Australian Journal of Basic and Applied Sciences, 5(9): 1153-4463, 2011
ISSN 1991-8178

An Ensemble Based Approach for Feature Selection

[1]Hamid Parvin, [2]Hamid Alinejad-Rokny, [3]Maryam Asadi

[1]Islamic Azad University, Nourabad Mamasani Branch, Nourabad, Iran.
[2]Department of Computer Engineering, Science and Research Branch,
Islamic Azad University, Tehran, Iran.
[3]Islamic Azad University, Nourabad Mamasani Branch, Nourabad, Iran.

Abstract: In this paper we propose an ensemble based approach for feature selection. We aim at overcoming the problem of parameter sensitivity of feature selection approaches. To do this we employ ensemble method. We get the results per different possible threshold values automatically in our algorithm. For each threshold value, we get a subset of features. We give a score to each feature in these subsets. Finally by use of ensemble method, we select the features which have the highest scores. This method is not a parameter sensitive one, and also it has been shown that using the method based on the fuzzy entropy results in more reliable selected features than the previous methods'. Empirical results show that although the efficacy of the method is not considerably decreased in most of cases (or it is even increased the performance in the most cases), the method becomes free from setting of any parameter.

Key words: Feature Selection, Ensemble Methods, Fuzzy Entropy

INTRODUCTION

We have to use features of a dataset to classify data points in pattern recognition and data mining. Some datasets have a large number of features. Processing these datasets is not possible or is very difficult. To solve this problem, the dimensionalities of these datasets should be reduced. To do this, some of the redundant or irrelevant features should be eliminated. By eliminating the redundant and irrelevant features, the classification performance over them will be improved. Three different approaches are available for feature selection mechanism (Tan, 2005). The first ones are embedded approaches. In these algorithms, feature selection is done as a part of the data algorithm. The second ones are filter approaches. These algorithm selected features before the data mining algorithm is run. The last ones are wrapper approaches. In these algorithms the target data mining algorithm is used to get the best subset of features.

In recent years, a lot of methods for subset selection have been presented, such as similarity measures (Tsang, 2003), gainentropies (Caruana, 1994), the relevance of features (Baim, 1988), the genetic algorithms method (Chaikla, 1999), the overall feature evaluation index (OFEI) (De, 1999), the feature quality index (FQI) (De, 1999), the mutual information-based feature selector (MIFS) (Battiti, 1994), classifiability measures (Dong, 2003), neuro-fuzzy approaches (De, 1997; Platt, 1999), fuzzy entropy measures (Shie, 2007), etc.

This paper is based on Shie-and-Chen's method (2007). In Shie-and-Chen's method by use of the previous fuzzy entropy measurements and also by explaining some new definitions, the authors present a new algorithm for feature selection problem. This algorithm can select appropriate features more accurately than the other algorithms. The new definitions are explained in the following section. This method uses boundary samples instead of the full set of samples. Boundary samples are some kinds of samples which are incorrectly classified samples of previously selected features. It uses two different threshold values to calculate the entropies. The most important weakness of the algorithm is raises from its sensitiveness to user-defined threshold values. User has to test different threshold values, and finally selects ones which cause the best performance. In other words this algorithm is a kind of the parameter sensitive algorithms. For a new dataset a lot of different values for two threshold values must be tested and then the best ones are selected.

In this paper we try to improve Shie-and-Chen's method. We try to solve the drawback of parameter sensitivity. To do this we use ensemble method. We get the results for different threshold values. For each threshold values, we get a subset of features. We give a score to each feature in these subsets. Finally by use of ensemble concept, we select the features which have the highest scores. This method is not a parameter sensitive one, and also it has been shown that using the method based on the fuzzy entropy results in more reliable selected features than the previous methods'.

Our contributions are three-folded.

1. We propose a novel ensemble approach in the feature selection.
2. We propose a novel method to be got rid of the drawback of parameter sensitivity for feature selection.
3. We will show empirically that the ensemble-based approach for feature selection is fully automated and parameterless, also it outperforms the original version.

Corresponding Author: Hamid Parvin, Islamic Azad University, Nourabad Mamasani Branch, Nourabad, Iran.
E-mail: {asadi, parvin}@iust.ac.ir , alinejad@ualberta.ca

2. Previous Fuzzy Feature Selection:

In this section, we review the existing fuzzy entropy measures. Fuzzy entropy is used to describe the impurity of datasets. We explain some definitions base on that one can find how to calculate the entropy value. Assume that *X* is a discrete random variable which contains *n* elements. If the probability distribution of x_i is $p(x_i)$, the amount of information $I(x_i)$ associated with x_i is defined as below.

$$I(x_i) = -\log_2 p(x_i)$$

The entropy *H(X)* of *X* is defined as below:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

There is a definition for fuzzy entropy concept. Let us is a fuzzy set. For a finite set $X = \{x_1, x_2, \ldots, x$ with respect to the probability distribution $P = \{p_1, p_2, \ldots, p$, the fuzzy entropy value for fuzzy set is calculated as below:

$$H = -\sum_{i=1}^{n} \mu_{\tilde{A}}(x_i) p_i \log p_i$$

where is the membership function of fuzzy set, $\mu_{\tilde{A}}($ is the grade membership of belonging to fuzzy set is the probability of and $1 \le i \le$.

