ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

Performance Evolution of Different Coding Methods with β - density Decoding Using Error Correcting Output Code Based on Multiclass Classification

Devangini Dave, M. Samvatsar, P. K. Bhanodia

***Abstract*— A common way to address a multi-class classification problem is to design a model that consists of handpicked binary classifiers and to combine them so as to solve the problem. Error-Correcting Output Codes (ECOC) is one such framework that deals with multi-class classification problems. ECOC framework is a powerful tool to deal with multi-class classification problems because of the recent works in the ECOC domain has shown promising results demonstrating improved performance. The error correcting ability improves and enhances the generalization ability of the base classifiers. The main goal of this paper is to compare Discriminate ECOC coding method with one-versus-one, one-versus-all, dense random, sparse random coding methods using pessimistic β-density as decoding method which shows significant performance improvement. The tool we have used for performing experiments is MATLAB.**

***Index Terms*— Error Correcting Output Codes (ECOC), Coding, Decoding, Discriminate ECOC (DECOC)**

I. INTRODUCTION

The task of supervised machine learning can be seen as the problem of finding an unknown function C(x) given the training set of example pairs < xi; C (xi) >. C(x) is usually a set of discrete labels. For example, in face detection, C(x) is a binary function c(x) belongs to {face, nonface} in optical digit recognition c(x) belongs to {0………9}.In order to address the binary classification task many techniques and algorithms have been proposed: decision trees, neural networks, large margin classification techniques, etc. Some of those methods can be easily extended to multiclass problems. However, some other powerful and popular classifiers, such as AdaBoost [15] and Support Vector machines [14], do not extend to multiclass easily. In those situations, the usual way to proceed is to reduce the complexity of the multiclass problem into multiple simpler binary classification problems. There are many different approaches for reducing multiclass to binary classification problems. The simplest approach considers the comparison between each class against all the others. This produces N_c binary problems, where Nc is the number of classes. Other researchers suggested the comparison of all possible pairs of classes [13], resulting in an N_c (N_c -1)/2 set of binary problems.

In this approach, the problem is transformed in n binary classification sub problems, where n is the error correcting output code length n ε $\{N_c,\ldots,\infty\}$. Then, the output of all classifiers must be combined—traditionally using Hamming distance. In the literature, one can find several powerful binary classifiers. However, when one needs to deal with multiclass classification problems, many learning techniques fail to manage this information. Instead, it is common to construct the classifiers to distinguish between just two classes and to combine them in some way. In this sense, Error-Correcting Output Codes (ECOCs) were born as a general framework to combine binary problems to address the multiclass problem. The strategy was introduced by Dietterich and Bakiri in 1995. Based on the error correcting principles and because of its ability to correct the bias and variance errors of the base classifiers, ECOC has been successfully applied to a wide range of applications such as face recognition, face verification, text recognition, and manuscript digit classification.

It was when Allwein et al. [2] introduced a third symbol (the zero symbols) in the coding process when the coding step received special attention. This symbol increases the number of partitions of classes to be considered in a ternary ECOC framework by allowing some classes to be ignored. Then the Ternary coding Matrix becomes M belongs to $\{-1, +1, 0\}^{N\times n}$.In this case, the symbol zero means that a particular class is not considered by a certain binary classifier. The rest of the paper is organized as follows: section-II describes the basic framework of Error Correcting Output Codes, section-III covers results and comparison with different coding methods on various datasets, section-IV includes conclusion.

II. THE BASIC FRAMEWORK OF ERROR CORRECTING OUTPUT CODES

1. Given a set of N_c classes, the basis of the ECOC framework consists of designing a codeword for each of the classes.

2. These code words encode the membership information of each class for a given binary problem.

3. Arranging the code words as rows of a matrix, we obtain a “coding matrix M_c”, where $M_c \in \{-1, 0, +1\}^{Nc\times n}$, being n the length of the code words codifying each classes.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

4. From the point of view of learning, M_c is constructed by considering n binary problems each one corresponding to a column of the matrix M_c.

5. Each of these binary problems splits the set of classes in two partitions (coded by +1 or -1 in M_c according to their class set membership or 0 if the class is not considered by the current binary problem).

6. Then at the decoding step, applying the n trained binary classifiers, a code is obtained for each data point in the test set.

