A Novel Data Representation for Effective Learning in Class Imbalanced Scenarios

Sri Harsha Dumpala, Rupayan Chakraborty and **Sunil Kumar Kopparapu**
TCS Reseach and Innovation - Mumbai, India
{d.harsha, rupayan.chakraborty, sunilkumar.kopparapu}@tcs.com

Abstract

Class imbalance refers to the scenario where certain classes are highly under-represented compared to other classes in terms of the availability of training data. This situation hinders the applicability of conventional machine learning algorithms to most of the classification problems where class imbalance is prominent. Most existing methods addressing class imbalance either rely on sampling techniques or cost-sensitive learning methods; thus inheriting their shortcomings. In this paper, we introduce a novel approach that is different from sampling or cost-sensitive learning based techniques, to address the class imbalance problem, where two samples are simultaneously considered to train the classifier. Further, we propose a mechanism to use a single base classifier, instead of an ensemble of classifiers, to obtain the output label of the test sample using majority voting method. Experimental results on several benchmark datasets clearly indicate the usefulness of the proposed approach over the existing state-of-the-art techniques.

1 Introduction

Class imbalance problem occurs when the number of samples in one class (majority class) greatly outnumbers the number of samples in the other class (minority class). Conventional classifiers such as artificial neural networks (ANNs), support vector machines (SVMs), etc., when trained on imbalanced data tend to get biased towards the majority class, ignoring the minority class [Wu and Chang, 2005]. This problem is often encountered in wide range of domains for example, computational biology and bio-informatics [Dubey *et al.*, 2014], anomaly detection [Tavallaee *et al.*, 2010], image annotation [Liu and Chen, 2005], etc, thus signifying the importance of solving this problem.

In literature, several approaches have been proposed to alleviate the effect of class imbalance problem. The most common approach is to use sampling-based methods, which involves either undersampling the majority class [Liu *et al.*, 2009] or oversampling the minority class [Chawla *et al.*, 2002] to obtain a balanced class distribution. These methods suffer from either the loss of useful information (undersampling) or the heavy dependency on the choice of samples to be duplicated (oversampling) which can lead to overfitting. Another approach is to use ensemble-based methods [Zhou, 2012; Galar *et al.*, 2012], which use the sampling-based techniques more effectively by employing an ensemble of classifiers, whose outputs are combined (usually by voting or by fusion of classifier scores) to obtain the final decision. But, an efficient way to combine these ensemble of classifiers is still an open problem [Wu *et al.*, 2017].

As an alternative to the sampling-based techniques, cost-sensitive learning based methods [Thai-Nghe *et al.*, 2010] are developed. These methods use a different cost parameter for the errors associated with the minority and the majority class samples. But selecting an appropriate value of the cost parameter is critical for the performance of these methods.

In this work, we propose a novel approach to address the class imbalance problem in which two arbitrary samples are simultaneously considered from the training set to train the classifier. First, a modified representation of the training set is obtained by considering the feature-based representation of two arbitrary samples simultaneously. Then we select the architecture of the classifier (in this work, multilayer perceptron (MLP) is used as the base classifier) so as to handle the modified data representation format. Finally to test the classifier, we developed a mechanism of combining the test sample with a set of reference samples on which majority voting based decision is obtained using only a single base classifier. We refer to the proposed data representation as simultaneous two sample (s2s) representation, and our approach for addressing the class imbalance problem as simultaneous two sample learning (s2sL) [Dumpala *et al.*, 2017]. The main contributions of this work are

- Novel data representation format to address the class imbalance problem.
- Selecting an appropriate MLP architecture to handle the proposed data representation.
- Mechanism to obtain the output label of the test sample using majority voting method by considering only a single base classifier.

2 Related Work

Previous approaches addressing the class imbalance problem can be classified into three broad categories [Wu *et al.*, 2017] namely, (1) Sampling-based methods, (2) Ensemble-based methods and (3) Cost-sensitive learning based methods.

Sampling-based methods: Generic sampling-based techniques randomly either select a fraction of samples from the majority class (undersampling) [Liu *et al.*, 2009] or oversample the minority class samples using synthetic minority oversampling technique (SMOTE) [Chawla *et al.*, 2002]. Critical SMOTE (CSMOTE) [Nanni *et al.*, 2015] and majority weighted minority oversampling technique (MWMOTE) [Barua *et al.*, 2014] are the most recent advancements of SMOTE. CSMOTE generates synthetic samples by using only the border and edge samples of the minority class whereas MWMOTE uses weighted informative minority class samples to generate synthetic patterns. These methods require significant amount of undersampling or oversampling for higher levels of class imbalance.

