Rough Set Subspace Error-Correcting Output Codes

Mohammad ali Bagheri*† Qigang Gao *‡ Sergio Escalera§¶
* *Faculty of Computer Science, Dalhousie University, Halifax, Canada*
§ *Centre de Visio per Computador, Campus UAB, Edifici O, Bellaterra, 08193 Barcelona, Spain*
Email: †*bagheri@cs.dal.ca,* ‡ *qggao@cs.dal.ca* ¶*sergio@maia.ub.es*

***Abstract*—Among the proposed methods to deal with multiclass classification problems, the Error-Correcting Output Codes (ECOC) represents a powerful framework. The key factor in designing any ECOC matrix is the independency of the binary classifiers, without which the ECOC method would be ineffective. This paper proposes an efficient new approach to the ECOC framework in order to improve independency among classifiers. The underlying rationale for our work is that we design three-dimensional codematrix, where the third dimension is the feature space of the problem domain. Using rough set-based feature selection, a new algorithm, named ”Rough Set Subspace ECOC (RSS-ECOC)” is proposed. We introduce the *QuickMultipleReduct* algorithm in order to generate a set of reducts for a binary problem, where each reduct is used to train a dichotomizer. In addition to creating more independent classifiers, ECOC matrices with longer codes can be built. The numerical experiments in this study compare the classification accuracy of the proposed RSS-ECOC with classical ECOC, one- versus-one, and one-versus-all methods on 24 UCI datasets.The results show that the proposed technique increases the classification accuracy in comparison with the state of the art coding methods.**

***Keywords*-Error Correcting Output Codes; Rough Set; Multiclass classification; Feature subspace;**

I. Introduction

A common task in many real-world pattern recognition problems is to discriminate among instances that belong to multiple classes. The predominant approach to deal with such problems is to recast the multiclass problem into a series of smaller binary classification tasks, which is referred to as ”class binarization” [1]. In this way, two-class problems can be solved by binary classifiers and the results can then be combined so as to provide a solution to the original multiclass problem. Among the proposed methods for approaching class binarization, there are three well-known methods including: one-versus-all (OVA) one-versus-one (OVO)and Error Correcting Output Codes.In one-versus-all, the multiclass problem is decomposed into several binary problems in the following way: for each class a binary classifier is trained to discriminate among the patterns of the class and the patterns of the remaining classes. In the one-versus-one approach, one classifier is trained to split each possible pair of classes. In both approaches, the final classification prediction is usually obtained by means of a voting or committee procedure. The third approach proposed by Dietterich and Bakiri [2] presents a general framework for class binarization approaches in order to enhance generalization ability of binary classifiers, which is known as Error Correcting Output Codes (ECOC).

The basis of the ECOC framework is to decompose a multiclass problem into a larger number of binary problems. In this way, each classifier is trained on a two meta-class problem, where each meta-class consists of some combinations of the original classes. The ECOC method can be broken down into two stages: encoding and decoding. The aim of the encoding stage is to design a discrete decomposition matrix (codematrix) for the given problem. Each row of the codematrix, namely codeword, is a sequence of bits representing each class, where each bit identifies the membership of the class to a classifier [3]. In the decoding stage, the final classification decision is obtained based on the outputs of binary classifiers. Given an unlabeled test sample, each binary classifier casts a vote to one of the two meta-classes used in its training. The output vector is compared to each class codeword of the matrix and the test sample is assigned to the class whose codeword is closest, according to some distance measure, to the output vector. Because of its ability to correct the bias and variance errors of the base classifiers [4], [5], the ECOC framework has been successfully applied to a wide range of applications.

