Evolutionary Fuzzy Feature Selection for Unsupervised Clustering

Ohm Sornil and Jitwadee Chaiyakarn

Department of Computer Science, National Institute of Development Administration
Bangkok 10240, Thailand

Abstract

In this article, we propose a novel filter-based technique to select a minimal set of features for unsupervised clustering problems. The proposed method employs fuzzification of original features using filter-based evolutionary irregular-shaped membership functions, and a feature selection process which employs two criterion functions to evaluate feature subsets. The first function is applied to select highly relevant but less redundant features, and the second function is applied to select a feature subset that defines the most well-formed data clusters. K-means is employed as the clustering algorithm. The proposed technique is evaluated using standard machine learning data sets in various sizes and complexities. The results show that the proposed technique is effective and performs well in comparisons with other research.

Keywords: Automatic Feature Selection, Unsupervised Clustering, Fuzzy Logic, Genetic Algorithm

1. Introduction

A feature selection method selects a small subset of highly predictive features from the original problem. It most of the time yields better results, due to the reduction of noises and distractions, and takes less time for clustering than does using the entire set of features. Feature selection has been widely studied in supervised classification. However, it is a rather recent and challenging research to select a set of predictive features for cluster analysis for two reasons. First, it is not an easy task to define a good criterion for evaluating the quality of a candidate feature set due to the absence of accurate labels of items. Second, feature selection needs to evaluate an exponential number of feature combinations which is impractical if a data set has a large number of features [10]. Feature selection approaches can be classified into three categories which are wrapper, filter, and hybrid approaches.

Given an unsupervised clustering problem, a wrapper method incorporates the clustering itself into the feature evaluation process. To evaluate the importance of a candidate feature subset, data clusters are built, validated, and used to evaluate the set. This approach is believed to generate a subset that yields high clustering validity; however, it is likely to take a long time. Hong et al. [10] use population based incremental learning algorithm (PBIL) to generate candidate feature subsets. There are two major steps of the proposed framework. In the first step, the population of clustering solutions is obtained through executing different clustering algorithms and then all the obtained clustering solutions are combined into a single consensus clustering solution. In the second step, framework searches for a subset of all features that best fits the obtained consensus clustering solution by using PBIL. The similarity measure between clustering solutions is used to evaluate feature subsets. Elghazel and Aussem [6] extend random forest to unlabeled data by introducing the clustering ensemble to combine data resampling, and random selection of features strategies to generate an ensemble of component clustering. The proposed method randomly selects features and applies a clustering ensemble technique to find a suitable set of clusters. For each cluster, features are selected according to the scree test. Liu et al [16] cluster the features and filter out irrelevant or redundant ones from each cluster according to their membership probabilities. New features are obtained by combining features from different clusters. K-means clustering algorithm is trained on instances using these features. The ratio of intra-cluster distance to inter-cluster distance is used to evaluate these clusters, and the feature set with the largest ratio is selected as the final set. Yang et al. [24] use K-means clustering to generate class labels and then use joint maximization margin criterion and sparse L_{2,1}-norm regularization to perform feature selection. These steps iterate until the algorithm converges to global optimum, and the feature subset, which maximizes the margin between classes, is selected.
A filter-based method generally runs faster and is more preferable for real-world problems, especially those with large data sets. In the filter method, instead of performing clustering as part of the feature selection process, a quality measure is used to evaluate each feature set. This approach composes of two important components, a selection algorithm and a criterion function. The selection algorithm is the candidate feature maker while the criterion function is used to select features and evaluate feature subsets; however, clustering several times is not being performed as the wrapper approach does. Many researchers had found that it yields less effectiveness than the other two approaches; nevertheless, it is not true to state that the filter approach always yields inferior clustering results. Some criterion functions may provide equivalent or better performance than others to guide the search process.

