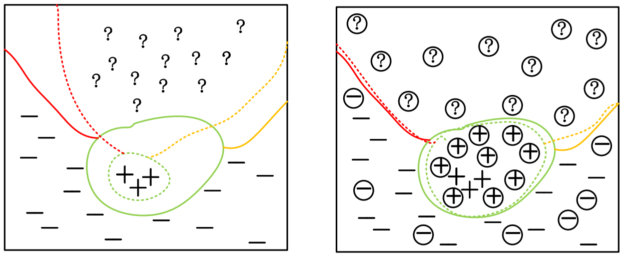
Imbalanced Augmented Class Learning with Unlabeled Data by Label Confidence Propagation

Si-Yu Ding[1,2], Xu-Ying Liu*[1,2], Min-Ling Zhang[1,2]
[1] *School of Computer Science and Engineering, Southeast University, Nanjing, China*
[2] *Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China*
{dingsy, liuxy, zhangml}@seu.edu.cn

***Abstract*—As a practical problem in open and dynamic environments, class-incremental learning has attracted much attention from many fields. Learning with augmented class (LAC) problem formulates one of the core difficulties of class-incremental learning: instances of augmented class need to be predicted with the restriction that only examples from seen classes are observed in training phase. LACU framework advances the study of LAC problem by exploiting unlabeled data, while it does not take into account an important practical problem widely-existing in real-world applications of LAC – imbalanced class distributions among seen classes, which will further increase the learning difficulties of LAC problem. We propose a novel approach Label Confidence Propagation (LCP) to tackle the problem of imbalanced augmented class learning with unlabeled data. LCP enlarges the labeled training data set by estimating class labels for unlabeled data, to meet the challenge of lacking supervision information of augmented classes via identifying some of their instances, and to alleviate the damage of class-imbalance via identifying more instances for each seen class. LCP firstly initializes label confidence, i.e., the posterior probability distributions of all classes (including augmented classes) for unlabeled data, then iteratively propagates label confidence to identify a valid label for each unlabeled instance to enlarge the labeled training data set. Finally, LCP predicts for unseen instances by linear neighborhood reconstruction to be robust to potential noise. Results on abundant experiments show that LCP is significantly superior to many state-of-the-art methods, and robust to high imbalance ratio and high open level. LCP can sufficiently unleash its strength especially when there are abundant unlabeled data available.**

***Keywords*-augmented class learning; LACU framework; class-imbalance; unlabeled data**



(a)

(b)

Fig. 1: (a) Illustration of the impact of class-imbalance on LAC problem. (b) Illustration of the motivation of LCP approach. Positive and negative symbol represents minority and majority seen classes, respectively. Question symbol represents augmented classes. A circled symbol represents an unlabeled instance is estimated as the symbol. A solid curve represents a ground-truth decision boundary and a dashed curve represents a possible decision boundary without sufficient data for training.

I. Introduction

Traditional supervised learning methods often assume that test data and training data share the same and fixed set of class labels. While in many real-world applications, the environments are open and dynamic, which may break such assumption. Incremental Learning (IL) is one of the learning paradigms for open environments. According to which type of things increases over time, IL is categorized into three main branches [1], attribute-incremental learning (A-IL) [2], example-incremental learning (E-IL) [3]–[5] and class-incremental learning (C-IL) [6]–[8]. Therein, C-IL has attracted increasing attention from various fields since it is widespread in more and more real-world applications. For example, a web text categorization system needs to predict the web texts from broad classes continuously emerging in the future.

As augmented classes will emerge over time, C-IL needs to predict instances of augmented classes with the restriction that only examples from seen classes are observed in training phase. Lacking of supervision information of augmented classes makes it hard to predict their instances correctly, which poses great challenge. This core difficulty of C-IL is formulated as LAC (Learning with Augmented Class) problem, and LACU (Learning with Augmented Class with Unlabeled data) framework advances the study of LAC problem by exploiting unlabeled data containing instances from all classes [9].

In many real-world applications of LAC, such as text categorization, image classification etc., class distributions among seen classes are often imbalanced. The class-imbalance problem [10], [11] will further increase learning difficulties, including but may not limited to: (1) minority seen class instances could be prone to be misclassified to majority seen classes or augmented classes; (2) augmented class instances could be more prone to be misclassified to majority seen classes; (3) inaccurate decision boundaries among seen classes caused by class-imbalance could probably hinder the learning of decision boundaries between augmented classes and seen classes. Fig. 1(a) illustrates the impact of class-imbalance on LAC problem. Class-imbalance problem will degenerate a classifier's performance obviously without being properly addressed. A reliable LAC method is expected to be adaptable to augmented classes as well as to be robust to class-imbalance. Though some research work studied the class-imbalance problem in IL with streaming data [12], there is no work concerning class-imbalance in LAC problem yet, as far as we know.

In this paper, we propose a novel approach Label Confidence Propagation (LCP) to tackle the problem of lacking of supervision information of augmented classes as well as the problem of class-imbalance simultaneously in LACU framework. The approach is inspired by the following idea: labeled training data set can be enlarged by estimating class labels for unlabeled data. Thus on one hand, the supervision information of augmented classes can be estimated, which will greatly reduce the difficulty of learning the concepts of augmented classes. And on the other hand, as a popular strategy in class-imbalance learning [11], when more training data is available to represent class concepts more sufficiently, the classification system is less sensitive to the level of imbalance among classes. When unlabeled instances are used to enlarge the labeled training data set according to their accurately predicted class labels, the learning difficulties caused by class-imbalance in LAC problem (please refer to the above paragraph) can all be addressed, as illustrated in Fig. 1(b).

LCP firstly initializes label confidence, i.e. the posterior probability distributions of all classes (including augmented class) for unlabeled data, then iteratively propagates label confidence to identify a valid label for each unlabeled instance to enlarge the labeled training data set. Finally, LCP classifies an unseen instance by linear neighborhood reconstruction to be robust to the noise potentially introduced. Results on abundant experiments show that LCP is significantly superior to many state-of-the-art methods, and robust to high imbalance ratio and high open level. LCP can sufficiently unleash its strength especially when there are abundant unlabeled data available.

The rest of this paper is organized as follows. Section 2 briefly describes the related work, Section 3 presents the LCP approach in detail, Section 4 reports the experimental results and Section 5 concludes.

II. Related Work

Incremental learning requires methods to be adaptable to the changes of open and dynamic environments. As an important branch, *class-incremental learning* (C-IL) focuses on the emerging classes, and has attracted more and more attention. According to whether and how the information of augmented classes is available in training phase, there are several settings in C-IL:

(1) A few of augmented class examples are observed in training phase. Some methods are designed for this setting. For example, in [7], base learners are trained from data containing augmented classes examples and are incrementally integrated as an ensemble. In [8], binary classifiers for each new class are incrementally added, which share the hypothesis of seen classes.

(2) No augmented class example is available in training data set. It is formulated as *learning with augmented class* (LAC) problem [9], and is especially hard to solve. Though some related learning paradigms could be of some help, such as the *classification with a reject option*, the *open set recognition*, the *outlier detection* and the *rare class discovery*, they are not consistent with the aim and scope of LAC problem and follow different principles[1].

(3) The unlabeled data from all classes (including augmented classes) are available in training phase. In [9], the *learning with augmented class with unlabeled data* (LACU) framework exploits unlabeled data to help LAC problem, and LACU-SVM approach is designed to utilize the large margin separators surrounding seen classes to help distinguish augmented classes.

In addition, some recent advancements in C-IL include the *classification with streaming emerging new class* [13]–[15] in data stream context, and the *learning with emerging new labels* in multi-label classification [16].

Though some research work studied the class-imbalance problem in IL with streaming data [12], there is no work concerning class-imbalance in LAC problem yet, as far as we know. An intuitive solution towards imbalanced augmented class learning with unlabeled data is to balance the skewness among seen classes in training data set, thus the problem can be converted to normal LAC problem to solve. Various popular class-imbalance learning techniques can be utilized for this purpose, such as random or synthetic undersampling or oversampling [10]. We compare LCP approach with some of these straightforward strategies in Section 4.

