2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

An Algorithmic Framework Based on the Binarization Approach for Supervised and Semi-supervised Multiclass Problems

Ayon Sen Md. Monirul Islam Kazuyuki Murase

***Abstract*— Using a set of binary classifiers to solve the multiclass classification problem has been a popular approach over the years. This technique is known as binarization. The decision boundary that these binary classifiers (also called base classifiers) have to learn is much simpler than the decision boundary of a multiclass classifier. But binarization gives rise to a new problem called the class imbalance problem. Class imbalance problem occurs when the data set used for training has relatively less data items for one class than for another class. This problem becomes more severe if the original data set itself was imbalanced. Furthermore, binarization has only been implemented in the domain of supervised classification. In this paper, we propose a framework called Binarization with Boosting and Oversampling (BBO). Our framework can handle the class imbalance problem arising from binarization. As the name of the framework suggests, this is achieved through a combination of boosting and oversampling. BBO framework can be used with any supervised classification algorithm. Moreover, unlike any other binarization approaches used earlier, we apply our framework with semi-supervised classification as well. BBO framework has been rigorously tested with a number of benchmark data sets from UCI machine learning repository. The experimental results show that using the BBO framework achieves a higher accuracy than the traditional binarization approach.**

I. Introduction

Machine learning is a branch of artificial intelligence which is concerned with the study and construction of systems which can learn from data. Classification is the machine learning task of identifying the class membership of data items. A classifier that differentiates between two classes is called a binary classifier. A multiclass classifier, on the other hand, differentiates between three or more classifiers. The learning task for a binary classifier is relatively easier compared to a multiclass classifier because the decision boundary that a binary classifier needs to learn is relatively simpler [1], [2].

How to solve the multiclass problem still remains an open issue [1]. One popular approach is to use a set of binary classifiers (*base classifiers*) to solve the problem. This technique is known as *binarizaiton*. Binarization not only helps in producing simple classifiers, but it also helps in reducing time to train classifiers. This is because there is a chance to train the binary classifiers in parallel. Over the years, many algorithms have been devised based on binarization [3]. There are mainly two popular decomposition techniques regarding binarization: one-vs-one (OVO) and one-vs-all (OVA).

The OVO approach divides the problem into as many binary problems as possible combinations between pairs of classes. One classifier is trained to discriminate between a pair of classes. If there are n classes then the number of base classifiers is ${}^{n}\mathrm{C}_{2}$. OVA consists in creating a base classifier to learn each class, where the class is distinguished from all other classes. OVA has the benefit of using less resources than OVO. If there are n classes then the number of base classifiers for OVA will also be n which is significantly less than ${}^{n}\mathrm{C}_{2}$ of OVO when the value of n is large. In both cases, the output of the base classifiers need to be combined to predict the output class. This is known as the aggregation strategy. OVA has mainly two aggregation strategies: the max confidence strategy (MAX) and the dynamically ordered OVA (DOO) [4] strategy. OVO, on the other hand, has multiple aggregation strategies with no clear winner. Even though many algorithms have been developed over the years based on binarization, most of them have been unable to address a problem arising from binarization: the class imbalance problem [5], i.e., the lack of data for a class in the training set. This can adversely effect the accuracy of classifiers. Moreover, to the best of our knowledge, binarization has only been implemented in the domain of supervised classification.

In this paper, we propose a framework based on OVA. The salient feature of our framework is that it can handle the class imbalance problem occurring due to binarization. Furthermore, it can be used with any supervised classification algorithm. It can also be used with any semi-supervised classification algorithm, which has not been used with binarization before. Our framework uses a combination of boosting and oversampling techniques to address the class imbalance problem. To predict the output class we use the DOO aggregation strategy.

The rest of the paper is organized as follows. A detailed discussion of related research work is given in Section II. Our proposed framework is presented in Section III. In Section IV, we give our experimental findings. We briefly conclude in Section V.

II. Related Works

Using binary classifiers to solve supervised multiclass problems has been a popular technique over the years. In this section, we give an overview of different OVO and OVA approaches to solve the multiclass problem. These approaches have been implemented to solve various problems using a multitude of binary classifiers. Since we implement our framework with semi-supervised classification also, we discuss the popular semi-supervised classification algorithms in this section.

OVO has been used by researchers to solve different multiclass problems. The main difference between the dif-

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

ferent approaches is the aggregation strategy, i.e, combining the output of the base classifiers to get a class prediction. Friedman [6] proposed an aggregation strategy where the output class was determined by counting the number of votes for all the classes. The authors of [7] proposed a modification to this method. Instead of voting for the winning class, each base classifier votes against the losing class. The class with the least number of votes is predicted as the output class. The weighted voting strategy is introduced by [8]. An adapting version of this weighted voting strategy is proposed by the authors of [9]. In [10], the authors proposed an aggregation strategy where the joint probability for all classes is determined to find the best approximation. The authors proposed a tree based approach in [11] where a rooted binary acyclic tree is constructed with each node being associated with a list of classes and a binary classifier.

OVA techniques have been used on multiple problem domains over the years. The authors of [4] proposed an OVA strategy to classify fingerprints. Their proposed algorithm uses support vector machine (SVM) for classification. They train a Naive Bayes classifier in parallel with the base classifiers. This classifier establishes a sequence in which the OVA base classifiers will be executed for a given data item. A data item is given to the base classifiers in the established sequence until a positive answer is obtained. This is the output class. The rest of the classifiers are not used for that data item. So, ties are avoided a priori by using the Naive Bayes classifier. [12] uses a text-query based method for Chinese handwriting detection using SVM. The authors of [13] investigated the problem of video categorization and Delechaux et al. [14] use neural networks as their base classifier to recognize indoor activities.

