On the Choice of Appropriate Combination of Classifier and Decomposition Scheme for Multiclass Imbalanced Data Classification : A Comparative Analysis

Sayantan Kumar

On the Choice of Appropriate Combination of Classifier and Decomposition Scheme for Multiclass Imbalanced Data Classification : A Comparative Analysis

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Computer Science

by

Sayantan Kumar
[Roll No: CS-1702]

under the guidance of

Dr. Swagatam Das
Associate Professor
Electronics and Communication Sciences Unit

Indian Statistical Institute
Kolkata-700108, India

July 2019

To my family and my guide

CERTIFICATE

This is to certify that the dissertation entitled **"On the Choice of Appropriate Combination of Classifier and Decomposition Scheme for Multiclass Imbalanced Data Classification : A Comparative Analysis"** submitted by **Sayantan Kumar** to Indian Statistical Institute, Kolkata,in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute and, in my opinion, has reached the standard needed for submission.

Swagatam Das
Associate Professor,
Electronics and Communication Sciences Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Acknowledgments

I would like to show my highest gratitude to my advisor, Dr. Swagatam Das, Associate Professor, Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. He has literally taught me how to do good research, and motivated me with great insights and innovative ideas.

My deepest thanks to all the teachers of Indian Statistical Institute, for their valuable suggestions and discussions which added an important dimension to my research work.

Finally, I am very much thankful to my parents and family for their everlasting supports.

Last but not the least, I would like to thank all of my friends for their help and support. I thank all those, whom I have missed out from the above list.

Sayantan Kumar
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

Classifying a multiclass data set with an imbalanced distribution of class representatives in the data set is a challenging problem which is prevalent in many real-world applications. In this study,we have made a comparative analysis of different decomposition techniques like OneVsAll(OVA), OneVsOne(OVO), Error Correcting Output Codes(ECOC), All-and-One(A&O) and One-Against-Lower-Order(OALO) to deal with the multiclass imbalance. While OVA and OVO have been used significantly in the multiclass imbalance domain, our work is the first to explore the remaining binarization approaches in this field. We have examined the performance of these decomposition methods on two types of learning : algorithmic approach and hybrid approach of both data-level and algorithmic solutions to solve the binary class imbalance classification problem. For the algorithmic approach learning we have used Hellinger Distance Decision Trees and for the hybrid method, we propose Balanced Ensemble Models (BEM) that combines both sampling and algorithm level modifications. It has been analyzed how effectively the decomposition methods when applied on our approach can counter the challenges of multiclass imbalance. A detailed experimental study, supported by statistical analysis has been carried out to determine which combination of classifier(between HDDT and our proposed ensemble method) and decomposition scheme work best to produce satisfactory classification performance on a multiclass imbalanced data set. From our research we conclude that ECOC decomposition strategy when applied on our proposed BEM outperforms all the other algorithms in dealing with multiclass imbalance problem.

Keywords: *Multiclass Imbalanced data classification, Decision tree, Hellinger distance, Balanced Ensemble Models, Binary Decomposition*

Contents

List of Algorithms

List of abbreviations

Table 1: Abbreviations of the algorithms given in the dissertation

Algorithm Name	**Description**
Calculate-Hellinger	Calculating the Hellinger Distance between two normalized frequency distribution of a feature across majority and minority classes.
HDDT	The method by which the decision tree is built using Hellinger Distance as the splitting criterion at each node.
PredictHellinger	Prediction of test data using HDDT.
HDDTOVA	One-vs-All decomposition applied to HDDT for multiclass imbalanced data.
HDDTOVO	One-vs-One decomposition applied to HDDT for multiclass imbalanced data.
HDDTECOC	Error Correcting Ouput Codes decomposition applied to HDDT for multiclass imbalanced data.
HDDTA&O	All and One (OVO + OVA) decomposition applied to HDDT for multiclass imbalanced data.
HDDTOALO	One Against Lower Order decomposition applied to HDDT for multiclass imbalanced data.
BEM	Our proposed algorithm Balanced Ensemble Models.
WeightedVoting	Weighted voting strategy for prediction of test data by BEM.
BEMOVA	One-vs-All decomposition applied to BEM for multiclass imbalanced data.
BEMOVO	One-vs-One decomposition applied to HDDT for multiclass imbalanced data.
BEMECOC	Error Correcting Ouput Codes decomposition applied to BEM for multiclass imbalanced data.
BEMA&O	All and One (OVO + OVA) decomposition applied to HDDT for multiclass imbalanced data.
BEMOALO	One Against Lower Order decomposition applied to HDDT for multiclass imbalanced data.

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter we begin with an introduction to the problem of class imbalance in classification. The next section provides a brief summary of the related work done so far in this domain. Next we give an idea about the various challenges in solving the class imbalance problem in multiclass data sets. In the final section, we present our contributions and conclude with an outline of the dissertation.

1.1 Imbalance Class Problem in Classification

Imbalanced data learning is a category of classification problem, where the number of representatives in some of the classes is very less compared to other classes. The class imbalance is one of the most challenging tasks in data classification and is prevalent in majority of real-world classification tasks. The skewed distribution of classes makes many conventional classifiers prone to high miss-classification error in predicting minority class examples. This is mainly due to the biased nature of the learning algorithm, especially if the majority class has over 90% representation of the data set.

The multiclass classification problem is a generalized form of the binary classification problem having k classes instead of two.In practice, many real-life domains have multiple classes having uneven representations of instances within each class. The next section deals with a brief summary of the work done so far in imbalanced data classification.

1.2 Related Work

A fair amount of research has been done to solve the data imbalance problem in classification. But most of the approaches deal with the binary class imbalance problem which contain only two classes [1–3]. After a brief literature review of research work done so far,the two-class class imbalance learning approaches can be categorized into four broad types as follows :

Algorithm Level Learning

These type of approaches deal with modification of the existing classification algorithms to make the learning biased towards the minority classes [4]. This is called the internal approach to solve the imbalanced data problem as it is mostly dependant on the problem and classifier without modifying the underlying data distribution [5]. A direct modification of the learning approach for a particular method is one of the most popular solutions [6].

Data Level Learning

The problem of imbalanced class classification originates from the uneven distribution of representative of each class, so many previous studies have considered the sampling method one of the easiest ways to tackle the problem [7]. Data level approaches balance the class distribution by using over-sampling and under-sampling techniques [8]. This is called the external approach. In these methods, a preprocessing step is applied to solve the class imbalance without modifying the learning method and is independent of the classification algorithm [9].

Cost Sensitive Learning

Cost sensitive learning approaches consider both data level and algorithmic level transformations [10]. The data level approaches include adding miss classification costs to individual samples and algorithmic techniques like assumption of higher miss classification cost for the minority classes [11]. A major problem of this method is that the miss classification costs are not defined manually in the data set [12].

Ensemble Level Learning

These approaches create a combination of one or more of the above mentioned strategies to create an ensemble learning solution. Many of the studies focus on create a balanced subset of data by sampling approaches and use algorithmic level modifications to ensure diversity within the pool of base learners. Some of the recent works by this approach include bagging combined with data level approach [13], randomized oversampling [14], hybrid combination of algorithmic techniques [15] and cost sensitive pruning for ensemble of decision trees [16].

Multiclass imbalanced data solutions

However,multiclass imbalance classification methods are relevant in many practical domains like text categorization [13], human activity recognition [14] and medical diagnosis [15]. In the recent years there had been a few studies on multiclass imbalance [16–19]. Adaboost.NC [18] combines boosting and over-sampling with ”multiminority” and ”multimajority” classes. DyS [17] is a neural network where the weights are updated by dynamically sampling the data during the learning procedure [19] had decomposed the multiclass imbalance problem using one-vs-one(OVO) approach and used binary ensemble learning algorithms. The next section provides an idea about the various challenges of class imbalance problem in multiclass domain.

1.3 Challenges in MultiClass Imbalanced Classification

Multiclass imbalance classification pose a few challenges which are not inherently observed in their corresponding binary class problems. They have been listed as follows :

Small number of samples

Number of samples in minority classes in imbalanced data sets is sometimes too low for a classification algorithm to learn discriminating rules to classify the minority class samples.

Overlapping between classes

If overlapping is absent between the classes,then any conventional classifier will be able to learn a better rule irrespective of the classes having imbalanced distribution.

Presence of small disjuncts

The complexity of the problem is enhanced if sub concepts are present within the concept of a minority class. This is due to the fact that the amount of representatives in the classes is not usually balanced.

Summary

Based on these difficulties, it can be concluded that multiclass imbalance can be manifested in two ways : a single majority and multiple minority classes, and a single minority and multiple majority classes. The problem becomes more serious when both the cases occur in a data set. Some important research that can be addressed are how these two cases create different challenges for a classifier and their individual effects on the classification of the majority and minority classes respectively.

1.4 Our Contributions

Cieslak et al. used Hellinger distance as a splitting criterion in decision trees. We provide an analysis on why Hellinger distance is a good choice for a splitting metric in decision trees over the usual splitting criteria like Gini Index and Information Gain and how Hellinger distance cannot be replaced by other divergences of the F-Divergence family like the Kullback–Leibler(KL) divergence and Jensen-Shanon(JS) Divergence. In this study,we have made a comparative study of the decomposition methods like OneVsAll(OVA) [21], OneVsOne(OVO) [22], Error Correcting Output Codes (ECOC) [23], OVA and OVO combined(A&O) [24] and One-Against-Lower-Order(OALO) [25]. The above techniques decompose the multiclass problem into a series of binary ensemble problems to effectively solve the multiclass. While OVA and OVO have been used significantly in the multiclass imbalance domain, our work is the first to explore the remaining binarization approaches in this field. We have examined the performance of these decomposition methods on two types of learning : algorithmic approach and hybrid approach of both data-level and algorithmic solutions to solve the binary class imbalance classification problem. For the algorithmic approach learning we have used Hellinger Distance Decision Trees and for the hybrid method, we present an ensemble method which combines both sampling and algorithm level modifications. It has been analyzed how effectively the decomposition methods when applied on our approach can counter the challenges of multiple majority and multiple minority cases. We have carried out a detailed experimental study and supported the findings by statistical analysis to determine which combination of classifier (between HDDT and our proposed ensemble method) and decomposition scheme work best to produce satisfactory classification performance on a multiclass imbalanced data set. To assert the superiority of the best performing method, we have compared it with AdaboostNC [18], a popular ensemble methods for solving the multiclass imbalance challenge.

Our contributions in this dissertation can be summarized as follows :

- Analysis of how Hellinger distance is a good splitting metric in decision trees for dealing the imbalanced data classification problem.
- Proposal of an ensemble technique for two-class imbalance problems.
- Comparative analysis of the different decomposition techniques on both HDDT and the proposed ensemble method and conclude which of the classifier-decomposition pair works best on multiclass imbalance classification problems.

1.5 Dissertation Outline

The rest of the dissertation has been organized as follows.