Ensemble-based methods: These methods use boosting and bagging algorithms to address the class imbalance problem. Generally, these methods obtain enhanced performance by using sampling-based techniques (for preprocessing the data) prior to the boosting/bagging algorithms [Błaszczyński *et al.*, 2010]. RUSBoost [Seiffert *et al.*, 2010], EUSBoost [Galar *et al.*, 2013] and more recently uncorrelated cost-sensitive multiset learning (UCML)[Wu *et al.*, 2017] are ensemble-based methods using undersampling as a preprocessing step. UCML uses multiset feature learning (MFL) to address highly imbalanced class problem. Hard ensemble (HE) [Nanni *et al.*, 2015] is also an ensemble-based method which uses ensemble of ensembles strategy in which both, undersampling and oversampling based methods are combined to address class imbalance.

Cost-sensitive learning based methods: Cost-sensitive multi-layer perceptron (CSMLP) method [Castro and Braga, 2013] is used to train MLP directly on the imbalanced data, which assigns higher cost to errors corresponding to minority class compared to that of majority class. Granular SVM (GSVM) [Tang *et al.*, 2009] is an asymmetric classifier which do not focus exclusively on assigning a different cost parameter. GSVM uses repetitive undersampling to achieve data cleaning and also to avoid loss of data.

Our approach to address class imbalance problem is motivated by co-ordination learning [Guo *et al.*, 2005], in which samples belonging to two different classes are considered in pairs to train classifiers such as SVMs. Co-ordination learning helps the classifiers to better learn the dependencies existing between samples across different classes [Guo *et al.*, 2005]. We show that learning these dependencies will improve the classifier performance in a class imbalanced scenario. Our approach mainly differs from co-ordination learning in terms of (a) not restricting the pair of samples to belong to different class, (b) re-purposing our data representation to address class imbalance problem, and (c) the approach adopted for testing the classifier, which are explained in Section 3. These are the major contributions of this paper. To the best of our knowledge, this is the first time s2s-based representation is used for addressing the class imbalance problem.

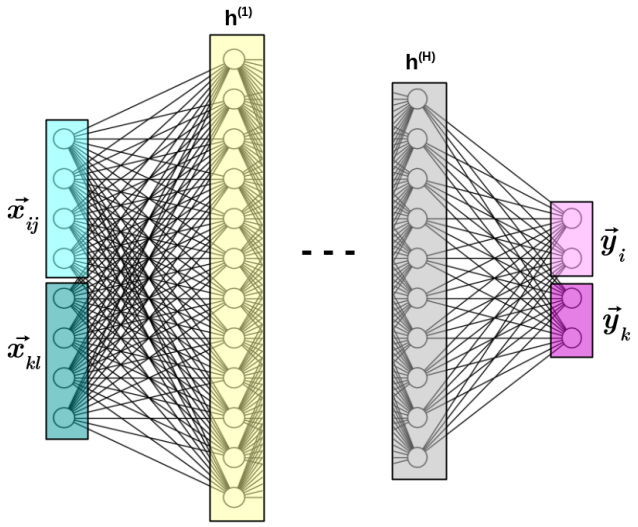

Figure 1: s2s-MLP architecture with 'H' hidden layers.

Proposed approach is very different from other approaches addressing class imbalance problem. Our approach neither eliminates the majority class samples nor generates duplicate/synthetic minority class samples. Moreover, majority voting based output is obtained in our approach using only a single base classifier but not an ensemble of classifiers.

3 Proposed Approach

The proposed approach to address the class imbalance problem is explained in three steps i.e., (1) data representation, (2) classifier training and (3) classifier testing. In this work, we use MLP architecture (as shown in Figure 1) as the base classifier. We refer to this MLP architecture as s2s-MLP.

3.1 Data Representation

Consider a two-class classification task with $C = \{C_1, C_2\}$ denoting the set of class labels, and let N_1 and N_2 be the number of samples corresponding to C_1 and C_2, respectively. In general, to train a classifier, the samples in the training set are provided as an input-output pair as follows.

$$(\vec{x}_{ij}^T, C_i^T), \quad i = 1, 2; \text{ and } \quad j = 1, 2, \cdots, N_i, \qquad (1)$$

where $\vec{x}_{ij} \in \mathbb{R}^{d\times 1}$ refers to the d-dimensional feature vector representing the j^{th} sample in i^{th} class, and $C_i \in C$ refers to output label of i^{th} class. T refers to the transpose of a vector.

