





Integrating Unsupervised Clustering and Label-Specific Oversampling to Tackle Imbalanced Multi-Label Data

Payel Sadhukhan¹^a, Arjun Pakrashi²^b, Sarbani Palit³^c and Brian Mac Namee²^d

¹*Institute for Advancing Intelligence, TCG CREST, Kolkata, India*

²*School of Computer Science, University College, Dublin, Ireland*

³*Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, India*

Keywords: Multi-Label, Imbalanced Learning, Unsupervised Clustering, Oversampling.

Abstract: There is often a mixture of very frequent labels and very infrequent labels in multi-label datasets. This variation in label frequency, a type class imbalance, creates a significant challenge for building efficient multi-label classification algorithms. In this paper, we tackle this problem by proposing a minority class oversampling scheme, UCLSO, which integrates Unsupervised Clustering and Label-Specific data Oversampling. Clustering is performed to find out the key distinct and locally connected regions of a multi-label dataset (irrespective of the label information). Next, for each label, we explore the distributions of minority points in the cluster sets. Only the intra-cluster minority points are used to generate the synthetic minority points. Despite having the same cluster set across all labels, we will use the label-specific class information to obtain a variation in the distributions of the synthetic minority points (in congruence with the label-specific class memberships within the clusters) across the labels. The training dataset is augmented with the set of label-specific synthetic minority points, and classifiers are trained to predict the relevance of each label independently. Experiments using 12 multi-label datasets and several multi-label algorithms shows the competency of the proposed method over other competing algorithms in the given context.

1 INTRODUCTION


In a multi-label dataset, a single datapoint is associated with more than one relevant label. This type of data is obtained naturally from real-world domains like text (Joachims, 1998; Godbole and Sarawagi, 2004), bioinformatics (Barutcuoglu et al., 2006), video (Qi et al., 2007), images (Boutell et al., 2004; Nasierding et al., 2009; Li et al., 2014) and music (Li and Ogihara, 2006). We denote a multi-label dataset as $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, 2, \dots, n\}$. Here, \mathbf{x}_i is the i^{th} input datapoint in d dimensions, and $\mathbf{y}_i = \{y_{i1}, y_{i2}, \dots, y_{iq}\}$ is the corresponding label assignment for \mathbf{x}_i among the possible q labels. y_{ik} indicates if the k^{th} label is applicable (or *relevant*) for the i^{th} datapoint: $y_{ik} = 1$ denotes that the k^{th} label is relevant to x_i , and $y_{ik} = 0$ denotes that the k^{th} label is not applicable, or is *irrelevant*, to x_i . The target of


multi-label learning is to build a model that can correctly predict all of the relevant labels for a datapoint \mathbf{x}_i .


Multi-label datasets are often found to possess an imbalance in the representation of the different labels—some labels are relevant to a very large number of datapoints while other labels are only relevant to a few. The quantitative disproportion in the representation of different classes of a dataset is known as class imbalance problem (Das et al., 2022). In a binary class-imbalanced dataset, the over-represented and the under-represented classes of the dataset are termed as the majority class and the minority class respectively. Let the set of the majority points and the set of the minority points in an imbalanced dataset be denoted by Maj and Min . *Imbalance ratio* quantifies the degree of disproportion in a dataset and it is represented as follows.


$$\text{imbalance ratio} = \frac{|Maj|}{|Min|} \quad (1)$$

In a multi-label dataset, the number of labels, \mathcal{L} is greater than 1. We have a positive class and a negative

^a <https://orcid.org/0000-0001-7795-3385>

^b <https://orcid.org/0000-0002-9605-6839>

^c <https://orcid.org/0000-0002-4105-6452>

^d <https://orcid.org/0000-0003-2518-0274>

class for each label. Often, the labels in a multi-label dataset have widely varying imbalance ratios and this is a challenging aspect for building multi-label classification models which will work good on all imbalance ratios.

Addressing label-specific imbalances to improve multi-label classification is an active field of research, and several methods have been proposed to address this problem (Zhang et al., 2020; Daniels and Metaxas, 2017; Liu and Tsoumakas, 2019; Pereira et al., 2020a). There is, however, room for significant improvement. Label-specific oversampling can be a solution to address the issue of varying label imbalances in multi-label datasets. In this light, we propose UCLSO, which integrates *Unsupervised Clustering and Label Specific data Oversampling*. The essence of the UCLSO approach is to integrate i) label-invariant information — information about the proximity of points (through clustering) and ii) label-specific information — about the class-memberships of the points, to address the issue of class imbalance in multi-label datasets. In this work, i) synthetic minority points are generated from local data clusters (obtained from unsupervised clustering of the feature space), and ii) the cardinality of the label-specific oversampled minority set obtained in a cluster will depend on the cluster’s share of label-specific minority data. In effect, the method oversamples the minority class by focusing on the cluster-specific distributions of the minority instances. The key highlights of our work are,

- We propose UCLSO, a new minority class oversampling method for multi-label datasets, which generates synthetic minority datapoints specifically in the minority regions of the input space.
- UCLSO works towards preserving the intrinsic class distributions of the local clusters. The goal is avoid generating synthetic minority instances in the majority region, or as outliers in the input space.
- UCLSO ensures that the number of synthetic minority points added in a region is in accordance with the original minority density in that region.
- In UCLSO, instances belonging to the clusters are same for all the labels but their label-specific class-information varies. We integrate this label-specific class information (by virtue of memberships) and the physical proximity of the points (obtained through clustering) to generate the *label-specific synthetic minority points*.
- An empirical study involving 12 well-known, real-world multi-label datasets and nine competing methods illustrates the competency of

UCLSO in handling label-specific imbalance of multi-label data over other competing methods.