As labeled data is expensive to come by, the practice of using unlabeled data to improve the effectiveness of a classifier has become popular in recent years. Hence, a lot of theoretical and practical work has been done in the field of semi-supervised classification. These algorithms can be broadly categorized into co-training, methods based on manifold assumption, methods based on cluster assumption and ensemble methods. In the co-training methods independent algorithms are trained and they learn from each other [15], [16], [17]. In semi-supervised classification it is assumed that the true structure of the data lies in a low-dimensional manifold embedded in the high-dimensional data space. This is known as the *manifold assumption*. Algorithms based on this assumption typically build graphs to represent all the instances [18], [19]. The cluster assumption states that classes are often separated by a low-density region. TSVM [20], SemiBSVM [21] and LLGC [22] algorithms are based on this assumption. LLGC uses the regularization framework. During each step the information of the unlabeled data set is gathered from its neighbors based on a parameter, α. Zhai et al. proposed a multiview version of LLGC in [23]. LLGC has been known to perform well in different domains like image and text classification. In [24], the authors proposed a technique based on regularization. Their algorithm takes the partition given by an algorithm as a regularization term in the loss function of an semi-supervised classifier. The authors of [25], [26], [27] proposed algorithms based on ensemble. The authors of [26] also address the issue of imbalanced data set. They work with multiclass classifiers and use boosting to address the issue. Moreover they mention using binarization for solving multiclass problems but do not present any existing work.

In this section, we gave overviews of research works related to binarization and semi-supervised classification. But most of the research work discussed here fail to address the issue of class imbalance problem. In our proposed framework our focus is to tackle this problem. Moreover, we extend the domain of binarization to the field of semi-supervised classification.

III. Our Approach

Binarization techniques have been proven to be very effective in handling multiclass classification problems. The main advantage is that the reformulation leads to classifiers who have to deal with decision boundaries which are simpler than their multiclass counterpart. An example of this simplicity is given in Figure 1. Here the binarization approach used is OVA. As can be seen from the figure, the decision boundary that an individual base classifier needs to learn is much simpler than the actual decision boundary of the problem. The decision boundary that a base classifier of OVO has to learn is even more simpler than the one shown in the figure as only two classes are involved. Hence, the classification task for binarization also becomes simpler. Moreover, there are many popular binary classifiers available. So, the individual classification task can be learnt with higher accuracy too.

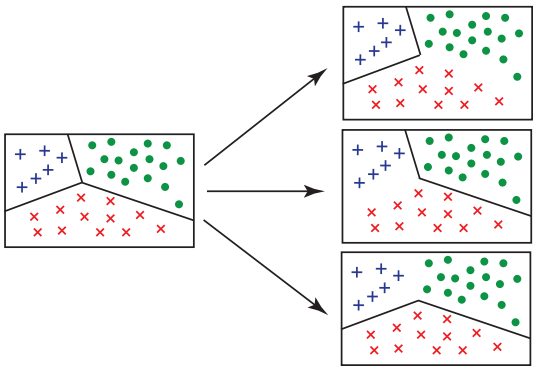

Fig. 1. Decision Boundary for OVA

But binarization has its drawbacks too. As it can be seen from Figure 1, after dividing the data set for OVA the proportion of data items for one class to others changes significantly, i.e., the data items for the class the base classifier is supposed to be trained for (target class) is significantly less than the data items for the other class. This problem is known as the *class imbalance problem*. This problem will occur in most cases of binarization unless the number of data items for one class is significantly greater than the number of data items in other classes. In that case the class

imbalance problem will arise for the data items of the other classes. Class imbalance problem is known to be responsible for reduced accuracy in many systems and it hampers the effectiveness of the base classifiers. The problem would be more severe if the original data set itself is imbalanced. Moreover, to the best of our knowledge binarization has only been used in the domain of supervised classification. It has not been used with unsupervised and semi-supervised classification.

Algorithm 1 Training Base Classifiers for Supervised Learning in the BBO Framework

$H \leftarrow$ the base classifier
$TC \leftarrow$ the target class
$DC \leftarrow$ the default class
$N \leftarrow$ total number of iterations
$BC \leftarrow$ number of times a data item would be copied for boosting
$N_1 \leftarrow$ iteration after which oversampling will start
$OP \leftarrow$ percentage of oversampled data created
$TS \leftarrow$ the binarized training data set
$CopyTS \leftarrow$ Copy(TS)
for $i = 1$ to N **do**
  $H \leftarrow$ Train$(H, CopyTS)$
  $S \leftarrow \phi$
  for $j = 1$ to Size(TS) **do**
    if $TS[j].class = TC$ **then**
      if Test$(H, TS[j]) \neq TC$ **then**
        $S \leftarrow S \cup TS[j]$
      end if
    end if
  end for
  for $j = 1$ to Size(S) **do**
    for $k = 1$ to BC **do**
      $CopyTS \leftarrow CopyTS \cup$ Copy$(S[j])$
    end for
  end for
  if $i > N_1$ **then**
    $CopyTS \leftarrow CopyTS \cup$ Oversample(S, OP)
  end if
end for
return H

In this paper, we address the issues arising from binarization. We propose a framework that handles the class imbalance problem. Furthermore, we also apply our framework for semi-supervised classification. Recently some solutions have been used to solve the class imbalance problem like boosting, oversampling and undersampling. Boosting is a technique where the data of the class with fewer data items are repeated, i.e., multiple copies of the same data is incorporated in the training set. Oversampling creates synthetic data for the class with fewer items. Undersampling deletes data items from the training set for the classes with more data items. We incorporate a combination of boosting and undersampling in our framework. Henceforth, we call our framework *Binarization with Boosting and Oversampling (BBO)*.