There is a fuzzy entropy measure [12] based on Shannon's entropy [13]. Let us assume *A* is a fuzzy set defined in the universe of discourse *X*, and denotes the membership function of the fuzzy set A, so $\mu_A($ denotes the grade of membership of *x* belonging to the fuzzy set *A*, and $x \in$. There are some axioms of a fuzzy entropy measure *H(A)* of a fuzzy set *A* which are shown as below:

Axiom 1: *H(A)* = 0 iff $A \in$ is a crisp set.

Axiom 2: *H(A)* is the maximum iff $\mu_{A(x)} = 0.5, \forall x \in$

Axiom 3: if is less fuzzy than, then $H(\tilde{A}) \le H($.

Axiom 4: $(A) = H($, where $A^c = 1 -$, i.e., denotes the complement of *A*.

Also fuzzy entropy concept is described as below [14]:

$$H = -K \sum_{j=1}^{n} [(\mu_A(x_j) \log \mu_A(x_j) + (1 - \mu_A(x_j)) \log (1 - \mu_A(x_j))]$$

There is another definition for fuzzy entropy concept based on the geometry of hypercube by using the concepts of overlap and underlap [15].

$$H(A) = \frac{\sum_{i=1}^{n} (\mu_A(x_i) \wedge \mu_A^c(x_i))}{\sum_{i=1}^{n} (\mu_A(x_i) \vee \mu_A^c(x_i))}$$

Another fuzzy entropy concept is presented based on Shannon's entropy measure [13] and Luca's axioms [2]. In this method *R* is the set of samples. Also *C* denotes the set of classes defined in *R*. Assume that a feature is divided into *I* intervals is a subset of *R* which is distributed in the *i*th interval is a subset of which is labeled as class $c \in$. In this definition *M* is the matching degree of the samples of class *c* in the *i*th interval belonging to the fuzzy set.

$$MD_c(\tilde{A}) = \frac{\sum_{r \in R_{ic}} \mu_{\tilde{A}}(r)}{\sum_{r \in R_i} \mu_{\tilde{A}}(r)}$$

Fuzzy entropy value of the samples of class *c* in the *i*th interval belonging to the fuzzy set is defined as below:

$$IFE_c(\tilde{A}) = -MD_c(\tilde{A}) \log_2 MD_c(\tilde{A})$$

Fuzzy entropy value for in the *i*th interval is calculated as below:

$$IFE(\tilde{A}) = \sum_{c \in C} IFE_c(\tilde{A})$$

Entropy fuzzy value for *i*th interval in a feature dimension is shown as below:

$$TFE_i = \sum_{v \in V_i} IFE(v)$$

Where is the set of fuzzy sets in the *i*th interval in a feature?

Another definition is presented for fuzzy entropy concept (Shie, 2007). Assume that *X* is a set of samples which is divided into a set of *C* classes. Class degree is shown by:

$$CD_c(\tilde{A}) = \frac{\sum_{x \in X_c} \mu_{\tilde{A}}(x)}{\sum_{x \in X} \mu_{\tilde{A}}(x)}$$

is the set of samples of class *c* defined by fuzzy set. $\mu_{\tilde{A}}($ denotes the membership grade for each *x* in fuzzy set and $\mu_{\tilde{A}}(x) \in [0.$

Fuzzy entropy value for the samples of the class *c* defined in fuzzy set is calculated as below:

$$FE_c(\tilde{A}) = -CD_c(\tilde{A}) \log_2 CD_c(\tilde{A})$$

Fuzzy entropy for fuzzy set is shown as below:

$$FE(\tilde{A}) = \sum_{c \in C} FE_c(\tilde{A})$$

Fuzzy entropy of feature *f* is defined as below:

$$FFE(f) = \sum_{v \in V} \frac{S_v}{S} FE(v)$$

The set of fuzzy sets for feature *f* is shown by *V*. *FE(v)* stands for the fuzzy entropy for each subset *v*. *S* is the summation of the membership grades of the samples belonging to each fuzzy set of the feature *f*, and is the summation of the membership grades for the samples belonging to the fuzzy set *v*.

The membership function of features is divided in two groups. Each feature is a nominal feature or a numeric feature. For nominal features, a fuzzy set is defined for each value of feature as below:

$$\mu_u(x) = \begin{cases} 1, & if\ x = u \\ 0, & otherwise \end{cases}$$

U is the set of nominal features. $\mu_u($ denotes the membership function of the fuzzy set u, and $u \in$.

We have to discretize numeric features to produce the membership function. To do this, K-means clustering algorithm is used. This algorithm divides the domain of the numeric feature to *k* clusters. Membership function can be produced by using the centers of these clusters. For this purpose, the centers of these clusters are used as the centers of fuzzy subsets. On one hand entropy decreases when the number of clusters increases. On the other hand increasing in the number of clusters causes over fitting. To solve this problem [11] uses a threshold value (T_c). This value for each data set is selected experimentally by the user. User has to test this algorithm for different threshold values, and finally select the best one which has the best result. When the decreasing rate of fuzzy entropy of a feature between two steps is less than T_c, they stop increasing the number of clusters. The algorithm is shown in Fig. 1.

First step initializes *k* by the number of clusters. Second step runs K-means Clustering algorithm. For this part, at first the centers of clusters are initialized. Then algorithm assigns each sample to a proper cluster. The center of the proper cluster should have the minimum Euclidean distance with its samples. Then cluster centers are recalculated. This step is done repeatedly until each cluster is not changed. Third step constructs the membership functions for each set of the possible fuzzy sets based on these k cluster centers. Fourth step calculates a fuzzy entropy value per each feature. Finally fifth step controls decreasing rate of the fuzzy entropy value by threshold value.

