A simple but powerful heuristic method for generating fuzzy rules from numerical data, Fuzzy Sets Syst. 88 (1997) 299–313.
[39] K. Bache, M. Lichman, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2013, ⟨http://www.archive.ics.uci.edu/ml⟩.
[40] K. Zielinski, R. Laur, Constrained single-objective optimization using differential evolution, in: Proceedings of the IEEE Congress Evolutionary Computation, 2006, pp. 927–934.
[41] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.
[42] N. Noman, H. Iba, Enhancing differential evolution performance with local search for high dimensional function optimization, in: Proceedings of the Conference on Genetic and Evolutionary Computation, 2005, pp. 967–974.
[43] Y. Gao, Y. Wang, A memetic differential evolutionary algorithm for high dimensional function spaces optimization, in: Proceedings of the 3rd ICNC, vol. 4, 2007, pp. 188–192.



Chun-Ru Dong received his B.Sc. degree in computational mathematics in 2002 and M.Eng. degree in computer science in 2005 from Hebei University. He works as a Lecturer in College of Mathematics and Computer Science, Hebei University, China, and is pursuing a Ph.D. degree in computer science from South China University of Technology, China. His research interests include sensitivity analysis for inductive learning, rule-based fuzzy reasoning system, neural networks, evolutionary algorithms and their applications, adversarial learning, etc.

Wing W. Y. Ng received his B.Sc. and Ph.D. degrees from Hong Kong Polytechnic University in 2001 and 2006, respectively. He is now an Associate Professor in the School of Computer Science and Engineering, South China University of Technology, China. His major research interests include business intelligence, large scale image retrieval and machine learning techniques. Dr. Ng is currently an associate editor of the International Journal of Machine Learning and Cybernetics. He is the principle investigator of two China National Nature Science Foundation projects and a Program for New Century Excellent Talents in University from China Ministry of Education. Dr. Ng served as the Board of Governor of IEEE Systems, Man and Cybernetics Society (SMCS) in 2011–2013.

Xi-Zhao Wang received his Ph.D. in computer science from Harbin Institute of Technology on September 1998. From October 1998 to September 2001 he worked at Department of Computing in Hong Kong Polytechnic University as a research fellow. He is presently the Dean and Professor of the Faculty of Mathematics and Computer Science, Hebei University, China. His research interests include machine learning, learning with fuzzy representation, approximate reasoning and expert systems, neural networks and their sensitivity analysis, statistical learning and support vector machine, fuzzy measures and fuzzy integrals, rough set theory, random weight network, and the recent topic: machine learning theories and methodologies in big data environment. He has edited more than 6 special issues and published 3 monographs, 2 textbooks, and over 150 peer-reviewed research papers including, IEEE Transactions on PAMI, Information Sciences, etc.

He is an IEEE Fellow; the chair of IEEE SMC Technical Committee on Computational Intelligence, the editor-in-chief for the International Journal of Machine Learning and Cybernetics; an Associate Editor of IEEE Transactions on SMC, Part B; an Associate Editor of Pattern Recognition and Artificial Intelligence; a member of editorial board of Information Sciences.

Patrick P.K. Chan received the B.Sc. degree in Information Technology and the Ph.D. degree from Hong Kong Polytechnic University. He is currently an Associate Professor in South China University of Technology, China. His current research interests include adversary learning, multiple classifier system and localized generalization error model. He is a member of the Board of Governors of IEEE SMC society and also the Chairman of IEEE SMC Hong Kong Chapter.

Daniel S. Yeung is currently a Chair Professor with School of Computer Science and Engineering, South China University of Technology. He has been a faculty member of Hong Kong Polytechnic University, Hong Kong, and Rochester Institute of Technology, USA. Prof. Yeung has also worked for TRW Inc., General Electric Corporation R&D Centre and Computer Consoles Inc. in USA. He is a Fellow of the IEEE and served as the President of IEEE SMC Society in 2008–2009.