Learning from Incomplete Labeled Data via Adversarial Data Generation

Wentao Wang*, Tyler Derr†, Yao Ma*, Suhang Wang‡, Hui Liu*, Zitao Liu§ and Jiliang Tang*
*Michigan State University, †Vanderbilt University, ‡Pennsylvania State University, §TAL Education Group
{wangw116, mayao4, liuhui7, tangjili}@msu.edu, tyler.derr@vanderbilt.edu, szw494@psu.edu, liuzitao@tal.com

***Abstract*—Positive and unlabeled (PU) learning aims to obtain a well-performed classifier via an incomplete binary training set, in which only a part of labels of one category is known while the rest are unknown. However, in many real-world applications such as image recognition, the collected data samples often involve more than two categories. Moreover, only a small portion of the collected samples might have associated labels due to some practical reasons, and these labeled samples cannot always cover all the categories. We refer to this type of data as *incomplete labeled data*. In this paper, we first formally define the incomplete labeled data learning problem and then aim to tackle it via adversarial data generation. Specifically, we propose a novel generative framework LILA, which can produce synthetic labeled samples for both partially labeled categories and unlabeled categories. To enforce that the generated samples for unlabeled categories can associate with correct labels, we integrate two active learning processes into the LILA framework for selecting unlabeled samples in the collected sample set to query their labels effectively. After LILA has been well trained, a classifier can be trained on the balanced augmented data set consisting of both generated and original labeled samples. Extensive experiments on real image data demonstrate the effectiveness of our proposed framework. We release the implementation of the proposed framework via `https://github.com/wentao-repo/LILA`.**

***Index Terms*—incomplete labeled data, generative model, active learning**

I. Introduction

We have recently witnessed the success of deep learning techniques in tackling many real-world tasks, such as image recognition [1] and machine translation [2]. These great achievements are in part contributed by the existence of large-scale labeled data sets since modern deep learning models require sufficient supervised information to learn their tens of thousands of parameters as well as complicated network architectures for benefiting downstream tasks. However, this requirement cannot always be satisfied in many real-world applications, as annotating labels for collected data samples is usually an expensive and time-consuming process [3]. In reality, we are likely to have only a small amount of the collected data samples with associated labels where these labeled samples cannot cover all categories in the collected data set. We refer to such data as *incomplete labeled data*.

In this paper, we focus on the problem of learning from incomplete labeled data. Positive and unlabeled (PU) learning [4], [5] can be treated as a special case of the incomplete labeled data learning problem, which aims to obtain a well-performed classifier via binary incomplete labeled data where only a part of labels of one category is annotated while the rest are unknown. As existing PU learning methods only focus on tackling incomplete data with binary categories, they cannot be directly extended to the incomplete labeled data learning problem. Moreover, due to limited governable resources like budget and time cost, in practice, only a small portion of collected samples being annotated with corresponding labels cannot guarantee that all categories of interest are covered by the labeled portion of the collected data set. Thus, traditional semi-supervised learning methods cannot handle such incomplete labeled data where multiple categories are totally unlabeled. In addition, the incomplete labeled data learning problem can also be regarded as one extreme case of the multi-class imbalanced learning problem [6], [7], in which all minority classes are unlabeled. To learn from incomplete labeled data, we are faced with two main challenges: 1) how to train a classifier when only given small amounts of labeled data; and 2) without any prior knowledge, how to build correct mappings between data samples of totally unlabeled categories and their corresponding labels.

Recently, generative adversarial learning models, such as Generative Adversarial Nets (GANs) [8] and its variants [9], [10], have shown their great power on generating various kinds of synthetic data samples, which paves us one possible way to tackle the aforementioned challenges. Specifically, we propose a GAN-based generative model that trains on the incomplete labeled data to produce high-quality synthetic labeled samples for both partial labeled categories and unlabeled categories. A complete and balanced augmented data set can be obtained by merging generated synthetic labeled samples with original labeled samples. Then, a classifier can be trained on this augmented data set, which correspondingly addresses the first challenge. In addition, since the incomplete labeled data cannot provide any guidance about the label information for the unlabeled categories, we introduce active learning techniques [11], [12] to assist the synthetic sample generation process, especially for samples of the unlabeled categories. In particular, we design novel selection strategies to choose unlabeled samples for querying their labels. With the help of active learning, the generated samples can have correct labels for all categories, which can address the second challenge. The key contributions of this work are summarized below.

