ENSEMBLE CREATION USING FUZZY SIMILARITY MEASURES AND FEATURE SUBSET EVALUATORS

Valerie Cross and Michael Zmuda

Computer Science and Software Engineering,
Miami University, Oxford, OH, USA

ABSTRACT

Current machine learning research is addressing the problem that occurs when the data set includes numerous features but the number of training data is small. Microarray data, for example, typically has a very large number of features, the genes, as compared to the number of training data examples, the patients. An important research problem is to develop techniques to effectively reduce the number of features by selecting the best set of features for use in a machine learning process, referred to as the feature selection problem. Another means of addressing high dimensional data is the use of an ensemble of base classifiers. Ensembles have been shown to improve the predictive performance of a single model by training multiple models and combining their predictions. This paper examines combining an enhancement of the random subspace model of feature selection using fuzzy set similarity measures with different measures of evaluating feature subsets in the construction of an ensemble classifier. Experimental results show that in most cases a fuzzy set similarity measure paired with a feature subset evaluator outperforms the corresponding fuzzy similarity measure by itself and the learning process only needs to occur on typically about half the number of base classifiers since the features subset evaluator eliminates those feature subsets of low quality from use in the ensemble. In general, the fuzzy consistency index is the better performing feature subset evaluator, and inclusion maximum is the better performing fuzzy similarity measure.

KEYWORDS

Feature selection, fuzzy set similarity measures, concordance correlation coefficient, feature subset evaluators, microarray data, ensemble learning.

1. INTRODUCTION

Feature selection (FS) is an important task for classification in machine learning (ML) since it reduces dimensionality with respect to the feature dimension. Its objective is to find a possibly optimal feature subset of relevant features that reduces the data size and increases, or maintains, the overall performance measures such as accuracy and sensitivity on the results of the classification. Reducing the data size decreases data storage requirements and training times for learning algorithms and can improve visualization and interpretation of the learning results. There are three main approaches to feature selection: filter, wrapper and embedded methods. In this research, a filter method is used due to its advantages of typically being fast and not tuned for a given learner [1].

Feature selection methods have typically been performed by evaluating a candidate feature subset and searching through the feature space to find a better subset. Existing algorithms adopt various

David C. Wyld: CSITY, NWCOM, SIGPRO, ASOFT, AIFZ, BDIoT, ITCCMA, CLSB, DTMN, MLNLP - 2021
pp. 73-86, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111407

measures to evaluate the quality of feature subsets [1][2][3]. The random subspace method (RSM) [4] for feature subset selection, however, does not use a search process. Instead, it randomly selects from an arbitrary sized subset of features that are ranked using an algorithm such as ReliefF [5], where ReliefF measures the relevance of a feature to the classification task. RSM techniques can be used to create an ensemble of base classifiers, each created from one of the randomly selected subsets of features [6]. Although such approaches are simple and fast, they do not consider possible correlations and dependencies that may exist between the features in the randomly selected subsets. In [7], ReliefF is used to rank the quality of the features and then the concordance correlation coefficient (CCC) [8] is used to group related features from the N top-ranked features into G disjoint subsets. An RSM-like approach is then used to randomly select a single feature from each of the G feature subsets instead of randomly selecting from all the top-ranked features. This process creates the feature subset for a single base classifier. This process is then repeated to create E base classifiers that are used in the ensemble. Extensions to that work examine the use of fuzzy set similarity measures (FSSM) along with the CCC to create the groups of related features and evaluate the difference in performance of the generated ensembles on four different datasets [9]. The FSSMs are modified to distance measure used in a hierarchical clustering process that creates the G feature subsets.

Some ML processes search through a space of feature subsets to find an optimal feature subset. They use feature subset evaluators to determine the quality of a feature subset. This paper further extends the research in [9] to employ different feature subset evaluators, not in a search process, but instead to assess the quality of each of the randomly generated feature subsets for use in base classifiers. Each feature subset evaluator is paired with a FSSM to determine its performance as compared to the corresponding FSSM alone. The hypothesis is that feature subset evaluators should reduce the number of base classifiers in the ensemble and improve the ensemble performance measures.

This research differs from the methods discussed in [10] which compares approaches used to create an ensemble of feature selectors and combine the produced feature subsets into one feature subset to be used in the machine learning process. In that research an ensemble of feature selectors is categorized as either homogenous or heterogenous. The homogenous selector uses the same feature selection method but on different training data subsets. The heterogenous feature selector ensemble uses a number of different feature selection methods but on the same training data.

Regardless of the type of feature selector ensemble, the resulting subsets of features must be aggregated into one feature subset. There are simple ways to approach this such as using the intersection or the union of the subsets of features [11]; however, these simple approaches may lead to a very restrictive set of features or to less reduction in the size of the set of features. A more sophisticated technique uses classification accuracy to combine the features produced by the various feature selectors [12]. This approach, however, is computationally expensive and may result in computational costs higher than that of the feature selection process.

The research presented here is similar to using homogeneous feature selectors in that the same modified RSM feature selector is used on different training datasets. It differs, however, since it does not combine the resulting subsets of features produced for each training dataset into one feature subset. Instead, each produced feature subset is used in the learning of a base classifier if it is of sufficient quality as determined by the feature subset evaluator. An ensemble is then created from those individual base classifiers learned using the feature subsets of sufficient quality. The ensemble is then applied on the test dataset, and the results of each of its base classifiers are combined using simple majority voting [13].