Step 1:
Initially, set the number k of clusters
Step 2:
For $i = 1$ *to* k *do*
Let $m_i = x;$
Repeat
{
For all $x \in$
Let $i = arg\,min_{x \in X} \|x - m_h;$
Let $Cluster_i = Cluster_i \cup \{;$
For $i=1$ *to* k *do*
Let $m_i = \frac{\sum_{x \in Cluste}}{n_i};$
} until each cluster set is not changed.
Step 3:
Let $m_L = \begin{cases} U_{min} - (m_i - U_{min}), & if\ i = 1 \\ m_{i-1}, & otherwi \end{cases}$
Let $m_R = \begin{cases} U_{max} + (U_{max} - m_i), & if\ i = i \\ m_{i+1}, & otherwi \end{cases}$
$\mu_{v_i}(x) = \begin{cases} Max\left\{1 - \frac{m_i - x}{m_i - m_L}, 0\right\}, & if\ x \leq \\ Max\left\{1 - \frac{x - m_i}{m_R - m_i}, 0\right\}, & if\ x \geq \end{cases}$
Step 4:
For $i=1$ *to* k *do*
Let $FE(v_i) = \sum_{c \in C} FE_c($
Let $FFE(f) = \sum_{v \in V} \frac{S_v}{S} FE($
Step 5:
If the decreasing rate of the fuzzy entropy is larger than the threshold value Tc,
Let k = k + 1 and go to Step 2. Otherwise, then let k = k - 1 and Stop

Fig. 1: Preprocessing phase of fuzzy feature selection algorithm.

The proposed algorithm for selecting a proper subset of the original features uses only boundary samples instead of all samples. Boundary samples are those samples which are incorrectly classified. Extension matrix is defined for each feature to show the membership grades of a feature value as below:

$$\mathrm{EM_f} = \begin{bmatrix} \mu_{v1}(r_{1f}) & \cdots & \mu_{vm}(r_{1f}) \\ \vdots & \vdots & \vdots \\ \mu_{v1}(r_{nf}) & \cdots & \mu_{vm}(r_{nf}) \end{bmatrix}_{n \times m}$$

In this matrix *m* denotes the number of fuzzy sets for feature *f* and *n* denotes the number of samples. $\mu_{vi}(r_$ stands for the membership grade of the value for the feature *f* of the sample belonging to the fuzzy set.

There is a definition for calculating class degree for the samples of class *c* belonging to the fuzzy set *v* as below:

$$\mathrm{CD_c(v)} = \frac{\sum_{r \in R_c} \mathrm{EM_f(|r|, |v|)}}{\sum_{r \in R} \mathrm{EM_f(|r|, |v|)}}$$

R is a set of samples, and denotes the samples of class *c* in *R*. The fuzzy entropy *FFE(f)* of a feature *f* can be calculated as below:

$$FFE(f) = \sum_{v \in V} \left[\frac{S_V}{S} \times \sum_{c \in C} (-CD_c(v) \log_2 CD_c(v)) \right]$$

To calculate entropy fuzzy of a feature subset, a new combined extension matrix is defined as below:

$\mathrm{CEM(f_1, f_2, '=}$

$$\begin{bmatrix} \mu_{v_{11}}(r_{1f_1}) \wedge \mu_{v_{21}}(r_{1f_2}) & \cdots & \mu_{v_{11}}(r_{1f_1}) \wedge \mu_{v_{2j}}(r_{1f_2}) & \cdots & \mu_{v_{1i}}(r_{1f_1}) \wedge \mu_{v_{21}}(r_{1f_2}) & \cdots & \mu_{v_{1i}}(r_{1f_1}) \wedge \mu_{v_{2j}}(r_{1f_2}) \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \mu_{v_{11}}(r_{nf_1}) \wedge \mu_{v_{21}}(r_{nf_2}) & \cdots & \mu_{v_{11}}(r_{nf_1}) \wedge \mu_{v_{2j}}(r_{nf_2}) & \cdots & \mu_{v_{1i}}(r_{nf_1}) \wedge \mu_{v_{21}}(r_{nf_2}) & \cdots & \mu_{v_{1i}}(r_{nf_1}) \wedge \mu_{v_{2j}}(r_{nf_2}) \end{bmatrix}_{n \times ij}$$

A threshold parameter given by the user is used to create this matrix. In this matrix i is the number of fuzzy sets defined in the feature f_1 whose maximum class degree is smaller than the given threshold value . j is the number of fuzzy sets defined in the feature f_2 whose maximum class degree is smaller than the given threshold value. $\mu_{v_{1x}}(r_p$ shows the membership grade of the value of the feature f_1 of the sample r_p belonging to a fuzzy set of the feature f_1, where $1 \leq x$, and $\mu_{v_{2y}}(r_p$ shows the membership grade of the value (of the feature f_2 of the sample r_p belonging to a fuzzy set of the feature f_2, where $1 \leq y$. Also $\mu_{v_{1x}}(r_{pf_1}) \wedge \mu_{v_{2y}}(r_p$ is the membership grade of the values of the feature subset $\{f_1, f_2\}$ of the sample r_p belonging to the combined fuzzy set $v_{1x/}$ of the feature subset $\{f_1, f_2\}$, where denotes the fuzzy minimum operator.

