ON THE MINIMAL SUPERVISION FOR TRAINING ANY BINARY CLASSIFIER FROM ONLY UNLABELED DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

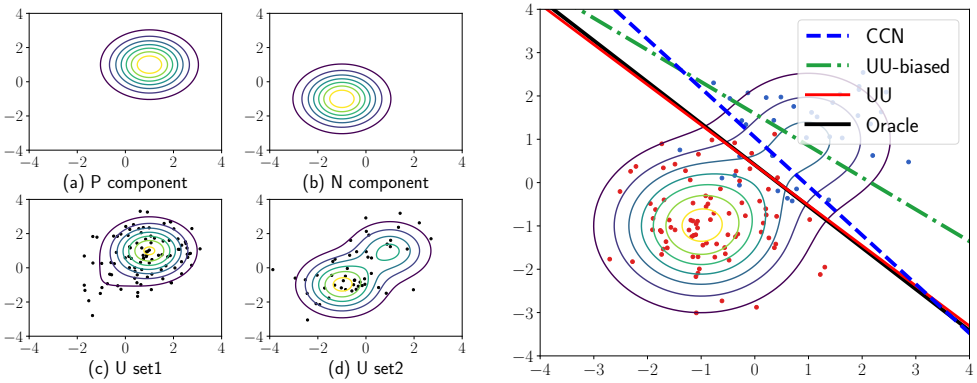
Empirical risk minimization (ERM), with proper loss function and regularization, is the common practice of supervised classification. In this paper, we study training *arbitrary* (from linear to deep) *binary classifier* from *only unlabeled* (U) *data* by ERM. We prove that it is impossible to estimate the risk of an arbitrary binary classifier in an *unbiased* manner given a single set of U data, but it becomes possible given two sets of U data with *different class priors*. These two facts answer a fundamental question—what the minimal supervision is for training any binary classifier from only U data. Following these findings, we propose an ERM-based learning method from two sets of U data, and then prove it is *consistent*. Experiments demonstrate the proposed method could train deep models and outperform state-of-the-art methods for learning from two sets of U data.

1 INTRODUCTION

With some properly chosen loss function (e.g., Bartlett et al., 2006; Tewari & Bartlett, 2007; Reid & Williamson, 2010) and regularization (e.g., Tikhonov, 1943; Srivastava et al., 2014), *empirical risk minimization* (ERM) is the common practice of supervised classification (Vapnik, 1998). Actually, ERM is used in not only supervised learning but also *weakly-supervised learning*. For example, in *semi-supervised learning* (Chapelle et al., 2006), we have very limited labeled (L) data and a lot of unlabeled (U) data, where L data share the same form with supervised learning. Thus, it is easy to estimate the risk from L data, and ERM can be done with regularization based on U data (including but not limited to Grandvalet & Bengio, 2004; Belkin et al., 2006; Mann & McCallum, 2007; Niu et al., 2013; Miyato et al., 2016; Laine & Aila, 2017; Tarvainen & Valpola, 2017; Luo et al., 2018).

Nevertheless, L data may differ from supervised learning in not only the amount but also the form. For instance, in *positive-unlabeled learning* (Elkan & Noto, 2008; Ward et al., 2009), all L data are from the positive class, and due to the lack of L data from the negative class it becomes impossible to estimate the risk from only L data. To this end, a two-step approach to ERM has been considered (du Plessis et al., 2014; 2015; Niu et al., 2016; Kiryo et al., 2017). Firstly, the risk is rewritten into an equivalent expression, such that it just involves the same distributions from which L and U data are sampled—this step leads to certain risk estimators. Secondly, the risk is estimated from both L and U data, and the resulted empirical training risk is minimized (e.g. by Robbins & Monro, 1951; Kingma & Ba, 2015). In this two-step approach, U data are used for *risk evaluation* that is the core of ERM and is mandatory in ERM, and hence *risk rewrite* enables ERM and is the key of success.