III. The LCP Approach

Let $D_s = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{L}$ denote the imbalanced labeled training data set, where $\boldsymbol{x}_i \in R^d$ is a training instance and its label is $y_i \in Y = \{1, 2, ..., K\}$. And the testing instances in $D_t = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{T}$ are sampled from an open data set $D_o = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{\infty}$, where $y_i \in Y_o = \{1, 2, ..., K, K+1, ..., M\}$ with $M > K$. As augmented classes are unobservable during training phase, one can access an unlabeled data set $D_u = \{\boldsymbol{x}_i\}_{i=L+1}^{L+U}$ sampled from D_o during training phase to help the problem of augmented class learning. In LACU framework [9], the task is to learn a multi-class classifier $f(\boldsymbol{x}) : X \to \mathcal{Y}$, where $\mathcal{Y} = \{1, 2, ..., K, novel\}$ and $novel$ indicates that $\boldsymbol{x}$ belongs to the augmented classes (we treat the $novel$ class as the $(K+1)$-th class), in order to minimize the expected risk

$$f^* = \arg\min_{f \in \mathcal{H}} \mathbb{E}_{(\boldsymbol{x}, y) \sim D_o} err(y, f(\boldsymbol{x})), \tag{1}$$

where $\mathcal{H}$ is a hypothesis space and err is LAC error

$$err(y, f(\boldsymbol{x})) = \begin{cases} I(f(\boldsymbol{x}) \neq y), & y \in Y \\ I(f(\boldsymbol{x}) \neq novel), & y \notin Y \end{cases}. \tag{2}$$

[1] Please refer to [9] for more description.

Here $I(x)$ is 1 when x is true and 0 otherwise.

In LCP, the labeled training data set D_s is enlarged by estimating class labels for the unlabeled data D_u, such that there are some estimated examples for $novel$ class to meet the challenge of lacking supervision information of augmented classes, and there are more examples for seen classes to alleviate the damage of class-imbalance. LCP has three stages: (1) *label confidence initialization* stage initializes label confidence, i.e., the posterior probability distributions of all classes in $\mathcal{Y}$; (2) *label confidence propagation* stage iteratively propagates label confidence to identify a valid label for each unlabeled instance in D_u to enlarge the labeled training data set D_s; (3) *prediction* stage classifies an unseen instance by linear neighborhood reconstruction in the enlarged training data set.

*1) **Label Confidence Initialization:*** Let $\boldsymbol{P}_{(L+U)\times(K+1)}$ be the posterior probability distribution matrix, or the initial label confidence matrix, for the instances in the whole training data set $D = D_s \bigcup D_u$. Also, denote $\boldsymbol{p}_i$ as the i-th row of $\boldsymbol{P}$ indicating the label confidence vector for instance $\boldsymbol{x}_i$, and $p_{ij} \in [0, 1]$ as the j-th element of $\boldsymbol{p}_i$ indicating the probability of $\boldsymbol{x}_i$ belonging to class j.

For a labeled instance $\boldsymbol{x}_i \in D_s$, the probability p_{ij} is set to 1 for its ground-truth class and 0 for other classes, i.e.,

$$p_{ij} = \begin{cases} 1, & \text{if } j = y_i \\ 0, & \text{otherwise} \end{cases} \quad (1 \le i \le L, 1 \le j \le K+1). \tag{3}$$

To meet the challenge of lacking supervision information of augmented classes, the probability of each unlabeled instance $\boldsymbol{x}_i \in D_u$ belonging to $novel$ class is initialized by a default value of 0.5[2]. According to cluster assumption of semi-supervised learning [17], similar instances have high probability of sharing the same class, so the probability of an unlabeled instance belonging to each seen class can be estimated via its similarity to each seen class. However, instance similarity estimation by direct distance measure can be problematic with high dimensionality [18]. Therefore, we apply LDA [19] on D to get data set D' with reduced dimensionality, and initializes the probability of $\boldsymbol{x}_i$ belonging to each seen class based on D'. In detail, the boundary radius γ_j of the seen class $j \in Y$ is defined as follows:

$$\gamma_j = max_{\boldsymbol{x}_i \in C_j} \parallel \boldsymbol{x}_i - \boldsymbol{\mu}_j \parallel_2 \quad (1 \le i \le L, 1 \le j \le K), \tag{4}$$

where C_j denotes the instance set of class j in data set D_s and $\boldsymbol{\mu}_j = \frac{1}{|C_j|}\sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{x}_i$. Generally speaking, an instance should be closer to the instances of its own class and far away from the instances of other classes. Therefore, $\boldsymbol{x}_i$ is considered to be uncorrelated to class j if $\parallel \boldsymbol{x}_i - \boldsymbol{\mu}_j \parallel_2 > \gamma_j$. According to cluster assumption, the probability of unlabeled instance $\boldsymbol{x}_i \in D_u$ belonging to class j depends on the similarity measure h_{ij}:

$$h_{ij} = \parallel \boldsymbol{x}_i - \boldsymbol{\mu}_j \parallel_2 - \gamma_j \quad (L+1 \le i \le L+U, 1 \le j \le K). \tag{5}$$

The label confidence vector $\boldsymbol{p}_i$ for an unlabeled instance $\boldsymbol{x}_i \in D_u$ can then be initialized as follows:

$$p_{ij} = \begin{cases} 0.5, & j = K+1 \\ 0, & 1 \le j \le K,\ h_{ij} \ge 0 \\ \frac{0.5\times|h_{ij}|}{\sum_{j=1}^{K}|h_{ij}|}, & 1 \le j \le K,\ h_{ij} < 0 \end{cases} \quad (L+1 \le i \le L+U), \tag{6}$$

Then, normalization is conducted to ensure $\boldsymbol{p}_i$ is an probability: $p_{ij} = p_{ij}/\sum_{j=1}^{K+1} p_{ij}$. If an unlabeled instance is uncorrelated to any seen class, the normalization step rescales its probability of belonging to $novel$ class to 1. It is worth mentioning that, D' with reduced dimensionality is only used in the first stage of LCP.

Algorithm 1 The LCP algorithm

```
Require:
    $D_s$ :  Training data set $\{\boldsymbol{x}_i, y_i\}_{i=1}^{L}, y_i \in \{1, 2, \ldots, K\}$
    $D_u$ :  Unlabeled data set $\{\boldsymbol{x}_i\}_{i=L+1}^{L+U}$
    $k$ :    The number of nearest neighbors
    $\alpha$ :    The balancing parameter in (0,1)
    $Q$ :    The number of iterations
Ensure:
    $f(\boldsymbol{x})$ :   The predicted label for $\boldsymbol{x}$ in $D_t = \{\boldsymbol{x}_i\}_{i=1}^{T}$
Stage 1. Label Confidence Initialization
Set $D = D_s \bigcup D_u$ and get $D'$ with reduced dimensionality by applying LDA on $D$
Initialize label confidence matrix $\mathbf{P} = [p_{ij}]_{(L+U)\times(K+1)}$ with $p_{ij} = 0$
Set the label confidence vector $\boldsymbol{p}_i$ for each instance $\boldsymbol{x}_i \in D_s$ according to Eq.(3)
Calculate the boundary radius $\gamma_j$ for each class $j$ ($j \in \{1, 2, \ldots, K\}$) according to Eq.(4)
for each $\boldsymbol{x}_i \in D_u$ $(L+1 \le i \le L+U)$ do
    Calculate $h_{ij}(1 \le j \le K)$ according to Eq.(5)
    Set the label confidence vector $\boldsymbol{p}_i$ for instance $\boldsymbol{x}_i$ according to Eq.(6) and perform normalization
end for
Stage 2. Label Confidence Propagation
Form the affinity matrix $\boldsymbol{W}$ according to Eq.(7)
Set $\boldsymbol{S}$ according to Eq.(8) and set $\boldsymbol{F}_0 = \boldsymbol{P}$
for $q = 1 \ldots Q$ do
    Set $\boldsymbol{F}_q$ according to Eq.(9)
    Rescale $\boldsymbol{F}_q$ according to Eq.(10)
end for
Set the final label confidence matrix $\hat{\boldsymbol{F}} = \boldsymbol{F}_Q$
Obtain the valid label $\hat{y}_i$ of each $\boldsymbol{x}_i \in D(1 \le i \le L+U)$ according to Eq.(12)
Update $D = \{(\boldsymbol{x}_i, \hat{y}_i)\}_{i=1}^{L+U}$
Stage 3. Prediction
for each $\boldsymbol{x}_i \in D_t(1 \le i \le T)$ do
    Identify the $k$-nearest neighbors $N(\boldsymbol{x_i})$ in $D$ for $\boldsymbol{x}_i$
    Return the predicted class label $f(\boldsymbol{x}_i)$ according to Eq.(13)
end for
```