Dash et al [2] propose an entropy measure and a forward search to select features. Entropy measure will be low if the data has distinct clusters while the measure will be high if otherwise. This measure is suitable for selecting the most important subset of features because it is invariant with number of dimensions, and is affected only by the quality of clustering. Features that give the minimum entropy for both of intra-cluster and inter-cluster distances are selected. Liu et al. [15] use the entropy measure proposed in [2] combined with the Laplacian score (LS) to select features for clustering. Features are ranked according to their LS in descending order. After that, the entropy measure of some features in the top of the list is evaluated and ranked in ascending order. Features with minimum entropy value are selected. This combination solves the drawbacks of using only the Laplacian score. Suri et al. [23] apply normalized mutual information (NMI) in [9] as an objective function for unsupervised feature selection in an outlier detection problem. Features with high NMI are considered as redundancy thus low NMI features are selected. In their work, average redundancy (AR) of a set of features is defined. One feature at a time is picked to form a feature subset and AR is used to evaluate subset and it is also used as stopping condition for the feature selection algorithm. Liu et al. [14] apply a forward search and a kernel least square error (LSE) for unsupervised feature selection. They propose an incremental LSE calculation to evaluate feature subsets in order to improve efficiency. To select a feature, the LSE of unselected features are evaluated. The feature with minimum value of LSE is selected; one feature at a time until stopping condition is reach.

The hybrid approach takes advantage of both wrapper and filter approaches. It applies a filter-based technique to select significant features, and applies a wrapper-based technique to add candidate features and evaluate candidate sets to select the best one. A few works are proposed in this approach for unsupervised learning. Dash and Liu [4] apply an entropy measure as a filter criterion. Data in well-formed clusters give low entropy values, so features are ranked in descending order of their entropy values. One feature at a time is picked to form a feature subset. Each subset is evaluated by K-means clustering, and a scattering criteria is used to evaluate cluster quality.

Problems usually found in real-world applications are mixtures of ambiguous and noisy data. This results in an inaccurate model which reduces its effectiveness. Fuzzy Logic, which is a multi-valued logic that allows intermediate values to be defined between conventional crisp evaluations, e.g., true/false, yes/no, etc., provides a simple way to define conclusions based upon vague, ambiguous, imprecise, noisy, or missing input information [8]. An element belongs to a set or class to a degree, indicating the certainty (or uncertainty) of membership. A membership function (MF), also referred to as the characteristic function of a fuzzy set, is used in the fuzzification process to produce the membership degrees. An inference process maps the level of membership received from the fuzzification process and produces a fuzzified output for each fuzzy value. Membership functions for fuzzy sets can be of any shape or type as determined by experts in the domain over which the sets are defined. In a fuzzy system, there are several patterns of MF such as triangular-, trapezoidal-, and Gaussian-shaped whose determination requires skills and knowledge about characteristics of the data to specify an appropriate pattern to use.

This paper proposes an unsupervised feature selection technique for clustering where features are first fuzzified using irregular-shaped membership functions evolved by a genetic algorithm which is suitable for high dimensional data. The technique is evaluated using standard machine learning data sets with varying sizes and complexities.

The rest of the paper is structured as follows. In the next section, a survey of feature selection methods and fuzzy feature selection is presented. Section 3 describes the proposed technique. Section 4, the technique is evaluated using standard data sets from the UCI repository and compared with past research. In the last section, the concluding remarks are drawn.
2. Backgrounds

Pudil et al. [20] introduce the sequential forward floating search (SFFS) using a criterion function to select a feature and compare candidate subsets. SFFS can be classified as a wrapper or a filter method, depending on the criterion function used. Its performance is good, but its computation time is high with a large data set. It consists of two phases: forward search and backward search. The forward search selects an unselected feature, and the backward search iteratively determines which member of the selected subset is to be removed if the remaining set improves the performance the most. The algorithm loops back to the forward search until the stopping condition is reached. Pseudo-code of SFFS is shown in Algorithm 1.

Algorithm 1

Input: $S$ is a feature set, $d$ is a predefined number of selected features, $J$ is a criterion function, and $J_k$ is the criterion function value of $k$ selected features.

