Journal of Engg. Research **Vol. 4 No. (2) June 2016** pp. 65-85

طريقة اختيار المتغير الآلية للتجميع الآني وميزة الاختيار

فيجاي كومار، **جيتندر كومار شاهابرا ودينيش كومار**

*قسم علوم الحاسوب و الهندسة، جامعة مانيبال، جايبور، راجستان، الهند

**قسم هندسة الحاسوب في المعهد الوطني للتكنولوجيا، كوروكشترا، هاريانا، الهند

***قسم علوم وهندسة الكمبيوتر، جامعة جورو جامبشاور للعلوم والتكنولوجيا، هاريانا، الهند

الخلاصة

في هذه الورقة، يتم اقتراح نسخة محسنة من برنامج NMA للتجميع الآني مع ميزة الاختيار CFS. ويتم تحديد المتغيرات مثل حجم مجموعة الاستبدال، حجم المجموعة المختارة وحجم السكان تجريبيا ويتم الحصول عليها يدويا بعد محاولات عديدة. ويتم اقتراح نهجا آلي لتحديد هذه المعايير من NMA_CFS. تكشف النتائج التجريبية أن نظام NMA_CFS المعدل لا يؤثر على تدهور الأداء الآلي لنظام NMA_CFS الأصلي.

An automated parameter selection approach for simultaneous clustering and feature selection

Vijay Kumar*, Jitender K. Chhabra** and Dinesh Kumar***

**Department of Computer Science and Engineering, Thapar University, Patiala, Punjab, India*
E-mail: vijaykumarchahar@gmail.com

***Department of Computer Engineering, National Institute of Technology, Kurukshetra, Haryana, India*
E-mail: jitenderchhabra@gmail.com

****Department of Computer Science and Engineering, Guru Jambheshwar University of Science & Technology, Haryana, India*
E-mail: dinesh_chutani@yahoo.com

Corresponding Author: Email: vijaykumarchahar@gmail.com

ABSTRACT

In this paper, an improved version of Niching Memetic Algorithm for Simultaneous Clustering and Feature Selection (NMA_CFS) is proposed. In NMA_CFS, the parameters such as replacement group size, selection group size and population size are determined empirically and are manually obtained after hit and trial experimentation. An automated approach is proposed to determine these parameters of NMA_CFS. The experimental results reveal that this modified NMA_CFS does not deteriorate the performance of NMA_CFS due to automation, compared to the original NMA_CFS.

Keywords: Data clustering; feature selection; memetic algorithm; niching.

INTRODUCTION

Clustering is the process of partitioning a set of data points into a finite number of groups (clusters) in such a way that it maximises the between cluster variability and minimizes the within cluster variability. It has been used in many engineering fields including image segmentation, data forecasting, information retrieval and bioinformatics (Xu & Wunsch II, 2009). Due to this wide applicability, researchers do a lot of efforts to design new clustering algorithms as well as to improve the performance of existing algorithms using newly developed meta-heuristic approaches. Classical and meta-heuristic are the two broad categories of the existing clustering algorithms (Hatamlou *et al.,* 2011b). Classical clustering algorithms can be broadly divided into five categories: hierarchical clustering, partitional clustering, density-

based clustering, grid-based clustering, and model-based clustering (Jain *et al.,* 1999). K-Means is a widely used classical clustering algorithm due to its simplicity and efficiency (Forgy, 1965; Kaufman & Rousseeuw, 1990). However, K-Means has the shortcoming of depending on the initial state and converges towards local optima (Kao *et al*., 2008; Selim & Ismail, 1984). The main focus of this paper is on partitional clustering technique. The partitional clustering algorithms assume that all features are equally important for clustering. These algorithms do not discriminate among the important features in the given set of features. Some features may be redundant and some features may be irrelevant, which deceive the clustering process. The selection of important features is required for efficient clustering and the process is known as feature selection. Another problem of partitional clustering technique is to find the number of clusters.

In last few decades, many meta-heuristic algorithms have been used to overcome the above-mentioned shortcomings. Meta-heuristic algorithms are believed to be able to solve NP-hard problems with satisfactory near-optimal solutions with less computational time as compared to other classical methods. Although many meta-heuristic algorithms for solving clustering problems have been proposed, the results are unsatisfactory (Omran *et al.,* 2006). A niching memetic algorithm for simultaneous clustering and feature selection (NMA_CFS) has been proposed by Sheng *et al.* (2008). A composite chromosome is used to encode both feature selection and cluster centers with a varying number of clusters. The local search operations are introduced to refine the features and cluster centers. A niching method is used to preserve the population diversity. The main contribution of this paper is to propose a novel automatic parameter setting approach for NMA_CFS. Some formulas for parameter setting are proposed. These formulas can be computed in a unit time and do not increase the time complexity of the algorithm. The performance of Improved NMA_CFS (INMA_CFS) has been tested on variety of datasets and compared with several other clustering algorithms.

The rest of this paper is structured as follows. First, the brief overview of previous work done in the field of simultaneous clustering and feature selection techniques is described. The next section describes niching based memetic algorithm for simultaneous clustering and feature selection. The automatic parameter setting approach for NMA_CFS is given next. Thereafter, the real-life datasets, parameter setting and experimentation results are described. The complexity analysis is described followed by significance of parameters on the performance of proposed approach. Finally, the contribution of this paper is summarized.

RELATED WORKS

This section provides a summary of related works on simultaneous clustering and feature selection to the clustering problem. Vaithyanathan & Dom (1999) described a Bayesian approach for model selection to determine both number of clusters and

features. They used marginal likelihood and cross-validated likelihood for evaluation. Kim *et al.* (2000) used an evolutionary local selection algorithm to search over the features and number of clusters using K-Means and gaussian mixture clustering. Dy & Brodley (2004) examined the issues entailed in developing wrapper methods and used the maximum likelihood and scatter separability criterion for selecting number of features and clusters. Roth & Lange (2004) used automatic relevance determination prior to select features, when there are two clusters.