The decomposition technique that we choose for the BBO framework is OVA. OVA creates n binary classifiers whereas OVO creates ${}^{n}\mathrm{C}_{2}$ classifiers. Thus OVO is more resource hungry than OVA. Moreover, as discussed in Section I there are multiple aggregation strategies for OVO with no clear winner. OVA has mainly two aggregation strategies. Keeping these in mind we prefer OVA over OVO for the BBO framework.

Algorithm 2 Training Base Classifiers for Semi-supervised Learning in the BBO Framework

$H \leftarrow$ the base classifier
$TC \leftarrow$ the target class
$DC \leftarrow$ the default class
$N \leftarrow$ total number of iterations
$BC \leftarrow$ number of times a data item would be copied for boosting
$N_1 \leftarrow$ iteration after which oversampling will start
$OP \leftarrow$ percentage of oversampled data created
$LD \leftarrow$ the binarized labeled data set
$UD \leftarrow$ the binarized unlabeled data set
$CopyLD \leftarrow$ Copy(LD)
for $i = 1$ to N **do**
  $H \leftarrow$ Train$(H, CopyLD, UD)$
  $S \leftarrow \phi$
  for $j = 1$ to Size(LD) **do**
    if $LD[j].class = TC$ **then**
      if Test$(H, LD[j]) \neq TC$ **then**
        $S \leftarrow S \cup LD[j]$
      end if
    end if
  end for
  for $j = 1$ to Size(S) **do**
    for $k = 1$ to BC **do**
      $CopyLD \leftarrow CopyLD \cup$ Copy$(S[j])$
    end for
  end for
  if $i > N_1$ **then**
    $CopyLD \leftarrow CopyLD \cup$ Oversample(S, OP)
  end if
end for
return H

At the start of training, for each base classifier the data set is divided into two classes: the class we want the base classifier to learn, i.e., the *target class* and the *default class*. So, for each classifier the training data set is also binarized i.e., the training data set for a base classifier contains two classes only: the target class and the default class. All data items not belonging to the target class are labeled as the default class. After dividing the training data set as such we check for class imbalance in the binarized data set. If the proportion of the number of data items between the target class and the default class is less than a predefined threshold then we apply oversampling to increase the number of data items for the target class.

The main goal of each base classifier is to learn the decision boundary for the target class. In the BBO framework we solely focus on this goal. The classification task can thus be said to learn this decision boundary better with each iteration. To help the base classifier learn this decision boundary better, we change the training set in between iterations. After each iteration we check which data items of the target class the base classifier has misclassified. Then the focus of the base classifier should be to be able to classify

these misclassified data items of the target class during the next iteration. For this purpose, during the next iteration we include multiple copies of these misclassified target class data items in the training set i.e., apply boosting, so that the classifier learns these items better during the next iteration. But if boosting is applied alone then it is possible for overfitting to occur. While using boosting BBO framework solely focuses on learning the misclassified data sets. Thus applying boosting alone may lead to overfitting. To avoid this problem we also oversample the misclassified data items and also add these new synthetic data items into the training data set. This way the base classifier learns to classify the misclassified data items better. It would have been possible only to use oversampling instead of using boosting and oversampling together. But we do not want the classifier to learn mainly the synthetic data. So, oversampling is not used alone and is also given less importance than boosting. We do this by introducing oversampling only during the latter iterations of the learning period. The training process of the base classifiers for supervised classification is presented in Algorithm 1.

We also use semi-supervised classifiers in the BBO framework. For such classification the framework remains almost the same. Semi-supervised classification uses both labeled and unlabeled data during training. As we do not know the class of the unlabeled data, boosting and oversampling is not performed on them. Boosting and oversampling is performed only on the labeled data. The rest of the BBO framework remains the same. The training process of base classifiers for semi-supervised classification is presented in Algorithm 2.

After training of the base classifiers is done, to find the output class of a data item we need to combine the outputs of the base classifiers. As discussed in Section I there are mainly two aggregation strategies for the OVA approach: MAX and DOO. These two strategies vary in the handling of ties. A tie occurs when two base classifiers return a positive response for their target classes. In such a case the tie needs to be resolved. In the MAX strategy the base classifiers also produce a probability or confidence of their output. When a tie occurs the target class of the base classifier with the highest confidence is chosen as the output class. But since a base classifier only learns one class, this strategy may lead to faulty outcomes. In the DOO strategy ties are handled a priori. As discussed in Section II a Naive Bayes classifier is trained in parallel with the base classifiers. This classifier produces a sequence in which the base classifiers will be executed for a given data item. The target class of a base classifier that first returns a positive response in the established sequence is chosen as the output class. The rest of the base classifiers are not used. Thus ties never occur. DOO is known to produce better results than MAX for OVA [3]. We use DOO as the aggregation strategy in the BBO framework.

IV. Experimental Studies

In this section, we evaluate BBO framework's performance on several well known data sets. We compare our results with several benchmark algorithms. We discuss the data sets, experimental details, results and comparisons.

