Pattern Recognition Letters 65 (2015) 51–59

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Simultaneous feature selection and weighting – An evolutionary multi-objective optimization approach☆

Sujoy Paul[a], Swagatam Das[b,*]

[a] Department of Electronics and Tele-Communication Engineering, Jadavpur University, Kolkata 700032, India
[b] Electronics and Communication Sciences Unit, India Statistical Institute, Kolkata 700108, India

ARTICLE INFO

Article history:
Received 8 December 2014
Available online 19 July 2015

Keywords:
Feature selection
Feature weighting
Evolutionary multi-objective optimization
MOEA/D
Inter- and intra-class distances

ABSTRACT

Selection of feature subset is a preprocessing step in computational learning, and it serves several purposes like reducing the dimensionality of a dataset, decreasing the computational time required for classification and enhancing the classification accuracy of a classifier by removing redundant and misleading or erroneous features. This paper presents a new feature selection and weighting method aided with the decomposition based evolutionary multi-objective algorithm called MOEA/D. The feature vectors are selected and weighted or scaled simultaneously to project the data points to such a hyper space, where the distance between data points of non-identical classes is increased, thus, making them easier to classify. The inter-class and intra-class distances are simultaneously optimized by using MOEA/D to obtain the optimal features and the scaling factor associated with them. Finally, *k*-NN (*k*-Nearest Neighbor) is used to classify the data points having the reduced and weighted feature set. The proposed algorithm is tested with several practical datasets from the well-known data repositories like UCI and LIBSVM. The results are compared with those obtained with the state-of-the-art algorithms to demonstrate the superiority of the proposed algorithm.

1. Introduction

Representing data with minimal number of meaningful and discriminative attributes or features is a crucial preprocessing step in exploratory data analysis. A dramatic growth in the high throughput technologies has led to the regular production of large datasets characterized by unprecedented number of features. When such high dimensional data are used as input to a decision making, learning or/and knowledge based system, high computational time is required, and this may even demand superior technology for real world applications. However, such data may have several features which are redundant, while some features may be misleading and give rise to confusion in decision making and classification process. The task of Feature Selection (FS) is to remove such features and select only those features which are sufficient for solving a problem, thus leading to a reduction in the computational overhead and higher classification accuracy of the classifiers.

Over the years, several FS and dimensionality reduction techniques have been proposed. They may be broadly classified as random [2,50,56], heuristic [9,33,49], or exhaustive [6,32]. In terms of function optimization perspective, FS algorithms may also be classified based on information [5,36,45], distance [26,57], similarity [30,14], consistency [58,3], and classifier error rate [68]. Mitra et al. [40] proposed an FS technique based on Feature Similarity (FSFS). Padungweang et al. [44] presented a new approach for FS, using the optical diffraction principle. Recently a non-parametric Bayesian error minimization scheme for FS was proposed by Yang and Hu [63]. Peng et al. [46] proposed a minimum Redundancy Maximum Relevance (mRMR) based FS technique that may use correlation, mutual information, and distance/similarity scores to select features. Wang et al. [55] presented a ranking based feature selection method, by using a convex hull for each class and computing the discriminative degree. A probabilistic prediction approach for FS was undertaken in [61]. Network based feature selection techniques have also flourished over the years. Romero and Sopena [48] used a multi-layer perceptron based wrapper approach for FS. They presented the advantage of network retraining at the cost of its computational overhead. Later, another network based FS technique was proposed by Hancock and Mamitsuka [19]. The boosted network classifier approach used by them was not extensively compared using a large number of datasets.

Wrapper method [22,28,62,18] is a popular FS technique, which optimizes a measure corresponding to the feature subset to obtain the optimal feature subset. This measure is directly related to the classifier learning process. But, in filter based FS methods [21], the measure to be optimized is not directly related to the classifier

☆ This paper has been recommended for acceptance by Dr. Y. Liu.

* Corresponding author. Tel/fax.: +91 3325752323.
E-mail address: swagatamdas19@yahoo.co.in (S. Das).

http://dx.doi.org/10.1016/j.patrec.2015.07.007
0167-8655/

learning process. Wrapper methods may obtain better performance, but requires much higher computational time than filter based algorithms. In [51], a comparative study on wrapper and filter based FS techniques was presented. Some works like [39,27] attempted to combine the wrapper and filter approaches and reported better results than the conventional methods.

Several fuzzy and rough set based FS and classifier design techniques have been proposed over the years. Few works relating to this topic may be found in [24,25,34]. Maji and Garai [37] used a fuzzy-rough set approach for feature selection. Chiang and Ho [11] used the concept of rough sets to design a classifier. Zhou and Khotanzad [67] proposed a fuzzy rule based classifier designing technique using the Genetic Algorithm (GA). A rule- based classifier using fuzzy-rough sets is proposed in [66]. An algorithm for FS and fuzzy rule extraction method for classification was proposed by Chen et al. [10].

Evolutionary Algorithms (EAs) have also been extensively used for FS. The basic mechanism of such algorithms is to optimize an objective function defined for FS, to obtain the best feature subset. Muni et al. [42] presented a simultaneous feature selection and tree-based classifier design algorithm based on genetic programming. Later Diao and Shen [16] presented a Harmony Search (HS) based algorithm for FS. They used some common classifiers like VQNN (Vaguely Quantified Nearest Neighbor, a noise tolerant fuzzy-rough classifier) and SVM (Support Vector Machine) to calculate the classification error, but did not conclude a single classifier, which is most compatible with their algorithm. A comparative study of some prominent metaheuristic approaches including GA and memetic algorithms for FS may be found in [54]. Nakamura et al. [43] proposed a Binary Bat Algorithm (BBA) for FS (BBA-FS), where they used a wrapper approach to combine BBA with optimum forest classifier to yield the feature subset by maximizing classification accuracy. In [4] a Hamming distance based binary Particle Swarm Optimization (PSO) algorithm was employed to obtain the important feature subset in gene expression data.

Cordón et al. [12] proposed a Multi-objective GA (MOGA) based FS technique and a Fuzzy Rule Based Classification System (FRBCS), but their work lacked extensive simulations. Spolaor et al. [52] developed an MOGA based method for FS. They used combinations of different measures like class correlation, inconsistent example pairs, entropy, etc., as their objective functions and optimized them to yield the optimal feature vectors. However, the classification accuracy offered by the different pairs of measures were not consistent over different datasets and the authors were unable to propose a single combination of measure, which would yield the best results for a wide range of datasets. Xue et al. [59] proposed a new multi-objective approach based on PSO for FS. They simultaneously maximized classification accuracy and minimized the cardinality of the feature subset. But their results reflect that they were unable to achieve a set of non-dominating solutions. This can be inferred from the fact that classification accuracy and number of features selected are not monotonically conflicting, that is if one of them decreases, the other does not increase monotonically. Instead, an optimal number of features are required to obtain maximum accuracy in classification for most datasets. A very similar work was done by the Cervante et al. [7], where they used binary PSO scheme to optimize the classification performance and number of features selected. Recently, Xue et al. [60] proposed a Multi Objective Differential Evolution based FS (DEMOFS) technique, which minimizes the classification error rate and the number of features selected. Instead of presenting a single feature subset, their algorithm offers a set of solutions with varying number of features and classification accuracy. Other multi-objective optimization based FS techniques have been proposed in [20,41,17].