- We formalize the problem of incomplete labeled data learning, which can be applied to many real applications;

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00170

- We propose an adversarial data generation solution for learning from incomplete labeled data; and
- We conduct extensive experiments on real image data sets to verify the effectiveness of our proposed framework.

II. Related Work

A. Active Learning

Active learning techniques are proposed to help select the valuable unlabeled data to annotate, as labeled data is often expensive to obtain, whereas unlabeled data is copious in many real-world applications [13]. Generally, they can be roughly classified into three different learning scenarios: stream-based selective sampling, membership query synthesis and pool-based sampling [14]. In the stream-based selective sampling setting, all unlabeled data samples are picked up one by one, and the learner decides whether or not they should be labeled [15]. In the membership query synthesis setting, the learner can generate synthetic samples from the entire space and make quires for these not pre-existing data [16]. The majority of existing active learning techniques belong to the pool-based sampling setting, where the learner maintains an unlabeled sample pool and selects from the pool based on some selection criteria [17]. More details about active learning can be found in a related survey [14].

B. Generative Adversarial Nets

The Generative Adversarial Nets (GANs) was first presented in [8], which consists of a generator G and a discriminator D. The generator G takes a random noise sampled from some probability distribution as input and generates a synthetic sample to fool the discriminator D, while the discriminator aims to differentiate if the input sample is from the generator or the real training set. These two components fight against each other and improve themselves gradually [18]. GANs have achieved impressive performance on the synthetic data generation task, and a large number of its variants have been proposed for handling various applications. For example, by concatenating label information with both real input data and random noises to feed the discriminator and the generator separately, Conditional GAN [9] can produce synthetic samples conditioned on class labels. More content about generative adversarial nets can be found in a related survey [19].

III. The Proposed Framework

A. Problem Formulation

***Definition** 1 (**Incomplete Labeled Data**):* Incomplete labeled data is defined as a set of data samples $\mathbf{D}$ coming from k different categories where only a small portion of the samples in $\mathbf{D}$ have been labeled to p categories with $(2 \leq p \leq k-2)$, and the rest of the samples are unlabeled.

***Definition** 2 (**Incomplete Labeled Data Learning**):* Given incomplete labeled data $\mathbf{D}$ as Definition 1, the goal is to learn a multi-class classifier that can accurately predict the labels of data samples in the test set from all k categories.

Without loss of generality, the p categories with partially labeled samples are denoted as *positive classes*, and the other

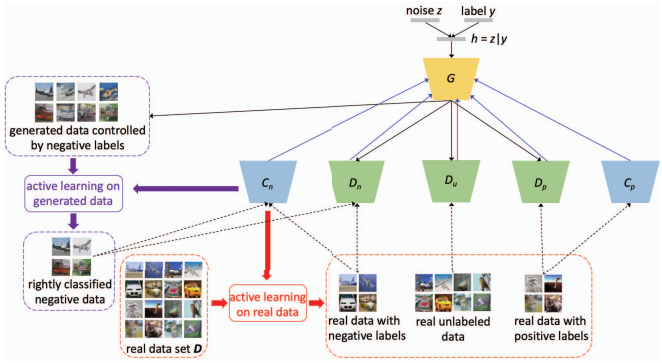


Fig. 1. An overview of our proposed framework LILA. The black solid lines indicate the direction of the generated data flow while the black dash lines denote the direction of the real data flow. The solid blue lines represent the supervision signals flow during framework training.

$k-p$ categories are referred to as *negative classes*. We, therefore, call the labels of positive classes as *positive labels* and others as *negative labels*. In this work, we target on an adversarial data generation solution that can synthetically produce a set of high-quality labeled data samples $\mathbf{D}^s$ covering all k categories. In this way, a multi-class classifier can be trained on the constructed balanced augmentation labeled data set $\mathbf{D}^a = \mathbf{D}^s \cup \mathbf{D}^o$, where $\mathbf{D}^o$ contains the labeled samples of positive classes from the incomplete data set $\mathbf{D}$.