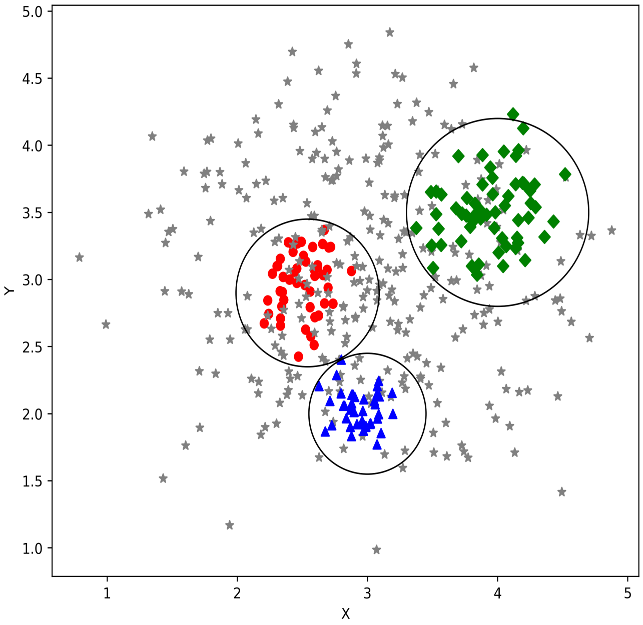

Figure 1.1: Presence of small disjuncts and class overlapping in a multiclass imbalanced data set

- In Chapter 2, we give an idea about Hellinger Distance Decision Trees(HDDT) and analyze why Hellinger distance is a good choice for a splitting metric in decision trees over the usual splitting criteria like Gini Index and Information Gain and how Hellinger distance cannot be replaced by other divergences of the F-Divergence family like the Kullback–Leibler(KL) divergence and Jensen-Shanon(JS) Divergence.
- Chapter 3 provides a brief idea about the different decomposition methods used with HDDT to classify a multiclass imbalanced data set.
- In Chapter 4, we present our proposed ensemble technique how the decomposition schemes can be applied on that deal with the multiclass imbalanced data classification challenge.
- Chapter 5 describes complete experimental framework of our study.
- In Chapter 6,we display the results and give a detailed performance analysis of our scheme.
- Chapter 7 summarizes our work and we discuss about the possible directions related to our work which can be explored in the future.

Chapter 2

Hellinger Distance Decision Trees

In this chapter, we give an idea about Hellinger Distance Decision Trees(HDDT) and analyze why Hellinger distance is a good choice for a splitting metric in decision trees how it cannot be replaced by other divergences of the F-Divergence family like the Kullback–Leibler(KL) divergence and Jensen-Shanon(JS) Divergence.

2.1 Decision Tree

Decision tree is one of the most important algorithms in classical machine learning,mainly because they are simple, efficient and easy to interpret. The most popular forms of decision trees are CART [26] and C4.5 [27], using Gini Index and Information Gain as the splitting metric respectively. In [28],the authors have recommended C4.4, a modification of C4.5 where unpruned decision trees have constructed with Laplacian smoothing at the leaves.

The most important thing that should be considered while building a decision tree is the splitting criterion. Although numerous studies have shown that C4.5 with sampling methods have performed reasonably well on imbalanced data sets, Gini index and Information Gain alone have been shown to be sensitive to the skewed distribution of representatives within the majority and minority classes. Cieslak et al. have used Hellinger distance as the splitting metric,which is member of the F-divergence family. In the next section how Hellinger distance is superior than some of the popular choices in the F-divergence family like Kulback-Leibler(KL) divergence and the Jensen-Shanon(JS) divergence.

2.2 Hellinger Distance

In this section,we have introduced the concept of Hellinger distance and we have performed a comparative analysis of Hellinger distance with KL Divergence and JS Shanon Divergence.

Definition 1. *Hellinger distance is a symmetric and non-negative measure of* *divergence or similarity between two probability distributions,related to the Bhattacharyya coefficient. Let X and Y be two continuous probability distributions with parameter γ in the measurable space (θ, γ). Hellinger distance can be defined as :*

$$d_H(X,Y) = \sqrt{\int_{\Omega} \left(\sqrt{X} - \sqrt{Y}\right)^2 d\gamma} \tag{2.1}$$

Here,the P and Q in Equation (2.1) are normalized values of feature values across the majority and minority classes. Hellinger distance quantifies the similarity measure between the two probability distributions on a finite event space. If X and Y are equal, then d_H = 0 (maximum similarity) and if $X \bigcap Y = \phi$, then d_H $=\sqrt{(2)}$ (zero similarity).

Before establishing the idea how Hellinger distance is a good splitting criterion, in the next two sections,we will prove that KL divergence and JS divergence, two of the most popular and widely used divergence metric cannot outperform Hellinger distance as the node splitting metric in decision trees.

2.3 Other divergences as splitting metric

Definition 2. *If X and Y are discrete probability distributions defined on the same probability space, then the Kullback-Leibler divergence between X and Y ican be defined as :*

$$D_{K_L}(X||Y) = \sum_i P(i) log \left(\frac{Y(i)}{X(i)}\right) \tag{2.2}$$

Theorem 1. *Kullback-Leibler Divergence cannot replace Hellinger distance as the splitting metric in decision trees.*

Proof. It can be proved that square of the Hellinger Distance is the lower bound of the Kullback-Leibler divergence(proof has been shown in the Appendix). The necessary condition for a metric to be used as the node splitting metric in a decision tree is that it must be symmetric. From the definition of KL divergence, we can conclude that it is not symmetric and so cannot replace Hellinger distance to be used as the splitting metric in decision trees. □

Definition 3. *Jensen Shanon Divergence is a symmetric form of the Kullback-Leibler Divergence and it can be expressed as :*

$$D_\gamma(X||Y) = \gamma D_{KL}(X||\gamma X + (1-\gamma)Y) + (1-\gamma) D_{KL}(Y||\gamma X + (1-\gamma)Y) \tag{2.3}$$

The value $\gamma = \frac{1}{2}$ in the above equation gives the JS Divergence :

$$D_{JS} = \frac{1}{2} D_{KL}(X||M) + \frac{1}{2} D_{KL}(Y||M) \tag{2.4}$$

where $M = \frac{(X+Y)}{2}$

Theorem 2. *Jensen-Shanon Divergence cannot replace Hellinger distance as the splitting metric in decision trees.*

Proof. Now from Equation 2.2, KL Divergence can be expressed as :

$$D_{KL}(x \parallel y) = \sum_i x(i) \log x(i) - \sum_i x(i) \log y(i) \tag{2.5}$$

$$= -H(x) - \sum_i x(i) \log y \tag{2.6}$$

H represents entropy

$$= -H(x) - \sum_i x(i) \log \frac{1}{n} \tag{2.7}$$

assuming y as uniform distribution

$$= H(x) + \log n \tag{2.8}$$

$$\tag{2.9}$$

As the KL Divergence is not symmetric, a symmetric version of it has been tested to check if it satisfies the criteria. From the above derivation, the KL divergence expression contains the Shannon entropy term. Since Hellinger distance performs better than Information Gain, it also works better than Shannon Entropy since the entropy term is present within Information Gain .The Jensen-Shanon Divergence contains the entropy term and so by intuitive argument it can agreed that performance will not be improved if Hellinger distance is replaced by JS divergence. □

2.4 Hellinger Distance Decision Trees(HDDT)

We have proved that neither of KL divergence or JS divergence can replace Hellinger distance as the node splitting criterion in decision trees. In this section we will explain how Hellinger distance can deal with the problem of class imbalance and then we will give an outline of the algorithm HDDT.

Definition 4. *In Decision trees Hellinger distance has been incorporated in decision trees as follows [20] :*

$$d_H(P_1, P_2) = \sqrt{\sum_{i=1}^{z} \left(\sqrt{\frac{|P_{1_i}|}{|P_1|}} - \sqrt{\frac{|P_{2_i}|}{|P_2|}} \right)^2} \tag{2.10}$$

where P_1 is the set of samples belonging to positive class,P_2 is the set of samples belonging to negative class. Assuming countable space,the feature values are discretized into z distinct bins.P_{1_i} represents the set of positive samples having the i-th value out of z distinct values of the chosen attribute.

Hellinger Distance to solve class imbalance

For splitting a node in a decision tree, we ideally want to select an attribute carrying the minimal similarity between the majority and minority classes. As Hellinger distance is a divergence between two normalized frequency distribution of feature values across classes, it can be redefined in the context of decision trees as the tendency of an attribute to discriminate between the feature distributions of the majority and minority class respectively. So the feature with maximum Hellinger distance is chosen for the split. Also,having no factor of prior probabilities for the classes as in Gini Index [26] and Information Gain [27], Hellinger distance is not affected to the imbalance aspect of the data set.

As a summary,the following things can be concluded about Hellinger distance which makes it a good selection for dealing with the imbalance class problem [20]

- Hellinger distance is symmetric and non-negative,which is a necessary criterion for a splitting metric in decision trees.
- No factor of class prior which makes it skew-insensitive.
- Represents maximal tendency of a feature to discriminate between the majority and minority class
- At each node split,the feature with the maximum Hellinger distance is chosen.
- Hellinger distance can outperform two other widely used divergence metric of the F-divergence family, the Kulber-Leibler Divergence and the Jensen-Shanon Divergence.

Algorithm 1 *Calculate-Hellinger*

```
Require: Training set $S$ , Feature chosen $f$
Let $d_H = -1$
$V_f \leftarrow$ set of values of feature $f$
for $i = 1$ to $length(V_f)$ do
    Let $j = V_f \setminus i$
    $current\_value = \left(\sqrt{\frac{|S_{f,i,1}|}{|S_1|}} - \sqrt{\frac{|S_{f,i,2}|}{|S_2|}}\right)^2 + \left(\sqrt{\frac{|S_{f,j,1}|}{|S_1|}} - \sqrt{\frac{|S_{f,j,2}|}{|S_2|}}\right)^2$
    if $current\_value > d_H$ then
        $d_H = current\_value$
    end if
end for
return $\sqrt{d_H}$
```

Algorithm 1 and 2 give an outline of the method by which Hellinger distance is calculated and how that calculated distance is incorporated into decision trees. In the algorithm,S_x represents the subset of the training set S which contain all the samples with class labels x. $S_{f_z=y}$ represents the training samples which have

Algorithm 2 *HDDT*

```
Require: Training Set $S$,Set of features $F$
for each feature $f \in F$ do
    $d_H(f) = Calculate - Hellinger(S, f)$
end for
$p = max(d_H)$                ▷ Feature chosen with maximum Hellinger distance
for each value $i \in p$ do
    $model = HDDT(S_{x_p=i}, F)$
end for
return $model$
```

value y for feature z. $T_{z,y,x}$ stores the samples which belong to class x and have value y for the z-th feature.

From the definition of Hellinger distance in Equation 2.1 which is based on a continuous space. Algorithm 1 considers binary splits for categorical attributes while building the decision tree. So when training samples with a continuous feature space is encountered, *Calculate-Hellinger* sorts the distinct values of the relevant feature and find all the relevant splits. The binary Hellinger distance is computed at each node split, and the maximum distance among them is taken as the output.

Hellinger Distance quantifies the separability or distance between two probability distributions. In a multiclass data set with c classes, calculating the pairwise distance between c probability distributions will be difficult. In the next section, we will show how the multiclass problem can be decomposed into binarization techniques on which Hellinger distance can be applied.

Algorithm 3 *PredictHellinger*

```
Require: model,testdata
for $i = 1$ to $length(testdata)$ do
    while $model.complete == False$ do
        if testdata($i, model.feature$) $\leq model.threshold$ then
            $model = model.leftbranch$
        else $model = model.rightbranch$
        end if
        $model.complete = True$
    end while
    $final\_pred(i) = model.label$
end for
return $final\_pred$
```

Algorithm 3 outlines the process by which prediction of test data is done by HDDT. For the sake of our implementation, HDDT model has been defined as class called HellingerNode, which has the attributes threshold, feature, leftbranch, rightbranch, complete and label.

Chapter 3

Hellinger Distance Decision Tree using Decomposition Techniques

In the previous section, we have explained why it is not possible for Hellinger Distance Decision Trees to perform well in multiclass imbalanced data classification. In this chapter, we will define each of the decomposition schemes used namely OneVsAll(OVA), OneVsOne(OVO), Error Correcting Output Codes(ECOC), Combined OVA and OVO(A&O) and One-Against-Lower-Order (OALO). Then we will discuss in brief how each of these binarization techniques are applied to Hellinger Distance Decision Tree.

Decomposition schemes : A good choice for multiclass imbalance

We have discussed in Chapter 1 that in multiclass data, the level of imbalance is much more complex than that of binary data due to small disjuncts being present in the data and classes having a lot of overlaps. Before going into the details of the decomposition methods, we list a few ways on how these methods can potentially solve the multilateral imbalance relation between the classes. These advantages serve as the motivation on why we have made a comparative analysis of these decomposition ensembles and choose the one that produces the best output.