The remainder of the paper is structured as follows. Section 2 discusses the relevant existing work in the multi-label domain. In Section 3 we first describe the motivations of our approach and then present the steps of the proposed UCLSO algorithm. The experiment design is described in Section 4 and the results of the experiments are discussed in Section 5. Finally, Section 6 concludes the paper and discusses some directions for future work.

2 RELATED WORK

Existing multi-label classification methods are principally classified into two types: i) *Problem transformation* methods that modify the multi-label dataset in different ways such that it can be used with existing multi-class classification algorithms (Tsoumakas et al., 2011; Fürnkranz et al., 2008; Read et al., 2011; Zhang and Zhou, 2013), and ii) *Algorithm adaptation* approaches that modify existing machine learning algorithms to directly handle multi-label datasets (Zhang and Zhou, 2007; Nam et al., 2014; Zhang and Zhou, 2006; Zhang and Zhou, 2013).

Multi-label algorithms can also be categorised based on if and how they take label associations into account, which allows algorithms to be categorised as: i) *first-order*, ii) *second-order* or iii) *higher-order* approaches based on the number of labels that are considered together to train the models. First order approaches do not consider any label association and learn a classifier for each label independently of all other labels (Zhang and Zhou, 2007; Tanaka et al., 2015; Zhang et al., 2018). In second order methods, pair-wise label associations are explored to achieve enhanced learning of multi-label data (Park and Fürnkranz, 2007; Fürnkranz et al., 2008). Higher order approaches considering associations between more than two labels (Boutell et al., 2004). A number of diversified techniques have facilitated higher order label associations through interesting schemes including classifier chains (Cheng et al., 2010; Read et al., 2013), RAKEL (Tsoumakas et al., 2011), random graph ensembles (Su and Rousu, 2015), DM-LkNN (Younes et al., 2008), IBLR-ML+ (Cheng and Hüllermeier, 2009), and Stacked-MLkNN (Pakrashi and Namee, 2017).

In recent years, data transformation has been a popular choice for handling multi-label datasets. The two principal ways of data transformation in multi-label domain are: i) feature extraction or selection, and ii) data oversampling or undersampling. One

of the earliest applications of feature extraction in multi-label learning was through LIFT (Zhang and Wu, 2015), which brought significant performance improvements. Most feature selection or extraction methods select a label-specific feature set for each label to improve the discerning capability of the label specific classifiers. Subsequently, a number of different feature selection and extraction approaches have been proposed (Huang et al., 2018; Xu, 2018; Xu et al., 2016; Li et al., 2017). In (Huang et al., 2019), label specific features are generated and the authors also address the issue of the missing labels. Recently, the class imbalance problem in multi-label learning has received more interest from the researchers. One common approach to handling imbalance is to balance the cardinalities of the relevant and irrelevant classes for each label. One way of achieving this is through the removal of points from the majority class of each label— for example using random undersampling (Tahir et al., 2012) or tomesk-link based undersampling (Pereira et al., 2020b). Another way to achieve this is by adding synthetic minority points to the minority class (Liu and Tsoumakas, 2019; Sadhukhan and Palit, 2019; Charte et al., 2015). Although this approaches have been shown to be effective there is still a lot of room for improvement.

3 UNSUPERVISED CLUSTERING AND LABEL SPECIFIC DATA OVERSAMPLING (UCLSO)

In this section we discuss the motivation and then present the proposed approach: Unsupervised Clustering and Label-Specific data Oversampling (UCLSO).

3.1 Motivation

Let us consider a two-dimensional toy dataset with two labels (1 and 2) shown in Figure 1(a). The imbalance ratios of labels 1 and 2 in this dataset are 24.7 and 14.4 respectively. Figure 1(b) shows 5 clusters in this dataset which are found using k-means. In Figures 1 (c) and (d), we mark the points with respect to their label-specific class memberships. The colours red and blue indicate the majority and the minority class points respectively. Data pre-processing via minority class oversampling is a popular choice to tackle the issue of imbalance in imbalanced datasets (He and Garcia, 2009). In a multi-label dataset, due to spatial and quantitative variation of class-memberships across the labels, we need a label-specific approach.

Figures 1 (e) and (f) show the label-specific SMOTE-based (Chawla et al., 2002) oversampling (synthetic points in yellow) for label 1 and label 2 respectively. It can be seen that SMOTE oversamples the synthetic minority points in the majority-populated regions on a number of occasions for both labels 1 and 2 (highlighted by black circles in Figures 1 (e) and (f)).

In order to achieve an effective learning of a dataset, we need to prevent the majority space encroachment during oversampling. We tackle this issue by clustering (using k-means) the feature space. Clustering the dataset will give us k localized sub-spaces. Oversampling only within each cluster can prevent the majority class encroachment.

This work is motivated by an effort to balance the cardinalities of the minority and majority classes of the labels without encroaching on the majority class spaces, as well as an effort to preserve the underlying distribution of the datapoints.

