A Comparison of Error Correcting Output Coding Methods for Multiclass Classification by Using Support Vector Machine: The Prediction of Self-Monitoring of Blood Sugar

Özge Aksehirli[a,#], Handan Ankarali[a,*], Duygu Aydin[b,#] and Davut Baltaci[c,#]

[a]*Department of Biostatistics, Faculty of Medicine, University of Düzce, Düzce, Turkey*

[b]*Department of Biostatistics, Faculty of Medicine, University of Hacettepe, Ankara, Turkey*

[c]*Department of Family Medicine, Faculty of Medicine, University of Düzce, Düzce, Turkey*

Abstract: SVMs were initially developed to perform binary classification. However, in many real-world problems, particularly pattern recocnation studies, aimed to determine the distinctive features of large number of class or group. For this reason, a number of methods to generate multiclass SVMs from binary SVMs have been proposed by researchers and this is still a continuing research topic. In this study we aimed to compare classification accuracy and computational cost of four multiclass approaches using a original and simulated data sets.

Error Corrected Output Coding (ECOC) based multiclass approaches that is used in this study creates many binary classifiers and combines their results to determine the class label of a test pixel.

As a result of the comparisons for all conditions examined in this study, it's found that the classification accuracy and computational cost of *One vs. One* multiclass approach is better than the other multiclass approaches.

In classification or pattern recognization problems, some of supervised machine learning methods or algorithms can be easily extended to multiclass problems. However, some other powerful and popular classifiers, such as AdaBoost and Support Vector Machines, 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.

Keywords: Support vector machines, Multiclass classification, Error correcting codes, Data mining, Kernel function, Bood sugar monitoring.

1. INTRODUCTION

Support vector machines (SVM) is a supervised learning kernel-based method proposed by Vapnic and introduced to solve binary-class problems or regression using structural risk minimization [1].

The SVM was initially developed to perform binary classification and its extension to multiclass problems was not straightforward. How to effectively extend it for solving multiclass classification problem is still an on-going research issue. Multiple class prediction is intrinsically more difficult than binary prediction because the classification algorithm has to learn to construct a greater number of separation boundaries or relations [2-4]. The popular methods for applying SVMs to multiclass classification problems usually decompose the multiclass problems into several two-class problems that can be addressed directly using several SVMs [5, 6]. In summary, currently there are two major approaches for extending SVM to multiclass classification: (*a*) considering all data in a single optimization. (*b*) Combining several binary SVM classifiers. Generally the first approach is called 'all-in-one' (AIO). But these methods are computationally much more expensive than solving several binary problems. The second is called 'divide-and-combine' and the main methods for divide-and-combine are Error Correcting Output Codes (ECOC) such as One-versus-All (OvA), One-versus-One (OvO), exhaustive correction codes, random correction codes etc. and tree based methods such as Directed Acyclic Graph (DAG) and binary hierarchical decision trees [7, 8]. A multiclass problem can be decomposed into a set of binary problems, and then combined to make a final multiclass prediction. The basic idea behind combining binary classifiers is to decompose the multiclass problem into a set of easier and more accessible binary problems. The main advantage in this divide-and-combine strategy is that any binary classification algorithm can be used [9-11].

The purpose of this research is to investigate the performances of multiclass SVM with the ECOC in term of classification accuracy and the computational cost using both orginal data set of patients with Diabetes and simulated data came from multivariate normal distribution.

*Address correspondence to this author at the Department of Biostatistics, Faculty of Medicine, University of Düzce, 81620, Konuralp, Düzce, Turkey Tel: +90 380 5421416 Fax: +90 380 5421302 E-mail: hankarali@yahoo.com
[#]Co-Authors E-mail: ozge_yilmaz85@hotmail.com, duyguaydin@duzce.edu.tr, davutbaltaci@hotmail.com

E-ISSN: 1929-6029/13

2. MATERIAL AND METHODS

Standard modern approaches for combining binary classifiers can be stated in terms of what is called output coding. Output coding is a general framework for solving multiclass categorization problems [12].

2.1. Error Correcting Output Coding Classifiers (ECOC)

Error Correcting Output Codes (ECOC) is the method to divide one multiclass classification problem into a certain number of subproblems of binary classifier and it has been proposed to enhance generalization ability of classifiers. Its advantage lies in requiring fewer classifiers. ECOC requires all classes to appear in each subproblem, and allows an arbitrary specification of how classes are reassigned to subproblems. The data structure used to specify how classes are reassigned to subproblems is called the coding matrix, *M*. The problem can be solved by associating each class with a row of a $k \times l$ "coding matrix" with all entries from {-1; +1} or {−1,0,+1}. Each column of the matrix represents a comparison between classes with "-1" and "+1", ignoring classes with "0". *k*: number of the classes and each row of this matrix, called a codeword, *l* is the number of classifiers or is also the number of binary classification problems to be constructed. Each class is assigned a unique binary string of length *l.* During testing an example is given to all of these *l* binary classifiers, then the outputs of these binary classifiers are combined to obtain a binary string of length *l*. This output string is compared to each of the *k* codewords, and the new example is assigned to the class whose codeword is closest, according to some distance measure such as Hamming distance which counts the number of bits that differ in two binary stings or Euclidian distance. Error correction is implemented *via* computing distance [11].