Law *et al.* (2004) presented an Expectation-Maximization algorithm to evaluate different features and clusters for gaussian-mixture clustering. Nanni (2006) developed a novel feature selection approach named as cluster-based pattern discrimination (CPD). Sheng *et al.* (2008) proposed an approach for simultaneous clustering and feature selection using a niching based memetic algorithm (NMA_CFS). They have made feature selection an integral part of global clustering search procedure and attempted to overcome the locally optimal solutions. They used a variable chromosome representation to encode both cluster centers and number of features. In addition, they also used local search operations to refine the chromosomes. A niching method was also integrated to avoid the premature convergence. Maugis *et al.* (2009) selected relevant features using backward stepwise selection for gaussian mixture models and an integrated likelihood criterion was used to search both number of clusters and features.

Sarvari *et al.* (2010) used the same concept as used in NMA_CFS except that harmony search algorithm was used instead of niching memetic algorithm. Breaban & Luchian (2011) introduced a new criterion to compute the number of clusters and provide ranking of partitions in feature subspaces of different cardinalities. This criterion is used to search both relevant features and optimal number of clusters. It minimizes the within-cluster variance and maximizes the between-cluster separation. Akarsu & Karahoca (2011) proposed a hybrid approach for clustering and feature selection using ant colony optimization (ACO). They have used ACO based clustering and then sequential backward selection technique for feature selection.

Javani *et al.* (2011) proposed a new approach for simultaneous clustering and feature selection using particle swarm optimization. They used weighting scheme for features to eliminate the irrelevant features. They proposed a new fitness function based on compactness and connectedness for finding the optimal number of features and clusters. Swetha & Devi (2012) used particle swarm optimization for feature selection and clustering. First, they carried out feature selection to select relevant features. Thereafter, clustering was performed on the selected features. Du & Shen (2013) proposed a unified framework based on fisher score and spectral clustering. They maximized fisher criterion for feature selection and minimized the spectral clustering criterion to preserve the manifold structure.

In this paper, a novel approach of automatic parameter selection for NMA_CFS has been proposed. The main difference between well-known feature selection approach CPD and proposed approach is that the former is used for feature selection only and this technique is used for classification in situations where training and testing has been done. It is more appropriate for classification rather than clustering. Moreover it is applicable to the datasets that consist two clusters. The proposed approach computes the number of clusters and features simultaneously during the run time and can be applied for datasets having any number of clusters.

NICHING MEMETIC ALGORITHM FOR CLUSTERING AND FEATURE SELECTION

Sheng *et al.* (2008) proposed a niching based memetic algorithm for simultaneous clustering and feature selection (NMA_CFS) via the clustering criterion optimization. The NMA_CFS used the variable length composite chromosomes to represents the solutions. The composite chromosome encodes both feature selection and cluster centers with a variable number of clusters. The main operations of NMA_CFS are reproduction, genetic operators, feature selection, and niching competition replacement. The description of the algorithm is given as follows (Sheng *et al.,* 2008):

NMA_CFS Algorithm

Step 1. Initialize the algorithm parameters such as population size, replacement size, selection size, and maximum number of clusters (K_{max}).

Step 2. Initialize the set of n chromosomes, which encode both feature selection and cluster centers with varying number of clusters.

Step 3. Compute the fitness value of each chromosome in the initial population.

Step 4. Repeat the following steps until the maximum number of iterations is reached

a) Compute the fitness value of each chromosome.

b) Select the pairing parents based on niching method.

c) Produce intermediate offspring by applying genetic operators on different parts of the paired parents.

d) Addition and removal process of features from the intermediate offspring.

e) Apply K-Means algorithm on the intermediate offspring.

f) Pair the offspring with the most similar solution found during a restricted competition replacement.

g) Compute the fitness value for each of the offspring. If the fitness of the offspring is better than its paired solution, then the latter is replaced.

Step 5. The best chromosome provides the optimal subset of features and cluster centers.

The steps of NMA_CFS are described in the following subsections.

Chromosome representation

NMA_CFS uses a variable length composite chromosome to encode both features and cluster centers for a varying number of clusters. For n data points, each have D dimensions, for user specified maximum number of cluster $K_{\max}$, a chromosome is a vector of $D_g + D \times k$. The first D_g entries represent an individual feature having values 0 or 1. The value 0 indicates the corresponding feature is ignored, otherwise it is selected. These are control bits for feature selection. The remaining bits are used for k cluster centers, each having D dimensions. k is the number of clusters, which is computed according to $RandInt(2, K_{\max})$. Here, $RandInt(\ )$ is a random number generator function that return a natural number in the range of 2 to $K_{\max}$. The vector $(V_i(t))$ of an agent i is demonstrated as

$V_i(t) =$

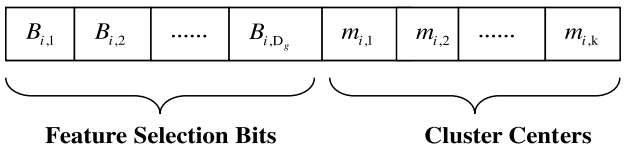

Fig. 1. Chromosome encoding scheme.

where $m_{i,j}$ is the j^{th} cluster center of i^{th} agent and Th_{i,D_g} is binary value of the corresponding feature.

For example, in four-dimensional dataset, the chromosome encodes three clusters as shown in Figure 2. The first, fourth and fifth features are being selected according to the selection bits. The cluster centers become (8.6,5.4,0.4), (3.7,1.9,0.9), (5.6,4.2,0.9).

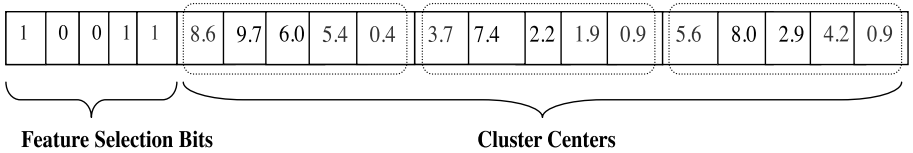

Fig. 2. A Chromosome encoding scheme consists of 2-dimensional dataset.

Niching process

The niching method is incorporated in the genetic algorithm to preserve the population diversity. During the niching selection, one parent (say p_1) is selected randomly from the population. Its mate (say p_2) is selected from a group of solutions called the selection group with the most similar number of clusters as p_1. The selection group is picked randomly from the population. During the restricted competition replacement, each offspring is compared with a group of solutions called the replacement group. The replacement group is picked randomly from the population and is paired with the most similar one. If the fitness value of the offspring is better than its paired solution, then the latter one is replaced.