One step further from positive-unlabeled learning is *learning from only U data* without any L data. This is significantly harder than previous learning problems (cf. Figure 1). However, we would still like to train *arbitrary binary classifier*, in particular, deep networks (Goodfellow et al., 2016). As a consequence, we prefer ERM to clustering methods (e.g., Xu et al., 2004; Gomes et al., 2010). The critical point is how to estimate the risk from only U data, and our solution is again ERM-enabling risk rewrite in the aforementioned two-step approach. The first step should lead to an *unbiased risk estimator* that will be used in the second step. Subsequently, we can evaluate the empirical training and/or validation risk by plugging only U training/validation data into the risk estimator. Thus, this two-step ERM needs *no L validation data for hyperparameter tuning*, which is a huge advantage in training deep models nowadays. Note that given only U data, by no means could we learn the class priors (Menon et al., 2015), so that we assume *all necessary class priors are also given*. This is the



In the left panel, (a) and (b) show positive (P) and negative (N) components of the Gaussian mixture; (c) and (d) show two distributions (with class priors 0.9 and 0.4) where U training data are drawn (marked as black points). The right panel shows the test distribution (with class prior 0.3) and data (marked as blue for P and red for N), as well as four learned classifiers. In the legend, "CCN" refers to Natarajan et al. (2013), "UU-biased" means supervised learning taking larger-/smaller-class-prior U data as P/N data, "UU" is the proposed method, and "Oracle" means supervised learning from the same amount of L data. See Appendix B for more information. We can see that the UU classifier is almost identical to the Oracle classifier and much better than the other two.

Figure 1: Illustrative example of classification from a Gaussian mixture dataset.

unique type of supervision we will leverage throughout this paper, which implies that the problem of interest still belongs to weakly-supervised learning rather than unsupervised learning.

In this paper, we raise a fundamental question in weakly-supervised learning—how many sets of U data with *different class priors* are necessary for rewriting the risk? Our answer has two aspects:

- Risk rewrite is impossible given a single set of U data (see Theorem 2 in Sec. 3);
- Risk rewrite becomes possible given two sets of U data (see Theorem 4 in Sec. 4).

This suggests that *three class priors*[1] *are all you need* to train deep models from only U data, while any two[2] should not be enough. The impossibility is a proof by contradiction, and the possibility is a proof by construction, following which we explicitly design an unbiased risk estimator. Therefore, with the help of this risk estimator, we propose an ERM-based learning method from two sets of U data. Thanks to the unbiasedness of our risk estimator, we derive an *estimation error bound* which certainly guarantees the *consistency of learning* (Mohri et al., 2012; Shalev-Shwartz & Ben-David, 2014).[3] Experiments demonstrate that the proposed method could train multilayer perceptron, All-ConvNet (Springenberg et al., 2015) and ResNet (He et al., 2016) from two sets of U data; it could outperform state-of-the-art methods for learning from two sets of U data. See Figure 1 for how the proposed method works on a Gaussian mixture of two components.

2 Problem Setting and Related Work

Consider the binary classification problem. Let X and Y be the input and output random variables such that $p(x, y)$ is the *underlying joint density*, $p_{\mathrm{p}}(x) = p(x \mid Y = +1)$ and $p_{\mathrm{n}}(x) = p(x \mid Y = -1)$ are the P and N *class-conditional densities*, $p(x)$ is the *marginal density*, and $\pi_{\mathrm{p}} = p(Y = +1)$ is the *class-prior probability*.

Data generation process Let θ and θ' be two class priors such that $\theta \neq \theta'$, and let

$$p_{\mathrm{tr}}(x) = \theta p_{\mathrm{p}}(x) + (1-\theta) p_{\mathrm{n}}(x), \quad p'_{\mathrm{tr}}(x) = \theta' p_{\mathrm{p}}(x) + (1-\theta') p_{\mathrm{n}}(x) \tag{1}$$

be the marginal densities from which U training data are drawn. Eq. (1) implies there are $p_{\mathrm{tr}}(x, y)$ and $p'_{\mathrm{tr}}(x, y)$, whose class-conditional densities are same and equal to those of $p(x, y)$, and whose class priors are different, i.e.,

$$p_{\mathrm{tr}}(x \mid y) = p'_{\mathrm{tr}}(x \mid y) = p(x \mid y), \quad p_{\mathrm{tr}}(Y = +1) = \theta \neq \theta' = p'_{\mathrm{tr}}(Y = +1).$$

[1] Two class-prior probabilities are of the training distributions and one is of the test distribution.