Optimizing inter- and intra-class distance measures is a common underlying technique for several FS as well as classification algorithms. However, use of this idea in the framework of a multi-objective evolutionary optimization algorithm with formulations similar to our algorithm is unknown till date, to the best of our knowledge. For example, among the commonly used classifiers, Linear Discriminant Analysis (LDA) uses the concept of inter- and intra-class distances. James et al. [23] optimized the area of overlap between the feature-wise inter- and intra-class distances for efficient selection of features. Liu et al. [31] used a similar concept and formulated their problem as a constrained Linear Programming (LP), but solved it by using a gradient descent approach, whose major drawback is trapping at local optima. Michael et al. [38] used intra- and inter-class distance ratio, as one of the criteria for iterative feature selection.

In this paper, a simultaneous feature selection and weighting method is proposed. Most feature selection algorithms focus only on selection rather than selection and weighting, where the later has the potential to further enhance the classifier performance [47]. Here, the formulated inter-class and intra-class distance measures are maximized and minimized simultaneously by using a Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D). Currently MOEA/D appears as one of the most competitive algorithms for multi-objective optimization. The population of MOEA/D comprises of vectors containing the candidate feature weights, which are varied according to the algorithm to optimize the inter-class and intra-class distance measures discussed subsequently. A repair mechanism is used to increase the probability of choosing lesser number of features. Also a penalty function is augmented with the fitness functions to reduce the number of features selected. After convergence to optima using MOEA/D, a fuzzy membership based technique [1] is incorporated to choose only a single feature subset (representing the best compromise solution) out of the Pareto optimal solution set. Finally it is shown that by using the k-NN classifier on the optimal feature subset, it is possible to produce higher classification accuracy than some of the state-of-art algorithms for FS.

The paper is organized as follows: Section 2 presents the proposed feature selection and weighting scheme. Section 3 provides a brief outline of MOEA/D followed by the description of the complete repairing and best compromise solution selection technique. Experimental settings and comparative studies have been presented and discussed in Section 4. Finally the paper is concluded in Section 5.

2. Feature selection and weighting

Appropriate classification of data points require the data points belonging to different classes be ideally far apart from each other and those belonging to the same class should be as near as possible. The extents of such inter- class separability and intra- class nearness depends on the features of the data points. Some of the features may be in accordance with this principle, (which may be selected) whereas the others may not (which can be eliminated as they may lead to poor classification). Thus, only a subset of the available features needs to be chosen depending on the inter- and intra-class distance measures of the resulting data points. Again, the features selected may be weighted in such a manner that the data points belonging to non-identical classes will be shifted far away and those of the same class will be constricted. Thus, the goal of the proposed feature selection algorithm is two-fold: selection of features as well as weighting them in such a way that the intra-class distance as well as the inter-class distance of the resultant data points can be optimized simultaneously.

Consider a dataset $\mathbb{Z}$ of size $m \times n$, m being the number of features of each data point and n being the number of data points. Then, each point $z_{i,j}$ denotes the ith feature of the jth data point. Also let the label of each data point of $\mathbb{Z}$ be denoted by the vector $\mathbb{L}$, such that l_j denote the label of the jth data point. It may be noted here that the different features of a data point can have values of different orders.

For this reason, the features are normalized within a range of [1, 10] following [47] as:

$$x_{i,j} = 1 + 9.\left(\frac{z_{i,j} - \min\limits_{1\le k\le n} z_{i,k}}{\max\limits_{1\le k\le n} z_{i,k} - \min\limits_{1\le k\le n} z_{i,k}}\right). \tag{1}$$

Any operations hereafter will be executed on the normalized data matrix $\mathbb{X}$. Let's consider the feature selection and weighting vector as $\boldsymbol{W}$, such that w_i is the feature selection or weighting factor for the ith feature, which may be defined as follows,

$$w_i = \begin{cases} 0, & if\ the\ feature\ is\ rejected, \\ [1, \mathcal{A}], & if\ the\ feature\ is\ selected. \end{cases} \tag{2}$$

Then, each feature may have a maximum weight of $\mathcal{A}$ and a minimum weight of 0 when the feature is eliminated. $\mathcal{A}$ is considered to be 10 in our algorithm. After applying the weights, the jth data point may be represented as:

$$\boldsymbol{Y}_j = \boldsymbol{W} \odot \boldsymbol{X}_j, \tag{3}$$

where $\odot$ denotes the element-wise multiplication operator between two vectors. Using the above notations, the l_1-norm distance between p^{th} and q^{th} data points may be formulated as:

$$\begin{aligned} d(\boldsymbol{Y}_p, \boldsymbol{Y}_q) &= \sum_{i=1}^{m} \left| y_{i,p} - y_{i,q} \right| \\ &= \sum_{i=1}^{m} \left| w_i x_{i,p} - w_i x_{i,q} \right| = \sum_{i=1}^{m} w_i \left| x_{i,p} - x_{i,q} \right|. \end{aligned}$$

$$\text{thus,} \quad d(\boldsymbol{Y}_p, \boldsymbol{Y}_q) = \boldsymbol{W}^T |\boldsymbol{X}_p - \boldsymbol{X}_q|. \tag{4}$$

Using Eq. (4) the total intra class distance may be formulated as:

$$\begin{aligned} D_{intra} &= \sum_{p=1}^{n} \sum_{\substack{q=p+1 \\ \forall l_p = l_q}}^{n} d(\boldsymbol{Y}_p, \boldsymbol{Y}_q) \\ &= \sum_{p=1}^{n} \sum_{\substack{q=p+1 \\ \forall l_p = l_q}}^{n} \boldsymbol{W}^T |\boldsymbol{X}_p - \boldsymbol{X}_q| = \boldsymbol{W}^T \sum_{p=1}^{n} \sum_{\substack{q=p+1 \\ \forall l_p = l_q}}^{n} |\boldsymbol{X}_p - \boldsymbol{X}_q| \end{aligned}$$

$$\text{thus,} \quad D_{intra} = \boldsymbol{W}^T \boldsymbol{\Delta}^{\boldsymbol{intra}}, \tag{5}$$

where $\boldsymbol{\Delta}^{\boldsymbol{intra}}$ is a vector containing the feature-wise intra class distance. It may be noted that there is a reason behind this approach of piggybacking the intra-class distance feature-wise and then applying the feature weight vector at the end, instead of finding the pair-wise distance of the data points and then summing them. The advantage of this method will be discussed in Section 4.4. The inter-class distance may be formulated in a similar manner as follows:

$$\begin{aligned} D_{inter} &= \sum_{p=1}^{n} \sum_{\substack{q=p+1 \\ \forall l_p \neq l_q}}^{n} d(\boldsymbol{Y}_p, \boldsymbol{Y}_q) \\ &= \sum_{p=1}^{n} \sum_{\substack{q=p+1 \\ \forall l_p \neq l_q}}^{n} \boldsymbol{W}^T . |\boldsymbol{X}_p - \boldsymbol{X}_q| = \boldsymbol{W}^T \sum_{p=1}^{n} \sum_{\substack{q=p+1 \\ \forall l_p \neq l_q}}^{n} |\boldsymbol{X}_p - \boldsymbol{X}_q| \end{aligned}$$

$$\text{thus,} \quad D_{inter} = \boldsymbol{W}^T \boldsymbol{\Delta}^{\boldsymbol{inter}}, \tag{6}$$

where $\boldsymbol{\Delta}^{\boldsymbol{inter}}$ is a vector containing the feature-wise intra class distance. In order to obtain an optimal subset of the selected and weighted features, the intra-class distance should be minimized and the inter-class distance should be maximized, by varying $\boldsymbol{W}$.