Feature addition and removal process

The two classical feature selection techniques, sequential forward selection (SFS) and sequential backward selection (SBS) (Foroutan & Sklasky, 1987), are incorporated in the Niching based genetic algorithm for addition and removal of features. These are specified as follows:

a) Feature addition: Select a feature from the unselected feature subset, that when combined with the currently selected features produces the largest value of fitness function and changes its status to "selected".

b) Feature removal: Select a feature from the selected feature subset that when combined with the currently selected features produces the small value of fitness function and changes its status to "ignored".

Genetic operators

The different genetic operators (crossover and mutation) are applied on feature selection and cluster center part. For the feature selection part, the *m*-point crossover and flip mutation are applied. The *m*-point crossover chooses the *m* points at random and alternately copies each segment from the two parents. For the cluster center part, the two-point crossover and Gaussian mutation are applied.

Fitness function

A large number of clustering criteria have been reported in literature. The most popular clustering criterion is $trace\left(S_w^{-1}S_b\right)$. S_w indicates how much scattered the data points are from their cluster center. S_b indicates how much scattered the cluster centers are from the mean of the whole dataset. However, $trace\left(S_w^{-1}S_b\right)$ is biased toward higher dimensionality. The value of this clustering criterion monotonically increases, as the number of features increases. To get rid off this problem, the penalty function suggested by Sheng *et al.* (2008) is incorporated in this clustering criterion and is

defined as follows.

$$Fit_1 = trace\left(S_w^{-1} S_b\right) \times \frac{\left(D-d\right)}{\left(D-1\right)} \tag{1}$$

where D is dimension of the given dataset and d is the number of features selected from the given dataset.

The Fit_1 is biased towards increasing the number of clusters. To overcome this problem, another penalty function is incorporated in the Fit_1 and the fitness function is rewritten as:

$$Fitness\ Function = Fit_1 \times \frac{\left(K_{\max} - k\right)}{\left(k-1\right)} \tag{2}$$

where $K_{\max}$ is the maximum number of clusters specified by user and k is the number of clusters, which is computed according to $RandInt\left(2, K_{\max}\right)$. Here, $RandInt\left(\ \right)$ is a random number generator function that returns a natural number in the range of 2 to $K_{\max}$.

PROPOSED APPROACH

This section first describes the motivation and mathematical foundation of proposed approach followed by proposed automatic clustering and feature selection technique.

MOTIVATION

The major contribution of this paper is a novel approach for automatic parameter selection scheme for NMA_CFS. It (NMA_CFS) needs tuning for getting the optimal value of the objective function, which itself is a difficult task, especially when the dataset consists of large number of data points and features. A lot of hit and trial experimentation needs to be done for proper tuning of NMA_CFS. To get rid off this problem of parameter setting, some formulas for parameter setting are proposed. These formulas can be computed in a unit time and do not increase the time complexity of the algorithm. Discussed below is first, the shortcoming of NMA_CFS and then the proposed formulas with justification are presented.

1. **The upper limit on the number of clusters**: Sheng *et al.* (2008) set the upper limit on the number of clusters ($K_{\max}$) to $\sqrt{No.\ of\ datapoints}$. The value of $K_{\max}$ greatly affects the performance of NMA_CFS as the fitness function is directly proportional to $K_{\max}$. If $K_{\max}$ is much larger than the actual number of clusters then the algorithm generates higher number of clusters than the actual count and takes more computational time as well. Otherwise, it generates small number of clusters.

As a matter of fact, the number of clusters not only depends upon the number of data points, but also on the number of features. The number of clusters is based on the combination of given features as well as number of data points and hence multiplying the value of the features and data points may become more useful. Based on these facts, we proposed an equation for $K_{\max}$ which is given below

$$K_{\max} = \sqrt[3]{(No.\ of\ \ datapoints \times No.\ of\ \ features)} \tag{3}$$

Here, we have used cube root instead of square root. There are mainly two reasons behind this. First, it produces the value of $K_{\max}$ that will generate the near optimal number of clusters (See section significance of the parameters on the performance of the INMA_CFS). Second, it reduces the computational time.

2. **Size of replacement group**: Sheng *et al.* (2008) mentioned that the size of replacement group is set experimentally. Improper size of the replacement group generates undesirable/wrong results. It is a tedious task to set the size of replacement group, as it is to be determined using hit and trail method.

To solve this problem, we propose an equation to determine automatically the size of the replacement group. The size of replacement group depends upon the number of data points, features and number of clusters. The number of clusters depends upon the features present in the data points. Based on these facts, the size of replacement group is defined as:

$$Replacement\ \ Size = \frac{No.\ of\ datapoints}{(K_{max} + No.\ of\ \ features)} \tag{4}$$

3. **Size of selection group**: The size of the selection group was also set empirically (determined experimentally) in Sheng *et al.* (2008). It was also determined by hit and trail method.

It is proposed that this size can be computed automatically by using the equation proposed below. We have analysed that the size of selection group should not be greater than the size of replacement group. When the size of selection group is larger, then the possibility of selecting parent pairs having same number of clusters increases. Hence, the size of selection group should be small to ensure the selection of solutions having different number of clusters. After a thorough analysis, we have found that Sheng *et al.* (2008) varied the value of selection group from 20% to 70% of the size of replacement group. We have analyzed that the size of selection group set to 30% of the size of replacement group gives optimal number of clusters. Hence, the proposed equation for size of selection group is as follows:

$$Selection\ Size = 0.30 \times Replacement\ Size \tag{5}$$

4. **Population size:** Sheng *et al.* (2008) used different values of population size for different datasets. But they have not mentioned any reason behind this. It is also a tedious task to determine an appropriate size of the population, especially there is no guideline available to determine the population size for new datasets, not used by Sheng *et al.* (2008).