In the proposed s2s data representation, we will simultaneously consider two samples as follows.

$$\left([\vec{x}_{ij}, \vec{x}_{kl}]^T, [C_i, C_k]^T\right), \quad \forall i, k = 1, 2; j = 1, 2, \cdots, N_i \text{ and } l = 1, 2, \cdots, N_k, \qquad (2)$$

where $\vec{x}_{ij}$, $\vec{x}_{kl} \in \mathbb{R}^{d\times 1}$ refer to the d-dimensional feature vectors representing the j^{th} sample in i^{th} class and l^{th} sample

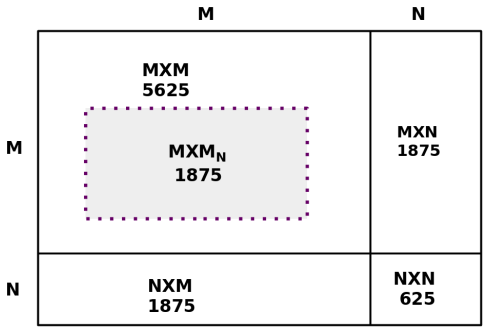

Figure 2: Distribution of class samples in s2s representation.

in the k^{th} class, respectively. $(C_i, C_k) \in C$ refers to the output labels of i^{th} and k^{th} class, respectively.

Hence, in s2s representation, we will have an input feature vector of length $2d$ i.e., $[\vec{x}_{ij}, \vec{x}_{kl}] \in \mathbb{R}^{2d\times 1}$, and output class labels as either $[C_1, C_1], [C_1, C_2], [C_2, C_1]$ or $[C_2, C_2]$. By representing the data in the s2s format, the number of samples in the train set increases from $(N_1 + N_2)$ to $(N_1 + N_2)^2$ samples. Additionally, s2s representation is hypothesized to provide the classifier with a better scope to learn the intra-class and inter-class variations [Guo *et al.*, 2005].

Class imbalance condition: Now let us consider the case where the two classes C_1 and C_2 are imbalanced with C_1 being the majority class and C_2 being the minority class, and $N_1 = M$ and $N_2 = N$ such that $M \gg N$. In s2s-based data representation (as in eq. (2)), the number of samples generated by simultaneously considering two samples is $(M+N)^2$.

The distribution of the samples across different combinations in s2s data representation is explained by taking an example case where $M = 75$ and $N = 25$. The four blocks, as shown in Figure 2, represent the four possible combinations that can be obtained by simultaneously considering the majority and minority class samples i.e., majority-majority, majority-minority, minority-majority and minority-minority class combinations with sizes $M \times M = 5625$, $M \times N = 1875$, $N \times M = 1875$ and $N \times N = 625$, respectively. Note that the total number of samples generated using s2s representation is $(M + N)^2 = 10000$ from $(M + N) = 100$ samples, where the total number of samples carrying majority class information are: $(M \times M) + (M \times N) + (N \times M) = 9375$ while the total samples carrying minority class information are: $(N \times N) + (M \times N) + (N \times M) = 4375$. Note that, here the class imbalance problem still persists.

To overcome this problem, we constrain the majority-majority sample combinations by combining each of the M samples corresponding to class C_1 with only M_N (where $M_N = N$) randomly chosen samples corresponding to class C_1 (as represented by the shaded portion bounded by the dotted line, labeled as $M \times M_N$, in Figure 2). This modifies the number of majority-majority samples in the s2s data representation to $M \times M_N = 1875$ while the number of samples in other class combinations remain the same. This modification in s2s representation is called as majority-constrained s2s results in a total of $(M \times M_N)+(M \times N)+(N \times M) = 5625$ samples carrying majority class information while maintaining a total of 4375 samples carrying minority class information. Thus, the majority-constraint reduces the proportion of samples carrying majority class information.

3.2 Classifier Training

MLP is one of the most common feed forward neural network which has been successfully used in various classification tasks. We will consider MLP as the base classifier to validate our s2s data representation. To train MLP on s2s data representation, the following architecture of MLP (see Figure 1) is considered.

- Input layer have $2 \times d$ linear units to accept the two samples i.e., $\vec{x}_{ij}$ and $\vec{x}_{kl}$, simultaneously.
- The number of units in the hidden layer is selected empirically by varying the hidden units from 2 to $4 \times d$ (twice the length of the input layer) and then selecting the number of units with the best performance on the validation set. Rectified linear units (ReLU) are used as the activation function for the hidden layers. We considered a single hidden layer for most of the experiments.
- The output layer will consist of units equal to twice the considered number of classes in the classification task i.e, the output layer will have four units for two-class classification task. Sigmoid activation function (not softmax) is used for the output units as the output labels in the proposed s2s data representation will have more than one unit active at a time (not one-hot encoded output).