In order to increase the affinity towards selecting lesser number of features, a penalty function may be added to both the intra-class and inter-class distances. According to the concept of penalty function, higher the number of selected features, higher will be the extent of penalization of the objective function. Thus, the objective functions required to be optimized are:

$$\begin{aligned} F_1(\boldsymbol{W}) &= D_{intra} + \lambda_1 [min\{1, \boldsymbol{W}\}]^T \boldsymbol{1}, \\ F_2(\boldsymbol{W}) &= -D_{inter} + \lambda_2 [min\{1, \boldsymbol{W}\}]^T \boldsymbol{1}, \end{aligned} \tag{7}$$

where λ_1 and λ_2 are the coefficients of penalty, $min\{\}$ is an element wise minimum operation on two vectors, and $\boldsymbol{1} = [1, 1, \ldots, 1]^T$ of the dimensions of $\boldsymbol{W}$. As D_{inter} should be maximized, the negative sign before the same indicates that F_2 should be minimized. Thus, both F_1 and F_2 should be minimized in order to obtain the optimal feature selection and weight vector. This may be stated as,

$$\begin{aligned} &Minimize\ \boldsymbol{F}(\boldsymbol{W}) = [\, F_1(\boldsymbol{W}),\ F_2(\boldsymbol{W})]^T \\ &subjected\ to\ \boldsymbol{W} = [w_1,\ w_2, \ldots, w_m]^T \in 0 \cup [1, \mathcal{A}]. \end{aligned} \tag{8}$$

It may be noted that all the elements of vectors $\boldsymbol{W}$, $\boldsymbol{\Delta}^{\boldsymbol{inter}}$ and $\boldsymbol{\Delta}^{\boldsymbol{intra}}$ are non-negative. Thus, if the elements of $\boldsymbol{W}$ are incremented, then by Eq. (6), $\boldsymbol{D}_{\boldsymbol{inter}}$ will increase, leading to a decrease in $\boldsymbol{F}_2(\boldsymbol{W})$. But, in such a case, $\boldsymbol{D}_{\boldsymbol{intra}}$ will also increase leading to an increase of $\boldsymbol{F}_1(\boldsymbol{W})$. Here minimization of only $\boldsymbol{F}_2(\boldsymbol{W})$ does not guarantee the minimization of $\boldsymbol{F}_1(\boldsymbol{W})$. Thus, Eq. (8) presents a Multi-objective Optimization Problem (MOP) which is solved as discussed next.

3. Multi-objective optimization

An MOP requires the optimization of multiple objective functions (usually in conflict) with or without the presence of constraints. Generally, these objective functions are in conflict with each other, i.e. optimization of one objective function towards its optimal value leads to the departure of another objective function away from its optimal value. The problem of comparing among candidate solutions is solved by the concept of Pareto-optimality, where a best trade-off among these objective functions is obtained, thus returning multiple solutions after optimization, known as the Pareto optimal set. Then, depending on the application, one solution is chosen from the Pareto optimal front.

In the proposed method, the well known MOEA/D ([64,65]) is used to obtain the Pareto optimal front. The algorithm decomposes an MOP into a number of single objective problems by the Tchebycheff approach, using equally spaced weight vectors to weight the objective functions. For detailed steps of MOEA/D, refer to [64].

In the proposed method, the initialization of the N feature selection and weighting vectors ($\boldsymbol{W}$) for the N sub problems of MOEA/D is done randomly within $[0, \mathcal{A}]$. Thereafter, MOEA/D contains a reproduction stage for gradual evolution of the population of weight vectors ($\boldsymbol{W}$) towards optima. In our method, this stage comprises of two steps namely: mutation and crossover. These two operations are carried out according to the mutation and crossover steps of DE/rand/bin/1, one of the simplest variant of Differential Evolution (DE) [53,13], a powerful single objective optimization algorithm. After every reproduction stage of MOEA/D, a repair step is carried out over the newly created offspring, so that it satisfies the constraints of the optimization problem. Consider $\boldsymbol{W} = [w_1, w_2, \ldots, w_m]^T$ to be the vector produced after reproduction. Then the ith element of the repaired vector $\boldsymbol{W'}$ may be computed as:

$$w'_i = \begin{cases} A, & if\ w_i > A, \\ 0, & if\ w_i < 1, \\ w_i, & otherwise, \end{cases} \tag{9}$$

$\forall\, i \in [1, m]$. It may be noted that when w_i is assigned a value in [0, 1), which is much smaller compared to its maximum possible value $\mathcal{A}$ (10 in the proposed method), it indicates that the weight to the ith feature is very small and may be considered as zero. This is done by the 2nd condition of Eq. (9), thus increasing the probability of lowering the number of selected features. After repair, the new offspring

Table 1
Summary of the datasets.

Name of dataset	No. of classes	No. of features	Size of the dataset
Iris	03	004	0150
WBDC	02	030	0569
Spambase	02	057	4701
Heart	02	006	0270
Glass	06	009	0214
WBC	02	009	0699
Ionosphere	02	034	0351
Arrhythmia	16	195	0452
Multiple features	10	649	2000
Breast cancer	02	010	0683
Australian	02	014	0690
German number	02	024	1000
DNA	02	180	2000
Wine	03	013	0178
Vehicle	04	018	0846
Waveform	03	040	5000
Zoo	07	017	0101
Hillvalley	02	100	0606
Sonar	02	060	0208
Musk 1	02	166	0476

is either considered as a parent for the next iteration or discarded depending on whether it is a better solution than its parent. Iterating the above steps a number of times lead to convergence to a set of Pareto optimal solutions, having different weights on the objective functions.

In order to obtain a single weight vector from the Pareto optimal front, the concept of the best compromise solution [1] is used. In this method, fuzzy membership values between 0 and 1 are assigned to each objective values for all the solutions of the Pareto optimal set. Thereafter, the solution having the highest total normalized fuzzy membership value may be selected as the best compromise solution. The fuzzy membership value for the kth member of the Pareto optimal set and for the ith objective value may be expressed as follows:

$$\mu_i^k = \begin{cases} 1 & if\ F_i^k \le F_i^{min}, \\ \frac{F_i^{max} - F_i^k}{F_i^{max} - F_i^{min}} & if\ F_i^{min} < F_i^k < F_i^{max}, \\ 0 & if\ F_i^k \ge F_i^{max}, \end{cases} \tag{10}$$

where F_i^{min} and F_i^{max} are the minimum and maximum objective values of all the Pareto optimal solutions for the ith objective function. The memberships of the kth solution, for the Obj number of objective functions ($Obj = 2$ in the proposed method: F_1, F_2) is represented as $\mu^k = [\mu_1^k, \mu_2^k, \ldots, \mu_{Obj}^k]^T$. It may be inferred from Eq. (10), that μ_i^k denotes the degree by which the kth solution satisfies the ith objective function. The degree of 'achievement' by the kth solution to optimize the objective functions, may be computed and normalized as follows,