- In some dichotomies, a few of the minority classes present in the original data set might merge or some of the original majority classes might be excluded. So,the imbalance level in a dichotomy is usually less than that of the original data set.

- The problem of small disjuncts can be removed when several classes in the actual data set are either combined together or are removed in some of the dichotomies.

3.1 OneVsAll(OVA) Decomposition

Definition 5. *One of the most simplest and natural technique to decompose a multiclass classification problem to a binary one is the OVA technique. For a C class problem, C binary classifiers are constructed corresponding to C classes. Each of the binary learners assigns one of the class as positive and all the remaining class are considered as negative. Each binary classifier returns a probability estimate for a new test sample. These probability estimates are then combined using the decision function :*

$$F(x) = \arg\max_{i=1,2,...C} f_i(x) \tag{3.1}$$

The class label with the maximum value of $F(x)$ is the predicted label of the test sample x. The OVA technique can also be generalized in the form of a Code matrix.This has been explained by an example.

Example 1. *Let C_1, C_2, C_3 be 3 classes and let D_1, D_2, D_3 be the 3 dichotomies. Here by dichotomy we mean the assignment of the classes as positive and negative in a particular binary classifier. Since C classifiers are built in OVA,C dichotomies are created.*

$$\begin{pmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \end{pmatrix}$$

The rows of the matrix represent a the configurations of a particular class C_1, C_2, C_3 in all the dichotomies (a class might be labelled positive in one dichotomy and negative in the others). Similarly, the columns of the matrix represent the configuration of each class(positive or negative) in a particular dichotomy. Algorithm 4 outlines the summary of the procedure in the form of a pseudo code.

Algorithm 4 *HDDTOVA*

```
Require: Training set $T$ , testdata, Set of features $F$
Let $C = C_1, C_2, ....C_k \leftarrow$ Set of labels corresponding to each of the $k$ classes.
$flag = 0$
for each pair of subsets $C_i \in C$ and $C_j = C \setminus C_i$ do
    $D_{ij} \leftarrow$ training data with samples of class $C_i$ and subset $C_j$
    $model\{flag\} = HDDT(D_{ij}, F)$
    $pd\{flag\} \leftarrow PredictHellinger(model, testdata)$
    $flag = flag + 1$
end for
$predicted = mode(pd)$                                  ▷ Majority voting
return $predicted$
```

3.2 OneVsOne(OVO) Decomposition

Definition 6. *The OVO is a decomposition scheme where pairwise combination of classes are created and binary classifier is trained for each of the pairs. Here* *one class is considered positive and the other negative. 66 A classifier f_{xy}, trained using the samples of classes x and y learns to distinguish between these two classes only.*

Majority Voting

For the prediction task of a test sample,a majority voting strategy is used [29]. When a classifier is trying between classes C_x, C_y, *Confidence degree* $r_{xy} \in [0, 1]$ is given by the classifier in favour of C_x to distinguish class x from class y. The confidence level in favour of C_y to correctly identify class y from class x is given by : $r_{yx} = 1 - r_{xy}$. For a 3 class problem,the confidence score matrix can be written as :

$$\begin{pmatrix} - & r_{12} & r_{13} \\ r_{21} & - & r_{23} \\ r_{31} & r_{32} & - \end{pmatrix}$$

If the confidence of a classifier to predict C_i is greater than that of C_j, then a vote is considered in favour of the class C_i. The votes received by each class is calculated and the class with the maximum votes is the assigned label for the test sample.

$$Class = \arg\max_{x=1,2,3..C} \sum_{y=1,y\neq x}^{C} s_{xy} \tag{3.2}$$

where $s_{xy} = 1$ if $r_{xy} > r_{yx}$ and 0 otherwise.

Example 2. *Like OVA, OVO can also be represented using the code matrix. An example for a* 3 *class problem has been shown :*

$$\begin{pmatrix} +1 & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & -1 \end{pmatrix}$$

Similarly as in OVA, the rows represent the class configurations in all the dichotomies and each column encodes a partition of C_1, C_2, C_3 into +1,-1 and 0 in a particular dichotomy where +1,-1 and 0 meaning positive class, negative class and class excluded from the dichotomy. According to the definition of OVO, only one class will labelled positive and one negative in a particular dichotomy. Algorithm 5 *outlines the summary of the procedure in the form of a pseudo code.*

3.3 Error-Correcting Output Codes Decomposition

ECOC is a popular method developed by [23] where a decomposition ensemble of classifiers can be learned by error correcting codes. OVO and OVA are special cases of ECOC and so, in this section we will explain the concept of error correcting codes in a bit detailed manner.

Algorithm 5 *HDDTOVO*

```
Require: Training set $T$ , testdata, Set of features $F$
Let $C = C_1, C_2, ....C_k \leftarrow$ Set of labels corresponding to each of the $k$ classes.
$flag = 0$
for $i = 1$ to $k - 1$ do
    for $j = i + 1$ to $k$ do
        $D_{ij} \leftarrow$ training data with samples of class $C_i$ and $C_j$
        $model\{flag\} = HDDT(D_{ij}, C)$
        $pd\{flag\} \leftarrow PredictHellinger(model, testdata)$
        $flag = flag + 1$
    end for
end for
$predicted = mode(pd)$                                        ▷ Majority voting
return $predicted$
```

The 3 stages of ECOC, coding,learning and decoding have been explained as follows :

Coding Stage

The coding stage decomposes a multiclass problem with c classes into n number of dichotomies.To be more precise,each of these c classes is assigned a n-bit string of -1 and $+1$ only, called a code word.These code words are generated in a way to ensure that the Hamming distance between all code words is maximized.Let $M \in \{-1, 0, +1\}^{c \times n}$ be a $c \times n$ matrix such that M_{ij} represents the j-th bit for code word of class i. For each dichotomy,the assignment of the c classes into positive and negative are denoted by the corresponding column of the code matrix M. If $M_{ij} = +1$, then the class i belongs to the positive class in the j-th classifier, and negative class for $M_{ij} = -1$. If $M_{ij} = 0$,then that particular class is excluded from the j-th dichotomy(classifier).

Learning Stage

The learning stage involves the training the dichotomy classifier by the learning rule for classification. $h_i : \chi \rightarrow \mathbb{R}$ corresponding to the i-th dichotomy.

Decoding Stage

In this final stage,each of the n classifiers(dichotomies) predict a value for a given test sample a, producing a code word for a : $Q(x) = (q_1(a), q_2(a), ...q_n(a))$.The t-th bit distance of example a to the C_k-th class is given by :

$$B_t(a, r) = d_{bit}(q_t(a), M(k, t)) \tag{3.3}$$

This quantifies the distance or separability between the output of the dichotomy classifier q_t and M(k,t), the dichotomy code of Class C_k in the code matrix M.

d_{bit} is the bit distance between two code words which is given by the formula of Hamming distance function :

$$d_{bit}(x, y) = \sum_{i=1}^{l} \frac{1 - sign(x_i, y_i)}{2} \tag{3.4}$$

where x and y are n-length vectors and sign(z) = +1 if $z > 0$ and -1 for $x < 0$.

The bit distance vector between test sample a and class C_k is given by :

$$B(a, k) = (B_1(a, k), B_2(a, k), ...B_n(a, k))^{\intercal} \tag{3.5}$$

Now, the class whose code word has the minimum magnitude of the bit distance vector with the code word predicted by the classifier is the class label that will be assigned to the test instance a.

$$\hat{y} = \arg\min_k B(a, k) \tag{3.6}$$

The error-correcting performance of the code matrix decides the performance of the ECOC decomposition method. To achieve a satisfactory performance, the ECOC matrix must satisfy the following 2 properties:

- Each row of the code matrix should be sufficiently separated from all the other rows. The Hamming distance between any two rows should be maximized as possible.
- Each of the bit functions q_i must be independent from the functions corresponding to the other bit positions $q_j, j \neq i$. To ensure that each dichotomy column has sufficient separability in terms of Hamming distance from the other columns,each dichotomy must have at least one +1 and -1 value and each column and its complementary should not be equal as it's previous columns.

Algorithm 6 outlines the summary of the procedure in the form of a pseudo code.

Length of Codewords

Fixing the length of the code words or dichotomies is an important aspect of the ECOC matrix. The maximum code word length is $2^{k-1} - 1$ for a k class problem. As the number of classes increases, the number of dichotomies formed also increases exponentially. To solve this problem, various strategies have been adopted to design the code matrix which are listed in Table 3.1.

Algorithm 6 *HDDTECOC*

```
Require: Training set $S$, Set of features $F$, testdata, type
Let there be $c$ classes and $n$ dichotomies,calculated according to type.
Generate the code matrix $M$ as explained in the coding stage.
for $t = 1, 2, 3..n$ do
    $D(t) = \phi$
    for every sample $i \in S$ do
        if $M(y_i, t) \neq 0$ then            ▷ If class is excluded from the dichotomy
            $D(t) = D(t) \cup (x_i, M(y_i, t))$
        end if
    end for
    $HDDT(D(t), F)$
end for
For a test sample $a$ whose label is to be predicted, compute the bit distance
  vector $B(a, k), \forall k = 1, ...c$ according to Equation 3.5
$\hat{y} = \arg\min_r B(a, k)$
return $\hat{y}$
```

Table 3.1: Different types of Error Correcting Code Matrix

Type	**Number of dichotomies**	**Description**
OneVsOne	$k(k-1)/2$	One class positive, others negative
OneVsAll	k	one class positive, one class negative
DenseRandom	$10log_2(k)$	For each dichotomy, all the classes are randomly assigned into positive and negative labels. Each dichotomy has at least one of each type.
BinaryComplete	$2^{k-1} - 1$	All possible binary combinations of classes are considered for partitioning the classes. For each dichotomy, all class assignments are either positive (+1) and negative (+1) with at least one of each type.
SparseRandom	$15log_2(k)$	For each dichotomy, classes are randomly assigned as positive or negative with probability 0.25 for each, and and classes are excluded from the dichotomy with probability 0.5.
TernaryComplete	$(3^k - 2^k + 1 - 1)/2$	The classes are partitioned into all possible combinations of 0, +1 and -1 with at least one positive and one negative class in each dichotomy.

3.4 All-and-One (A&O)

Definition 7. *A&O is a decomposition method where both OVA and OVO strategies are utilized. The motivation behind this method is to take advantage of both of the methods such that one can somewhat negate the drawbacks of the other. The A&O method have been previously used in multiclass classification problems but has never been used in multiclass imbalanced data classification, to the best of our knowledge. The steps of the technique have been listed as follows :*

- *The data set is trained using both OVA and OVO separately.*
- *For a test sample with unknown label, the OVA approach is used to calculate the top two output classes* (C_i, C_j).
- *The corresponding OVO classifier* f_{ij} *is used to determine the final output label.*

Motivation behind using A&O

In OVA, when there is a high proportion of miss classified instances the second best output is actually the correctly predicted label. Also the individual binary classifiers of OVO give satisfactory performances when trained individually, but usually produce incorrect results when trained in combination, as they often fail to capture the complexity of the multiclass imbalance. As discussed in the beginning of the section, the A&O method can be used to combine the effectiveness of OVO and OVA to deal with the multiclass imbalance problem. The algorithm has been explained in *HDDTA&O.*

Example 3. *Let us consider that a multiclass imbalanced data set with* 3 *classes. For the OVA,*3 *classifiers are constructed,* OVA_1, OVA_2 *and* OVA_3 *corresponding to each class. In the training data of* OVA_i *all the samples of class* C_i *are considered as positives and the samples of all the other classes as negatives. For OVO,* 3 *classifiers are built for each pair of classes,* OVO_{12}, OVO_{23} *and* OVO_{13}. *Algorithm* 7 *outlines the summary of the procedure in the form of a pseudo code.*

The total number of dichotomies required in A&O is $k(k-1)/2$ for OVA and k, for OVO for a k class problem. However,the number of dichotomies can be reduced if we only train the OVA classifiers and obtaining the best and second best classes C_i and C_j and then train the corresponding OVO classifier OVO_{ij}.