A measure of quality of an error–correcting code is the minimum Hamming distance between any pair of codewords. If the minimum Hamming distance between the pair of codewords is *m*, then the code can correct up to (*m*-1)/2. This is known as row separability. Therefore, it is desired to have well separated columns as well as rows, which can be achieved by maximizing the minimum Hamming distance among the columns of the ECOC matrix. The Hamming distance between two binary strings is maximum when they are complements of each other.

For a *k*-class problem, the number of possible columns is 3^k, since each one of the *k* entries can take a value from the set {-1; 0; +1}. But some of these columns do not correspond to binary classification problems; e.g. all zero or all one columns. In fact, all columns which do not contain at least one +1 and one -1 are useless. Therefore, the number of effective columns is $(½)(3^k - 2^{(k+1)} + 1)$. The factor 1/2 is a result of not using the columns which are complements of each other. The ECOC matrix which contains all the $(½)(3^k - 2^{(k+1)} + 1)$ columns is called the full code [13-15]. Dietterich and Bakiri [14] recommend using all possible dichotomies when the number of classes is 7 or less; when there are more classes, a random sampling of dichotomies is typically used. Various algorithms were proposed to select the best subset and studies on this subject are still continues.

A good error-correcting output code for a k-class problem should satisfy two properties:

- Row separation: Each codeword should be well-separated in Hamming decoding distance from each of the other codewords.
- Column separation: Each column should be uncorrelated with all the other column. This property is achieved if the Hamming decoding distance between a column and the rest—including their complementaries—is large.

Most of the previous work on output coding has concentrated on the problem of solving multiclass problems using *predefined* output codes used to construct the binary classifiers [10]. This paper gives the four commonly-used encodings.

2.2. Application Independent Design of Error Correcting Output Codes

2.2.1. One-vs-All Classifiers (OvA)

OvA formulation is a special case of error correcting codes with no error correcting capability, and by introducing "don't" care bits, also is pairwise formulation. This method is also called *winner-take-all* classification. For the *k*-class problems (*k*>2), *k* two-class SVM classifiers are constructed. The i^{th} SVM is trained while labeling the samples in the i^{th} class as positive examples and all the rest as negative examples. In the testing phase, a test example is presented to all *k* SVMs and is labelled according to the maximum output among the *k* classifiers [4].

That is:

$$Class = arg \max_{i=1,\ldots k} f_i(x)$$

where $f_i(x)$ is the decision function obtained from SVM or signed confidence measure of the i^{th} classifier [5]. Coding matrix used in OvA method for three group clasification problem is located in Table **1**.

Table 1: Code Matrix for a Three Group Problem in *OvA* Precedure

Classes/Classifiers	Codewords		
	f_1	f_2	f_3
C_1	+1	-1	-1
C_2	-1	+1	-1
C_3	-1	-1	+1

Decision boundaries obtained using *OvA* method for three classes as C_1, C_2 and C_3 are located In Figure **1**. In the first stage, straight line equations that are belonging to the three thin lines shown in the figure are obtained, then by combining these equations a general function of SVM that discriminated three classes and shown in bold lines is reached.

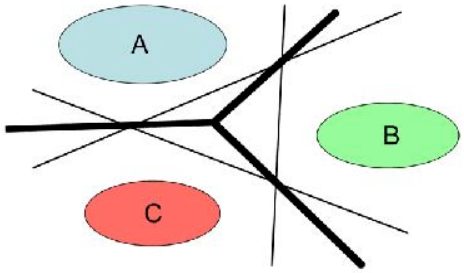

Figure 1: *OvA* result for three groups.

The disadvantage of this method is its training complexity, as the number of training samples is large. Each of the *k* classifiers is trained using all available samples. If the outputs corresponding to two or more classes are very close to each other, those points are labeled as *unclassified*, and a subjective decision may have to be made by the analyst [3, 6, 8, 16].

2.2.2. One-vs-One Classifiers (OvO)

In the *OvO* approach, ½ k(k −1) classifiers are constructed, with each classifier trained to discriminate between a class pair *i* and *j*. To combine these classifiers, the *Max Wins* algorithm is adopted [14, 17]. This can be thought of as a $k \times \frac{1}{2} k(k-1)$ matrix, where the ij^{th} entry corresponds to a classifier that discriminates between classes *i* and *j*. The codebook, in this case, is used to simply sum the entries of each row and select the row for which this sum is maximal [18].

$$Class = arg \max_{i=1,\ldots k} \left[\sum_{j=1}^{k} f_i(x) \right]$$

where f_{ij} is the signed confidence measure for the ij^{th} classifier.