$$\mu^k = \frac{\left[\mu^k\right]^T 1}{\sum_{l=1}^{N} \left[\mu^l\right]^T 1}, \tag{11}$$

with $k \epsilon [1, N]$, N being the number of Pareto optimal solutions obtained from MOEA/D and $1 = [1, 1, \ldots, 1]^T$ of length Obj. Thus, μ^k may be considered as a single objective function with normalized degree of 'achievement' by the kth solution. Hence, finding the maximum of μ^k, $\forall k$ gives the index of the best compromise solution among the set of Pareto optimal solutions. As each of the N Pareto optimal solutions yielded by MOEA/D corresponds to a feature selection and weighting vector ($\boldsymbol{W}^k$), the vector that is chosen to be optimal ($\boldsymbol{W}^*$) satisfies the following,

$$k^* = \max_k \mu^k, \quad \boldsymbol{W}^* = \boldsymbol{W}^{k^*}. \tag{12}$$

Thus, only those features which have been chosen in $\boldsymbol{W}^*$ forms the new subset of features after applying the corresponding weights on them. Hereafter, any classification algorithm may be applied on this subset of weighted features. Pseudo-code for the entire algorithm is provided below.

Input ← Training dataset
1. Normalize the training dataset
2. Compute $\boldsymbol{\Delta}^{intra}$ and $\boldsymbol{\Delta}^{intra}$
3. Randomly initialize N Feature Selecting-Weighting vector ($\boldsymbol{W}$) which comprise the population of MOEA/D
Repeat (4) to (8) for each individual of the MOEA/D population
4. Mutation and Crossover to create new feature selecting-weighting vectors
5. Repair the newly generated selecting-weighting vectors
6. Compute the Objective values
8. Update solutions
9. *If* termination criterion not reached, go to (4) *else* go to (10) *end if*
10. Use Best Compromise Solution technique to find the optimal feature selection-weighting vector ($\boldsymbol{W}^*$) from the Pareto optimal set
11. Normalize the test dataset. Select the features and apply $\boldsymbol{W}^*$ on the test and training dataset to select and weight only the required features. Then use k-NN ($k = 5$) to classify the data points of test dataset
Output ← Classification Accuracy

Table 2
Setup for MOEA/D.

Number of weight vectors	100
Dimension of the search space	Number of features of the dataset
Scaling Factor (F)	0.70
Crossover Rate (Cr)	0.95
Upper bound of search space	10
Lower bound of search space	0
Termination criteria	150 iterations

4. Experimental results

4.1. Datasets

The 20 real world datasets used for comparison and validation are taken from the UCI and LIBSVM database ([8,35]) and they are summarized in Table 1. It may be noted that in

a. *Heart* dataset: Chen et al. [10] used the 6 real valued features out of total 13 features,
b. *Arrhythmia* dataset: Mitra et al. [40] used the 195 real valued features out of total 279 features.

In order to have a proper comparison, we have also used the same features as used by the above mentioned authors.

4.2. Parameter setup for MOEA/D

A choice of the set of parameters for proper convergence of the MOEA/D algorithm is a difficult and challenging task. After a series of experimentations on the proposed algorithm, and considering the suggestions presented in ([53,64,65]), a set of parameters is found to deliver appropriate result and they are listed in Table 2. A higher value of F has been used in this algorithm for faster exploration of the search space. In the proposed method, F within 0.6 and 0.8 produced best results and thus it may be considered to be 0.7. A high value of Cr has been considered as it produced best results in and around that value. Requirement in high value of Cr depicts the requirement in higher diversity among the populations of vectors. In such cases, a low value of Cr near 0 may lead to slower rate of convergence to optima. The number of weight vectors and thus the population size can be chosen in between [100, 200], although not much difference was noted in classification accuracy with its change from 100 to 200. But if the population size is significantly lesser than 100, there might be a risk of missing out potential solutions, as the number of subproblems the MOEA/D algorithm decomposes the multi-objective optimization problem is equal to the population size. Next, the upper

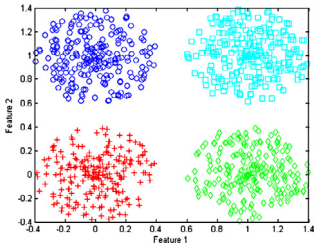

Fig. 1.1. Dataset "Simple Class".

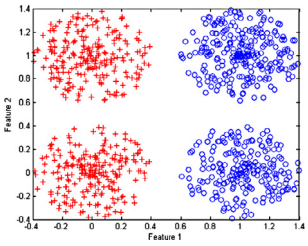

Fig. 1.2. Modified dataset "Simple Class".

bound of the search space may be considered to be 10 since values higher than that may decrease the probability of a feature weight in $[0, 1)$, thus increasing the probability of selecting a feature (see Eq. (9)). Again, the upper bound lesser than 10 will restrict the weights offered to the features and may hamper the actual objective of feature weighting, which is to scale the features for enhanced classification. The lower bound should be kept at 0 in order to comply with the repair and feature selection mechanism of the proposed method. The termination criteria may be selected on a trial basis, depending on the average number of iterations required for a dataset to converge and this has been considered to be 150 for all the results provided in this paper. After a series of simulations by changing λ_1 and λ_2 of (7) within the range [200, 1000], a combination of $\lambda_1 = \lambda_2 = 500$ is found to produce the best results over a wide range of datasets considered here.

4.3. Comparisons

4.3.1. Testing with synthetic datasets

In order to classify the test data using the selected and weighted feature vectors, k-NN classifier is used, with $k = 5$. At first to demonstrate the ability of the proposed algorithm, we tested our algorithm on a synthetic dataset called "Simple Class" (available in MATLAB 2012a), plotted in Fig. 1.1. It may be noted from the plot that there exist two feature vectors and none can be eliminated, as both are relevant for proper classification. The optimal feature weights are $[\,8.62,\ 10.00]^{\,T}$ with 100% accuracy on test data. Slight difference in the weights offered to the feature vectors is due to the slight difference in the symmetricity among the classes.

Now two pairs of classes of the "Simple Class" dataset are merged, resulting in a new dataset, as shown in Fig. 1.2, where red and blue represent the two classes. It may be noted that feature 1 alone is capable of classifying the data points with high accuracy. After running our proposed algorithm on this dataset, the optimal weight vector is $[9.67,\ 0]^T$ with 100% accuracy on test data.

4.3.2. Testing with real world datasets

The proposed algorithm is compared with the following FS methods:

- Two wrapper based feature selection method – Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) [29]
- Feature similarity technique (FS-FS) [40],
- Fuzzy rule based (FR-FS) method [10],
- Bat algorithm based method (BBA-FS) [43]
- Ranking Based method (RB-FS) [55],
- Differential Evolution based Multi-Objective Feature Selection (DEMOFS) [60] and
- Multi-objective Genetic Algorithm (MOGA-FS) based technique [52].

Table 3
Comparison with SFS and SBS FS algorithm.