7. This code is compared to the base code words of each class defined in the matrix M_c, and the data point is assigned to the class with the closest codeword.

Figure 1 shows ECOC coding design for a 4-class problem. White, black and grey positions corresponds to the symbols +1, -1 and 0, respectively. Once the four binary problems are learnt, at the decoding step a new test sample X is tested by the n classifiers. Then the new codeword x={x1… xn} is compared with the class code words {C1… C4}, classifying the new sample by the class Ci which codeword minimizes the decoding measure.

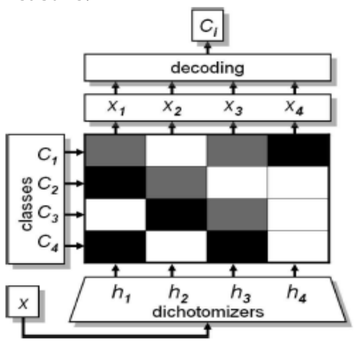

Fig. 1. ECOC Coding Design for 4-class Problem.

A. Coding Design

Here the ECOC coding design covers the state-of-the art of coding strategies, mainly divided in two main groups: problem-independent approaches, which do not take into account the distribution of the data to define the coding matrix, and the problem-dependent designs, where information of the particular domain is used to guide the coding design [1].

Problem-Independent ECOC Coding Designs.

- One-versus-all (Rifkin and Klautau, 2004): *Nc* dichotomizers are learnt for *Nc* classes, where each one splits one class from the rest of classes.
- One-versus-one(Nilsson,1965):$n=Nc(Nc-1)/2$ dichotomizers are learnt for *Nc* classes, splitting each possible pair of classes.
- Dense Random (Allwein et al., 2002): n=10.logNc dichotomizers are suggested to be learnt for *Nc* classes, where $P(-1) = 1-P(+1)$, being $P(-1)$ and $P(+1)$ the probability of the symbols -1 and +1 to appear, respectively. Then, from a set of defined random matrices, the one which maximizes a decoding measure among all possible rows of *Mc* is selected.
- Sparse Random (Escalera et al., 2009): n = 15.logNc dichotomizers are suggested to be learnt for *Nc* classes, where $P(0) = 1-P(-1) -P(+1)$, defining a set of random matrices *Mc* and selecting the one which maximizes a decoding measure among all possible rows of *Mc*.

Problem-Dependent ECOC Coding Designs

- ECOC (Pujol et al., 2006): problem-dependent design that uses n = Nc−1 dichotomizers. The partitions of the problem are learnt by means of a binary tree structure using exhaustive search or a SFFS criterion. Finally, each internal node of the tree is embedded as a column in Mc.
- Forest-ECOC (Escalera et al., 2007): problem-dependent design that uses $n = (Nc-1) \cdot T$ dichotomizers, where T stands for the number of binary tree structures to be embedded. This approach extends the variability of the classifiers of the DECOC design by including extra dichotomizers.
- ECOC-ONE (Pujol et al., 2008): problem-dependent design that uses $n = 2 \cdot Nc$ suggested dichotomizers. A validation sub-set is used to extend any initial matrix *Mc* and to increase its generalization by including new dichotomizers that focus on difficult to split classes.

In this paper we have analyzed the performance by using a problem-dependent Discriminate ECOC (DECOC) Design. The goal of this work is to find a compact—in terms of codeword length—matrix M with high discriminative power. DECOC is born as an answer to three demands: First, a heuristic for the design of the ECOC matrix, second, the search for high-performance classification using the minimum number of classifiers, and, third, a tool to describe the classification domain in terms of class dependencies. The proposed method renders each column of the output code matrix to the problem of finding the binary partition that divides the whole set of classes so that the discriminability between both sets is maximum. The criterion used for achieving this goal is based on the mutual information between the feature data and its class label. Since the problem is defined as a discrete optimization process, we propose using the floating search method as a suboptimal search procedure for finding the partition that maximizes the mutual information. The whole ECOC matrix is created with the aid of an intermediate step formulated as a binary tree. With this formulation, we ensure that we decompose the multiclass problem into Nc - 1 binary problems.

