CLASSIFICATION FROM POSITIVE, UNLABELED AND BIASED NEGATIVE DATA*/ REPRODUCIBILITY CHALLENGE

Shiquan Zhang
School of Computer Science
Mcgill University
Montréal, Québec, Canada
shiquan.zhang@mail.mcgill.ca

Ce Zhang
School of Civil Engineering
Mcgill University
Montréal, Québec, Canada
ce.zhang@mail.mcgill.ca

Xiao Deng
School of Electrical and Computer Engineering
Mcgill University
Montréal, Québec, Canada
xiao.deng2@mail.mcgill.ca

ABSTRACT

Binary classification is an essential issue in machine learning. When we have a full knowledge of both the positive (P) set and the Negative (N) set, it is quite easy for us to design an algorithm to minimize the classification risk in many cases. However, the full knowledge of the whole negative set is so difficult to obtain in reality that the researchers introduce an effective learning strategy named positive-unlabeled (PU) learning. Selecting a smaller class of negative class is not only easy to implement but also can represent the class properly if we give the data with suitable biased labels. Based on the idea of such biased (bN) data in PU learning, we followed the steps of the empirical risk minimization algorithm proposed by Hsieh et al. (2019) using the MNIST data set. Simulation results have proved that the improvement of the performances in terms of the mean and standard deviation of misclassification rate is achievable and reproducible.

1 INTRODUCTION AND MOTIVATION

In a traditional binary classification scenario, the input instances are labeled as either positive (P) or negative (N) , then we can train a classifier on these labeled instances. Conversely, in positive-unlabeled (PU) learning, we addresses the problem of learning a classifier from P and unlabeled (U) data, without the need to know exactly about the N data set Elkan & Noto (2008) and Chapelle et al. (2010).

Instead of obtaining the full knowledge of the N set, PU learning raises much more attentions in many real-world problems. However, the most difficult step in PU learning is to collect a fully representative N set, leading to a trade-off that picking up a small portion of all possible N data is relatively easy while selecting too many N data leads to huge complexity Plessis et al. (2014). Therefore, we focus on this paper to study the learning problem from P, U and biased N (bN) data, which is named PUbN learning by its authors. In addition to P and U data, the authors also gather a set of bN samples, which is governed by a distribution distinct from the true N distribution and this can be viewed as an extension of PU learning.

In fact, they are some other researchers interested in the learning problem from bN data. Fei & Liu (2015) attempted to solve similar problems in the context of text classification. They considered even gathering unbiased U data is difficult and learned the classifier from only P and bN data. However, their method is specific to text classification because it relies on the use of effective similarity

*Hsieh et al. (2019)

measures to evaluate similarity between documents. Tao et al. (2007) split all the negative samples into a small number of subsets, each of which has a simple prior distribution. For each such small group, the authors build a marginal convex machine subclassifier to distinguish data from the subset of negative data to the single positive data. However, their work only solves the relevance feedback problem in content-based image retrieval.

In this paper, the authors first develop an empirical risk minimization-based algorithm that combines both PU learning and importance weighting to solve the $PUbN$ classification problem. They then estimate the probability that an example is sampled into the P or the bN set. Based on this estimate, they regard bN and U data as N examples with instance-dependent weights and assign larger weights to U examples. P data are treated as P examples with unity weight but also as N examples with usually small or zero weight whose actual value depends on the same estimate. Next, we theoretically establish an estimation error bound for their proposed method. Finally, they perform advantages of the algorithm and state that it can be easily adapted to ordinary PU learning by ignoring all the terms related to the biased negative data.

Our reproducibility challenge has done the following part of their work:

1.We test their $PUbN$ algorithm through Python given the same data set, i.e., the MNIST, and prove that their results are reproducible.

2.We analyze our simulation results of the test error, the false negative rate and the false positive rate according to their tendencies, values and convergence.

3.Based on the differences of our simulation results with the author's experiments, we try to explain the reasons.

2 METHOD

2.1 PN CLASSIFICATION

Positive-negative(PN) classification is the standard supervised binary classification problem. The aim of the PN classification is to find an arbitrary decision function for binary classification $g : \mathbb{R}^d \to \mathbb{R}$, such that minimizes the classification risk:

$$R(g) = \mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}[l(yg(\boldsymbol{x})] \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$, $y \in \{+1, -1\}$ is random variables following unknown probability distribution with density $p(\boldsymbol{x}, y)$, $l : \mathbb{R} \to \mathbb{R}_+$ is a loss function of margin $yg(\boldsymbol{x})$ and $\mathbb{E}_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y)}[\cdot]$ is the expectation over the joint distribution $p(\boldsymbol{x}, y)$. Loss function is usually represented by the sigmoid loss $l_{sig}(z) = 1/(1 + \exp(z))$.