[2] One of the training distribution and one of the test distribution, or two of the training distributions.

[3] Learning is consistent (more specifically the learned classifier is asymptotically consistent), if and only if as the amount of training data approaches infinity, the risk of the learned classifier converges to the risk of the optimal classifier, where the optimality is defined over a given hypothesis class.

If we could sample L data from $p_{\mathrm{tr}}(x,y)$ or $p'_{\mathrm{tr}}(x,y)$, it would reduce to supervised learning under *class-prior change* (Quiñonero-Candela et al., 2009).

Nonetheless, the problem of interest belongs to weakly-supervised learning—U training (and validation) data are supposed to be drawn according to (1). More specifically, we have

$$\mathcal{X}_{\mathrm{tr}} = \{x_1, \ldots, x_n\} \sim p_{\mathrm{tr}}(x), \quad \mathcal{X}'_{\mathrm{tr}} = \{x'_1, \ldots, x'_{n'}\} \sim p'_{\mathrm{tr}}(x), \tag{2}$$

where n and n' are two natural numbers as the sample sizes of $\mathcal{X}_{\mathrm{tr}}$ and $\mathcal{X}'_{\mathrm{tr}}$. This is exactly same as du Plessis et al. (2013) and Menon et al. (2015) with some different names. In Menon et al. (2015), θ and θ' are called *corruption parameters*, and if we assume $\theta > \theta'$, $p_{\mathrm{tr}}(x)$ is called the *corrupted P density* and $p'_{\mathrm{tr}}(x)$ is called the *corrupted N density*. Despite the same data generation process in (2), a vital difference between the problem settings is performance measures to be optimized.

Performance measures Let $g : \mathbb{R}^d \to \mathbb{R}$ be an arbitrary *decision function*, i.e., g may literally be any binary classifier. Let $\ell : \mathbb{R} \to \mathbb{R}$ be the *loss function*, such that the value $\ell(z)$ means the loss by predicting $g(x)$ when the ground truth is y where $z = yg(x)$ is the *margin*. The *risk* of g is

$$R(g) = \mathbb{E}_{(X,Y)\sim p(x,y)}[\ell(Yg(X))] = \pi_{\mathrm{p}}\mathbb{E}_{\mathrm{p}}[\ell(g(X))] + (1-\pi_{\mathrm{p}})\mathbb{E}_{\mathrm{n}}[\ell(-g(X))], \tag{3}$$

where $\mathbb{E}_{\mathrm{p}}[\cdot]$ means $\mathbb{E}_{X\sim p_{\mathrm{p}}}[\cdot]$ and $\mathbb{E}_{\mathrm{n}}[\cdot]$ means $\mathbb{E}_{X\sim p_{\mathrm{n}}}[\cdot]$ respectively. If ℓ is the *zero-one loss* that is defined by $\ell_{01}(z) = (1-\mathrm{sign}(z))/2$, the risk is named the *classification error* that is the standard performance measure in classification problems. A balanced version of (3) is

$$B(g) = \tfrac{1}{2}\mathbb{E}_{\mathrm{p}}[\ell(g(X))] + \tfrac{1}{2}\mathbb{E}_{\mathrm{n}}[\ell(-g(X))], \tag{4}$$

and if ℓ is ℓ_{01}, (4) is named the *balanced error* (Brodersen et al., 2010). The vital difference is that (3) is chosen in the current paper whereas (4) is chosen in du Plessis et al. (2013) and Menon et al. (2015) as the performance measure to be optimized.

We argue that (3) is more natural than (4) as the performance measure for binary classification. By the phrase "binary classification", we mean π_{p} is neither very large nor very small. Otherwise, due to extreme values of π_{p} (i.e., either $\pi_{\mathrm{p}} \approx 0$ or $\pi_{\mathrm{p}} \approx 1$), the problem under consideration should be retrieval or detection rather than binary classification. Note that $B(g) \neq R(g)$ unless $\pi_{\mathrm{p}} = \frac{1}{2}$ since g is arbitrary, which implies that (4) is misleading so long as (3) is the performance measure.