B. An Overview

Inspired by the impressive performance of GANs and its variants on numerous data generation tasks, we present a novel generative model, i.e., Learning from Incomplete Labeled dAta (LILA), to produce realistic synthetic data samples for all k categories involved in the incomplete data set $\mathbf{D}$. Figure 1 demonstrates an overview of our LILA framework. Next, we will introduce each component in LILA and the details of two active learning processes will be later discussed in Section IV.

C. Generator

In order to generate realistic synthetic labeled data samples for all k categories in $\mathbf{D}$, the generator G adopted in LILA takes both random noises $\mathbf{z}$ and one-hot embeddings of labels $\mathbf{y}$ as input and aims to produce synthetic samples with expected labels. The one-hot embedding of any label $\mathbf{y}_i$ controls the category of the generated synthetic samples as $\mathbf{y}_i$ and the random noises $\mathbf{z}$ follow some prior distribution that allows generated samples within the same category to be diverse.

As shown in Figure 1, to make sure the generator can generate realistic labeled data samples, especially for negative categories, we designed three discriminators and two classifiers to guide the generator training phase: (1) the discriminator D_u is designed to guarantee that the synthetic data samples generated by G can satisfy the real data distribution of the unlabeled data set; (2) the discriminator D_p and D_n are used for making sure that the synthetic data samples can follow the data distribution of the real positive labeled data set and real negative labeled data set, respectively; and (3) the classifiers

C_p and C_n are targeted on guiding the generator G to produce labeled positive and negative samples with expected labels, separately.

D. Discriminators

In order to generate synthetic labeled data for all k categories, we need prior knowledge about mappings between data samples and categorical labels. However, in the incomplete labeled data learning problem, we only have a small amount of positive labeled data while the data samples for the negative classes are totally unlabeled. Moreover, as we mentioned before, the annotation process in practice is often expensive and time-consuming; and typically we cannot select as many unlabeled real samples as we want from the incomplete data set $\mathbf{D}$ and query their associated labels. Hence, as shown in Figure 1, before training our LILA framework, we first perform an active learning process on the incomplete data set $\mathbf{D}$ to obtain several labeled real data samples for negative classes. After this active learning process, we obtain three types of real data that can be used to train LILA, i.e., a set of real positive labeled data $\mathbf{D}^p$, a set of real negative labeled data $\mathbf{D}^n$ and an unlabeled real data set $\mathbf{D}^u$.

The discriminator D_u is trained on unlabeled real data $\mathbf{D}^u$ and all synthetic data samples generated by the generator G. It is used for differentiating whether input samples are real or fake. If a sample comes from the real unlabeled sample set $\mathbf{D}^u$, the discriminator D_u regards it as a real sample; otherwise it will be considered as a fake sample. The loss function for training both the discriminator D_u and the generator G can be written as

$$\begin{aligned}\mathcal{L}_{(G,D_u)} = &\frac{1}{|\mathbf{D}^u|}\sum_{\mathbf{x}_i\in\mathbf{D}^u}(D_u(\mathbf{x}_i)-1)^2 \\ &+\frac{1}{|\mathbf{H}|}\sum_{\mathbf{h}_i\in\mathbf{H}}(D_u(G(\mathbf{h}_i))-0)^2,\end{aligned} \tag{1}$$

where $\mathbf{h}_i$ denotes that the random noise $\mathbf{z}_i$ is conditioned on the label $\mathbf{y}_i$, $\mathbf{H}$ is a set of random noise that is conditioned on labels of any categories.

The discriminator D_p is trained on real positive labeled data $\mathbf{D}^p$ and generated synthetic data samples controlled by positive labels. The discriminator D_p is to identify each input sample from the set $\mathbf{D}^p$ or generated by the generator G under the control of positive labels. The loss function for training both the discriminator D_p and the generator G is formulated as

$$\begin{aligned}\mathcal{L}_{(G,D_p)} = &\frac{1}{|\mathbf{D}^p|}\sum_{\mathbf{x}_i\in\mathbf{D}^p}(D_p(\mathbf{x}_i)-1)^2 \\ &+\frac{1}{|\mathbf{H}^p|}\sum_{\mathbf{h}_i\in\mathbf{H}^p}(D_p(G(\mathbf{h}_i))-0)^2,\end{aligned} \tag{2}$$

where $\mathbf{H}^p$ indicates a set of random noise that is conditioned on different positive labels.