Output: $S_{sel}$ is the selected feature set

Initialize: $k = 0$ and $S_{sel} = \emptyset$

(1) Inclusion
   a. If $k = d$ then terminate
   b. $x^* = \arg\max_{x \in S \setminus S_{sel}} J(S_{sel} \cup x)$
   c. $S_{sel} = S_{sel} \cup x^*, k = k + 1$

(2) Conditional exclusion
   a. $x^- = \arg\max_{x \in S_{sel}} J(S_{sel} \setminus x)$
   b. If $J(S_{sel} \setminus x^-) > J_{k-1}$ then
      i. $S_{sel} = S_{sel} \setminus x^-, k = k - 1$
      ii. Go to (2)
   Else go to (1)

Improved versions of SFFS have been proposed in various researches to provide better performance. Somol et al. [21] present ASFFS which keeps selecting features to be added and removed from the list. ASFFS adds one feature at a time until the number of features reaches a specified point. To remove features, a reverse of the inclusion is applied. Nakariyakul and Casasent [18] present IFFS which adds an extra step after the removing step to check whether replacing a weak feature can improve the criterion function value. If the replacement improves performance of the feature subset, the algorithm steps back to find another feature to remove again until the replacement can no longer improve the criterion function and goes back to the inclusion step to select a new feature from unselected ones. Sun et al. [22] propose an improvement to SFFS without the need to predefine the number of selected features. Mutual information and Parzen window estimator are the criterion functions to handle a mixture of continuous and categorical input features and continuous target features.

From the benefits of fuzzy logic in handling ambiguous and noisy data, a number of papers use the fuzzy set theory to improve performance of feature selection algorithm. Parthaláin and Jensen [19] propose an unsupervised subset evaluation measure based on the fuzzy rough set theory and a backward search to eliminate redundant features. A feature is removed if its removal does not worsen the effectiveness of the set. Ansari et al. [1] apply linear-shaped fuzzy MFs as weighting schemes to eliminate insignificant features which in their context are WWW URLs and also to remove noises from the data (very small user sessions), while preserving information of user sessions. Fuzzy c-means clustering is used to discover user profiles in this work.

3. The Proposed Method

The proposed technique is shown in Figure 4. An irregular-shaped membership function for every continuous attribute is evolved. Values of the attributes are thus fuzzified to create a suitable set of value ranges. All attributes are passed into a filter-based feature selection algorithm which employs two criterion functions to generate the best subset of features.
3.1. A Filter-Based Irregular-Shaped Membership Function Generation

An MF shape determined in advance by experts may not be suitable for a specific problem at hand, especially those with large and complex search spaces. We modify the wrapper-based hierarchical co-evolutionary genetic algorithm (HCGA) by [11] to generate an irregular-shaped MF where a criterion function is used, i.e., filter-based, instead of training and testing many classifiers, in order to improve efficiency.

The algorithm partitions and encodes possible solutions as populations in different levels, allowing for different kinds of chromosomes and genetic operations. A higher level chromosome selects a set of lower-level chromosomes to form a solution. In this case, a highly complicated search task can be properly partitioned into several subtasks which are simultaneously and effectively handled.

An MF shape is represented as one pivot point, left shoulder points and right shoulder points, depicted in Figure 1. Fuzzy partitions on each input variable are encoded in genetic segmentations and concatenated into one chromosome in the first level for the corresponding variable. A chromosome in the second level composes of genes pointing to chromosomes for all variables in L1-level. All point values are stored in genes. With GA operations, coordinate point values will change and result in changing shape. Constraints and repair schemes are applied before decoding the genetic representation. An L2-level gene contains the integer value of an index of the L1-level chromosome. The structure of two-level of HCGA is shown in Figure 2.

Furthermore, regarding out-bound cross point repairing, from experiments we find that if two consecutive shapes have pivot point values close to the new horizontal offset value obtained using the formula provided in the original paper; the repaired point can be moved to the next point, as shown in Figure 3(a). This results in multiple membership values, as shown in Figure 3(b). Therefore, in this research the horizontal coordinate of the midpoint between the left and the right points is used as the horizontal offset for the new cross point.
There are two membership values for the crisp data between $\mathbf{p}_m$ and $\mathbf{p}_{m+1}$.