Dataset	Avg. classification accuracy (Avg. No. of features selected)		
	SFS	SBS	Proposed
Iris	93.33 (3.30)	93.33 (2.60)	**97.27** (**2.00**)
WBDC	90.12 (13.9)	89.77 (17.8)	**94.06** (**13.5**)
Spambase	87.40 (35.7)	87.01 (37.3)	**88.48** (**26.0**)
Heart	65.12 (**1.90**)	62.59 (2.30)	**80.00** (5.30)
Glass	63.10 (5.80)	63.61 (7.00)	**67.76** (**4.40**)
WBC	95.99 (6.40)	95.13 (7.30)	**96.05** (**4.20**)
Ionosphere	**88.70** (**7.20**)	85.92 (9.10)	88.31 (11.5)
Arrhythmia	59.89 (89.4)	58.04 (**49.2**)	**65.77** (100)
Multiple features	90.26 (**210**)	91.22 (305)	**97.88** (270)
Breast cancer	95.14 (6.10)	94.85 (6.10)	**96.53** (**4.30**)
Australian	83.02 (3.70)	82.83 (**3.00**)	**84.64** (4.70)
German number	68.20 (12.2)	65.80 (10.8)	**71.30** (**10.5**)
DNA	82.16 (**18.8**)	82.33 (20.6)	**83.08** (71.8)
Wine	91.44 (**6.00**)	91.44 (7.50)	**96.05** (6.90)
Vehicle	**68.56** (10.8)	67.34 (10.7)	65.26 (**9.10**)
Waveform	77.82 (18.4)	78.46 (18.3)	**83.65** (**16.0**)
Zoo	94.89 (**9.00**)	**98.00** (13.0)	95.42 (11.0)

Table 4
Comparison with fuzzy rule based FS.

Dataset	Avg. classification accuracy (Avg. number of features selected)	
	Fuzzy rule based FS	Proposed
Iris	93.84 (3.39)	**97.27** (**2.00**)
WBDC	93.55 (**2.14**)	**94.06** (13.5)
Heart	68.64 (**3.94**)	**80.00** (5.30)
Glass	61.54 (6.96)	**67.76** (**4.40**)
Wine	95.51 (**4.38**)	**96.05** (6.90)
Ionosphere	84.40 (**4.33**)	**88.31** (11.5)

Table 5
Comparison with RB-FS.

Dataset	Avg. classification accuracy	
	RB-FS	Proposed
Multiple features	89.88	**97.88**
Spambase	83.21	**88.48**

Comparison with SFS and SBS based FS methods with classification error as the criterion is presented in Table 3, using the 10 Fold Cross Validation (FCV). It may be noted that these two algorithms are computationally inefficient and is of the order $O(n^2)$. Next, Table 4 presents the comparison with FR-FS, where the authors used 10 FCV to compute the classification accuracy, and we follow the same convention. Table 5 presents the comparison with RBFS. Number of features selected cannot be provided as the authors of this algorithm did not mention it. Table 6 presents the comparison with FS-FS, where again we use the 10 FCV.

Comparison of the proposed algorithm with BBA-FS is summarized in Table 7. The authors of BBA-FS used 30% of a dataset for testing, and the rest for training and validation. We here also resort to the same scheme.

We further compare our algorithm with DEMOFS [60] and the results are summarized in Table 8. This table presents two sets of results for DEMOFS: 'A' corresponds to the maximum classification accuracy

Table 6
Comparison with FS-FS.

Dataset name	Avg. classification accuracy (Avg. number of features selected)	
	Feature Similarity	Proposed
Iris	96.80 (**2.00**)	**97.27** (**2.00**)
WBC	95.56 (**4.00**)	**96.05** (**4.20**)
Ionosphere	78.77 (16.0)	**88.31** (**11.5**)
Arrhythmia	58.93 (**100**)	**65.77** (**100**)
Multiple features	78.34 (**325**)	**97.88** (270)
Waveform	75.20 (20.0)	**83.65** (**16.0**)

Table 7
Comparison with BBA-FS.

Dataset name	Avg. classification accuracy (Avg. number of features selected)	
	BBAFS	Proposed
Breast cancer	96.31 (07)	**96.34** (**4.50**)
Australian	77.25 (08)	**81.59** (**4.70**)
German number	70.24 (**07**)	**70.97** (11.3)
DNA	83.02 (**18**)	**83.08** (71.8)

Table 8
Comparison with DEMOFS.

Dataset name	Classification accuracy (Number of features selected)		
	DEMOFS		Proposed
	A	B	
Wine	89.65 (**6.00**)	89.65 (**6.00**)	**96.05** (6.90)
Australian	77.30 (**4.00**)	77.30 (**4.00**)	**84.64** (4.70)
Zoo	95.38 (**11.0**)	85.00 (**10.0**)	**95.42** (**11.0**)
German number	70.10 (**1.00**)	67.94 (12.0)	**71.30** (10.5)
Hill Valley	**60.46** (**26.0**)	57.04 (43.0)	57.50 (40.0)
Sonar	78.60 (**10.0**)	76.67 (22.0)	**82.74** (20.0)
Musk1	**83.45** (**58.0**)	82.30 (90.0)	81.52 (59.3)

and 'B' corresponds to the classification accuracy with the same number of chosen features as that of the proposed method, both chosen from the set of solutions offered by DEMOFS.

Next we compare our algorithm with the Multi-Objective Genetic Algorithm based FS (MOGA-FS) technique [52]. The authors of MOGA-FS have used ten combinations of five distinct objective measures, and optimized them using MOGA to obtain the optimal features. The measures used by them were - Inconsistent Example Pairs (IP), Attribute Class Correlation (AC), Inter-Class Distance (IE), Representation Entropy (RE) and Laplacian Score (LS). Spolaor et al. used a 10 FCV to compute the average classification accuracy and we do the same here. However, the results produced by the different combination of measures in MOGA-FS were not consistent and varied highly with a change of datasets. Also they did not propose a single combination of measure, which may perform well for a wide variety of datasets. For this reason, we have used a rank based method for comparison. The classification accuracy produced by MOGA-FS and the proposed algorithm is ranked from 1 to 11 for each dataset (with a rank of 1 assigned to the method having highest classification accuracy and increasing rank for decreasing classification accuracy). Then an average is taken over the ranks assigned for different datasets. Thus, the method having the least average rank may be considered to produce best results among others. The same ranking procedure is employed for Percentage of Features Reduction (PFR).

Table 9 presents the results of the ten methods of MOGA-FS and the proposed algorithm along with their ranks for 3 datasets. It may be seen that the proposed algorithm has the best average rank of 2.67 among the other methods of MOGA-FS for classification accuracy. It may also be observed that the AC+LS method ranks 1st with respect to PFR, but last according to classification accuracy. This suggests that decrease in number of features selected does not guarantee increase in classification accuracy; instead, an optimal number of features are required for better classification.

