Fuzzy-rough Classifier Ensemble Selection

Ren Diao and Qiang Shen
Department of Computer Science
Aberystwyth University
Aberystwyth, SY23 3DB, UK
Email: {rrd09, qqs}@aber.ac.uk

***Abstract*—Classifier ensembles constitute one of the main research directions in machine learning and data mining. Ensembles allow higher accuracy to be achieved which is otherwise often not achievable with a single classifier. A number of approaches have been adopted for constructing classifier ensembles and aggregate ensemble decisions. In most cases, these constructed ensembles contain redundant members that, if removed, may further increase ensemble diversity and produce better results. Smaller ensembles also relax the memory and storage requirements of an ensemble system, reducing its run-time overhead while improving overall efficiency. In this paper, a new approach to classifier ensemble selection based on fuzzy-rough feature selection and harmony search is proposed. By transforming the ensemble predictions into training samples, classifiers are treated as features. Harmony search is then used to select a minimal subset of such artificial features that maximises the fuzzy-rough dependency measure. The resulting technique is compared against the original ensemble and ensembles formed using random selection, under both single algorithm and mixed classifier ensemble environments.**

Keywords: Classifier Ensemble Selection; Feature Selection; Harmony Search; Fuzzy-rough Sets

I. Introduction

The main purpose of a classifier ensemble is to improve the performance of single classifier systems. Different classifiers usually make different predictions on certain samples, caused by their diverse internal models. Combining such classifiers became the natural way of trying to increase classification accuracy, by exploiting their uncorrelated errors. Also, each ensemble member can potentially be trained using a subset of training samples, which may reduce the computational complexity issue which arises when a single classification algorithm is applied to very large datasets. Additionally, an ensemble can operate in a distributed environment, where datasets are physically separated and are cost ineffective or technically difficult to combine into one database, in order to train a single classifier. A typical approach to building classifier ensembles involves building a group of classifiers with diverse training backgrounds, before aggregating their decisions together to produce the final prediction.

The target of Classifier Ensemble Selection (CES) [21] is to select a subset from a pre-constructed pool of base classifiers, in order to form a reduced group that is still capable in producing good classification results. This is an intermediate stage between the ensemble building and aggregation. Efficiency is one of the obvious gains from CES. Having a reduced number of classifiers can eliminate a portion of run-time overheads, making the ensemble processing quicker; having fewer models also means relaxed memory and storage requirements. Removing redundant ensemble members can also lead to improved diversity within the group, further increase potential ensemble prediction performance. A number of existing techniques use clustering [9] to discover groups of models that share similar predictions, and subsequently prune each cluster separately. Alternative selection methods [22] focus on selecting potentially optimal subsets to maximise a pre-defined diversity measures.

Feature selection [4] (FS) aims to determine a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original features. In dealing with real-world problems FS is usually necessary due to the abundance of noisy, irrelevant, redundant or misleading features [13]. For instance, by removing these factors, learning from data techniques such as text processing and web content classification can benefit greatly. Given a feature set size n, the task of FS can be seen as a search for an "optimal" feature subset through the competing 2^n candidate subsets. The definition of an optimal subset may vary, depending on the problem at hand. A data reduction approach for fuzzy-rough feature selection (FRFS), based on fuzzy-rough sets [7], has been developed [14]. Fuzzy-rough sets encapsulate the related but distinct concepts of vagueness (for fuzzy sets) and indiscernibility (for rough sets [19]),both of which occur as a result of uncertainty in knowledge. The fuzzy-rough set-based techniques consider the extent to which fuzzified values are similar. By employing fuzzy-rough sets, it is possible to perform better feature selection. An unsupervised FRFS method was also proposed [17].

Many existing FS algorithms employ heuristic search strategies in an attempt to avoid the prohibitive complexity of exhaustive method. For instance, FRFS uses an incremental hill-climbing algorithm to discover the best feature subset. However, this often led to the discovery of non-optimal feature subsets, both in terms of the resulting subset quality and the subset size. Other search mechanisms may help, harmony search (HS) [8] in particular, is a meta-heuristic algorithm that mimics the improvisation process of music players. It imposes only limited mathematical requirements and is insensitive to initial value settings. Due to its simplistic structure and powerful performance, HS algorithm has been very successful in a wide variety of optimisation problems [8], presenting several advantages with respect to traditional optimisation techniques. The basic HS algorithm has been improved by

introducing methods to tune parameters dynamically [6] and also successfully applied to solve FS problems [5].

In this paper, a new CES approach based on FRFS is proposed. Inspired by the analogies in between CES and FS, fuzzy-rough CES aims to tackle CES problems from a different angle: by transforming the classifier predictions, each classifier is treated as an artificial feature in the transformed dataset, and classifier predictions as feature values. FS algorithms can then be used to remove redundant features (classifiers) in the present context, in order to select a minimal classifier subset while maintaining original ensemble diversity, and preserving ensemble prediction accuracy.