The paper is organized is as follows: Section II discusses the machine learning system which uses a combination of FSSMs with feature subset evaluators. Section III explains the FSSMs for grouping of similar features. The evaluators used to assess the quality of a feature subset are presented in section IV. This evaluation is over the feature set as a whole and differs from filtering methods applied to individual features. Section V discusses the experimental design and its parameters. Section VI presents the experimental results in terms of several views: 1) individual feature subset evaluators over FSSMs and datasets, 2) individual FSSMs over feature subset evaluators and datasets, 3) ensemble performance across datasets, 4) datasets across ensemble performance measures, and 5) highest ensemble performance values for pairs of feature subset evaluator and FSSM within datasets. Finally, section VII presents conclusions and possible future work.

2. MACHINE LEARNING COMPONENTS

A machine learning system can have different structures and use a variety of methods. Here in this research the structure consists of 1) pre-process filtering, 2) a feature subset selection algorithm, 3) ensemble building algorithm, and 4) a learning algorithm. Figure 1 illustrates the system using four processes, which are described in the following four subsections.

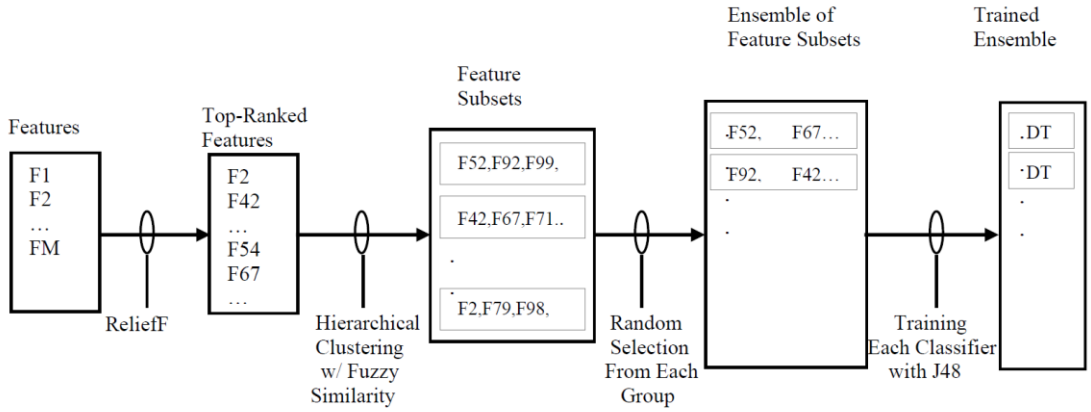

Figure 1. Machine Learning Process without Evaluators

2.1. Pre-processing Filtering

Filtering methods [14], also referred to as feature ranking methods, examine intrinsic properties of datasets to rank the features on their relevance to the classification task. This ranking is independent of the choice of learning algorithms.

ReliefF [5] is a well-known and often-used filtering method and is used to rank the features based on their numerical value. From this ranking, a specified number of top ranked features are selected as input to a feature subset selection algorithm; the others are discarded. Although ReliefF does provide a good way to assess the merit of individual features, it does not assess the merit of a *collection* of features. That is, a good feature set will include high-quality features in addition to having a diverse set of features; ReliefF is not designed to address this issue. The next step is to form subsets of these top-ranked features to use in training individual base classifiers.

2.2. Feature Subset Selection

Feature subset selection using feature subset evaluation produces candidate feature subsets based on a given strategy and can address feature redundancy in addition to feature relevance. A search strategy typically is used to search through feature subsets. Searching is time consuming due to feature subset generation and the evaluation of the feature subset. Methods to evaluate feature subsets are a distinguishing factor among feature selection algorithms using searching.

As done in [9], the top-ranked features undergo a hierarchical clustering algorithm to assign each feature uniquely into one of the G groups. When clustering, *distance* is defined using the specific FSSM used. During the clustering process, the two clusters that are merged are those that have smallest distance between the most distant members of the two individual clusters. When complete, the features that reside in a particular group can be viewed as being like the others in that group as defined by the FSSM, while being relatively dissimilar to the members of the other groups. The degree in which this is true is highly dependent on the underlying data and the value of G used.

This work does not use searching in the feature subset selection process. The RSM method, in combination with the grouping of related features, produces a feature subset by randomly selecting one feature from each of the G groups. In [9], no quality assessment is performed on each generated feature subset; each one is simply used in a base classifier to be trained for use in an ensemble. This current research instead uses different evaluators to assess the quality of a feature subset. If the feature subset's evaluation score is not sufficient, the feature subset is eliminated from use in a base classifier.

2.3. Ensemble Building

An ensemble can be built using a data partition, a feature partition, or hybrid approach [15]. Here, feature partitioning is used. Each feature subset of sufficient quality is associated with a base classifier. Each base classifier must be trained before used in the ensemble. An ensemble with multiple high-quality-only trained base classifiers is expected to have higher performance results and reduce learning times due to the elimination of low-quality feature subsets. The ensemble aggregates the predictions from its set of base classifiers using simple majority voting [13].

2.4. Machine Learning

Various machine learning algorithm with the training data can be used on a base classifier and its feature subset. Weka's J48 decision tree (DT) classifier [16][10] with default parameter settings is used for training the base classifiers. J48 is used for consistency with its use in [9].

3. FUZZY SET SIMILARITY MEASURES

Each feature is represented as a fuzzy set over the instances in the sample data sets. The feature values must be normalized to specify a degree of membership in [0, 1]. Similarity between the fuzzy sets representing each feature is determined using a FSSM. The fuzzy similarity measures are using during the hierarchical clustering process. In [9] the FSSMs used are presented in more detail. For completeness, they are briefly described here.

The *concordance correlation coefficient* (CCC) measures a bivariate relationship in terms of agreement between two values [8]. It differs from the Pearson correlation which measures the

degree of linear relationship. CCC measures the degree to which pairs of values are close to the 45 degrees line of perfect concordance in a scatterplot. This line runs diagonally to the scatterplot. It is a very specific linear relationship, not just any linear relationship. A zero value indicates no agreement.