Coding matrix used in *OvO* method for three group clasification problem is located in Table **2**.

Table 2: Code Matrix for a Three Group Problem in *OvO* Precedure

Classes/Classifiers	Codewords		
	f_1	f_2	f_3
C_1	+1	+1	0
C_2	-1	0	+1
C_3	0	-1	-1

Classification with *OvO* approach for three class SVM is seen in Figure **2**. 3(3-1)/2 = 3 classifiers are used in this classification and the first classifier separates the classes labelled with C_1 and C_2, the second separates the classes labelled with C_1 and C_3 and the third classifier separates C_2 and C_3 [19].

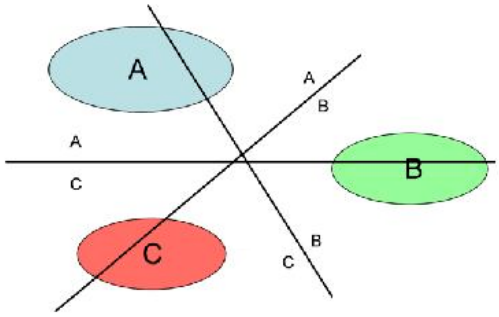

Figure 2: *OvO* result for three groups.

The lower number of samples causes smaller nonlinearity, resulting in shorter training times. The disadvantage of this method is that every test sample has to be presented to large number of classifiers *k*(*k*-1)/2. This results in slower testing, especially when the number of the classes in the problem is big [1, 5]. *OvA* and *OvO* formulations, however, unclassifiable regions exist. Instead of discrete decision functions, proposed to use continuous decision functions.

2.3. Application Dependent Design of Error Correcting Output Codes

Several methods such as Exhaustive correction codes, Randomized correction codes, Column

selection from exhaustive codes BCH codes etc are proposed to generate application dependent error correcting codes.

2.3.1. Exhaustive Correction Codes (ECC)

Detailed information of exhaustive codes for *k* class was depicted as follows;

- Codeword for first class, assigns ones to all bits;
- Codeword for second class, consists of $2^{(k-2)}$ zeros followed by $2^{(k-2)}$ - 1 ones;
- Codeword for third class, consists of $2^{(k-3)}$ zeros, followed by $2^{(k-3)}$ ones, followed by $2^{(k-3)}$ zeros, followed by $2^{(k-3)}$ - 1 ones;
- Codeword for *i.* class, alternatively runs of $2^{(k-i)}$ zeros and ones [11, 14, 20, 21]

Table **3** shows an example Exhaustive code matrix for a task with 4 classes (C_i), using 5 base classifiers (f_i).

Table 3: Code Matrix for a Four Group Problem in Exhaustive Correction Codes Precedure

Classes/Classifiers	Codewords				
	f_1	f_2	f_3	f_4	f_5
C_1	1	1	1	1	1
C_2	0	0	0	0	1
C_3	0	0	1	1	0
C_4	0	1	0	1	0

In the code matrix above, each class C_i is associated with a codeword. Each classifier f_i is trained to perform a binary classification task, that is, to distinguish the two subsets of the classes labeled with 1 and 0, respectively. During testing, a vector of scores [o_1, o_2, o_3 o_4, o_5] is generated by the 5 binary classifiers for each test sample. This vector is then compared to each codeword, and the one with the minimum distance is chosen as the hypothesis. There are two problems in this framework, the design of the code matrix and the distance measure. To design a good code matrix, a general idea is to have large row and column separation. The rows in the code matrix are the codewords (i.e., corresponding to different classes), hence the larger the distance among them, the more likely that a correct hypothesis is obtained even with errors from some classifiers during testing.

In the above code matrix, the Hamming distance m(00110, 01010) is 2. The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words. If the min Hamming distance is d, then the code can correct at least (m-1)/2 single bit errors. There are *k* classes, there will be at most 2^{k-1}-1 usable columns after removing complements and the all-zeros or all-ones column. That is in the *ECOC* approach, up to $2^{k-1}-1$ SVMs are trained, each of them aimed at separating a different combination of classes [22].

2.3.2. Randomized Correction Codes

The number of the base classifier is set and the samplings are assigned at random. Then the rows and columns are deleted which do not accord with the condition of coding matrix. *Random Codes* are randomly generated codes, where the probability distribution of the set of possible symbols {-1, 0, 1}can be specified. Allwein *et al.* [23] propose two methods for generating the matrix *M*. The first, called the *dense method*, generates codewords of length [10 $\log_2 k$]. Each element is generated randomly from {+1,−1}. A code matrix is generated by randomly generating 10,000 matrices and choosing the one with the highest minimum hamming distance among its rows. The second method, called the *sparse method*, has codewords of length [15 $\log_2 k$]. Each entry in the matrix *M* is generated randomly to get the value of 0 with probability 1/2 and {-1,+1} with probablility of 1/4 each. As before, the matrix with the highest minimum hamming distance is chosen among 10,000 randomly generated matrices [23].