The rest of this paper is structured as follows. Section II describes the theory behind FRFS, and a most recently proposed unsupervised FRFS method which will be utilised in this work. Section III explains the basic structure of HS, and how it is applied to FRFS. Section IV introduces the key concepts of fuzzy-rough CES, demonstrates how CES can be modelled as an FS problem, and details the approaches developed to tackle the problem. Section V presents the experimentation results along with discussions. Section VI concludes the paper and proposes further work in the area.

II. Fuzzy-Rough Feature Selection

Rough set theory (RST) has been successfully used as an attribute selection tool to discover data dependencies and reduce the number of attributes contained in a dataset by purely structural means [20]. Given a dataset with discretised attribute values, RST can find a subset (termed *reduct*) of the original attributes that are the most informative; all other attributes can be removed from the dataset with minimal information loss. However, it is most often the case that the values of attributes may be both crisp and real-valued, and this is where traditional rough set theory encounters a problem. It is not possible in the theory to say whether two different attribute values are similar and to what extent they are the same. For example, two close values may only differ as a result of noise, but in the standard RST-based approach they are considered to be as different as two values of a different order of magnitude. Dataset discretisation must therefore take place before reduction methods based on crisp rough sets can be applied. This is often still inadequate, however, as the degrees of membership of values to discretised values are not considered and thus can result in information loss. In order to combat this, extensions of rough sets based on fuzzy-rough sets [7] have been developed. A fuzzy-rough set is defined by two fuzzy sets, fuzzy lower and upper approximations, obtained by extending the corresponding crisp rough set notions. In the crisp case, elements either belong to the lower approximationwith absolute certainty or not. In the fuzzy-rough case, elements may have a membership in the range [0,1], allowing greater flexibility in handling uncertainty.

A. Supervised Fuzzy-rough Feature Selection

Fuzzy-rough feature selection [14] (FRFS) is concerned with the reduction of information or decision systems through the use of fuzzy-rough sets. Let $I = (\mathbb{U}, \mathbb{A})$ be an information system, where $\mathbb{U}$ is a non-empty set of finite objects (the universe) and $\mathbb{A}$ is a non-empty finite set of attributes such that $a : \mathbb{U} \rightarrow V_a$ for every $a \in \mathbb{A}$. V_a is the set of values that attribute a may take. For decision systems, $\mathbb{A} = \{\mathbb{C} \cup \mathbb{D}\}$ where $\mathbb{C}$ is the set of input features and $\mathbb{D}$ is the set of decision features.

$$\mu_{\underline{R_P}X}(x) = \inf_{y \in \mathbb{U}} I(\mu_{R_P}(x,y), \mu_X(y)) \tag{1}$$

$$\mu_{\overline{R_P}X}(x) = \sup_{y \in \mathbb{U}} T(\mu_{R_P}(x,y), \mu_X(y)) \tag{2}$$

Here, I is a fuzzy implicator and T is a t-norm. R_P is the fuzzy similarity relation induced by the subset of features P,

$$\mu_{R_P}(x,y) = T_{a \in P}\{\mu_{R_a}(x,y)\} \tag{3}$$

where $\mu_{R_a}(x,y)$ is the degree to which objects x and y are similar for feature a.

FRQuickReduct($\mathbb{C}$,$\mathbb{D}$).
$\mathbb{C}$, the set of all conditional features;
$\mathbb{D}$, the set of decision features.

(1) $R \leftarrow \{\}$, $\gamma'_{best} \leftarrow 0$, $\gamma'_{prev} \leftarrow 0$
(2) **do**
(3) $T \leftarrow R$
(4) $\gamma'_{prev} \leftarrow \gamma'_{best}$
(5) $\forall x \in (\mathbb{C} - R)$
(6) **if** $\gamma'_{R \cup \{x\}}(\mathbb{D}) > \gamma'_T(\mathbb{D})$
(7) $T \leftarrow R \cup \{x\}$
(8) $\gamma'_{best} \leftarrow \gamma'_T(\mathbb{D})$
(9) $R \leftarrow T$
(10) **until** $\gamma'_{best} == \gamma'_{prev}$
(11) **return** R

Fig. 1. The fuzzy-rough QuickReduct algorithm

A fuzzy-rough QuickReduct algorithm, based on the crisp version [20], has been developed as given in Fig. 1. It employs a quality measure termed the fuzzy-rough dependency function γ' to choose which features to add to the current reduct candidate, which is defined by:

$$\gamma'_{P(Q)} = \frac{\sum_{x \in \mathbb{U}} \mu_{POS_{R_P}(Q)}(x)}{\|\mathbb{U}\|} \tag{4}$$

where the fuzzy positive region is defined as:

$$\mu_{POS_{R_P}(Q)}(x) = \sup_{q \in \mathbb{Q}} \mu_{\underline{R_P}Q}(x) \tag{5}$$

The algorithm terminates when the addition of any remaining feature does not increase the dependency. As with the original algorithm, for a dimensionality of n, the worst case dataset will result in $(n^2 + n)/2$ evaluations of the dependency function. However, as fuzzy-rough set based feature selection is used for dimensionality reduction prior to any involvement of a given

application which will exploit those features belonging to the resultant reduct, this operation has no negative impact upon the run-time efficiency of the system.