Zadeh's consistency index, also known as the sup-min or partial matching index [17], roughly estimates the similarity between two fuzzy sets by finding at what domain values they intersect and determines their similarity by taking the highest membership degree among their intersection points.

The *fuzzy Jaccard similarity measure* is a fuzzy extension of the Jaccard index [18] between two crisp sets. It replaces set cardinality with fuzzy set cardinality. It is the ratio between the fuzzy set cardinality of the intersection and that of the union.

A *fuzzy inclusion measure* determines how much one fuzzy set is included in another [17]. Another way to create a FSSM is to use a symmetric aggregation of the two directions of inclusion. The aggregation operators used are *average*, *minimum*, and *maximum*.

The *cosine* measure [19] views each fuzzy set as a vector in *n* dimensional space and computes the cosine of the angle between the two vectors. Because the feature values are values in the range [0, 1], the cosine can never be negative

4. FEATURE SUBSET SELECTION

Many feature selection methods contain two important aspects: evaluation of a candidate feature subset and searching through the feature space. The RSM approach does not use a search process but instead iteratively produces a randomly generated feature subset for a candidate base classifier to use in the ensemble. This current research incorporates the use of evaluation functions, referred to as *evaluators*, on the randomly generated feature subset. In [20], feature subset evaluators are classified into five categories: distance, information (or uncertainty), dependence, consistency**,** and classifier error rate. Evaluators in the classifier error rate category are referred to as wrapper methods [20]. Although wrapper methods produce high accuracy, due to their high computational cost, they are not considered here. The following describes the three evaluators used in this work.

Interclass distance (ICD) [2] in the distance category, also known as separability, divergence, or discrimination, is based on the assumption that instances of a different class should be distant in the instance space. Most often the distance measure *d* is in the Euclidean family:

$$\mathrm{ICD}(+,-) = \frac{1}{N_+ N_-} \sum_{k_1}^{N_+} \sum_{k_2}^{N_-} d\left(x_{(+,k_1)}, x_{(-,k_2)}\right) \quad (1)$$

where, + and - are the two class labels. $x_{(+,\,k1)}$ represents an instance k_1 of class +. $x_{(-,\,k2)}$ respresents an instance k_2 of class -. N_+ is the number of positive instances. N_- is the number of negative instances. This formula is for two classes since the datasets in this study are binary classification problems. It takes the distance between each positive instance with each negative instance and sums over all possible pairs. Then the average over all the distances is taken.

Maximal information compression index (MiCi)[20] appears in both the information and dependence categories. An evaluator in the dependence category computes the dependence of a feature on other features. Its value measures the degree of redundancy of the feature. All evaluators in the dependence category can also be classified as information measures. MiCi

measures the amount of error produced by reducing the pair of features (x, y) to a single feature. The greater the error means the less redundant are the two features. For features x and y, the formula is given as

$$MiCi(x,y) = \frac{1}{2}(var(x) + var(y) - \sqrt{(var(x) + var(y))^2 - 4(var(x) * var(y) * (1 - \rho(x,y)^2))}) \quad (2)$$

var(x) is the variance of the values for feature x and similarly for y. $\rho(x,y)$ is the covariance for the values of features x and y. This measure is performed for every pair of features in the feature subset, and the total is accumulated as a measure of error over the feature subset as a whole.

Fuzzy consistency (FC) is an adaptation of a crisp consistency measure [21] on a feature subset. Consistency for a feature subset F determines how many identical instances have the same class value for each group of identical instances, i.e., measures how consistent is the classification for each set of identical instances using the given F. Since this work only deals with binary classification problems, if there is a pattern A of identical instances, then for pattern A the crisp consistency is the maximum of either the positive class count or the negative class. The consistency formula for an identical pattern A is

$$C_F(A) = max_{k=+,-}[F_k(A)] \quad (3)$$

where $F_k(A)$ is the number of instances in class k equal to A. The consistency rate is given as

$$CR(F) = \frac{\sum_{A \in S} C_F(A)}{|S|} \quad (4)$$

for each pattern A in the set of unique patterns S and |S| is the cardinality of the set S. Consistency measures rely on discrete-valued features where continuous features must first be discretized. To simplify measuring the consistency of the fuzzy instances, clustering is used to group instances into similar groups, i.e., |S| clusters, based on their feature membership values over features. Although the fuzzy instances in a group are not identical, the fuzzy consistency value is calculated for each similar group A as in Eq. 3. Fuzzy consistency rate is calculated as the average over all similar groups as in Eq. 4.

5. EXPERIMENTAL DESIGN AND DATASETS

The research objective is to compare the effects of 1) the different FSSMs used to group features and 2) the evaluators used on the randomly generated features subsets from these groups on the performance when poorer feature subsets are eliminated. The results from the combinations of FSSMs and feature subset evaluators are compared to the results of just using FSSMs in creation of the base classifiers (i.e., without the evaluators). To perform this analysis, a systematic series of machine learning experiments were conducted, with the main control variables being the fuzzy similarity measure and the evaluator. The datasets used in these experiments are: breast, CNS, colon, and leukemia. The details of the data sets can be found in [9].

The objective in the previous research with FSSMs for grouping was to compare how the different similarity measures performed in creating ensemble classifiers. The reported results were based on finding the greatest performance values for accuracy, sensitivity, specificity, and F-measure for each fuzzy set similarity measure and dataset and the parameter values for which they occur. The input parameters for that research are S, N, G, and E. S is the FSSM. N is the

number of top ranked features from ReliefF and ranged from 10 to 100 by an increment of 10. G is the number of clusters for grouping related features and ranged from 2 to 10 by an increment of 1. E is the number of base classifiers used to create the ensemble classifier and was fixed at 101. Note that the ensemble performance measures of accuracy, sensitivity, specificity, and F-measure may achieve their highest values at different N and G values for a FSSM and dataset.