B. Decoding Design

- decoding: $HD(x,y_i)=\sum_{j=1}^{n}(1-sign(x^j y_i^j))/2$, being x a test codeword and yi a codeword from M_c corresponding to class C_i.
- Inverse Hamming decoding: $IHD(x, yi) = \max(\Delta^{-1}D^T)$, where $\Delta(i_1, i_2) = HD(y_{i1}, y_{i2})$, and D is the vector of Hamming decoding values of the test codeword x for each of the base code words y_i.
- Euclidean decoding: $ED(x,y_i)=\sqrt{\sum_{j=1}^{n}(x_j-y_j)}^2$
- Attenuated Euclidean decoding:

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

$$AED(x, y_i) = \sqrt{\sum_{j=1}^{n} |y_i^j|\,|x^j|\,(x^j - y_i^j)^2}$$

- Loss-based decoding:

  $LB(\rho, v_i) = \sum_{j=1}^{n} L(y_i^j . f^j(\rho))$,

  Where ρ is a test sample, L is a loss function, and f is a real-valued function $f: R^n \rightarrow R$.
- Probabilistic-based decoding:

  $PD(y_i, x) = -\log(\pi_j \in [1,\ldots,n] : M_c(i,j) \neq 0$

  $P(x^j = M_c(i,j) \mid f^j) + K)$,

  Where K is a constant factor that collects the probability mass dispersed on the invalid codes, and the probability $P(x^j = Mc\ (i, j) \mid f^j)$ is estimated by means of

  $P(x^j = y_i^j \mid f^j) = 1/1 + e^{y_i^j (v^j f^j + w^j)}$,

  Where vectors υ and ω are obtained by solving an Optimization problem (Passerini et al., 2004).
- Laplacian decoding:

  $LAP(x,\ y_i) = \dfrac{\alpha_i + 1}{\alpha_i + \beta_i + K}$, where α_i is the number of matched positions between x and y_i, β_i is the number of miss-matches without considering the positions coded by 0, and K is an integer value that codifies the number of classes considered by the classifier.
- Pessimistic β-Density Distribution decoding:

  Accuracy $s_i : \int_{vi-si}^{vi} \psi i(v, \alpha_i, \beta_i) dv = \frac{1}{3}$, where

  $\psi_i(v, \alpha_i, \beta_i) = \frac{1}{K} v^{\alpha i} (1-v)^{\beta i}$, ψ_i is the β-Density Distribution between a codeword x and a class codeword y_i for class c_i, and $v \in R : [0,1]$.
- Loss-Weighted decoding:

  $LW(\rho, i) = \sum_{j=1}^{n} M_W(i,j) L(y_i^j . f(\rho, j))$,

  Where

  $M_w(i,j) = H(i,j) / \sum_{j=1}^{n} H(i,j)$,

  $H(i,j) = 1/m_i \sum_{k=1}^{m_i} \varphi(h^j(\rho_k^i), i, j)$,

  $$\varphi(x^j, i, j) = \begin{cases} -1, x^j = y_i^j \\ 0, otherwise \end{cases}$$

  m_i is the number of training samples from class C_i, and ρ_k^i is the kth sample from class C_i.

C. Discussion of Some ECOC coding methods

- One-Versus-All strategy

  The most well-known binary coding strategies are the one-versus-all strategy, where each class is discriminated against the rest of classes.

  In Figure2 (a), the one-versus-all ECOC design for a four-class problem is shown. The white regions of the coding matrix M correspond to the positions coded by 1 and the black regions to -1. Thus, the code word for class C1 is {1,-1,-1,-1}. Each column i of the coding matrix codifies a binary problem learned by its corresponding dichotomizer h_i. For instance, dichotomizer h_1 learns C1 against classes C2, C3, and C4, dichotomizer h_2 learns C2 against classes C1, C3, and C4, etc.
- The Dense Random Strategy

  The dense random strategy, where a random matrix M is generated, is maximizing the rows and columns separability in terms of the Hamming distance.

An example of a dense random matrix for a four class problem is shown in Figure 2(c).