One of the key factors to the success of ECOC methods is the independence of binary classifiers, without which the output coding approach would be ineffective [4]. In ECOC methods, the codematrix can be considered as the core component to generate independent classifiers. Accordingly, most previous methods to design the ECOC matrix try to build an optimal codematrix, usually by optimizing the row and column separation criteria. Many researchers, however, agree that random generation of codematrix is a ”reasonably” accurate approach, and that ”more sophisticated methods might have only marginal effect on testing error” [2], [1], [6]. It has also been shown that large random codes would not be outperformed by codes designed for their error-correcting capabilities [7]. Therefore, the overall performance of ECOC codes built by different strategies for a same base classifier tends to be very similar, especially as the length of codewords increases. One efficient approach to increase diversity among an ensemble of classifiers is to train each learner with data that consist of different feature subsets, leading to uncorrelated errors of base learners

[8], [9]. This idea, usually called subspace approach, can effectively make use of diversity of base learners to reduce the variance as well as the bias errors.

The key problem of this approach is how to find a set of feature subsets with high prediction power. Rough set theory, introduced by Pawlak [10], has shown to be a powerful tool to deal with uncertainty, in particular to the feature selection problem [11]. In the rough set structure, reducts are the minimal feature subsets which keep the discernibility of the original dataset and have no redundant features. It is worth noting that there are usually multiple reducts for a given dataset. All reducts can be employed for building base classifiers. However, most of the applications select the reduct with the fewest features to construct a classifier. In this way, we may lose information hidden in other reducts.

Inspired by the subspace idea and by utilizing the rough set-based features selection, this paper proposes a new approach to the ECOC method, named Rough-Set Subspace ECOC. The strategy consists of using different rough-based feature subsets for base dichotomizers, leading to more independent classifiers and, in consequence, increasing the overall system accuracy. In addition to the design of more independent classifiers, this new technique also allows for the design of larger codes compared to classical methods. Results on several public and challenging data sets show the benefits and better performance of the proposed method in comparison to state-of-the-art approaches.

The rest of this paper is organized as follows: Section 2 provides a brief introduction to the ECOC framework. The preliminary knowledge on rough set as well as rough set feature selection algorithm is given in Section 3. The proposed ECOC approach is explained in detail in Section 4. Section 5 reports the experimental results and Section 6 concludes the paper.

II. Error Correcting Output Codes

First, we briefly describe some notations used in this paper:

- $T = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_m}, y_m)\}$. A training set; where $\mathbf{x_i} \in R^n$; and each label,y_i, is an integer belongs to $Y = \{1, 2, \ldots, N_c\}$, where N_c is the number of classes
- $h = \{h_1, h_2, \ldots, h_L\}$: A set of L binary classifiers.

The basis of the ECOC framework consists of designing a codeword for each of the classes. This method uses a matrix M of $\{1, -1\}$ values of size $N_c \times L$, where L is the number of codewords codifying each class. This matrix is interpreted as a set of L binary learning problems, one for each column. That is, each column corresponds to a binary classifier, called *dichotomizer* h_j, which separates the set of classes into two metaclasses. Instance $\mathbf{x}$, belonging to class i, is a positive instance for the jth classifier if and only if $M_{ij} = 1$ and is a negative instance if and only if $M_{ij} = -1$. Table 1 shows a possible binary coding matrix for a 4-class problem $\{c_1, \ldots, c_4\}$ with respective codewords $\{M(r, .)\}$ that uses six dichotomizers $\{h_1, \ldots, h_6\}$. In this table, each column is associated with a dichotomy classifier,h_j, and each row is a unique codeword that is associated with an individual target class. The white cells of the table refer to $+1$ and the dark cells stand for -1. For example, h_3 recognizes two meta-classes: original classes 1 and 4 form the first meta-class, and the other two form the second one. When testing an unlabeled pattern,$\mathbf{x}^*$, each classifier outputs a "-1" or "1" value, creating a L long output code vector. This output vector is compared to each codeword in the matrix, and the class whose codeword has the closest distance to the output vector is chosen as the predicted class. The process of merging the outputs of individual binary classifiers is called decoding. The most commonly decoding method is the Hamming distance. Several decoding strategies (combination methods other than distance methods) have been proposed in the literature such as probabilistic approaches and loss-functions strategies.