In this current experiment, the objective is not to find the best achievable performance in the various ensemble performance measures. Instead, it is to compare performance results of a fuzzy set similarity used by itself in ensemble creation to those produced with the identical FSSM paired with each of the feature subset evaluators. This type of experiment can be used to assess the evaluators' effectiveness. To reduce the number of experiments and with this objective in mind, N is fixed at 50. This number was selected since across the experiments in the previous research, the majority of the highest performance measures were achieved at $N \leq 50$. G is fixed at 10 since it was the highest value in the previous experiments and some highest performance values were achieved at 10. E is fixed at 101 where an odd E eliminates possible ties in majority voting.

The leave-one-out cross validation method is used in the experiments. An ensemble is to be created by independently learning with up to E base classifiers for each fold if the feature subset is of sufficient quality. Each feature subset is formed by randomly selecting one feature from each of the G feature subsets. The quality of the feature subset is then measured by using one of the three feature subset evaluators selected for the experiment. The acceptable quality level of a feature subset evaluator for it to be used in a base classifier is defined here for these experiments as

$$Quality(eval) = max_E(eval) - \frac{max_E(eval) - min_E(eval)}{2} \quad (5)$$

where *eval* is the evaluator used. The $max_E(eval)$ is the maximum value of the subset evaluator *eval* over all the feature subsets for the potential base classifiers in the ensemble, and likewise, $min_E(eval)$ is the minimum value. If a feature subset's quality value is greater than or equal to the acceptable quality level, that feature subset is included for training a base classifier in the ensemble. The maximum and minimum values for *eval* are over those values it produces for each generated feature subset. Only those feature subsets meeting the quality level are used to learn a base classifier. Each base classifier in the ensemble participates in a simple majority vote on the test sample.

6. EXPERIMENTAL PERFORMANCE RESULTS

Baseline experiments were conducted to obtain performance results from using only FSSMs for grouping the top-ranked ReliefF features. These results are then used for comparison with those results from the combination of FSSMs and feature subset evaluators used to eliminate poor feature subsets. Using FSSMs by themselves, E=101 base classifiers are always used. FSSMs combined with evaluators, the number of base classifiers may vary since the evaluation step can eliminate feature subsets that are determined to be of poor quality.

Tables I, II, III, and IV for the four datasets show comparisons on *accuracy, sensitivity, specificity* and *F-measure*, respectively. Each table shows each FSSM measure paired with a feature subset evaluator. The columns labeled "None" correspond to using the FSSM without any evaluator. These columns serve as the baseline. The other columns, ICD, MiCi, and FC, are the associated results when using the corresponding evaluator. For these columns, the values

shown in bold for a value of FSSM and a feature subset evaluator pair meet or exceed those of the corresponding baseline result with no evaluator. Thus, bold entries are considered favorable and referred to as a "win." While matching the performance might be considered a "tie," it is viewed as a win because the same performance is obtained with fewer base classifiers.

The following analysis focuses on performance of the FSSMs and evaluators within a dataset based on its number of wins. The number of base classifiers used is discussed if there is no difference in wins analysis between two evaluators or between two FSSMs. Later analysis includes the actual number of base classifiers used when examining the highest ensemble performance measures.

Table I. Accuracy

FSSM	Breast				CNS				Colon				Leukemia			
	ICD	MiCi	FC	None	ICD	MiCi	FC	None	ICD	MiCi	FC	None	ICD	MiCi	FC	None
CCC	0.633	0.653	**0.714**	0.673	**0.517**	0.483	**0.500**	0.500	0.806	0.790	0.823	0.839	**0.986**	**0.972**	**0.972**	0.972
Cos	**0.673**	**0.673**	**0.673**	0.673	**0.450**	0.433	**0.450**	0.450	0.677	**0.694**	0.677	0.694	**0.875**	**0.875**	**0.875**	0.875
IncAve	**0.592**	**0.653**	**0.592**	0.551	**0.583**	**0.583**	**0.600**	0.583	0.823	0.823	0.823	0.839	0.903	0.903	**0.917**	0.917
IncMax	**0.612**	**0.571**	**0.592**	0.510	**0.617**	0.567	0.583	0.600	**0.855**	**0.871**	**0.871**	0.855	**0.972**	**0.972**	**0.986**	0.972
IncMin	**0.633**	**0.612**	**0.633**	0.612	0.617	**0.633**	**0.650**	0.633	**0.839**	0.823	**0.856**	0.839	**0.903**	**0.903**	**0.903**	0.903
Jaccard	**0.571**	**0.673**	**0.633**	0.551	**0.600**	**0.600**	**0.600**	0.600	**0.839**	**0.839**	0.823	0.839	0.917	**0.944**	0.931	0.944
Zadeh	0.633	0.673	0.694	0.714	**0.600**	0.583	0.583	0.600	**0.871**	**0.855**	**0.855**	0.855	0.903	0.903	**0.917**	0.917
Wins	5	5	6	n/a	6	3	5	n/a	4	4	3	n/a	4	5	6	n/a