Related work Learning from only U data is previously regarded as *discriminative clustering* (Xu et al., 2004; Valizadegan & Jin, 2006; Li et al., 2009; Gomes et al., 2010; Sugiyama et al., 2014; Hu et al., 2017). Their goals are to maximize the margin or the mutual information between X and Y. As a result, they rely on the *cluster assumption* (Chapelle et al., 2002) and the assumption that *one class corresponds to exactly one cluster*. The second assumption is rarely satisfied in practice.

As mentioned earlier, learning from two sets of U data is already studied in du Plessis et al. (2013) and Menon et al. (2015). Both of them adopt (4) as the performance measure. In the former paper, g is learned by estimating $\mathrm{sign}(p_{\mathrm{tr}}(x) - p'_{\mathrm{tr}}(x))$. In the latter paper, g is learned by taking noisy L data from $p_{\mathrm{tr}}(x)$ and $p'_{\mathrm{tr}}(x)$ as clean L data from $p_{\mathrm{p}}(x)$ and $p_{\mathrm{n}}(x)$, and then its threshold is moved to the correct value by post-precessing. In summary, instead of ERM, they evidence the possibility of *empirical balanced risk minimization*, and no impossibility is proven.

Our findings are compatible with *learning from label proportions* (Quadrianto et al., 2009; Yu et al., 2013). In Quadrianto et al. (2009), it is proven the minimal number of U sets equals the number of classes. However, their finding only holds for the linear model, the logistic loss, and their proposed method based on mean operators. On the other hand, Yu et al. (2013) is not ERM-based; it is based on discriminative clustering together with expectation regularization (Mann & McCallum, 2007).

At first glance, our data generation process, using the names from Menon et al. (2015), looks quite similar to that of *learning with noisy labels* (cf. Natarajan et al., 2013). In fact, these two are fairly different, and the differences are reviewed and discussed in Menon et al. (2015) and van Rooyen & Williamson (2018). Along this line of research, just a few papers explore instance-dependent noise models (Menon et al., 2016; Cheng et al., 2017), and the vast majority of papers employ instance-independent noise models (e.g., Natarajan et al., 2013; Sukhbaatar et al., 2015; Menon et al., 2015; Liu & Tao, 2016; Goldberger & Ben-Reuven, 2017; Patrini et al., 2017; Han et al., 2018a) or have no noisy model but many heuristics (e.g., Reed et al., 2015; Jiang et al., 2018; Ren et al., 2018; Han et al., 2018b). There exist two instance-independent noise models: *class-conditional noise* (CCN)

in Angluin & Laird (1988) and *mutually contaminated distributions* (MCD) in Scott et al. (2013). Denote by $\tilde{y}$ and $\tilde{p}(\cdot)$ the corrupted label and distributions. Then, CCN and MCD are defined by

$$\begin{pmatrix}\tilde{p}(\tilde{Y}=+1 \mid x)\\ \tilde{p}(\tilde{Y}=-1 \mid x)\end{pmatrix} = T_{\mathrm{CCN}}\begin{pmatrix}p(Y=+1 \mid x)\\ p(Y=-1 \mid x)\end{pmatrix} \quad \text{and} \quad \begin{pmatrix}\tilde{p}(x \mid \tilde{Y}=+1)\\ \tilde{p}(x \mid \tilde{Y}=-1)\end{pmatrix} = T_{\mathrm{MCD}}\begin{pmatrix}p_{\mathrm{p}}(x)\\ p_{\mathrm{n}}(x)\end{pmatrix},$$

where both of T_{CCN} and T_{MCD} are 2-by-2 matrices but T_{CCN} is column normalized and T_{MCD} is row normalized. It has been proven in Menon et al. (2015) that CCN is a strict special case of MCD. To be clear, $\tilde{p}(\tilde{y})$ is fixed in CCN once $\tilde{p}(\tilde{y} \mid x)$ is specified while $\tilde{p}(\tilde{y})$ is free in MCD after $\tilde{p}(x \mid \tilde{y})$ is specified. Furthermore, $\tilde{p}(x) = p(x)$ in CCN but $\tilde{p}(x) \neq p(x)$ in MCD. Due to this *covariate shift*, CCN methods do not fit MCD problem setting, though MCD methods fit CCN problem setting. To the best of our knowledge, the proposed method is the first MCD method based on ERM.