   After a thorough analysis, we have found that Sheng *et al.* (2008) have taken large population size for large number of data points. However, they have also mentioned that large value of population sizes may lead to a longer runtime, but no improvement in the performance of algorithm. We propose a general formula for setting the size of the population. As the replacement group is always a subset of population, the population size should not be smaller or equal to the replacement group. In order to maintain the balance between exploration and exploitation, population size should be at least 30-40% higher than replacement size. At the same time, taking this too higher will result in higher computation time. Hence we have proposed this size,

$$Population\ Size = 1.50 \times Replacement\ Size \tag{6}$$

Automatic clustering and feature selection using improved NMA_CFS

The main strength of improved NMA_CFS is that it sets the desired parameters using the above mentioned proposed equations for automatic clustering and feature selection and the time consuming hit and trial method is eliminated. The steps of the proposed approach of automatic parameter selection for NMA_CFS (INMA_CFS) are illustrated with a flow-chart in Figure 3. In INMA_CFS, K-Means (KM) clustering algorithm is used. Expectation-maximization (EM) algorithm is also another good option. K-Means can be treated as a special case of EM under a spherical Gaussian mixture, where the dimensions have the same variance. It has been observed that KM, as compared to EM, does not yield better results particularly while handling overlapping data points. To solve this problem, the membership function can be incorporated in the KM.

COMPLEXITY ANALYSIS

In this section, the complexity analysis of INMA_CFS is presented. The time complexity of INMA_CFS basically depends upon the three major processes, such as feature addition and removal process, one step of K-Means algorithm, and fitness computation process. The feature addition and removal process requires $O\left(nK_{\max}D^2\right)$ time. The K-Means algorithm requires $O\left(nK_{\max}D\right)$ time. The fitness computation process requires $O\left(nD\right)$ time. The proposed equations for parameter setting require

$O(1)$ time. Hence, the overall complexity of INMA_CFS is $O\left(nK_{\max}D^2PG\right)$. where P is the population size and G is the number of generations.

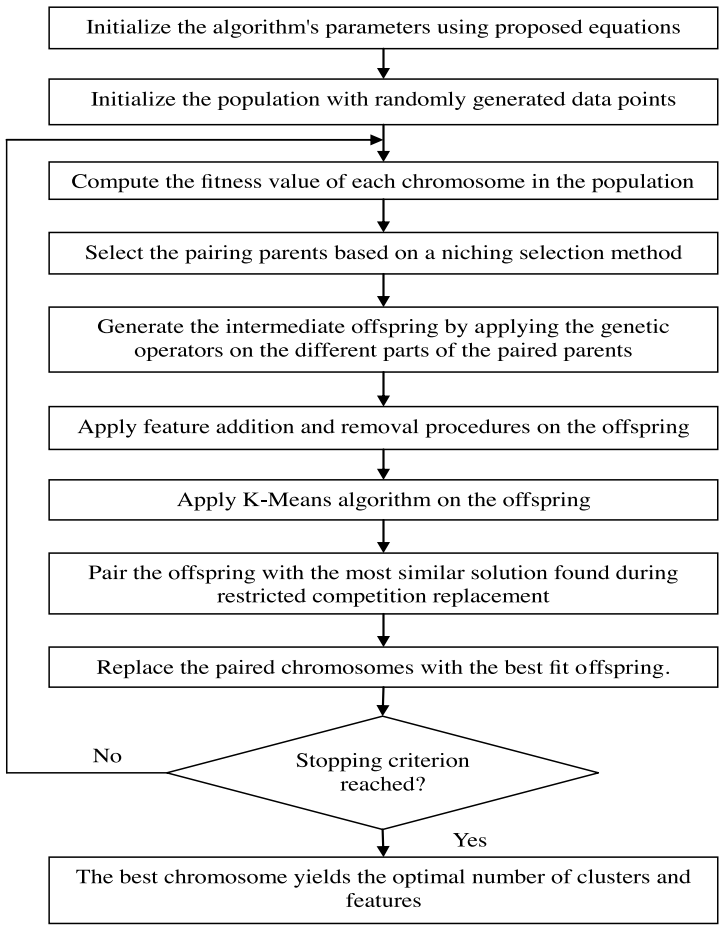

Fig. 3. Flowchart of the proposed INMA_CFS.

EXPERIMENTAL RESULTS

This section describes the experimentation to evaluate the performance of improved NMA_CFS technique on twelve real-life datasets. These datasets are described in preceding subsection. The results are evaluated and compared with well known clustering techniques.

Datasets used

Twelve real-life datasets with a variety of complexity are used to evaluate the performance of the improved NMA_CFS clustering technique. The real life datasets are *Iris*, *Wine, Glass, Haberman, Bupa, Libras, Musk (V.1), WDBC, Hill Valley,*

Cancer, Vowel, Contraceptive Method Choice (CMC), and *Image Segmentation*, which are available in the UCI machine learning repository (Blake & Merz, 1998). Table 1 summaries the main characteristics of the used datasets.

Table 1. Main characteristics of datasets used

Dataset Name	Number of Instances	Number of Features	Number of Classes	Type
Iris	150	4	3	Real
Wine	178	13	3	Real
Glass	214	9	6	Real
Haberman	306	3	2	Real
Bupa	345	6	2	Real
Libras	360	90	15	Real
Musk (V.1)	476	168	2	Real
WDBC	576	30	2	Real
Hill Valley	606	101	2	Real
Cancer	683	9	2	Real
Vowel	871	3	6	Real
CMC	1473	9	3	Real
Image Seg.	2310	19	7	Real

Algorithms used for comparisons

The performance of the improved NMA_CFS is compared against four well known algorithms reported in literature, including K-Means (KM), modified harmony search-based clustering (MHSC) (Kumar *et al.,* 2014), Feature selection wrapped around the K-Means algorithm (FS_K-Means) (Sheng *et al.,* 2008), and Niching memetic algorithm for clustering and feature selection (NMA_CFS) (Sheng *et al.,* 2008). The performance of the algorithms is evaluated and compared using three cluster quality measures as number of clusters, number of features and classification accuracy.