### 3.2. Feature Selection Algorithm

A filter-based method most of the time yields less cluster validity. This research attempts to improve the quality of the feature sets selected from a filter-based method in order to achieve high cluster validity. We improve upon the filter-based SFFS by employing two criterion functions with different characteristics to complement each other and allowing more thorough search for features by introducing the candidate sets. During the inclusion pass, the first function ($J_{11}$) is used to evaluate and rank unselected features. The top-ranked features are candidates for inclusion. The second function ($J_{12}$) is then used to evaluate each candidate feature set of size $k+1$ created by combining a candidate feature with currently the best subset of size $k$. The feature to be selected for inclusion is the one that yields the best subset of size $k+1$ when evaluated by $J_{12}$.

Next, in the exclusion pass, $J_{12}$ is used to identify targeted features for exclusion. Each feature in the selected subset of size $k+1$ is evaluated and ranked in a descending order with $J_{12}$. Then, $J_{12}$ determines the best feature set of size $k$ by removing one feature at a time in the order and evaluates the remaining subset. The feature to be excluded is the one when removed gives the best subset of size $k$. The process iterates to smaller feature subsets if the best new subset of size $k$ is better than currently the best subset of the same size; then goes back to the inclusion step. The architecture of the propose technique is shown in Figure 4. Calculation by function $J_{12}$ usually takes a long time. The use of two criterion functions also helps reduce the calculation time since $J_{11}$ and $J_{12}$ which runs faster screens candidate features. This reduces the amount of calculations to be performed by $J_{2}$.

![Figure 4](image-url)
Input: $S$ is the original feature set, $S_{amnd}$ is a candidate set, $d$ is the predefined number of selected features, $J_1$ is the first criterion function, $J_2$ is the second criterion function.

Let $D$ be the total number of original features. $d_{sel}$ is the number of selected features. $d_{amnd}$ is the number of members in candidate set where $d_{amnd} \geq 1$. $S$ is the original set of features. $S_{sel}$ is the selected feature subset. $S_{amnd}$ is the candidate set in the backward step and $S_{amnd}^+$ is the candidate set in the forward step.

A feature to be added must be less redundancy to features already picked. $J_1$ in forward step is computed according to (1).

$$J_{11}(x_n) = m \cdot AR(x_n, S_{sel}) \text{ where } x_n \in S \setminus S_{sel}$$

In the forward step, each unselected feature is evaluated its $J_1$ and sorted in an ascending order according to its values of $J_1$. A candidate set is generated according to (2).

$$S_{amnd}^+ = \{x_n|x_n \in S \setminus S_{sel} \text{ and } n = [1..d_{amnd}] \text{ and } J_{11}(x_1) \leq J_{11}(x_2) \leq \cdots \leq J_{11}(x_{d_{amnd}})\}$$

A feature to be removed must be one providing the most redundancy where it produces the largest increase in MI with respect to the remaining $|S_{sel}| - 1$ features. Therefore, $J_1$ in backward step is computed according to (3).

$$J_{12}(x_n) = m \cdot \alpha \left\{ \sum_{n=1, n \neq x}^{d_{amnd}} NM(x_n, x_s) \right\} \text{ where } x_n \in S_{sel}, x_s = S_{sel} \setminus x_n$$

In backward step, each selected feature is evaluated its $J_1$ and sorted in a descending order. A candidate set is generated according to (4).

$$S_{amnd}^- = \{x_n|x_n \in S_{sel} \text{ and } n = [1..d_{amnd}] \text{ and } J_{12}(x_1) \geq J_{12}(x_2) \geq \cdots \geq J_{12}(x_{d_{amnd}})\}$$

### 3.3. Criterion Functions

Criterion functions play significant roles in feature selection. The purpose of the first function is to eliminate redundant features, thus the average redundancy (AR) measure is chosen [23] which can be calculated as follows.

$$AR(f_i, F_x) = \frac{1}{|F_x|} \sum_{f_j \in F_x} NM(f_i, f_j)$$

$$NM(f_i, f_j) = \frac{I(f_i, f_j)}{\min[H(f_i), H(f_j)]}$$

$$H(x) = -\sum_{j} p(x_j) \log(p(x_j))$$

$$I(x, y) = \sum_{i} \sum_{j} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$$

where $F_x$ is a set of selected feature, and $f_i$ is the feature to be evaluated.