A. The Dataset

In this study, we selected 18 data sets from the UCI machine learning repository [28]. A summary of the data sets is given in Table I. These data sets were used for experimentation in [3]. We downloaded the data sets from http://sci2s.ugr.es/ovo-ova.

TABLE I
SUMMARY DESCRIPTION OF DATA SETS.

Data Set	#Example	#Attributes	#Numeric	#Nominal	#Classes
autos	159	25	15	10	6
car	1728	6	6	0	4
cleveland	297	13	5	8	5
dermatology	366	33	1	32	6
ecoli	336	7	7	0	8
flare	1389	10	0	10	6
glass	214	9	9	0	6
led7digit	500	7	0	7	10
lymphography	148	18	3	15	4
nursery	1296	8	0	8	5
pageblocks	548	10	10	0	5
penbased	1099	16	16	0	10
satimage	643	36	36	0	7
segment	2310	19	19	0	7
shuttle	2175	9	9	0	7
vehicle	846	18	18	0	4
vowel	990	13	13	0	11
zoo	101	16	0	16	7

B. Experimental Setup

For all our experiment purposes we used the Waikato Environment for Knowledge Analysis (Weka) tool [29]. It is a data mining tool written in java. There are popular approaches available for oversampling. One of these approaches is Synthetic Minority Overs-sampling Technique (SMOTE) [30]. SMOTE does not generate examples in an application specific manner. New synthetic examples are generated by considering all the data items of the minority class and creating new data items along the line segments joining any/all of the k minority class nearest neighbors. SMOTE has been known to perform well in various class imbalance problems. That is why we have chosen SMOTE as our oversampling algorithm of choice for the BBO framework. Other than SMOTE it is also possible to use MWMOTE [31] as the oversampling technique.

For supervised classification we chose neural network (NN) [32] as our classifier of choice. NN learns through iterations. The classifier tries to reduce the output error after each iteration. We have chosen NN as our classifier for this natural iterative process. The total number of iterations (N) used for NN is 1000. BC was set to 100, N_1 to 500 and OP to 400. The accuracy rate was obtained by means of a five-fold cross-validation. The data partitions used can be found in [33] and http://sci2s.ugr.es/ovo-ova. We have compared our framework with four baseline algorithms: a multi-class NN (M-NN), a binarized NN (B-NN), a binarized

TABLE II
SUPERVISED CLASSIFICATION ACCURACY.

Data Set	Best Binarization Algorithm		M-NN	B-NN	BB-NN	BO-NN	BBO-NN
	Name	Accuracy					
autos	1NN-DOO	82.96	74.21	69.84	69.23	78.59	76.73
car	PDFC-VOTE	100.00	99.71	97.05	97.86	98.96	99.31
cleveland	SVM-PC	58.59	55.54	54.90	57.59	45.11	59.60
dermatology	3NN-DOO	97.49	95.81	96.37	95.81	96.37	96.37
ecoli	PDFC-MAX	83.05	81.56	82.16	80.36	79.47	84.53
flare	SVM-MAX	75.42	74.20	69.70	71.11	72.51	72.98
glass	RIPPER-LVPC	74.77	61.25	49.55	48.62	57.96	67.77
led7digit	SVM-VOTE	73.40	71.00	69.80	69.20	71.00	70.60
lymphography	1NN-MAX	87.08	83.03	83.72	82.37	82.37	83.03
nursery	PDFC-PC	96.84	95.99	92.59	92.44	95.22	95.99
pageblocks	C45-DDAG	95.79	93.97	95.25	95.07	93.97	96.53
penbased	PDFC-PE	98.36	93.36	91.82	91.27	91.91	96.36
satimage	PDFC-PC	87.41	87.55	83.98	84.13	84.76	86.94
segment	1NN-PE	97.23	95.37	91.65	92.42	95.32	96.80
shuttle	1NN-KNN	99.77	98.11	98.02	96.64	97.70	99.40
vehicle	PDFC-PE	83.69	83.10	71.99	71.75	81.20	79.91
vowel	1NN-PE	99.19	87.68	81.45	84.11	80.00	92.73
zoo	PDFC-PC	96.00	97.00	96.00	96.00	95.05	96.05
Average	PDFC-VOTE	85.74	84.91	81.99	82.00	83.19	86.20

TABLE III
G-MEAN VALUES FOR NEURAL NETWORK

Data Set	M-NN	B-NN	BB-NN	BO-NN	BBO-NN
autos	0.7649	0.7338	0.7494	0.8405	0.8060
car	0.9867	0.9350	0.9691	0.9389	0.8254
cleveland	0.3459	0.3344	0.3423	0.2910	0.3813
dermatology	0.9552	0.9611	0.9578	0.9611	0.9611
ecoli	0.7568	0.7919	0.7969	0.7840	0.8269
flare	0.5489	0.5074	0.5467	0.5362	0.5257
glass	0.6106	0.5484	0.5600	0.7254	0.7743
led7digit	0.7214	0.7111	0.7281	0.7221	0.7164
lymphography	0.8276	0.9230	0.9256	0.9207	0.8948
nursery	0.8991	0.6585	0.6701	0.6760	0.6836
pageblocks	0.5615	0.7436	0.7439	0.7236	0.7546
penbased	0.9237	0.9152	0.9591	0.9162	0.9648
satimage	0.8595	0.8371	0.8462	0.8488	0.8733
segment	0.9502	0.9141	0.9153	0.8597	0.9113
shuttle	0.8500	0.8514	0.8820	0.8484	0.9133
vehicle	0.8075	0.6979	0.6869	0.7934	0.7788
vowel	0.8412	0.6125	0.8128	0.6188	0.9328
zoo	0.9158	0.9353	0.9521	0.9260	0.9443
Average	0.7848	0.7562	0.7802	0.7739	0.8038

NN that uses only boosting (BB-NN) and a binarized NN that uses only oversampling (BO-NN). N was set to 1000 for all baseline algorithms. BC was set to 100 for BB-NN. For BO-NN N_1 and OP was set to 400. Furthermore, we have compared our results with the most popular binarization algorithms. The results of these algorithms were obtained from [3].