Table I
An example of an ECOC matrix

Class	h_1	h_2	h_3	h_4	h_5	h_6
c_1		■		■	■	
c_2			■	■		■
c_3	■		■		■	
c_4	■	■		■		

The ECOC method was then extended by Allwein et al. [12] using a coding matrix with three values, {1,0,-1}, where the zero value means that a given class is not considered in the training phase of a particular classifier. This extended codeword is denominated sparse random code and binary ECOC is named dense random code. Thanks to this unifying approach, classical multiclass classification methods, such as one-versus-one and one-versus-all, can be represented as an ECOC matrix.

III. Rough set feature selection

Rough set feature selection (RSFS) is a filter based procedure by which knowledge can be extracted from a domain in a concise way: retaining the information content while reducing the amount of knowledge involved [13]. For a better understanding of RSFS, we first explain the basic concepts of the rough set theory in the next subsection.

A. Rough set

In the rough set framework, datasets are usually given in the form of tables, called information system. An information system formulated as a four-tuple $IS =< U, A, V, F >$, where $U = \{x_1, x_2, ..., x_m\}$ is a set of finite and nonempty objects, called the universe; A is the set of features characterizing the objects; V is the set of values that feature a may take; and F is the set of information functions $f : U \times A \to V$.

Central to RSFS is the concept of indiscernibility. With any arbitrary feature set $P \in A$, there is an associated indiscernibility relation $IND(P)$:

$$IND(P) = \{(x, y) \in U^2 | \forall a \in P, f(x, a) = f(y, a)\} \quad (1)$$

If $(x, y) \in IND(P)$, then x and y are indiscernible by features in P. Clearly, an indiscernibility relation is an equivalent relation and satisfies symmetry, reflexivity and transitivity. The equivalent class induced by P is denoted by $[x]_p$: $[x]_p = \{x| < x_i, x >\in IND(P), x \in U\}$

The set of equivalence classes forms a concept system, which is used to characterize feature subsets in the information system. Let $X \in U$. X can be approximated by the two sets of equivalence classes:

$$\begin{aligned} \underline{P}(X) &= \{x \in U | [x]_P \subseteq X\} \\ \overline{P}(X) &= \{x \in U | [x]_P \cap X \neq \emptyset\} \end{aligned} \quad (2)$$

$\underline{P}(X)$ and $\overline{P}(X)$ are called lower and upper approximations of X in terms of feature set P, respectively. Let P and Q be two sets of features inducing equivalence relations over U. Then the positive and negative regions can be defined as:

$$POS_P(Q) = \bigcup_{X \in U/Q} \underline{P}(X) \quad (3)$$

$$NEG_P(Q) = U - \bigcup_{X \in U/Q} \overline{P}(X) \quad (4)$$

The positive region contains all objects that can be classified into classes of U/Q using the information in feature set P. In other words, having only the values of feature set P, we can predict the value of feature set Q. The negative region, $NEG_P(Q)$, is the set of objects that cannot be classified to classes of U/Q.

An important issue in data mining, especially in feature selection, is discovering dependencies between features. Obviously, a set of features Q depends totally on a set of features P, which are denoted by $P \to Q$, if all feature values from Q are uniquely determined by values of features from P. If a functional dependency exists between values of Q and P, then Q depends totally on P. In the rough set framework, dependency is defined in the following way: For $P, Q \subset A$, it is said that Q depends on P in a degree $k(0 \leq k \leq 1)$, which is denoted by $P \Rightarrow_k Q$, where:

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|} \quad (5)$$

where $|A|$ stands for the cardinality of set A. The dependency coefficient,k, measures the approximation power of a set of conditional features. If $k = 0$, then Q does not depend on P; if $0 < k < 1$, Q depends partially (in a degree k) on P, and if $k = 1$, Q depends totally on P.