Algorithm 7 *HDDTA&O*

```
Require: Training set $T$ , testdata, Set of features $F$
Let $C = C_1, C_2, ....C_k \leftarrow$ Set of labels corresponding to each of the $k$ classes.
$flag = 0$
for each pair of subsets $C_i \in C$ and $C_j = C \setminus C_i$ do
    $D_{ij} \leftarrow$ training data with samples of class $C_i$ and subset $C_j$
    $model\_OVA\{flag\} = HDDT(D_{ij}, F)$
    $flag = flag + 1$
end for
for $i = 1$ to $k - 1$ do
    for $j = i + 1$ to $k$ do
        $T_{ij} \leftarrow$ training data with samples of class $C_i$ and $C_j$
        $model\_OVO\{i, j\} = HDDT(T_{ij}, C)$
        $pd\_OVO\{i, j\} \leftarrow PredictHellinger(model\_OVO, testdata)$
    end for
end for
Find best two classes from OVA,index1 and index2
$final\_pred = pd\_OVO(index1, index2)$
return $final\_pred$
```

3.5 One-Against-Low-Order(OALO) Decomposition

Definition 8. *In the OALO method, a hierarchy of classifiers is built based on the distribution of instances within the classes. The following steps provide a brief explanation of the decomposition technique.*

- $k - 1$ *classifiers are constructed for* k *classes,* $C_1, C_2, ...C_k$ *in decreasing order of the number of representatives in each class.*

- *The first classifier is trained considering the instances of the first class* C_1 *as positives and the samples of all the other classes as negatives. Similarly in the second classifier, the samples of the second class* C_2 *are trained as positives against the samples belonging to the higher ordered classes in the hierarchy* C_3,C_4,.. *and so on. The last classifier is trained assuming* C_{K-1} *as the positive class and* C_K *negative.*

- *Likewise the hierarchical approach for building the classifiers, a similar approach is used to predict the class of a new sample. Initially the first classifier is used to classify the sample. If the predicted label is* C_1*, then it can be concluded that the sample belongs to class* C_1 *and the process is terminated. Otherwise, the second classifier classifies the sample, and this process is repeated till the last* $K - 1$*-th classifier.*

Motivation behind using OALO

The main motivation behind using this method in multiclass imbalance problem is the grouping of the minority classes against the minority classes at each hierarchy. The hierarchical order of the classifiers plays a major role in the performance, as miss-classification errors made by the higher order classifiers is propagated to the lower order classifiers. Hence, selection of the learning algorithm is important. Algorithm 8 outlines the summary of the procedure in the form of a pseudo code.

Algorithm 8 *HDDTOALO*

```
Require: Training set $T$, testdata, Set of features $F$
Let $C_i, i \in [1, k] \leftarrow$ Set of labels corresponding to each of the $k$ classes.
$C_1, C_2, C_3....C_k \leftarrow$ class labels in decreasing order of training samples.
$flag = 0$
for $i = 1$ to $k - 1$ do
    $D_{ij} \leftarrow$ Subset of training data where Class $i$ is trained against classes $i + 1$ to $k$
    $model\{flag\} = HDDT(D_{ij}, C)$
    $pd\{flag\} \leftarrow PredictHellinger(model, testdata)$
    $flag = flag + 1$
end for
for $i = 1$ to $length(testdata)$ do
    for $j = 1$ to $flag$ do
        $final\_pred = pd(i, j)$
        if $final\_pred(i) == C_j$ then
            $break$
        end if
    end for
end for
return $final\_pred$
```

Chapter 4

Balanced Ensemble Models(BEM) to Solve MultiClass Imbalance Problem

In the last section,we discussed in detail about how Hellinger distance can be incorporated as a node splitting metric in decision trees. In the Related Work subsection of Chapter 1*, three categories of solutions have been discussed to deal with the binary class imbalanced data classification : data-level approach, algorithmic level approach and a hybrid approach. Hellinger Distance Decision tree belongs to the second category, without affecting the underlying data distribution. Numerous studies over the years have demonstrated the advantages of data level and algorithmic approaches in the class imbalance domain. We, hereby propose an ensemble approach which combines both data-level and algorithmic level operations. For the remaining part of the dissertation, we will refer our proposed Balanced Ensemble Models as* ***BEM****.*

4.1 Proposed Method

Our algorithm has been designed as an ensemble of models, where both data level and algorithmic modifications have been applied on a two-class imbalanced data set. The entire method can be summarized in 3 steps as follows :

- Each component of the proposed ensemble is a subset of the original data set having balanced proportion of the two classes. The subsets are created by including all the instances belonging to the minority class and an equal number of instances sampled randomly from the majority class without replacement.

- C4.5 learning algorithm is then used for training all the subsets. As each subset is balanced, C4.5 has been considered as a reliable learner to give good performance.

- After the training phase, the output of all the models are aggregated by a weighting voting technique. Each model is assigned a different weight based on their performances on the remaining subsets and the model having higher weight is used to classify the test sample.

Algorithm 9 and 10 outline our method *BEM* and the weighted voting strategy respectively. If C_maj and C_min be the labels corresponding to the majority and minority class respectively, number of balanced subsets that will be formed is :

$$R = \frac{|C\_maj|}{|C\_min|} \tag{4.1}$$

For estimating the weight for each learning model, each of them is tested on all the other subsets except the one on which is trained on. In the *WeightedVoting* algorithm, the weights are initialized as follows :

$$weight = \frac{accuracy}{\sum_i^R accuracy} \tag{4.2}$$

For each test sample, the weights are calculated corresponding to each class label C_maj and C_min. The sample is assigned the class for which it has higher weight.

Algorithm 9 *BEM*

```
Require: traindata,testdata
  Let $C\_maj$ and $C\_min$ be the majority and minority class label respectively.
  $R = \frac{|C\_maj|}{|C\_min|}$
  for $i = 1$ to $R$ do
      $data(i) \leftarrow$ each subset of the training data having equal proportions of the two classes.
      $train(i) \leftarrow data(i)$
      $test(i) \leftarrow$All other subsets $data(j)$ for all $j \neq i$
      $tree(i) = C4.5(train(i))$
      $accuracy(i) \leftarrow$Performance on $test(i)$
  end for
  $pred = WeightedVoting(testdata, accuracy, R, C\_maj, C\_min)$
  return $pred$
```

4.2 Decomposition Techniques on *BEM*

In this section we describe the algorithms on how the decomposition methods OVA, OVO, ECOC, A&O and OALO can be applied on *BEM* to solve the multiclass imbalanced data classification problem. The main aim of the thesis is to perform a comparative analysis of not only the different decomposition techniques but also between an algorithmic level approach(*HDDT*) and hybrid approach (*BEM*). For

Algorithm 10 *WeightedVoting*

```
Require: testdata, Accuracy of each subset $acc$, Number of subsets $R$, $C\_maj$
    and $C\_min$
$weight = \frac{acc}{\sum acc}$
for $i = 1$ to $R$ do
    $prec = evalC4.5(testdata)$                    ▷ Prediction strategy of C4.5
    for $j = 1$ to length(testdata) do
        if $prec(j) == C\_maj$ then
            $results(j, 1) = results(j, 1) + weight(i)$
        else $results(j, 2) = results(j, 2) + weight(i)$
        end if
    end for
end for
for $j = 1$ to $length(testdata)$ do
    if $results(j, 1) > results(j, 2)$ then
        $prec(j) = C\_maj$
    else $prec(j) = C\_min$
    end if
end for
return $prec$
```

this reason, our proposed method has been designed to solve the binary class imbalance problem like HDDT to make a fair comparison between the two algorithms for each decomposition method. All the methods have been listed in algorithm form in Algorithm $11 - 15$.

Algorithm 11 *BEMOVA*

```
Require: Training set $T$,testdata
Let $C = C_1, C_2, ....C_k \leftarrow$ Set of labels corresponding to each of the $k$ classes.
$flag = 0$
for each pair $\{C_i, C_j\}, C_i \in C$ and $C_j = C \setminus C_i$ do
    $D_{ij} \leftarrow$ subset of training data with samples of class $C_i$ and subset $C_j$
    $pd\{flag\} = BEM(D_{ij}, testdata)$
    $flag = flag + 1$
end for
$predicted = mode(pd)$                              ▷ Majority voting
return $predicted$
```

Algorithm 12 *BEMOVO*

```
Require: Training set $T$,test data
Let $C = C_1, C_2, ....C_k \leftarrow$ Set of labels corresponding to each of the $k$ classes.
$flag = 0$
for $i = 1$ to $k - 1$ do
    for $j = i + 1$ to $k$ do
        $D_{ij} \leftarrow$ subset of training data with samples of class $C_i$ and $C_j$
        $pd\{flag\} = BEM(D_{ij}, testdata)$
        $flag = flag + 1$
    end for
end for
$predicted = mode(pd)$                                    ▷ Majority voting
return $predicted$
```

Algorithm 13 *BEMECOC*

```
Require: Training set T,testdata,type
Let there be $c$ classes and $n$ dichotomies,calculated according to type.
Generate the code matrix $M$.
for $t = 1, 2, 3..n$ do
    $D(t) = \phi$
    for every sample $i \in S$ do
        if $M(y_i, t) \neq 0$ then            ▷ If class is excluded from the dichotomy
            $D(t) = D(t) \cup (x_i, M(y_i, t))$
        end if
    end for
    $BEM(D(t), testdata)$
end for
For a test sample $a$ whose label is to be predicted, compute the bit distance
    vector $B(a, k), \forall k = 1, ...c$ according to Equation 3.5
$\hat{y} = \arg\min_r B(a, k)$
return $\hat{y}$
```

Algorithm 14 *BEMA&O*

```
Require: traindata,testdata
Let $C = C_1, C_2, ....C_k \leftarrow$ Set of labels corresponding to each of the $k$ classes.
$flag = 0$
for each pair $\{C_i, C_j\}, C_i \in C$ and $C_j = C \setminus C_i$ do
    $D_{ij} \leftarrow$ subset of traindata with samples of class $C_i$ and subset $C_j$
    $pd\_OVA\{flag\} = BEM(D_{ij}, testdata)$
    $flag = flag + 1$
end for
for $i = 1$ to $k - 1$ do
    for $j = i + 1$ to $k$ do
        $T_{ij} \leftarrow$ subset of traindata with samples of class $C_i$ and $C_j$
        $pd\_OVO\{i, j\} = BEM(T_{ij}, testdata)$
    end for
end for
Find best two classes from OVA,index1 and index2
$final\_pred = pd\_OVO(index1, index2)$
return $final\_pred$
```

Algorithm 15 *BEMOALO*

```
Require: Training set T,testdata
Let $C_i, i \in [1, k] \leftarrow$ Set of labels corresponding to each of the $k$ classes.
$C_1, C_2, C_3....C_k \leftarrow$ class labels in decreasing order of training samples.
$flag = 0$
for $i = 1$ to $k - 1$ do
    $D_{ij} \leftarrow$ Subset of training data having Class $i$ as positive class and claases $(i + 1)$ to $k$ as negative classes.
    $pd\{flag\} = BEM(D_{ij}, testdata)$
    $flag = flag + 1$
end for
for $i = 1$ to $length(testdata)$ do
    for $j = 1$ to $flag$ do
        $final\_pred = pd(i, j)$
        if $final\_pred(i) == C_j$ then
            $break$
        end if
    end for
end for
return $final\_pred$
```

Chapter 5

Experimental Protocols

This section describes the framework for the experimental study done in our work. The multiclass imbalanced data sets chosen for our experiments have been described in Section 5.1. *Section* 5.2 *presents the evaluation metrics used to test the performance of the algorithms. In the final section, we have explained how statistical tests have been used to make a significant comparison of the results obtained from our experimental study. A total of* 10 *algorithms have been tested in our work,the* 5 *decomposition techniques described in Section* 3*, applied to both HDDT and BEM. In order to make a fair comparison with the ensemble learning techniques dedicated to multiclass imbalanced data classification tasks, we have compared the best performing algorithm with a state-of-the-art method AdaboostNC.*

5.1 Data sets Used

In our study, datasets taken from various sources like the UCI repository, KEEL and Openml have been used to test the algorithms. The data sets have been chosen in such a way that they reflect variable levels of imbalance. Table 5.1 shows a detailed description of all the data sets. For each example, it includes the number of samples(# Exm), number of features(#Attr), the number of classes(#Class), the distribution of representatives within each class(Distribution) and the imbalance ratio(IR). For the data sets having missing feature values, we have removed them before doing our experiments.