As indicated in Figures 1 (e) and (f), a generic oversampling for all labels will not be fruitful as different labels have different quantitative and spatial distribution of the minority points. There are two aspects we need to keep in mind. i) Where should we perform the oversampling? To answer this, we cluster the feature space in an unsupervised manner (only the feature attributes of the points are taken into account). ii) If there is more than one subspace in which to perform oversampling, how much should we oversample in each subspace? We look into the distribution of the minority points (label-specific) in the clusters to decide this. The degree of label-specific oversampling in a cluster should be proportional to its original minority class distribution for that label. Figures 1 (g) and (h) show the oversampling on labels 1 and labels 2 through the proposed method UCLSO. The degree of encroachment in the majority class region is much less for UCLSO compared to SMOTE.

3.2 An Intuitive Description of UCLSO

- *We add synthetic minority points to each imbalanced label.*

Our aim is to reduce the bias of our classifier towards the well and over-represented majority class against the quantitatively scarce minority class. To achieve this, a synthetic minority set is selected for each label.

- *What is the modus operandi of the proposed oversampling?*

We randomly select two existing minority points and sample a synthetic minority point at a random location on their direction vector.

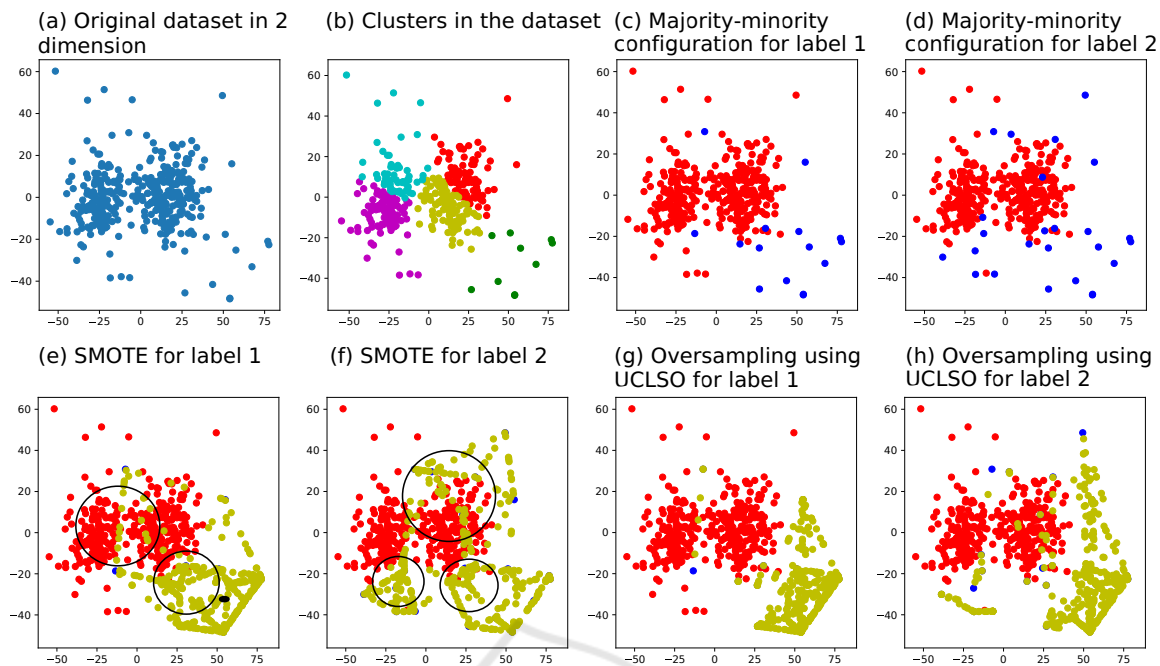


Figure 1: A toy dataset illustrating the problems with oversampling, and how UCLSO addresses them.

- *Where should we initiate the sampling so that we are more confident of sampling in the minority class region rather than encroaching the majority space?*

Adding any synthetic minority point cannot guarantee that it is being added to the minority region in the input space and it is not "encroaching" into the majority class. We can avoid regions of the input space where adding a synthetic minority point is equivalent to adding an outlier or being in a majority region. Intuitively, if the two original minority points involved in the oversampling lie within a small neighbourhood, we can ensure oversampling of a "non-encroaching synthetic minority point".

- *How do we arrange for the two original minority points to lie within a locality?*

A common choice of selecting two points within a locality or neighbourhood is by selecting the first point and then selecting the second from the first one's nearest neighbourhood. But, for a label with high imbalance ratio and sparsely distributed minority points, the neighbours from the same class can lie far apart. In this case, it is highly likely that the neighbours encompass a significant volume of feature space and are not actually local (in spite of being neighbours). Oversampling a synthetic minority point on the direction vectors of two such neighbours can lead to the generation of synthetic minority point in some arbitrary location causing encroachment into the majority spaces.

To tackle this issue, we resort to clustering of the original points. We employ k-means clustering for this purpose and use it in an unsupervised format. By unsupervised, we mean that the clustering process does not involve any class or label information of the points. Clustering is done solely on the basis of inter-point euclidean distances between the points.

After segregating the points into a pre-fixed number of clusters, we inspect the label-specific class-memberships of the points in the clusters. We select two intra-cluster minority points from a cluster and compute the synthetic minority point at a random location on their direction vector. We compute the synthetic minority points from the original minority points lying in the k clusters. The number of synthetic minority points generated from a cluster will depend on the share of original minority points in the clusters.