B. Unsupervised Fuzzy-rough Feature Selection

The central idea behind unsupervised FRFS (U-FRFS) [17] is that, as with the supervised FRFS, the fuzzy dependency measure can also be used to discover the inter-dependency of features. This can be achieved by simply substituting the decision feature(s) $\mathbb{D}$ of the supervised approach for any given feature or group of features Q. In this case, the fuzzy positive region defined in Eq 5 becomes:

$$\mu_{POS_{R_P}(Q)}(x) = \sup_{z \in \mathbb{U}} \mu_{\underline{R_P} R_Q z}(x) \tag{6}$$

where $R_Q z$ indicates the fuzzy tolerance class (or fuzzy equivalence class) for object z, and the lower approximation now becomes:

$$\mu_{\underline{R_P} R_Q z}(x) = \inf_{y \in \mathbb{U}} I(\mu_{R_P}(x, y), \mu_{R_Q}(y, z)) \tag{7}$$

III. Harmony Search for Fuzzy-rough Feature Selection

Harmony Search (HS) [8] mimics the improvisation process of musicians, during which, each musician plays a note for finding a best harmony all together. The basic concepts of HS and application of such concepts in performing optimisation are outlined below, together with an introduction to the dynamic parameter control involved in HS.

A. Key Concepts

The key concepts of HS are musicians, notes, harmonies and harmony memory. In most optimisation problems solvable by HS, the musicians are the decision variables of a certain function being optimised. The notes played by the musicians are the values each decision variable can take. The harmony contains the notes played by all musicians, or an emerging solution vector containing the values for each decision attribute. The harmony memory contains harmonies played by the musicians, or a storage place for potential solution vectors. A more concrete representation of harmony memory is a two dimensional matrix, where the rows contain harmonies (solution vectors) with the number of rows being predefined and bounded by the harmony memory size. Each column is dedicated to one musician, and the entire column stores all the notes played by the musician in all saved harmonies, referred to as the working note domain for each musician in this paper.

Harmony Search for Feature Selection (HSFS) [5] treats musicians as independent experts, and each musician can vote for one feature to be included in the feature subset when improvising a new harmony. The harmony is then the combined vote from all musicians, indicating which features are being nominated. The entire pool of original features forms the range of notes available to the musicians. Multiple musicians are allowed to choose the same attribute, and they may opt to choose no attribute at all. For example, the harmony {A,-,B,B,C,-} will translate into feature subset {A,B,C}, − here represents a null note.

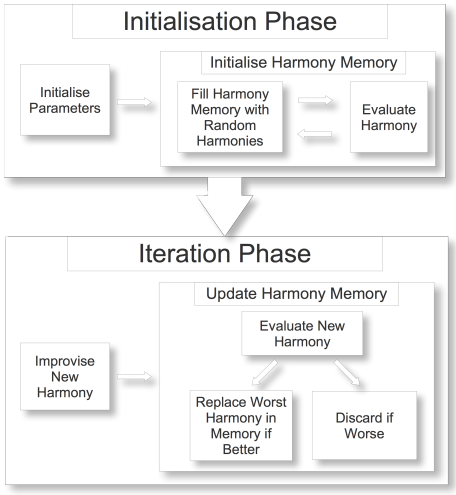

Fig. 2. Illustration of Harmony Search

B. Iteration Steps

HS can be divided into two core phases, initialisation and iteration, as illustrated in Fig 2.

a) Initialise Problem Domain: The parameters of HS are assigned according to the problem, including: size of harmony memory, number of musicians, max iteration, and the harmony memory considering rate (HMCR). The harmony memory of size m is initialised by random generation. This provides each musician a working note domain of m values, which may include identical notes, and nulls. A new harmony is produced by each musician randomly choosing one attribute from their note domain. The new harmony is then evaluated using the given cost function. It is used to replace the worst harmony in the harmony memory if a better score is achieved, or discarded otherwise.

b) Improvise New Harmony: A new value is chosen randomly by each musician out of their note domain, and together forms a new harmony. There are two factors which affect the note choice of a musician, HMCR and PAR. HMCR, ranging from 0 to 1, is the rate of choosing one value from the historical notes stored in the harmony memory, with $(1 - HMCR)$ set to be the rate of randomly selecting one value from the range of all possible notes of the corresponding variable. If HMCR is set low, the musicians will constantly explore other areas of the solution space, and a higher HMCR will restrict the musicians to historical choices.