As can be seen from Figure 1, the output layer has outputs $\vec{y}_i$ and $\vec{y}_k$ corresponding to the output labels associated with the input feature vectors $\vec{x}_{ij}$ and $\vec{x}_{kl}$, respectively. For a two-class classification task, the output layer will have 4 output units with the possible output labels as $[0, 1, 0, 1]$, $[0, 1, 1, 0]$, $[1, 0, 0, 1]$, $[1, 0, 1, 0]$ corresponding to the class labels $[C_1, C_1], [C_1, C_2], [C_2, C_1]$ and $[C_2, C_2]$, respectively. Note that in s2sL, s2s-based data representation is used for training the s2s-MLP to address the class imbalance problem.

For training s2s-MLP, we use Adam algorithm with an initial learning rate of 0.001. Binary cross-entropy is used as the cost function. The batch size and other hyper-parameters are selected considering the performance on the validation set.

3.3 Classifier Testing

Generally, the feature vector corresponding to the test sample is provided as input to the trained MLP in the testing phase and the class label is decided based on the output obtained. In case of ensemble-based methods, the same test sample is provided as input to an ensemble of classifiers and the final class label is obtained using majority voting on the outputs obtained from the ensemble of classifiers.

In s2sL method, the feature vector corresponding to the test sample should also be converted to the s2s-based representation for testing the s2s-MLP. We obtain the s2s representation of the test sample by concatenating the test sample with a set of pre-selected reference samples, whose class label is known a priori, as follows.

$$[\vec{t}_i, \vec{r}_j]^T, \quad \forall i = 1, 2, \cdots, S; j = 1, 2, \cdots, R, \qquad (3)$$

Low Imbalanced Datasets							
	Generic data representation				s2s-based data representation		
Dataset	Attributes (d)	#Majority	#Minority	IR	Attributes (d)	#Majority	#Minority
Pima	8	500	268	1.90	16	402000	339824
Glass0	9	144	70	2.01	18	30240	25080
Vehicle0	18	647	199	3.23	36	386259	297107
Ecoli1	7	259	77	3.36	14	59829	45815
Yeast3	8	1321	163	8.11	16	645969	457215
Highly Imbalanced Datasets							
	Generic data representation				s2s-based data representation		
Dataset	Attributes (d)	#Majority	#Minority	IR	Attributes (d)	#Majority	#Minority
Pageblock	10	444	28	15.85	20	37296	25648
Glass5	9	205	9	22.81	18	5535	3771
Yeast5	8	1440	44	32.78	16	190080	128656
Yeast6	8	1449	35	39.15	16	152145	102655
Abalone	8	4142	32	128.87	16	397632	266112

Table 1: Details of the considered imbalanced datasets. (# refers to number of samples).

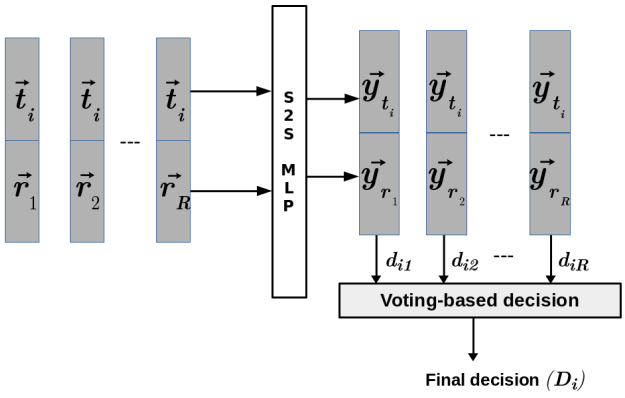

Figure 3: Block diagram for testing s2s-MLP.

where $\vec{t}_i \in \mathbb{R}^{d\times 1}$, $\vec{r}_j \in \mathbb{R}^{d\times 1}$ refer to the d-dimensional feature vector corresponding to the i^{th} test sample and j^{th} reference sample, respectively. S and R refers to the considered number of test and reference samples, respectively. The preselected reference samples can belong to either majority or minority class. But better performance is obtained when the reference samples are selected from the majority class samples. Also, the performance obtained by selecting the reference samples from train set is similar to that obtained when samples are unseen by the network during training.

For testing the s2s-MLP (as shown in Figure 3), each test sample $\vec{t}_i$ is combined with all the R reference samples $(\vec{r}_1, \vec{r}_2, \cdots, \vec{r}_R)$ to form R different instances of the same test sample $\vec{t}_i$. The corresponding outputs $(d_{i1}, d_{i2}, \cdots, d_{iR})$ obtained from s2s-MLP for the R generated instances of $\vec{t}_i$ are combined by majority voting approach to obtain the final decision D_i. The class label with maximum number of votes is considered as the output label. It is important to note that the majority voting based decision is obtained using a single base classifier but not an ensemble of classifiers.

