LEARNING FROM POSITIVE AND UNLABELED DATA USING OBSERVER-GAN

Omar Zamzam, Haleh Akrami, Richard Leahy

University of Southern California
Department of Electrical Engineering
Los Angeles, CA, USA

ABSTRACT

In the case of learning from positive and unlabeled data, the input data consist of (1) observations from the positive class and their corresponding labels and (2) unlabeled observations from both positive and negative classes. Generative Adversarial Networks (GANs) have been used to reduce the problem to the supervised setting with the advantage that supervised learning has state-of-the-art accuracy in classification tasks. In order to generate pseudo-negative observations, GANs are trained on positive and unlabeled observations with a modified loss. Using both positive and pseudo-negative observations leads to a supervised learning setting. The generation of pseudo-negative observations that are realistic enough to replace missing negative class samples is a bottleneck for current GAN-based algorithms. By including an additional classifier into the GAN architecture, we describe a novel GAN-based approach. The GAN discriminator instructs the generator to only produce samples that fall into the unlabeled data distribution, while a second classifier (observer) network monitors the GAN training to: (i) prevent the generated samples from falling into the positive distribution; and (ii) learn the features that are the key distinction between the positive and negative observations. Experiments on four image datasets demonstrate that our trained observer network performs better than existing techniques in discriminating between real unseen positive and negative samples.

Index Terms— PU learning, GANs

1. INTRODUCTION

In real-world binary classification problems, it is not unusual to find the cost of labeling one of the classes considerably higher than the cost of labeling the other class. It is sometimes difficult even to label a subset of the data that belongs to one of the classes. This can result in an abundance of data for which we do not know the associated class and a subset of data with labels corresponding to only one class (here, we will assume such labels are ‘positive’). This is referred to as the problem of PU Learning or learning from positive and unlabeled input [1]. A common example of the PU data set is recommendation systems, where previous purchases or user clicks are direct indicators of user interest (positive label). In contrast, labels for all other instances remain unknown (unlabeled) [2]. Another important example of PU datasets is in automatic diagnostic systems, where a specific symptom of disease appears in only a subgroup of patients; the presence of this symptom can be used to label positive label cases, but the absence of this symptom does not necessarily indicate the absence of the disease [3]. PU learning also has applications in the field of gene identification [4], matrix completion [5], clustering [6], and spam detection [7].

In the current study, we provide a technique that uses a Generative Adversarial Network (GAN) [8] to detect features from the unlabeled data for distinguishing between positive and negative classes. The key role for a GAN is to have the generator network learn the distribution of a training set so as to generate additional samples from the same distribution. This is achieved by training a generator and a discriminator network adversarially, i.e. the discriminator tries to identify a data sample as real (coming from the training set) or fake (generated by the generator network). The objective of the generator network is to generate data samples that the discriminator network fails to classify as fake. We use the unlabeled dataset in this context as the input training set to the discriminator. In addition, we use an additional classifier, here called an observer network, to determine whether a data sample comes from the positive set or is fake (generated by the generator network). The objective of our generator network is modified to generate data samples that: (i) the discriminator network fails to classify as fake and (ii) the observer network successfully identifies as being "non-positive". These samples should then be generated exclusively from the negative distribution in the unlabeled data. The discriminator network cannot determine that these generated samples are fake, and the observer network can separate them from the positive samples. After training these three networks iteratively, the observer network learns features that are sufficient to classify between unseen positive and negative samples.

We evaluate the proposed method, Observer-GAN, on four image datasets and show that it outperforms state-of-the-art PU learning methods on binary classification tasks.
2. RELATED WORK

The PU learning problem has been investigated since at least 1998 [9]. Recently there has been a growing interest in developing PU learning methods as a result of the increased use of machine learning/deep learning in a variety of applications where the cost of labeling is high, such as in medicine, marketing, and advertising ([10], [11], [12], [13], [6] [5], [14],[15], [16]).

One of the common approaches in PU learning leverages biased learning, in which unlabeled examples are considered negative examples with label noise, and a binary classifier is trained using a biased cost function that assigns a higher penalty for misclassification of positive examples (clean labels) [10], [5], [17]. Another class of methods for PU learning assumes a positive class prior (\(P_y = 1\)) is known, which facilitates the training and tuning of a binary classifier. Using this information, training can be stopped when the proportion of identified positive examples in an unlabeled validation set is equal to the positive class prior [18], [16], [19]. However, the true class prior is usually difficult to obtain and so an alternative is to estimate the positive class prior as a first step and then train a binary classifier using this estimated prior [20]. Convergence can be achieved by iterating between these two steps (TED) [15].