Similarly, the real negative labeled data set $\mathbf{D}^n$ obtained by the active learning process can be utilized to train the discriminator D_n. However, the number of labeled negative data samples in set $\mathbf{D}^n$ may not be sufficient for training a good D_n when given the limited budget for querying labels. Therefore, instead of only performing active learning process on the real unlabeled data samples, we design another active learning process on the synthetic generated data for selecting generated negative data samples $\mathbf{D}^{n_g}$ that could help train an even better discriminator D_n and then adding them into the real negative data set $\mathbf{D}^n$. Using $\mathbf{H}^n$ to denote a set of random noise that are conditioned on different negative labels and $\overline{\mathbf{D}}^n$ to represent the enlarged real negative data set $\mathbf{D}^n$ after appending new selected negative data samples, i.e., $\overline{\mathbf{D}}^n = \mathbf{D}^n \cup \mathbf{D}^{n_g}$, the loss function for training both the discriminator D_n and the generator G can be defined as

$$\begin{aligned}\mathcal{L}_{(G,D_n)} = &\frac{1}{|\overline{\mathbf{D}}^n|}\sum_{\mathbf{x}_i\in\overline{\mathbf{D}}^n}(D_n(\mathbf{x}_i)-1)^2 \\ &+\frac{1}{|\mathbf{H}^n|}\sum_{\mathbf{h}_i\in\mathbf{H}^n}(D_n(G(\mathbf{h}_i))-0)^2.\end{aligned} \tag{3}$$

E. Classifiers

To ensure that the generated data samples have expected labels for all the k categories, we integrate two classifiers C_p and C_n into our LILA framework for providing useful supervision information to guide the generator training process.

C_p is a p-class classifier trained on the real positive labeled data set $\mathbf{D}^p$ and used for predicting labels for generated data samples controlled by positive labels. The loss function to train C_p can be written as

$$\mathcal{L}_{C_p} = \frac{1}{|\mathbf{D}^p|}\sum_{\mathbf{x}_i\in\mathbf{D}^p}\|C_p(\mathbf{x}_i)-\Gamma_{\mathbf{x}_i}\|_2^2, \tag{4}$$

where $\Gamma_{\mathbf{x}_i}$ is the one-hot embedding of the true label of real sample $\mathbf{x}_i$ from one of the positive classes.

Similarly, the $(k-p)$-classes classifier C_n is trained on the enlarged real negative data set $\overline{\mathbf{D}}^n$ to facilitate the training process. The loss function for training C_n is given as

$$\mathcal{L}_{C_n} = \frac{1}{|\overline{\mathbf{D}}^n|}\sum_{\mathbf{x}_i\in\overline{\mathbf{D}}^n}\|C_n(\mathbf{x}_i)-\Gamma_{\mathbf{x}_i}\|_2^2. \tag{5}$$

The classifiers C_p and C_n are designed for guiding the generator G to produce labeled positive and negative samples with expected labels, respectively. Let $\Gamma_{\mathbf{h}_i}$ denote one input label fed to the generator G. The loss functions for training the generator G with respect to these two classifiers can be defined as

$$\mathcal{L}_{(G,C_p)} = \frac{1}{|\mathbf{H}^p|}\sum_{\mathbf{h}_i\in\mathbf{H}^p}\|C_p(G(\mathbf{h}_i))-\Gamma_{\mathbf{h}_i}\|_2^2, \tag{6}$$

and

$$\mathcal{L}_{(G,C_n)} = \frac{1}{|\mathbf{H}^n|}\sum_{\mathbf{h}_i\in\mathbf{H}^n}\|C_n(G(\mathbf{h}_i))-\Gamma_{\mathbf{h}_i}\|_2^2. \tag{7}$$