For semi-supervised classification we used LLGC as our classifier of choice. We chose LLGC as our semi-supervised classifier of choice as it was the only available multiclass semi-supervised classifier available in WEKA and we wanted to compare our results with the results of the original algorithm. LLGC does not have any natural iterations. So, it is not possible to update the training set while training the algorithm. Therefore we retrain the classifier multiple times. The training set is updated after each training process. The total number of iterations (N) used for LLGC was 3. BC was set to 100, N_1 to 2 and OP to 400. We have compared our algorithm with four baseline algorithms: a multi-class LLGC (M-LLGC), a binarized LLGC (B-LLGC), a binarized LLGC that uses only boosting (BB-LLGC) and a binarized LLGC that uses only oversampling (BO-LLGC). For the latter two algorithms N was set to 3. BC was set to 100 for BB-LLGC. For BO-LLGC N_1 and OP was set to 0 and 400 respectively. The test accuracy for each data set was averaged over 10 trials. Each data set was divided into partitions with 10%, 20%, 30%, 40% and 50% labeled data which were selected randomly. The rest of the data were treated as unlabeled. The task of the classifier was to find the label of the unlabeled data set i.e., transductive learning.

C. Supervised Classification Results

Table II gives the accuracy results for supervised classification. Table III shows the G-mean values for the same experiment. BBO-NN is the NN classifier that uses the BBO framework. It can be seen from Table II that neither M-NN nor BB-NN perform better than the other binarization algorithms overall. But though the base algorithm that we have incorporated in our framework was not the best overall algorithm in terms of accuracy, BBO-NN provides the best overall accuracy. The table also shows that both BB-NN and BO-NN gives a higher accuracy than B-NN. Moreover, the overall G-mean value also increases as evident from Table III. So, it appears that a class imbalance problem does occur due to binarization. Boosting and oversampling can help to handle this class imbalance problem. But the overall accuracy and G-mean is higher when both are used in tandem. So, our assumption that boosting alone can lead to overfitting and oversampling alone may mislead because of the synthetic data appears to be correct also. For statistical analysis we

TABLE IV
SEMISUPERVISED CLASSIFICATION ACCURACY FOR 10% LABELED DATA.

	Accuracy					G-mean				
Data Set	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC
autos	28.89	27.78	43.47	27.78	43.47	0.2978	0.2919	0.4706	0.2919	0.4706
car	70.21	74.20	74.20	74.20	74.20	0.6947	0.7348	0.7348	0.7348	0.7348
cleveland	54.18	38.43	49.29	38.43	49.29	0.3374	0.2341	0.2930	0.2341	0.2930
dermatology	29.35	39.72	51.39	51.39	51.39	0.2926	0.3962	0.5138	0.5138	0.5138
ecoli	40.26	30.26	41.45	41.45	41.45	0.3736	0.2917	0.4110	0.4090	0.4110
flare	30.24	27.87	61.31	27.87	61.31	0.0324	0.0385	0.1120	0.0385	0.1120
glass	32.90	28.86	39.64	28.86	39.64	0.3280	0.3194	0.4566	0.3194	0.4566
led7digit	15.16	11.00	48.38	11.00	48.38	0.1540	0.1121	0.5090	0.1121	0.5090
lymphography	52.69	58.28	58.13	58.28	58.13	0.5252	0.6425	0.6533	0.6425	0.6533
nursery	38.44	46.87	63.92	46.87	64.02	0.0599	0.0976	0.2425	0.0976	0.2633
pageblocks	90.04	90.04	90.04	90.04	90.04	0.5381	0.7046	0.7046	0.7046	0.7046
penbased	9.52	10.29	47.78	10.29	47.78	0.0941	0.1026	0.5021	0.1026	0.5021
satimage	21.49	35.18	53.59	35.18	53.59	0.2109	0.3507	0.5390	0.3507	0.5390
segment	15.82	18.47	58.44	18.47	58.44	0.0234	0.0163	0.1562	0.0163	0.1562
shuttle	78.43	80.34	80.34	80.34	80.34	0.6794	0.6978	0.6978	0.6978	0.6978
vehicle	24.91	36.61	39.80	36.61	39.80	0.2420	0.3549	0.3811	0.3549	0.3811
vowel	8.53	8.88	29.08	8.88	29.08	0.0818	0.0668	0.2810	0.0668	0.2810
zoo	20.88	25.71	57.33	42.86	57.33	0.1971	0.2505	0.5685	0.4175	0.5685
Average	36.77	38.27	54.87	40.49	54.87	0.2868	0.3168	0.4570	0.3392	0.4582

TABLE V
SEMISUPERVISED CLASSIFICATION ACCURACY FOR 20% LABELED DATA.