Finally the entropy fuzzy of the feature subset $\{f_1, f_2\}$ is calculated as below:

$$\text{BSFFE}\ (f_1, f_2, T_r) = \begin{cases} \frac{S_{1B}}{S_1} \times \sum_{w \in V_{FS}} \frac{S_w}{S_{FS}} FE(w) + \sum_{v1 \in V_{1UB}} \frac{S_{v1}}{S_1} FE(v_1), & \text{if } \frac{S_{1B}}{S_1} \leq \frac{S_{2B}}{S_2} \\ \frac{S_{2B}}{S_2} \times \sum_{w \in V_{FS}} \frac{S_w}{S_{FS}} FE(w) + \sum_{v2 \in V_{2UB}} \frac{S_{v2}}{S_2} FE(v_2), & \text{otherwise} \end{cases}$$

The summation of the membership grades of different values of the feature f_1 for the samples belonging to each fuzzy set of feature f_1 is shown by S_1. is the summation of the membership grades of the values of the feature f_1 for the samples belonging to the fuzzy sets of the feature f_1 whose maximum class degree is smaller than the threshold value given by the user. shows the set of combined fuzzy sets of feature $\{f_1, f_2\}$, denotes the summation of the membership grades of the set of the values of the feature subset $\{f_1, f_2\}$ for the samples belonging to each combined fuzzy set of the feature subset $\{f_1, f_2\}$. is the summation of the membership grades of the values of the feature subset $\{f_1, f_2\}$ for the samples belonging to a combined fuzzy set w, $FE(w)$ denotes the fuzzy entropy of a combined fuzzy set w, is set of fuzzy sets of the feature f_1 whose maximum class degree is larger than or equal to the threshold value . is the summation of the membership grades of the values of the feature f_1 of the samples belonging to a fuzzy set v_1 of the feature f_1. $FE(v_1)$ is the fuzzy entropy of a fuzzy set v_1 of the feature f_1. One can defines some equivalent definitions like these ones for f_2. The algorithm is presented in Fig. 2.

3. Proposed Algorithm:

The algorithm which is presented Fig. 2 is parameter sensitive. So if these parameters change, the result of algorithm can be changed significantly. When these parameters are given by the user, the quality of algorithm results will be even weaker. Because user selects the parameters randomly and experimentally, so it is possible that they are not proper values for an exemplary dataset. So the result of algorithm is not trustable. Also the proper values are not available for some datasets which are not used in this algorithm. So to find the best result we need to test the algorithm for a lot of possible threshold values. Then we must select the threshold values which cause the best results. To solve this problem we use ensemble method.

We do not select threshold values experimentally in our algorithm. Our algorithm test different possible values for thresholds and then by doing some steps, it selects the subset of features. This algorithm has 5 steps. We employ Shie-and-Chen's method by a little change in our algorithm. The result of their algorithm is a subset of features. But we get a sequence of features instead of a subset. Actually the order of feature appearance is important in our algorithm.

First step runs Shie-and-Chen's method for each pair of (T_r , T_c). The result of algorithm at this step is a table of feature sequences which are selected for each pair of threshold values. For example the result of our algorithm for Iris is shown in Table 1. We obtained this result for 5 different values for T_c and T_r. Each element in this table is a feature sequence selected by the algorithm of Fig. 2 with a different pair of threshold values. The first step of the algorithm is as Fig. 3.

It has two loops. One of them slides over T_r and the other one slides over T_c. Two parameters *base_t*$_r$ and *base_t*$_c$ are the minimum values used for T_r and T_c respectively. Two parameters *step_t*$_r$ and *step_t*$_c$ determine the distance between two consecutive threshold values of parameters T_r and T_c respectively. *FSeq* is a two dimensional matrix whose elements are features sequences obtained by the algorithm of Fig. 4 with each possible tested pair of threshold values.

FSeq= Shie-and-Chen's Algorithm(T_r, T_c)
F is a set of candidate feature, FS is the selected feature subset
FSeq is the selected feature sequence.
Step 1:
For each f ∈ *do*
Let $EM_f = \begin{bmatrix} \mu_{v1}(r_{1f}) & \cdots & \mu_{vm}(r_{1f}) \\ \vdots & \vdots & \vdots \\ \mu_{v1}(r_{nf}) & \cdots & \mu_{vm}(r_{nf}) \end{bmatrix}_n$
Let *E(f)= FFE(f)*
Step 2:
Let *i=1*
Let $\ddot{f} = arg\,min_{f \in F} E($
Let $E_{FS} = E($
Let $FS = FS \cup \{$
Let $FSeq(i) =$
Let *i= i+1*
Let $F = F - \{$
Step 3:
repeat
for each $f \in$ *do*
Let EM_{temp}*= CEM(FS,f,*T_r*)*
Let *E(f)= BSFFE(FS,f)*
Let $\ddot{f} = arg\,min_{f \in F} E($
Let $FS = FS \cup \{$
Let $FSeq(i) =$
Let *i= i+1*
Let $F = F - \{$
Let $D = E_{FS} - E($
Let $E_{FS} = E($
until $(FFE(FS) = 0$ *or* $D \leq 0$ *or* $F =$
Let FS be the selected feature subset and FSeq be the selected feature sequence.

Fig. 2: Shie-and-Chen's Algorithm with a simple modification suitable for proposed fuzzy feature selection algorithm.