Let us consider two clusters C_1 and C_2 with differing shares of minority instances for label l . Let the shares of C_1 and C_2 be denoted by s_1 and s_2 respectively such that $s_1 > s_2$. The total number of minority points, minority points in C_1 and minority points in C_2 are denoted by n , n_1 and n_2 respectively. The share of minority instances in a cluster does not depend on the total number of points in a cluster, rather it depends on how many out of the total original minority points we have in that cluster. We will oversample more points in C_1 than C_2 because, C_1 has more original minority

points than that of C_2 , and we have more confidence on adding a point to C_1 over C_2 if we want non-encroachment to the majority class as much as possible. On a similar logic, if we get a cluster with zero minority instances share for some label, we will not oversample any synthetic minority point in that cluster as it represent a majority region.

3.3 Approach: UCLSO

Algorithm 1: UCLSO.

```

1: procedure UCLSO( $\mathcal{D}, k$ )  $\triangleright$   $\mathcal{D}$ : Training dataset,
    $k$ : number of clusters in k-means clustering
2:    $\{C_1, C_2, \dots, C_k\} = \text{k.means}(\{\mathbf{x}_i | 1 \leq i \leq n\}, k)$ 
    $\triangleright$  Cluster the input space
3:   for  $l \in 1, 2, \dots, \mathcal{L}$  do
4:      $\mathcal{S}_l = \{\}$ 
5:     for  $p \in 1, 2, \dots, k$  do
6:        $n_{lp} =$  number of original minority
       points for label  $l$  in  $C_p$ 
7:       _____  $\triangleright$  Find the synthetic
       minority instance shares of each cluster
8:        $min_l = \{\mathbf{x}_i | \forall_i y_{il} = 1\}$ 
9:        $major_l = \{\mathbf{x}_i | \forall_i y_{il} = 0\}$ 
10:       $syn_{lp} = \lceil n_{lp} \times \frac{|major_l| - |min_l|}{|min_l|} \rceil$ ,  $p =$ 
        $1, 2, \dots, k$ 
11:      _____  $\triangleright$  Generate synthetic
       points
12:      for  $j \in 1, 2, \dots, syn_{lp}$  do
13:         $\mathbf{u}_p$  is selected randomly from  $C_p$ 
14:         $\mathbf{v}_p$  is  $\mathbf{u}_p$ 's randomly selected near-
        est neighbor in  $C_p$ 
15:         $r \in (0, 1)$  selected randomly
16:         $\mathbf{s}_{pj}^{(l)} = \mathbf{u}_p + (\mathbf{v}_p - \mathbf{u}_p) \times r$   $\triangleright j^{th}$ 
        synthetic pt. for label  $l$  from  $C_p$ 
17:         $\mathcal{S}_l = \mathcal{S}_l \cup \{(\mathbf{s}_{pj}^{(l)}, y_{pj}^{(l)} = 1)\}$ 
18:      end for
19:    end for
20:     $\mathcal{A}_l = \mathcal{D} \cup \mathcal{S}_l$   $\triangleright$  Augment original data
21:  end for
22:  return  $\{\mathcal{A}_l | l = 1, 2, \dots, q\}$   $\triangleright$  Per label
       augmented synthetic datasets
23: end procedure

```

The main idea of this oversampling method is to generate the synthetic minority points in the minority populated regions of the input space. We follow this scheme to introduce more synthetic minority points in the minority regions, thereby avoiding the introduction of the synthetic minority points in non-minority regions. This should ideally improve the detection of

the minority points with respect to the majority points — as the error optimization in the classifier modelling phase will have equivalent contribution from both the minority points and the majority points. This will in turn help in mitigating the bias of the majority class and learning a better decision boundary for an imbalanced label.

A common approach for generating synthetic minority datapoints is to select two points within a neighbourhood and then generate a synthetic point by interpolation at a random location on the direction vector connecting the two (Chawla et al., 2002). For a label with a high imbalance ratio and sparsely distributed minority points, the neighbours from the same class for this label can lie far apart. Consequently, the neighbourhood can encompass a large volume of feature space. Therefore, oversampling in the given manner may lead to the generation of synthetic minority points which end up in the majority region of the input space.

To tackle this issue, we partition the original points into k clusters $\{C_1, C_2, \dots, C_k\}$, on the basis of their Euclidean distances. We use the k-means algorithm to perform this clustering. Once we get the clusters, for each cluster C_p , we randomly select \mathbf{u}_p , a minority point from a cluster, and \mathbf{v}_p , which is a randomly chosen nearest neighbour of $\mathbf{u}_p \in C_p$ (randomly chosen minority neighbor of \mathbf{u}_p from the same cluster). We compute the synthetic minority point by interpolation at a random location of the direction vector connecting \mathbf{u}_p and \mathbf{v}_p . The synthetic point is computed as follows

$$\mathbf{s}_{pj}^{(l)} = \mathbf{u}_p + (\mathbf{v}_p - \mathbf{u}_p) \times r \quad (2)$$

where $\mathbf{s}_{pj}^{(l)}$ is the j th synthetic datapoint generated in cluster C_p for the label l , and $r \in (0, 1)$ is a random number sampled from the uniform distribution, which decides the location of the synthetic point between \mathbf{u}_p and \mathbf{v}_p .

The number of synthetic minority points generated from a cluster is directly proportional to the share of original minority points in that cluster. Therefore, more synthetic minority points will be introduced in the clusters with more original minority points. This is because, we are more confident about adding minority points in a region which originally had more original minority points. The number of synthetic minority points to be added is computed as.

$$syn_{lp} = \lceil n_{lp} \times \frac{|major_l| - |min_l|}{|min_l|} \rceil \quad (3)$$

where min_l and $major_l$ are the sets of minority and majority datapoints for the label l respectively. Here n_{lp}

is the number of original minority datapoints for label l in cluster C_p . This way, the clusters which have more original minority points will be populated with more synthetic minority point.