3 Learning from one set of U data

From now on, we prove that knowing π_{p} and θ is insufficient for rewriting $R(g)$.

3.1 A brief review of ERM

To begin with, we review ERM (Vapnik, 1998) by imaging that we are given $\mathcal{X}_{\mathrm{p}} = \{x_1, \ldots, x_n\} \sim p_{\mathrm{p}}(x)$ and $\mathcal{X}_{\mathrm{n}} = \{x'_1, \ldots, x'_{n'}\} \sim p_{\mathrm{n}}(x)$. Then, we would go through the following procedure:

1. Choose a *surrogate loss* $\ell(z)$, so that $R(g)$ in Eq. (3) is defined.
2. Choose a model $\mathcal{G}$, so that $\min_{g\in\mathcal{G}} R(g)$ is achievable by ERM.
3. Approximate $R(g)$ by
$$\widehat{R}_{\mathrm{pn}}(g) = \tfrac{\pi_{\mathrm{p}}}{n}\textstyle\sum_{i=1}^{n} \ell(g(x_i)) + \tfrac{1-\pi_{\mathrm{p}}}{n'}\sum_{j=1}^{n'} \ell(-g(x'_j)). \tag{5}$$
4. Minimize $\widehat{R}_{\mathrm{pn}}(g)$, with appropriate regularization, by favorite optimization algorithm.

Here, ℓ should be *classification-calibrated* (Bartlett et al., 2006), in order to guarantee that $R(g;\ell)$ and $R(g;\ell_{01})$ have the same minimizer over all measurable functions. This minimizer is the *Bayes optimal classifier* and denoted by $g^{**} = \arg\min_g R(g)$. The Bayes optimal risk $R(g^{**})$ is usually unachievable by ERM as $n, n' \to \infty$. That is why by choosing a model $\mathcal{G}$, $g^* = \arg\min_{g\in\mathcal{G}} R(g)$ is changed as the target to which $\widehat{g}_{\mathrm{pn}} = \arg\min_{g\in\mathcal{G}} \widehat{R}_{\mathrm{pn}}(g)$ converges as $n, n' \to \infty$. In statistical learning, the *approximation error* is $R(g^*) - R(g^{**})$, and the *estimation error* is $R(\widehat{g}_{\mathrm{pn}}) - R(g^*)$. Learning is consistent if and only if the estimation error converges to zero as $n, n' \to \infty$.

3.2 Impossibility of risk rewrite

Recall that $R(g)$ is approximated by (5) given $\mathcal{X}_{\mathrm{p}}$ and $\mathcal{X}_{\mathrm{n}}$, which does not work given $\mathcal{X}_{\mathrm{tr}}$ and $\mathcal{X}'_{\mathrm{tr}}$. We might rewrite $R(g)$ such that it can be approximated given $\mathcal{X}_{\mathrm{tr}}$ and/or $\mathcal{X}'_{\mathrm{tr}}$. This is known as the *backward correction* (Natarajan et al., 2013; Patrini et al., 2017) in learning with noisy labels.

Definition 1. *We say that $R(g)$ in (3) is rewritable given p_{tr}, if and only if*[4] *there exist constants a and b, such that for any g it holds that*

$$R(g) = \mathbb{E}_{p_{\mathrm{tr}}}[\bar{\ell}(g(X))], \tag{6}$$

where $\mathbb{E}_{p_{\mathrm{tr}}}[\cdot]$ means $\mathbb{E}_{X\sim p_{\mathrm{tr}}}[\cdot]$ and $\bar{\ell}(z) = a\ell(z) + b\ell(-z)$ is the corrected loss function.