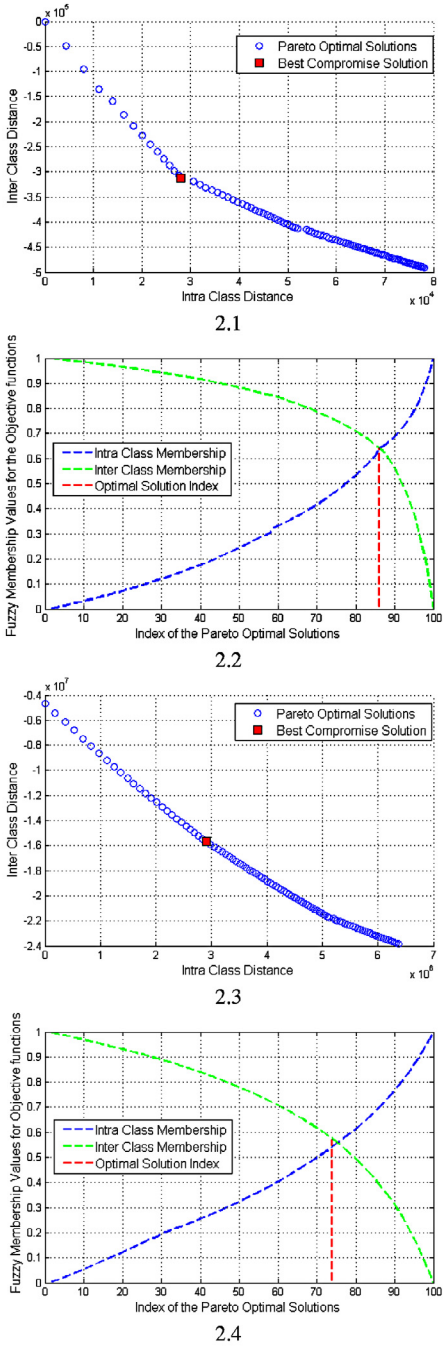

Fig. 2. Pareto optimal solution set and their corresponding fuzzy membership values for finding best compromise solution for Iris (2.1–2.2) and Breast Cancer (2.3–2.4) datasets.

The Pareto optimal solution set along with their fuzzy membership values for finding best compromise solution are presented for two datasets in Fig. 2.

4.4. Analysis of complexity

The proposed algorithm is an iterative based method of gradual evolution towards convergence. A deterministic number of iterations

Table 9
Comparison with MOGA-FS with respect to classification accuracy and percentage of feature reduction (PFR).

Feature selection procedure	Dermatology		Vehicle		Wine		Average rank (Accuracy (PFR))
	Accuracy (PFR)	Rank	Accuracy (PFR)	Rank	Accuracy (PFR)	Rank	
IE + AC	91.03 (23.14)	06 (09)	72.43 (06.85)	04 (08)	93.24 (37.69)	02 (09)	04.00 (8.67)
IE + IP	93.28 (00.00)	05 (11)	**74.00** (00.00)	**01** (10)	92.12 (00.00)	03 (10)	03.00 (10.3)
IE + RE	94.11 (01.76)	03 (10)	73.96 (00.19)	02 (09)	92.12 (00.00)	04 (10)	03.00 (9.67)
IE + LS	90.67 (33.53)	07 (08)	72.74 (14.07)	03 (07)	89.35 (64.62)	07 (07)	05.67 (7.33)
AC + IP	62.75 (85.39)	10 (02)	69.99 (62.59)	05 (05)	89.93 (78.46)	06 (03)	07.00 (3.33)
AC + RE	93.46 (40.59)	04 (07)	63.84 (88.89)	09 (03)	90.02 (69.74)	05 (06)	06.00 (4.67)
AC + LS	65.38 (**94.12**)	11 (**01**)	52.49 (**94.44**)	10 (**01**)	83.10 (**88.46**)	10 (**01**)	10.33 (**1.00**)
IP + RE	96.08 (52.35)	02 (04)	67.62 (76.30)	08 (04)	77.48 (69.23)	11 (04)	07.00 (4.00)
IP + LS	80.73 (71.67)	09 (03)	68.44 (63.33)	07 (03)	89.31 (83.85)	08 (02)	08.00 (2.67)
RE + LS	88.99 (43.73)	08 (06)	52.49 (**94.44**)	11 (**01**)	87.61 (69.23)	09 (05)	09.33 (4.00)
Proposed	**96.13** (45.76)	**01** (05)	65.26 (49.44)	06 (06)	**96.05** (46.90)	**01** (08)	**02.67** (6.33)

Table 10
Comparison of NSGAII and MOEA/D.

Dataset	NSGAII		MOEA/D	
	Accuracy (No. of Features)	Time (s)	Accuracy (No. of features)	Time (s)
Iris	96.03 (**2.00**)	05.08	**97.27** (**2.00**)	**01.56**
WBDC	**96.34** (30.2)	06.97	94.06 (**13.5**)	**02.10**
Spambase	88.25 (55.0)	60.17	**88.48** (**26.0**)	**35.22**
Heart	78.89 (12.0)	05.19	**80.00** (5.30)	**01.69**
Glass	66.77 (7.40)	05.51	**67.76** (**4.40**)	**01.62**
WBC	95.86 (8.60)	06.50	**96.05** (**4.20**)	**02.24**
Ionosphere	87.06 (33.0)	06.39	**88.31** (**11.5**)	**01.83**
Arrhythmia	62.87 (192)	13.18	**65.77** (**100**)	**02.48**
M. features	**97.88** (576)	63.12	**97.88** (**270**)	**31.23**
Br. cancer	96.05 (9.80)	06.39	**96.53** (**4.30**)	**02.27**
Australian	84.45 (13.3)	06.42	**84.64** (**4.70**)	**02.22**
German No.	71.02 (23.0)	08.10	**71.30** (**10.5**)	**03.04**
DNA	80.18 (177)	27.15	**83.08** (**71.8**)	**10.75**
Wine	95.90 (11.2)	05.60	**96.05** (**6.90**)	**01.67**
Vehicle	**68.06** (17.3)	06.70	65.26 (**9.10**)	**02.30**
Waveform	80.14 (39.4)	35.01	**83.65** (**16.0**)	**39.82**
Zoo	94.00 (13.9)	**05.68**	**95.42** (**11.0**)	06.50

β is used for all the datasets. In each iteration, the algorithm calls for computation of the objective functions in (7), this in turn may be computed by using Eqs. (5) and (6). Computing Δ^{intra} and Δ^{inter} has a time complexity of $O(n^2)$ (n being the number of data points). But, these two vectors are computed only once at the beginning of the algorithm, by piggybacking the distances for each feature separately, instead of computing the pair-wise distance between data points and summing up them. So, during each iteration, computation of the objective functions requires only two vector inner-products (for two objective functions), which is of the order l (the number of features in each data point). Thus, the running time for computing the objective function for β iterations is $T(n) \approx 2\beta l$. Now, running time and time complexity are related as,

$$T(n) = O(f(n)),$$
$$if\ \exists\ constant\ c\ such\ that\ T(n) \leq cf(n), \quad \forall n \geq n_o.$$

Using this formulation, time complexity of the iterative part of the proposed algorithm may be found to be of $O(l)$ (with $c = 2\beta, \forall n \geq 1$). Thus, summing this along with the time complexity of computing inter- and intra-class distances, the worst case complexity of the proposed method is $O(n^2 + l)$. But as the inter- class and intra- class distances are required to be computed only once, the time required is smaller compared to the other algorithms, whose complexity orders are discussed next.