In the PN classification setting, we can sample the data with label P and N independently from $p(\boldsymbol{x}|y = +1)$ and $p(\boldsymbol{x}|y = -1)$. The P and N sampled data are denoted by $\mathcal{X}_P = \{\boldsymbol{x}_i^P\}_{i=1}^{n_P}$ and $\mathcal{X}_N = \{\boldsymbol{x}_i^N\}_{i=1}^{n_N}$. The prior of P data is denoted by $\pi = p(y = 1)$ and the partial risks are denoted by $R_P^+(g) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=+1)}[l(g(\boldsymbol{x}))]$, $R_N^-(g) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=-1)}[l(-g(\boldsymbol{x}))]$. Thus the empirical risk of PN classification can be written as:

$$\hat{R}_{PN}(g) = \pi\hat{R}_P^+(g) + (1-\pi)\hat{R}_N^-(g) = \frac{\pi}{n_P}\sum_{i=1}^{n_P} l(g(\boldsymbol{x}_i^P)) + \frac{1-\pi}{n_N}\sum_{i=1}^{n_N} l(-g(\boldsymbol{x}_i^N)) \tag{2}$$

We can obtain the standard PN classifier, or empirical risk minimizer $\hat{g}_{PN}$ by minimizing empirical risk $\hat{R}_{PN}(g)$.

2.2 PU CLASSIFICATION

In some cases, we sample data from the whole data set and only are able to label the positive data. In other words, we can only sample the P data $\mathcal{X}_P$ and the unlabeled data $\mathcal{X}_U = \{\boldsymbol{x}_i^U\}_{i=1}^{n_U} \sim p(\boldsymbol{x})$. In the PU classification setting, du Plessis et al. (2014) and Plessis et al. (2015) proposed an unbiased risk estimator and the unbiased risk of N data can be represented by:

$$(1-\pi)R_N^-(g) = R_U^-(g) - \pi R_P^-(g) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[l(-g(\boldsymbol{x}))] - \pi\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=+1)}[l(-g(\boldsymbol{x}))] \tag{3}$$

Therefore, the empirical risk of PU classification can be written by:

$$\begin{aligned}\hat{R}_{PU}(g) &= \pi\hat{R}_P^+(g) + (1-\pi)\hat{R}_N^-(g) = \pi\hat{R}_P^+(g) + \hat{R}_U^-(g) - \pi\hat{R}_P^-(g) \\ &= \frac{\pi}{n_P}\sum_{i=1}^{n_P} l(g(\boldsymbol{x}_i^P)) - \frac{\pi}{n_P}\sum_{i=1}^{n_P} l(-g(\boldsymbol{x}_i^P)) + \frac{1}{n_U}\sum_{i=1}^{n_U} l(-g(\boldsymbol{x}_i^U))\end{aligned} \tag{4}$$

Similarly, we can obtain the PU classifier $\hat{g}_{PU}$ by minimizing the empirical risk $\hat{R}_{PU}(g)$.

However, the model g_{PU} overfits the training data such that the empirical risk $\hat{R}_{PU}(g)$ goes to negative. To avoid overfitting, Kiryo et al. (2017) proposed the non-negative PU classification ($nnPU$). The $nnPU$ classification requires that $R_U^-(g) - \pi R_P^-(g) = (1-\pi)R_N^-(g) \geq 0$. The empirical risk estimator can be modified as:

$$\tilde{R}_{PU}(g) = \pi\hat{R}_P^+(g) + \max\{0, \hat{R}_U^-(g) - \pi\hat{R}_P^-(g)\} \tag{5}$$