The PAR parameter causes the musicians to select a neighbouring value based on the following formula $a + (random * bw)$, where bw is an arbitrary distance bandwidth, with $(1 - PAR)$ set to be the probability of using the chosen value without further alteration. The pitch adjustment is applied after a note is chosen by the musician, either from the HM or from the domain of all possible values. For FS problems that are dealt with in this paper, the PAR parameter is not used. This is because the underlying assumption of using PAR is that values that are very close by may provide more optimal solutions. However, when such values represent feature

indices, each index, and its neighbouring features, have no such general dependency in between.

c) Update Harmony Memory: If the new harmony is better than the worst harmony in the harmony memory, judged by the objective function, the new harmony is then included in harmony memory and the existing worst harmony is removed. The algorithm continues to iterate until the maximum number of iterations has been reached.

C. Parameter Control

TABLE I
PARAMETER SETTINGS IN DIFFERENT SEARCH STAGES

	Initialisation	Intermediate	Termination
HMCR	Small	Medium	Large
MS	Small	Medium	Large
Effect	High diversity. Deep Exploration	Steady improvement in harmonies	Fine tuning. Fast convergence

To improve HS and eliminate the drawbacks lying with the use of fixed parameter values, a dynamic parameter adjustment scheme [6] was proposed to modify parameter values at run time. Parameters are dynamically changed, and different settings are provided for different search stages. Table I shows the parameters relevant to FS. These parameters then gradually change through the initial solution space exploration, intermediate solution refinement, and fine tune optimal solution towards termination.

IV. FUZZY-ROUGH CLASSIFIER ENSEMBLE SELECTION

The overall approach developed in this work can be summarised in Fig 3, with each of the four key components described below.

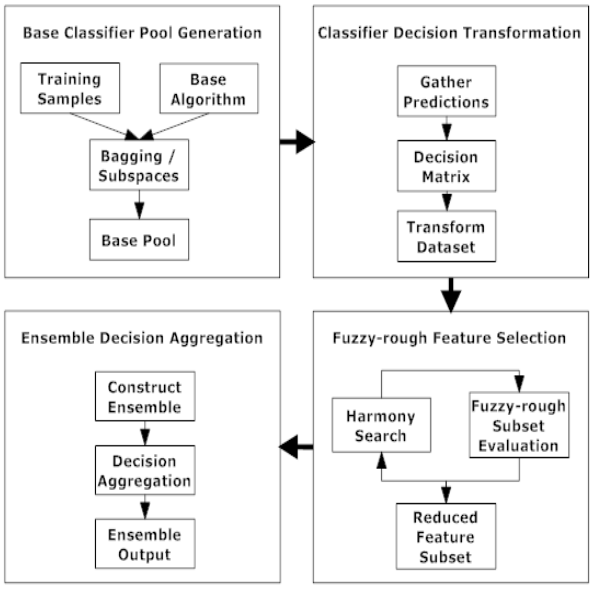

Fig. 3. Fuzzy-rough CES Flow Chart

A. Base Classifier Pool Generation

Forming a diverse base classifier pool (BCP) is the first step in producing a good classifier ensemble. Any preferred methods can be used to build the base classifiers, *Bagging* [2] and *Random Subspaces* [11] methods are adapted here. BCP can be created using a single classification algorithm as well as a mixed classifier scheme. *Bagging* randomly selects different subsets of training samples in order to build diverse classifiers. Differences in the training data present extra or missing information for different classifiers, resulting in different models. The *Random Subspaces* method randomly generates different subsets of domain attributes and builds various classifiers on top of each of such subsets. The differences between the subsets creates different view points of the same problem [3], typically resulting in different borders for classification. For a single base classification algorithm, these two methods both provide good diversities. In addition, a mixed classifier scheme is implemented in the presented work. By selecting classifiers from different schools of classification algorithms, the diversity is naturally achieved through the various foundations of the algorithms themselves.