F. Objective Function of Training LILA

With the components introduced above, the objective function of training our LILA framework is given as

$$\begin{aligned}\min_{\theta_G,\theta_{C_p},\theta_{C_n}}\ \max_{\theta_{D_u},\theta_{D_p},\theta_{D_n}}\ &\mathcal{L}_{(G,D_u)}+\lambda_1\mathcal{L}_{(G,D_p)}+\lambda_2\mathcal{L}_{(G,D_n)} \\ &+\lambda_3\mathcal{L}_{(G,C_p)}+\lambda_4\mathcal{L}_{(G,C_n)} \\ &+\lambda_5\mathcal{L}_{C_p}+\lambda_6\mathcal{L}_{C_n},\end{aligned} \tag{8}$$

where θ_G, θ_{C_p}, θ_{C_n}, θ_{D_u}, θ_{D_p} and θ_{D_n} are parameters to control the generator G, classifiers C_p and C_n, and discriminators D_u, D_p and D_n, respectively. $\lambda_1, \ldots, \lambda_6$ are hyperparameters to control the contribution of each component.

After LILA is well trained, the generator G is able to produce realistic labeled data samples for all the k categories.

IV. Active Learning Strategies

Since more than two categories are unlabeled in the incomplete data set $\mathbf{D}$, we need to build correct mappings between negative labels and unlabeled data samples belonging to negative classes. As active learning techniques have been successfully applied in annotating unlabeled data, we introduce active learning techniques to select and query labels for data samples of negative classes.

A. Active Learning on Real Data

The first active learning process in LILA is to gather labeled data samples for negative classes. Moreover, the generated synthetic labeled data from LILA should be diverse; thus, the diversity among these selected query candidates from $\mathbf{D}$ should also be considered. Our designed adaptive active learning method is consisting of two phases. In the first phase, we use a small budget to randomly sample unlabeled data from $\mathbf{D}$ and query their labels. The first phase can be regarded as an initialization phase since we aim to ensure that all the k categories have labeled data samples after random sampling.

The second phase is illustrated in Figure 2. We first train the classifier C_n based on the real negative labeled samples obtained from the first phase. Although the number of negative labeled data may be insufficient for training a good classifier at this time, the classifier C_n can be roughly trained and has an initial capacity to distinguish different negative data samples. With the classifier C_n, we use all unlabeled samples in the set $\mathbf{D}$ to feed the classifier C_n and form a query candidate sample set based on the outputs of the classifier C_n. Specifically, for each sample, the classifier C_n will produce a $(k-p)$-dimensional output, where the value in each dimension denotes the likelihood of that sample belonging to the t-th category in the $k-p$ negative categories. Here we name the likelihood as the certainty score. Then within each category, we sort the certainty scores in descending order and the resulted sequence is called the categorical certainty sequence $\mathcal{S}$. Thus, for each category, the unlabeled sample corresponding to the largest certainty score has the largest likelihood of belonging to this category. Next, we select unlabeled samples from the front part of the categorical certainty sequence of each category as our selected query candidate samples and query their labels. Then, we utilize all available real negative labeled data to again train the classifier C_n and in an iterative cycle, we again select another set of candidate samples to query the labels before again training C_n. In this way, each category can have the same number of query candidate samples and each query candidate sample has a larger likelihood to that category.

In terms of the sample diversity, we adaptively change the position of each categorical certainty sequence $\mathcal{S}$ we used for selecting query candidates. For example, in the beginning, we choose the unlabeled samples from the first position of $\mathcal{S}$. This is because the classifier C_n at this stage only has the limited capacity to calculate certainty scores. However, as C_n is trained better and better, we can move to latter positions of $\mathcal{S}$ to select query candidate samples. The reason we do not always choose the first position is that C_n is likely to give larger certainty scores for unlabeled samples which are closer to existing negative labeled data samples used to train C_n. Via the above active learning process, we can obtain three real data sets, i.e., a real positive labeled data set $\mathbf{D}^p$, a real negative labeled data set $\mathbf{D}^n$ and an unlabeled real data set $\mathbf{D}^u$.