2.3 PNU Classification

In another case, when the P, N and U data are all available, which we called as the PNU classification, and it becomes a semi-supervised problem. Sakai et al. (2016) proposed a PNU risk estimator, which linearly combines the PN and PU/NU risk estimator. With another parameter $\gamma \in [0, 1]$, the PNU risk estimator is expressed as follows:

$$\begin{aligned}\hat{R}_{PNU}^{\gamma}(g) &= \gamma\hat{R}_{PN}(g) + (1-\gamma)\hat{R}_{PU}(g) \\ &= \gamma[\pi\hat{R}_P^+(g) + (1-\pi)\hat{R}_N^-(g)] + (1-\gamma)(\pi\hat{R}_P^+(g) - \pi\hat{R}_P^-(g) + \hat{R}_U^-(g)) \\ &= \pi\hat{R}_P^+(g) + \gamma(1-\pi)\hat{R}_N^-(g) + (1-\gamma)(\hat{R}_U^-(g) - \pi\hat{R}_P^-(g)) \\ &= \frac{\pi}{n_P}\sum_{i=1}^{n_P} l(g(\boldsymbol{x}_i^P)) - \frac{(1-\gamma)\pi}{n_P}\sum_{i=1}^{n_P} l(-g(\boldsymbol{x}_i^P)) + \frac{1-\gamma}{n_U}\sum_{i=1}^{n_U} l(-g(\boldsymbol{x}_i^U)) \\ &+ \frac{\gamma(1-\pi)}{n_N}\sum_{i=1}^{n_N} l(-g(\boldsymbol{x}_i^N))\end{aligned} \tag{6}$$

Similarly, to alleviate overfitting, we can also modified the estimator to a non-negative PNU classifier ($nnPNU$):

$$\hat{R}_{PNU}^{\gamma}(g) = \pi\hat{R}_P^+(g) + \max\{0, \gamma(1-\pi)\hat{R}_N^-(g) + (1-\gamma)(\hat{R}_U^-(g) - \pi\hat{R}_P^-(g))\} \tag{7}$$

2.4 $PUbN$ Classification

In some cases, the situation sits between semi-supervised PNU problem and pure PU classification. That is, we can sample the unlabelled data $\mathcal{X}_U \sim p(\boldsymbol{x})$ and the positive data $\mathcal{X}_P \sim p(\boldsymbol{x}|y=+1)$. Besides, parts of the negative data are also available. The problem under this setting is named positive-unlabelled-biased-negative ($PUbN$) classification. Since the biased negative data cannot represent the whole negative data set, we need to introduce another latent random variable $s \in \{+1, -1\}$. $s = +1$ means the data is available to sample, and $s = -1$ means the data cannot be sampled. Thus the original distribution is modified to a joint distribution $p(\boldsymbol{x}, y, s)$ with constraints $p(s=+1|\boldsymbol{x}, y=+1) = p(y=-1|\boldsymbol{x}, s=-1) = 1$. In this joint distribution, the prior $\pi = p(\boldsymbol{x}|y=+1) = p(\boldsymbol{x}|y=+1, s=+1)$ represents the distribution of the positive data and $(1-\pi)$ represents the distribution of the negative data. And we introduce another prior $\rho = p(\boldsymbol{x}|y=-1, s=+1)$, which represents the distribution of the biased negative data. The relation among positive data, biased negative data and the rest of negative data is shown in Fig. 1.

Thus, the bN samples are denoted by:

$$\mathcal{X}_{bN} = \{\boldsymbol{x}_i^{bN}\}_{i=1}^{n_{bN}} \sim p(\boldsymbol{x}|y=-1, s=+1) \tag{8}$$

And the classification risk is:

$$R_{PUbN}(g) = \pi R_P^+(g) + \rho R_{bN}^-(g) + (1-\pi-\rho)R_{N\setminus bN}^-(g) \tag{9}$$

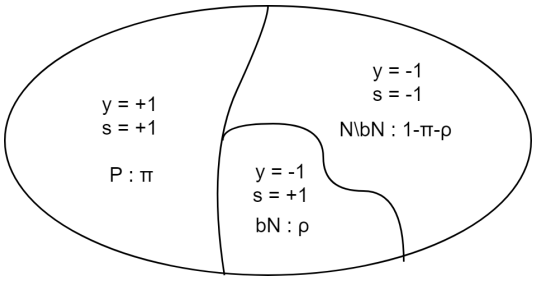


Figure 1: the $PUbN$ diagram

The empirical risk can be expressed as:

$$\begin{aligned}\hat{R}_{PUbN}(g) &= \pi\hat{R}_P^+(g) + \rho\hat{R}_{bN}^-(g) + (1-\pi-\rho)\hat{R}_{N\setminus bN}^-(g) \\ &= \frac{\pi}{n_P}\sum_{i=1}^{n_P} l(g(\boldsymbol{x}_i^P)) + \frac{\rho}{n_{bN}}\sum_{i=1}^{n_{bN}} l(-g(\boldsymbol{x}_i^{bN})) + (1-\pi-\rho)\hat{R}_{s=-1}^-(g)\end{aligned} \tag{10}$$