B. Classifier Decision Transformation

TABLE II
DECISION MATRIX

	C_1	C_2	$\cdots$	C_i	$\cdots$	C_{N_C}
I_1	D_{1j}	D_{21}	$\cdots$	D_{1j}	$\cdots$	D_{N_C1}
I_2	D_{12}	D_{22}	$\cdots$	D_{2j}	$\cdots$	D_{N_C2}
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
I_j	D_{1j}	D_{2j}	$\cdots$	D_{ij}	$\cdots$	D_{N_Cj}
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
I_{N_I}	D_{1N_I}	D_{2N_I}	$\cdots$	D_{iN_I}	$\cdots$	$D_{N_CN_I}$

Once the base classifiers are built, their decisions on the training instances are also gathered. For base classifiers $C_i, i = 1, 2, \ldots, N_C$, and training instances $I_j, j = 1, 2, \ldots, N_I$, where N_C is the total number of base classifiers, and N_I is the total number of training instances, a decision matrix as shown in Table II can be constructed. The value D_{ij} represents the ith classifier's decision on the jth instance. For supervised FS, a class label is required for each training sample, the same class attribute is taken from the original dataset, and assigned to each of the instances. Note that both the total number of instances and the relations between instances and their class labels remain unchanged. Although all attributes and values are completely replaced by transformed classifier predictions, the original class labels remain the same. A new dataset is therefore constructed, each column represents an artificially generated feature, each row corresponds to a training instance, the cell then stores the transformed feature value.

C. Feature Selection on the Transformed Dataset

FRFS is then performed on the transformed dataset, evaluating each artificial feature subset using the fuzzy-rough dependency measure. HS optimises the quality of

discovered subsets, while trying to reduce subset sizes. A detailed explanation is already given in Section III. When HS terminates, its best harmony is translated into a feature subset and returned as the FS result. The features then indicate their corresponding classifiers which should be included in the learnt classifier ensemble. For example, if the best harmony given by HS is $\{-, C_9, C_3, C_{23}, C_3, C_5, C_{17}, -\}$, the translated artificial feature subset is then $\{C_3, C_5, C_9, C_{17}, C_{23}\}$. Thus, the 3rd, 5th, 9th, 17th and 23rd classifiers will be chosen from the BCP to construct the classifier ensemble. Note that the application of unsupervised FRFS is the same as above and hence is omitted to avoid repetition.

D. Ensemble Decision Aggregation

Once the classifier ensemble is constructed, new objects are classified by the ensemble members, and their results are aggregated to form the final ensemble decision output. The *Average of Probability* aggregation method is used in the paper. Given ensemble members $E_i, i = 1, 2, \ldots, N_E$, and decision classes $D_j, j = 1, 2, \ldots, N_D$, where$N_D$ is the ensemble size and N_D is the number of decision classes, classifier decisions can be viewed as a matrix of probability distributions $\{P_{ij}\}$. Here, P_{ij} indicates prediction from classifier C_i for decision class D_j. The final aggregated decision is the winning classifier that has the highest averaged prediction across all classifiers: $\{\sum_{i=1}^{N_E} P_{i1}/N_E, \sum_{i=1}^{N_E} P_{i2}/N_E, \ldots, \sum_{i=1}^{N_E} P_{iN_D}/N_E\}$. Note that this is effective because redundant classifiers are now removed. As such, the usual alternative aggregation method: *Majority Vote* is no longer favourable since the "majority" has been significantly reduced.

V. Experimentation and Discussion

To demonstrate the capability of fuzzy-rough CES, a number of experimentations have been carried out. These experiments are divided into three groups, based on the main methods adapted to generate the BCP: *Bagging*, *Random Subspaces*, and the mixed classifier scheme. For single base algorithm experiments, the facilitate comparative studies, the state of the art fuzzy-rough nearest neighbour (FRNN) [12] classifier is used. Both ordinary FRFS and unsupervised FRFS methods are examined. Classification accuracies are collected under stratified 10 fold cross validation, and each set of experiments is run 10 times with results averaged. Ten 10-fold cross validations are used as random factors are involved in creating the *Bagging* and *Random Subspaces* BCPs, as well as in the artificial feature selection process using HS. For comparison, the accuracy of the base algorithm, when trained using the original dataset without alteration is included, the averaged accuracy of the full BCP is also included. A total of 9 real-valued UCI datasets [18] are used in the experiments. Parameters used in the experiments and dataset information are summarised in Table III.

TABLE III
HS Parameter Settings and Dataset Information

Harmony Memory Size	# Musicians	HMCR	Max Iteration
10-20	$\frac{2}{3}$ # features	0.5-1	500
Dataset	**Objects**	**Attributes**	**# Classes**
cleveland	297	14	5
ecoli	336	8	8
glass	214	9	6
heart	270	13	2
ionosphere	230	35	2
sonar	208	60	2
water 3	390	38	3
water 2	390	38	2
wine	178	14	3

A. Bagging

In this set of experiments, the BCP is built using *Bagging* with FRNN being the base algorithm. Table IV shows the results of applying fuzzy-rough CES and unsupervised fuzzy-rough CES, as compared against the results of using: (1) the base algorithm, (2) the full base classifier pool, and (3) randomly formed ensembles. Entries annotated with *v* indicates that the selected ensemble performance is statistically significant when compared against random pick, using paired t test with two-tailed P threshold = 0.005.