Parameter setting for the algorithms

The parameters setting for K-Means (Jain & Dubes, 1988) and MHSC (Kumar *et al.,* 2014) are set the same as are in their original paper. All the parameters values of NMA_CFS were determined experimentally. For NMA_CFS and INMA_CFS, flip and Gaussian mutation rates are set to 0.01. For both NMA_CFS and INMA_CFS, the number of iterations is set to 50. Based on experimentation, the optimal parameter setting for NMA_CFS is reported in Table 2.

Table 2. Parameters setting for NMA_CFS

Datasets Used	Replacement Size	Selection Size	Population Size
Iris	10	4	40
Wine	11	5	40
Glass	22	10	35
Haberman	4	2	40
Bupa	12	6	50
Libras	25	12	50
Musk (V.1)	21	12	40
WDBC	20	8	70
Hill Valley	24	13	50
Cancer	9	3	45
Vowel	20	15	60
CMC	10	4	70
Image Seg.	40	16	120

Results and discussions

Table 3 show the performance of above mentioned algorithms over 20 independent runs on 13 real-life datasets. Experimental results reveal that the proposed method is able to generate equally good results as produced by Sheng's method. The latter used hit and trail for parameter setting, whereas the former (proposed) method is based upon the automated calculations of parameters and takes very less time as compared to that of hit and trial method. The results further strengthen our belief that automation does not deteriorate the performance of the proposed algorithm.

Table 3. Number of clusters, number of features selected and classification accuracy obtained from clustering algorithms.

Dataset	Method	Evaluation Method		
		Number of Clusters	Number of Features	Classification Accuracy (%)
Iris	K-Means	Fixed 3	Fixed 4	84.4 (±6.3)
	MHSC	Fixed 3	Fixed 4	86.7 (±5.7)
	FS-K Means	4.0 (±0.7)	2.5 (±0.6)	90.6 (±3.9)
	NMA_CFS	3.0 (±0.0)	1.9 (±0.2)	95.9 (±1.5)
	INMA_CFS	3.0 (±0.0)	1.7 (±0.7)	95.0 (±2.0)
Wine	K-Means	Fixed 3	Fixed 13	65.9 (±5.9)
	MHSC	Fixed 3	Fixed 13	64.1 (±5.4)
	FS-K Means	3.9 (±1.8)	7.2 (±1.7)	64.6 (±6.1)
	NMA_CFS	3.5 (±1.1)	5.8 (±1.5)	65.6 (±8.9)
	INMA_CFS	2.9 (±0.3)	6.6 (±1.7)	66.2 (±9.5)

Glass	K-Means	Fixed 6	Fixed 9	50.8 (±3.6)
	MHSC	Fixed 6	Fixed 9	49.4 (±3.6)
	FS-K Means	6.9 (± 0.7)	4.3 (±1.5)	45.2 (±7.3)
	NMA_CFS	5.9 (± 1.5)	3.5 (±1.1)	43.1 (±5.4)
	INMA_CFS	5.6 (± 0.9)	3.6 (±1.2)	42.8 (±6.0)
Haberman	K-Means	Fixed 2	Fixed 3	50.9 (±1.1)
	MHSC	Fixed 2	Fixed 3	55.4 (±4.0)
	FS-K Means	2.7 (±0.9)	1.9 (±0.7)	54.6 (±6.3)
	NMA_CFS	2.5 (±0.5)	1.3 (±0.5)	55.8 (±8.1)
	INMA_CFS	2.0 (±0.0)	1.0 (±0.0)	54.3 (±4.9)
Bupa	K-Means	Fixed 2	Fixed 6	53.1 (±0.0)
	MHSC	Fixed 2	Fixed 6	53.8 (±1.9)
	FS-K Means	3.1 (±1.1)	2.9 (±0.9)	51.0 (±3.0)
	NMA_CFS	3.0 (±0.7)	2.4 (±0.8)	51.5 (±3.9)
	INMA_CFS	2.2 (±0.5)	2.2 (±0.8)	52.1 (±4.7)
Libras	K-Means	Fixed 15	Fixed 90	22.5 (±3.6)
	MHSC	Fixed 15	Fixed 90	22.6 (±3.1)
	FS-K Means	11.5 (±1.0)	51.8 (±1.6)	29.5 (±2.4)
	NMA_CFS	10.2 (±1.9)	49.1 (±0.8)	30.3 (±4.3)
	INMA_CFS	10.9 (±2.2)	51.0 (±1.1)	31.6 (±3.9)
Musk (V.1)	K-Means	Fixed 2	Fixed 168	51.2 (±0.0)
	MHSC	Fixed 2	Fixed 168	51.6 (±1.5)
	FS-K Means	2.0 (±0.0)	85.9 (±9.2)	51.9 (±1.8)
	NMA_CFS	2.0 (±0.0)	83.4 (±7.5)	53.3 (±2.2)
	INMA_CFS	2.0 (±0.0)	84.8 (±7.5)	52.9 (±2.1)
WDBC	K-Means	Fixed 2	Fixed 30	85.4 (±0.0)
	MHSC	Fixed 2	Fixed 30	85.7 (±1.9)
	FS-K Means	2.0 (±0.0)	15.2 (±2.1)	86.3 (±2.1)
	NMA_CFS	2.0 (±0.0)	14.8 (±0.9)	90.8 (±0.4)
	INMA_CFS	2.0 (±0.0)	13.4 (±2.6)	90.4 (±0.8)
Hill Valley	K-Means	Fixed 2	Fixed 101	45.5 (±0.0)
	MHSC	Fixed 2	Fixed 101	50.9 (±0.3)
	FS-K Means	2.5 (±0.9)	52.3 (±6.4)	48.1 (±2.3)
	NMA_CFS	2.3 (±0.5)	50.1 (±5.0)	49.2 (±1.7)
	INMA_CFS	2.2 (±0.5)	48.6 (±4.2)	49.4 (±1.5)
Cancer	K-Means	Fixed 2	Fixed 9	94.0 (±0.0)
	MHSC	Fixed 2	Fixed 9	94.4 (±0.9)
	FS-K Means	2.5 (±1.2)	6.2 (±2.3)	93.0 (±2.0)
	NMA_CFS	2.2 (±0.4)	4.1 (±1.4)	94.6 (±2.9)
	INMA_CFS	2.1 (±0.3)	3.4 (±1.6)	94.3 (±1.0)
Vowel	K-Means	Fixed 6	Fixed 3	53.0 (±5.0)
	MHSC	Fixed 6	Fixed 3	53.6 (±5.8)
	FS-K Means	6.7 (±2.8)	2.6 (±1.0)	52.9 (±4.9)
	NMA_CFS	6.0 (±1.4)	1.1 (±0.3)	53.4 (±3.6)
	INMA_CFS	6.3 (±0.8)	1.1 (±0.3)	53.8 (±4.1)