4 Experiments and Results

Experiments are conducted on ten standard two-class imbalanced datasets, with different degree of imbalance, belonging to wide range of domains: computational biology and bio-informatics (Pima, Ecoli, Yeast and Abalone), document classification (Pageblock), image annotation (Vehicle) and object classification (Glass). All datasets used in this work are obtained from KEEL dataset repository [Fernández *et al.*, 2008]. Table 1 summarizes the details of the datasets using original/generic representation and s2s (majority-constrained s2s) based representation. The datasets are listed in the increasing order of their imbalance ratio (IR). Datasets with IR between 1.5 and 9 are considered as relatively low imbalanced datasets, and datasets with IR above 9 are considered as highly imbalanced datasets [Fernández *et al.*, 2008]. Attributes (d) refers to the number of feature attributes. In s2s representation, #Majority and #Minority refer to the number of samples carrying majority class information and minority class information, respectively. We can observe from Table 1 that in generic representation, the majority class samples heavily outnumber the minority class samples in most datasets. Whereas in s2s representation with majority-constraint, #Majority and #Minority are comparable. Further, the actual number of minority class samples are very low in most of the datasets (which make it difficult for the classifiers to learn the characteristics of the minority class) but are increased to a higher value in s2s approach.

For each dataset, we use 5-fold (the folds as provided in the KEEL dataset repository are directly used) cross-validation approach to compare the performance of all the methods considered for analysis. In KEEL dataset repository, each fold is obtained by randomly selecting the samples from the dataset but maintaining the same IR value across all folds as in the original dataset and there is no overlap between any of the two folds. Hence, at any time 80% of the data is used for training (75% as train set and 5% as validation set) and remaining 20% of the data is used for testing. The validation

	Pima		**Glass0**		**Vehicle0**		**Ecoli1**		**Yeast3**	
	F_1	F_2	F_1	F_2	F_1	F_2	F_1	F_2	F_1	F_2
MLP	.63±.03	.56±.02	.51±.02	.48±.01	.91±.02	.90±.02	.70±.03	.67±.03	.60±.04	.55±.03
CSM	.65±.02	.66±.03	.64±.03	.73±.05	.93±.03	.96±.04	.75±.03	.81±.06	.65±.03	.74±.05
MWM	.68±.03	**.71±.04**	**.68±.03**	.74±.04	.94±.03	.96±.03	.74±.03	.78±.05	.69±.03	.81±.07
RUSB	.65±.03	.68±.04	.64±.03	.69±.04	.92±.02	.95±.04	.74±.04	.80±.05	.66±.03	.75±.05
EUSB	.64±.03	.67±.03	.65±.03	.72±.05	.91±.04	.94±.05	.76±.04	.79±.05	.67±.04	.76±.04
HE	.66±.03	.69±.03	.66±.03	.71±.04	.93±.03	.96±.04	.74±.03	.80±.06	.64±.03	.75±.05
UCML	.65±.03	.67±.03	.65±.03	.69±.03	.91±.02	.93±.03	.77±.03	.83±.05	.66±.03	.78±.06
CMLP	.66±.03	.68±.05	.66±.04	**.76±.07**	.92±.03	.96±.03	.76±.03	.82±.06	.68±.02	.78±.05
GSVM	.67±.03	.70±.04	.67±.03	.75±.06	.93±.03	.95±.03	.78±.03	.85±.05	.68±.03	.79±.07
s2sL	**.69±.03**	**.71±.04**	**.68±.03**	.75±.04	**.96±.02**	**.98±.03**	**.79±.03**	**.86±.05**	**.72±.02**	**.83±.05**

Table 2: Experimental results (F_1 and F_2 values with standard deviation) for low imbalanced datasets.