Table II. Sensitivity

FSSM	Breast				CNS				Colon				Leukemia			
	ICD	MiCi	FC	None	ICD	MiCi	FC	None	ICD	MiCi	FC	None	ICD	MiCi	FC	None
CCC	0.600	**0.680**	**0.680**	0.640	**0.095**	**0.048**	**0.048**	0.048	**0.900**	**0.900**	**0.900**	0.900	**1.000**	**0.979**	**0.979**	0.979
Cos	**0.680**	**0.680**	**0.680**	0.680	**0.238**	0.143	**0.190**	0.190	**0.775**	**0.775**	**0.775**	0.775	**0.897**	**0.894**	**0.894**	0.894
IncAve	**0.520**	**0.680**	**0.520**	0.520	**0.286**	0.238	**0.286**	0.286	**0.925**	**0.925**	**0.925**	0.925	0.915	0.915	**0.936**	0.936
IncMax	**0.640**	**0.560**	**0.560**	0.480	**0.238**	**0.238**	0.095	0.143	0.900	**0.925**	**0.925**	0.925	**0.979**	**0.979**	**1.000**	0.979
IncMin	**0.640**	0.520	**0.600**	0.600	**0.238**	0.190	**0.286**	0.238	**0.925**	**0.925**	**0.925**	0.925	0.912	**0.936**	0.915	0.936
Jaccard	**0.560**	**0.680**	**0.640**	0.520	0.143	0.143	0.143	0.190	**0.925**	**0.925**	0.900	0.925	0.936	0.957	0.957	0.979
Zadeh	0.640	0.720	0.760	0.800	**0.238**	0.190	**0.286**	0.238	**0.925**	0.900	**0.925**	0.925	0.936	0.936	0.918	0.957
Wins	5	5	6	n/a	6	2	5	n/a	6	6	6	n/a	3	4	4	n/a

Table III. Specificity

FSSM	Breast				CNS				Colon				Leukemia			
	ICD	MiCi	FC	None	ICD	MiCi	FC	None	ICD	MiCi	FC	None	ICD	MiCi	FC	None
CCC	0.667	0.625	**0.750**	0.708	**0.744**	0.718	**0.744**	0.744	0.636	0.591	0.682	0.727	**0.960**	**0.960**	**0.960**	0.960
Cos	**0.667**	**0.667**	**0.667**	0.667	0.564	**0.590**	**0.590**	0.590	0.500	**0.545**	0.500	0.545	**0.840**	**0.840**	**0.840**	0.840
IncAve	**0.667**	**0.625**	**0.667**	0.583	**0.744**	**0.769**	**0.769**	0.744	0.636	0.636	0.636	0.682	**0.880**	**0.880**	**0.880**	0.880
IncMax	**0.583**	**0.583**	**0.625**	0.542	0.821	0.744	**0.846**	0.846	**0.773**	**0.773**	**0.773**	0.727	**0.960**	**0.960**	**0.960**	0.960
IncMin	**0.625**	**0.708**	**0.667**	0.625	0.821	**0.872**	**0.846**	0.846	**0.682**	0.636	**0.727**	0.682	**0.880**	**0.840**	**0.880**	0.840
Jaccard	**0.583**	**0.667**	**0.625**	0.583	**0.846**	**0.846**	**0.846**	0.821	**0.682**	**0.682**	**0.682**	0.682	**0.880**	**0.920**	**0.880**	0.880
Zadeh	**0.625**	**0.625**	**0.625**	0.625	**0.795**	**0.795**	0.744	0.795	**0.773**	**0.773**	**0.727**	0.727	**0.840**	**0.840**	**0.840**	0.840
Wins	6	6	7	n/a	4	5	6	n/a	4	4	4	n/a	7	7	7	n/a

Table IV. F-Measure

FSSM	Breast				CNS				Colon				Leukemia			
	ICD	**MiCi**	**FC**	**None**	**ICD**	**MiCi**	**FC**	**None**	**ICD**	**MiCi**	**FC**	**None**	**ICD**	**MiCi**	**FC**	**None**
CCC	0.625	**0.667**	**0.708**	0.667	**0.121**	0.061	**0.063**	0.063	0.857	0.847	0.867	0.878	**0.989**	**0.979**	**0.979**	0.979
Cos	**0.680**	**0.680**	**0.680**	0.680	**0.233**	0.150	**0.195**	0.195	0.756	**0.765**	0.756	0.765	**0.903**	**0.903**	**0.903**	0.903
IncAve	**0.565**	**0.667**	**0.565**	0.542	**0.324**	0.286	**0.333**	0.324	0.871	0.871	0.871	0.881	0.925	0.925	**0.936**	0.936
IncMax	**0.627**	**0.571**	**0.583**	0.500	**0.303**	**0.278**	0.138	0.200	0.889	**0.902**	**0.902**	0.892	**0.979**	**0.979**	**0.989**	0.979
IncMin	**0.640**	0.578	**0.625**	0.612	0.303	0.267	**0.367**	0.313	**0.881**	0.871	**0.892**	0.881	0.925	**0.926**	0.925	0.926
Jaccard	**0.571**	**0.680**	**0.640**	0.542	0.200	0.200	0.200	0.250	**0.881**	**0.881**	0.867	0.881	0.936	**0.957**	0.947	0.958
Zadeh	0.640	0.692	0.717	0.741	**0.294**	0.242	**0.324**	0.294	**0.902**	0.889	**0.892**	0.892	0.926	0.926	**0.938**	0.938
Wins	5	5	6	n/a	5	1	5	n/a	3	3	3	n/a	3	5	5	n/a

6.1. Subset Evaluator Performance

First, the performance of evaluators is analyzed using the overall number of wins (*per column*) over all seven FSSMs and is discussed in terms of ensemble performance measures for each dataset. The following analysis includes numerous comparisons that, overall, show the evaluators can improve performance and reduce the size of the ensemble.

For the *breast* dataset, the *fuzzy consistency* evaluator is consistently the best or tied for the best in terms of the number wins against the baseline. For example, in *accuracy FC* has 6 wins. This is seen in the last row in Table 1 for the breast dataset. *ICD* and *MiCi* each have 5 wins. *FC* also performs better for the other performance measures with respect to the breast dataset.