For T_r *= base_*t_r*: step_*t_r *:1*
For T_c *= base_*t_c*: step_*t_c *:1*
AllFSeq(T_r*,* T_c*) = Shie-and-Chen's algorithm(*T_r*,* T_c*);*

Fig. 3: Pseudo code of the first step of algorithm.

Table 1: Feature subsets selected for some pairs of threshold values over Iris dataset.

Tc, Tr	0.01	0.21	0.41	0.61	0.81
0.01	4, 3	3, 4	3, 4	3, 4	3, 4
0.21	4, 3	4, 3	3, 4	3, 4	3, 4
0.41	3, 4	4, 3	3, 4	3, 4	3, 4
0.61	4, 1	4, 2	3, 1	3, 1	3, 1
0.81	3, 1	4, 3, 1	3, 4	3, 4	3, 4

As it is inferred from Table 1, at the first and the last rows of each column we have some similar results for some threshold values. There is a similar discussion about the first and the last columns of each row. The results of algorithm for the first and the last columns of each row and the first and the last rows of each column are not trustable to reach some proper threshold values. Since these results have strongly negative effect on the final evaluation, at the second step we have to remove these repetitions. This step has two parts. The first part removes the repetitions of columns and the second part removes the repetitions of rows. First part keeps only the results at the beginning and ending of each column to reach a dissimilar result at the beginning and ending of each column. And the second part keeps only the results at the beginning and ending of a row to reach a dissimilar result at the beginning or ending of each row. In other words, we use only one of the same results at the beginning and ending parts of each row and each column in final evaluation.

Equation 1 is a function that checks the similarity of its inputs. It has two input parameters which can be two sequences of features. If they are similar, the output will be 1 and if they are not similar the output is 0.

$$is\_same(a, b) = \begin{cases} 1, & if\ x = \\ 0, & otherwi \end{cases}$$

The following pseudo code is the first part of second step of the algorithm.

```
New_AllFSeq = AllFSeq
For        Tr = base_tr: step_tr :1
   q = base_ tc
   While (true)
          q = q + step_ tc
          if       is_same ( AllFSeq ( Tr , base_tc ) , AllFSeq ( Tr, q ))
                   New_AllFseq ( Tr , q ) = EmptySeq
          else
                   break
   q = last_ tc
   While (true)
          q = q - step_ tc;
          if       is_same ( AllFSeq ( Tr , last_tc ) , AllFSeq ( Tr, q ))
                   New_AllFseq ( Tr , q ) = EmptySeq
          else
                   break
```

Fig. 4: Pseudo code of the first part of the second step of the algorithm.

It checks the similarity between the first sequence of a column and the consecutive sequences of that column. By reaching the first dissimilar sequence at the beginning or ending of a column, this part of algorithm is done for each column. Output for Iris example of doing the first part of the second step of the algorithm is available in Table 2.

Table 2: Delete repetitions in columns of Table 1.

T_c, T_r	0.01	0.21	0.41	0.61	0.81
0.01	4, 3	3, 4	3, 4	3, 4	3, 4
0.21	+	+	+	+	+
0.41	3, 4	4, 3	+	+	+
0.61	4, 1	4, 2	3, 1	3, 1	3, 1
0.81	3, 1	4, 3, 1	3, 4	3, 4	3, 4

Also the second part of the second step of the algorithm is as the algorithm of Fig. 5. It is like the first part of the second step. It checks the similarity between the first sequence of a row and the other sequences in that column. By reaching the first dissimilar sequence at the beginning or ending of a row, this part of algorithm is done for each row.

Result of doing the second part of the second step of the algorithm over the Iris dataset which is obtained from the first step is shown in Table 3.

Table 3: Deleting repetitions in rows of Table 2.

Tc, Tr	0.01	0.21	0.41	0.61	0.81
0.01	4, 3	*	*	*	3, 4
0.21	+	+*	+*	+*	+
0.41	3, 4	4, 3	+*	+*	+
0.61	4, 1	4, 2	*	*	3, 1
0.81	3, 1	4, 3, 1	*	*	3, 4

Third step uses majority voting to reach the best subset of features. We have to give a score to each feature. There is a subset of selected features for each pair of T_r and T_c. We change this subset to a sequence of features by their ranks of appearing at the first step. In other words each feature that appears sooner has more effect on output, so it is given a higher score. Then we sum all given scores to features for each pair of threshold values. We define the score of each feature as equation 2.

```
For        T_c = base_t_c: step_t_c :1
    q = base_ t_r
    While (true)
            q = q + step_ t_r
            if      is_same ( AllFSeq ( T_c , base_t_r ) , AllFSeq ( T_c, q ))
                    New_AllFseq ( T_c , q ) = EmptySeq
            else
                    break;
    q = last_ t_r
    While (true)
            q = q - step_ t_r
            if      is_same ( AllFSeq ( T_c , last_t_r ) , AllFSeq ( T_c, q ))
                    New_AllFseq ( T_c , q ) = EmptySeq
            else
                    break
AllFseq=New_AllFseq
```

Fig. 5: Pseudo code of the second part of the second step of the algorithm.