AR is based on mutual information (MI) which has two main properties: its capacity of measuring relationships between variables and its invariance under space transformations [13]. Therefore, using AR selects only relevant features and eliminates redundant features that may misguide the search process.

In classification, feature selection aims at identifying features that predict class labels with the highest accuracy while in clustering feature selection aims at finding those features that discover the
natural grouping structures of the data. The entropy theory \cite{2} states that entropy of a system measures the disorder in the system. If the probability of each point is equal, we are most uncertain about the outcome, and the entropy value is maximum. On the other hand, when the data has well-formed clusters, the uncertainty is low and so is the entropy \cite{2}. Therefore, an entropy measure is chosen as the second criterion function for clustering problems. Entropy measure can be calculated as follows.

$$E = \sum_{x_i} \sum_{x_j} E_{ij} = \begin{cases} e^{(\beta \cdot D_{ij})} - e^0 & 0 \leq D_{ij} \leq \mu \\ e^{(\beta \cdot (\mu - D_{ij}))} - e^0 & \mu \leq D_{ij} \leq 1 \end{cases}$$

where $\beta$ is set to 10 from experiments, and $\mu$ is calculated as proposed in \cite{15}. $D_{ij}$ is the distance between 2 instances $i$ and $j$. Feature values after fuzzification are nominal, thus the hamming distance \cite{3} is used as the distance. $D_{ij}$ is calculated as follows.

$$D_{ij} = \frac{\sum_{k=1}^{M} |x_k \neq x_{jk}|}{M}$$

where $|x_k = x_{jk}|$ is 0 if $x_k$ equals $x_{jk}$, and 1, otherwise. M is the number of variables in the subset under consideration. The purpose of using entropy as the second criterion function is to assign low entropy to intra- and inter-cluster distances, and to assign higher entropy to noise \cite{15}. Therefore, this measure helps the proposed algorithm discover natural groupings of data.

### 3.4. The Proposed Algorithm

The proposed algorithm is shown in Algorithm 2. For every feature in $S_{at}$, we store the best of $J_{11}$, $J_{12}$ and $J_2$ values, and the feature subset in a lookup table to speed up the calculation.

**Algorithm 2**

Let $S_d$ be subset containing $d$ features

Initialization

$S_{at} = \text{empty set}$

$d = 0$

Begin

**Step 1: Inclusion step**

Use (2) to generate $S_{am+d}$

$x_i = \text{argmax}_{x \in S_{am+d}} J_2(S_{at} \cup x)$

If $J_2(S_{at} \cup x_i) < J_2(S_{at+d})$ then

$S_{at} = S_{at} \cup x_i$

$d = d + 1$

Else

$S_{at} = S_{at+d}$

**Step 2: Exclusion step**

Use (4) to generate $S_{am-d}$

$x_i = \text{argmin}_{x \in S_{am-d}} J_2(S_{at} \setminus x)$

If $J_2(S_{at} \setminus x_i) < J_2(S_d)$ then

$S_{at} = S_{at} \setminus x_i$

$d = d - 1$

Else

$S_{at} = S_{at-d}$

$J_2(S_{at-d}) = J_2(S_{at})$
$d = d - 1$

Go to step 2

else

Go to step 3

Step 3: Stopping criterion checking

If ($d = d_{\text{stop}} + \Delta$) or ($f_3(S_{\text{stop}}) > ALL_{LRF(F)}$) then exit an algorithm

Else go to step 1.

End.

The algorithm begins with an empty set of selected feature ($S_{\text{sel}}$). In the inclusion step, equation (2) is used to generate a candidate set of features ($S_{\text{and}}^+$) for inclusion. Each feature in $S_{\text{and}}^+$ is combined with the best subset of size $k$ to create a feature subset of size $k+1$ which is then evaluated by $f_2$. The best new subset of size $k+1$ will become the best subset if it is better than currently the best subset of the same size, according to $f_2$ function.