For *CNS*, the *ICD* evaluator is best, or tied for the best, for *accuracy, sensitivity, and F-measure* as seen in the last row for the *CNS* column for Tables I, II, IV. *FC* is a close second for *accuracy*. *ICD* and *FC* are best for the *F-measure* as seen in last row for the *CNS* column of Table IV. *ICD,* however, is the worst for *specificity* as seen in Table III where *FC* is the best.

For the *colon* dataset, less variety exists between the evaluators. For example, in *accuracy*, both *ICD* and *MiCi* have 4 wins and *FC* only has 3 as seen in the last row in the *colon* column in Table I. For *sensitivity*, all evaluators have 6 wins, but *MiCi* might be judged the best if the least number of base classifiers required is considered. *F*or *specificity* all three evaluators have 4 wins. For *F-measure*, all three evaluators have 3 wins.

For the *leukemia* data set, the *FC* evaluator is consistently the best, or tied for the best, with respect to all of the ensemble performance measures. *FC* has: 6 wins for accuracy in the leukemia column in Table I, 4 wins for sensitivity in Table II, 7 wins for specificity in Table III, and 5 wins for *F-measure* in Table IV. *MiCi* is nearly equal to *FC* but has only 5 wins for *accuracy*.

To summarize, *FC* is the best performer with respect to all datasets and ensemble performance measures. Its exceptions are for *sensitivity* in *CNS* and for *accuracy* in the *colon* dataset.

6.2. Fuzzy Set Similarity Performance

Next, the performance of FSSMs is analyzed using the overall number of wins (*per row)* for each FSSM across all the three evaluators and the four ensemble performance measures; therefore, there are a maximum of 12 wins for each data set. This analysis is done across all ensemble performance measures for a dataset since doing it per dataset for each ensemble performance measures provides only 3 cases to examine.

For the *breast* dataset, *Cos, IncAve, IncMax,* and *Jaccard* perform the best across all evaluators and all ensemble performance measures with 12 wins. For example, each row for *cos* is bold across all evaluators in each of the four tables for the *breast* dataset.

For the *CNS, IncAve* performs the best across all evaluators with 10 wins followed by *CCC* with 9 wins. For the *colon* dataset, *IncMax* and Zadeh perform the best across all evaluators with 10 wins.

For the *leukemia* dataset, *CCC, Cos,* and *IncMax* are the best performing FSSMs over all of the ensemble performance measures and evaluators with 12 wins.

6.3. Ensemble Performance Measures Across Datasets

Analysis across the ensemble performance measures can be examined across the 7 different FSSMs, each with 3 evaluators for a total of 21 cases. These 21 cases exist for each ensemble performance measures for each dataset. The range for the percentage of wins for each ensemble performance measure with respect to each dataset is presented.

For example, for *accuracy*, using pairs of FSSMs and evaluators, there are 16 wins for the *breast*, 15 wins for *leukemia* datasets, 14 wins for CNS, and 11 wins for colon. 16 wins corresponds to a 76% win rate (16/21). However, for accuracy in the colon dataset, there are only 11 wins or 52% of the 21 pairings. Thus, the range is from 52% (*colon*) to 76% ((*breast*). This result indicates that for most pairings of an evaluator with a FSSM the *accuracy* increases for all the datasets.

For *sensitivity*, the percentage of wins ranges from 52% (*Leukemia*) to 86% (*colon*). For *specificity*, the percentage of wins ranges from 57% (*colon*) to 100% (*leukemia*). For F-measure, the percentage of wins ranges from 43% (colon) to 76% (breast). Only for the *colon* dataset, is the percentage less than 50%. To summarize, pairing an evaluator with a FSSM has the most effect on *specificity* with the highest bottom and top values for the range. F-measure and accuracy have the lowest top range value at 76% and sensitivity has the lowest bottom range values at 43%.

6.4. Dataset Performance Across Ensemble Performance Measures

The range for the percentage of wins for each dataset with respect to each ensemble performance measure is presented. For the *breast* dataset, the percentage of wins ranges from 76% (*accuracy, sensitivity, F-measure*) to 90% (*specificity*). For *CNS*, the range is 52% (*F-measure*) to 71% (*specificity*). For the *colon* dataset, the range is 43% (*F-measure*) to 86% *(sensitivity*). For l*eukemia*, the range is 52% (*sensitivity*) to 100% (*specificity*). To summarize, the pairing of an evaluator with a FSSM has the most effect on *leukemia for* the highest top range; however, it does not have the highest bottom range. Breast has the highest bottom range and the second highest top range. Breast also has the smallest range where leukemia has the largest range. The smallest effect is on *CNS* since it has the lowest top range and almost the lowest bottom range but is second to the *colon* dataset.

6.5. Fuzzy Similarity and Evaluator Pairs with Highest Performance Measures

Finally, the performance of pairs of FSSMs and evaluators with respect to the highest values for each dataset and each ensemble performance measure is presented in Tables V, VI, VII, and VIII showing results for *accuracy, sensitivity, specificity,* and *F-measure,* respectively. The average number of base classifiers was not reported in previous tables. The number of base classifiers has

been recorded for the experiments, but due to space limitations is only presented in Table V in the column labeled #BCs for the highest performing combinations.