Theorem 2. *Let ℓ be ℓ_{01}, or any bounded surrogate loss satisfying that*

$$0 \leq \ell(+\infty) = \lim_{z\to+\infty} \ell(z) < \lim_{z\to-\infty} \ell(z) = \ell(-\infty) < +\infty. \tag{7}$$

Assume p_{p} and p_{n} are almost surely separable and θ is arbitrary. Then, $R(g)$ is not rewritable.[5]

Theorem 2 shows that under the separability assumption of p_{p} and p_{n}, $R(g)$ is not rewritable. As a consequence, we lack a learning objective, that is, the empirical training risk. It is even worse—we cannot access the empirical validation risk of g after it is trained by other learning methods such as discriminative clustering. In particular, ℓ_{01} satisfies (7), which implies that the common practice of hyperparameter tuning is disabled by Theorem 2, since U validation data also follow p_{tr}.

[4] This is because the backward correction in (6), if exists, would be unique.

[5] Please find in Appendix A the proofs of theorems.

4 LEARNING FROM TWO SETS OF U DATA

From now on, we prove that knowing π_{p}, θ and θ' is sufficient for rewriting $R(g)$.

4.1 POSSIBILITY OF RISK REWRITE, AND UNBIASED RISK ESTIMATORS

We have proven that $R(g)$ is not rewritable given p_{tr}, and Quadrianto et al. (2009) has proven that $R(g)$ can be estimated from $\mathcal{X}_{\mathrm{tr}}$ and $\mathcal{X}'_{\mathrm{tr}}$, where g is a linear model and ℓ is the logistic loss. These facts motivate us to investigate the possibility of rewriting $R(g)$, where g and ℓ are both arbitrary.[6]

Definition 3. *We say that $R(g)$ is rewritable given p_{tr} and p'_{tr}, if and only if*[7] *there exist constants a, b, c and d, such that for any g it holds that*

$$R(g) = \mathbb{E}_{p_{\mathrm{tr}}}[\bar{\ell}_+(g(X))] + \mathbb{E}_{p'_{\mathrm{tr}}}[\bar{\ell}_-(-g(X))], \tag{8}$$

where $\bar{\ell}_+(z) = a\ell(z) + b\ell(-z)$ and $\bar{\ell}_-(z) = c\ell(z) + d\ell(-z)$ are the corrected loss functions.

Theorem 4. *Assume θ and θ' are arbitrary but satisfy $\theta > \theta'$; otherwise, swap p_{tr} and p'_{tr} to make sure $\theta > \theta'$. Then, $R(g)$ is rewritable, by letting*

$$a = \frac{(1-\theta')\pi_{\mathrm{p}}}{\theta-\theta'}, \quad b = -\frac{\theta'(1-\pi_{\mathrm{p}})}{\theta-\theta'}, \quad c = \frac{\theta(1-\pi_{\mathrm{p}})}{\theta-\theta'}, \quad d = -\frac{(1-\theta)\pi_{\mathrm{p}}}{\theta-\theta'}. \tag{9}$$

Theorem (4) immediately leads to an unbiased risk estimator, namely

$$\begin{aligned}\widehat{R}_{\mathrm{uu}}(g) = {} & \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\left(\tfrac{(1-\theta')\pi_{\mathrm{p}}}{\theta-\theta'}\ell(g(x_i)) - \tfrac{\theta'(1-\pi_{\mathrm{p}})}{\theta-\theta'}\ell(g(-x_i))\right) \\ & + \tfrac{1}{n'}\textstyle\sum_{j=1}^{n'}\left(-\tfrac{(1-\theta)\pi_{\mathrm{p}}}{\theta-\theta'}\ell(g(x'_j)) + \tfrac{\theta(1-\pi_{\mathrm{p}})}{\theta-\theta'}\ell(-g(x'_j))\right).\end{aligned} \tag{10}$$

Eq. (10) is useful for both training (by plugging U training data into it) and hyperparameter tuning (by plugging U validation data into it). We hereafter refer to the process of obtaining the empirical risk minimizer of (10), i.e., $\widehat{g}_{\mathrm{uu}} = \arg\min_{g\in\mathcal{G}} \widehat{R}_{\mathrm{uu}}(g)$, as *unlabeled-unlabeled* (UU) *learning*. The proposed UU learning is by nature ERM-based, and consequently $\widehat{g}_{\mathrm{uu}}$ can be obtained by powerful *stochastic optimization* algorithms (e.g., Duchi et al., 2011; Kingma & Ba, 2015).