In the exclusion step, equation (4) is used to create a candidate feature set ($S_{\text{and}}^-$) for exclusion. Each feature in $S_{\text{and}}^-$ is removed from the best feature subset of size $k+1$ to give a new subset of size $k$. This subset is evaluated with $f_2$. The feature that is once removed gives the best subset of size $k$ is then excluded from the subset. The resulting subset of size $k$ is compared with currently the best subset of the same size, based on $f_2$. If the former subset is better, the algorithm iterates to refine the subset of size $k-1$. Otherwise, the algorithm goes back to the inclusion step to find the best feature of size $k+2$.

3.5. K-means Clustering

K-means clustering is an unsupervised learning algorithm. It is the most widely used technique to cluster data [12]. It is simple but efficient to partition data into clusters. The algorithm is an iterative process to group data into $K$ clusters, where the sum of within-cluster distances between point-to-cluster centroid over all clusters is minimized. In this work, we apply the squared Euclidean distance to measure the distance between a point and its centroid of cluster, which is the mean value of the points in that cluster. Squared Euclidean distance is evaluated as follows.

$$d = \sum_{i=1}^{n} (x_i - c)^2$$

where $x_i$ is a data instance, and $c$ is the group centroid.

The K-means algorithm starts by randomly select $K$ representative data from the raw data to be the centroid for each of the $K$ data groups. Then, it assigns each data to the closet group based on the distance measure. The algorithm updates the centroid value of the group for the mean of the data in that group, and repeatedly reassigns groups and updates centroids until the assignment cannot be changed.

4. Experimental Evaluations

To evaluate the effectiveness of our method, we use the evaluation method previously used in [5], [25], [7], and [17] where data sets with predefined classes are used. The data sets used in the evaluations of the proposed technique consists of 15 data sets from the UCI machine learning repository. Six data sets with small numbers of attributes are Iris (4 attributes), EColi (7 attributes), Pima (7 attributes), Wine (13 attributes), Parkinson (22 attributes), and Breast Cancer (30 attributes). Five data sets with moderate numbers of attributes are Ionosphere (34 attributes), Dermatology (34 attributes), Spambase (58 attributes), Sonar (60 attributes), and Movement Libras (90 attributes). And, three data set with large numbers of attributes is Hill Valley with Noise (100 attributes), Arrhythmia (279 attributes) and Madelon (500 attributes).

After performing clustering in each data set, each cluster will be assigned a label of class majority. The total number of correctly labeled data divided by the total number of instances in the data equals the accuracy of clustering. Since the K-means algorithm is sensitive to the initialization of cluster
centroids and the value of K, K is set as the number of classes, and the final accuracy is obtained from an average value after twenty runs.

After exploring various combinations of system parameters, the followings are used in all experiments. For the hierarchical co-evolutionary genetic algorithm, the population sizes of the first and second layer are set to 100 and 30, respectively, upper bound value = 0.8, lower bound = 0.1. Uniform mutation operator is used for both layers, and the candidate set size for the feature selection is \( d_{\text{amd}} = 5 \). For the data set without a separate test set provided, a 10-fold cross validation is used to measure the performance.

### 4.1. Effectiveness of the Proposed Technique

This section, we study the benefit of feature selection in comparison to the original set of features, the effectiveness of the proposed two-criterion function feature selection algorithm. Three feature selection approaches, i.e., the original filter-based; non-fuzzified two-criterion \( (f_1+f_2 \text{ without fuzzy}) \); and the proposed fuzzified two-criterion \( (f_1+f_2 \text{ with fuzzy}) \), are studied. The results (in Table 1) show that feature selection is beneficial and gives higher accuracy than does the original set of features that the proposed feature selection algorithm is effective where in 9 of 15 cases \( f_1+f_2 \text{ without fuzzy} \) yields better results than does \( f_2 \) alone. In 13 of 15 cases fuzzified \( f_1+f_2 \) gives better results than does the best of \( f_2 \) and \( f_1+f_2 \) without fuzzy configurations except for Spambase and Madelon data sets. For Madelon, the proposed method gives slightly lower accuracy (0.16%) with 20 less features than does the best method.