After obtaining Table 3 for each dataset, we give a score to each of its features according to equation 2. In the equation 2, we give the higher weight to the first feature which appears sooner, and we give the lower weight to the last feature which appears at the end of the sequence. For example if there are 10 features, the weight of the first feature is considered 10, and the weight of the last feature is considered 1.

$$Score\ (f) = \sum_{T_r}\sum_{T_c}\sum_{i=1}^{MaxSF} isequal(\ AllFSeq\ (T_r,T_c)(i), f) * (|AllFSeq| - i + 2$$

where *MaxSF* is obtained by equation 3.

$$MaxSF = max_{T_r,T_c,i}(|AllFSeq\ (T_r,T_c)(i)|)$$

Finally we sum all the weighted scores obtained by the algorithm for different pairs of threshold values. For example, in the Iris example the *MaxFS* is 3. In the example we get these results:

Score (3) = 21, Score (4) = 21, Score (1) = 7 and Score (2) = 2

Then we sort all features by their scores. After that we select the features with maximum scores. We select the same number of features as the Shie-and-Chen's method. In Iris example the subset of {3, 4} features is selected as final selected subset, because these features have the highest scores, and Shie-and-Chen's method selected two features for this example.

3 Experimental Results:

In (Shie, 2007) they tested their algorithm in two stages. Their first experiment is compared with some previous methods in Table 4. These methods are OFFSS, OFEI, FQI and MIFS. This table shows the feature subsets selected by some methods. They use four datasets in this stage. These data sets are Iris, Breast Cancer Diagnostic, Pima Diabetes and Mile Per Gallon (MPG).

Table 4: A comparison of feature subsets selected by previous methods (Shie, 2007).

Data set	Feature subsets selected by different methods				
	Shie-and-Chen's	MIFS	FQI	OFEI	OFFSS
Iris	{4, 3}	{4, 3}	{4, 3}	{4, 3}	{4, 3}
Breast Cancer	{6, 2, 1, 8, 5, 3}	{6, 3, 2, 7}	{6, 1, 8, 3}	{6, 1, 3, 2}	{6, 3, 1, 2}
Pima Diabetes	{2, 6, 8, 7}	{2, 6, 8}	{8, 2, 1}	{2, 3, 6}	{2, 6, 7}
MPG	{4, 6, 3}	{4, 6, 2, 1}	{4, 6, 3, 2}	{4, 5, 6, 2}	{6, 2, 5, 4}

Table 5 shows the accuracies of different classifiers on the selected features obtained by the methods used in Table 4. It shows that the different classifiers on the selected features obtained by Shie-and-Chen's method have better accuracies than the other methods. It uses four classifiers to compare these methods. These classifiers are LMT, Naive Bayes, SMO and C4.5.

The second experiment is on five datasets and three problems. These datasets are Pima Diabetes, Cleve, Correlated, M of N-3-7-10 and Crx datasets, also Monk-1, Monk-2 and Monk-3 problems. They compare their method with Dong and Kothari's method. Table 6 shows the feature subsets which are selected by these algorithms.

Table 5: A comparison between the average classification accuracy rates of previous methods (Shie, 2007).

Data sets	Classifiers	Average classification accuracy rates of different methods				
		OFFSS	OFEI	FQI	MIFS	Shie-and-Chen's method
Iris	LMT	94.67 4.27%	94.67 4.27%	94.67 4.27%	94.67 4.27%	94.67 4.27%
	Naive Bayes	96.00 ± 4.00%	96.00 ± 4.00%	96.00 ± 4.00%	96.00 ± 4.00%	96.00 ± 4.00%
	SMO	96.00 ± 4.00%	96.00 ± 4.00%	96.00 ± 4.00%	96.00 ± 4.00%	96.00 ± 4.00%
	C4.5	96.00 ± 5.33%	96.00 ± 5.33%	96.00 ± 5.33%	96.00 ± 5.33%	96.00 ± 5.33%
Breast cancer	LMT	95.90 ± 2.15%	95.90 ±2.15%	96.49 ±2.09%	95.46 ± 1.79%	96.49 ± 2.08%
	Naive Bayes	96.19 ± 2.56%	96.19 ±2.56%	96.49 ±1.88%	95.31 ± 1.58%	96.63 ± 1.97%
	SMO	96.34 ± 2.19%	96.34 ±2.19%	97.07 ±1.85%	96.05 ± 2.62%	97.07 ± 2.27%
	C4.5	95.61 ± 2.70%	95.61 ±2.70%	96.93 ±1.90%	95.16 ± 2.86%	96.02 ± 2.57%
Pima di-abetes	LMT	76.83 ± 3.79%	76.04 ±3.63%	73.56 ±4.68%	75.53 ± 4.39%	77.22 ± 4.52%
	Naive Bayes	76.57 ± 3.65%	76.83 ±4.36%	74.09 ±5.43%	76.44 ± 5.50%	77.47 ± 4.93%
	SMO	75.91 ± 4.96%	75.91 ±3.80%	75.39 ±4.93%	75.91 ± 4.97%	77.08 ± 5.06%
	C4.5	75.01 ± 3.72%	74.36 ±4.27%	71.74 ±3.18%	74.61 ± 4.86%	74.88 ± 5.89%
MPG	LMT	81.13 ± 5.67%	81.13 ±5.67%	82.38 ±7.28%	84.17 ± 7.26%	81.87 ± 6.74%
	Naive Bayes	78.31 ± 7.63%	78.31 ±7.63%	79.59 ±6.79%	76.28 ± 8.25%	80.60 ± 7.01%
	SMO	80.58 ± 7.21%	80.58 ±7.21%	81.61 ±6.99%	76.77 ± 4.12%	81.86 ± 8.25%
	C4.5	79.83 ± 7.84%	79.83 ±7.84%	79.58 ±8.24%	81.37 ± 9.05%	79.93 ± 7.78%

Table 6: Feature subsets selected by Dong-and-Kothari's method and Shie-and-Chen's method.