Simplification Note that (10) may require some efforts to implement. Fortunately, it can be simplified by employing ℓ that satisfies a *symmetric condition*:

$$\ell(z) + \ell(-z) = 1. \tag{11}$$

Eq. (11) covers ℓ_{01}, the ramp loss $\ell_{\mathrm{ramp}}(z) = \max\{0, \min\{1, (1-z)/2\}\}$ (du Plessis et al., 2014; Niu et al., 2016) and the sigmoid loss $\ell_{\mathrm{sig}}(z) = 1/(1+\exp(z))$ (Kiryo et al., 2017). With the help of (11), we can simplify (10) as

$$\widehat{R}^{\mathrm{Sym}}_{\mathrm{uu}}(g) = \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\alpha\ell(g(x_i)) + \tfrac{1}{n'}\sum_{j=1}^{n'}\alpha'\ell(-g(x'_j)) - \tfrac{\theta'(1-\pi_{\mathrm{p}})+(1-\theta)\pi_{\mathrm{p}}}{\theta-\theta'}, \tag{12}$$

where $\alpha = (\theta' + \pi_{\mathrm{p}} - 2\theta'\pi_{\mathrm{p}})/(\theta-\theta')$ and $\alpha' = (\theta + \pi_{\mathrm{p}} - 2\theta\pi_{\mathrm{p}})/(\theta-\theta')$. Just like (10), (12) is an unbiased risk estimator, and it is easy to implement in many deep learning frameworks.

Special cases Consider some special cases of (10) by specifying θ and θ'. It is obvious that (10) reduces to (5) for supervised learning, if $\theta = 1$ and $\theta' = 0$. Next, (10) reduces to

$$\widehat{R}_{\mathrm{uu}}(g) = \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\pi_{\mathrm{p}}\ell(g(x_i)) - \tfrac{1}{n}\sum_{i=1}^{n}\pi_{\mathrm{p}}\ell(-g(x_i)) + \tfrac{1}{n'}\sum_{j=1}^{n'}\ell(-g(x'_j)),$$

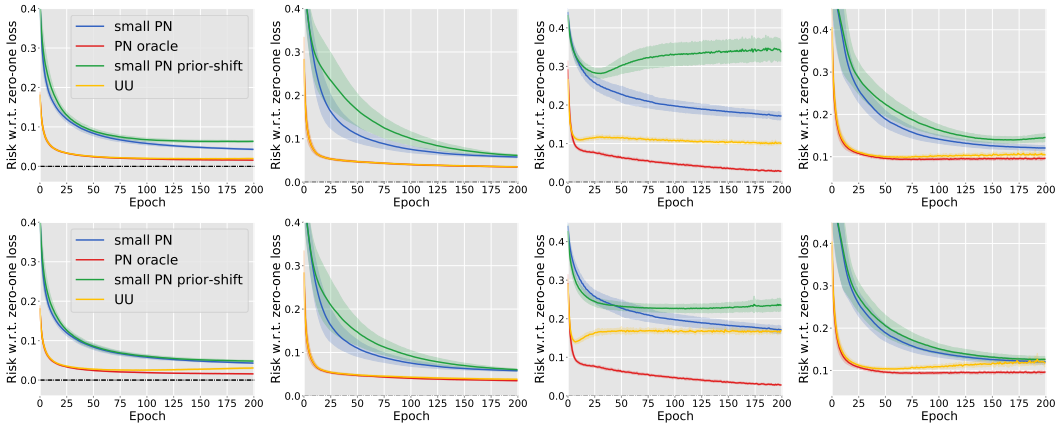
if $\theta = 1$ and $\theta' = \pi_{\mathrm{p}}$, and we recover the unbiased risk estimator in positive-unlabeled learning (du Plessis et al., 2015; Kiryo et al., 2017). Additionally, (10) reduces to a fairly complicated unbiased risk estimator in similar-unlabeled learning (Bao et al., 2018), if $\theta = \pi_{\mathrm{p}}, \theta' = \pi_{\mathrm{p}}^2/(2\pi_{\mathrm{p}}^2 - 2\pi_{\mathrm{p}} + 1)$ or vice versa. Therefore, UU learning is a very general framework in weakly-supervised learning.