	Pageblock		**Glass5**		**Yeast5**		**Yeast6**		**Abalone19**	
	F_1	F_2	F_1	F_2	F_1	F_2	F_1	F_2	F_1	F_2
MLP	.73±.04	.56±.03	.67±.05	.58±.03	.49±.04	.42±.02	.21±.03	.16±.02	0±0	0±0
CSM	.84±.02	.85±.05	.72±.03	.79±.05	.55±.03	.63±.04	.28±.03	.37±.05	.04±.01	.07±.01
MWM	.88±.03	.88±.06	.75±.03	.81±.05	.56±.03	.68±.06	.25±.02	.39±.06	.05±.01	.10±.02
RUSB	.86±.03	.87±.05	.70±.03	.77±.06	.54±.03	.60±.04	.28±.03	.38±.05	.04±.01	.09±.01
EUSB	.87±.02	.89±.06	.76±.04	.83±.06	.57±.03	.67±.05	.32±.03	.46±.07	.06±.01	.10±.01
HE	.85±.03	.87±.05	.74±.04	.82±.05	.56±.03	.65±.05	.28±.04	.37±.06	.05±.02	.11±.02
UCML	.91±.03	.92±.05	.81±.03	.88±.07	.58±.02	.76±.06	.34±.04	.48±.05	.07±.01	.12±.01
CMLP	.88±.03	.89±.07	.76±.04	.86±.08	.57±.03	.73±.07	.27±.04	.41±.06	.06±.01	.09±.02
GSVM	.89±.02	.91±.05	.75±.03	.83±.06	.60±.03	.69±.05	.30±.03	.46±.05	.07±.01	.11±.02
s2sL	**.95±.03**	**.96±.04**	**.88±.02**	**.93±.04**	**.69±.02**	**.79±.06**	**.44±.03**	**.58±.06**	**.08±.01**	**.15±.03**

Table 3: Experimental results (F_1 and F_2 values with standard deviation) for highly imbalanced datasets.

set is used for selecting network architecture and for hyper-parameter tuning. It is to be noted that the s2s-based representation is obtained separately on the train, validation and test sets, respectively. Majority-constrained s2s-based data representation is obtained on train set considering $M_N = N$; and for the test set, majority voting is used, where $R = 20$ randomly selected majority class samples from the training set are used as the pre-selected reference set. The choice of $R = 20$ is obtained empirically by experimenting with different values of R.

We employed the widely used $F - measure$ (F_β) as a metric which is defined as:

$$F_\beta = (1 + \beta^2)\frac{Precision \times Recall}{(\beta^2)Precision + Recall}, \qquad (4)$$

where $Precision$ refers to the ratio of number of minority class samples correctly classified to the total number of samples classified as minority class by the classifier, and $Recall$ refers to the ratio of the number of minority class samples that are correctly classified to the total number of minority class samples in the test set. In eq. (4), a higher value of β indicates higher importance to $Recall$ over $Precision$. In this paper, we will consider the widely used F_1 (when $\beta = 1$) along with F_2 (when $\beta = 2$) as performance metrics, as previously used in [Maratea *et al.*, 2014; Wu *et al.*, 2017]. The higher the values of F_β, the better is the performance of the system. F_β values reported in this paper are the mean scores obtained from the 5-fold cross validation.

Methods for comparison: We compare the performance of our proposed s2sL based approach with nine state-of-the-art methods, namely, Sampling-based methods (CSMOTE (CSM) [Nanni *et al.*, 2015], MWMOTE (MWM) [Barua *et al.*, 2014]), ensemble-based methods (RUSBoost (RUSB) [Seiffert *et al.*, 2010], EUSBoost (EUSB) [Galar *et al.*, 2013], UCML [Wu *et al.*, 2017] and Hard-Ensemble (HE) [Nanni *et al.*, 2015]) and cost-sensitive learning based methods (CSMLP (CMLP) [Castro and Braga, 2013] and GSVM [Tang *et al.*, 2009]).

Further, a conventional MLP is also considered to show the significance of s2s-based data representation in handling the class imbalance problem. Proposed s2sL along with MLP and CSMLP techniques are implemented using Keras deep learning toolkit [KER, 2016], and most of the other methods are implemented as provided in KEEL software [Fernández *et al.*, 2008].

4.1 Results and Analysis

Experimental results obtained on low and highly imbalanced datasets are provided in Table 2 and Table 3, respectively. It is to be noted that the results for the proposed approach (in Table 2 and 3) are obtained by considering majority-constrained s2sL method. It can be observed that the s2sL approach outperforms all the state-of-the-art methods across all the datasets (for both low and highly imbalanced datasets). Also, s2sL achieves a huge improvement in performance com-

Dataset	Minority class	Majority class
Pima	76.1	80.4
Glass0	78.2	79.1
Vehicle0	99.0	97.63
Ecoli1	94.1	90.9
Yeast3	87.6	95.1
Pageblock	96.6	99.8
Glass5	100.0	98.8
Yeast5	93.9	98.1
Yeast6	86.1	95.2
Abalone	69.1	85.8

Table 4: Class-wise accuracies (in %) obtained using s2sL approach.