Synthetic Data Sets

For our experiments, we have also used 3 artificially created data sets apart from the ones taken taken from some original source to check the robustness of the algorithms. The data sets were created by randomly assigning two-dimensional data points in the X-Y plane to 4 classes such that the number of representatives within each class is imbalanced. In Table 5.1, $D16$, $D17$ and $D18$ are the 3 synthetic data sets having various levels of imbalance. While $D16$ and $D18$ have the challenge of multiple minority classes and single majority class challenge, $D17$

Table 5.1: Description of the data sets used in the study

ID	Dataset	# Exm	#Attr	#Class	Distribution	IR
D1	Abalone19	4174	8	2	4142,32	130
D2	Balance	625	4	3	49,288,288	5.9
D3	Contraceptive	1473	9	3	629,333,511	1.9
D4	Dermatology	362	34	6	111,60,71,48,52,20	5.5
D5	Ecoli	344	7	8	143,77,52,35,20,5,6,6	29
D6	Glass7	214	9	7	70,76,17,13,9,29	8.5
D7	Hayes-roth	132	4	3	51,51,30	1.7
D8	Led7digit	500	7	8	37,51,57,52,52,47,57,53	1.5
D9	New-thyroid	215	5	3	150,30,35	5
D10	Pageblocks	551	10	5	492,33,6,8,12	82
D11	Satimage	6435	36	6	1533,703,1358,626,707,1508	2.5
D12	Thyroid	720	21	3	17,37,366	39.2
D13	Wine	178	13	3	59,71,48	1.5
D14	Winequality-red	1599	11	6	10,53,681,638,199,18	68.1
D15	Yeast	1484	8	9	244,429,463,44,51,163,35,30,25	18.6
D16	Rand_Imbalance	450	2	4	50,300,40,60	6
D17	Rand_Imbalance_1	2618	2	4	1200,50,1000,368	24
D18	Rand_Imbalance_2	1311	2	4	10,1200,30,71	120

displays the case of single minority class and multiple majority class.

K-fold Cross validation

We have used the stratified five-fold cross validation (SCV) technique [30] in our experiments. Each data set has been divided into five folds and each fold has 20% of the representatives of the data set. For each fold, the training data includes samples belonging to the remaining 4 folds having 80% of the instances of the data set. The current fold is the test data of the algorithm for that particular fold. We have considered five-fold SCV more suited to our experimental studies than a ten-fold SCV [31]. If we increase the number of folds, size of partitions will become smaller. This may result in the test set in some of the folds having having no representatives from some of the minority classes.

5.2 Evaluation Metrics

In classification, accuracy is the most common measure for evaluating the performance of a learning algorithm. However, a learner trained on an imbalanced class problem is mainly biased towards the classes having the major proportion of the total data. In that case, the accuracy will be high if most of the majority class samples are classified correctly but very few minority class samples are assigned their correct labels. Hence accuracy cannot be considered an appropriate performance metric in imbalanced class problems.

Some of the common performance measures used to deal with binary class imbalanced data are Precision [32] and Recall [33]. Precision is the correctly classified fraction of test points which are predicted as members of the positive class. Recall is the measure of the class-specific accuracies of the minority class. To combine the properties of these indices, different metric like Geometric Mean(GMean) [34], F-measure [35] and Area Under the Receiver Operating Characteristic Curve(AUC) [36–38] and Area Under the Recall-Precision Curve(AURPC) [39]. G-mean represents the geometric mean of the class-wise accuracies. F-measure is calculated as the harmonic mean of precision and recall. The AUC value measures the area under the Receiver Operating Characteristic (ROC) curve, which plots Recall against False Positive Rate(FPR) for different parameter settings of a classifier. Similarly AURPC measures the area under the Recall-Precision Curve obtained by varying the parameter settings of a classifier.

A direct extension of G-Means is available for multiclass classification [34]. The multiclass analog of recall is called the Average Class Specific Accuracy(ACSA) [40]. The index AUC and AURPC have been extended to the multiclass case by One-vs-All(OVA) strategy which considers each of the remaining class as negative class for a given positive class [41].

We have formally defined the metrics for multiclass classification which have been used for our experiments as follows :

Definition 9. *A confusion matrix over a test set T for a C-class problem can be defined as $M_C = [m_{ij}]_{C\times C}$, where m_{ij} is the count of the samples which have actual class label i but are predicted as member of class j. The diagonal elements m_{ii} are the samples belonging to class i and are correctly predicted.*

From Definition 1,for a multiclass classification problem with c classes and n test samples,

$$Precision = \frac{1}{C}\sum_{i=1}^{C}\left(\frac{m_{ii}}{k_i}\right) \tag{5.1}$$

$$ACSA = \frac{1}{C}\sum_{i=1}^{C}\left(\frac{m_{ii}}{n_i}\right) \tag{5.2}$$

$$FMeasure = \frac{2 \times Precision \times ACSA}{Precision + ACSA} \tag{5.3}$$

$$GMean = \left(\prod_{i=1}^{C}\frac{m_{ii}}{n_i}\right)^{\frac{1}{C}} \tag{5.4}$$

where

$$n_i = \sum_{j=1}^{C} m_{ij} \tag{5.5}$$

$$k_i = \sum_{j=1}^{C} m_{ji} \tag{5.6}$$

5.3 Statistical Tests

Statistical test is an important tool to analyze the results obtained from the experimental study. We have made an attempt to compare the outputs of the classifiers across multiple data sets and find out if there exist any significant differences between them [42]. To evaluate the significance of our experimental findings, the **Friedman Rank test** has been used. This is a non-parametric test which is first applied on a metric to provide information of any statistically significant difference between the rankings of the algorithms for that metric [43].

Average Rankings

Computing the average rankings is complementary to the statistical analysis. Here,the mean value of the ranking of the algorithms is calculated to estimate the superiority of the algorithm compared to the rest. Ranks are assigned to the classifiers as per the produced output, from best to worst. We use the **Tied Rank test** where the average ranking of an algorithm is computed by taking the mean value of the ranks across all the data-sets. For each evaluation metric, the algorithm with the minimum average ranking has the best performance.

Pairwise comparison of best method with others

After obtaining the best performing algorithm by Friedman and Tied Rank test, we proceed to perform a pairwise comparison of the best method with the other algorithms. For this we have used the **Wilcoxon Signed Rank Test** [44] is a paired two-sided non parametric test that tests the null hypothesis that two dependant samples were selected from population having the same distribution.

Chapter 6

Results and Analysis

In this section, we present our experimental findings and then perform a thorough comparative analysis of the different algorithms to check which combination of classifier and decomposition scheme works best overall for both natural and synthetic data sets. We have tabulated the results for the three most standard evaluation metrics for MultiClass Imbalance problems : G-Means, F-Measure and Average Class Specific Accuracy (ACSA).

6.1 G-Means

Table 6.1 and 6.2 presents the G-Means values of the decomposition algorithms applied to HDDT and BEM respectively on all the data sets $D1$-$D18$. The value marked in bold represents the best G-mean value for a particular data set.Table 6.3 shows the rankings of the algorithms for each of the data sets along with the average ranks for all the data sets combined. Rank 1 in bold value corresponds to the best algorithm for a data set. For proper visualization of the G-mean values,we have created a box plot and a bar plot (Figure 6.1) corresponding to Table 6.1 and 6.2. The bar plot represents the G-mean value of a particular algorithm averaged across all the 18 data sets while the box plot provides an idea about the range of G-Means values.

Analysis

1. Analyzing the values from both Table 6.1 and 6.2, we find out that **BEME-COC** outputs the best G-Means value in 11 out of 18 data sets.

2. If we compare classifier wise for all the decomposition schemes combined, our proposed method BEM **beats** HDDT in 15 out of 18 data sets, clearly demonstrating its superior performance.

3. For data sets like Abalone 19, Pageblocks, Yeast etc as well the synthetic data sets having high levels of imbalance, BEM values are significantly greater than that of HDDT.

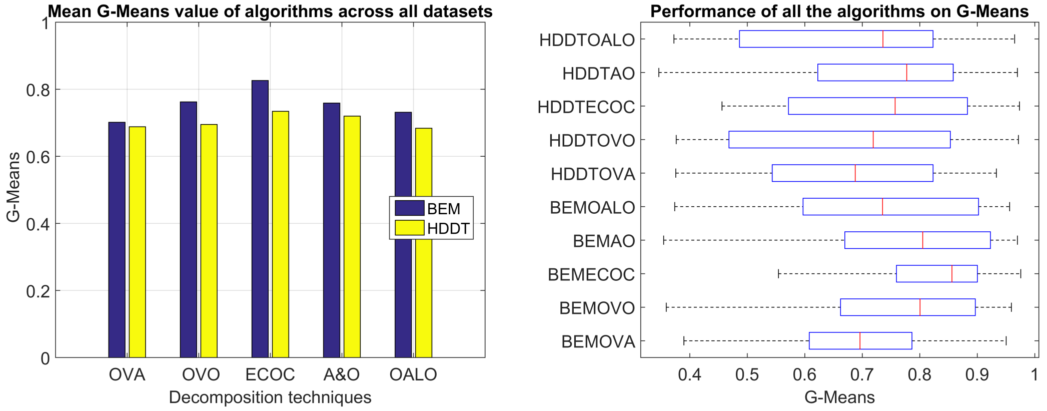

Figure 6.1: Visualization of the G-means performance of algorithms for all data sets using Bar plot and Box plot

Table 6.1: Performance of **HDDT** algorithms on **G-Means**. The bold values in each row represents the best performing algorithm for that particular data set.

Dataset	**HDDT**				
	OVA	**OVO**	**ECOC**	**A&O**	**OALO**
D1	0.47094	0.37636	0.5675	0.49726	0.40603
D2	0.56855	0.46814	0.5716	0.62262	0.43254
D3	0.53674	0.43561	0.47899	0.40745	0.48641
D4	0.90348	0.97156	0.94425	0.94722	0.93121
D5	0.70698	0.75809	0.83109	0.7857	0.73616
D6	0.68299	0.71727	0.70551	0.78168	0.73996
D7	0.8565	0.8634	0.8829	0.8581	0.84312
D8	0.67007	0.70961	0.68422	0.63917	0.67422
D9	0.8082	0.85308	0.90543	0.88435	0.82305
D10	0.82309	0.79412	0.7863	0.79847	0.73144
D11	0.81253	0.78989	0.82364	0.84253	0.81869
D12	0.93323	0.95441	0.97339	0.9698	0.91921
D13	0.9299	0.9446	0.93067	0.96241	0.96498
D14	0.3756	0.38421	0.456	0.4287	0.3724
D15	0.4056	0.45817	0.55166	0.34612	0.43379
D16	0.66651	0.68144	0.72813	0.77295	0.74905
D17	0.69261	0.72075	0.78607	0.73769	0.73569
D18	0.54336	0.62949	0.6087	0.6767	0.50926
Average	0.6882	0.6950	**0.7342**	0.7199	0.6837

4. From the rankings table (Table 6.3), ECOC has minimum average rankings for both HDDT and BEM.

5. From the box plot (Figure 6.1), it can be observed that BEMECOC has the smallest minimum and 1st quartile value and the highest median value. This justifies the consistent performance of BEMECOC across all data sets, significantly better than the others.