By evaluating the change in dependency when a feature is removed from the set of considered conditional features, a measure of the significance of the feature can be achieved. The higher the change in dependency, the more significant the feature is. If the significance is 0, then the feature is dispensable. More formally, given P, Q and a feature $a \in P$:

$$\sigma_P(Q, a) = \gamma_P(Q) - \gamma_{P-a}(Q) \quad (6)$$

B. *Rough set-based feature selection*

Rough set feature selection is achieved by comparing equivalence relations generated by sets of features. Features are removed so that the reduced set provides the same predictive capability of the decision feature as the original. A reduct is defined as a minimal subset, R, of the initial feature set, C, if:

$$\begin{aligned} &1. \quad \gamma_R(D) = \gamma_C(D) \\ &2. \quad \forall a \in R : \gamma_R(D) > \gamma_{R-a}(D) \end{aligned} \quad (7)$$

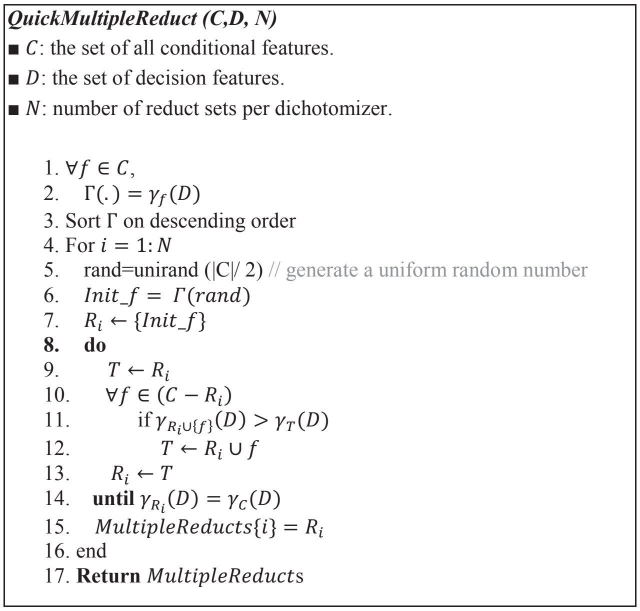
This means that no features can be removed from the reduct set without affecting the dependency degree. Therefore, a minimal subset by this definition may not be the global minimum (a reduct of smallest cardinality). Thus, a given dataset may have a number of reduct sets. The intersection of all the reducts is called the *core*, the feature subset which cannot be removed from any reduct because the discernibility of the system will decrease.

IV. Rough Set Subspace ECOC (RSS-ECOC)

As we stated earlier, the key success point in ECOC design is that the error committed by each of binary classifiers needs to be uncorrelated, which makes the ECOC approach effective in correcting some errors. In this paper, we have proposed a novel approach to the ECOC design. The key idea of the proposed Rough Set Subspace ECOC is based on using feature space in the design process of the ECOC matrix. That is, each dichotomizer is trained with a different feature subset, leading to better classification accuracy. From the design process point of view, we generate three-dimensional codematrix, where the third dimension is the feature space of the problem domain. To generate this framework, first, a two-dimensional codematrix is generated from a previous set of matrices that maximizes the minimum distances between any pair of codewords. Then, for each dichotomizer, we find a number of rough-set-based reduct set. Each reduct set is used to train the corresponding dichotomizer. The main goal is to define an algorithm to look for multiple reducts. This is explained in detail in the following subsection.