Alternatively, we can use a distance metric to find reliable negative examples, simplifying the problem to the supervised setting. Reliable negative examples are the unlabeled examples that are most different from the positive samples [11] [21]. To generate negative samples, a generative model can also be used [14]. In the first step in [14], the so-called D-GAN is trained with a generator network together with a discriminator network that takes both unlabeled and positive data as training set. This discriminator network is trained to classify between the unlabeled data as one class, and both the positive data and generated data as another class. Using this setup, the generator learns to fool the discriminator by generating only negative samples. In the second step, a binary classifier is trained given the positive samples and the generated "negative" samples. We find in experiments that this setup fails when the target dataset is complicated and difficult for a GAN to generate, so the generated samples are not realistic enough to learn from. In these cases, the second-step classifier only learns the difference between real and fake images, as this task is easier to learn than the real-positive versus real-negative sample classification.

Our suggested framework does not directly depend on the quality of the generated images for the classification task. Our Observer network is progressively trained with the generator network, it is forced to learn only the features of the generated images that are consistently generated throughout the training and point to the "non-positive" samples as described in Section 4.

3. PROBLEM SETUP

We denote an unlabeled observation as \(x_U \in X_U \sim P_U\), such that \(X_U\) is the unlabeled dataset and \(P_U\) is the unlabeled data distribution. We defined the positive data sample \(x_P \in X_P \sim P_P\), such that \(X_P\) is the positive dataset and \(P_P\) is the positive data distribution. \(z \sim P_Z\) is a random noise vector where \(P_Z = N(\bar{z}, I)\) is the multivariate normal distribution (\(\bar{z}\) is a zero vector and \(I\) is the identity matrix). \(x_Z\) is a generated sample obtained from the generator. We denote the label for each observation as \(y \in \{0, 1\}\). \(D(\cdot)\) is the output probability of the discriminator, \(G(z)\) is the output sample of the generator network, and \(Ob(\cdot)\) is the output probability of the observer network. \(\alpha\) is the proportion of the positive samples in the unlabeled dataset, i.e., \(P_U = \alpha P_P + (1 - \alpha) P_N\), where \(P_N\) is the negative data distribution.

We aim to learn \(f(x) = p(y = 1 \mid x)\), a classifier that estimate the true label of an observation \(x\). We claim that at the end of the training of our Observer-GAN, we have \(Ob(x) \approx f(x)\).

4. PROPOSED METHOD

The standard GAN [8] discriminator network \((L_D)\) and the generator network \((L_G)\) loss functions are as follows:

\[
L_D = E_{x_U \sim P_U} [H(D(x_U), 1)] + E_{z \sim P_Z} [H(D(G(z)), 0)]
\]  
\[
L_G = E_{z \sim P_Z} [H(D(G(z)), 1)]
\]

where \(H(\hat{y}, y) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})\) is the binary cross entropy loss between \(\hat{y}\) and \(y\). Minimizing the loss in (1) makes the discriminator able to learn the difference between the data in \(P_U\) and the output of the generator \(G(z)\), while minimizing the loss in (2) attempts to fool the classifier into believing its samples are real. Minimizing the two loss functions iteratively, results in a generator that is able to generate new samples that are indistinguishable from the real unlabeled data samples. Our Observer-GAN has the same objective for the discriminator. In addition to this setup, as illustrated in Figure 1, we added a second classifier network (the
Observer). The loss function of the Observer network \( L_{Ob} \) is defined as follows:

\[
L_{Ob} = E_{x \sim P_{p}}[H(Ob(x, 0))] + E_{z \sim P_{z}}[H(Ob(G(z), 1))]
\]

Minimizing this loss forces the observer network to learn features that separate the positive dataset and the generated samples. We refer to these features as "positive features". We also replace the loss function of the generator network defined in (2) with:

\[
L_G = E_{z \sim P_{z}}[H(D(G(z), 1))] + E_{z \sim P_{z}}[H(Ob(G(z), 1))]
\]

The first term in (4) is the same as the first term in (2) while the second term in (4) ensures that the samples that the generator produces are different from positive samples and this difference can be learned by the observer. The generator in this setup attempts to fool the discriminator network, while keeping the loss of the observer network minimal. In this way, the generator network captures the distribution of the samples that are present in the unlabeled dataset (to fool the discriminator), and absent in the positive dataset (to keep \( L_{Ob} \) minimal). Minimizing (1), (3), and (4) iteratively, ensures that the observer network learns appropriate features from these generated samples to distinguish between positive and generated samples.