Data sets	Feature subsets selected by different methods	
	Dong-and-Kothari's method	Shie-and-Chen's method
Pima diabetes data set	{2, 8, 1}	{2, 6, 8, 7}
Cleve data set	{10, 13, 12, 3, 9}	{13, 3, 12, 11, 1, 10, 2, 5, 6}
Correlated data set	{6, 1, 2, 3, 4}	{6, 1, 2, 3, 4}
M of N-3-7-10 data set	{4, 9, 5, 8, 3, 6, 7}	{4, 9, 8, 5, 3, 6, 7}
Crx data set	{8, 9, 13, 10}	{9}
Monk-1 data set	{5, 1, 2}	{5, 1, 2}
Monk-2 data set	{3, 6, 1, 2, 4, 5}	{5}
Monk-3 data set	{2, 5, 4, 1}	{5, 2, 4}

Table 7 shows the accuracy of these two methods. It uses the same classifier as Table 4.

Table 7: A comparison between the average classification accuracy rates of Dong-and-Kothari's method and Shie-and-Chen's method.

Data sets	Classifiers	Average classification accuracy rates of different methods	
		Dong-and-Kothari's method	Shie-and-Chen's method
Pima diabetes data set	LMT	73.56 ± 4.68%	77.22 ± 4.52%
	Naive Bayes	73.43 ± 1.57%	77.47 ± 4.93%
	SMO	75.39 ± 4.93%	77.08 ± 5.06%
	C4.5	71.74 ± 3.18%	74.88 ± 5.89%
Cleve data set	LMT	83.17 ± 4.24%	82.87 ± 6.23%
	Naive Bayes	84.17 ± 1.82%	84.48 ± 3.93%
	SMO	84.47 ± 5.59%	83.51 ± 6.09%
	C4.5	76.90 ± 8.71%	76.90 ± 8.40%
Correlated data set	LMT	100.00 ± 0.00%	100.00 ± 0.00%
	Naive Bayes	86.03 ± 3.75%	86.03 ± 3.75%
	SMO	89.87 ± 6.88%	89.87 ± 6.88%
	C4.5	94.62 ± 4.54%	94.62 ± 4.54%
M of N-3-7-10 data set	LMT	100.00 ± 0.00%	100.00 ± 0.00%
	Naive Bayes	89.33 ± 1.56%	89.33 ± 1.56%
	SMO	100.00 ± 0.00%	100.00 ± 0.00%
	C4.5	100.00 ± 0.00%	100.00 ± 0.00%
Crx data set	LMT	85.22 ± 4.04%	85.22 ± 4.04%
	Naive Bayes	84.06 ± 1.33%	85.51 ± 4.25%
	SMO	85.80 ± 3.71%	85.80 ± 3.71%
	C4.5	85.36 ± 4.12%	85.51 ± 4.25%
Monk-1 data set	LMT	100.00 ± 0.00%	100.00 ± 0.00%
	Naive Bayes	74.97 ± 1.95%	74.97 ± 1.95%
	SMO	75.02 ± 5.66%	75.02 ± 5.66%
	C4.5	100.00 ± 0.00%	100.00 ± 0.00%
Monk-2 data set	LMT	67.36 ± 1.17%	67.36 ± 1.17%
	Naive Bayes	66.22 ± 2.80%	67.14 ± 0.61%
	SMO	67.14 ± 0.61%	67.14 ± 0.61%
	C4.5	67.14 ± 0.61%	67.14 ± 0.61%
Monk-3 data set	LMT	99.77 ± 0.10%	99.77 ± 0.10%
	Naive Bayes	97.22 ± 0.47%	97.21 ± 2.71%
	SMO	100.00 ± 0.00%	100.00 ± 0.00%
	C4.5	100.00 ± 0.00%	100.00 ± 0.00%

We have implemented our feature selection algorithm by Matlab. We use weka to evaluate the mapped datasets into the selected features obtained by our feature selection algorithm. We compare the feature subsets selected by our method with those selected by Shie-and-Chen's method in table (8) for all of datasets which are used to compare in (Shie, 2007).

Table 8: A comparison of feature subsets selected by our algorithm and Shie-and-Chen's method.

Data sets	Feature subsets selected by two methods	
	Shie-and-Chen's method	Our method
Iris	{4,3}	{4,3}
Breast cancer data set	{6, 2, 1, 8, 5, 3}	{6, 2, 3, 1, 9, 5}
Pima	{2, 6, 8, 7}	{2, 4, 6, 3}
MPG data set	{4, 6, 3}	{2, 4, 1}
Cleve data set	{13, 3, 12, 11, 1, 10, 2, 5, 6}	{13, 1, 12, 3, 9}
Crx data set	{9}	{9}
Monk-1 data set	{5, 1, 2}	{5, 1, 2}
Monk-2 data set	{5}	{5}
Monk-3 data set	{5, 2, 4}	{2,5,1}

Also Table 9 shows that the obtained accuracies of different classifiers on the selected features obtained by proposed method are better that the obtained accuracies of the same classifiers on the selected features obtained by Shie-and-Chen's algorithms the most datasets.