A. *Rough set-based multiple feature subset selection*

The key component of rough set based feature selection in the proposed framework is how to generate multiple reducts for a binary classifier. Several algorithms for finding reduct sets have been proposed based on heuristic strategies, such as discernibility matrix evolutionary methods (for example Genetic Algorithm and Particle Swarm Optimization



```
QuickMultipleReduct (C,D, N)
■ $C$: the set of all conditional features.
■ $D$: the set of decision features.
■ $N$: number of reduct sets per dichotomizer.

$\forall f \in C$,
    $\Gamma(.) = \gamma_f(D)$
Sort $\Gamma$ on descending order
For $i = 1{:}N$
    rand=unirand (|C|/ 2) // generate a uniform random number
    $Init\_f = \Gamma(rand)$
    $R_i \leftarrow \{Init\_f\}$
    do
        $T \leftarrow R_i$
        $\forall f \in (C - R_i)$
            if $\gamma_{R_i \cup \{f\}}(D) > \gamma_T(D)$
                $T \leftarrow R_i \cup f$
        $R_i \leftarrow T$
    until $\gamma_{R_i}(D) = \gamma_C(D)$
    $MultipleReducts\{i\} = R_i$
end
Return $MultipleReducts$
```

Figure 1. The pseudo-code of the *QuickMultipleReduct* algorithm

information entropy and dependency function. Many of these methods are computationally intensive and are not appropriate for relatively large datasets. Here we utilize a heuristic algorithm, named *QUICKREDUCT* [14]. The *QUICKREDUCT* algorithm tries to find reduct sets without exhaustively generating all possible subsets. The algorithm starts off with an empty reduct set. Then, it evaluates the change in dependency caused by adding a feature to the current set. The next feature to be added is the most significant feature. This procedure will continue until the maximum possible significance value for the dataset is obtained. The heuristic used is based on (7), where $\sigma_{P \cup a}(Q, a)$ is evaluated for each feature, given reduct candidate P.

However, this algorithm will find a unique reduct set for a given dataset. In this work, we modified the *QUICKREDUCT* algorithm in order to obtain different reduct sets. The modified algorithm is named *QuickMultipleReduct* and works as follows: instead of finding the most significant feature in the first iteration of the algorithm, we sort the available features in terms of their significance values in descending order and randomly choose one feature from the top half features. This modification results in diverse reduct sets. One may argue that reduct sets generated by this procedure may not be optimal in terms of prediction power. However, we should take into account that, while the feature selection seeks to find an optimal subset of features, the goal of the RSS-ECOC approach is to build diverse and accurate dichotomizers. This modification results in independent errors of classifiers and makes the ECOC approach effective in correcting the errors. Figure 1 shows the proposed *QuickMultipleReduct* algorithm.

Table II

SUMMARY OF THE USED DATASETS

	Dataset	# instances	# features	# classes
1	**Abalone**	4177	8	3
2	**Cleafs**	4758	64	8
3	**Cmc**	1473	9	3
4	**Derm**	358	34	6
5	**Ecoli**	336	7	8
6	**Glass**	214	9	6
7	**Lymph**	148	18	4
8	**Mfeat-fac**	2000	216	10
9	**Mfeat-fou**	2000	76	10
10	**Mfeat-kar**	2000	64	10
11	**Mfeat-mor**	2000	6	10
12	**Mfeat-pix**	2000	240	10
13	**Mfeat-zer**	2000	47	10
14	**Optdigits**	5620	64	10
15	**Pendigits**	10992	16	10
16	**Sat**	6435	36	6
17	**Thyroid**	215	5	3
18	**Vehicle**	846	18	3
19	**Vertebral**	310	6	3
20	**Vowel**	528	10	11
21	**Waveforms**	5000	40	3
22	**Wine**	178	13	3
23	**Yeast**	1484	8	10
24	**Zoo**	101	16	7

V. EXPERIMENTAL COMPARISON

A. Experimental settings

- **Data:** The proposed RSS-ECOC method is validated on 24 multiclass datasets from the UCI machine learning repository [15]. The UCI datasets are widely used by the Machine Learning community for evaluating different methods. Table 2 shows the number of classes, instances, and features for each UCI dataset.
- **Methods:** we compared our proposed method with OVO, OVA, and classical ECOC methods. The class of an instance in the ECOC schemes is chosen using the Exponential Loss-Weighted (ELW) decoding [16]. To limit the computational complexity of the experiments in the RSS-ECOC design, we set the maximum number of different reduct sets per each nontrivial dichotomizer to 10. Thus, codewords are about 10 times longer in Subspace ECOC design for both dense and sparse ECOC methods. In this study, two base learners were chosen: a classification and regression tree (CART) with the Gini-index as a split criterion and a multi-layer perceptron (MLP) with 10 hidden nodes and the hyperbolic tangent transfer function.
- **Evaluation measurements:** The classification performance was obtained by means of 10-fold cross-validation. Moreover, we include statistical tests to look for statistical significance among the obtained performances.