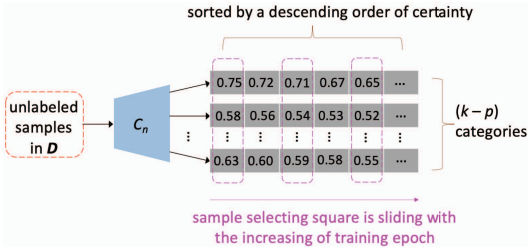


Fig. 2. An illustration of the second phase of our proposed adaptive active learning method on the real unlabeled data.

B. Active Learning on Generated Data

The classifier C_n may not be well trained based solely on $\mathbf{D}^n$. Moreover, our ultimate goal is to train a good generator G to produce realistic synthetic data samples for all the k categories as compared to just obtaining a good classifier C_n for prediction. Thus, the classifier C_n should assist the training process of the generator G. In order to gather more negative labeled data to train the classifier C_n and make the training of the generator G to be more effective, we introduce another active learning process into LILA to obtain the generated negative labeled data set $\mathbf{D}^{n_g}$.

Specifically, similar to the second phase of the first active learning process, during the training process of the classifier C_n, we feed C_n with generated data samples controlled by negative labels and form categorical certainty sequences for each $k-p$ category of negative classes based on the outputs of C_n. Then, for each categorical certainty sequence $\mathcal{S}$, we select the generated data sample that is located in the i-th position of $\mathcal{S}$ and query its associated label. If the queried label is the same as the original label used for feeding the generator G to produce this sample, all the generated data samples that are located before the i-th position of $\mathcal{S}$, i.e., these samples have a larger likelihood to this negative category, will be added into $\mathbf{D}^{n_g}$. In experiments, we note that this process even further improves the efficiency of LILA to obtain samples for the negative categories, which is important due to the limited label querying budget. Following this way, the generated negative labeled data $\mathbf{D}^{n_g}$ can be constructed of considerable size by using only a small amount of query budgets. Finally, the enlarged negative labeled data set $\overline{\mathbf{D}}^n$ is a combination of both real negative labeled data set $\mathbf{D}^n$ and generated negative labeled data set $\mathbf{D}^{n_g}$, i.e., $\overline{\mathbf{D}}^n = \mathbf{D}^n \cup \mathbf{D}^{n_g}$.

TABLE I
STATISTICAL INFORMATION OF DATA SETS.

Statistic index	Data set	
	MNIST-In	CIFAR10-In
# features	784	3,072
# total categories	10	10
# positive classes	5	5
# training positive labeled data	250	2,500
# training unlabeled data	49,750	47,500
# total budgets for querying	250	2,500
# test data	10,000	10,000

V. EXPERIMENTS

A. Experiment Settings

1) Data Sets: Our experiments are based on two real image data sets, including MNIST [20] and CIFAR10 [21]. Since both MNIST and CIFAR10 are fully labeled data sets and cannot be directly used in the incomplete labeled data learning problem, based on these two data sets, we construct two incomplete data sets MNIST-In and CIFAR10-In for evaluation. We will take the MNIST as an example to introduce how to construct its corresponding incomplete data set MNIST-In. First, among the total ten categories, we randomly choose five of them as positive classes and the other five categories as negative classes. Second, for data samples belonging to negative classes, we remove their corresponding labels and regard them as unlabeled data; and for data samples belonging to positive classes, we randomly select a very small number of them as labeled data. Specifically, we choose 1% labeled data in each positive class as partial labeled positive samples while treating all remaining data as unlabeled data. Third, we set the total available budgets for querying labels in the active learning process equal to the total number of positive labeled samples in the incomplete data set. The key statistics of MNIST-In and CIFAR10-In are summarized in Table I.

2) Evaluation Metrics: Due to the fact that our goal is to solve the incomplete labeled data learning problem via generating synthetic labeled data, we adopt a classification task to evaluate the quality of the generated samples. We exploit two popular classifiers LeNet-5 [20] and Pre-activation ResNet [22], abbreviated as PreActResNet, to verify whether our LILA framework can generate high-qualify synthetic labeled data to better train the classifiers LeNet-5 and PreActResNet on the MNIST-In and CIFAR10-In, respectively.