Dataset	s2sL$_{\overline{C}}$	s2sL$_C$	ΔF_1
Pageblock	.87	**.95**	0.08
Glass5	.80	**.88**	0.08
Yeast5	.57	**.69**	0.12
Yeast6	.35	**.44**	0.09
Abalone	.06	**.08**	0.02

Table 5: F_1 values for different variants of s2sL (i.e., without majority-constraint (s2sL$_{\overline{C}}$) and with majority-constraint (s2sL$_C$)).

pared to the other methods on highly imbalanced datasets. This is evident from the large difference in F_β values obtained between s2sL and the other methods, for highly imbalanced datasets. Higher F_1 and F_2 values obtained by the s2sL approach compared to the other methods signifies that the proposed approach not only classifies the minority class samples accurately, but also has a good performance on the majority class samples which is reflected by the higher values of F_2. The class-wise accuracies (in %) obtained for highly imbalanced datasets using s2sL (as provided in Table 4) further validate this analysis. These class-wise accuracies are obtained using 5-fold cross validation. Moreover, the results in Tables 2 and 3 are analyzed by conducting statistical tests [Hodges *et al.*, 1962; Demšar, 2006]. These test results indicate that the s2sL approach makes a statistically significant difference in comparison to other methods.

The enhanced performance of s2sL compared to previous approaches may be attributed to the s2s-based data representation. s2s-based data representation seems to help the classifiers not only to learn the independent class-specific information (as in generic data representation) but in addition the *"similarities and differences"* that might exist between the classes [Guo *et al.*, 2005].

4.2 Significance of Majority Constraint

In the proposed s2sL approach, constraining the [majority, majority] combinations is one of the main component. Here, we will discuss the significance of this component on the performance of s2sL approach. Proposed approach without considering majority constraint (i.e., $M_N = N$) is referred to as s2sL$_{\overline{C}}$. s2sL method considering majority constraint (i.e., $M_N = M$) is represented as s2sL$_C$ (Note: s2sL$_C \equiv$ s2sL) in Table 5.

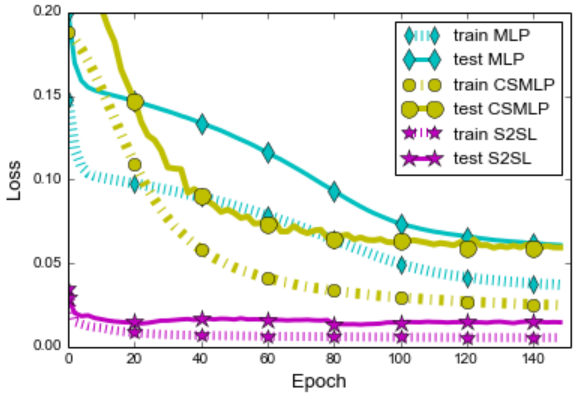

Figure 4: Train and test loss of MLP, CSMLP and s2s-MLP.

Table 5 provides the performance obtained for s2sL$_{\overline{C}}$ and s2sL$_C$ on 5 datasets. ΔF_1 is computed as: ΔF_1 = F_1(s2sL$_C$) - F_1(s2sL$_{\overline{C}}$). It can be observed that the majority constraint improves the performance of the s2sL approach by large margin (ΔF_1 are higher across all datasets as shown in Table 5. This shows the importance of the proposed majority constraint on s2sL for addressing the class imbalance problem.

4.3 Computational Complexity

Generally, the quadratic increase in the number of training samples generated by the proposed s2s-based data representation, and the increase in the number of input and output layer units of the MLP (i.e., s2s-MLP) may raise concerns about the computational complexity of the proposed s2sL approach. Computational time taken by s2sL to train the s2s-MLP for a single epoch may be higher compared to that taken to train MLP and CSMLP, but the s2sL based system converges much faster and to a better local minima compared to MLP and CSMLP as can be observed from Figure 4. Figure 4 shows the loss (binary cross-entropy) obtained on the train and test sets, of Yeast6 dataset, at every epoch for MLP, CSMLP and s2sL based systems. It can be observed from Figure 4 that s2sL requires lower number of epochs (23 epochs) to reach the minima of the training loss (loss = 0.017) compared to MLP (which required 132 epochs to reach the minima of the training loss = 0.062) and CSMLP (which required 124 epochs to reach the minima of the training loss of 0.056). Further, the average training time (in seconds) for convergence (using i5-3210M 3.1GHz cpu with 4-GB RAM) on Yeast6 for different techniques are: 98.7 (s2sL), 38.5 (MLP), 43.7 (CS-MLP), 213.8 (CSM), 95.3 (GSVM), 146.4 (EUSB). This shows that s2s is not that computationally complex compared to other state-of-the-art techniques.