- One-Versus-One and Random Sparse Strategy

It was when Allwein et al. Introduced a third symbol (the zero symbols) in the coding process when the coding step received special attention. This symbol increases the number of partitions of classes to be considered in a ternary ECOC framework by allowing some classes to be ignored. Then, the ternary coding matrix becomes $M \in \{-1,0,1\}^{N \times n}$. In this case, the symbol zero means that a particular class is not considered by a certain binary classifier. Thanks to this, strategies such as one-versus-one and random sparse coding can be formulated in the framework. Figure 2(b) shows the one-versus-one ECOC configuration for a four-class problem. In this case, the gray positions correspond to the zero symbol. A possible sparse random matrix for a four-class problem is shown in Figure 2(d).

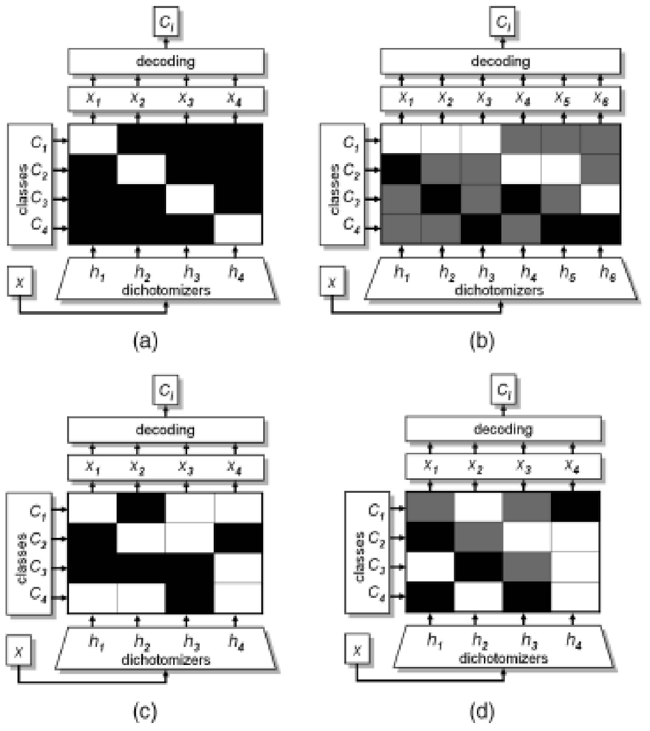

Fig .2 (a) one-versus-all, (b) one-versus-one, (c) dense random, and (d) sparse random ECOC designs.

D. An Algorithm for Problem Dependent ECOC based on DECOC Method

Create the trivial partition $\{\wp_0^1, \wp_0^2\}$ of the set of classes $\{C_i\}$:

$\{\wp_0^1, \wp_0^2\} = \{\{\emptyset\}, \{C_1, C_2, \ldots, C_{N_c}\}\}$

$L_0 = \{\wp_0^2\}$

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

$k = 1$

Step 1. S_k is the first element of L_{k-1}

$L'_k = L_{k-1} \setminus \{ S_k \}$

Step 2. Find the optimal binary partition BP (S_k):

$\{ {}_{k}^{-1}, {}_{k}^{-2} \} = \underset{BP(S_k)}{argmax} (I(\mathbf{x},d(BP(S_k))))$

Where I is the mutual information criterion, x is the random variable associated to the features and d is the discrete random variable od the dichotomy labels. [a]

Step 3. $L_k = \{ L'_k \cup {}_{k}^{-i} \}$ if $|{}_{k}^{-i}| > 1$ $\forall_i \in \{1,2\}$

Step 4. If $|L'_k| \neq 0$

K=k+1 go to Step-1.

[a Use SFFS algorithm as the maximization procedure and MI of to estimate I.]

Table-I Terms Used

C – set of classes
I – Mutual Information
i – index
J – data matrix of the problem
L – set of class labels
ξ - error function
J_i – data of class C_i
h_j – jth dichotomizer
M – coding matrix
m – number of object instances
n – number of dischotomizers
P – total number of experiments
N_c – number of classes
$\wp_i$ – { $\wp_{i1}$, $\wp_{i2}$ }, set of positive & negative sub-set of the ith binary problem

The goal of this work is to find a compact—in terms of codeword length—matrix M with high discriminative power. The general algorithm can be described as follows:

General procedure

- Create the Column Code Binary Tree—recursively, find the most discriminant binary partition of each parent node class set { $\wp k^1$,$\wp k^2$ }—using floating search with fast quadratic mutual information criterion.
  - Assign to the column k of matrix M the code obtained by the partition { $\wp k^1$,$\wp k^2$ }.