2.4. Data Used in this Study

2.4.1. Orginal Data

Data used for this study are taken from the article by [24]. In this data Patients with type 2 Diabetes Mellitus (DM) admitted to the outpatient clinic of Internal Medicine Department and Family Medicine Department of Duzce University, Turkey, for diabetes care were enrolled in 2011 year. The patients were assigned into three groups according to the status of Self-monitoring of blood sugar (SMBG). Group 1 included the patients who had regularly used SMBG for at least 6 months; group 2 included the patients had irregularly used SMBG for at least 6 moths; group 3 included the patients who had never used SMBG.

Socio-demographic characteristics and clinic features of patients are given in Table **4**.

Table 4: Descriptives for Categorical and Numerical Demografic and Clinical Features in Groups

	Group 1 (n = 111) (N, %)	Group 2 (n = 133) (N, %)	Group 3 (n = 105) (N, %)
Gender			
Male	51 (45.9)	56 (42.1)	37 (35.2)
Female	60 (54.1)	77 (57.9)	68 (64.8)
Education			
Illiterate	9 (8.1)	16 (12.0)	21 (20.0)
Literate	14 (12.6)	13 (9.8)	18 (71.1)
Primary-secondary school	57 (51.4)	74 (55.6)	55 (52.4)
High school	18 (16.2)	21 (15.8)	7 (6.7)
University	13 (11.7)	9 (6.8)	4 (3.8)
Hypertension			
Yes	74 (66.7)	98 (73.7)	78 (74.3)
No	37 (33.3)	35 (26.3)	27 (25.7)
Complication			
Yes	59 (53.2)	87 (65.4)	65 (61.9)
No	52 (46.8)	46 (34.6)	40 (38.1)
Smoking Status			
Never smokers	66 (59.5)	93 (69.9)	84 (80.0)
Current smokers	28 (25.2)	15 (11.3)	16 (15.2)
Former smokers	17 (15.3)	25 (18.8)	5 (4.8)
HbA1c (%)			
< 7.5	41 (37.3)	48 (41.4)	45 (36.6)
≥ 7.5	69 (62.7)	68 (58.6)	78 (63.4)
Spot Urinary ACR			
< 30 mg/g	89 (80.9)	92 (79.3)	92 (74.8)
30-300 mg/g	17 (15.5)	19 (16.4)	24 (19.5)
≥ 300 mg/g	4 (3.6)	5 (4.3)	7 8 (5.7)
	Mean±SD	**Mean±SD**	**Mean±SD**
Age (years)	53.8 ± 9.3	54.8 ±9.3	53.5 ± 9.7
Duration of DM (years)	6.9 ± 5.1	6.5 ± 4.9	61.3 ± 4.4
BMI (kg(m^2)	31.1 ± 4.9	31.3 ± 5.7	32.1 ± 6.1
FBG (mg/dL)	157.3 ± 7.1	163.1 ± 6.7	161.3 ± 7.8
PBG (mg/dL)	218. 9 ± 11.1	247.8 ± 11.1	237.8 ± 12.6
LDL-chol (mg/dL)	114.7 ± 3.9	110.4 ± 3.7	114.3 ± 4.4
HDL-chol (mg/dL)	45.4 ±1.5	43.4 ± 1.2	44.1 ± 1.4
TG (mg/dL)	181.1 ±14.9	174.8 ± 14.6	187.8 ±16.6
T-chol (mg/dL)	189.4 ± 12.7	196.6 ± 12.5	183.2 ± 14.1
HbA1c (%)	7.3 ± 0.2	7.6 ± 0.1	7.5 ± 0.3
SBP (mm-Hg)	135.6 ± 23.3	134.7 ± 24.1	137.2 ± 22.5
DBP (mm-Hg)	86.3 ± 14.7	85.6 ± 13.9	86.4 ± 15.1
ACR (mg/g)	65.1 ± 22.4	73.7 ±21.8	78.5 ± 24.9

2.4.2. Empirical Data

Random numbers from the multivariate normal distribution were generated by writing MINITAB macro (ver. 15). Six numerical predictor variable (or attribute) were simulated and the correlation between them were produced at various levels (Table **5**). So, considered the heterogeneity of a correlation matrix. Samples, taken from three standart multivariate normal

Table 5: Empirical Data Characteristics and Corelation Matrix for Six Attributes in Simulation Study

	Correlation matrix					
	x1	**x2**	**x3**	**x4**	**x5**	**x6**
x1	1	0.9	0	0	0	0
x2		1	0	0	0	0
x3			1	0.5	0	0
x4				1	0	0.2
x5	Symetric				1	0
x6						1
	Other data characteristics					
					Sample size	
					n=30	**n=100**
Mean Vector*	1th Sample				μ = 0	μ = 0
	2th Sample				μ = 1	μ = 1
	3rd Sample				μ = 2	μ = 2
Standard Deviation Vector	1th Sample				σ =1	σ =1
	2th Sample				σ =1	σ =1
	3rd Sample				σ =1	σ =1

*written in terms of standard deviation.

populations which have different from each other about mean vectors but same standart deviations. Differences between three populations, referred to as effect size. In this simulation study we took into account two effect size (1 standard deviation and 2 standard deviation). Sample size to number of attribute ratio was determined as 5 and 17. The selection of these values, the ratio should be at least about 5 the explanations have been used. An empirical test of the utility of the observations-to-attributes ratio in factor and components analysis [25]. Accordingly, the size of the selected samples set at 30 and 100. Four different ECOC SVM procedures were applied 500 times in each sample size. Simulation study have been described with details in Table **5**.