B. Experimental results

The average accuracy of the six methods for the 24 datasets is presented in Table 3 and Table 4. In these tables, the means of prediction accuracy over 10 runs (expressed in %) are reported for each ECOC design method on

Table III
CLASSIFICATION ACCURACIES OF DIFFERENT METHODS USING CART

	OVO	**OVA**	**dense ECOC**	**sparse ECOC**	**dense RSS-ECOC**	**sparse RSS-ECOC**
Abalone	65.28	66.07	65.16	65.74	66.77	66.38
Cleafs	75.71	74.90	76.07	76.21	77.45	77.69
Cmc	51.98	52.00	49.39	52.27	53.38	54.57
Derm	92.04	94.36	96.27	95.57	97.26	97.42
Ecoli	81.95	84.44	85.54	85.90	87.71	87.19
Glass	58.33	57.92	65.68	65.20	66.32	65.63
Lymph	72.16	77.51	78.03	82.71	84.42	85.79
Mfeat-fac	95.19	92.21	95.07	96.40	97.37	98.20
Mfeat-fou	74.69	63.67	75.69	76.31	79.54	80.64
Mfeat-kar	91.33	82.05	92.69	91.90	95.93	95.95
Mfeat-mor	72.95	73.26	74.17	74.10	74.13	73.99
Mfeat-pix	91.07	86.00	91.21	94.21	94.11	96.62
Mfeat-zer	77.10	77.00	81.81	82.33	84.13	84.59
Optdigits	92.73	93.81	95.95	95.49	97.94	97.47
Pendigits	98.18	97.44	98.94	98.92	99.31	99.54
Sat	85.96	85.55	87.38	87.91	88.97	89.52
Thyroid	92.30	95.51	95.08	94.98	95.34	96.37
Vehicle	80.47	79.92	79.80	80.92	81.41	81.63
Vertebral	79.74	83.23	81.84	83.01	84.37	84.55
Vowel	84.89	74.32	87.17	88.56	87.53	91.47
Waveforms	82.95	83.76	79.74	84.06	86.44	86.54
Wine	94.74	96.02	94.53	97.37	98.38	98.62
Yeast	56.10	57.76	59.72	59.18	60.12	60.61
Zoo	81.88	91.60	93.57	92.08	94.43	94.51
Average accuracy	**80.40**	**80.01**	**82.52**	**83.39**	**84.70**	**85.23**