The first step is the creation of the Column Code Binary Tree (CCBT), where each node of the tree defines a partition of the classes. The partition at each node must satisfy the condition of being highly separable in terms of discrimination. This division is obtained as the result of the maximization of the quadratic mutual information between the data x and the labels created for such partition d. The algorithm used for the discrete maximization is the floating search method, which will be introduced in the next section. Above the basic algorithm for creating the code column binary tree (CCBT) has been shown. In the algorithm, d is a discrete random variable, so that, given a binary partition { $\wp k^1$,$\wp k^2$}of the set Sk, { $\wp k^1$,$\wp$k2 }=BP(Sk) , d is defined in the following terms,

$$d = d(x, BP(Sk)) = \begin{cases} 1 \; if \; x \in C_i | \; C_i \in \wp_k^1 \\ -1 \; if \; x \in C_i | C_i \in \wp_k^2 \end{cases}$$

The tree must be seen as a means to finding the codewords. The second step is the process of filling the ECOC matrix. The final matrix M is composed by the codes obtained at each node—except for the leaves. Those codes are placed as columns in the coding matrix, M(. , i). In order to create each column code, we use the relationship between a parent node and its children. Therefore, given a certain class C_r and the class set associated to node $k : \{\wp_k^1 \cup \wp_k^2\}$ (where $\wp_k^1$ and $\wp_k^2$ are the sets of classes for each one of the children of the node k, respectively), matrix M is filled as follows:

$$M(r,i) = \begin{cases} 0 \; if \; C_r \notin \wp_i \\ +1 \; if \; C_r \in \wp_k^1 \\ -1 \; if \; C_r \in \wp_k^2 \end{cases}$$

Note that the number of columns—n—coincides with the number of internal nodes. It is easy to see that, in any binary tree, the number of internal nodes is N_c - 1 given that the number of leaves is N_c. Therefore, by means of the CCBT, we can assure that the codeword will have length N_c - 1. Figure 3 shows an example of a CCBT for eight classes. On the right side of the figure, we show the resulting discriminant ECOC matrix. The white squares are +1, black squares are -1, and gray squares have 0 value. Observe, for instance, that column N5 corresponds to the partition $\wp_5^2 = \{c5, c6\}$ and $\wp_5^2 = \{c2\}$. On the other hand, if we look at the rows of the matrix, the codeword associated to class 6 (c6) is {+1,0,-1,0,-1,0,+1}.From a more general point of view, the creation of the ECOC matrix is only one of the parts involved in the multiclass classification technique. The other two remaining parts to be defined are the dichotomy learning technique and the decoding strategy. Here chosen classifier is AdaBoost for each dichotomy. A maximization process is needed to obtain the division of the classes in two sets. Although looking for the best partition set requires of an exhaustive search among all possible partitions, due to the impracticability of this endeavor a suboptimal strategy must be used. The strategy chosen is the floating search method. The following subsection details this method that allows the problem to be computationally feasible.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

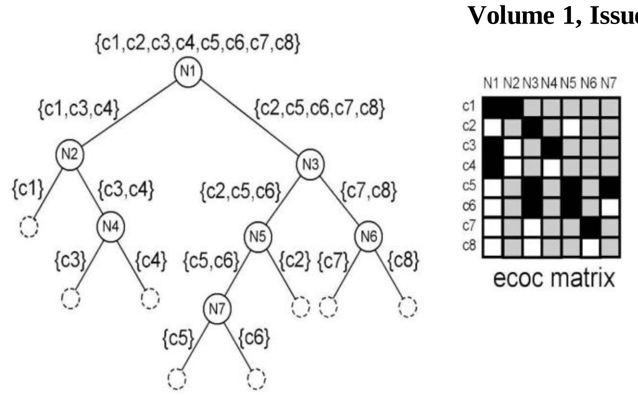

Fig. 3Example of conversion from binary tree to the ECOC matrix.