	Accuracy					G-mean				
Data Set	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC
autos	26.88	30.39	47.42	30.39	47.42	0.2654	0.3297	0.5201	0.3297	0.5201
car	70.27	77.32	77.32	77.32	77.32	0.3870	0.4550	0.4550	0.4550	0.4550
cleveland	54.20	41.85	51.22	41.85	51.22	0.2698	0.2558	0.3311	0.2558	0.3311
dermatology	31.29	41.54	57.70	57.70	57.70	0.3010	0.4096	0.5754	0.5754	0.5754
ecoli	42.64	38.14	54.61	54.61	54.61	0.3614	0.3642	0.5385	0.5385	0.5385
flare	30.79	28.46	57.14	28.46	57.14	0.0213	0.0325	0.0896	0.0325	0.0896
glass	33.72	36.69	42.62	36.69	42.62	0.3477	0.4081	0.4805	0.4081	0.4805
led7digit	16.68	10.00	50.37	10.00	50.37	0.1686	0.1018	0.5078	0.1018	0.5078
lymphography	52.86	57.39	56.47	57.39	56.47	0.4739	0.6406	0.5747	0.6406	0.5747
nursery	32.70	25.04	66.10	25.04	66.10	0.0522	0.0588	0.2692	0.0588	0.2692
pageblocks	89.86	89.86	89.86	89.86	89.86	0.6465	0.6465	0.6465	0.6465	0.6465
penbased	9.23	10.56	48.13	10.56	48.13	0.0919	0.1057	0.4784	0.1057	0.4784
satimage	22.37	28.08	55.57	28.08	55.57	0.2150	0.2765	0.5346	0.2765	0.5346
segment	15.11	15.51	58.53	15.51	58.53	0.0337	0.0123	0.1641	0.0123	0.1641
shuttle	78.32	80.17	72.34	80.17	80.17	0.6450	0.7506	0.6401	0.7506	0.7506
vehicle	24.12	31.31	32.25	31.31	32.25	0.2295	0.2983	0.3095	0.2983	0.3095
vowel	8.31	9.12	33.65	9.12	33.65	0.0822	0.0912	0.3374	0.0912	0.3374
zoo	37.90	32.35	63.46	32.35	63.46	0.3244	0.3090	0.6176	0.3090	0.6176
Average	37.62	37.99	56.38	39.80	56.81	0.2731	0.3081	0.4483	0.3270	0.4545

performed the Wilcoxon signed-rank test [34]. The p-value returned are 0.0002 and 0.0059 for accuracy and G-mean respectively for B-NN vs BBO-NN. So, the null hypothesis can be rejected as we have chosen the α-value as 0.05 and BBO-NN can be said to perform better than B-NN.

D. Semi-supervised Classification Results

Table IV-VIII shows the average accuracy and G-mean of our framework and the different base classifiers for 10 runs. BBO-LLGC is the LLGC classifier that uses the BBO framework. As it can be seen from this information BBO-LLGC provides the best overall accuracy and G-mean in all five cases. For most of the data sets our framework gives the best accuracy and G-mean value. But it can also be observed from the tables that for some data sets (like cleaveland) the best accuracy was achieved with M-LLGC. Furthermore, it can be noted that using only boosting with binarization (BB-LLGC) gives an overall accuracy close to our framework. Moreover, oversampling along with binarization (BO-LLGC) gives better accuracy than binarization alone (B-LLGC). This shows that boosting or oversampling can handle the class imbalance problem occurring from binarization. But the best result can only be obtained by combining them both. It can also be seen that the difference between the accuracy of the different algorithms remains almost the same for different percentage of labeled data. Wilcoxon's signed-rank test returned p-values 0.0004, 0.0004, 0.0005, 0.0005 and 0.0005 respectively for accuracy for B-LLGC vs BBO-LLGC. For G-mean the values were 0.0003, 0.0006, 0.0008, 0.0005 and 0.0005 respectively. So, for α-value 0.05 the null hypothesis can be rejected and BBO-LLGC can be said to have outperformed B-LLGC in terms of both accuracy and

TABLE VI

SEMISUPERVISED CLASSIFICATION ACCURACY FOR 30% LABELED DATA.

	Accuracy					G-mean				
Data Set	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC
autos	27.55	22.36	47.64	22.36	47.64	0.2800	0.2328	0.4585	0.2328	0.4585
car	70.23	77.64	77.64	77.64	77.64	0.5985	0.6367	0.6367	0.6367	0.6367
cleveland	53.13	49.19	49.19	49.19	49.19	0.2619	0.2403	0.2403	0.2403	0.2403
dermatology	31.30	41.58	62.68	62.68	62.68	0.3070	0.4095	0.6247	0.6247	0.6247
ecoli	43.79	30.27	47.58	48.13	62.41	0.3325	0.2922	0.3647	0.4729	0.5260
flare	31.27	29.56	56.74	29.56	56.74	0.0251	0.0325	0.0890	0.0325	0.0890
glass	33.01	36.64	42.59	36.64	42.59	0.3399	0.3834	0.4475	0.3834	0.4475
led7digit	12.10	8.44	48.02	8.44	48.02	0.1221	0.0405	0.4841	0.0405	0.4841
lymphography	51.52	54.34	57.88	54.34	57.88	0.5172	0.5387	0.5097	0.5912	0.5042
nursery	32.18	32.50	84.32	32.50	84.32	0.0516	0.0590	0.4474	0.0590	0.4343
pageblocks	90.00	90.00	90.00	90.00	90.00	0.7338	0.7338	0.7338	0.7338	0.7338
penbased	9.33	10.42	47.58	10.42	47.58	0.0932	0.1043	0.4729	0.1043	0.4729
satimage	22.12	39.63	55.76	39.63	55.76	0.2138	0.3926	0.5364	0.3926	0.5364
segment	15.48	14.23	58.69	14.23	58.69	0.0397	0.0113	0.1651	0.0113	0.1651
shuttle	78.38	79.18	79.18	79.18	79.18	0.6520	0.6520	0.6520	0.6520	0.6520
vehicle	24.29	38.05	40.95	38.05	40.95	0.2326	0.3634	0.3913	0.3634	0.3913
vowel	8.02	9.44	34.85	9.44	34.85	0.0790	0.0945	0.3432	0.0945	0.3432
zoo	36.18	38.24	67.06	38.24	67.06	0.3149	0.3721	0.5740	0.3721	0.5740
Average	37.21	38.98	58.24	41.15	59.07	0.2886	0.3105	0.4540	0.3354	0.4619