*2) **Label Confidence Propagation:*** After initializing the label confidence matrix $\boldsymbol{P}$, LCP refines the label confidence

[2] Without any prior information, it is reasonable to do so. If the prior probability of $novel$ class p_N is given, it is reasonable to initialize the above probability by p_N.

by iterative propagation, and then identifies the valid label for each unlabeled instance based on the refined label confidence to enlarge the labeled training data set. Inspired by [20], we design an Iterative Label Confidence Propagation (ICLP) process to transfer the supervised information from the labeled examples in D_s to help the learning task on data set D.

In detail, a weighted graph $G = (V, E)$ is constructed on data set D, where the vertex set $V = \{\boldsymbol{x}_i | 1 \leq i \leq L+U\}$, and the weights of edges in E are determined by affinity matrix $\boldsymbol{W}_{(L+U)\times(L+U)}$ which incorporates the distance relationships among instances. Note that label confidence initialization stage may introduce potential noise. To alleviate the potential accumulative damage caused by propagating noise, we restrict $\boldsymbol{W}$ to be the 1-nearest neighbor affinity matrix, i.e.,

$$W_{ab} = \begin{cases} exp(\frac{-\|\boldsymbol{x}_a - \boldsymbol{x}_b\|^2}{2\sigma^2}), & \text{if } \boldsymbol{x}_b \text{ is } \boldsymbol{x}_a\text{'s nearest neighbor} \\ 0, & \text{otherwise} \end{cases}. \quad (7)$$

To guarantee the convergence in subsequent ICLP process, $\boldsymbol{W}$ need to be symmetrically normalized by row:

$$\boldsymbol{S} = \boldsymbol{R}^{-1/2}\boldsymbol{W}\boldsymbol{R}^{-1/2}, \quad (8)$$

where $\boldsymbol{R}$ is a diagonal matrix with diagonal element $R_{aa} = \sum_{b=1}^{L+U} W_{ab}$.

Denote $\boldsymbol{F}_{(L+U)\times(K+1)}$ as a matrix with non-negative entries, whose element F_{ij} corresponds to the probability of $\boldsymbol{x}_i$ in D belonging to class j $(j \in \{1, 2, ..., K+1\})$. Initializing $\boldsymbol{F}$ with the $\boldsymbol{P}$ obtained above, i.e. $\boldsymbol{F}_0 = \boldsymbol{P}$, then $\boldsymbol{F}$ can be updated as follows at the q-th iteration:

$$\boldsymbol{F}_q = \alpha \cdot \boldsymbol{S}\boldsymbol{F}_{q-1} + (1-\alpha) \cdot \boldsymbol{P}, \quad (9)$$

where parameter $\alpha \in (0, 1)$ controls the relative amount of information that instance inherited from its only nearest neighbor and its initial label information. After that, $\boldsymbol{F}_q$ is rescaled based on the initial label confidence matrix $\boldsymbol{P}$, i.e.,

$$F_{ij} = \frac{F_{ij}.\delta_{ij}}{\sum_{k=1}^{K+1} F_{ik} \cdot \delta_{ik}} (1 \leq i \leq L+U, 1 \leq j \leq K+1), \quad (10)$$

where $\delta_{ij} = 1$ if $p_{ij} > 0$ and 0 otherwise. This step not only guarantees normalization, but also ensures the instances in D being immune to noise propagation when they are uncorrelated to particular classes, including that the F_{ij} values for the seen class examples in D_s are totally consistent with their ground-truth class labels. Such procedure of ICLP will continue until convergence and we have $\hat{\boldsymbol{F}} = \boldsymbol{F}_Q$, where Q is the maximum number of iterations. After that, the valid label for each instance in D can be identified based on the final label confidence matrix $\hat{\boldsymbol{F}}$ as follows:

$$\hat{y_i} = \arg \max_{j\in\{1,2,...,K+1\}} \hat{F}_{ij} \quad (1 \leq i \leq L+U). \quad (11)$$

However, the unlabeled data from *novel* class are prone to be misclassified since there is no ground-truth *novel* class examples in the training data set. To meet this challenge, LCP initializes the probability of an unlabeled instance belonging to *novel* class by 0.5 or 1. Hence, in order to inherit this advantage, we use the class mass normalization mechanism [21] to take the prior information of class distribution into account, which is reflected by $\boldsymbol{P}$:

$$\hat{y_i} = \arg \max_{j\in\{1,2,...,K+1\}} \frac{v_j}{\hat{v}_j} \cdot \hat{F}_{ij} \quad (1 \leq i \leq L+U), \quad (12)$$

where $v_j = \sum_{i=1}^{L+U} p_{ij}$ and $\hat{v}_j = \sum_{i=1}^{L+U} \hat{F}_{ij}$. Note that, the class labels of the seen class examples in D_s remain unchanged via this step.

3) ***Prediction:*** During testing phase, the class label of $\boldsymbol{x}$ in D_t is predicted based on the enhanced examples $(\boldsymbol{x}_i, \hat{y}_i)$ in D. Let $N(\boldsymbol{x})$ be the set of $\boldsymbol{x}$'s k-nearest neighbors in D. However, not all pre-estimated labels of instances in D are correct. In order be more robust to noise, LCP makes prediction by reconstructing the unseen instance $\boldsymbol{x}$ with the instances belonging to the same class in $N(\boldsymbol{x})$. Specially, the instance $\boldsymbol{x}$ will be predicted to the class with the minimum linear reconstruction error, i.e.,

$$f(\boldsymbol{x}) = \arg \min_{j\in\{1,2,...,K+1\}} \|\boldsymbol{x} - \sum_{\boldsymbol{x}_n \in E_j} \beta_n^j \cdot \boldsymbol{x}_n\|_2, \quad (13)$$

where, E_j is the instance set of class j in $N(\boldsymbol{x})$, and β_n^j is the corresponding reconstruction coefficient. The $\boldsymbol{\beta}^j = [\beta_1^j, ..., \beta_{|E_j|}^j]$ is determined by solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\beta}^j} \|\boldsymbol{x} - &\sum_{\boldsymbol{x}_n \in E_j} \beta_n^j \cdot \boldsymbol{x}_n\|_2 \\ s.t.\ \ &\beta_n^j \geq 0 \end{aligned}. \quad (14)$$

The pseudo code of the proposed LCP approach is summarized in $Algorithm1$.

IV. Experiments

A. Experimental Settings

Data sets 4 controlled imbalanced data sets and 8 real-world imbalanced data sets are collected to assess the performance of comparison methods. The controlled data sets are MNIST[3], Letter[4], CIFAR-10 and CIFAR-100[5]. They are pre-processed to have different levels of class-imbalance. The real-world imbalanced data sets are Font, Vicon Physical Action, UJI-IndoorLoc, Sat and Turkiye[6], Caltech101[7], Caltech256[8] and Forest Cover[9]. All of them are pre-processed to have different configurations of open environments in order to thoroughly investigate the comparison methods for imbalanced augmented class learning with unlabeled data. TABLE I and TABLE II shows the data set information.

For the controlled data sets, we take MNIST as an example to explain how to obtain the imbalanced data sets with different configurations for investigation. We consider 3 different imbalance levels, i.e. the maximum imbalance ratio $ImR = 3, 9, 15$.