The first two terms of the empirical risk can be estimated by the positive and biased negative samples. Now we focus on the third term. Let $\bar{R}_{s=-1}^-(g) = (1-\pi-\rho)R_{s=-1}^-(g)$, $\sigma(\boldsymbol{x}) = p(s=+1|\boldsymbol{x})$, $\eta \in [0,1]$ and $h : \mathbb{R}^d \to [0,1]$ s.t. $h(\boldsymbol{x}) > \eta \Rightarrow \sigma(\boldsymbol{x}) > 0$. Then the risk is:

$$\begin{aligned}\bar{R}_{s=-1}^-(g) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\mathbf{1}_{h(\boldsymbol{x})\leq\eta}\, l(-g(\boldsymbol{x}))(1-\sigma(\boldsymbol{x}))] \\ + \pi\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=+1)}\Big[\mathbf{1}_{h(\boldsymbol{x})>\eta}\, l(-g(\boldsymbol{x}))\frac{1-\sigma(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\Big] \\ + \rho\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y=-1,s=+1)}\Big[\mathbf{1}_{h(\boldsymbol{x})>\eta}\, l(-g(\boldsymbol{x}))\frac{1-\sigma(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\Big]\end{aligned} \tag{11}$$

Practically, we use $\hat{\sigma}(\boldsymbol{x})$ to estimate $\sigma(\boldsymbol{x})$ and $h(\boldsymbol{x})$. And $\hat{\sigma}(\boldsymbol{x})$ can be obtained by training the classifier of random variables s. We apply $nnPU$ classifier to samples $\mathcal{X}_P$, $\mathcal{X}_{bN}$, which represent the positive sample of s and $\mathcal{X}_U$, which represent the unlabelled sample of s. Therefore, we can train an $nnPU$ classifier of s, $\hat{\sigma}(\boldsymbol{x})$, by the P, bN and U samples to approximate $\sigma(\boldsymbol{x})$ and $h(\boldsymbol{x})$ in the risk estimator. The empirical risk of $\bar{R}_{s=-1}(g)$ can be approximated from the samples by:

$$\begin{aligned}\hat{\bar{R}}_{s=-1,\eta,\hat{\sigma}(\boldsymbol{x})}(g) = \frac{1}{n_U}\sum_{i=1}^{n_U}\Big[\mathbf{1}_{\hat{\sigma}(\boldsymbol{x}_i^U)\leq\eta}\, l(-g(\boldsymbol{x}_i^U))(1-\hat{\sigma}(\boldsymbol{x}_i^U))\Big] \\ + \frac{\pi}{n_P}\sum_{i=1}^{n_P}\Big[\mathbf{1}_{\hat{\sigma}(\boldsymbol{x}_i^P)>\eta}\, l(-g(\boldsymbol{x}_i^P))\frac{1-\hat{\sigma}(\boldsymbol{x}_i^P)}{\hat{\sigma}(\boldsymbol{x}_i^P)}\Big] \\ + \frac{\rho}{n_{bN}}\sum_{i=1}^{n_{bN}}\Big[\mathbf{1}_{\hat{\sigma}(\boldsymbol{x}_i^{bN})>\eta}\, l(-g(\boldsymbol{x}_i^{bN}))\frac{1-\hat{\sigma}(\boldsymbol{x}_i^{bN})}{\hat{\sigma}(\boldsymbol{x}_i^{bN})}\Big]\end{aligned} \tag{12}$$

Finally, we can get the complete expression of the empirical risk as follows:

$$\hat{R}_{PUbN,\eta,\hat{\sigma}}(g) = \frac{\pi}{n_P}\sum_{i=1}^{n_P} l(g(\boldsymbol{x}_i^P)) + \frac{\rho}{n_{bN}}\sum_{i=1}^{n_{bN}} l(-g(\boldsymbol{x}_i^{bN})) + \hat{\bar{R}}_{s=-1,\eta,\hat{\sigma}}(g) \tag{13}$$