5 Summary and Conclusions

In this paper, we proposed a novel approach of data representation, called s2s to address the class imbalance problem. Our proposed s2sL approach increases the number of training sample instances quadratically by simultaneously considering two samples to train the classifier (MLP). For training,

we considered the architecture of MLP which we referred to as s2s-MLP to handle the s2s data representation. During testing, multiple instances of the same test sample are generated by concatenating the test sample with a set of pre-selected reference samples. Further, in s2sL approach, a single base classifier is used and is sufficient to obtain majority voting based decision on these test sample instances. Experiments conducted on ten bench-mark datasets, with different degrees of imbalance, illustrate the significance of the proposed s2sL approach over the existing state-of-the-art approaches.

References

[Barua *et al.*, 2014] S Barua, MdM Islam, X Yao, and K Murase. Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Trans. on Knowledge and Data Engineering*, 26(2):405–425, 2014.

[Błaszczyński *et al.*, 2010] J Błaszczyński, M Deckert, J Stefanowski, and S Wilk. Integrating selective pre-processing of imbalanced data with ivotes ensemble. In *International Conference on Rough Sets and Current Trends in Computing*, pages 148–157. Springer, 2010.

[Castro and Braga, 2013] CL Castro and AP Braga. Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE Trans. on neural networks and learning systems*, 24(6):888–899, 2013.

[Chawla *et al.*, 2002] NV Chawla, KW Bowyer, LO Hall, and WP Kegelmeyer. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[Demšar, 2006] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[Dubey *et al.*, 2014] R Dubey, J Zhou, Y Wang, PM Thompson, J Ye, et al. Analysis of sampling techniques for imbalanced data: an n= 648 adni study. *NeuroImage*, 87:220–241, 2014.

[Dumpala *et al.*, 2017] Sri Harsha Dumpala, Rupayan Chakraborty, and Sunil Kumar Kopparapu. A novel approach for effective learning in low resourced scenarios. *arXiv preprint arXiv:1712.05608*, 2017.

[Fernández *et al.*, 2008] A Fernández, S García, MJ del Jesus, and F Herrera. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems*, 159(18):2378–2398, 2008.

[Galar *et al.*, 2012] M Galar, A Fernandez, E Barrenechea, H Bustince, and F Herrera. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2012.

[Galar *et al.*, 2013] M Galar, A Fernández, E Barrenechea, and F Herrera. Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition*, 46(12):3460–3471, 2013.

[Guo *et al.*, 2005] Yuhong Guo, Russell Greiner, and Dale Schuurmans. Learning coordination classifiers. In *IJCAI*, pages 714–721, 2005.

[Hodges *et al.*, 1962] JL Hodges, Erich L Lehmann, et al. Rank methods for combination of independent experiments in analysis of variance. *The Annals of Mathematical Statistics*, 33(2):482–497, 1962.

[KER, 2016] François chollet "keras". `https://github.com/fchollet/keras/`, 2016.

[Liu and Chen, 2005] Y Liu and Y Chen. Total margin based adaptive fuzzy support vector machines for multiview face recognition. *IEEE Conference on Systems, Man and Cybernetics*, 2:1704–1711, 2005.

[Liu *et al.*, 2009] X Liu, J Wu, and Z Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

[Maratea *et al.*, 2014] A Maratea, A Petrosino, and M Manzo. Adjusted f-measure and kernel scaling for imbalanced data learning. *Information Sciences*, 257:331–341, 2014.

[Nanni *et al.*, 2015] L Nanni, C Fantozzi, and N Lazzarini. Coupling different methods for overcoming the class imbalance problem. *Neurocomputing*, 158:48–61, 2015.

[Seiffert *et al.*, 2010] C Seiffert, TM Khoshgoftaar, J Van Hulse, and A Napolitano. Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Trans. on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197, 2010.

[Tang *et al.*, 2009] Y Tang, Y Zhang, NV Chawla, and S Krasser. Svms modeling for highly imbalanced classification. *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):281–288, 2009.

[Tavallaee *et al.*, 2010] M Tavallaee, N Stakhanova, and AA Ghorbani. Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(5):516–524, 2010.

[Thai-Nghe *et al.*, 2010] N Thai-Nghe, Z Gantner, and L Schmidt-Thieme. Cost-sensitive learning methods for imbalanced data. In *Neural Networks, The International Joint Conference on*, pages 1–8. IEEE, 2010.

[Wu and Chang, 2005] G Wu and EY Chang. Kba: Kernel boundary alignment considering imbalanced data distribution. *IEEE Trans. on knowledge and data engineering*, 17(6):786–795, 2005.

[Wu *et al.*, 2017] F Wu, X Jing, S Shan, W Zuo, and J Yang. Multiset feature learning for highly imbalanced data classification. In *AAAI*, pages 1583–1589, 2017.

[Zhou, 2012] Z Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.