TABLE IV
Fuzzy-rough Nearest Neighbour using Bagging

Dataset	Base	Full	FRFS	U-FRFS	Random
cleveland	53.20	57.91	57.21 (11.8) *v*	56.30 (8.9)	56.03
ecoli	81.85	86.01	85.97 (11.9)	85.52 (7.9)	85.52
glass	72.43	73.36	73.18 (11.3) *v*	73.04 (8.1)	73.04
heart	76.30	81.19	79.78 (8.0) *v*	79.22 (8.2)	78.69
ionosphere	90.43	88.43	88.70 (7.4) *v*	88.83 (7.5) *v*	87.85
sonar	85.10	85.72	85.96 (8.0)	85.91 (7.9)	85.32
water 3	80.26	81.51	81.38 (8.8)	81.62 (7.9) *v*	80.94
water 2	84.62	85.23	85.05 (7.9) *v*	84.69 (7.8)	84.35
wine	97.19	96.91	96.46 (7.6)	96.63 (7.8) *v*	96.35

The results show that the *Bagging* method improved over the base classifier. Column *Full* displays aggregated predictions from the entire base classifier pool of size 50. For 7 out of 9 datasets, greater classification accuracies have been achieved using the ensemble approach. The CES+FRFS method successfully selected classifier ensembles with reduced size, as shown in column *FRFS* with the numbers in the brackets indicating the selected ensemble sizes. Because ten fold cross validation is used, different classifiers are constructed and selected for each individual fold, hence performed in general differently, but the reported sizes are an average of 100 ensembles. Despite the reduced ensemble size, classification accuracies are further improved for the *sonar* and *ionosphere* datasets. For the other datasets, the accuracies are reduced very marginally, with less than 0.2% in 5 of the 7 total cases, and the greatest reduction being 0.7% (*cleveland*).

When compared against the use of supervised FRFS, CES equipped with unsupervised FRFS finds more compact ensembles with an averaged size of 8. This is possibly due to the fact that the class label is no longer considered in the dependency calculation, and therefore less consistency constraints are placed upon the construction of ensembles

(artificial feature subsets). The reduced ensembles still maintain reasonable classification accuracies, in comparison to the use of supervised FRFS, and the base pool. Both approaches generally deliver better results than randomly picked ensembles, except for *ecoli* and *glass* datasets, the unsupervised method seems to have an equal performance as random picking. One possible explanation is that the number of classes for a given training dataset has direct impact upon unsupervised FRFS performance. The more classes there are, the more various the classifier predictions become, thereby providing more available values for the artificial features. The transformed dataset is therefore more complex than ones originated from the training datasets with 2 or 3 classes. For such datasets like *cleveland* (5), *ecoli* (8), and *glass* (6), the extra consistency constraint from the class label itself must have aided the supervised method in selecting better artificial feature subsets, with the sacrifice of having larger ensembles.

B. Random Subspaces

TABLE V
FUZZY-ROUGH NEAREST NEIGHBOUR USING RANDOM SUBSPACES

Dataset	Base	Full	FRFS	U-FRFS	Random
cleveland	53.20	58.25	57.00 (8.1)	57.78 (7.4) *v*	56.37
ecoli	81.85	80.33	81.85 (9.8) *v*	78.63 (7.8)	77.98
glass	72.43	76.96	76.78 (7.6) *v*	76.87 (7.6) *v*	75.05
heart	76.30	82.59	80.89 (7.7)	79.96 (7.6)	80.28
ionosphere	90.43	89.87	89.87 (7.8) *v*	89.83 (7.7) *v*	89.48
sonar	85.10	88.89	88.22 (7.1) *v*	87.88 (7.4) *v*	87.24
water 3	80.26	80.97	81.90 (7.7) *v*	81.59 (7.8) *v*	80.63
water 2	84.62	85.46	85.15 (7.6) *v*	85.00 (7.6)	84.83
wine	97.19	97.81	97.02 (7.5) *v*	97.02 (7.8)	96.41

Table V shows the use of *Random Subspaces* builds slightly better quality base classifiers. In comparison with the base algorithm, the pool gains better accuracy in 8 out of 9 cases, except for dataset *ionoshpere*; and in 7 out of 9 cases, the averaged accuracies are higher than pools using *Bagging*. The FRFS method successfully reduced ensembles, and the ensemble sizes are smaller than previous discoveries for all datasets. A likely indication from these results is that the classifiers created using *Random Subspaces* contain more redundancy. Although the overall accuracies of the reduced ensembles are higher than those obtained by their counterpart via using the *Bagging* method, greater accuracy decrease is observed when compared with the base pools.

Unsupervised FRFS also finds smaller ensembles, with comparable accuracies except for *ecoli*, the reduced ensembles suffered more than 3% loss of accuracy when compared against the case when supervised FRFS is used. Both selection methods however entail better results than random selection.