Table IV
CLASSIFICATION ACCURACIES OF DIFFERENT METHODS USING CART

	OVO	**OVA**	**dense ECOC**	**sparse ECOC**	**dense RSS-ECOC**	**sparse RSS-ECOC**
Abalone	58.29	56.26	58.26	59.47	61.05	62.24
Cleafs	69.00	64.20	73.91	73.91	75.78	75.62
Cmc	49.13	47.62	48.72	49.98	51.39	51.67
Derm	96.00	91.80	97.60	97.63	97.68	97.81
Ecoli	83.12	76.30	85.10	83.62	85.69	83.27
Glass	65.31	59.63	65.06	67.67	67.24	70.53
Lymph	76.32	72.95	76.54	79.43	81.03	81.65
Mfeat-fac	88.02	82.83	93.95	93.29	95.45	94.76
Mfeat-fou	74.57	69.31	77.90	79.10	81.55	81.40
Mfeat-kar	82.05	80.10	90.83	89.45	94.36	93.36
Mfeat-mor	67.90	62.29	69.36	69.33	69.52	69.83
Mfeat-pix	87.26	82.12	92.05	91.79	95.07	94.71
Mfeat-zer	68.33	63.62	75.07	74.76	78.26	77.71
Optdigits	91.00	85.94	95.98	94.74	97.45	96.73
Pendigits	94.23	89.85	97.86	97.30	98.42	97.99
Sat	84.64	81.47	88.63	88.18	89.69	89.85
Thyroid	92.56	91.30	91.30	91.60	92.36	92.81
Vehicle	72.35	69.51	72.36	74.61	76.99	76.05
Vertebral	79.46	77.60	77.71	79.63	78.80	80.54
Vowel	71.26	71.11	83.58	82.34	86.97	86.71
Waveforms	74.86	71.52	71.08	78.08	82.01	83.07
Wine	94.43	89.97	92.01	95.56	97.41	96.67
Yeast	54.22	49.97	57.85	58.94	58.90	58.54
Zoo	85.77	88.22	91.84	91.16	93.38	93.02
Average accuracy	**77.50**	**73.98**	**80.19**	**80.90**	**82.77**	**82.77**

the considered datasets.For each dataset, the best accuracy achieved among all tested algorithms is bolded.

In order to show the superiority of the proposed RSS-ECOC method, statistical analysis is necessary. According to the recommendations of Demsar [17], we consider the use of non-parametric tests. Non-parametric tests are safer than parametric tests, such as ANOVA and t-test, since they do not assume normal distribution or homogeneity of variance. In this study, we employ the Iman-Davenport test. If there are statistically significant differences in the classification performance, then we can proceed with the Nemenyi test as a post-hoc test, which is used to compare all methods with each other [17].

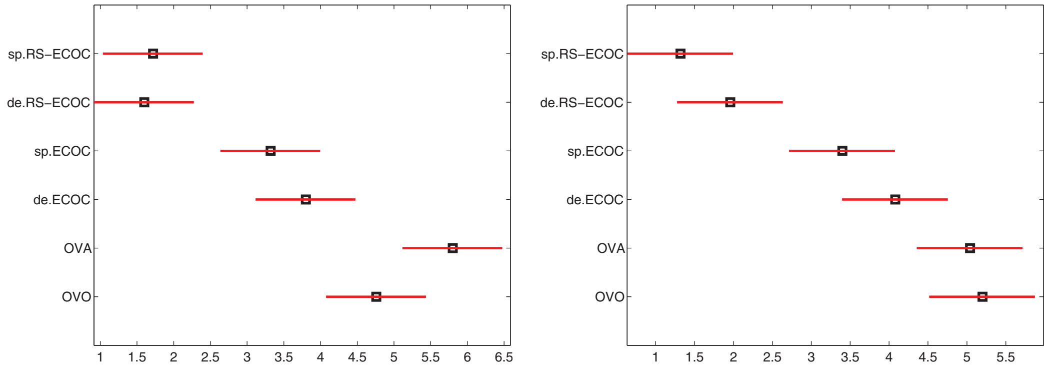
To do that, we first rank competing methods for each dataset. The method's mean rank is obtained by averaging its ranks across all experiments. Then, we use the Friedman test to compare these mean ranks to decide whether to reject the null hypothesis, which states that all considered methods have equivalent performance. Iman and Davenport [18] found that this statistic is undesirably conservative, and proposed a corrected measure. Applying this method, we can reject the null hypothesis, and show that there exists significant statistical difference among the rival methods.

Further, to compare rival methods with each other, we apply the Nemenyi test, as illustrated in Figure 4. In this figure, the mean rank of each method is indicated by a square. The horizontal bar across each square shows the critical difference. Two methods are significantly different if their corresponding average ranks differ by at least the critical difference value. i.e., their horizontal bars are not overlapping. The results in Table 3 and Table 4, along with the statistical tests presented in Figure 4, indicate that in general the proposed ECOC approach receives the best performance among all methods for both MLP and CART as the base learners. It is also ranked as the preferred method in these classification algorithms for most of the UCI datasets. The results also show significant difference between RSS-ECOC and classical ECOC for both dense and sparse schemes.