The Floating search method [9] was born as a suboptimal sequential search method for alleviating the prohibitive computation cost of exhaustive search methods in feature selection. Furthermore, these methods allowed the search criterion to be non monotonic, thus solving the main constraint of many sequential methods. Floating search methods can be described as a dynamically changing number of forward steps and backward steps as long as the resulting subsets are better than the previously evaluated ones at that level. . In this sense, this method avoids nesting effects that are typical of sequential forward and backward selection while being equally step-optimal since the best (worst) item is always added (discarded) to (from) the set. Our goal is to maximize the mutual information between the data in the sets and the class labels created for each subset.

Fast Quadratic Mutual Information

Mutual information (MI) is a well-known criterion to compute the amount of information that one random variable tells about another one. In classification theory, this measure has been shown to be optimal in terms of class separation [6], [4], allowing to take into account high-order statistics. MI also bounds the optimal Bayes error rate. Finally, to decode problem-dependent design of Discriminate ECOC, we take advantage of the recently proposed Pessimistic β – density distribution decoding .Pessimistic Beta Density Distribution Decoding (β-DEN), is based on estimating the probability density functions between code words.

III. RESULT AND COMPARISON

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. In this section the proposed method is compared with the different ECOC Coding Methods [17] which are already available. For Comparison One Vs One, One Vs All and Random (Dense Random/Sparse Random) Coding strategies have been used with DECOC Coding strategy. Table-II shows the characteristics of the datasets which have been used to perform experiments.

Table-II Dataset Characteristics

Problem	#Attributes	#classes	Attribute Types	#instances
Iris	4	3	Numeric	150
Vehicle	18	4	Numeric	15000
Segment	19	7	Numeric	1500
Two Patterns	128	4	Numeric	5000

The accuracy comparison and results are shown in Table III and Figure 4, Table IV and Figure 5 show the TP_rate, FP_rate, Precision and F-Measure calculations for all the datasets with various coding methods which indicates that by using DECOC with Pessimistic β - Density Distribution Decoding for Multiclass classification, maximum numbers of instances of data have been correctly classified. Table Iv is shown in Appendix.

Table-III Classification Accuracy on the datasets

Dataset	Accuracy			
	OneVsOne	OneVsAll	Random	DECOC
Vehicle	74.1467	92.7333	97.5467	99.0733
Segment	68.3333	93.4000	85.0667	96.9333
Iris	96.6667	96.6667	98.6667	99.3333
Two Patterns	61	54.3800	65.3400	55.0600

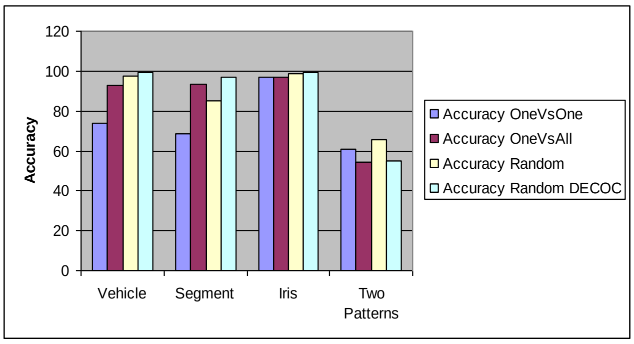

Fig.4. Accuracy chart of various coding methods

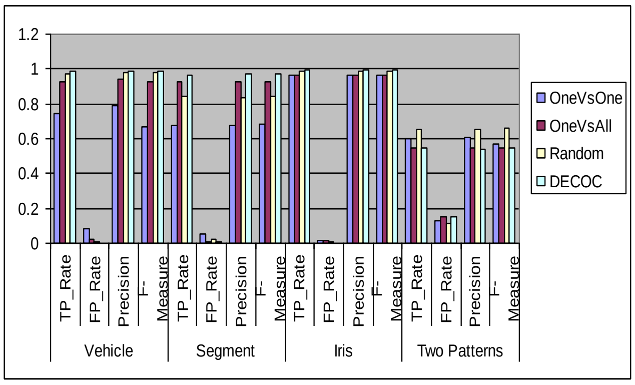

Fig. 5. TP_Rate, FP_Rate, Precision and F-Measure Chart

IV. CONCLUSION

Discriminate ECOC design with β-density Decoding is a very promising alternative to other ECOC methods, frequently outperforming most of them. This ECOC strategy is a novel way to model complex multiclass classification problems. The method is based on embedding dichotomizers in a problem-dependent ECOC design. The results are even more significant when one has a sufficiently large training size. The zero symbol produces serious inconsistencies when using the traditional decoding strategies so pessimistic β-density distribution decoding is used which gives significant performance improvement. By performing the various experiments on the selected datasets, it is observed that accuracy can be increased by 5 to 6% with compare to other strategies.