C. Mixed Classifier Scheme

A total of 10 different base algorithms are selected for this experiment, one or two distinctive classifiers from each representative classifier groups, including fuzzy-based FuzzyNN [16], FuzzyRoughNN [12], VQNN [12], lazy-based IBk [1], tree-based J48 [23], REPTree [23], rule-based JRip [23], PART [23], NaiveBayes [15] and MultilayerPerceptron [10]. Given only 10 base algorithms, more variations are needed to produce as pool of size 50, *Bagging* and *Random Subspaces* are again used to create differentiation between classifiers. Tables VI and VII show the experimental results, using these two methods respectively.

For mixed classifiers created using *Bagging*, the FRFS method finds ensembles with much greater size variation and overall, considerably larger ensembles than previous experiments. For the *ecoli* dataset in particular, the averaged ensemble size is 15.98. Even with such a large ensemble, the underlying artificial feature subset still did not achieve 1 for the fuzzy-rough dependency measure. However, the results indicate that many distinctive features (i.e. classifiers) are present, therefore a larger subset is necessary to maintain consistency. For classifier ensembles, this means good diversity and many distinctive classifiers co-exist in the base pool, and thus less redundancy exists. This particular ensemble also results in the highest accuracy for *ecoli* compared against other approaches, with 87.67% BCP accuracy, and 86.66% ensemble accuracy. Although most ensembles can achieve comparable performance, large performance decreases are also noticed for the *sonar* and *heart* datasets. Interestingly, the use of the unsupervised method achieves better overall performance than its supervised counterpart, with smaller selected ensemble sizes. Reasons for such performance is unknown yet, but further investigation into this aspect is currently ongoing.

TABLE VI
MIXED CLASSIFIERS USING BAGGING

Dataset	Full	FRFS	U-FRFS
cleveland	54.94	52.92 (11.2)	54.27 (9.32)
ecoli	87.67	86.66 (15.98)	85.77 (8.7)
glass	71.12	69.62 (12.2)	69.62 (9.2)
heart	82.07	75.40 (8.76)	77.62 (8.42)
ionosphere	87.73	88.17 (8.36)	88.17 (8.56)
sonar	80.96	73.55 (8.5)	80.76 (8.68)
water 3	78.15	78.71 (9.26)	78.20 (8.72)
water 2	80.61	81.58 (8.72)	82.10 (8.84)
wine	98.31	97.52 (7.74)	97.40 (7.46)

TABLE VII
MIXED CLASSIFIERS USING RANDOM SUBSPACES

Dataset	Full	FRFS	U-FRFS
cleveland	56.57	57.85 (11.82)	57.10 (9.08)
ecoli	79.17	84.64 (12.16)	84.40 (7.8)
glass	75.61	71.50 (11.28)	73.08 (8.18)
heart	82.44	80.89 (7.96)	80.44 (8.16)
ionosphere	89.30	87.39 (8.1)	88.00 (7.58)
sonar	82.69	86.06 (7.86)	83.17 (7.88)
water 3	80.26	80.41 (9.16)	80.92 (8.04)
water 2	83.18	85.85 (7.86)	85.74 (7.92)
wine	98.09	97.53 (7.82)	97.75 (7.46)

The *Random Subspaces* based mixed classifier scheme produces better base pools in 7 out of 9 cases. Both FRFS and U-FRFS find smaller ensembles on average than the case where *Bagging* is used. Neither method suffers from extreme performance decreases following reduction unlike the results obtained when single base algorithm is employed. Despite having a BCP that under performs for the *ecoli* dataset, both methods manage to achieve an increase of 5% in accuracy.

However, the general quality of the mixed classifier group is lower than that of the FRNN based single algorithm approach. This is largely caused by the employment of non-optimised base classifiers. It can be expected that the results achievable after optimisation would be even better. Note that certain ensemble methods are less desirable for some datasets. For example, using the mixed classifier scheme can achieve an accuracy of 86.66% for *ecoli*, but using partition based approach only leads to an accuracy of 81.85%. Yet, the partition based approach results in 88.22% for *sonar*, while the mixed classifier scheme obtains a result as low as 73.55%. Further experimental investigation revealed that *ecoli* involves the least number (8) of attributes. Therefore *Random Subspaces*, which creates attribute subsets randomly, can easily remove an essential attribute, making it less suitable for ensemble generation. The massive performance gain for *sonar* dataset can be explained in a similar way: *Random Subspaces* can construct diverse classifiers reasonably well with 60 attributes.

VI. Conclusion

This paper has presented a new approach to classifier ensemble selection. It works by applying fuzzy-rough feature selection technique to minimise redundancy in an artificial dataset generated via transforming a given classifier pool's decision matrix. The aim is to further reduce the size of a classifier ensemble, while maintaining and improving classification accuracy, making the ensemble more efficient. Experimental comparative studies show that both supervised and unsupervised FRFS approaches can entail good solutions. Reduced ensembles are found with comparable classification accuracies as the base pools, and in most cases provide good improvements over the performance achievable by the base algorithms. In particular, the use of the unsupervised FRFS method can help create smaller ensembles, especially when complex mixed classifiers are used.