[3] $http : //yann.lecun.com/exdb/mnist/$

[4] $http : //archive.ics.uci.edu/ml/datasets.html$

[5] both are at $https : //www.cs.toronto.edu/\ kriz/cifar.html$

[6] they are all at $http : //archive.ics.uci.edu/ml/datasets.html$

[7] $http : //www.vision.caltech.edu/Image_Datasets/Caltech101$

[8] $http : //www.vision.caltech.edu/Image_Datasets/Caltech256$

[9] $http : //kdd.ics.uci.edu/databases/covertype/covertype.html$

TABLE I: Controlled data set information. M: the number of classes in original data sets, dim: dimensionality, ImR: imbalance ratio (the level of class-imbalance on controlled data sets is characterized by $|majC|/|minC|$), $minC$: a minority seen class with $|minC|$ being its size, #$minC$: the number of minority seen classes, $majC$: a majority seen class with $|majC|$ being its size, #$majC$: the number of majority seen classes, $AugC$: an augmented class with $|AugC|$ being its size, #$AugC$: the number of augmented classes, $size$: data set size.

controlled data sets[9]	M	dim	ImR	(#$minC$, #$majC$, #$AugC$)	$\|minC\|$	$\|majC\|$	$\|AugC\|$	$size$
MNIST	10	784	3	(1,4,5)	50	150	150	3450
			9	(1,4,5),(3,2,5),(5,2,3)		450	450	10050,7650,6150
			15	(1,4,5)		750	750	16650
Letter	26	16	5	(1,4,5)	40	200	200	4520
				(2,3,5)				4040
				(3,2,5)				3560
CIFAR-10	10	3072	5	(1,4,5)	100	500	500	11300
			10			1000	1000	22300
CIFAR-100	100	3072	3	(1,4,5)	50	150	150	3450
			5			250	250	5650

TABLE II: Real-world data set information. M: the number of classes in original data sets, dim: dimensionality, ImR: imbalance ratio (the level of class-imbalance on real-world data sets is characterized by $minimum$, $maximum$ and $average$ imbalance ratio [22]), #$seenC$: the number of all seen classes, #$AugC$: the number of augmented classes, $|C|$: class size in real-world imbalanced data sets, which is characterized by $minimum$, $maximum$ and $average$ class sizes.

real-world data sets[10]	M	dim	(#$seenC$, #$AugC$)	ImR			$\|C\|$		
				min	max	avg	min	max	avg
Font	153	409	(5,5)	1.0	113.7	6.4	824	93688	5443
Vicon Physical Action	20	27	(5, 15)	1.0	1.4	1.1	26334	35251	30270
UJIIndoorLoc	13	527	(5, 8)	1.0	2.8	1.4	995	2749	1620
Sat	6	36	(3, 3)	1.0	2.5	1.7	626	1553	1073
Forest Cover	7	54	(3, 4)	1.2	103.2	16.4	2747	283301	83002
Turkiye	13	32	(5, 8)	1.0	22.1	3.8	41	904	448
Caltech101	101	1984	(5, 5)	1.0	23.5	2.4	31	800	86
Caltech256	256	1764	(5, 5)	1.0	10.4	1.5	80	827	120

For $ImR = 3$, 5 from 10 classes are randomly selected as seen classes, with the remaining 5 classes being treated as augmented classes. Then, 1 from 5 seen classes is randomly selected as a minority seen class, with the remaining 4 being majority seen classes. To obtain the desired imbalance level $ImR = 3$, we perform undersampling so that each minority and majority seen class has 50 and 150 examples, respectively. For the sake of simplicity, the size of an augmented class is kept the same as that of a majority seen class. Thus, we have an imbalanced data set where the size of the labeled training data set, the unlabeled data set and the test data set is $1 \times 50 + 4 \times 150 = 650$, $1 \times 50 + 9 \times 150 = 1400$, $1 \times 50 + 9 \times 150 = 1400$, respectively. Under the same set of constrains described above, 10 imbalanced data sets are obtained from 10 configurations with different seen classes and augmented classes randomly selected. For $ImR = 9$ and 15, we have 3 and 1 configurations of (#$minC$, #$majC$, #$AugC$), respectively, with each configuration further deriving 10 imbalanced data sets. Therefore, we have 50 imbalanced data sets derived from MNIST.

Since the 8 real-world imbalanced data sets have naturally imbalanced class distributions, they only need to be pre-processed to have different configurations of open environments. Taking Font as an example, we randomly select 5 classes from all 153 classes as seen classes and 5 classes from the remaining 148 classes as augmented classes. In order to maintain i.i.d. distributions, the instances of each seen class are evenly allocated to training data set, unlabeled data set and test data set, and the instances of each augmented class are evenly allocated to unlabeled data set and test data set. Similar to the controlled data sets, each real-world imbalanced data set randomly generates 10 different configurations with different seen classes and augmented classes for a fixed configuration of (#$seenC$, #$AugC$). That means there are 10 imbalanced data sets derived for each real-world imbalanced data set.

There are 18 configurations altogether deriving 180 imbalanced data sets in total. For each imbalanced data set, 10 times repeated experiments are carried out and the average results are recorded. Pairwise t-tests at 0.05 significance level are also conducted. Macro-F1, a common performance measure for class-imbalanced learning tasks, is used in the experiments.

Comparison methods To compare with LCP, we investigate the classic method LACU-SVM for LACU framework, and 5 other methods from related learning paradigms (such as outlier detection), LOF, MOC-SVM, OVR-SVM, iForest and 1-vs-Set, since they can also be helpful to LAC problem and served as comparison methods for LACU-SVM [9]. As introduced in Section 2, a general straightforward strategy for imbalanced LACU problem is to convert it to normal LAC problem to solve by utilizing an appropriate class-imbalance learning technique to balance seen classes. Here, we adopt random undersampling and a very popular synthetic oversampling technique SMOTE [24] and combine each of them with each

[9] We use the normalized features of MNIST data set. And for the rest three data sets, we use the given features of the data sets downloaded from their websites without further processing.

[10] We extract the features of Caltech256 data set by HOG [23]. And for the rest six data sets, we use the given features of the data sets downloaded from their websites without further processing.

TABLE III: Performance (mean ± std. deviation) of each comparison method on MNIST data set. In addition, $\bullet/\circ$ indicates whether LCP is significantly superior/inferior to the comparison methods (pairwise t-test at 0.05 significance level).

Macro-F1	$ImR = 3$	$ImR = 9$	$ImR = 9$	$ImR = 9$	$ImR = 15$
	($\#minC = 1$)	($\#minC = 1$)	($\#minC = 3$)	($\#minC = 5$)	($\#minC = 1$)
LCP	0.908±0.017	0.915±0.017	0.872±0.022	0.870±0.019	0.913±0.022
LCP-SMOTE	0.907±0.013	0.922±0.001	0.890±0.014	0.885±0.015	0.927±0.013
LACU-SVM	0.631±0.064●	0.620±0.038●	0.424±0.066●	0.298±0.023●	0.641±0.067●
LACU-SVM+SMOTE	0.689±0.051●	0.705±0.048●	0.564±0.033●	0.637±0.072●	0.688±0.042●
LACU-SVM+USAM	0.488±0.114●	0.486±0.112●	0.460±0.087●	0.281±0.053●	0.541±0.122●
MOC-SVM	0.581±0.052●	0.584±0.054●	0.542±0.066●	0.511±0.056●	0.567±0.054●
MOC-SVM+SMOTE	0.539±0.050●	0.543±0.050●	0.417±0.043●	0.329±0.033●	0.540±0.039●
MOC-SVM+USAM	0.567±0.044●	0.555±0.045●	0.528±0.056●	0.499±0.052●	0.546±0.046●
OVR-SVM	0.663±0.040●	0.635±0.029●	0.502±0.034●	0.625±0.052●	0.619±0.036●
OVR-SVM+SMOTE	0.662±0.039●	0.635±0.029●	0.495±0.033●	0.620±0.052●	0.620±0.032●
OVR-SVM+USAM	0.653±0.042●	0.612±0.036●	0.479±0.022●	0.586±0.054●	0.615±0.024●
LOF	0.669±0.046●	0.675±0.045●	0.517±0.048●	0.499±0.074●	0.680±0.059●
LOF-SMOTE	0.584±0.036●	0.616±0.034●	0.396±0.039●	0.305±0.022●	0.646±0.028●
LOF-USAM	0.614±0.046●	0.600±0.056●	0.503±0.054●	0.493±0.063●	0.584±0.060●
iForest	0.507±0.044●	0.490±0.044●	0.382±0.051●	0.389±0.062●	0.482±0.050●
iForest-SMOTE	0.506±0.001●	0.482±0.029●	0.378±0.057●	0.361±0.052●	0.473±0.030●
iForest-USAM	0.497±0.042●	0.478±0.036●	0.374±0.022●	0.351±0.054●	0.477±0.032●
1-vs-Set	0.487±0.046●	0.475±0.025●	0.414±0.046●	0.399±0.044●	0.480±0.051●
1-vs-Set+SMOTE	0.504±0.032●	0.516±0.014●	0.426±0.014●	0.405±0.062●	0.546±0.048●
1-vs-Set+USAM	0.463±0.026●	0.450±0.036●	0.363±0.057●	0.349±0.023●	0.414±0.060●