6. It can be observed from the bar plot (Figure 6.1) that if we compare the average values of HDDT and BEM for all the decomposition techniques, BEM always exceeds HDDT.

Thus Error Correcting Output Codes can be considered as a reliable decomposition technique that can be applied on both algorithmic and ensemble approaches to produce a satisfactory G-Means output on multiclass imbalanced data sets.

Table 6.2: Performance of **BEM** algorithms on **G-Means**. The bold values in each row represents the best performing algorithm for that particular data set.

Dataset	**BEM**				
	OVA	**OVO**	**ECOC**	**A&O**	**OALO**
D1	0.68641	0.7176	0.7594	0.6696	0.62444
D2	0.62567	0.66217	0.72682	0.75525	0.5588
D3	0.54019	0.61253	0.68695	0.45458	0.59695
D4	0.94008	0.9352	0.93338	0.92379	0.90214
D5	0.6078	0.6456	0.87344	0.7364	0.7011
D6	0.65545	0.79874	0.85424	0.80497	0.70016
D7	0.78048	0.80674	0.8314	0.8305	0.83289
D8	0.5104	0.6949	0.76198	0.6475	0.4526
D9	0.93308	0.95936	0.94075	0.9513	0.95229
D10	0.70562	0.89659	0.89954	0.92291	0.92007
D11	0.60822	0.8256	0.90015	0.80809	0.77593
D12	0.95027	0.94459	0.97561	0.96984	0.95626
D13	0.90422	0.95279	0.9484	0.94829	0.94097
D14	0.38973	0.35906	0.55424	0.35453	0.3738
D15	0.51577	0.49037	0.6581	0.4987	0.5497
D16	0.73316	0.86499	0.88337	0.86866	0.76936
D17	0.75455	0.80218	0.82432	0.80525	0.8755
D18	0.78636	0.75244	0.85762	0.70744	0.6825
Average	0.7015	0.7656	**0.8187**	0.7588	0.7286

Table 6.3: Rankings of algorithms based on **G-Means** value over all the 18 data sets.The best algorithm for each data set has been marked in bold(Rank 1).

Dataset	**Proposed Ensemble Method**					**Hellinger Distance Decision Tree**				
	OVA	**OVO**	**ECOC**	**A&O**	**OALO**	**OVA**	**OVO**	**ECOC**	**A&O**	**OALO**
D1	3	2	**1**	4	5	8	10	6	7	9
D2	4	3	2	**1**	8	7	9	6	5	10
D3	4	2	**1**	8	3	5	9	7	10	6
D4	4	5	6	8	10	9	**1**	3	2	7
D5	10	9	**1**	5	8	7	4	2	3	6
D6	10	3	**1**	2	8	9	6	7	4	5
D7	10	9	7	8	6	4	2	**1**	3	5
D8	9	3	**1**	7	10	6	2	4	8	5
D9	5	**1**	4	3	2	10	8	6	7	9
D10	10	4	3	**1**	2	5	7	8	6	9
D11	10	3	**1**	7	9	6	8	4	2	5
D12	7	8	**1**	3	5	9	6	2	4	10
D13	10	3	4	5	7	9	6	8	2	**1**
D14	4	9	**1**	10	7	6	5	2	3	8
D15	4	6	**1**	5	3	9	7	2	10	8
D16	7	3	**1**	2	5	10	9	8	4	6
D17	6	4	2	3	**1**	10	9	5	7	8
D18	2	3	**1**	4	5	9	7	8	6	10
Average	6.611	4.445	**2.111**	4.778	5.833	7.667	6.389	4.945	5.167	7.056

6.2 F-Measure

In Table 6.4 and 6.5 we present the F-Measure values of the 10 algorithms on all the data sets. The value marked in bold represents the best F-Measure value for each data set.Table 6.6 shows the rankings of the algorithms for each of data sets along with the average ranks for all the data sets combined. Rank 1 in bold value corresponds to the best algorithm for a data set. Likewise in G-Means, a box plot and a bar plot (Figure 6.1) corresponding to Table 6.4 and 6.5 has been shown.

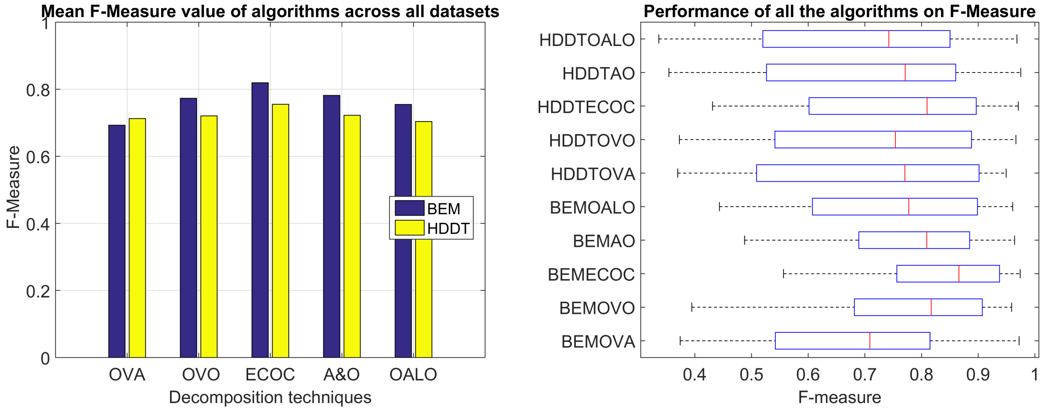

Figure 6.2: Visualization of the F-Measure performance of algorithms for all data sets using Bar plot and Box plot

Table 6.4: Performance of **HDDT** algorithms on **F-Measure**. The bold values in each row represents the best performing algorithm for that particular data set.

Dataset	**BEM**				
	OVA	**OVO**	**ECOC**	**A&O**	**OALO**
D1	0.50897	0.40969	0.44825	0.46843	0.39572
D2	0.61773	0.55415	0.69677	0.58175	0.52002
D3	0.43057	0.47429	0.47834	0.48011	0.50137
D4	0.90155	0.96652	0.95144	0.94245	0.93913
D5	0.72674	0.74703	0.79954	0.75573	0.70854
D6	0.76514	0.69168	0.80763	0.70116	0.66522
D7	0.90508	0.88823	0.89652	0.85995	0.85032
D8	0.71618	0.71677	0.78707	0.69703	0.703
D9	0.91226	0.90907	0.92047	0.91634	0.9111
D10	0.80884	0.78885	0.85538	0.81184	0.77599
D11	0.81655	0.85078	0.82561	0.86033	0.83401
D12	0.94931	0.95901	0.97072	0.97509	0.92551
D13	0.93433	0.94791	0.95212	0.95577	0.96831
D14	0.36982	0.37278	0.43127	0.35427	0.33645
D15	0.37456	0.54131	0.50336	0.47759	0.46684
D16	0.776	0.7608	0.81193	0.7867	0.7933
D17	0.83513	0.87208	0.85497	0.85085	0.83421
D18	0.47767	0.52021	0.60152	0.5265	0.54017
Average	0.712579	0.72062	**0.755162**	0.722327	0.703845

Analysis

1. It can be observed from Table 6.4 and 6.5 that **BEMECOC** has the maximum F-Measure in 9 out of 18 data sets.

2. Comparing classifier wise for all the decomposition strategies, our proposed method BEM **outperforms** HDDT in 14 out of 18 data sets.

3. The bar plot and the box plot show similar trends to that of G-Means. In terms of average F-Measure values, BEM is always superior. BEMECOC has the minimum range and maximum median value across all data sets.

4. Observing the rankings table (Table 6.6), we can claim that for both BEM and HDDT, A&O shows the 2nd best output after ECOC.

Table 6.5: Performance of **BEM** algorithms on **F-Measure**. The bold values in each row represents the best performing algorithm for that particular data set.

Dataset	BEM				
	OVA	OVO	ECOC	A&O	OALO
D1	0.75522	0.76012	0.79562	0.68927	0.62513
D2	0.60574	0.71493	0.65748	0.64192	0.60265
D3	0.47794	0.48763	0.58276	0.57434	0.60753
D4	0.85918	0.95898	0.97411	0.96421	0.93005
D5	0.49398	0.6216	0.8922	0.7997	0.7547
D6	0.71886	0.68158	0.75641	0.77225	0.75956
D7	0.81497	0.87947	0.85647	0.8521	0.84634
D8	0.6539	0.72066	0.84905	0.7694	0.5742
D9	0.97015	0.954	0.94439	0.95722	0.95409
D10	0.78188	0.90724	0.93765	0.73213	0.89865
D11	0.67935	0.83403	0.88299	0.81878	0.79539
D12	0.97217	0.94882	0.96676	0.95308	0.96084
D13	0.91561	0.95521	0.93799	0.95944	0.94551
D14	0.37414	0.39471	0.5565	0.48781	0.4681
D15	0.38463	0.54753	0.60781	0.5191	0.4434
D16	0.77238	0.87249	0.88793	0.86787	0.80535
D17	0.6987	0.87636	0.87549	0.88482	0.88902
D18	0.5421	0.80015	0.78768	0.82765	0.7250
Average	0.692828	0.773084	**0.819405**	0.781727	0.754751

Table 6.6: Rankings of algorithms based on **F-Measure** value over all the 18 data sets.The best algorithm for each data set has been marked in bold(Rank 1).

Dataset	Proposed Ensemble Method					Hellinger Distance Decision Tree				
	OVA	OVO	ECOC	A&O	OALO	OVA	OVO	ECOC	A&O	OALO
D1	3	2	1	4	5	6	9	8	7	10
D2	6	1	3	4	7	5	9	2	8	10
D3	8	5	2	3	1	10	9	7	6	4
D4	10	4	1	3	8	9	2	5	6	7
D5	10	9	1	2	5	7	6	3	4	8
D6	6	9	5	2	4	3	8	1	7	10
D7	10	4	6	7	9	1	3	2	5	8
D8	9	4	1	3	10	6	5	2	8	7
D9	1	4	5	2	3	8	10	6	7	9
D10	8	2	1	10	3	6	7	4	5	9
D11	10	4	1	7	9	8	3	6	2	5
D12	2	9	4	7	5	8	6	3	1	10
D13	10	4	8	2	7	9	6	5	3	1
D14	6	5	1	2	3	8	7	4	9	10
D15	9	2	1	4	8	10	3	5	6	7
D16	9	2	1	3	5	8	10	4	7	6
D17	10	3	4	2	1	8	5	6	7	9
D18	6	2	3	1	4	10	9	5	8	7
Average	7.389	4.167	**2.722**	3.778	5.389	7.222	6.500	4.333	5.889	7.611

As observed in the results for G-Means, in terns of F-Measure too, Error Correcting Output Codes decomposition is significantly superior than the rest. As a 2nd choice, the A&O strategy can be recommended.

6.3 Average Class Specific Accuracy (ACSA)

Table 6.7 and 6.8 displays the Average Class Specific Accuracy values of all the decomposition algorithms of HDDT and BEM respectively. The value marked in bold represents the best ACSA value for a particular data set.Table 6.9 shows the average rankings of the algorithms for all the data sets combined. Like G-Means and F-Measure, the corresponding box plot and bar plot of ACSA has been shown for better visualization of the results.