Following the above steps, we obtain the synthetic minority set S_l for the label l . The original training dataset \mathcal{D} is appended with S_l to get an augmented dataset \mathcal{A}_l for each label l . This augmented training set, \mathcal{A}_l , is used to train a binary classifier model for the corresponding label l . The above process is summarised in Algorithm 1.

4 EXPERIMENTS

We performed a set of experiments to evaluate the effectiveness of the proposed UCLSO method. This section describes the datasets, algorithms, experimental setup, and evaluation processes used for the experiments.

Several well-known multi-label datasets were selected which are listed in Table 1¹. Here, *instances*, *inputs* and *labels* indicate the total number of datapoints, the number of predictor variables, and the number of potential labels respectively in each dataset. *Type* indicates if the input space is numeric or nominal. *Distinct labelsets* indicates the number of unique combinations of labels. *Cardinality* is the average number of labels per datapoint, and *Density* is achieved by dividing *Cardinality* by the *Labels*.

The datasets are modified as recommended in (Zhang et al., 2020; He and Garcia, 2009). Labels having a very high degree of imbalance (50 or greater) or having too few positive samples (20 in this case) are removed. For text datasets (*medical*, *enron*, *rcv1*, *ibtex*), only the input space features with high degree of document frequencies are retained.

To compare the performance of different approaches, we have selected the label-based macro-averaged F-Score and label-based macro-averaged AUC scores recommended in (Zhang et al., 2020). For the experiments evaluating the proposed algorithm we have performed a 10×2 fold cross-validation experiment. The experiment setup and environment was kept identical to Zhang et al. (Zhang et al., 2020). For clustering, the number of clusters was set to 5 for the k-means step of UCLSO. In the classification phase, a set of linear SVM classifiers are used, one for each label.

We compare the performance of UCLSO against several state-of-the-art multi-label classification algorithms – COCOA (Zhang et al., 2020), THRSEL (Pil-

lai et al., 2013), IRUS (Tahir et al., 2012), SMOTE-EN (Chawla et al., 2002), RML (Pettersson and Caetano, 2010), and binary relevance (BR), calibrated label ranking (CLR) (Fürnkranz et al., 2008), ensemble classifier chains (ECC) (Read et al., 2011) and RAKEL (Tsoumakas et al., 2011). We base our experiments on the experiment presented in Zhang et al. (Zhang et al., 2020), and extend the results of that paper by adding the performance of UCLSO.

5 RESULTS

Tables 2 and 3 shows the label-based macro-average F-Score and label-based macro-averaged AUC results respectively², along with the relative ranks in brackets (lower ranks are better) of the algorithms compared for each dataset. The last row of both tables indicate the average rank for the algorithms. The best values are highlighted in boldface.

Also, to further analyse the differences between the algorithms, we performed a non-parametric statistical test for a multiple classifier comparison test. Following (García et al., 2010), we have performed a Friedman test with Finner p -value adjustments, and the critical difference plots from the test results are shown in Figure 2³.

Table 2 clearly shows that the overall performance of the proposed UCLSO algorithm is better than all the other algorithms, attaining the best average rank of 1.25. The second best rank is attained by COCOA (avg. rank 3). Also, the proposed method UCLSO achieved much better performance than the other approaches for many datasets and attained the top rank for nine of the datasets, and on the remaining three datasets it attained the second rank. these results also show that methods that attempt to explicitly consider the label imbalance issue perform better than those that do not. The other algorithms which specifically address label imbalance attained the following order: RML (avg. rank 3.29), THRSEL (avg. rank 4.62), SMOTE-EN (avg. rank 5.12) and IRUS (arg. rank 6.33). The algorithms which do not consider the label imbalances like BR (avg. rank 7.42), RAKEL (avg. rank 7.5), ECC (avg. rank 8.12), and CLR (avg. rank 8.33) all performed poorly.

Multiple classifier comparison results in Figure 2 show that when UCLSO is compared with other al-

²Note that results for Table 3 does not have the results RML (Pettersson and Caetano, 2010) as the implementation does not provide prediction scores.

³The full result tables in supplementary material: https://github.com/phoxis/uclso/blob/main/UCLSO_Supplementary_Material.pdf

¹<http://mulan.sourceforge.net/datasets-mlc.html>

Table 1: Description of datasets.

Dataset	Instances	Inputs	Labels	Type	Cardinality	Density	Distinct Labelsets	Proportion of Distinct Labelsets	Imbalance Ratio		
									min	max	avg
yeast	2417	103	13	numeric	4.233	0.325	189	0.078	1.328	12.500	2.778
emotions	593	72	6	numeric	1.869	0.311	27	0.046	1.247	3.003	2.146
medical	978	144	14	numeric	1.075	0.077	42	0.043	2.674	43.478	11.236
cal500	502	68	124	numeric	25.058	0.202	502	1.000	1.040	24.390	3.846
rcv1-s1	6000	472	42	numeric	2.458	0.059	574	0.096	3.342	49.000	24.966
rcv1-s2	6000	472	39	numeric	2.170	0.056	489	0.082	3.216	47.780	26.370
rcv1-s3	6000	472	39	numeric	2.150	0.055	488	0.081	3.205	49.000	26.647
enron	1702	50	24	nominal	3.113	0.130	547	0.321	1.000	43.478	5.348
bibtex	7395	183	26	nominal	0.934	0.036	377	0.051	6.097	47.974	32.245
llog	1460	100	18	nominal	0.851	0.047	109	0.075	7.538	46.097	24.981
corel5k	5000	499	44	nominal	2.241	0.050	1037	0.207	3.460	50.000	17.857
slashdot	3782	53	14	nominal	1.134	0.081	118	0.031	5.464	35.714	10.989

Table 2: Each cell indicates the averaged *label-based macro-averaged F-Scores* scores (best score in bold) along with the relative rank of the corresponding algorithm in brackets. The last row indicates the overall average ranks.