An analysis of the results shows that when the number of training patterns is relatively small compared with the dimensionality of data, the subspace approach is usually a better choice. In these cases, the training sample size relatively increases with the dimensionality of data. [8] showed that while most classification approaches suffer from the curse of dimensionality, the subspace approach can benefit from high dimensionality.

VI. CONCLUSION

In this paper, we presented a novel approach to Error-Correcting Output Codes to deal with multi-class classification problems. The proposed technique is based on designing the ECOC matrix code using different random subspace in order to generate more independent classifiers. For this



(a) CART Classifier

(b) MLP Classifier

Figure 2. Comparison results of rival methods based on the Nemenyi test

task, each dichotomizer is trained using different rough-based feature subsets computed using the proposed Quick-MultipleReduct procedure, leading to a more independent classifiers and, in consequence, increasing the overall accuracy of the ensemble. In addition to creating more independent classifiers, in the proposed Subspace ECOC technique, ECOC matrices with longer codes can be generated. The experimental evaluation over several UCI Machine Learning repository datasets show that using a neural network and decision tree as the base classifier, significant performance improvements can be obtained compared to the one-versus-one, one-versus-all, and classical dense and sparse ECOC methods.

REFERENCES

[1] N. Garca-Pedrajas and D. Ortiz-Boyer, "An empirical study of binary classifier fusion methods for multiclass classification," *Information Fusion*, vol. 12, no. 2, pp. 111–130, 2011.

[2] T. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research*, vol. 2, pp. 263–286, 1995.

[3] S. Escalera, O. Pujol, and P. Radeva, "Separability of ternary codes for sparse designs of error-correcting output codes," *Pattern Recognition Letters*, vol. 30, no. 3, pp. 285–297, 2009.

[4] E. Kong and T. Dietterich, "Why error-correcting output coding works with decision trees," Technical Report, Department of Computer Science, Oregon State University, Corvallis, OR., Tech. Rep., 1995.

[5] T. Windeatt and R. Ghaderi, "Coding and decoding strategies for multi-class learning problems," *Information Fusion*, vol. 4, no. 1, pp. 11–21, 2003.

[6] R. E. Shapire, "Using output codes to boost multiclass learning problems," in *14th Int. Conference on Machine Learning*, D. H. Fisher, Ed., 1997, pp. 313–321.

[7] G. M. James and T. Hastie, "The error coding method and pict's," *Computational and Graphical Statistics*, vol. 7, pp. 377–387, 1998.

[8] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattem Analysis and Machine Intelligence*, vol. 20, pp. 832–844, 1998.

[9] C. Zor, T. Windeatt, and B. Yanikoglu, "Bias-variance analysis of ecoc and bagging using neural nets," *Studies in Computational Intelligence*, vol. 373, pp. 59–73, 2011.

[10] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Norwell, MA: Kluwer, 1991.

[11] R. Swiniarski and A. Skowron, "Rough set methods in feature selection and recognition," *Pattern Recognition Letters*, vol. 24, no. 6, pp. 833–849, 2003.

[12] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," *Journal of Machine Learning Research*, vol. 1, pp. 113–141, 2001.

[13] R. Jensen and Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, 2009.

[14] A. Chouchoulas and Q. Shen, "Rough set-aided keyword reduction for text categorization," *Applied Artificial Intelligence*, vol. 15, no. 9, pp. 843–873, 2001.

[15] C. Blake and C. Merz, "Uci repository of machine learning databases, department of information and computer sciences, university of california, irvine," 1998.

[16] S. Escalera, O. Pujol, and P. Radeva, "On the decoding process in ternary error-correcting output codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 120–134, 2010.

[17] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[18] R. Iman and J. Davenport, "Approximations of the critical regions of the friedman statistic," *Communications in Statistics*, vol. 6, pp. 571–595, 1980.