The FS algorithm proposed by Mitra et al. [40] has a complexity of $O(nl^2)$, where l is the number of features. In the BBA-FS method, due to its iterative nature, a constant β (for a finite number of iterations), appears in its running time, which is $T(n) = \beta n^2$, leading to a runtime complexity of $O(n^2)$. The authors of MOGA-FS presented a comparative study of optimizing 10 different combinations of 5 different objective functions. The running times for these combinations are $T(n) = \beta * (n^2,\ n^2,\ l^2 + n,\ n^2 l,\ n^2 + l,\ n^2 + l^2,\ n^2 l,\ n^2 + l^2,\ n^2 l,\ n^2 l + l^2)$ in the order listed in Table 7. Again, the constant factor β appears due to the iterative optimization process of the algorithm. Thus, the corresponding time complexities are $O(n^2)$, $O(n^2)$, $O(l^2 + n)$, $O(l^2 + n)$, $O(n^2 l)$, $O(n^2 + l)$, $O(n^2 + l^2)$, $O(n^2 l)$, $O(n^2 + l^2)$, $O(n^2 l)$ and $O(n^2 l + l^2)$ respectively, in the order listed in Table 9. n is the number of data points and l is the number of features of the dataset. It may be noted that although the time complexity of the proposed method is comparable and even better than the other methods, the running time of the proposed algorithm is much smaller than the other algorithms due to the piggybacking of feature wise inter- and intra-class distance measures. Due to differences in the simulation environment, operating systems and processor speeds on which the competing algorithms were simulated by the respective authors, we have refrained from mentioning the computational time of the other methods.

MOEA/D has been used in the proposed method due to its lower computational time and better selection of a non-dominated set of solutions. This is reflected in Table 10, where it is compared with NSGA-II [15], a well known evolutionary multi-objective optimizer, using the same set of objective functions as in the proposed method. It may be viewed that the number of selected features and time consumed by MOEA/D is much lesser than by NSGA-II. Both the methods have been simulated under similar system conditions.

5. Conclusions

In this paper, two measures - inter and intra class distance of a dataset are used for feature selection and weighting. The feature selection-weighting vectors form the population of MOEA/D. The two measures are optimized in order to obtain the optimal feature vectors. A penalty is added to the objective functions and a repair mechanism is used to reduce the number of features selected. The *k*-NN classifier is used for classification of the test set. Besides offering better classification accuracy, the proposed method is computationally efficient. Future work may focus on developing a classifier, which exploits the intra- and inter-class distance property of the selected and weighted subset of features.

Supplementary Materials

The dependence of classification accuracy with the number of best weighted features has been studied and graphical plots representing the same are included in the supplementary material for the datasets used in the article. Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2015.07.007.

References

[1] M.A. Abido, A novel multiobjective evolutionary algorithm for environmental/economic power dispatch, Electric Power Systems Research, vol. 65, Elsevier, 2003, pp. 71–81. 2003.
[2] G.H. Altun, H.J. Hu, S. Gremalschi, R.W. Harrison, Y. Pan, A feature selection algorithm based on graph theory and random forests for protein secondary structure predictionpp, Bioinf. Res. Appl., LNCS 4463 (2007) 590–600.
[3] A.A. Azofra, J.M. Benitez, J.L. Castro, Consistency measures for feature selection, J. Intell. Inf. Syst. 30 (2008) 273–292.
[4] H. Banka, S. Dara, A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation, Pattern Recogn. Lett. 52 (2015) 94–100.
[5] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Trans. Neural Netw. 5 (4) (1994) 537–550.
[6] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence, Elsevier, 1997, pp. 245–271.
[7] L. Cervante, B. Xue, L. Shang, M. Zhang, A Multi-objective feature selection approach based on binary pso and rough set theory, Evolut. Comput. Comb. Optim. LNCS 7832 (2013) 25–36.
[8] Chang, C.C. and Lin, C.J. (2001). Retrieved from LIBSVM – A Library for Support Vector Machines: www.csie.ntu.edu.tw/~cjlin/libsvm/.
[9] Y. Chang, R.J. Stanley, R.H. Moss, W.V. Stoecker, A systematic heuristic approach to feature selection for melanoma discrimination using clinical images, Skin Res. Technol 11 (2005) 165–178.
[10] Y.C. Chen, N.R. Pal, I.F. Chung, An Integrated mechanism for feature selection and fuzzy rule extraction for classification, IEEE Trans. Fuzzy Syst. 20 (4) (2012) 683–698.
[11] J.H. Chiang, S.H. Ho, A combination of rough-based feature selection and rbf neural network for classification using gene expression data, IEEE Trans. Nanobiosci. 7 (1) (2008) 91–99.
[12] O Cordón, M.J.D. Jesus, F. Herrera, L. Magdalena, P. Villar, A multiobjective genetic learning process for joint feature selection and granularity and contexts learning in fuzzy rule-based classification systems, Interpret. Issues Fuzzy Model. Stud. Fuzziness Soft Comput. (2003) 79–99.
[13] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evolut. Comput. 15 (1) (2011) 4–31.
[14] M. Dash, H. Liub, Consistency-based search in feature selection, Artif. Intell. Elsevier 151 (2003) 155–176.
[15] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolut. Comput. 6 (2) (Apr. 2002) 182–197.
[16] R. Diao, Q. Shen, Feature selection with harmony search, IEEE Trans. Syst., Man, Cybern.—Part B: Cybern. 42 (6) (2012) 1509–1523.
[17] A. Ekbal, S. Saha, M. Hasanuzzaman, Multiobjective approach for feature selection in maximum entropy based named entity recognition, in: Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 1, 2010, pp. 323–326.
[18] Y. Han, K. Park, Y.K. Lee, Confident wrapper-type semi-supervised feature selection using an ensemble classifier, in: 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce, (pp. 4581–4586), 2011.
[19] T. Hancock, H. Mamitsuka, Boosted Network Classifiers for Local Feature Selection, IEEE Trans. Neural Netw. Learn. Syst. 23 (11) (2012) 1767–1778.
[20] J. Handl, J. Knowles, Semi-supervised feature selection via multiobjective optimization, in: International Joint Conference on Neural Networks (ICJNN, pp. 3319–3326), 2006.
[21] J. Huang, S.J. Tong, X. XuA, Filter Approach to Feature Selection Based on Mutual Information, in: Proceedings of the 5th IEEE International Conference on Cognitive Informatics, 1, 2006, pp. 84–89.
[22] C.L. Huanga, C.J. Wang, A GA-based feature selection and parameters optimization for support vector machines, Expert Systems with Apps., 31, Elsevier, 2006, pp. 231–240.
[23] James, A.P. and Dimitrijev, S. (2012). Feature selection using nearest attributes. Retrieved from arXiv.org: http://arxiv.org/ftp/arxiv/papers/1201/1201.5946.pdf.
[24] J. Jensen, Q. Shen, New approaches to fuzzy-rough feature selection, IEEE Trans. Fuzzy Syst. 17 (4) (2009) 824–838.
[25] Jensen, R. (2005). Combining rough and fuzzy sets for feature selection. Thesis for PhD, School of Informatics, University of Edinburgh.
[26] W. Jiang, G. Er, Q. Dai, J. Gu, Similarity-based online feature selection in content-based image retrieval, IEEE Trans. Image Process. 15 (3) (2006) 702–712.
[27] P. Bermejo, J.A. Gámez, J.M. Puerta, A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets, Pattern Recogn. Lett. 32 (5) (2011) 701–711.
[28] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell., Elsevier 97 (1–2) (1997) 273–324.
[29] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recogn. 33 (1) (2000) 25–41.
[30] Y. Li, S.J. Hu, W.J. Yang, G. Sun, F. Yao, G. Yang, Similarity-based feature selection for learning from examples with continuous values, Adv. Knowl. Discov. Data Min. LNCS (2009) 957–964.
[31] Z. Liu, H. Bensmail, M. Tan, Efficient feature selection and multiclass classification with integrated instance and model based learning, Evolut. Bioinf. 8 (2012) 197–205.
[32] H. Liu, H.H. Motoda, Computational Methods of Feature Selection, Chapmane and Hall/CRC, 2009, pp. 4–7.
[33] H.L.H. Liu, J. Li, L. Wong, A comparative study of feature selection and classification methods using gene expression profiles and proteomic patterns, Genome Inf. 13 (2002) 51–60.
[34] Z. Lu, Z. Qin, Y. Zhang, J. Fang, A fast feature selection approach based on rough set boundary regions, Pattern Recogn. Lett. 36 (2014) 81–88.
[35] Machine Learning Repository. Retrieved from University of California, Irvine: http://archive.ics.uci.edu/ml/.
[36] P. Maji, S.K. Pal, Feature selection using f-information measures in fuzzy approximation spaces, IEEE Trans. Knowl. Data Eng. 22 (6) (2010) 854–867.
[37] P. Maji, P. Garai, Fuzzy–rough simultaneous attribute selection and feature extraction algorithm, IEEE Trans. Cybern. 43 (4) (2013) 1166–1177.
[38] M. Michael, W.-C. Lin, Experimental study of information measure and inter-intra class distance ratios on feature selection and orderings, IEEE Trans. Syst., Man Cybern., SMC 3 (2) (1973) 172–181.
[39] H. Min, W. Fangfang, Filter-wrapper hybrid method on feature selection, in: Proceedings of the 2nd WRI Global Congress on Intelligent Systems, 3, 2010, pp. 98–101.
[40] P. Mitra, C.A. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2004) 301–312.
[41] A. Mukhopadhyay, U. Maulik, An SVM-Wrapped multiobjective evolutionary feature selection approach for identifying cancer-microrna markers, IEEE Trans. Nanobiosci. (2013) PP 99.
[42] D.P. Muni, N.R. Pal, J. Das, Genetic programming for simultaneous feature selection and classifier design, IEEE Trans. Syst., Man, Cybern.—Part B: Cybern. 36 (1) (2006) 106–117.
[43] R.Y.M. Nakamura, L.A.M. Pereira, K.A. Costa, D. Rodrigues, J.P. Papa, X.S. Yang, BBA: a binary bat algorithm for feature selection, in: Proceedings of the 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI, pp. 291–297), 2012.
[44] P. Padungweang, C. Lursinsap, K. Sunat, A Discrimination analysis for unsupervised feature selection via optic diffraction principle, IEEE Trans. Neural Netw. and Learn. Syst. 23 (10) (2012).
[45] N. Parthalain, Q. Shen, R. Jensen, A Distance measure approach to exploring the rough set boundary region for attribute reduction, IEEE Trans. Knowl. Data Eng. 22 (3) (2010) 305–317.
[46] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[47] M.L. Raymer, W.F. Punch, E.D. Goodman, L.A. Kuhn, A.K. Jain, Dimensionality reduction using genetic algorithms, IEEE Trans. Evolut. Comput. 4 (2) (2000).
[48] E. Romero, J.M. Sopena, Performing feature selection with multilayer perceptrons, IEEE Trans. Neural Netw. 19 (3) (2008).
[49] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, Heuristic search over a ranking for feature selection, Comput. Intell. Bioinspired Syst. LNCS 3512 (2005) 742–749.
[50] S. Li, E.J. Harner, D.A. Adjeroh, Random KNN feature selection - a fast and stable alternative to random forests, BMC Bioinf. 14 (2013).
[51] P. Somol, B. Baesens, P. Pudil, J. Vanthienen, International journal of intelligent systems, Int. J. Intell. Syst. 20 (10) (2005).
[52] N. Spolaor, A.C. Lorena, A.D. Lee, Use of multiobjective genetic algorithms in feature selection, in: Proceedings of the11th Brazilian Symposium on Neural Networks (SBRN, pp. 146–151), 2010.
[53] R. Storn, K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Opt. 11 (4) (1997).
[54] S.C. Yusta, Different metaheuristic strategies to solve the feature selection problem, Pattern Recogn. Lett. 30 (5) (2009) 525–534.
[55] Z. Wang, M. Sun, J. Jiang, Automatically fast determining of feature number for ranking-based feature selection, Electron. Lett. 48 (23) (2012) 1462–1463.