**Table 1.** The results of the original filter method \( f_2 \), \( f_1+f_2 \) without fuzzification, and \( f_1+f_2 \) with fuzzification

<table>
<thead>
<tr>
<th>Data set</th>
<th>Original data</th>
<th>( f_2 )</th>
<th>( f_1+f_2 ) Without Fuzzy</th>
<th>( f_1+f_2 ) With Fuzzy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>Accuracy</td>
<td>91.67</td>
<td>93.57</td>
<td>91.37</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>EColi</td>
<td>Accuracy</td>
<td>80.54</td>
<td>82.89</td>
<td>79.87</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Pima</td>
<td>Accuracy</td>
<td>66.02</td>
<td>73.18</td>
<td>66.02</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>7</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Wine</td>
<td>Accuracy</td>
<td>72.72</td>
<td>80.98</td>
<td>76.69</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>13</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Image Segmentation</td>
<td>Accuracy</td>
<td>32.68</td>
<td>56.83</td>
<td>62.45</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>19</td>
<td>16</td>
<td>15</td>
</tr>
<tr>
<td>Parkinson</td>
<td>Accuracy</td>
<td>75.39</td>
<td>75.39</td>
<td>75.39</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>22</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>Accuracy</td>
<td>84.28</td>
<td>87.36</td>
<td>85.41</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>30</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>Dermatology</td>
<td>Accuracy</td>
<td>35.70</td>
<td>75.79</td>
<td>85.85</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>34</td>
<td>29</td>
<td>17</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>Accuracy</td>
<td>71.22</td>
<td>74.93</td>
<td>74.93</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>34</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Spambase</td>
<td>Accuracy</td>
<td>60.81</td>
<td>62.23</td>
<td>68.74</td>
</tr>
<tr>
<td></td>
<td>Features</td>
<td>57</td>
<td>21</td>
<td>40</td>
</tr>
<tr>
<td>Sonar</td>
<td>Accuracy</td>
<td>57.89</td>
<td>59.93</td>
<td>57.57</td>
</tr>
</tbody>
</table>
4.2. Comparisons with Previous Research

Lastly, the proposed method is compared against 3 recent research on feature selection for clustering which consist of [10], [6], and [16], using common data sets and performance numbers reported in each paper. All of these methods use the wrapper approach where clustering are performed to evaluate each feature subset. Results are shown in Table 2, we can see that our method outperforms [10] in 4 of 5 data sets. Comparing with [6], our method outperforms in Iris data set but performs worse on the Ecoli data set. However, the proposed method gives higher performance than does [16] in 2 out of 3 common data sets.

Table 2. Experimental results of the proposed method compared with previously proposed methods.
The number of selected features is shown in parentheses.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>4</td>
<td>92.56 (2)</td>
<td>93.2 (1.78)</td>
<td>92.53 (2)</td>
<td>96 (1)</td>
</tr>
<tr>
<td>EColi</td>
<td>7</td>
<td>82.51 (6)</td>
<td>84.02 (5.3)</td>
<td>-</td>
<td>82.07 (6)</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
<td>87.07 (6)</td>
<td>-</td>
<td>94.46 (4)</td>
<td>89.89 (2)</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>30</td>
<td>77.86 (12)</td>
<td>-</td>
<td>-</td>
<td>90.33 (2)</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>34</td>
<td>59.17 (19)</td>
<td>-</td>
<td>-</td>
<td>77.35 (2)</td>
</tr>
</tbody>
</table>

5. Conclusions

As data sets grow in size and complexity, a feature selection technique is needed to select a small subset of highly representative features from the entire set of features. The technique is expected to reduce noises and distractions thus improve both effectiveness and efficiency. This article presents a filter-based technique to select a minimal set of features for clustering problems. The proposed technique employs fuzzification of original features using evolved irregular-shaped membership functions and a novel filter-based feature selection using two criterion functions. The first function is applied to eliminate features with redundant effects, and the second function is used to select a feature subset that yields well-formed clusters. The technique is evaluated using 15 standard UCI data sets and compared with 3 recent research papers. The results show that feature selection is beneficial to clustering, that the two-criterion feature selection algorithm is generally effective, and that the fuzzification improves the performance of the feature selection algorithm. In addition, the proposed technique performs well in comparison with the wrapper-based feature selection methods previously proposed in common data sets.
6. Reference