	UCLSO	COCOA	THRSEL	IRUS	SMOTE-EN	RML	BR	CLR	ECC	RAKEL
yeast	0.505 (1)	0.461 (3)	0.427 (5)	0.426 (6)	0.436 (4)	0.471 (2)	0.409 (9)	0.413 (8)	0.389 (10)	0.420 (7)
emotions	0.658 (2)	0.666 (1)	0.560 (9)	0.622 (5)	0.575 (8)	0.645 (3)	0.550 (10)	0.595 (7)	0.638 (4)	0.613 (6)
medical	0.783 (1)	0.759 (2)	0.733 (3.5)	0.537 (10)	0.700 (8)	0.707 (7)	0.718 (6)	0.724 (5)	0.733 (3.5)	0.672 (9)
cal500	0.273 (2)	0.210 (5)	0.252 (3)	0.277 (1)	0.235 (4)	0.209 (6)	0.169 (8)	0.081 (10)	0.092 (9)	0.193 (7)
rcv1-s1	0.443 (1)	0.364 (3)	0.292 (5)	0.252 (8)	0.313 (4)	0.387 (2)	0.285 (6)	0.227 (9)	0.192 (10)	0.272 (7)
rcv1-s2	0.432 (1)	0.342 (3)	0.275 (5)	0.234 (8)	0.305 (4)	0.363 (2)	0.272 (6)	0.226 (9)	0.173 (10)	0.263 (7)
rcv1-s3	0.480 (1)	0.339 (3)	0.275 (5)	0.225 (8)	0.302 (4)	0.371 (2)	0.271 (6)	0.211 (9)	0.163 (10)	0.257 (7)
enron	0.352 (1)	0.342 (2)	0.291 (5)	0.293 (4)	0.266 (8)	0.307 (3)	0.246 (9)	0.244 (10)	0.268 (6)	0.267 (7)
bibtex	0.442 (1)	0.318 (3)	0.303 (4)	0.253 (8)	0.283 (5)	0.326 (2)	0.263 (7)	0.265 (6)	0.212 (10)	0.252 (9)
llog	0.181 (1)	0.082 (6)	0.096 (3)	0.124 (2)	0.095 (4.5)	0.095 (4.5)	0.031 (7)	0.024 (8)	0.022 (10)	0.023 (9)
corel5k	0.209 (2)	0.196 (3)	0.146 (4)	0.105 (6)	0.125 (5)	0.215 (1)	0.089 (7)	0.049 (10)	0.054 (9)	0.084 (8)
slashdot	0.443 (1)	0.374 (2)	0.355 (4)	0.257 (10)	0.366 (3)	0.343 (5)	0.291 (8)	0.290 (9)	0.304 (6)	0.296 (7)
Avg. rank	1.25	3.00	4.62	6.33	5.12	3.29	7.42	8.33	8.12	7.5

gorithms, except for COCOA and RML, the null hypothesis can be rejected with a significance level of $\alpha = 0.05$. Therefore, based on the statistical test, UCLSO is significantly better than the other algorithms, except COCOA and RML.

Table 3 shows the label-based macro-averaged AUC scores, which shows that proposed method UCLSO was able to attain the second best average rank of 2.42, being very close to COCOA attaining the best rank of 2.21. Interestingly UCLSO attained more rank ones (six) than COCOA (two rank ones). Also, interestingly ECC was able to perform better than UCLSO in six of the datasets, but was able to perform better in nine datasets when compared to COCOA. It is also interesting to notice that ECC and CLR had higherrankings for the label-based macro-averaged AUC metric than for macro-averaged F-Scores. It seems that a simple BR still performed poorly. As ECC and CLR takes label associations into consideration in a binary relevance and ranking fashion, respectively, it helped improve the comparative performances. RAKEL, on the other hand, taking label associations into account is sensitive on the label subset size (value of k) and the specific combination,

which can lead to an even higher degree of imbalance. The difference in the results of the label-based macro-average AUC compared to the F-Score also indicates the importance of thresholding the predictions when deciding the relevance of a certain label.

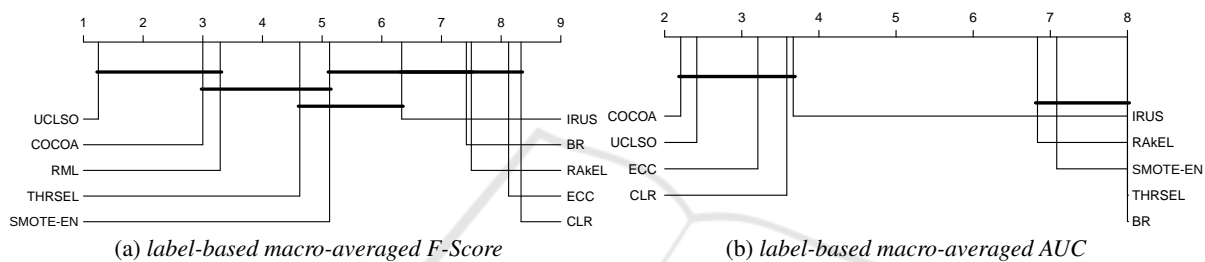
Multiple classifier comparison results show in Figure 2 that when UCLSO is compared with others, the null hypothesis could not be rejected for COCOA, ECC, CLR and IRUS in this case with a significance level of $\alpha = 0.05$. Although, UCLSO performed significantly better than RAKEL, SMOTE-ML, THRSEL and BR. Overall, the experiments demonstrate the effectiveness of the proposed method UCLSO, as it outperforms the compared state of the art algorithms in almost all cases.