This training strategy forces the generator network to generate new samples that are closer in distribution to the unlabeled samples, while keeping the features that the observer network did not previously identify as "positive features". The persistence of these features in the generated samples also forces the observer network to only pay attention to the persistently generated negative features, as the rest of the features in the generated samples change more drastically as training progresses.

In the first training steps of Observer-GAN, the generated samples are very different from the positive data samples, and since the generator and observer are not training adversarially, the value of \( L_{Ob} \) is always small, hence, the second term in (4) is almost disabled. As the training proceeds, when the generator starts learning to generate samples that are closer to samples of the unlabeled dataset (by minimizing the first term in (4)), the value of \( L_{Ob} \) increases for the samples \( x \in Z \) that are closer to the positive dataset, penalizing the generator for generating these samples. We chose the name "Observer" because of this behaviour of the network.

This training procedure can result in undesirable over-fitting in which the observer network starts to memorize positive samples. This is due to the fact that the positive training has a small size compared to the (infinite) number of generated samples. In that case, the value of \( L_{Ob} \) never increases, and never affects the learning of the generator. To address this issue and avoid undesirable over-fitting, we randomly re-initialize the observer network weights every \( T \) training epochs. This step has the specific purpose of removing any memory the observer network has about positive samples, so that it always learns meaningful features from the input samples, instead of simply memorizing positive samples. This reinitialization trick has previously been used to enhance GAN performance [22]. Since the idea of using a validation set to monitor the training and avoid overfitting cannot be used in a PU data setting, this reinitialization step represents a solution to ensure the validity of the final classifier. The observed continuous improvement of the quality of the generated samples, justifies the use of reinitialization. We observed that the performance of the Observer-GAN is not very sensitive to the choice of \( T \). We train for 1000 epochs with \( T \in \{50, 100, 200\} \).

5. RESULTS

5.1. Data Preparation

We evaluate the performance of the Observer network on four different datasets: MNIST [23], Fashion-MNIST [24], CIFAR-10 [25], and animal faces (AFHQ) [26]. The positive and negative classes are defined respectively as: even versus odd digits on MNIST dataset, last five classes vs first five classes on Fashion-MNIST (classes: T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot), animal versus not animal images on CIFAR-10, and cat versus dog images on AFHQ.

We define the whole training dataset to be

\[
X = \{x_1, x_2, \ldots, x_p, x_{p+1}, x_{p+2}, \ldots, x_{p+n}\}, \text{ where } p \text{ is the number of positive samples, and } n \text{ is the number of negative samples.}
\]

We randomly select \( \alpha |X_U| \) positive samples to be in the unlabeled set, where \( \alpha \) is the proportion of positive samples in \( X_U \), and \( |X_U| \) is the size of \( X_U \). The remainder \( (1-\alpha)|X_U| \) samples in \( X_U \) are negative. We use \( \alpha = 0.5 \) in the following experiments.

5.2. Baselines

We compare the performance of the Observer network to D-GAN, which trains a binary classifier on the positive sample and previously generated negative samples, and (TED"), which uses an iterative method between estimating \( \alpha \) and training a binary classifier.

Because of the nature of the problem, as explained previously, the use of a validation set to choose the best learner is not feasible. Instead, we use the Fréchet Inception Distance (FID) score [27] as a tool to monitor training. Since the FID score accounts for both the quality and diversity of the generated images, it is a reasonable way to ensure that mode collapse does not happen during training. Monitoring the FID score of the generated images together with reinitialization, allows us to avoid mode collapse and overfitting of the Observer network. As a result, the Observer network tends to improve with training for more epochs. To evaluate the validity of this claim, we train all models in all of the
experiments for 1000 epochs, and assess the average performance of the Observer network on the test set using the last 50 and 100 epochs. We follow the same approach when training (TED\textsuperscript{a}). However, for D-GAN, the second step classifier cannot avoid the overfitting problem, hence, we use a fully-labeled validation set to pick the best performing classifier to compare to. We train the GAN model in D-GAN for 1000 epochs, and we train a classifier for 1000 epochs on the output of the GAN after 100, 200, and 1000 epochs.