2.5 REVISIT PU CLASSIFICATION

If the biased negative data are unavailable, we go back to the PU classification problem. The $PUbN$ risk estimator discussed above degrades to $PUbN\backslash N$ estimator:

$$\begin{aligned}\hat{R}_{PUbN\backslash N,\eta,\hat{\sigma}}(g) &= \pi\hat{R}_P^+(g) + \hat{\tilde{R}}^-_{y=-1,\eta,\hat{\sigma}}(g) \\ &= \frac{\pi}{n_P}\sum_{i=1}^{n_P} l(g(\boldsymbol{x}_i^P)) + \frac{1}{n_U}\sum_{i=1}^{n_U}\Big[\mathbf{1}_{\hat{\sigma}(\boldsymbol{x}_i^U)\leq\eta}\, l(-g(\boldsymbol{x}_i^U))(1-\hat{\sigma}(\boldsymbol{x}_i^U))\Big] \\ &+ \frac{\pi}{n_P}\sum_{i=1}^{n_P}\Big[\mathbf{1}_{\hat{\sigma}(\boldsymbol{x}_i^P)>\eta}\, l(-g(\boldsymbol{x}_i^P))\frac{1-\hat{\sigma}(\boldsymbol{x}_i^P)}{\hat{\sigma}(\boldsymbol{x}_i^P)}\Big]\end{aligned} \tag{14}$$

where $\hat{\sigma}(\boldsymbol{x})$ also degrades to an estimate of $p(y=+1|\boldsymbol{x})$.

3 EXPERIMENT PROCEDURES

3.1 DATA SET INTRODUCTION

3.1.1 MNIST SET

The MNIST database is a data set of handwritten digits, containing 60,000 training images and 10,000 testing imagesKussul & Baidyk (2004). Half of the training set and half of the test set were taken from NIST's training data set, while the other half of the training set and the other half of the test set were taken from NIST's testing data set. The MNIST database was constructed from NIST's Special Database 3 and Special Database 1 which contain binary images of handwritten digits. NIST originally designated SD-3 as their training set and SD-1 as their test set. However, SD-3 is much cleaner and easier to recognize than SD-1. The reason for this can be found on the fact that SD-3 was collected among Census Bureau employees, while SD-1 was collected among high-school students.

3.1.2 DATA SET SETTING

For every a same learning task, different methods are compared using the same 10 random samplings from the data set. Biased N data uniformly to the latent categories: [1,3,5,7,9] and the probability is [0.03, 0.15, 0.3, 0.02, 0.5].

3.2 BASELINE

For indicating the performance of $PUBN$, we use the following two relative baselines:

3.2.1 $nnPNU$/$nnPU$

When biased negative database was given, the first baseline is $nnPNU$ as the semi-supervised learning:

$$\hat{R}^{\gamma}_{PNU}(g) = \gamma\hat{R}_{PN}(g)+(1-\gamma)\hat{R}_{PU}(g) = \pi\hat{R}_P^+(g)+\gamma(1-\pi)\hat{R}_N^-(g)+(1-\gamma)(\hat{R}_U^-(g)-\pi\hat{R}_P^-(g)) \tag{15}$$

If the biased negative was not given, the first baseline becomes $nnPU$:

$$\hat{R}_{PU}(g) = \pi\hat{R}_P^+(g) - \pi\hat{R}_P^-(g) + \hat{R}_U^-(g) \tag{16}$$

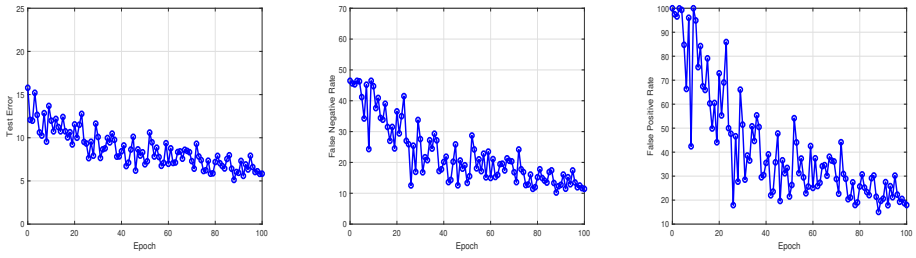
3.2.2 $PU\rightarrow PN$

This baseline only existed when biased negative database was given. In this method, we need to train two binary classifiers, first is to classify positive and biased negative dataset $(s=1, y=\pm 1)$ from the whole database and then classify the positive data $(y=1)$ from the dataset we classified before.