6 CONCLUSION AND FUTURE WORK

In this work we have proposed an algorithm to address the class imbalance of labels in multi-label classification problems. The proposed algorithm, Un-

Table 3: Each cell indicates the averaged *Label-based macro-averaged AUC* scores (best score in bold) along with the relative rank of the corresponding algorithm in brackets. The last row indicates average ranks.

	UCLSO	COCOA	THRSEL	IRUS	SMOTE-EN	BR	CLR	ECC	RAKEL
yeast	0.666 (3)	0.711 (1)	0.576 (8.5)	0.658 (4)	0.582 (7)	0.576 (8.5)	0.650 (5)	0.705 (2)	0.641 (6)
emotions	0.819 (3)	0.844 (2)	0.687 (8.5)	0.802 (4)	0.698 (7)	0.687 (8.5)	0.796 (6)	0.850 (1)	0.797 (5)
medical	0.967 (1)	0.964 (2)	0.869 (7.5)	0.955 (3.5)	0.873 (6)	0.869 (7.5)	0.955 (3.5)	0.952 (5)	0.856 (9)
cal500	0.550 (4)	0.558 (2)	0.509 (8.5)	0.545 (5)	0.512 (7)	0.509 (8.5)	0.561 (1)	0.557 (3)	0.528 (6)
rcv1-s1	0.919 (1)	0.889 (3)	0.643 (7.5)	0.882 (4)	0.626 (9)	0.643 (7.5)	0.891 (2)	0.881 (5)	0.728 (6)
rcv1-s2	0.912 (1)	0.882 (2.5)	0.640 (7.5)	0.880 (4)	0.622 (9)	0.640 (7.5)	0.882 (2.5)	0.874 (5)	0.721 (6)
rcv1-s3	0.956 (1)	0.880 (2)	0.633 (7.5)	0.872 (4.5)	0.628 (9)	0.633 (7.5)	0.877 (3)	0.872 (4.5)	0.718 (6)
enron	0.719 (5)	0.752 (1)	0.597 (8.5)	0.738 (3)	0.619 (7)	0.597 (8.5)	0.720 (4)	0.750 (2)	0.650 (6)
bibtex	0.844 (4)	0.877 (2)	0.673 (8.5)	0.894 (1)	0.706 (6)	0.673 (8.5)	0.811 (5)	0.873 (3)	0.696 (7)
llog	0.721 (1)	0.663 (4)	0.518 (7.5)	0.676 (2)	0.561 (6)	0.518 (7.5)	0.612 (5)	0.673 (3)	0.514 (9)
corel5k	0.695 (4)	0.718 (3)	0.559 (7.5)	0.687 (5)	0.596 (6)	0.559 (7.5)	0.740 (1)	0.723 (2)	0.552 (9)
slashdot	0.806 (1)	0.774 (2)	0.632 (8.5)	0.753 (4)	0.714 (6)	0.632 (8.5)	0.742 (5)	0.765 (3)	0.638 (7)
Avg. ranks	2.42	2.21	8.00	3.67	7.08	8.00	3.58	3.21	6.83

Figure 2: Critical difference plots. The scale indicates the average ranks. The methods which are not connected with the horizontal lines are significantly different with a significance level of $\alpha = 0.05$.

supervised Clustering and Label-Specific data Oversampling (UCLSO), oversamples label-specific minority datapoints in a multi-label problem to balance the sizes of the majority and the minority classes of each label. The oversampling of the minority classes for each label is done in a way such that more minority class samples are generated in regions (or clusters) where the density of minority points is high. This avoids the introduction of minority datapoints in majority regions in the input space. The number of samples introduced per cluster also depends on the share of the minority class for that cluster.

An experiment with 12 well-known multi-label datasets and other state of the art algorithms demonstrates the efficacy of UCLSO with respect to label-based macro-averaged F-Score. UCLSO attained the best average rank and the degree of its improvement over existing approaches was significant. This shows that UCLSO has successfully improved the classification of imbalanced multi-label data. In future, we would specifically like to incorporate some imbalance informed clustering to extend our scheme. Moreover, it would be interesting to amalgamate the oversampling technique with label associated learning, another key component of multi-label data.