Table 1. Details of the experiments; Left-most column is the dataset, and upper-most row is the method used. We report the mean and standard deviation of the accuracy (%) of the last 50 and 100 epochs when using TED\textsuperscript{a} or Observer network, and the best performing model when using D-GAN.

| Method       | Early Stop | 50  | 100 | 50  | 100  |
|--------------|------------|-----|-----|-----|------|
| AFIN         | ~50        | 86.8±1.2 | 89.9±1.3 | 91.1±1.1 | 90.1±1.2 |
| CIFAR        | 82         | 88.2±5.6 | 87.7±4.6 | 90.6±0.2 | 88.8±1.4 |
| MINIST       | 98.3       | 97.7±0.4 | 97.7±0.4 | 98.3±0.2 | 97.8±1.6 |
| Fashion MINIST | 89.6     | 88.3±0.9 | 88.1±1  | 92.0±0.2 | 92.2±1  |

Table 1 shows the positive-versus-negative classification accuracy of each of the methods, applied on each of the datasets. The second step classifier of D-GAN fails for more complicated datasets since it learns only the difference between the real and fake samples and therefore predicts a positive label for all unseen realistic data. D-GAN appears to work well only for datasets that can easily be generated, which is not always the case in real-world PU datasets. The Observer-GAN shows the best performance using the average accuracy of last 50 and 100 epochs, which empirically supports our claim about avoiding overfitting.

Figure 2 shows sample output images from the generator of both Observer-GAN and D-GAN. It is evident that D-GAN does fall into a mode collapse, where the generator network generates almost identical images. Although image quality is still limited in Observer-GAN, the classification task is not affected.

6. DISCUSSION

A human being can classify between cats and dogs with reasonable accuracy by looking only at a specific features (e.g. the tail). In general, neural networks are trained to classify between two classes of images in a similar way, i.e., given a labeled training set, the neural network is trained to highlight the important parts of the images concerning the classification task. Looking at Figure 2, it is easy to see that all the generated images from the generator of Observer-GAN contain a noticeable amount of noise, however, the Observer network can still achieve good accuracy on the unseen test set. Because of the noisy output of any fixed generator network, a classifier trained with the positive dataset (e.g. cat images) and the output of a fixed generator network (e.g. fake dog images) as the input, learns the noise in the fake samples as it provides more evident differences between the two input classes. This holds for the output of any fixed generator in both Observer-GAN and D-GAN. However, in Observer-GAN, the learning is not done on a fixed generator network, which leads the Observer network to only learn the features of the images that the generators are consistently generating rather than the remaining, consistently-changing parts of the output images.

Fig. 2. Randomly chosen (no cherry picking) images generated by the generator of Observer-GAN and D-GAN after training for 1000 epochs. The bottom row contains example images from the training unlabeled set.

While Table 1 shows that TED\textsuperscript{a} has comparable (although slightly lower) performance to the Observer network for all of the datasets, the relatively high standard deviation of the accuracy makes it less reliable when choosing a final model, as there is no validation set to rely on.

7. CONCLUSION

In this paper, we present Observer-GAN, a method for learning from positive and unlabeled datasets. It employs a GAN architecture with two discriminators. One of the discriminators (the Observer) contributes to generator learning by deviating its output from the positive class distribution, and it is also used as a final stage classifier between positive and negative testing samples.

We demonstrate that the Observer network can learn to distinguish between the positive and negative distributions even before having to generate realistic-looking images. This is because the loss function of the generator network drives the generated images to contain features that can be used by the two discriminators. These features resemble the differences between positive and negative images, and are generated before the images look realistic to the human eye.

We also address the issue of overfitting that most existing PU learning methods suffer from because of the absence of a labeled validation set, and we assess the performance of our method averaged over multiple models to validate the insensitivity of our approach to the training stopping point.
8. REFERENCES

[1] Jessa Bekker and Jesse Davis, “Learning from positive and unlabeled data: A survey,” Machine Learning, vol. 109, no. 4, pp. 719–760, 2020.

[2] Yao Zhou, Jianpeng Xu, Jun Wu, Zeinab Taghavi, Evren Korpeoglu, Kannan Achan, and Jingrui He, “Pure: Positive-unlabeled recommendation with generative adversarial network,” in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2409–2419.

[3] Marc Claesen, Frank De Smet, Pieter Gillard, Chantal Mathieu, and Bart De Moor, “Building classifiers to predict the start of glucose-lowering pharmacotherapy using belgian health expenditure data,” arXiv preprint arXiv:1504.07389, 2015.

[4] Fantine Mordelet and Jean-Philippe Vert, “Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples,” BMC bioinformatics, vol. 12, no. 1, pp. 1–15, 2011.

[5] Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit Dhillon, “Pu learning for matrix completion,” in International conference on machine learning. PMLR, 2015, pp. 2445–2453.