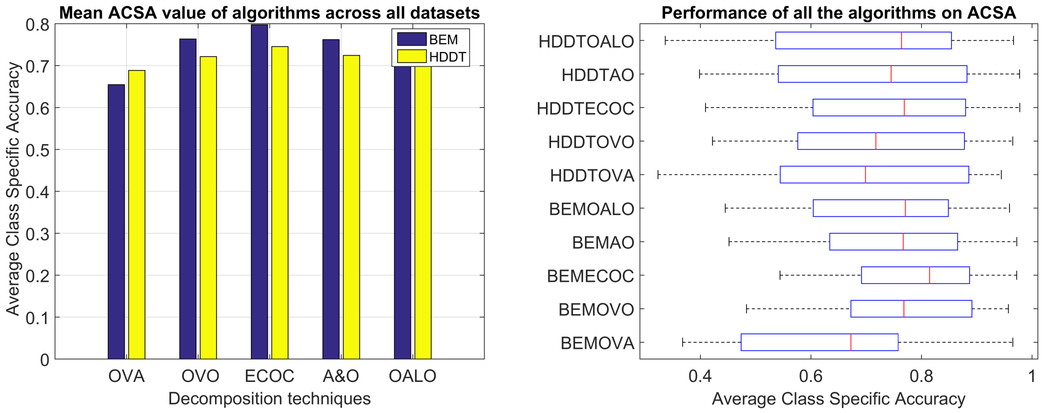

Figure 6.3: Visualization of the ACSA performance of algorithms for all data sets using Bar plot and Box plot

Table 6.7: Performance of **HDDT** algorithms on **ACSA**. The bold values in each row represents the best performing algorithm for that particular data set.

Dataset	**BEM**				
	OVA	**OVO**	**ECOC**	**A&O**	**OALO**
D1	0.54451	0.51187	0.62425	0.52534	0.51377
D2	0.62837	0.57616	0.60375	0.60051	0.56189
D3	0.45186	0.50529	0.51612	0.47622	0.53625
D4	0.88563	0.96506	0.95981	0.93838	0.93554
D5	0.67333	0.71663	0.77939	0.74901	0.76527
D6	0.59389	0.69786	0.71518	0.65627	0.61718
D7	0.89697	0.87758	0.8798	0.85455	0.83879
D8	0.66921	0.70422	0.69601	0.66857	0.68658
D9	0.91079	0.8784	0.91746	0.88905	0.9181
D10	0.72409	0.72859	0.80572	0.7411	0.77362
D11	0.80894	0.85031	0.87365	0.88234	0.83323
D12	0.94439	0.95661	0.9778	0.97742	0.91753
D13	0.93434	0.94608	0.93341	0.97397	0.96653
D14	0.32332	0.42194	0.40923	0.39833	0.33646
D15	0.36953	0.47708	0.53512	0.49831	0.48504
D16	0.73833	0.71833	0.75833	0.79875	0.7625
D17	0.86088	0.86213	0.82937	0.869	0.85425
D18	0.42411	0.59315	0.60283	0.54095	0.47497
Average	0.688472	0.721516	**0.745402**	0.724337	0.709861

Analysis

1. Like the F-measure table, **BEMECOC** is the best algorithm in terms of ACSA in 9 out of 18 data sets.

Table 6.8: Performance of **BEM** algorithms on **ACSA**. The bold values in each row represents the best performing algorithm for that particular data set.

Dataset	**BEM**				
	OVA	**OVO**	**ECOC**	**A&O**	**OALO**
D1	0.7227	0.72606	0.78462	0.634	0.641
D2	0.45527	0.65366	0.54407	0.5975	0.60423
D3	0.47378	0.48341	0.61897	0.54496	0.50344
D4	0.68529	0.95723	0.97222	0.97237	0.93443
D5	0.65162	0.72567	0.81144	0.7698	0.66783
D6	0.68903	0.67226	0.69153	0.75462	0.76292
D7	0.76525	0.7503	0.79343	0.8455	0.84859
D8	0.4228	0.68779	0.75237	0.6533	0.4450
D9	0.96524	0.94206	0.96254	0.95429	0.95921
D10	0.72877	0.81939	0.8652	0.75915	0.82923
D11	0.65933	0.83502	0.88128	0.81754	0.79235
D12	0.90832	0.94741	0.94029	0.97132	0.95825
D13	0.90582	0.95336	0.90095	0.96886	0.94317
D14	0.36785	0.51407	0.58017	0.45186	0.5115
D15	0.45462	0.59301	0.67216	0.6093	0.5132
D16	0.75792	0.89125	0.88708	0.86542	0.77875
D17	0.63875	0.80413	0.87792	0.78275	0.81338
D18	0.53003	0.78598	0.81786	0.76417	0.6638
Average	0.654577	0.763448	**0.79745**	0.762039	0.731682

Table 6.9: Rankings of algorithms based on **Average Class Specific Accuracy (ACSA)** value over all the 18 data sets.The best algorithm for each data set has been marked in bold(Rank 1).

Dataset	**Proposed Ensemble Method**					**Hellinger Distance Decision Tree**				
	OVA	**OVO**	**ECOC**	**A&O**	**OALO**	**OVA**	**OVO**	**ECOC**	**A&O**	**OALO**
D1	3	2	**1**	5	4	7	10	6	8	9
D2	10	**1**	9	6	3	2	7	4	5	8
D3	9	7	**1**	2	6	10	5	4	8	3
D4	10	5	2	**1**	8	9	3	4	6	7
D5	10	6	**1**	3	9	8	7	2	5	4
D6	6	7	5	2	**1**	10	4	3	8	9
D7	9	10	8	6	5	**1**	3	2	4	7
D8	10	4	**1**	8	9	6	2	3	7	5
D9	**1**	5	2	4	3	8	10	7	9	6
D10	8	3	**1**	6	2	10	9	4	7	5
D11	10	5	2	7	9	8	4	3	**1**	6
D12	10	6	8	3	4	7	5	**1**	2	9
D13	9	4	10	2	6	7	5	8	**1**	3
D14	8	2	**1**	4	3	10	5	6	7	9
D15	9	3	**1**	2	5	10	8	4	6	7
D16	8	**1**	2	3	5	9	10	7	4	6
D17	10	7	**1**	6	3	9	8	4	5	2
D18	8	2	**1**	3	4	10	6	5	7	9
Average	8.222	4.445	**3.167**	4.056	4.945	7.833	6.167	4.278	5.556	6.33

2. Combining all the decomposition methods, BEM has higher ACSA value than HDDT in 14 data sets and the average value for BEM is always higher than that of HDDT.

3. Similar to F-Measure, it can be observed that for both BEM and HDDT, A&O has the 2nd best rankings after ECOC.

4. The bar plot for ACSA shows BEMECOC having the minimum range and

maximum medium value among all and hence justifies the superiority and consistency of BEMECOC.

Summary

Analyzing the values, corresponding ranks, bar plots and box plots for G-Means,F-Measure and ACSA,the following things can be summarized about the performance of the different algorithms :

- Error Correcting Output Codes decomposition when applied to our proposed method (BEMECOC) is the best choice algorithm for dealing with a multiclass imbalanced data set.
- If we compare HDDT with our proposed ensemble scheme BEM for all the 5 decomposition methods combined, BEM outperforms HDDT in every case.
- When the same classifier is combined with different decomposition methods,increase in the number of dichotomies leads to better results.The number of classifiers trained by ECOC,A&O,OVO,OVA and OALO are $2^n - 1$, $[\frac{n(n-1)}{2} + n]$, $\frac{n(n-1)}{2}$, n and $n - 1$ respectively.With the maximum number of dichotomies trained, ECOC decomposition yields the best results followed by A&O and OVO.In OVA and OALO, less number of dichotomies are trained compared to the other methods and hence the metric values for these techniques are significantly less than ECOC,A&O and OVO.
- For applications having time efficiency constraints and in cases where classification performance can be adjusted to some extent, the One-and-All (A&O) decomposition technique will be the preferred choice.

6.4 Statistical Analysis

Friedman Test

To test whether our results are statistically significant or not we perform the Friedman test on the values of Table 6.1,6.3 and 6.5.The *p*-values obtained from the tests at a confidence interval of 95% have been listed in Table 6.7.As evident from the table, the *p*-values corresponding to each of G-means, F-Measure and ACSA are less than 5%(0.05) and so it can be claimed there is significant difference between the rankings of the algorithms.

Pairwise Comparison by Wilcoxon Signed Rank Test

After determining the algorithm producing the best results we have compared BEMECOC with the remaining 9 methods by the Wilcoxon Signed Rank Test at 95% confidence interval.The results of the test for G-Means, F-Measure and ACSA

Table 6.10: p-values obtained from Friedman test corresponding to G-Means,F-Measure and ACSA respectively.

Sl No	**Evaluation Metric**	**p value**	**Statistically significant(less than 0.05)**
1	G-Means	0.0019	**Yes**
2	F-Measure	0.0079	**Yes**
3	ACAS	0.0056	**Yes**

have been given in Table 6.8.Observing the p-values,we notice that for G-means, all the p-values are significantly less than 0.05.Similar situation can be observed for F-Measure and ACSA.Thus it can be concluded from the results that BEMECOC significantly outperforms the other methods.

Table 6.11: Wilcoxon Signed Rank test for pairwise comparison of BEMECOC with the other 9 algorithms at 95% confidence interval

G-MEANS		**F-MEASURE**		**ACSA**	
Method	**p-value**	**Method**	**p-value**	**Method**	**p-value**
BEMOVA	0.000233	*BEMOVA*	0.000386	*BEMOVA*	0.000455
BEMOVO	0.000863	*BEMOVO*	0.019809	*BEMOVO*	0.015647
BEMA&O	0.005684	*BEMA&O*	0.034669	*BEMA&O*	0.052624
BEMOALO	0.002471	*BEMOALO*	0.004969	*BEMOALO*	0.019809
HDDTOVA	0.000276	*HDDTOVA*	0.000863	*HDDTOVA*	0.003285
HDDTOVO	0.000629	*HDDTOVO*	0.000863	*HDDTOVO*	0.008418
HDDTECOC	0.000863	*HDDTECOC*	0.007398	*HDDTECOC*	0.034669
HDDTA&O	0.000863	*HDDTA&O*	0.000535	*HDDTA&O*	0.010843
HDDTOALO	0.000455	*HDDTOALO*	0.000276	*HDDTOALO*	0.002471

6.5 Comparison with state-of-the-art ensemble method

We have established the superiority of our ensemble learning paradigm and to ensure a fair comparison with other ensemble techniques dedicated to multiclass imbalanced data classification tasks, we have compared BEM with one of the state-of-the art methods **AdaboostNC**. AdaboostNC is a combination of multiclass boosting and negative correlation learning. AdaboostNC is different from the classical Adaboost algorithm by the fact that here, after each classifier is built, the difference between the classifiers within the ensembles is given a penalty term. Along with the miss classification information, the diversity within the ensemble is taken into account for updating the weights. For our experiments, we have taken the following parameters of AdaboostNC as suggested by the authors.