TABLE IV: Performance (mean ± std. deviation) of each comparison method on Letter data set. In addition, $\bullet/\circ$ indicates whether LCP is significantly superior/inferior to the comparison methods (pairwise t-test at 0.05 significance level).

Macro-F1	$ImR = 5$	$ImR = 5$	$ImR = 5$
	($\#minC = 1$)	($\#minC = 2$)	($\#minC = 3$)
LCP	0.911±0.019	0.879±0.012	0.866±0.023
LCP-SMOTE	0.913±0.018	0.890±0.009	0.878±0.011
LACU-SVM	0.850±0.041●	0.801±0.038●	0.768±0.034●
LACU-SVM+SMOTE	0.876±0.029	0.831±0.027	0.800±0.042●
LACU-SVM+USAM	0.778±0.041●	0.755±0.033●	0.749±0.042●
MOC-SVM	0.554±0.023●	0.498±0.015●	0.456±0.023●
MOC-SVM+SMOTE	0.546±0.019●	0.492±0.010●	0.443±0.019●
MOC-SVM+USAM	0.322±0.026●	0.324±0.019●	0.332±0.021●
OVR-SVM	0.644±0.048●	0.583±0.032●	0.535±0.044●
OVR-SVM+SMOTE	0.646±0.042●	0.574±0.030●	0.530±0.040●
OVR-SVM+USAM	0.622±0.051●	0.558±0.039●	0.523±0.051●
LOF	0.712±0.024●	0.615±0.037●	0.568±0.047●
LOF-SMOTE	0.709±0.022●	0.641±0.014●	0.584±0.012●
LOF-USAM	0.577±0.040●	0.531±0.035●	0.516±0.043●
iForest	0.597±0.030●	0.555±0.048●	0.524±0.059●
iForest-SMOTE	0.592±0.002●	0.544±0.010●	0.510±0.053●
iForest-USAM	0.572±0.011●	0.534±0.009●	0.466±0.045●
1-vs-Set	0.852±0.021●	0.808±0.039●	0.767±0.042●
1-vs-Set+SMOTE	0.868±0.025●	0.838±0.014●	0.816±0.012●
1-vs-Set+USAM	0.827±0.031●	0.781±0.025●	0.736±0.063●

of the above 6 comparison methods in turn. In addition, we also combine LCP with SMOTE in order to further investigate the proposed approach. We have 20 methods in comparison in total.

- **LACU-SVM** [9]: LACU-SVM trains an binary SVM classifier $f_j(\cdot)$ for the j-th seen class, $j \in \{1, 2, ..., K\}$. It predicts the seen class if $\max_{j=1,...,K} f_j(x) > 0$, and *novel* class otherwise.
- **MOC-SVM**: As introduced in [9], MOC-SVM trains an one-class SVM for each seen class to detect *novel* class.
- **OVR-SVM** [25]: According to [9], the original one-vs-rest SVM, which trains a single binary SVM classifier $f_j(\cdot)$ for each class, can be adapted to predict *novel* class by returning the class *novel* only if $max_j f_j(x) < 0$, otherwise returning the class $\arg\max_{j=1,...,K} f_j(x)$.
- **LOF** [26]: LOF is proposed for finding anomalous data points by measuring the local deviation between the given data point and its neighbours. We use LOF for detecting *novel* class, and use the one-vs-rest SVM for seen classes. This strategy is also employed for iForest and 1-vs-Set.
- **iForest** [27]: iForest is a state-of-the-art outlier detection algorithm and it makes the most of two outliers quantitative properties, i.e., few and different, by exploring the concept of isolation of samples.
- **1-vs-Set** [28]: 1-vs-Set Machine takes the risk over open space into consideration by introducing extra decision boundaries to minimize the regions for seen classes.
- **LACU-SVM/MOC-SVM/OVR-SVM/LOF/iForest/1-vs-Set + SMOTE/Undersampling** (abbreviated as USAM): In this series of comparison methods, SMOTE or undersampling is used to fully balance the data distribution of the training data set D_s. Then LACU-SVM/ MOC-SVM/ OVR-SVM/ LOF/ iForest/ 1-vs-Set is applied on the balanced training data set respectively.
- **LCP**
- **LCP-SMOTE**: SMOTE is used to fully balance the data distribution of D_s. Then LCP is applied.

For MOC-SVM, OVR-SVM and their variants, we use the implementations in LIBSVM software [29]. For LACU-SVM, iForest, 1-vs-Set and their variants, we use the code released by the corresponding authors. The coefficient C in SVM is selected via cross validation on training data, and the width for Gaussian kernel ν is set to $1/dim$. In LACU-SVM, other parameters are set according to the paper: $ramp_s = -0.3$, $\eta = 1.3$, $\lambda = 0.1$, $max\_iter = 10$, C_1 is set to C, C_2 is set to $C_1 L/U$. For LOF and its variants, the minimum and maximum number of neighbors are 3 and 5, respectively. And the Euclid distance d_e is replaced with $1 - exp(-\nu d_e)$ for Gaussian kernel since the original LOF does not have a kernel version. For iForest, 1-vs-Set and their variants, we use the default parameters introduced in the corresponding papers. The

TABLE V: Performance (mean ± std. deviation) of each comparison method on CIFAR-10 data set. In addition, $\bullet/\circ$ indicates whether LCP is significantly superior/inferior to the comparison methods (pairwise t-test at 0.05 significance level).

Macro-F1	$ImR = 5$	$ImR = 10$
LCP	0.271±0.027	0.272±0.016
LCP-SMOTE	0.271±0.027	0.341±0.016○
LACU-SVM	0.117±0.000●	0.118±0.000●
LACU-SVM+SMOTE	0.117±0.000●	0.118±0.000●
LACU-SVM+USAM	0.117±0.000●	0.118±0.000●
MOC-SVM	0.117±0.000●	0.118±0.000●
MOC-SVM+SMOTE	0.117±0.000●	0.118±0.000●
MOC-SVM+USAM	0.117±0.000●	0.118±0.000●
OVR-SVM	0.117±0.000●	0.118±0.000●
OVR-SVM+SMOTE	0.117±0.000●	0.118±0.000●
OVR-SVM+USAM	0.117±0.000●	0.118±0.000●
LOF	0.157±0.026●	0.146±0.014●
LOF-SMOTE	0.152±0.018●	0.147±0.007●
LOF-USAM	0.181±0.029●	0.170±0.017●
iForest	0.117±0.000●	0.118±0.000●
iForest-SMOTE	0.117±0.000●	0.118±0.000●
iForest-USAM	0.117±0.000●	0.118±0.000●
1-vs-Set	0.117±0.000●	0.118±0.000●
1-vs-Set+SMOTE	0.117±0.000●	0.118±0.000●
1-vs-Set+USAM	0.117±0.000●	0.118±0.000●

TABLE VI: Performance (mean ± std. deviation) of each comparison method on CIFAR-100 data set. In addition, $\bullet/\circ$ indicates whether LCP is significantly superior/inferior to the comparison methods (pairwise t-test at 0.05 significance level).