B. Data Classification Performance

For evaluating the effectiveness of LILA on generating high-quality labeled samples, we explore whether the synthetic labeled samples generated by LILA can improve the classification performance. Obviously, classifiers LeNet-5 and PreActResNet cannot be trained on MNIST-In and CIFAR10-In directly as negative classes are totally unlabeled in these two incomplete data sets. A simple solution is that we can apply a random sampling process on unlabeled data (for their respective data sets) to obtain query candidate samples and then query labels for them. After that, all categories involved in the incomplete data sets are most likely to have labeled samples; hence, classifiers LeNet-5 and PreActResNet can be trained as normal. We denote this simple solution as *Original + RS*. However, because of the limited budget for querying labels, the number of labeled data samples after a random sampling process for training classifiers may be insufficient. Therefore, we utilize our LILA framework to produce sufficient synthetic labeled data for all categories and train the classifiers LeNet-5 and PreActResNet on the balanced augmented data sets which consist of synthetic generated samples and original positive labeled samples. Similarly, we do the same for several representative generative baseline methods: 1) CGAN [9], which produces synthetic samples with expected labels by concatenating data and label information in the model learning process; 2) ACGAN [10], which can produce more clear and diverse synthetic images conditioned on different labels via introducing an auxiliary classifier into GANs; 3) SSGAN [23], which is a semi-supervised version of CGAN that is able to utilize all available data samples during the training process; 4) CVAE [24], which can generate synthetic samples controlled by labels using a Variational Autoencoder (VAE) based framework; 5) SSVAE [25], which is a semi-supervised generative model that improves the performance of VAE framework on the semi-supervised scenario.

Table II shows the classification accuracy of two classifiers trained on their corresponding data sets formed by baseline methods and our LILA framework. The total budget on MNIST-In data set is 250 and on CIFAR10-In data set is 2,500. We repeat experiments 5 times and report the average accuracy results and associated standard deviation. Since all baselines cannot work on the incomplete data sets MNIST-In and CIFAR10-In directly, we first perform a random sampling process on incomplete data sets to obtain labeled data samples for negative classes and then train baseline methods separately on the new data sets which contain both positive and negative labeled data samples. As shown in Table II, all generative models can produce effective synthetic labeled data to help train classifiers LeNet-5 and PreActResNet better, as compared to only performing random sampling on the incomplete data sets and then training classifiers directly on the original data paired with the randomly sampled, queried, and labeled data. Among them, the classifier trained on the synthetic labeled data generated by semi-supervised generative models SSGAN and SSVAE that exploiting unlabeled data in the model training process can achieve relatively better performance than these trained on other baseline methods that only utilize labeled data samples. Most importantly, the synthetic labeled samples generated by our LILA framework on two data sets help classifiers LeNet-5 and PreActResNet achieve the best classification performance, separately. The main reason is that our designed two active learning processes is able to select more representative unlabeled real data and generated data for negative classes to query labels separately, and these selected data samples can help train the discriminator D_n and the classifier C_n greatly, and thus make contributions to the training process of the generator G in LILA. In addition, our LILA framework utilizes all available real data including unlabeled data. Thus, LILA has a much larger capacity to

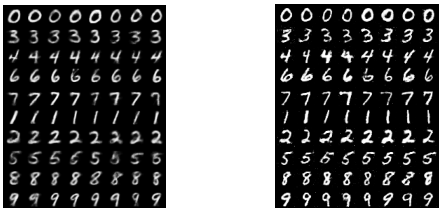
TABLE II
CLASSIFICATION ACCURACY OF CLASSIFIERS LENET-5 AND PREACTRESNET TRAINING ON DIFFERENT DATA SETS.

Methods	Accuracy	
	MNIST-In	CIFAR10-In
Original + RS	0.8476 ± 0.0072	0.7366 ± 0.0110
RS + CGAN	0.8642 ± 0.0216	0.7519 ± 0.0086
RS + ACGAN	0.8582 ± 0.0046	0.7527 ± 0.0092
RS + SSGAN	0.8895 ± 0.0061	0.7532 ± 0.0085
RS + CVAE	0.8749 ± 0.0044	0.7559 ± 0.0040
RS + SSVAE	0.8935 ± 0.0027	0.7537 ± 0.0120
LILA	**0.9100 ± 0.0025**	**0.7566 ± 0.0080**

learn the real data distribution of the incomplete data set and enforces the generated samples to be more realistic that will be shown in the next subsection.