The type of the kernel function determines the feature space into which the training data is going to be mapped and the parameter *C* controls the trade–off between margin maximization and training–error minimization. The regularization parameter *C* controls the trade–off between the complexity of the decision function and the number of training examples misclassified. As *C* increases, the number of training errors will decrease. The kernel function chosen determines the type of the decision surface, hence has a very important effect on the performance. Because frequently used kernel functions are the polynomial kernels and radial basis function kernel we used this kernels in multiclass SVM.

MINITAB for Windows (ver. 15) packet programme macro was used for random data generated from multivariate normal distribution and Weka (ver. 3.7.5) packet programme was used for ECOC SVM methods.

3. RESULTS

3.1. Original Data Results

In original data set, age, height and weight variables have normal distrubution, but AKS, TKS, LDL, HDL, TG, KOL, microalbuminuria1, ACR and HbA1C1 variables have logarithmic distribution. When logarithmic transformation is applied to the variables that have nonnormal distribution, the variables shows normal distribution. To examine the effect of the shape of distribution on the results, the results are are interpreted comparatively by applying multiclass SVM methods to both transformated data and raw data sets.

The results obtained from four different approaches by using nonlinear support vector machines for more than two classes are given in Table **6**. When these results were examined, it was seen that true rate, false rate, predictive value (precision) and area under the ROC curve values were relatively better in *OvO*

Table 6: Classification Performance Measures of Methods by Using Original Data

Performance Measures for Classes and Model with OvA approach (Run time: 0.2 seconds)			
	Group 1 (n = 111)	**Group 2 (n = 133)**	**Group 3 (n = 105)**
True Rate	%65	%44	%96
False Rate	%25	%20	%7
Predictive Value (Precision)	%55	%58	%85
Area Under ROC Curve	0.713	0.665	0.950
Accuracy	%66		
Kappa	0.49		
Mean absolute error	0.35		
Performance Measures for Classes and Model with RCC approach(Run time: 0.53 seconds)			
	Group 1 (n = 111)	**Group 2 (n = 133)**	**Group 3 (n = 105)**
True Rate	%60	%49	%96
False Rate	%22	%22	%7
Predictive Value (Precision)	%55	%58	%85
Area Under ROC Curve	0.690	0.654	0.949
Accuracy	%66		
Kappa	0.50		
Mean absolute error	0.35		
Performance Measures for Classes and Model with ECC approach (Run time: 0.2 seconds)			
	Group 1 (n = 111)	**Group 2 (n = 133)**	**Group 3 (n = 105)**
True Rate	%65	%44	%96
False Rate	%25	%20	%7
Predictive Value (Precision)	%55	%58	%85
Area Under ROC Curve	0.713	0.665	0.950
Accuracy	%66		
Kappa	0.49		
Mean absolute error	0.35		
Performance Measures for Classes and Model with OvO approach(Run time: 0.16 seconds)			
	Group 1 (n = 111)	**Group 2 (n = 133)**	**Group 3 (n = 105)**
True Rate	%60	%57	%97
False Rate	%17	%22	%7
Predictive Value (Precision)	%55	%58	%85
Area Under ROC Curve	0.810	0.675	0.950
Accuracy	%70		
Kappa	0.55		
Mean absolute error	0.29		

approach, but it can be said that the results of four approaches were similar. In addition, accuracy, kappa statistics and error values as an indicator of the model performances are also similar in the four approaches.

Table 7: Classification Performance Measures of Methods After Being Log-Transformed Some Variables have Nonnormal Distribution