REFERENCES

[1] Oriol Pujol, Petia Radeva, Member, IEEE, and Jordi Vitria—Discriminant ECOC: A Heuristic Method for Application Dependent Design of Error Correcting Output Codes, IEEE Transactions on pattern analysis and machine intelligence , Vol. 28, No. 6, June 2006.

[2] E. Allwein, R. Schapire, and Y. Singer, —Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers, J. Machine Learning Research, vol. 1, pp 113 141, 2002.

[3] E.L Allwein, R.E Shapire, and Y. Singer, —Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers, J. Machine Learning Research, vol. 1, pp 113-141, 2000.

[4] J. Principe, D. Xu, and J. Fisher III, —Information Theoretic Learning, Unsupervised Adaptive Filtering, Wiley, 2000.

[5] K. Crammer and Y. Singer. On the learn ability and design of output codes for multiclass problems. Machine Learning, 47(2-3):201–233, 2002.

[6] K. Torkkola, —Feature Extraction by Non-Parametric Mutual Information Maximization,‖ J. Machine Learning Research, vol. 3, pp. 1415-1438, 2003.

[7] N.J. Nilsson, Learning Machines. McGraw-Hill, 1965.

[8] O. Pujol and P. Radeva. Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. PAMI, 28(6):1007–1012, 2006.

[9] P. Pudil, F. Ferri, J. Novovicova´, and J. Kittler, —Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions,‖ Proc. Int'l Conf. Pattern Recognition, pp. 279-283, 1994.

[10] R. Ghaderi and T.Windeatt. Circular ecoc: A theoretical and experimental analysis. In ICPR, pages 2203–2206, 2000.

[11] R. Schapir and Y. Singer. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research, 2:263–286, 1995.

[12] T.G. Dietterich and G. Bakiri, —Solving Multiclass Learning Problems via Error-Correcting Output Codes‖ J. Artificial Intelligence Research, vol. 2, pp. 263-286, 1995.

[13] T. Hastie and R. Tibshirani, —Classification by Pair wise Coupling,‖ Annals of Statistics, vol. 26, no. 2, pp. 451- 471, 1998.

[14] V.N. Vapnik, the Nature of Statistical Learning Theory, Springer 1995.

[15] Y. Freund and R.E. Shapire, —A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,‖ J. Computer and System Sciences, vol. 55, no. 1, pp. 119-139, 1997.

[16] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In NIPS, 2008.

[17] Sergio Escalera ,Oriol Pujol ,Petia Radeva ,Error-Correcting Ouput Codes Library Journal of Machine Learning Research 11 (2010) 661-664 Submitted 8/09; Revised 1/10; Published 2/10.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 4, April 2012

APPENDIX

Table-IV. TP_Rate, FP_Rate, Precision and F-Measure Calculation for All the Datasets.

Dataset		OneVsOne	OneVsAll	Random	DECOC
Vehicle	TP_Rate	0.7415	0.9273	0.9755	0.9907
	FP_Rate	0.0862	0.0242	0.0082	0.0031
	Precision	0.7875	0.9381	0.9776	0.9909
	F-Measure	0.6719	0.9265	0.9765	0.9907
Segment	TP_Rate	0.6794	0.9253	0.8457	0.9682
	FP_Rate	0.0524	0.0061	0.0248	0.0051
	Precision	0.6770	0.9251	0.8330	0.9731
	F-Measure	0.6798	0.9257	0.8451	0.9689
Iris	TP_Rate	0.9667	0.9667	0.9867	0.9933
	FP_Rate	0.0167	0.0167	0.0067	0.0033
	Precision	0.9668	0.9668	0.9872	0.9935
	F-Measure	0.9667	0.9667	0.9867	0.9933
Two Patterns	TP_Rate	0.6029	0.5478	0.6536	0.5473
	FP_Rate	0.1309	0.1507	0.1154	0.1510
	Precision	0.6097	0.5439	0.6553	0.5369
	F-Measure	0.5714	0.5456	0.6588	0.5437