Macro-F1	$ImR = 3$	$ImR = 5$
LCP	0.358±0.034	0.360±0.026
LCP-SMOTE	0.368±0.043	0.371±0.013
LACU-SVM	0.116±0.000●	0.117±0.000●
LACU-SVM+SMOTE	0.116±0.000●	0.117±0.000●
LACU-SVM+USAM	0.116±0.000●	0.117±0.000●
MOC-SVM	0.116±0.000●	0.117±0.000●
MOC-SVM+SMOTE	0.116±0.000●	0.117±0.000●
MOC-SVM+USAM	0.116±0.000●	0.117±0.000●
OVR-SVM	0.116±0.000●	0.117±0.000●
OVR-SVM+SMOTE	0.116±0.000●	0.117±0.000●
OVR-SVM+USAM	0.116±0.000●	0.117±0.000●
LOF	0.226±0.037●	0.204±0.021●
LOF-SMOTE	0.215±0.027●	0.197±0.034●
LOF-USAM	0.244±0.022●	0.230±0.037●
iForest	0.116±0.000●	0.117±0.000●
iForest-SMOTE	0.116±0.000●	0.117±0.000●
iForest-USAM	0.116±0.000●	0.117±0.000●
1-vs-Set	0.116±0.000●	0.117±0.000●
1-vs-Set+SMOTE	0.116±0.000●	0.117±0.000●
1-vs-Set+USAM	0.116±0.000●	0.117±0.000●

number of neighbors k in SMOTE is set to 5. For LCP, $k = 20$, $\alpha = 0.9$ and $Q = 100$ for all the experiments. We implement LCP and LCP-SMOTE models via Matlab, and the data and source code can be downloaded at the following address[10].

B. Results and Analysis

We first discuss the results of LCP and the 18 comparison methods excluding LCP-SMOTE on controlled data sets and real-world data sets respectively. The impact analysis of open level (the number of augmented classes), further analysis of LCP via comparison with LCP-SMOTE, and parameter analysis deserve separate threads of discussion later in this part.

1) Controlled Data Sets: **MNIST and Letter MNIST and Letter** The learning tasks on the MNIST handwritten digit data set and the Letter data set are much easier than those on CIFAR-10 and CIFAR-100. The mean and the standard variance of performance are reported in TABLE III and TABLE IV respectively. The results show that LCP is significantly superior to all of the 18 comparison methods, followed by the second best method LACU-SVM+SMOTE. It is also obvious that LCP is insensitive to the change of imbalance ratio, and is only slightly affected by the change of the number of minority seen classes. While other methods generally have obvious performance degeneration in the latter case. This shows the strong robustness of LCP. SMOTE boosts performance obviously for LACU-SVM and 1-vs-Set, but the opposite phenomenon is generally observed in MOC-SVM, OVR-SVM, LOF and iForest, which may because SMOTE introduces potential noise. While undersampling is almost surely damaging to all the methods since it reduces large amount of useful information.

CIFAR-10 and CIFAR-100 Both CIFAR-10 and CIFAR-100 are complex data sets usually used in the experiments for deep learning. The results are reported in TABLE V and TABLE VI respectively. It shows that, LCP still significantly outperforms all the 18 comparison methods on these very difficult learning tasks, followed by the second best method LOF-USAM. LAUC-SVM-based methods no longer have the advantage observed on MINST and Letter, which is took over by LOF-based methods. LACU-SVM, MOC-SVM, OVR-SVM, iForest and 1-vs-Set all fail on these two data set, because they always predict *novel* class. Besides, SMOTE and undersampling can neither boost nor damage their performance.

2) Real-World Data Sets: **Font, Vicon Physical Action, UJIIndoorLoc, Sat and Forest Cover** As shown in TABLE I and TABLE II, each class in these 5 real-world imbalanced data sets has abundant instances. The results are reported in TABLE VII. It shows that, LCP is significantly superior to all the comparison methods and is very robust to high imbalance levels. The absolute advantages of LCP over other methods lie in its ability of fully utilizing the unlabeled data. These 5 data sets have abundant instances in each class, therefore LCP can greatly enlarge the labeled training set, leading to sufficient representation of class concepts, which will meet the challenge of lacking supervision information of augmented classes as well as alleviating the impact of class-imbalance simultaneously. It is important to note that, LCP can sufficiently unleash its strength especially when there are abundant unlabeled data. The runner up is LOF. The performance of LACU-SVM-based methods is not as well as that on the previous controlled data sets. It performs especially poor on data set Font and Forest Cover, both of which have high imbalance levels. One possible reason is that the real-world data sets are naturally imbalanced and are more complex than the controlled data sets. Besides, LACU-SVM is not designed for imbalanced LACU problem, and balancing class distributions by applying SMOTE only can marginally help since each seen class already has sufficient instances.

[10] $http://cse.seu.edu.cn/PersonalPage/xyliu/codes/LCP.htm$

TABLE VII: Performance (mean ± std. deviation) of each comparison method on Font, Vicon Physical Action, UJIIndoorLoc, Sat and Forest Cover data sets. In addition, ●/○ indicates whether LCP is significantly superior/inferior to the comparison methods (pairwise t-test at 0.05 significance level).

Macro-F1	$Font$	$ViconPhysicalAction$	$UJIIndoorLoc$	Sat	$ForestCover$
LCP	0.570±0.043	0.936±0.005	0.894±0.025	0.752±0.041	0.608±0.029
LCP-SMOTE	0.571±0.044	0.937±0.005	0.897±0.009	0.749±0.042	0.583±0.035
LACU-SVM	0.233±0.082●	0.297±0.076●	0.631±0.064●	0.540±0.064●	0.194±0.047●
LACU-SVM+SMOTE	0.234±0.082●	0.298±0.078●	0.676±0.068●	0.577±0.020●	0.194±0.047●
LACU-SVM+USAM	0.214±0.061●	0.286±0.075●	0.345±0.065●	0.189±0.021●	0.194±0.047●
MOC-SVM	0.191±0.051●	0.219±0.045●	0.217±0.045●	0.189±0.021●	0.194±0.047●
MOC-SVM+SMOTE	0.190±0.052●	0.219±0.046●	0.232±0.048●	0.189±0.021●	0.194±0.047●
MOC-SVM+USAM	0.177±0.038●	0.214±0.044●	0.208±0.042●	0.189±0.021●	0.194±0.047●
OVR-SVM	0.239±0.080●	0.400±0.096●	0.410±0.066●	0.126±0.058●	0.123±0.055●
OVR-SVM+SMOTE	0.239±0.081●	0.400±0.096●	0.411±0.066●	0.189±0.021●	0.194±0.047●
OVR-SVM+USAM	0.220±0.061●	0.389±0.095●	0.378±0.064●	0.189±0.021●	0.194±0.047●
LOF	0.291±0.029●	0.572±0.010●	0.572±0.028●	0.525±0.020●	0.471±0.031●
LOF-SMOTE	0.284±0.024●	0.542±0.012●	0.447±0.027●	0.423±0.047●	0.392±0.047●
LOF-USAM	0.283±0.029●	0.570±0.012●	0.592±0.020●	0.522±0.019●	0.424±0.039●
iForest	0.149±0.067●	0.331±0.085●	0.284±0.043●	0.308±0.049●	0.165±0.049●
iForest-SMOTE	0.165±0.077●	0.348±0.075●	0.295±0.041●	0.451±0.057●	0.166±0.041●
iForest-USAM	0.146±0.057●	0.342±0.079●	0.271±0.043●	0.257±0.069●	0.161±0.042●
1-vs-Set	0.234±0.082●	0.319±0.082●	0.207±0.052●	0.526±0.087●	0.123±0.055●
1-vs-Set+SMOTE	0.234±0.082●	0.320±0.082●	0.203±0.049●	0.530±0.041●	0.194±0.047●
1-vs-Set+USAM	0.214±0.061●	0.310±0.080●	0.204±0.050●	0.189±0.021●	0.194±0.047●

TABLE VIII: Performance (mean ± std. deviation) of each comparison method on Turkiye, Caltech101 and Caltech256 data sets. In addition, ●/○ indicates whether LCP is significantly superior/inferior to the comparison methods (pairwise t-test at 0.05 significance level).