TABLE VII

SEMISUPERVISED CLASSIFICATION ACCURACY FOR 40% LABELED DATA.

	Accuracy					G-mean				
Data Set	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC
autos	27.38	27.50	33.50	27.50	48.13	0.2992	0.2896	0.3615	0.2896	0.4500
car	69.79	77.81	77.81	77.81	77.81	0.4868	0.5428	0.5428	0.5428	0.5428
cleveland	54.83	46.85	46.85	46.85	46.85	0.3253	0.3222	0.3222	0.3222	0.3222
dermatology	31.17	52.15	61.43	64.47	65.47	0.3083	0.5198	0.6140	0.6177	0.6231
ecoli	40.89	43.51	42.62	47.12	55.12	0.3901	0.4275	0.4162	0.4496	0.5215
flare	31.16	31.16	53.28	31.16	53.28	0.0313	0.0313	0.0694	0.0313	0.0694
glass	30.65	29.07	35.14	29.07	42.14	0.3431	0.3198	0.3754	0.3253	0.4027
led7digit	21.32	8.56	47.60	8.56	47.60	0.2184	0.0871	0.4809	0.0871	0.4839
lymphography	54.86	58.78	65.27	58.78	65.27	0.6163	0.6604	0.6913	0.6604	0.6913
nursery	31.82	22.79	77.55	22.79	77.55	0.0624	0.0515	0.4048	0.0515	0.4048
pageblocks	89.85	89.85	89.85	89.85	89.85	0.7032	0.7032	0.7032	0.7032	0.7032
penbased	9.05	10.25	48.57	10.25	48.57	0.0904	0.1023	0.4828	0.1023	0.4828
satimage	21.68	35.43	56.02	35.43	56.02	0.2144	0.3521	0.5567	0.3504	0.5395
segment	16.25	14.03	58.87	14.03	58.87	0.0549	0.0125	0.1674	0.0125	0.1674
shuttle	78.90	79.84	79.84	79.84	79.84	0.7515	0.7323	0.7323	0.7323	0.7323
vehicle	23.83	37.54	40.57	37.54	40.57	0.2272	0.3590	0.3888	0.3590	0.3888
vowel	7.62	9.60	34.97	9.60	34.97	0.0762	0.0960	0.3440	0.0960	0.3440
zoo	42.75	32.16	63.53	32.16	63.53	0.4124	0.3103	0.6183	0.3103	0.6183
Average	37.99	39.27	56.29	40.16	58.41	0.3117	0.3289	0.4596	0.3358	0.4716

G-mean value.

V. CONCLUSION

In this paper, we proposed a framework called BBO based on OVA. The main feature of this framework is that it can handle the class imbalance problem arising from binarization. BBO framework can be used with any supervised algorithm. We also implemented the BBO framework with semi-supervised algorithm which to the best of our knowledge has not used has not been used with binarization before. We perform extensive experimental study of BBO framework with a large number of benchmark data sets. Our experimental results show that using the BBO framework can increase the accuracy of a classifier significantly. In future we would like to use more classifiers to gain more insight about our framework. Furthermore, we would like to venture into the domain of unsupervised classification.

Acknowledgments. The work has been done in the Computer Science & Engineering Department of Bangladesh University of Engineering and Technology (BUET). The authors would like to acknowledge BUET for its generous support.

REFERENCES

[1] J. Fürnkranz, "Round robin classification," *Journal of Machine Learning Research*, vol. 2, pp. 721–747, 2002.

[2] J. Furnkranz, "Round robin ensembles," *Intelligent Data Analysis*, vol. 7, no. 5, pp. 385–403, 2003.

[3] M. Galar, A. Fernández, E. B. Tartas, H. B. Sola, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.

TABLE VIII

SEMISUPERVISED CLASSIFICATION ACCURACY FOR 50% LABELED DATA.