Table V. Configurations With Best Accuracy

Data	Acc.	FSSM	Eval	#BCs
Breast	0.714	CCC	FC	59.6
		Zadeh	None	101
CNS	0.650	Jaccard	FC	52.8
Colon	0.871	IncMax	FC	71.8
			MiCi	43.5
		Zadeh	ICD	49.6
Leukemia	0.986	CCC	ICD	46.2
		IncMax	FC	78.6

Table VI. Configurations With Best Sensitivity

Data	Sens	FSSM	Eval	#BCs
Breast	0.800	Zadeh	None	101
CNS	0.286	IncAve	FC	48.8
			ICD	50.5
		IncMin	FC	52.8
		Zadeh	FC	52.1
Colon	0.925	IncAve	MiCi	43.1
			ICD	54.3
			FC	75.3
		IncMax	MiCi	43.5
			FC	71.8
		IncMin	MiCi	38.7
			ICD	53.2
			FC	75.8
		Jaccard	MiCi	42.4
			ICD	53.8
		Zadeh	ICD	49.6
			FC	66.5
Leukemia	1.00	CCC	ICD	46.2
		IncMax	FC	78.6

Table VII. Configurations With Best Specificity

Data	Spec	FSSM	Eval	#BCs
Breast	0.750	CCC	FC	59.6
CNS	0.872	IncMin	MiCi	30.1
Colon	0.773	IncMax	FC	71.8
			MiCi	43.5
			ICD	57.5
		Zadeh	MiCi	38.9
Leukemia	0.960	IncMax	FC	78.6
			MiCi	34.0
			ICD	46.6
			None	101
		CCC	FC	77.4
			MiCi	32.8
			ICD	46.2
			None	101

Table VIII Configurations With Best F-Measures

Data	F-Meas	FSSM	Eval	#BCs
Breast	0.741	Zadeh	none	101
CNS	0.367	IncMin	FC	52.8
Colon	0.902	IncMax	FC	71.8
			MiCi	43.5
		Zadeh	ICD	49.6
Leukemia	0.989	IncMax	FC	78.6
		CCC	ICD	46.2

With respect to evaluators, over all the datasets, *FC* has the highest number of instances for *accuracy* where it had the highest at 4. This can be seen by counting the number of rows in Table V that list FC as a top-ranked evaluator. For *sensitivity* (Table VI), FC has 8 followed by *ICD* with 6. For the *F-measure* (Table VIII), FC has at 3.

For *specificity* (Table VII), *MiCi* has the highest number of instances at 5 followed by *FC* at 4. When *MiCi* does produce one of the highest performance values, it always uses the least number of base classifiers.

Overall *FC* occurs at a highest performance value 19 times across all datasets: 4 times for *accuracy* and *specificity*, 8 times for *sensitivity*, and 3 times for *F-measure*. *FC* occurs with at least one FSSM for all ensemble performance measures over all datasets except for *F-measure* for *breast* and *specificity* for *CNS*.

With respect to FSSMs producing highest ensemble performance values, *IncMax* has its highest values occurring 8 times, 2 times for each ensemble performance measures and these were only for the *colon* and *leukemia* datasets *Zadeh* also has the highest values over all the ensemble performance measures with 8 occurrences, but only the *colon* dataset has all of the four ensemble performance measures at their highest values. All the other FSSMs occur 6 or fewer times with a highest performance value. Only *Cos* never produces a high for any ensemble performance measure.

Without evaluators (None), the FSSMs *CCC, IncMax* and *Zadeh* produce the highest, or tied for the highest, performance values. *Zadeh* produces the highest performance values for *sensitivity* and *F-measure* for the *breast* dataset and matches *CCC* for *accuracy* for the *breast* data set. *CCC* and *IncMax* without evaluators produce a highest *specificity* value for the *leukemia* dataset. When no evaluator is used with a FSSM, all 101 base classifiers are used, as shown in the columns labeled #BCS.

To summarize, for feature subset evaluators, overall *FC* produces the highest ensemble performance values. For FSSMs, *IncMax* and *Zadeh* produce the highest ensemble performance values. In terms of the number of wins as analyzed in Section 6.1 for ensemble performance measures, generally *FC*, regardless of the FSSM it is paired with, is the better performing feature subset evaluator. As analyzed in Section 6.2 for ensemble performance measures, *IncMax*, regardless of the feature subset evaluator it is paired with, is the better performing FSSM.

7. CONCLUSIONS

The research presented in this paper extends that in [9] where fuzzy set similarity measures (FSSMs) are used for grouping related features for an ML process. This current research employs the use of three feature subset evaluators in combination with seven FSSMs to examine their effects on the ensemble performance measures *accuracy*, *sensitivity*, *specificity* and *F-measure*.

First FSSMs are used to create groups of related features from the best ReliefF-ranked features. Next features are randomly selected from each group to produce a feature subset as in [9]. This random selection process occurs a fixed number of times to generate feature subsets to be associated with a fixed number of base classifiers for the ensemble. Typically, all base classifiers would be used in the ensemble. Instead, the quality of a feature subset associated with a base classifier is assessed using an evaluator. Those that have low quality are eliminated because they are likely to reduce the ensemble's performance.

Much research exists that discusses the use of feature subset evaluators in the search process of finding an optimal set of features for machine learning. Here three feature subset evaluators: interclass distance (*ICD*), maximal information compression index (*MiCi*), and fuzzy consistency (*FC*) are used, not in a search process, but to determine the quality of the feature subsets produced by the random subspace method of feature selection as applied to the feature groups formed using FSSMs. *FC* is an adaption of the crisp consistency measure which requires the discretization of feature values.

The experimental results showed that in most cases the FSSM paired with a feature subset evaluator outperforms the corresponding FSSM by itself, although it is acknowledged that it is difficult to know which combination will yield the most improvement. An added benefit is that the learning process only needs to occur on typically about half the number of base classifiers since the evaluator produces a quality assessment and those of low quality are eliminated from the ensemble.

From this study, in general the *FC* measure is the best performing feature subset evaluator paired with the FSSMs. As for FSSMs, in general *IncMax* paired with feature subset evaluators is the best performing for the colon and leukemia datasets. *CCC* and *Zadeh* with FC perform the best the breast and CNS datasets.