CMC	K-Means	Fixed 3	Fixed 9	39.9 (±0.2)
	MHSC	Fixed 3	Fixed 9	40.1 (±1.4)
	FS-K Means	3.9 (±1.6)	4.8 (±2.1)	39.6 (±4.3)
	NMA_CFS	2.9 (±0.8)	3.6 (±1.4)	40.4 (±3.1)
	INMA_CFS	3.6 (±0.8)	2.4 (±1.2)	40.2 (±2.5)
Image Seg.	K-Means	Fixed 7	Fixed 19	61.4 (±3.8)
	MHSC	Fixed 7	Fixed 19	60.7 (±2.7)
	FS-K Means	8.5 (±1.8)	3.7 (±0.6)	63.1 (±2.8)
	NMA_CFS	6.6 (±1.4)	2.4 (±0.5)	64.8 (±1.8)
	INMA_CFS	6.6 (±0.6)	7.5 (±1.8)	69.7 (±1.5)

Statistical evaluation

Here, we have done statistical test to show the performance of improved NMA_CFS and existing NMA_CFS algorithms is same. The unpaired *t*-tests have been done to determine whether the proposed INMA_CFS approach is statistically different or not. We have taken 20 as the sample size for unpaired *t*-tests. Table 4 shows the results of unpaired t-tests based on the accuracy presented in Table 3. As can be seen from Table 4, INMA_CFS is statistically equivalent to NMA_CFS for all the datasets except *Image Seg.* dataset. For *Image Seg.* dataset, INMA_CFS performs better than NMA_CFS.

Table 4. Unpaired t-test between INMA_CFS and NMA_CFS algorithms for each dataset based on the data presented in Table 3.

Dataset	***Standard Error***	***t***	***95% Confidence Interval***	***Two-tailed P***	***Difference***
Iris	0.559	1.610	-0.232 to 2.032	0.1157	Not Statistical Significant
Wine	2.911	0.206	-6.493 to 5.293	0.8378	Not Statistical Significant
Glass	1.805	0.166	-3.354 to 3.954	0.8689	Not Statistical Significant
Haberman	2.117	0.708	-2.785 to 5.785	0.4829	Not Statistical Significant
Bupa	1.366	0.439	-3.365 to 2.165	0.6629	Not Statistical Significant
Libras	1.298	1.002	-3.928 to 1.328	0.3229	Not Statistical Significant
Musk (V.1)	0.680	0.588	-0.977 to 1.777	0.5599	Not Statistical Significant
WDBC	0.200	2.000	-0.005 to 0.805	0.0527	Not Statistical Significant
Hill Valley	0.507	0.394	-1.226 to 0.826	0.6954	Not Statistical Significant
Cancer	0.686	0.437	-1.089 to 1.689	0.6643	Not Statistical Significant
Vowel	1.220	0.328	-2.870 to 2.070	0.7448	Not Statistical Significant
CMC	0.891	0.225	-1.603 to 2.003	0.8235	Not Statistical Significant
Image Seg.	0.524	9.352	-5.961 to -3.839	<0.0001	Statistical Significant

SIGNIFICANCE OF PARAMETERS ON THE PERFORMANCE OF INMA_CFS

In this section, we discuss the significance of the parameters such as maximum number of user specified clusters, size of selection group, and population size. The impact of above-mentioned parameters is analyzed on the performance of INMA_CFS.

Significance of K_{max} on the performance of INMA_CFS

Other parameters are kept fixed, which are computed from proposed equations, INMA_CFS was run for different values of K_{max} (2, 4, 6, 8, 10, 12, 14, 16, 18, and 20). Figure 4 shows the effect of K_{max} over the number of clusters. From Figure 4, we can see that the optimal values of K_{max} for *Iris, Wine, Glass, WDBC*, and *Vowel* are 8, 13, 12, 25, and 14 respectively. These values of K_{max} generate the optimal number of clusters and obtained from our proposed equation. The value of K_{max} for *Iris* dataset is smaller than other datasets as the combination of data points and features are smaller. Whereas, the value of K_{max} for *WDBC* dataset is higher than other datasets as the combination of data points and features are higher. Figure 5 shows the effect of K_{max} over the accuracy obtained from the proposed method. The results depict that the optimal values of K_{max} for *Iris, Wine, Glass, WDBC*, and *Vowel* are 8, 13, 12, 25, and 14 respectively. Thus we can say that K_{max} depends on both the number of data points and features. Hence it has been analytically proved that the proposed equation for K_{max} provides appropriate value for finding optimal number of clusters in the specified dataset.

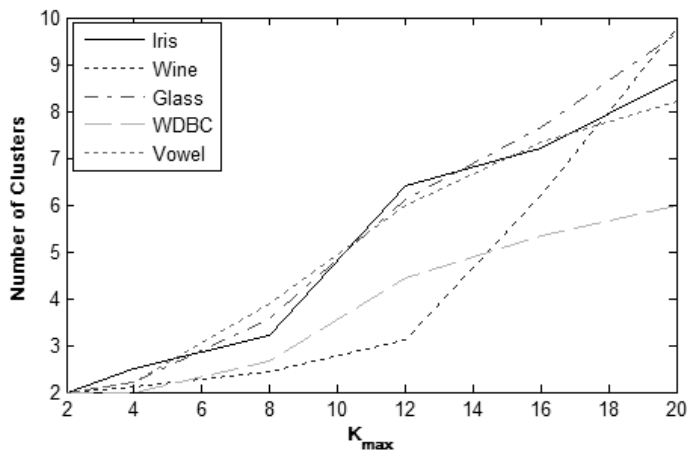

Fig. 4. Effect of K_{max} on the number of clusters obtained from proposed method.