REFERENCES

- Barutcuoglu, Z., Schapire, R. E., and Troyanskaya, O. G. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836.
- Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771.
- Charte, F., Rivera, A. J., del Jesus, M. J., and Herrera, F. (2015). MLSMOTE: approaching imbalanced multi-label learning through synthetic instance generation. *Knowledge-Based Systems*, 89:385–397.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.
- Cheng, W. and Hüllermeier, E. (2009). Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225.
- Cheng, W., Hüllermeier, E., and Dembczynski, K. J. (2010). Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 279–286.
- Daniels, Z. and Metaxas, D. (2017). Addressing imbalance in multi-label classification using structured hellinger forests. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

- Das, S., Mullick, S. S., and Zelinka, I. (2022). On supervised class-imbalanced learning: An updated perspective and some key challenges. *IEEE Transactions on Artificial Intelligence*, 3(6):973–993.
- Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., and Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Mach. Learn.*, 73(2):133–153.
- García, S., Fernández, A., Luengo, J., and Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information sciences*, 180(10):2044–2064.
- Godbole, S. and Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30.
- He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Trans. on Knowl. and Data Eng.*, 21(9):1263–1284.
- Huang, J., Li, G., Huang, Q., and Wu, X. (2018). Joint feature selection and classification for multilabel learning. *IEEE Transactions on Cybernetics*, 48(3):876–889.
- Huang, J., Qin, F., Zheng, X., Cheng, Z., Yuan, Z., Zhang, W., and Huang, Q. (2019). Improving multi-label classification with missing labels by learning label-specific features. *Information Sciences*, 492:124–146.
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.
- Li, F., Miao, D., and Pedrycz, W. (2017). Granular multi-label feature selection based on mutual information. *Pattern Recognition*, 67:410 – 423.
- Li, T. and Ogihara, M. (2006). Toward intelligent music information retrieval. *Multimedia, IEEE Transactions on*, 8(3):564–574.
- Li, X., Zhao, F., and Guo, Y. (2014). Multi-label image classification with a probabilistic label enhancement model. In *Uncertainty in Artificial Intelligence*.
- Liu, B. and Tsoumakas, G. (2019). Synthetic oversampling of multi-label data based on local label distribution. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 180–193. Springer.
- Nam, J., Kim, J., Mencía, E. L., Gurevych, I., and Fürnkranz, J. (2014). Large-scale multi-label text classification—revisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer.
- Nasierding, G., Tsoumakas, G., and Kouzani, A. Z. (2009). Clustering based multi-label classification for image annotation and retrieval. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 4514–4519.
- Pakrashi, A. and Namee, B. M. (2017). Stacked-MLkNN: A stacking based improvement to multi-label k-nearest neighbours. In *LIDTA@PKDD/ECML*.
- Park, S. and Fürnkranz, J. (2007). Efficient pairwise classification. In *ECML 2007. LNCS (LNAI)*, pages 658–665. Springer.
- Pereira, R. M., Costa, Y. M., and Silla Jr, C. N. (2020a). Mtl: A multi-label approach for the tomek link undersampling algorithm. *Neurocomputing*, 383:95–105.
- Pereira, R. M., Costa, Y. M., and Silla Jr, C. N. (2020b). MLTL: A multi-label approach for the tomek link undersampling algorithm. *Neurocomputing*, 383:95–105.
- Petterson, J. and Caetano, T. S. (2010). Reverse multi-label learning. In *Advances in Neural Information Processing Systems 23*, pages 1912–1920. Curran Associates, Inc.
- Pillai, I., Fumera, G., and Roli, F. (2013). Threshold optimisation for multi-label classifiers. *Pattern Recogn.*, 46(7):2055–2065.
- Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., Mei, T., and Zhang, H.-J. (2007). Correlative multi-label video annotation. In *Proceedings of the 15th ACM International Conference on Multimedia, MM '07*, pages 17–26. New York, NY, USA. ACM.
- Read, J., Martino, L., and Luengo, D. (2013). Efficient monte carlo optimization for multi-label classifier chains. pages 3457–3461.
- Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine learning*, 85(3):333.
- Sadhukhan, P. and Palit, S. (2019). Reverse-nearest neighborhood based oversampling for imbalanced, multi-label datasets. *Pattern Recognition Letters*, 125:813 – 820.
- Su, H. and Rousu, J. (2015). Multilabel classification through random graph ensembles. *Machine Learning*, 99(2).
- Tahir, M. A., Kittler, J., and Yan, F. (2012). Inverse random under sampling for class imbalance problem and its application to multi-label classification. 45(10):3738–3750.
- Tanaka, E. A., Nozawa, S. R., Macedo, A. A., and Baranauskas, J. A. (2015). A multi-label approach using binary relevance and decision trees applied to functional genomics. *Journal of Biomedical Informatics*, 54:85–95.
- Tsoumakas, G., Katakis, I., and Vlahavas, I. (2011). Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089.
- Xu, J. (2018). A weighted linear discriminant analysis framework for multi-label feature extraction. *Neurocomputing*, 275:107–120.
- Xu, J., Liu, J., Yin, J., and Sun, C. (2016). A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously. *Knowledge-Based Systems*, 98:172–184.
- Younes, Z., Abdallah, F., and Denœux, T. (2008). Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies. In *2008 16th European Signal Processing Conference*, pages 1–5. IEEE.

- Zhang, M. and Zhou, Z. (2006). Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18:1338–1351.
- Zhang, M.-L., Li, Y.-K., Liu, X.-Y., and Geng, X. (2018). Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, 12(2):191–202.
- Zhang, M.-L., Li, Y.-K., Yang, H., and Liu, X.-Y. (2020). Towards class-imbalance aware multi-label learning. *IEEE Transactions on Cybernetics*.
- Zhang, M.-L. and Wu, L. (2015). Lift: Multi-label learning with label-specific features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(1):107–120.
- Zhang, M.-L. and Zhou, Z.-H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recogn.*, 40(7):2038–2048.
- Zhang, M.-L. and Zhou, Z.-H. (2013). A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837.