C. Case Studies

For checking whether our LILA framework can generate realistic synthetic labeled data, we visualize the synthetic labeled samples generated from the MNIST-In data set. As a comparison, we also visualize the synthetic labeled samples generated by one baseline method *RS + SSVAE*, since it helps the classifier LeNet-5 achieve the best classification accuracy among all baseline methods as shown in Table II. For both figures contained in Figure 3, the first five rows show generated digits for five positive classes and the remaining five rows are generated digits of negative classes. As shown in Figure 3, comparing with the baseline method, our LILA framework can produce more realistic and diverse synthetic data samples for all ten classes with expected labels, which effectively verifies the generation capacity of our LILA framework.



(a) By RS + SSVAE (b) By LILA

Fig. 3. Synthetic image samples generated by one baseline method and our LILA framework from the MNIST-In data set.

VI. CONCLUSIONS

In this paper, we present a challenging real-world problem of learning from incomplete labeled data. We propose a novel framework LILA to deal with the incomplete labeled data learning problem via generating high-quality synthetic labeled data. Experimental results demonstrate the effectiveness of our LILA framework on generating synthetic labeled data for all categories contained in the given incomplete labeled data set. In the future, we plan to extend our LILA framework to solve the multi-class imbalance problem via generating synthetic labeled data for minority classes.

ACKNOWLEDGMENT

Wentao Wang, Yao Ma, Hui Liu and Jiliang Tang are supported by the National Science Foundation (NSF) under grant numbers IIS1907704, IIS1714741, IIS1715940 and IIS1845081. Suhang Wang is supported by the NSF under grant number IIS1909702. Zitao Liu is supported by Beijing Nova Program (Z201100006820068) from Beijing Municipal Science & Technology Commission.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
[2] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *ICML*. JMLR. org, 2017, pp. 1243–1252.
[3] V. C. Raykar and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 491–518, 2012.
[4] X.-L. Li and B. Liu, "Learning from positive and unlabeled examples with different data distributions," in *ECML*, 2005, pp. 218–229.
[5] J. Bekker and J. Davis, "Learning from positive and unlabeled data: A survey," *arXiv preprint arXiv:1811.04820*, 2018.
[6] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 687–719, 2009.
[7] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.
[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.
[9] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
[10] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *ICML*. JMLR. org, 2017, pp. 2642–2651.
[11] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *CVPR*, 2009, pp. 2372–2379.
[12] X. Li and Y. Guo, "Adaptive active learning for image classification," in *CVPR*, 2013, pp. 859–866.
[13] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and S. Y. Philip, "Active learning: A survey," in *Data Classification: Algorithms and Applications*. CRC Press, 2014, pp. 571–605.
[14] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.
[15] S. Dasgupta, D. J. Hsu, and C. Monteleoni, "A general agnostic active learning algorithm," in *NIPS*, 2008, pp. 353–360.
[16] J.-J. Zhu and J. Bento, "Generative adversarial active learning," *arXiv preprint arXiv:1702.07956*, 2017.
[17] Y. Zhang, M. Lease, and B. C. Wallace, "Active discriminative text representation learning," in *AAAI*, 2017.
[18] Z. Gan, L. Chen, W. Wang, Y. Pu, Y. Zhang, H. Liu, C. Li, and L. Carin, "Triangle generative adversarial networks," in *NIPS*, 2017, pp. 5247–5256.
[19] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (gans): A survey," *IEEE Access*, vol. 7, pp. 36 322–36 333, 2019.
[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[21] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *ECCV*. Springer, 2016, pp. 630–645.
[23] K. Sricharan, R. Bala, M. Shreve, H. Ding, K. Saketh, and J. Sun, "Semi-supervised conditional gans," *arXiv preprint arXiv:1708.05789*, 2017.
[24] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *NIPS*, 2015, pp. 3483–3491.
[25] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *NIPS*, 2014, pp. 3581–3589.