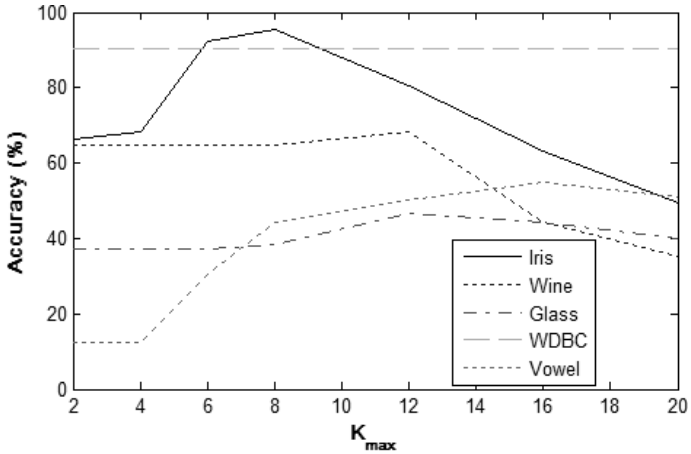

Fig. 5. Effect of $K_{\max}$ on the accuracy obtained from proposed method.

Significance of selection group size on the performance of INMA_CFS

The INMA_CFS was run for different values of selection group size keeping other parameters fixed. The values of selection group size used in experimentation are 20% to 70% of the size of replacement group. Figure 6 shows the effect of selection group size over the number of clusters. From Figure 6, we can see that the size of selection group set to 30% of the size of replacement group gives optimal number of clusters. Figure 7 shows the effect of selection group size over the accuracy obtained from INMA_CFS. From results it is revealed that the selection group size set to 30% of replacement group produces best accuracy. Hence it has been analytically proved that the proposed equation for selection group size provides appropriate value for finding optimal number of clusters in the specified dataset.

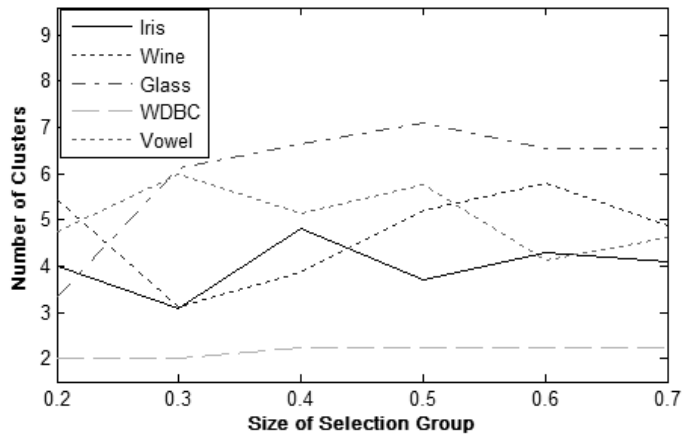

Fig. 6. Effect of selection group size on the number of clusters obtained from proposed method.

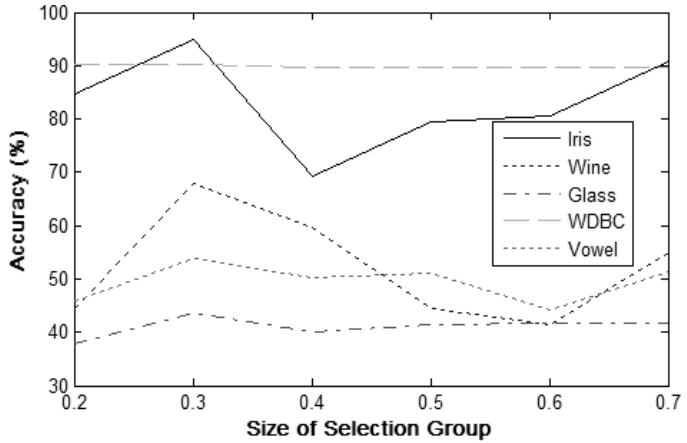

Fig. 7. Effect of selection group size on the accuracy obtained from proposed method.

Significance of population size on the performance of INMA_CFS

The INMA_CFS was run for different values of population size keeping other parameters fixed. The values of population size used in experimentation are 100% to 250% of the size of replacement group. Figures 8 and 9 show the effect of population size over the number of clusters and execution time. From Figure 8, we can see that the population size set to 150% of the size of replacement group gives optimal number of clusters. The results obtained from Fig. 9 show that the large value of population sizes lead to a longer runtime but no improvement in the performance of algorithm. Hence it has been analytically proved that the proposed equation for population size provides appropriate value for finding optimal number of clusters in the specified dataset.

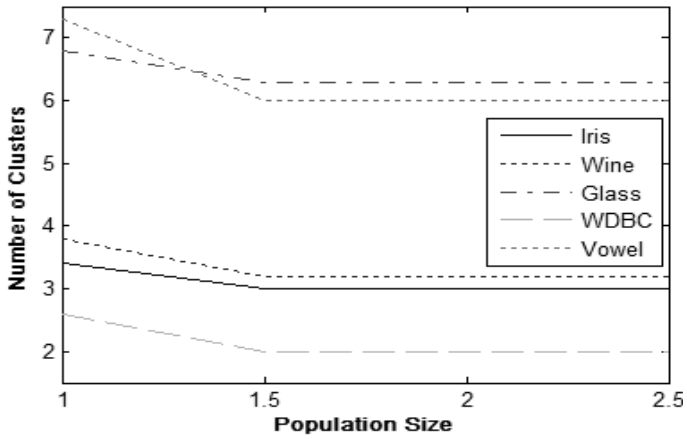

Fig. 8. Effect of population size on the number of clusters obtained from proposed method.