[56] B. Waske, S. Schiefer, M. Braun, Random feature selection for decision tree classification of multi-temporal SAR data, in: Proceedings of the IEEE International Conference on Geoscience and Remote Sensing Symposium, (pp. 168–171), 2006.
[57] L. Xu, E. Hung, Distance-based feature selection on classification of uncertain objects, Adv. Artif. Intell. LNCS 7106 (2011) 172–181.
[58] Yan Xu, Gareth J.F. Jones, JinTao Li, Bin Wang, ChunMing Sun, A study on mutual information-based feature selection for text categorization, J. Comput. Inf. Syst. 3 (3) (2007) 1007–1012.
[59] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: a multi-objective approach, IEEE Trans. Cybern. 43 (6) (2012) 1656–1671.
[60] Bing Xue, Wenlong Fu, Mengjie Zhang, Multi-objective feature selection in classification: a differential evolution approach, Simul. Evol. Learn., LNCS 8886 (2014) 516–528.
[61] J.B. Yang, C.J. Ong, Feature selection using probabilistic prediction of support vector regression, IEEE Trans. Neural Netw. 22 (6) (2011).
[62] P. Yang, W. Liu, B.B. Zhou, S. Chawla, A.Y. Zomaya, Ensemble-based wrapper methods for feature selection and class imbalance learning, Adv. Knowl. Discov. Data Min. LNCS 7818 (2013) 544–555.
[63] S.H. Yang, B.G. Hu, Discriminative feature selection by nonparametric bayes error minimization, IEEE Trans. Knowl. Data Eng. 24 (8) (2012) 1422–1434.
[64] Q. Zhang, H. Li, MOEA/D: a multi-objective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[65] Q. Zhang, W. Liu, H. Li, The performance of a new MOEA/D on CEC09 MOP test instances, in: IEEE Congress Evolutionary Computation, (pp. 203–208), 2009.
[66] S. Zhao, E.C.C. Tsang, D. Chen, X. Wang, Building a rule-based classifier—A fuzzy-rough set approach, IEEE Trans. Knowl. Data Eng. 22 (5) (2010) 624–638.
[67] E. Zhou, A. Khotanzad, Fuzzy classifier design using genetic algorithms, Pattern Recogn. Elsevier 40 (2007) 3401–3414.
[68] P.F. Zhu, T.H. Meng, Y.L. Zhao, R.X. Ma, Q.H. Hu, Feature selection via minimizing nearest neighbor classification error, in: Proceedings of the 9th International Conference on Machine Learning and Cybernetics, (pp. 506–511), 2010.