Future work will investigate other feature subset evaluators and the application of this pairing with FSSMs on other datasets. Initial experimental results also suggest that an aggregation of evaluators on a feature subset might present even higher quality feature subsets for an ensemble's base classifiers. The idea is that these higher quality feature subsets could further improve ensemble performance measures and reduce the number of needed base classifiers. In addition, a study to investigate possible relationships both between evaluators and ensemble performance measures and between evaluators and datasets might provide better insight to their use.

REFERENCES

[1] Molina, L. C., Belanche, L., Nebot, A. (2002) "Feature Selection Algorithms: A Survey and Experimental Evaluation," *IEEE International Conference on Data Mining*, Maebashi City, Japan, Dec. 9 – 12.

[2] Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A. (2015) *Feature Selection for High-Dimensional Data*, Springer International Publishing AG, Switzerland.

[3] Wan, Cen (2019) *Hierarchical Feature Selection for Knowledge Discovery Application of Data Mining to the Biology of Ageing*, Springer Nature Switzerland AG.

[4] Ho, T. K. (1998) "The random subspace method for constructing decision forests," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20 no. 8, pp. 832–844.

[5] Robnik-Sikonja, M. & Kononenko, I. (1997) "An adaptation of relief for attribute estimation in regression," in: D. H. Fisher 635 (Ed.), *Fourteenth International Conference on Machine Learning*, Morgan Kaufmann, pp. 296–304.

[6] Kuncheva, L. I., Rodríguez, J. J., Plumpton, C. O., Linden, D. E. J., & Johnston, S. J. (2010) "Random Subspace Ensembles for fMRI Classification," *IEEE Transactions on Medical Imaging*, Vol. 29, No. 2, pp. 531- 542.

[7] Chaudhury, B., Goldgof, D. B., Hall, L. O., Gatenby, R. A., Gillies, R. J., & Drukteinis, J. S. (2015) "Correlation based random subspace ensembles for predicting number of axillary lymph node metastases in breast dce-mri tumors," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2164–2169.

[8] Bland, J. M. & Altman, D. (1986) "Statistical methods for assessing agreement between two methods of clinical measurement," *The Lancet*, vol. 327 no. 8476, pp. 307–310.

[9] Cross, V., Zmuda, M., Paul, R., & Hall, L. O. (2020) "Fuzzy Set Similarity for Feature Selection in Classification", *2020 International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 19 – 24, Glasglow, United Kingdom.

[10] Bolón-Canedo, Verónica & Alonso-Betanzos, Amparo (2019) "Ensemble for feature selection: Review and Trends," *Information Fusion*, Vol 52, pp. 1-12.

[11] Álvarez-Estévez, Diego, Sánchez-Maroño, Noelia, Alonso-Betanzos, Amparo, & Moret-Bonillo, Vicente (2011) "Reducing dimesionality in a database EEG sleep arousals," *Expert Systems with Applications*, 38(6), pp. 7746-7754.

[12] Morán-Fernández, L., Bolón-Canedo, V., & Alonso-Betanzos, A. (2017) "Centralized vs. distributed feature selection methods based on data complexity measures," *Knowl. Based Syst.* 117 pp. 27–45.

[13] Liu, H., Liu, L., & Zhang, H. (2010) "Ensemble gene selection by grouping for microarray data classification," *Journal of Biomedical informatics*, 43 (1) pp, 81–87.

[14] Duch, W. (2006) "Filter Methods," in *Feature Extraction Foundations and Applications*, Eds. I. Guyon, M. Nikravesh, S. Gunn, L. Zadeh, Berlin: Springer Berlin Heidelberg.

[15] L. Rokach (2010), "Ensemble-based classifiers," *Artif Intell Rev*, vol. 33, pp. 1–39.

[16] Holmes, G., Donkin, A., & Witten, I. H. (1994) "Weka: A machine learning workbench," *Proceedings of the Second Australia and New Zealand Conference on Intelligent Information Systems*, Brisbane, Australia, 1994 Web link: https://www.cs.waikato.ac.nz/ml/weka/

[17] Dubois, D. & Prade, H. (1982) "A unifying view of comparison indices in a fuzzy set-theoretic framework," in R Yager Ed. *Fuzzy Set and Possibility Theory: Recent Developments*, Pergamon Press, New York, NY pp. 3-13.

[18] P. Jaccard (1912) "The distribution of the flora in the alpine zone", *New Phytologist*, vol. 11, pp. 37–50.

[19] Han, Jiawei, Kamber, M., & Pei, Jian (2012) *Data Mining: Concepts and Techniques*, 3rd Ed. Morgan Kaufmann, Burlington, MA.

[20] Mitra, P., Murthy, C.A., & Pal, S.K. (2002) "Unsupervised Feature Selection Using Feature Similairty," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 3.

[21] Dash, M. & Liu, Huan (2003) "Consistency-based search in feature selection," *Artificial Intelligence*, Vol. 151, pp. 155-176.

AUTHORS

Dr. Valerie Cross is an Associate Professor in the Computer Science and Software Engineering department at Miami University in Oxford, OH. She earned a B.S in Computer Science and a B.S. in Statistics from West Virginia University, a Masters in Computer Science at the University of Colorado, Boulder and a PhD in Computer Science from Wright State University. Her research interests include fuzzy set theory and approximate reasoning, ontology alignment, ontologies in biomedical and bioinformatics applications and the use of fuzzy set theory in machine learning.

Dr. Zmuda is an Associate Professor in the Computer Science and Software Engineering department at Miami University in Oxford, OH. He earned a B.S in Computer Science and a B.S. in Mathematics from Eastern Michigan University and a M.S. and Ph.D. in Computer Science and Engineering at Wright State University. His research interests include the application of AI techniques such as fuzzy set theory and optimization to problems in medicine and virtual reality.