[6] Ke Zhou, Gui-Rong Xue, Qiang Yang, and Yong Yu, “Learning with positive and unlabeled examples using topic-sensitive pls,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 1, pp. 46–58, 2009.

[7] Zhiang Wu, Jie Cao, Yaqiong Wang, Youquan Wang, Lu Zhang, and Junjie Wu, “hpsd: a hybrid pu-learning-based spammer detection model for product reviews,” IEEE transactions on cybernetics, vol. 50, no. 4, pp. 1595–1606, 2018.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” Advances in neural information processing systems, vol. 27, 2014.

[9] Francesco De Comité, François Denis, Rémi Gilleron, and Fabien Letouzey, “Positive and unlabeled examples help learning,” in International conference on algorithmic learning theory. Springer, 1999, pp. 219–230.

[10] Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S Yu, “Building text classifiers using positive and unlabeled examples,” in Third IEEE international conference on data mining, IEEE, 2003, pp. 179–186.

[11] Hwanjo Yu, Jiawei Han, and KC-C Chang, “Pebi: Web page classification without negative examples,” IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 1, pp. 70–81, 2004.

[12] Dell Zhang and Wee Sun Lee, “A simple probabilistic approach to learning from positive and unlabeled examples,” in Proceedings of the 5th annual UK workshop on computational intelligence (UKCI), 2005, pp. 83–87.

[13] Charles Elkan and Keith Noto, “Learning classifiers from only positive and unlabeled data,” in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 213–220.

[14] Florent Chiaroni, Ghazaleh Khodabandelou, Mohamed-Cherif Rahal, Nicolas Hueber, and Frederic Dufaux, “Counter-examples generation from a positive unlabeled image dataset,” Pattern Recognition, vol. 107, pp. 107527, 2020.

[15] Saurabh Garg, Yifan Wu, Alex Smola, Sivaraman Balakrishnan, and Zachary Lipton, “Mixture proportion estimation and PU learning: A modern approach,” in Advances in Neural Information Processing Systems (NeurIPS), 2021.

[16] Yunrui Zhao, Qianqian Xu, Yangbangyang Jiang, Peisong Wen, and Qingming Huang, “Dist-pu: Positive-unlabeled learning from a label distribution perspective,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14461–14470.

[17] Fantine Mordelet and J-P Vert, “A bagging svm to learn from positive and unlabeled examples,” Pattern Recognition Letters, vol. 37, pp. 201–209, 2014.

[18] Ryuichi Kiro, Gang Niu, Matthinus C Du Plessis, and Masashi Sugiyama, “Positive-unlabeled learning with non-negative risk estimator,” Advances in neural information processing systems, vol. 30, 2017.

[19] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama, “Convex formulation for learning from positive and unlabeled data,” in Proceedings of the 32nd International Conference on Machine Learning, Francis Bach and David Blei, Eds., Lille, France, 07–09 Jul 2015, vol. 37 of Proceedings of Machine Learning Research, pp. 1386–1394, PMLR.

[20] Dmitry Ivanov, “Dedpul: Difference-of-estimated-densities-based positive-unlabeled learning,” in 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2020, pp. 782–790.

[21] Olesya Grinenko, Jian Li, John C Mosher, Irene Z Wang, Juan C Bucio, Jorge Gonzalez-Martinez, Dileep Nair, Imad Najm, Richard M Lealy, and Patrick Chauvel, “A fingerprint of the epileptogenic zone in human epilepsies,” Brain, vol. 141, no. 1, pp. 117–131, 2018.

[22] Shuhan Wang, Tianyang Gai, Tong Qu, Bojie Ma, Xiaojing Su, Lisong Dong, Libin Zhang, Peng Xu, Yajuan Su, and Yaye Wei, “Data augmentation in hotspot detection based on generative adversarial network,” Journal of Micro/Nanopatterning, Materials, and Metrology, vol. 20, no. 3, pp. 034201, 2021.

[23] Li Deng, “The mnist database of handwritten digit images for machine learning research,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141–142, 2012.

[24] Han Xiao, Kashif Rasul, and Roland Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” arXiv preprint arXiv:1708.07747, 2017.

[25] Alex Krizhevsky, Geoffrey Hinton, et al., “Learning multiple layers of features from tiny images,” 2009.

[26] Yunjey Choi, Youngjung Uh, Jaeeun Yoo, and Jung-Woo Ha, “Stargan v2: Diverse image synthesis for multiple domains,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 8188–8197.

[27] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, “Gans trained by a two-time-scale update rule converge to a local nash equilibrium,” Advances in neural information processing systems, vol. 30, 2017.