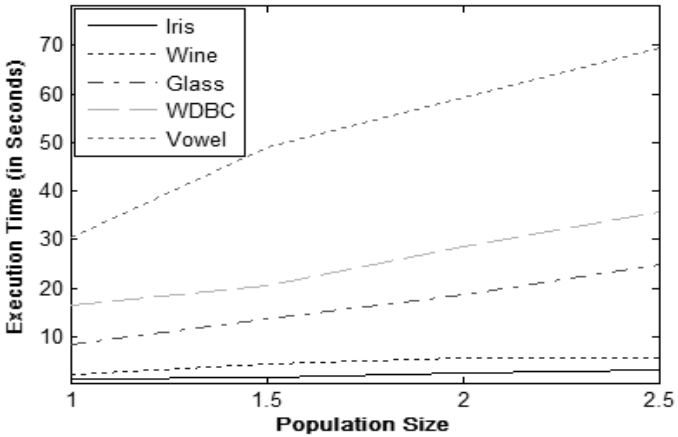

Fig. 9. Effect of population size on the execution time of the proposed method.

CONCLUSIONS

An automatic approach of parameter setting for niching memetic algorithm for simultaneous clustering and feature selection is proposed. The four novel formulae for parameter settings are proposed in the automatic approach. These formulae are computed in a unit time and did not increase the time complexity of the algorithm. The performance of proposed approach has been tested on thirteen real-life datasets and compared with several other clustering algorithms. The experimental results show that the proposed approach is able to detect the correct number of clusters and features. The automation saves lot of time, which would otherwise have been wasted in parameter settings using hit and trial method. The proposed approach does not deteriorate the performance of existing algorithm. This has been proved using statistical tests also. Further, the effect of these parameters, such as user specified number of clusters, selection group size, and population size, has also been analyzed. The results reveal that the proposed formulae are efficient in determining the optimal number of clusters and features as well.

REFERENCES

Akarsu, E. & Karahoca, A. 2011. Simultaneous feature selection and ant colony clustering. Procedia Computer Science,, **3**:1432-1438.

Breaban, M. & Luchian, H. 2011. A unifying criterion for unsupervised clustering and feature selection. Pattern Recognition, **44**:854-865.

Blake, C.L. & Merz, C.J. 1998. UCI Repository of Machine Learning. ⟨http:/www.ics.uci.edu/_mlearn/databases/⟩.

Du, L. & Shen, Y.D. 2013. Joint clustering and feature selection. In: Wang, J., Xiong, H., Ishikawa, Y., Xu, J. & Zhou, J.(Eds.). Web-Age Information Management. Pp. 241-252. Springer-Berlag, Berlin.

Dy, J.G. & Brodley, C.E. 2004. Feature selection for unsupervised learning. Journal of Machine Learning

Research, 5:845-889.

Forgy, E. 1965. Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. Biometrics, **21**:768-769

Foroutan, I. & Sklasky, J. 1987. Feature selection for automatic classification of non-gaussian data. A robust competitive clustering algorithm with applications in computer vision. IEEE Transactions on Systems, Man and Cybernetics, **17**:187-198.

Hatamlou, A., Abdullah, S. & Nezamabadi-pour, H. 2011b. Application of gravitational search algorithm on data clustering. In: Yao, J., Ramanna, S., Wang, G. & Suraj, Z. (Eds.), *Rough Set and Knowledge Technology*, Lecture Notes in Computer Science, Berlin, Germany: Springer-Verlag, Vol. 6954, pp. 337-346.

Jain, A.K. & Dubes, R.C. 1988. Algorithms for clustering data. Prentice-Hall, NJ, USA.

Jain, A.K., Murty, M.N. & Flynn, P.J. 1999. Data clustering: a review. ACM Computing Survey, **31**(3):264–323.

Javani, M., Faez, K. & Aghlmandi, D. 2011. Clustering and feature selection via PSO algorithm. Proceedings of the Artificial Intelligence and Signal Processing, Pp. 71-76, Tehran.

Kao, Y.T., Zahara, E. & Kao, I.W. 2008. A hybridized approach to data clustering. Expert Systems with Applications, **34**(3):1754–1762.

Kaufman, L. & Rousseeuw, P. 1990. Finding groups in data: An introduction to cluster analysis. John Wiley & Sons Inc.

Kim Y., Street, W. & Menczer, F. 2000. Feature selection in unsupervised learning via evolutionary search. Proceedings of Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Pp. 365-369.

Kumar, V., Chhabra, J.K. & Kumar, D. 2014. Clustering using modified harmony search algorithm. International Journal of Computational Intelligence Studies, **3**(2/3): 113-133.

Law, M.H.C., Figueiredo, M.A.T. & Jain, A.K. 2004. Simultaneous feature selection and clustering using mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence. **26**(9): 1154-1165.

Maugis, C., Celeux, G. & Martin-Magniette, M.L. 2009. Variable selection for clustering with gaussian mixture models. Biometrics, **65**: 701-709.

Nanni, L. 2006. Cluster-based pattern discrimination: a novel technique for feature selection. Pattern Recognition Letters, 27: 682-687.

Omran, M.G.H., Salman, A. & Engelbrecht, A.P. 2006. Dynamic clustering using particle swarm optimization with application in image segmentation. Pattern Analysis and Applications. **8**(4): 332-344.

Roth, R.V. & Lange, T. 2004. Feature Selection in Clustering Problems. Proceedings of Advances in Neural Information Processing Systems, Cambridge.

Sarvari, H., Khairdoost, N. & Fetanat, A. 2010. Harmony search algorithm for simultaneous clustering and feature selection. Proceedings of the International Conference of Soft Computing and Pattern Recognition. 202-207, Paris.

Selim, S.Z., & Ismail, M.A. 1984. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. IEEE Transactions on Pattern Analysis and Machine Intelligence, **1**:81-87.

Sheng, W., Liu, X. & Fairhurst, M. 2008. A niching memetic algorithm for simultaneous clustering and feature selection. IEEE Transactions on Knowledge and Data Engineering, **20**(7): 868-879.

Swetha, K.P. & Devi, V.S. 2012. Simultaneous feature selection and clustering using particle swarm optimization. Proceedings of the International Conference on Neural Information Processing. Pp. 509-515, Doha, Qatar.

Vaithyanathan, S. & Dom, B. 1999. Generalized model selection for unsupervised learning in high dimensions. Proceeding of the Neural Information Processing Systems, Pp. 970-976, Cambridge.

Xu, R. & Wunsch II, D.C. 2009. Clustering, John Wiley and Sons, USA.

Submitted: 10/4/2015
Revised: 29/6/2015
Accepted: 29/9/2015