Macro-F1	$Turkiye$	$Caltech101$	$Caltech256$
LCP	0.317±0.044	0.506±0.101	0.411±0.065
LCP-SMOTE	0.316±0.044	0.508±0.084	0.408±0.070
LACU-SVM	0.226±0.050●	0.478±0.091	0.312±0.127●
LACU-SVM+SMOTE	0.276±0.068	0.447±0.084●	0.323±0.120●
LACU-SVM+USAM	0.198±0.042●	0.438±0.090●	0.314±0.123●
MOC-SVM	0.235±0.034●	0.312±0.070●	0.181±0.053●
MOC-SVM+SMOTE	0.247±0.033●	0.261±0.039●	0.170±0.048●
MOC-SVM+USAM	0.226±0.022●	0.219±0.075●	0.182±0.058●
OVR-SVM	0.280±0.073	0.250±0.085●	0.154±0.034●
OVR-SVM+SMOTE	0.287±0.072	0.297±0.114●	0.167±0.036●
OVR-SVM+USAM	0.266±0.071●	0.145±0.053●	0.143±0.026●
LOF	0.188±0.017●	0.270±0.087●	0.198±0.057●
LOF-SMOTE	0.194±0.007●	0.233±0.063●	0.171±0.044●
LOF-USAM	0.186±0.027●	0.287±0.072●	0.198±0.055●
iForest	0.244±0.041●	0.333±0.110●	0.297±0.070●
iForest-SMOTE	0.274±0.051	0.412±0.074●	0.304±0.073●
iForest-USAM	0.228±0.050●	0.407±0.077●	0.302±0.073●
1-vs-Set	0.244±0.055●	0.248±0.057●	0.334±0.083●
1-vs-Set+SMOTE	0.289±0.068	0.275±0.071●	0.377±0.091
1-vs-Set+USAM	0.222±0.049●	0.167±0.057●	0.310±0.078●

Turkiye, Caltech101 and Caltech256 Different from the above data sets, in each of these 3 data sets, the minority seen class has rare instances, and other classes generally have non-sufficient instances. The results are shown in TABLE VIII. By no surprise, LCP has the best performance. It is followed by 1-vs-Set+SMOTE and LACU-SVM+SMTOE. While the advantages of LCP over other methods are not as much are those on the 5 previous real-world data sets. LCP's ability is relatively restricted because there are no sufficient instances in each class, so that the labeled training set can only be limitedly enlarged leading to restricted improvements to represent class concepts.

*3) **The impact analysis of open level**:* More augmented classes means higher open level and greater learning difficulty. In order to illustrate the impact of open level on methods' performance, we conduct the experiments on data set CIFAR-100 and Caltech256. Specifically, the number of seen classes is fixed to 5 on both of them, and the number of augmented classes varies from 5 to 30 with step 5 on CIFAR-100 data set ($ImR = 3$), and from 5 to 65 with step 10 on Caltech256 data set. The results are plotted in Fig. 2. As shown in Fig. 2(a), the performance of valid methods LCP-based and LOF-based methods declines with the increasing number of augmented classes. However, it is still noteworthy that LCP is relatively robust across a wild range of the number of augmented classes, i.e., from 5 to 20, until it encounters a abrupt decline when the environment is overly open. And as shown in Fig. 2(b), the performance of valid methods LCP-based and LACU-SVM-based methods generally declines with the increasing number of augmented classes. Different from the previous case, LCP encounters an immediate abrupt decline while subsequently keeps relatively robust performance across a wild range of the number of augmented classes, i.e., from 15 to 65. Generally speaking, there are wide ranges of open level for LCP to have relatively robust performance.

*4) **Further analysis of LCP**:* Generally speaking, LCP-SMOTE is comparable to LCP. Though the former has marginal advantage over the latter on few data sets, the advantage tends to vanish when the number of augmented classes is large, as shown in Fig. 2(a). Besides, LCP-SMOTE is inferior to LCP on few data sets, such as Forest Cover (TABLE VII). The results verify that the effectiveness of LCP mainly relies upon the substantial method designed for enlarging the training data set by sufficiently precise supervision information for unlabeled data, especially the sufficiently precise supervision information for augmented class. And on the other hand, SMOTE might provide further assistance in some cases, while it is not recommended when there are sufficient instances in each class or the learning task is complex.

*5) **Parameter analysis**:* We further study the influence of different parameters in LCP, i.e., the number of neighbors k, balancing parameter α and the number of iterations Q in Fig.

3. We let k vary from 16 to 24, α vary from 0.55 to 0.95 and Q vary from 1 to 100. For clarity of illustration, MNIST and Letter are employed here for sensitivity analysis while similar observations can be made on other data sets. It is obvious that the performance of LCP is stable across a broad range of the parameter k (Fig. 3(a)). While for parameter α. there is an obvious increasing tendency in performance until α approaches 0.9 (Fig. 3(b)). It indicates that we can choose an α value in $[0.9, 1]$ to achieve effective and stable performance. In Fig. 3(c), $||\mathbf{F}_Q - \mathbf{F}_{Q-1}||_2$ is used to quantify the difference between label confidence matrices $\mathbf{F}_Q$ and $\mathbf{F}_{Q-1}$ obtained in the Q- and $(Q-1)$-th iterations. Its changing tendency shows the convergence rate.

V. Conclusion

We propose LCP approach to tackle the problem of imbalanced augmented class learning with unlabeled data. The labeled training data set is enlarged by estimating class labels for unlabeled data, to meet the challenge of lacking the supervision information of augmented classes, and to alleviate the damage of class-imbalance by identifying more examples to each seen class. LCP firstly estimates label confidence, i.e., the posterior probability distributions of all classes (including augmented classes) for unlabeled data, then iteratively propagates label confidence to identify a valid label for each unlabeled instance to enlarge the labeled training data set. Finally, LCP predicts for unseen instances by linear neighborhood reconstruction to be robust to potential noise. The results on abundant experiments verified the significant superiority of LCP over many state-of-the-art comparison methods, showed its robustness to high imbalance ratio and high open level. LCP can sufficiently unleash it strength especially when there are abundant unlabeled data available.

There are several issues worth considering in the future, including conducting additional experiments with more seen classes and augmented classes, exploring other ways to obtain the label confidence vector of unlabeled data, and adapting LCP approach to streaming data, etc.

References

[1] Z. Zhou and Z. Chen, "Hybrid decision tree," *Knowledge-Based Systems*, vol. 15, no. 8, pp. 515–528, 2002. [Online]. Available: https://doi.org/10.1016/S0950-7051(02)00038-2

[2] V. Vapnik, A. Vashist, and N. Pavlovitch, "Learning using hidden information (learning with teacher)," in *Proceedings of the International Joint Conference on Neural Networks*, Atlanta, USA, 2009, pp. 3188–3195. [Online]. Available: https://doi.org/10.1109/IJCNN.2009.5178760

[3] S. Rüping, "Incremental learning with support vector machines," in *Proceedings of the International Conference on Data Mining*, San Jose, USA, 2001, pp. 641–642. [Online]. Available: https://doi.org/10.1109/ICDM.2001.989589

[4] R. Polikar, L. Upda, S. S. Upda, and V. G. Honavar, "Learn++: an incremental learning algorithm for supervised neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 31, no. 4, pp. 497–508, 2001. [Online]. Available: https://doi.org/10.1109/5326.983933

[5] A. Fern and R. Givan, "Online ensemble learning: An empirical study," *Machine Learning*, vol. 53, no. 1-2, pp. 71–109, 2003. [Online]. Available: https://doi.org/10.1023/A:1025619426553

[6] M. Fink, S. Shalev-Shwartz, Y. Singer, and S. Ullman, "Online multiclass learning by interclass hypothesis sharing," in *Proceedings of the International Conference on Machine Learning*, Pittsburgh, USA, 2006, pp. 313–320. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143884

[7] M. D. Muhlbaier, A. Topalis, and R. Polikar, "Learn++.nc: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 152–168, 2009. [Online]. Available: https://doi.org/10.1109/TNN.2008.2008326

[8] I. Kuzborskij, F. Orabona, and B. Caputo, "From N to N+1: multiclass transfer incremental learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Portland, USA, 2013, pp. 3358–3365. [Online]. Available: https://doi.org/10.1109/CVPR.2013.431

[9] Q. Da, Y. Yu, and Z. Zhou, "Learning with augmented class by exploiting unlabeled data," in *Proceedings of the Conference on Artificial Intelligence*, Québec City, Canada, 2014, pp. 1760–1766. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8388

[10] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009. [Online]. Available: https://doi.org/10.1109/TKDE.2008.239

[11] S. L. Phung, A. Bouzerdoum, and G. H. Nguyen, "Learning pattern classification tasks with imbalanced data sets," *Patern Recognition*, pp. 193–208, 2009.