Performance Measures for Classes and model with OvA approach (Run time: 0.2 seconds)			
	Group 1 (n = 111)	**Group 2 (n = 133)**	**Group 3 (n = 105)**
True Rate	%62	%52	%95
False Rate	%20	%22	%7
Predictive Value (Precision)	%59	%60	%85
Area Under ROC Curve	0.711	0.669	0.950
Accuracy	%68		
Kappa	0.52		
Mean absolute error	0.35		
Performance Measures for Classes with RCC approach (Run time: 0.44 seconds)			
	Group 1 (n = 111)	**Group 2 (n = 133)**	**Group 3 (n = 105)**
True Rate	%56	%55	%96
False Rate	%19	%24	%7
Predictive Value (Precision)	%59	%59	%85
Area Under ROC Curve	0.740	0.683	0.946
Accuracy	%68		
Kappa	0.51		
Mean absolute error	0.34		
Performance Measures for Classes with ECC approach (Run time: 0.36 seconds)			
	Group 1 (n = 111)	**Group 2 (n = 133)**	**Group 3 (n = 105)**
True Rate	%62	%52	%95
False Rate	%20	%22	%7
Predictive Value (Precision)	%59	%60	%85
Area Under ROC Curve	0.713	0.665	0.950
Accuracy	%68		
Kappa	0.52		
Mean absolute error	0.35		
Performance Measures for Classes with OvO approach (Run time: 0.17 seconds)			
	Group 1 (n = 111)	**Group 2 (n = 133)**	**Group 3 (n = 105)**
True Rate	%60	%57	%97
False Rate	%17	%22	%7
Predictive Value (Precision)	%55	%58	%85
Area Under ROC Curve	0.810	0.675	0.950
Accuracy	%70		
Kappa	0.55		
Mean absolute error	0.29		

The variables that have nonnormal distribution were converted to normal distrubution with logarithmic transformation and support vector classification was applied again for four different approach. Results of

these applications are given in Table **7**. Although the results obtained from *OvO* approach are generally better, model performances can be said to be similar to each other and also the results obtained from untransformed data. According to this result, it can be said that multi-class support vector classification isn't affected by the shape of distribution.

Classification performance measures computed with four different approaches from original data and the data applied logarithmic transformation are compared separately and it is determined that these measurements show no significant difference from each other (all p values > 0.05).

3.1. Simulation Results

Three groups with means that differ from each other by 1 and 2 standard deviation and contain 30 and 100 observations respectively (mean of first group is 0 standart deviation, second is 1 standart deviation and third is 2 standart deviations) were classified with OvA, RCC, ECC and *OvO* approach and performance measures are obtained as Table **8**.

When classification success, model error and analysis time were taken into consideration together, it was seen that for all conditions most successful classification is obtained with *OvO* approach. In addition, the results obtained from the other three approaches were very similar to each other. The performance measures of group 1 and group 3 obtained with *OvO* approach were similar for both two sample size. This is an expected result and sample size increases, classification successes that is nearest expected value are again obtained from *OvO* approach. In the other approaches, success of correct classification to third group was lower than to first group. In addition, it is determined that as the number of observations in the groups increase, classification successes for three groups increase and number of incorrectly classified observations decrease. When sample size increases from 30 to 100, even though analysis time elongates a little, there is no significant elongating according to researchers can be said. It has been observed that the most appropriate method is *OvO* again in terms of analysis time. When the model errors were examined, it was seen that the smallest errors are observed in *OvO* approach and *OvA*, *ECC* approaches follow this. As a result, when the methods were sorted from good to bad considering all of the performance measures, the rank was determined as *OvO*, *ECC*, *OvA* and *RCC*.

4. DISCUSSION

In this study various multiclass classification algorithms such as *OvO*, *OvA*, Exhaustive correction codes and Random codes were compared by their predictive accuracy, model errors and their analyse times by using original data (came from Type 2 Diabetes Mellitus patients) and simulated data.

In this study, the training time taken by *OvO* technique is less than that with the other three techniques in all conditions. This study also concludes that the highest classification accuracy is achieved with *OvO* approach by using both original and simulated data.

However, these results are valid for the conditions tried in this study. Simulation conditions when a broader, methods, would be more accurately compared. We suggest that researchers are compare these methods by using different data sets.

There is some work in the literature comparing these methods for classical datasets like iris, wine, glass, letter etc. [2, 4] *OvA*, *OvO* MaxWins (with majority voting), DAG and AIO are compared. The authors show that there is not one method that performs best for every dataset but that *OvO* MaxWins and *DAG* perform better with large number of classes. *OvO* MaxWins, *OvA*, DAG and Neural Networks are compared. The authors Show that the methods have comparable performance on accuracy and error rate but that *OvO* and DAG need less time for training phases [4].

OvO is considered more symmetric than the *OvA* method. Moreover, the memory required to create the kernel matrix is much smaller. However, the main disadvantage of this method is the increase in then number of classifiers as the number of class increases [8].

Experiments performed show that the *OVA* scheme is generally inferior to the other approaches, while no one approach generally outperforms the others. This suggests that the best coding strategy is problem dependent [23]. Several algorithms for constructing good ECOC matrices for multiclass classification problems have been proposed [13, 14, 26].