Table 6.12: Parameter settings of AdaboostNC

Parameter	**Value**
Number of iterations	20
Penalty term	2
Base Classifier	C4.5

Table 6.13: Comparison between BEMECOC and AdaboostNC

Dataset	**G-Means**		**F-Measure**		**ACSA**	
	BEMECOC	**AdaboostNC**	**BEMECOC**	**AdaboostNC**	**BEMECOC**	**AdaboostNC**
D1	0.7594	0.39206	0.79562	0.4561	0.78462	0.52507
D2	0.72682	0.51706	0.65748	0.57435	0.54407	0.60636
D3	0.68695	0.46781	0.58276	0.47023	0.61897	0.47066
D4	0.93338	0.94531	0.97411	0.92274	0.97222	0.9265
D5	0.87344	0.87834	0.8922	0.88887	0.81144	0.73191
D6	0.85424	0.72989	0.75641	0.72554	0.69153	0.68704
D7	0.8314	0.90645	0.85647	0.91873	0.79343	0.91394
D8	0.76198	0.41469	0.84905	0.42042	0.75237	0.43386
D9	0.94075	0.89952	0.94439	0.91648	0.96254	0.90921
D10	0.89954	0.81463	0.93765	0.87274	0.8652	0.73227
D11	0.90015	0.87079	0.88299	0.88491	0.88128	0.88263
D12	0.97561	0.83564	0.96676	0.88112	0.94029	0.8594
D13	0.9484	0.94595	0.93799	0.94764	0.90095	0.94545
D14	0.55424	0.37561	0.5565	0.46013	0.58017	0.3761
D15	0.6581	0.39486	0.60781	0.47018	0.67216	0.45615
D16	0.88337	0.7206	0.88793	0.77948	0.88708	0.73542
D17	0.82432	0.79101	0.87549	0.88346	0.87792	0.85479
D18	0.85762	0.62372	0.78768	0.56428	0.81786	0.52348

Analysis

Table 6.12 shows the comparative results between BEM and AdaboostNC in terms of G-Means, F-Measure and ACSA. As observed from the table, in terms of the 3 evaluation metrics, BEMECOC significantly outperforms AdaboostNC in majority of the data sets. AdaboostNC has won marginally in data sets which have less degree of imbalance. On highly imbalanced data sets like Abalone 19, Contraceptive Led7digit as well as on the artificial ones, our method comprehensively beats AdaboostNC. These observations assert the superiority and dominance of BEMECOC.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In a multiclass imbalanced data set with, the level of imbalance is much more complex than that of binary data due to small disjuncts being present in the data and classes having a lot of overlaps. One of the most common strategies for tackling the problem is to decompose the multiclass problem into binary imbalance sub problems and use the classifiers which are generally suited to deal with binary class imbalanced data sets. The proven efficacy of the classifiers to tackle a two-class imbalance problem motivated us to make a comparative analysis of these decomposition ensembles and choose the appropriate combination of classifier and decomposition strategy that produces the best output.

In our work, we have used two types of learning algorithms to classify a two class imbalanced data set. For an algorithmic level learning approach we have used Hellinger Distance Decision Tree proposed by Cieslak et al.. To validate the author's claim why Hellinger Distance is a good choice for splitting criterion in decision trees, we have compared it with other members of the F-Divergence family like the Kullback-Leibler Divergence and the Jensen-Shannon Divergence. To utilize the advantages of both data level and algorithmic level modifications, we have proposed an ensemble paradigm *Balanced Ensemble Models*(BEM) where balanced subsets are created by including all the minority class samples and an equal number of majority class samples picked without replacement and then a weighted voting strategy is applied to the ensemble models to classify the test data. We have used five decomposition techniques One-vs-All(OVA), One-vs-One(OVO), Error Correcting Output Codes(ECOC), One-and-All(A&O) and One-Against-Lower-Order(OALO) and applied each of these strategies to both HDDT and BEM.

For our experimental study we have used 3 popular evaluation metrics for multiclass imbalance problems : G-Means, F-Measure and Average Class Specific Accuracy which is a multiclass equivalent of Average Recall. For each of these metrics, the Tied Rank test have been used to compute the rankings of the classifiers followed by the Friedman Rank to test whether there is any significant differences between the rankings of the algorithms. From the average rankings across all data sets, we can conclude that ECOC when applied to our proposed method BEM yields the best result. To validate the claim that BEMECOC outperforms all the other methods, we have used the Wilcoxon Signed Rank test at a confidence level of 5% to perform a pairwise comparison of BEMECOC with the rest of the algorithms.

From our experimental findings and statistical tests, we conclude the fact that Error Correcting Output Codes as a decomposition strategy yields the best result. A&O and OVO producing comparable performances comes after that followed by OALO and OVO. We can highlight the fact that more the number of dichotomies trained in each decomposition scheme, better the result. If we compare the two types of learning algorithms, algorithmic and hybrid approaches, the superior performance of BEM in majority of the data sets clearly prove that an ensemble paradigm of both data level and algorithmic modifications have a better chance to effectively deal with the challenges of a multiclass imbalanced classification data set.

7.2 Scope for Future Work

Despite its good performance, our proposed algorithm BEMECOC has a few limitations and there is scope of improvement.One of the challenges of the ECOC decomposition strategy is that it is quite time-consuming. As the number of classes increase, the number of dichotomies trained increases exponentially. But in applications where time efficiency constraints can be tolerated and classification accuracy is the main focus, BEMECOC can be considered as the first choice. In future work, we plan to use parallel based methods to reduce the training time without compromising on the accuracy.

While OVA and OVO have been widely used as decomposition strategies in multiclass imbalance problems, the rest of the techniques have not been utilized much in this domain. To the best of our knowledge, this is the first study where a thorough comparative analysis of all the 5 techniques has been made to highlights their advantages and limitations. A scope for future work can be combining one or more of these decomposition schemes to boost up classification accuracy. We sincerely hope that this study should have important reference value for researchers in the imbalance learning domain.

Chapter 8

Appendix

Theorem 3. *Square of Hellinger Distance is the lower bound of the Kullback-Leibler Divergence.*

Proof. The proof will be complete if we can show that the Bhattacharyya distance is the lower bound of the Kullback-Leibler Divergence.

The Bhattacharyya coefficient can be defined as :

$$D_B(x, y) = \int \sqrt{x(i)y(i)}\,\mathrm{d}i \tag{8.1}$$

From Equation 2.1,Hellinger distance can be reformulated as follows:

$$d_H(x, y) = \{1 - \left(\int \sqrt{x(i)y(i)}\,\mathrm{d}i\right)\}^{1/2} \tag{8.2}$$

Therefore,Hellinger distance can be expressed in terms of Bhattacharyya coefficient as follows :

$$d_H(x, y) = \{1 - D_B(x, y)\}^{1/2} \tag{8.3}$$

Bhattacharyya Distance :

$$\begin{aligned}
d_B(x, y) &= -\log D_B(x, y) \\
&= -\log \int \sqrt{x(i)y(i)}\,\mathrm{d}i \\
&\overset{\text{def}}{=} -\log \int z(i)\,\mathrm{d}i \\
&= -\log \int \frac{z(i)}{x(i)}\,x(i)\,\mathrm{d}i \\
&\leq \int -\log \left\{\frac{z(i)}{x(i)}\right\}\,x(i)\,\mathrm{d}i \\
&= \int \frac{-1}{2} \log \left\{\frac{z^2(i)}{x^2(i)}\right\}\,x(i)\,\mathrm{d}i \\
&= \int \frac{-1}{2} \log \left\{\frac{y(i)}{x(i)}\right\}\,x(i)\,\mathrm{d}i \qquad\qquad = \frac{1}{2} d_{KL}(x \| y)
\end{aligned}$$

Thus,it can be said that Bhattacharyya distance is the lower bound of the Kullback-Leibler Divergence.

$$d_{KL}(x\|y) \geq 2d_B(x,y) \tag{8.4}$$

From the graph of $log(x)$,

$$-log(i) \geq 1 - i \qquad\qquad 0 \leq i \leq 1 \tag{8.5}$$

So,finally we can conclude from Equations 2.7 and 2.8,

$$d_B(x,y) \geq d_H(x,y)^2 \tag{8.6}$$

□

Bibliography

[1] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, (9):1263–1284, 2008.

[2] Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113 – 141, 2013. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2013.07.007. URL `http://www.sciencedirect.com/science/article/pii/S0020025513005124`.

[3] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[4] Bing Liu, Yiming Ma, and Ching Kian Wong. Improving an association rule based classifier. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 504–509. Springer, 2000.

[5] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Machine learning*, 46(1-3):191–202, 2002.

[6] Ricardo Barandela, José Salvador Sánchez, V Garca, and Edgar Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36 (3):849–851, 2003.

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[8] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008.

[9] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

[10] Nitesh V Chawla, David A Cieslak, Lawrence O Hall, and Ajay Joshi. Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery*, 17(2):225–252, 2008.

[11] Charles X Ling, Victor S Sheng, and Qiang Yang. Test strategies for cost-sensitive decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1055–1067, 2006.

[12] Shichao Zhang, Li Liu, Xiaofeng Zhu, and Chen Zhang. A strategy for attributes selection in cost-sensitive decision trees induction. In *2008 IEEE 8th International Conference on Computer and Information Technology Workshops*, pages 8–13. IEEE, 2008.

[13] Part Pramokchon and Punpiti Piamsa-nga. Reducing effects of class imbalance distribution in multi-class text categorization. In *Recent Advances in Information and Communication Technology*, pages 263–272. Springer, 2014.

[14] B Abidine M'hamed and Belkacem Fergani. A new multi-class wsvm classification to imbalanced human activity dataset. *Journal of Computers*, 9(7), 2014.

[15] Bartosz Krawczyk, Mikel Galar, Łukasz Jeleń, and Francisco Herrera. Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. *Applied Soft Computing*, 38:714–726, 2016.

[16] Alberto FernáNdez, Victoria LóPez, Mikel Galar, MaríA José Del Jesus, and Francisco Herrera. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-based systems*, 42:97–110, 2013.

[17] Minlong Lin, Ke Tang, and Xin Yao. Dynamic sampling approach to training neural networks for multiclass imbalance classification. *IEEE Transactions on Neural Networks and Learning Systems*, 24(4):647–660, 2013.

[18] Shuo Wang and Xin Yao. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1119–1130, 2012.

[19] Zhongliang Zhang, Bartosz Krawczyk, Salvador Garcìa, Alejandro Rosales-Pérez, and Francisco Herrera. Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data. *Knowledge-Based Systems*, 106:251–263, 2016.

[20] David A Cieslak, T Ryan Hoens, Nitesh V Chawla, and W Philip Kegelmeyer. Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*, 24(1):136–158, 2012.

[21] Rangachari Anand, Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, 6(1):117–124, 1995.

[22] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In *Advances in neural information processing systems*, pages 507–513, 1998.

[23] Thomas G Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, 2:263–286, 1994.

[24] Nicolas Garcia-Pedrajas and Domingo Ortiz-Boyer. Improving multiclass pattern recognition by the combination of two strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1001–1006, 2006.

[25] Yi L Murphey, Haoxing Wang, Guobin Ou, and Lee A Feldkamp. Oaho: an effective algorithm for multi-class learning from imbalanced data. In *2007 International Joint Conference on Neural Networks*, pages 406–411. IEEE, 2007.

[26] L Breiman, J Friedman, R Olshen, and C Stone. Classification and regression trees (chapman y hall, eds.). *Monterey, CA, EE. UU.: Wadsworth International Group*, 1984.

[27] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[28] Foster Provost and Pedro Domingos. Tree induction for probability-based ranking. *Machine learning*, 52(3):199–215, 2003.

[29] Jerome H Friedman. Another approach to polychotomous classification. *Technical Report, Statistics Department, Stanford University*, 1996.

[30] Jose García Moreno-Torres, José A Sáez, and Francisco Herrera. Study on the impact of partition-induced dataset shift on k-fold cross-validation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1304–1312, 2012.

[31] Victoria López, Alberto Fernández, and Francisco Herrera. On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. *Information Sciences*, 257: 1–13, 2014.

[32] Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):31, 2016.

[33] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

[34] Ricardo Barandela, Rosa Maria Valdovinos, and José Salvador Sánchez. New applications of ensembles of classifiers. *Pattern Analysis & Applications*, 6 (3):245–256, 2003.

[35] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[36] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27 (8):861–874, 2006.

[37] Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978.

[38] Waleed A Yousef, Robert F Wagner, and Murray H Loew. Estimating the uncertainty in the estimated mean area under the roc curve of a classifier. *Pattern Recognition Letters*, 26(16):2600–2610, 2005.

[39] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[40] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[41] David J Hand and Robert J Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45 (2):171–186, 2001.

[42] Salvador García, Alberto Fernández, Julián Luengo, and Francisco Herrera. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Computing*, 13(10):959, 2009.

[43] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

[44] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.