	Accuracy					G-mean				
Data Set	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC	M-LLGC	B-LLGC	BB-LLGC	BO-LLGC	BBO-LLGC
autos	27.75	27.75	27.75	27.75	52.75	0.2783	0.2783	0.2783	0.2783	0.5321
car	69.79	78.54	78.54	78.54	78.54	0.4651	0.5346	0.5346	0.5346	0.5346
cleveland	53.83	54.50	54.50	54.50	54.50	0.3146	0.3598	0.3598	0.3598	0.3598
dermatology	29.44	54.77	63.75	68.75	68.75	0.2302	0.5312	0.6320	0.6852	0.6852
ecoli	42.44	27.62	49.57	50.13	65.43	0.3069	0.2579	0.4745	0.4782	0.6228
flare	31.14	27.82	55.48	27.82	55.48	0.0314	0.0330	0.0891	0.0330	0.0891
glass	33.46	31.21	37.18	31.21	43.25	0.3528	0.3493	0.4032	0.3493	0.4661
led7digit	16.64	9.16	49.44	9.16	49.44	0.1682	0.0927	0.4989	0.0939	0.4995
lymphography	54.73	67.97	70.95	67.97	70.95	0.6293	0.7378	0.7622	0.7378	0.7622
nursery	32.21	20.26	75.29	20.26	75.29	0.0638	0.0514	0.4034	0.0514	0.4034
pageblocks	89.31	89.31	89.31	89.31	89.31	0.7396	0.7396	0.7396	0.7396	0.7396
penbased	9.09	10.38	48.13	10.38	48.13	0.0894	0.1034	0.4811	0.1036	0.4814
satimage	24.72	33.91	56.22	33.91	56.22	0.2399	0.3341	0.5575	0.3354	0.5616
segment	16.37	14.21	58.89	14.21	58.89	0.0557	0.0125	0.1678	0.0125	0.1678
shuttle	78.20	78.20	78.20	78.20	78.20	0.7736	0.7736	0.7736	0.7736	0.7736
vehicle	23.78	31.30	33.66	31.30	33.66	0.2227	0.2971	0.3232	0.2984	0.3236
vowel	8.18	9.13	35.12	9.13	35.23	0.0819	0.0909	0.3524	0.0909	0.3535
zoo	37.65	39.41	56.27	39.41	56.27	0.3630	0.3803	0.5471	0.3803	0.5471
Average	37.71	39.19	56.57	41.22	59.46	0.3004	0.3310	0.4655	0.3520	0.4946

[4] J. H. Hong, J. K. Min, U. K. Cho, and S. B. Cho, "Fingerprint classification using one-vs-all support vector machines dynamically ordered with naive bayes classifiers," *Pattern Recognition*, vol. 41, no. 2, pp. 662–671, 2008.

[5] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka, "An improved algorithm for neural network classification of imbalanced training sets," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 962–969, 1993.

[6] J. Friedman, "Another approach to polychotomous classification," Technical report, Stanford University, Department of Statistics, Tech. Rep., 1996.

[7] F. Cutzu, "Polychotomous classification with pairwise classifiers: A new voting principle," in *Multiple Classifier Systems*, 2003, pp. 115–124.

[8] J. Fürnkranz and E. Hüllermeier, "Pairwise preference learning and ranking," in *ECML*, 2003, pp. 145–156.

[9] E. Hüllermeier and S. Vanderlooy, "Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting," *Pattern Recognition*, vol. 43, no. 1, pp. 128–142, 2010.

[10] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," in *NIPS*, 1997.

[11] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *NIPS*, 1999, pp. 547–553.

[12] H. Zhang, D. H. Wang, and C. L. Liu, "Keyword spotting from online chinese handwritten documents using one-vs-all trained character classifier," in *ICFHR*, 2010, pp. 271–276.

[13] P. Xu, Y. Shi, and M. A. Larson, "Tud at mediaeval 2012 genre tagging task: Multi-modality video categorization with one-vs-all classifiers," in *MediaEval*, 2012.

[14] B. Delachaux, J. Rebetez, A. Pérez-Uribe, and H. F. S. Mejia, "Indoor activity recognition by combining one-vs.-all neural network classifiers exploiting wearable and depth sensors," in *IWANN (2)*, 2013, pp. 216–223.

[15] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.

[16] Z. H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, 2005.

[17] Y. Li and M. Guo, "A new relational tri-training system with adaptive data editing for inductive logic programming," *Knowl.-Based Syst.*, vol. 35, pp. 173–185, 2012.

[18] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, vol. 2, p. 3, 2006.

[19] T. Joachims, "Transductive learning via spectral graph partitioning," in *ICML*, 2003, pp. 290–297.

[20] R. Basili, "Learning to classify text using support vector machines: Methods, theory, and algorithms by thorsten joachims," *Computational Linguistics*, vol. 29, no. 4, pp. 655–661, 2003.

[21] S. Chakraborty, "Bayesian semi-supervised learning with support vector machine," *Statistical Methodology*, vol. 8, no. 1, pp. 68–82, 2011.

[22] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NIPS*, 2003.

[23] D. Zhai, H. Chang, S. Shan, X. Chen, and W. Gao, "Multiview metric learning with global consistency and local smoothness," *ACM TIST*, vol. 3, no. 3, p. 53, 2012.

[24] R. G. F. Soares, H. Chen, and X. Yao, "Semisupervised classification with cluster regularization," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 23, no. 11, pp. 1779–1792, 2012.

[25] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "Semiboost: Boosting for semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2000–2014, 2009.

[26] H. Valizadegan, R. Jin, and A. K. Jain, "Semi-supervised boosting for multi-class classification," in *ECML/PKDD (2)*, 2008, pp. 522–537.

[27] K. Chen and S. Wang, "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 129–143, 2011.

[28] A. Asuncion and D. J. Newman, "Uci machine learning repository," 2007.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res. (JAIR)*, vol. 16, pp. 321–357, 2002.

[31] S. Barua, M. M. Islam, X. Yao, and K. Murase, "Mwmote-majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, 2014.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," DTIC Document, Tech. Rep., 1985.

[33] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.

[34] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.