[12] P. Kulkarni and R. Ade, "Incremental learning from unbalanced data with concept class, concept drift and missing features : A review," *International Journal of Data Mining and Knowledge Management Process*, vol. 4, no. 6, pp. 15–29, 2014.

[13] A. Haque, L. Khan, and M. Baron, "SAND: semi-supervised adaptive novel class detection and classification over data stream," in *Proceedings of the Conference on Artificial Intelligence*, Phoenix, USA, 2016, pp. 1652–1658. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12335

[14] X. Mu, F. Zhu, J. Du, E. Lim, and Z. Zhou, "Streaming classification with emerging new class by class matrix sketching," in *Proceedings of the Conference on Artificial Intelligence*, San Francisco, USA, 2017, pp. 2373–2379. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14514

[15] X. Mu, K. M. Ting, and Z. Zhou, "Classification under streaming emerging new classes: A solution using completely-random trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1605–1618, 2017. [Online]. Available: https://doi.org/10.1109/TKDE.2017.2691702

[16] Y. Zhu, K. M. Ting, and Z. Zhou, "Multi-label learning with emerging new labels," in *Proceedings of the International Conference on Data Mining*, Barcelona, Spain, 2016, pp. 1371–1376. [Online]. Available: https://doi.org/10.1109/ICDM.2016.0188

[17] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, Bridgetown, Barbados, 2005. [Online]. Available: http://www.gatsby.ucl.ac.uk/aistats/fullpapers/198.pdf

[18] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *SIGKDD Explorations*, vol. 6, no. 1, pp. 90–105, 2004. [Online]. Available: http://doi.acm.org/10.1145/1007730.1007731

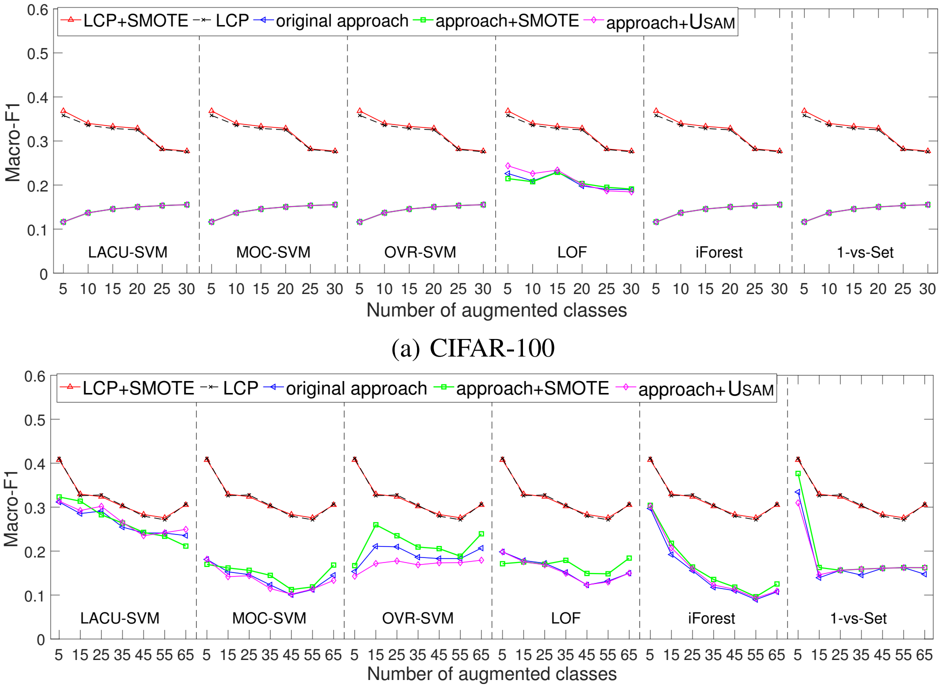
[19] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of human genetics*, vol. 7, no. 2, pp. 179–188, 1936.

[20] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proceedings of the Advances in Neural Information Processing Systems*, Vancouver, Canada, 2003, pp. 321–328. [Online]. Available: http://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency

[21] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. Morgan & Claypool Publishers, 2009. [Online]. Available: https://doi.org/10.2200/S00196ED1V01Y200906AIM006

[22] M. Zhang, Y. Li, and X. Liu, "Towards class-imbalance aware multi-label learning," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2015, pp. 4041–4047. [Online]. Available: http://ijcai.org/Abstract/15/567

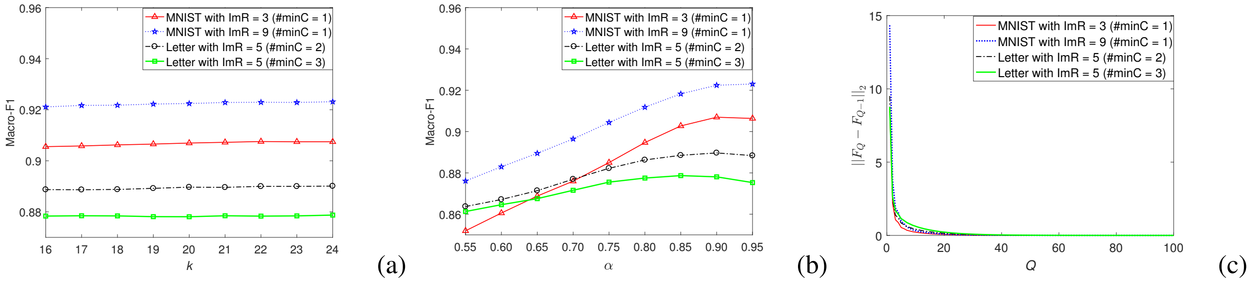
[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on*



(a) CIFAR-100

(b) Caltech256

Fig. 2: The impact analysis of open level. (a) Performance of comparison methods with different number of augmented classes on CIFAR-100 data set. (b) Performance of comparison methods with different number of augmented classes on Caltech256 data set.



With different k

With different α

With different Q

Fig. 3: The influence of parameters k, α and Q in LCP on MNIST data set and Letter data sets.

Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA, 2005, pp. 886–893. [Online]. Available: https://doi.org/10.1109/CVPR.2005.177

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002. [Online]. Available: https://doi.org/10.1613/jair.953

[25] R. M. Rifkin and A. Klautau, “In defense of one-vs-all classification,” *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004. [Online]. Available: http://www.jmlr.org/papers/v5/rifkin04a.html

[26] M. M. Breunig, H. Kriegel, R. T. Ng, and J. Sander, “LOF: identifying density-based local outliers,” in *Proceedings of the International Conference on Management of Data*, Dallas, USA, 2000, pp. 93–104. [Online]. Available: http://doi.acm.org/10.1145/342009.335388

[27] F. T. Liu, K. M. Ting, and Z. Zhou, “Isolation forest,” in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, 2008, pp. 413–422. [Online]. Available: https://doi.org/10.1109/ICDM.2008.17

[28] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, “Toward open set recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, 2013. [Online]. Available: https://doi.org/10.1109/TPAMI.2012.256

[29] C. Chang and C. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961199