Pal [8] found that the training time taken by *OvO and DAG* techniques is less than that with the *OvA* strategy and the highest classification accuracy is achieved with exhaustive ECOC approach but requires

Table 8: Classification Performance Measures of Methods by Using Simulated Data

Performance Measures for Classes and Model with OvA approach (n=30 in each group)			
	Group 1	**Group 2**	**Group 3**
True Rate	%87	%50	%77
False Rate	%21	%14	%7.7
Area Under ROC Curve	0.877	0.739	0.871
Mean absolute error	0.34		
Root mean squared error	0.38		
Run Time	0,084 seconds		
Performance Measures for Classes and Model with ECC approach (n=30 in each group)			
	Group 1	**Group 2**	**Group 3**
True Rate	%87	%51	%77
False Rate	%21	%13	%7
Area Under ROC Curve	0.879	0.744	0.872
Mean absolute error	0.34		
Root mean squared error	0.38		
Run Time	0,08 seconds		
Performance Measures for Classes and Model with RCC approach (n=30 in each group)			
	Group 1	**Group 2**	**Group 3**
True Rate	%82	%56	%79
False Rate	%14	%16	%10
Area Under ROC Curve	0.868	0.721	0.866
Mean absolute error	0.33		
Root mean squared error	0.39		
Run Time	0,15 seconds		
Performance Measures for Classes and Model with OvO approach (n=30 in each group)			
	Group 1	**Group 2**	**Group 3**
True Rate	%81	%63	%81
False Rate	%9.7	%18	%8.8
Area Under ROC Curve	0.91	0.73	0.91
Mean absolute error	0.28		
Root mean squared error	0.36		
Run Time	0,08 seconds		
Performance Measures for Classes and Model with OvA approach (n=100 in each group)			
	Group 1	**Group 2**	**Group 3**
True Rate	%84	%62	%81
False Rate	%13	%15	%7
Area Under ROC Curve	0.881	0.765	0.882
Mean absolute error	0.32		
Root mean squared error	0.37		
Run Time	0,13 seconds		

(Table 8). Continued.

Performance Measures for Classes and Model with ECC approach (n=100 in each group)			
	Group 1	**Group 2**	**Group 3**
True Rate	%84	%62	%81
False Rate	%13	%15	%7
Area Under ROC Curve	0.881	0.765	0.882
Mean absolute error	0.32		
Root mean squared error	0.37		
Run Time	0,13 seconds		
Performance Measures for Classes and Model with RCC approach (n=100 in each group)			
	Group 1	**Group 2**	**Group 3**
True Rate	%82	%64	%81
False Rate	%10	%17	%8
Area Under ROC Curve	0.87	0.75	0.87
Mean absolute error	0.32		
Root mean squared error	0.38		
Run Time	0,21 seconds		
Performance Measures for Classes and Model with OvO approach (n=100 in each group)			
	Group 1	**Group 2**	**Group 3**
True Rate	%83	%66	%83
False Rate	%8	%17	%8
Area Under ROC Curve	0.92	0.75	0.92
Mean absolute error	0.27		
Root mean squared error	0.35		
Run Time	0,11 seconds		

very large training time. A comparison of accuracy achieved by exhaustive *ECOC* approach suggests no significant improvement in comparison to *OvO* approach. The main problem with the 'one against the rest' strategy is that it may produce unclassified data, and hence lower classification accuracies. Finally, results suggest the suitability of *OvO* approach for this type of data in term of classification accuracy and the computational cost [8]. The code design does not have a significant impact on performance. The combination of multiple binary SVMs *via* ECOC achieves better performance than a direct multiclass SVM [27].

For the blood cell data with polynomial kernels and for the hiragana data, ECOC support vector machines did not perform better than *OvA* support vector machines [10].

The predicting accuracy of *OvO* is the highest of all, and those of Dense Random and Sparse Random are similar. These results demonstrate that ECOC provide a better approach for improving the performance of speech recognition [15].

Despite classical approaches such as *OvO* or *OvA* partially solve the multi-class problem, these approaches lose their attractiveness when there are lots of classes. Recently, *hierarchical based classification* method, which is more effective than traditional methods, is used in case of many classes.

REFERENCES

[1] Burges C. A tutorial on support vector machines for pattern recognition. Data Min Knowldisc 1998; 2: 121-67. http://dx.doi.org/10.1023/A:1009715923555

[2] Hsu CW, Lin CJ. A comparison of methods for multiclass. IEEE T Neural Networ 2002; 13(2): 415-25. http://dx.doi.org/10.1109/72.991427

[3] Erastö P. Support vector machines - Academic Dissertation for the Degree of Licentiate of Philosophy, Helsinki: Rolf Nevanlinna Institute 2001; pp.1-70.

[4] Platt JC. Probabilistic outputs for support vector machines and comparison to regularize likelihood methods. Advances in Large Margin Classifiers 2000; 61-74.

[5] Madzarov G, Gjorgjevikj D, Chorbev I. A multi-class SVM classifier utilizing binary decision tree. Informatica-Lithuan 2009; 33(2): 225-33.

[6] Pujol O, Radeva P, Vitria J. Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes. IEEE T Pattern Anal 2006; 28(6): 1007-12. http://dx.doi.org/10.1109/TPAMI.2006.116

[7] Demirkesen C, Cherifi H. A Comparison of Multiclass SVM Methods for Real World Natural Scenes. In: Blanc-Talon J, *et al.* Eds. Advanced Concepts for Intelligent Vision Systems, 10th International Conference; 2008; France: Juan-les-Pins 2008; pp. 752-63.