Although promising, much can be done to further improve the potential of the presented work. In particular, an alternative classifier decision matrix transformation procedure may be formulated. Some classifiers first produce a distribution of the likelihood that a particular instance may belong to the available classes, and the class with highest probability is then taken as the final prediction. These probability distributions may contain more information and are potentially more suitable to be used as the artificial feature values. In addition, other statistical information from the classifiers such as variance, may also be good candidates for use as part of the artificially generated features, in order to create a more comprehensive dataset for feature selection. Experimental evaluation of these ideas remains as active research, as well as the aforementioned investigation into the underlying reasons why the use of unsupervised FRFS helps more significantly overall, in simplifying the complexity of the leant ensembles.

Finally, it is worth noting that, instead of fully relying on FRFS and HS methods, the proposed approach can be generalised to work with other feature selection techniques and heuristic search strategies, making it a generic CES framework for future extensions. Fuzzy-rough CES can also be used in conjunction with other CES methods, in particular those that maximise a certain diversity measure, in order to further reduce ensemble size, whilst preserving ensemble performance.

References

[1] D. Aha and D. Kibler, Instance-based learning algorithms, Machine Learning, vol. 6, no. 1, pp. 37–66, 1991.

[2] L. Breiman, Bagging predictors, Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.

[3] C. Christoudias, R. Urtasun, and T. Darrell, Multi-view learning in the presence of view disagreement, Proceedings of UAI 2008, 2008.

[4] M. Dash and H. Liu, Feature Selection for Classification, Intelligent Data Analysis, vol. 1, no. 3, pp. 131–156, 1997

[5] R. Diao and Q. Shen, Two New Approaches to Feature Selection with Harmony Search, Proceedings of the 19th International Conference on Fuzzy Systems, pp. 3161–3167, 2010.

[6] R. Diao and Q. Shen, Deterministic Parameter Control in Harmony Search, Proceedings of the 2010 UK Workshop on Computational Intelligence, 2010.

[7] D. Dubois and H. Prade, Putting rough sets and fuzzy sets together, Intelligent Decision Support, Kluwer Academic Publishers, Dordrecht, 1992.

[8] Z.W. Geem, J.H. Kim and G.V. Loganathan, A New Heuristic Optimization Algorithm: Harmony Search, Simulation, vol. 76, no. 2, pp. 60–68, 2001.

[9] G. Giacinto, F. Roli, and G. Fumera, Design of effective multiple classifier systems by clustering of classifiers, 15th International Conference on Pattern Recognition, ICPR 2000, pp. 160–163, 2000.

[10] S. Haykin, Neural Networks: A Comprehensive Foundation, Second Edition, Prentice Hall, 1998.

[11] T.K. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832–844, 2002.

[12] R. Jensen and C. Cornelis, A New Approach to Fuzzy-Rough Nearest Neighbour Classification, Transactions on Rough Sets XIII, LNCS 6499, pp. 56-72, 2011.

[13] R. Jensen and Q. Shen, Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches, IEEE Press and Wiley & Sons, 2008.

[14] R. Jensen and Q. Shen, New Approaches to Fuzzy-Rough Feature Selection, IEEE Transactions on Fuzzy Systems, vol. 17, no. 4, pp. 824–838, 2009.

[15] G.H. John and P. Langley, Estimating Continuous Distributions in Bayesian Classifiers, Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345, 1995.

[16] J. Keller, M. Gray, and J. Givens, A fuzzy k-nearest neighbor algorithm, IEEE Transactions on Systems, Man, And Cybernetics, vol. 15, no. 2, pp. 580–588, 1985.

[17] N. Mac Parthalain and R. Jensen, Measures for unsupervised fuzzy-rough feature selection. International Journal of Hybrid Intelligent Systems, vol. 7, pp. 1C11, 2010.

[18] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, UCI Repository of machine learning databases http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[19] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishing, Dordrecht, 1991.

[20] Q. Shen and A. Chouchoulas, A rough-fuzzy approach for generating classification rules, Pattern Recognition, vol. 35, no. 2, pp. 2425–2438, 2002.

[21] G. Tsoumakas, I. Partalas, and I. Vlahavas, A Taxonomy and Short Review of Ensemble Selection, ECAI 2008, Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications, 2008.

[22] A. Tsymbal, M. Pechenizky, and P. Cunningham, Diversity in search strategies for ensemble feature selection, Information Fusion, vol. 6, pp. 83–98, 2005.

[23] I.H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, San Francisco, 2005.