3.3 PREPROCESSING

In this experiment, $\hat{\sigma}$ and g always use the same model and are trained for the same number of epochs. Meanwhile, to determine the value of η, we also introduced another hyperparameter τ and choose η such that $\#\{x \in \mathcal{X}_U | \hat{\sigma}(x) \leq \eta\} = \tau(1 - \pi - \rho)n_U$.

4 SIMULATION RESULTS AND ANALYSIS



(a) Test Error (b) False Negative Rate (c) False Positive Rate

Figure 2:The three metrics along with the epoches

We run the $PUbN\backslash N$ algorithm proposed by the authors in terms of the test error, the false negative rate and the false positive rate using the MNIST data set, respectively, which has been shown as Fig.2(a) through Fig.2(c). As we can see from these three figures, all the three metrics decrease while the epoches increase. Compared our results with the author's, we find that the final values of these metrics are approximately the same and so do the tendencies of these three curves. However, there are still some differences. The first one is the larger oscillation phenomenon in our simulation, which mainly because of the less number of the simulation times. The next difference is the convergence rate. In the author's statement, the epoch which reaches sharp decrease of the metrics occurs less than 20 in their simulations, however, our simulation claims that the descending rate is almost the same in different periods, but can converge eventually. The main reason of that is because of the setting of the hyperparameter η, which is an indicator of how lowest the empirical risk of the estimator can reach finally. In the additional experiments part of the paper, the authors discuss the influence of the hyperparameter of η and ρ. However, the optimal setting of these two hyperparameters depends mainly on the intermediate variable τ, but the different values of the set τ can lead to a diverge convergence rate, but they fail to explain the reasons. So, the different setting of that hyperparameter may result in the difference of the convergence rate. In general, the algorithm can converge and indeed perform better than uPU and $nnPU$.

5 CONCLUSION

In our reproducibility challenge, we investigate the $PUbN\backslash N$ learning problem in binary classification area and use the MNIST data set to reproduce the authors' results. The tendencies and the values of the test error, the false negative rate and the false positive rate are almost the same with the author's. However, the oscillation phenomenon and the convergence rate are different, resulting from the simulation times and the setting of the hyperparameter, respectively. In sum, the author's algorithm can perform better than the traditional algorithms and can converge eventually when using the MNIST data set.

REFERENCES

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. *Semi-Supervised Learning*, volume 1. MIT Press, 2010. URL `http://www.acad.bg/ebook/ml/MITPress-%20SemiSupervised%20Learning.pdf`.

Marthinus C du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence,

and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 703–711. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5509-analysis-of-learning-from-positive-and-unlabeled-data.pdf.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. *KDD*, 2008. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.9201&rep=rep1&type=pdf.

Geli Fei and Bing Liu. Social media text classification under negative covariate shift. In *Proceedings of the 2015 Conference on Empirical Mehods in Natural Language Processing*, pp. 2347–2356, 2015. URL https://pdfs.semanticscholar.org/7307/df9edffcd23c692b3367565b680bc2682490.pdf.

Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. Classification from positive, unlabeled and biased negative data. In *Submitted to International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1ldNoC9tX. under review.

Ryuichi Kiryo, Gang Niu, Marthinus Christoffel du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. *CoRR*, abs/1703.00593, 2017. URL http://arxiv.org/abs/1703.00593.

Ernst M. Kussul and Tatyana Baidyk. Improved method of handwritten digit recognition tested on MNIST database. *Image Vision Comput.*, 22(12):971–981, 2004. doi: 10.1016/j.imavis.2004.03.008. URL https://doi.org/10.1016/j.imavis.2004.03.008.

Marthinus C. du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 703–711, 2014.

Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1386–1394, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/plessis15.html.

Tomoya Sakai, Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Beyond the low-density separation principle: A novel approach to semi-supervised learning. *CoRR*, abs/1605.06955, 2016. URL http://arxiv.org/abs/1605.06955.

Dacheng Tao, Xuelong Li, and Stephen J. Maybank. Negative samples analysis in relevance feedback. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):568–580, 2007. URL https://ieeexplore-ieee-org.proxy3.library.mcgill.ca/stamp/stamp.jsp?tp=&arnumber=4118712.