Table 9: A comparison between the average classification accuracy rates of our algorithm and Shie-and-Chen's method

Data sets	Classifiers	Average classification accuracy rates of different methods	
		Our method	Shie-and-Chen's method
Pima diabetes data set	LMT	76.30 ±4.84%	77.22 ± 4.52%
	Naive Bayes	76.30 ±4.84%	77.47 ± 4.93%
	SMO	75.65 ±5.61%	77.08 ± 5.06%
	C4.5	94.62 ±2.12%	74.88 ± 5.89%
Cleve data set	LMT	82.42 ± 5.34%	82.87 ± 6.23%
	Naive Bayes	80.41 ± 3.95%	84.48 ± 3.93%
	SMO	80.00 ± 5.99%	83.51 ± 6.09%
	C4.5	76.90 ± 8.40%	76.90 ± 8.40%
Correlated data set	LMT	100.00 ± 0.00%	100.00 ± 0.00%
	Naive Bayes	86.03 ± 3.75%	86.03 ± 3.75%
	SMO	89.87 ± 6.88%	89.87 ± 6.88%
	C4.5	94.62 ± 4.54%	94.62 ± 4.54%
M of N-3-7-10 data set	LMT	100.00 ± 0.00%	100.00 ± 0.00%
	Naive Bayes	89.33 ± 1.56%	89.33 ± 1.56%
	SMO	100.00 ± 0.00%	100.00 ± 0.00%
	C4.5	100.00 ± 0.00%	100.00 ± 0.00%
Crx data set	LMT	86.53 ±3.87%	85.22 ± 4.04%
	Naive Bayes	86.53 ±3.87%	85.51 ± 4.25%
	SMO	86.53 ±3.87%	85.80 ± 3.71%
	C4.5	85.36 ± 4.12%	85.51 ± 4.25%
Monk-1 data set	LMT	100 ± 0.00%	100.00 ± 0.00%
	Naive Bayes	72.22 ± 6.33%	74.97 ± 1.95%
	SMO	72.22 ± 6.33%	75.02 ± 5.66%
	C4.5	100.00 ± 0.00%	100.00 ± 0.00%
Monk-2 data set	LMT	67.14 ± 0.61%	67.36 ± 1.17%
	Naive Bayes	67.14 ± 0.61%	67.14 ± 0.61%
	SMO	67.14 ± 0.61%	67.14 ± 0.61%
	C4.5	67.14± 0.61 %	67.14 ± 0.61%
Monk-3 data set	LMT	97.22 ± 0.47%	99.77 ± 0.10%
	Naive Bayes	97.21 ± 2.71%	97.21 ± 2.71%
	SMO	97.22 ± 0.47%	100.00 ± 0.00%
	C4.5	100.00 ± 0.00%	100.00 ± 0.00%

4. Conclusion:

In this paper we improved one of the existing feature selection algorithms, Shie-and-Chen's method. This feature selection algorithm uses fuzzy entropy concept. The problem of Shie-and-Chen's method is that it is a parameter sensitive algorithm. User should select threshold values in that algorithm experimentally. The result of algorithm for some threshold values is very weak and it is not trustable. To solve this problem we use ensemble method. Our paper runs Shie-and-Chen's algorithm for different values as thresholds and then gives a weight to each selected features according its rank. Finally by using one of the ensemble methods, majority voting, it selects the best features which have the highest scores. So this algorithm does not need any input parameter. Also the obtained accuracies of different classifiers on the selected features obtained by proposed method are

better that the obtained accuracies of the same classifiers on the selected features obtained by Shie-and-Chen's algorithms.

REFERENCES

Baim, P.W., 1988. A method for attribute selection in inductive learning systems. IEEE Trans Pattern Anal Mach Intell., 10(6): 888-896.

Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Trans Neural Netw., 5(4): 537–550.

Caruana, R. and D. Freitag, 1994. Greedy attribute selection. In: Proceedings of international conference on machine learning, New Brunswick, NJ, pp: 28-33.

Chaikla, N. and Y. Qi, 1999. Genetic algorithms in feature selection. In: Proceedings of the 1999 IEEE international conference on systems, man, and cybernetics, Tokyo, Japan, 5: 538-540.

De, R.K., J. Basak and S.K. Pal, 1999. Neuro-fuzzy feature evaluation with theoretical analysis. Neural Netw., 12(10): 1429–1455.

De, R.K., N.R. Pal and S.K. Pal, 1997. Feature analysis: neural network and fuzzy set theoretic approaches. Pattern Recognit, 30(10): 1579-1590.

Dong, M. and R. Kothari, 2003. Feature subset selection using a new definition of classifiability. Pattern Recognit Lett., 24(9): 1215–1225.

Platt, J.C., 1999. Using analytic QP and sparseness to speed training of support vector machines. In: Proceedings of the thirteenth annual conference on neural information processing systems, Denver, CO, pp: 557-563.

Shie, J.D. and S.M. Chen, 2007. Feature subset selection based on fuzzy entropy measures for handling classification problems, Springer Science+Business Media.

Tan, P.N., M. Steinbach and V. Kumar, 2005. Introduction to Data Mining, first ed. Addison-Wesley Longman Publishing Co. Inc.

Tsang, E.C.C., D.S. Yeung and X.Z. Wang, 2003. OFFSS: optimal fuzzyvalued feature subset selection. IEEE Trans Fuzzy Syst., 11(2): 202-213.