[8] Pal M. Multiclass approaches for support vector machine based land cover classification. CoRR 2008; abs/0802.

[9] Duan K-B, Keerthi SS. Which is the best multiclass SVM method? An empirical study; 2005: Proceedings of the Sixth International Workshop on Multiple Classifier Systems 2005: June 13-15; Seaside, CA, US, Springer-Verlag. pp. 278-85.

[10] Kikuchi T, Abe S. Comparison between error correcting output codes and fuzzy support vector machines. Pattern Recogn Lett 2005; 26(12): 1937-45. http://dx.doi.org/10.1016/j.patrec.2005.03.014

[11] Wang Z, Xu W, Hu J, Guo J. A multiclass SVM method *via* probabilistic error-correcting output codes. 2010 International Conference on Internet Technology and Applications 2010; pp. 1-4.

[12] Dietterich TG, Bakiri G. Error-correcting output codes: A general method for improving multiclass inductive learning programs: 1991: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91); 1991: July 14-19; Anaheim, California, pp. 572-77.

[13] Crammer K, Singer Y. On the Learnability and Design of Output Codes for Multiclass Problems: 2000: Proceedings of the Thirteenth Annual Conference on Computational Learning Theory; 2000: June 28-July 1; San Francisco, CA, USA; pp. 35-46.

[14] Dietterich TG, Bakiri G. Solving multiclass learning problems *via* error-correcting output codes. J Artif Intell Res 1995; 2: 263-86.

[15] Xiao-Feng L, Xue-Ying Z, Ji-Kang D, Eds. Speech recognition based on support vector machine and error correcting output codes 2010: First International Conference on Pervasive Computing Signal Processing and Applications (PCSPA); 2010: Sep 17-19; Harbin Institute of Technology, Harbin, China, pp. 336-9. http://dx.doi.org/10.1109/PCSPA.2010.88

[16] Angulo C, Parra X, Català A. K-SVCR. A support vector machine for multi-class classification. Neurocomputing 2003; 55(1-2): 57-77. http://dx.doi.org/10.1016/S0925-2312(03)00435-1

[17] Jiang Z-Q, Fu H-G, Li L-J. Support vector machine for mechanical faults classification. J Zhejiang Univ Sci 2005; 6(5): 433-9. http://dx.doi.org/10.1631/jzus.2005.A0433

[18] Chang C-C, Lin C-J. LIBSVM: A Library for Support Vector Machines. Taipei, Taiwan. 2001, Available from: http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf and LIBSVM FAQ: http://www.csie.ntu.edu.tw/~cjlin/libsvm/faq.html

[19] Bilgisayarkavramlari [homepage on the Internet]. Seker S.E: Çok sınıflı DVM (Multiclass SVM). 2008, Available from: http://www.bilgisayarkavramlari.com/2008/12/01/cok-sinifli-dvm-multiclass-svm/ 05.10.2012.

[20] Bose RC, Ray-Chaudhuri DK. On a class of error correcting binary group codes. Inf Technol Control 1960; 3(1): 68-79. http://dx.doi.org/10.1016/S0019-9958(60)90287-4

[21] Peterson WW, Weldon EJ. Error-correcting codes. 2nd ed. MIT Press. Cambridge, Mass 1972.

[22] Übeyli DE. ECG beats classification using multiclass support vector machines with error correcting output codes. Digit Signal Process 2007; 17(3): 675-84. http://dx.doi.org/10.1016/j.dsp.2006.11.009

[23] Allwein E, Shapire R, Singer Y. Reducing multiclass to binary: A unifying approach for margin classifiers. J Mach Learn Res 2000; 1: 113-41.

[24] Baltaci D, Kutlucan A, Ozturk S, Saritas A, Celer A, Celbek G, Ankarali H. Effectiveness for self-monitoring of blood sugar on blood glucose control in Turkish patients with type 2 diabetes mellitus. Med Glas 2012; 9(2): 213-19.

[25] Arrindell WA, Ende JV. An empirical test of the utility of the observations-to-variables ratio in factor and components analysis. Appl Psych Meas 1985; 9(2): 165-78. http://dx.doi.org/10.1177/014662168500900205

[26] Masulli F, Valentini G. Effectiveness of error-correcting output coding methods in ensemble and monolithic learning machines. Pattern Anal Appl 2003; 6(4): 285-300. http://dx.doi.org/10.1007/s10044-003-195-9

[27] Liu Y. Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus. INTERSPEECH – ICSLP; 2006: Sep 17-21; Pittsburgh, Pennsylvania: ISCA. pp. 1938-41.

Received on 28-02-2013 Accepted on 05-03-2013 Published on 30-04-2013

http://dx.doi.org/10.6000/1929-6029.2013.02.02.06