[6] The technique that underlies Theorem 4 is totally different from Quadrianto et al. (2009). We shall obtain (9) by solving a linear system resulted from Definition 3. As previously mentioned, Quadrianto et al. (2009) is based on mean operators, and it cannot be further generalized to handle nonlinear g or arbitrary ℓ.

[7] This is similarly because the backward correction in (8), if exists, would be unique.

Table 1: Specification of benchmark datasets, models, and optimization algorithms.

Dataset	# Train	# Test	# Feature	π_{p}	Model $g(x;\theta)$	Opt.
MNIST	60,000	10,000	784	0.49	5-layer FC with ReLU	SGD
Fashion-MNIST	60,000	10,000	784	0.50	5-layer FC with ReLU	SGD
SVHN	100,000	26,032	3,072	0.27	12-layer CNN with ReLU	Adam
CIFAR-10	50,000	10,000	3,072	0.60	32-layer ResNet with ReLU	Adam



(a) MNIST (b) Fashion-MNIST (c) SVHN (d) CIFAR-10

Figure 2: Experimental results of training the deep neural networks with ℓ_{sig}. The top/bottom row corresponds to $(\theta, \theta') = (0.9, 0.1)/(0.8, 0.2)$.

4.2 Consistency and convergence rate

The consistency of UU learning is guaranteed due to the unbiasedness of (10). In what follows, we analyze the estimation error $R(\widehat{g}_{\mathrm{uu}}) - R(g^*)$ (see Sec. 3.1 for the definition). To this end, assume there are $C_g > 0$ and $C_\ell > 0$ such that $\sup_{g\in\mathcal{G}} \|g\|_\infty \leq C_g$ and $\sup_{|z|\leq C_g} \ell(z) \leq C_\ell$, and assume $\ell(z)$ is Lipschitz continuous for all $|z| \leq C_g$ with a Lipschitz constant L_ℓ. Let $\mathfrak{R}_n(\mathcal{G})$ and $\mathfrak{R}'_{n'}(\mathcal{G})$ be the *Rademacher complexity* of $\mathcal{G}$ over $p_{\mathrm{tr}}(x)$ and $p'_{\mathrm{tr}}(x)$ (Mohri et al., 2012; Shalev-Shwartz & Ben-David, 2014). For convenience, denote by $\chi_{n,n'} = \alpha/\sqrt{n} + \alpha'/\sqrt{n'}$.

Lemma 5. *For any $\delta > 0$, let $C_\delta = \sqrt{(\ln 2/\delta)/2}$, then we have with probability at least $1-\delta$,*

$$\sup_{g\in\mathcal{G}} |\widehat{R}_{\mathrm{uu}}(g) - R(g)| \leq 2L_\ell\alpha\mathfrak{R}_n(\mathcal{G}) + 2L_\ell\alpha'\mathfrak{R}'_{n'}(\mathcal{G}) + C_\ell C_\delta \chi_{n,n'}, \tag{13}$$

where the probability is over repeated sampling of data for evaluating $\widehat{R}_{\mathrm{uu}}(g)$.

Theorem 6. *For any $\delta > 0$, let $C_\delta = \sqrt{(\ln 2/\delta)/2}$, then we have with probability at least $1-\delta$,*

$$R(\widehat{g}_{\mathrm{uu}}) - R(g^*) \leq 4L_\ell\alpha\mathfrak{R}_n(\mathcal{G}) + 4L_\ell\alpha'\mathfrak{R}'_{n'}(\mathcal{G}) + 2C_\ell C_\delta \chi_{n,n'}, \tag{14}$$

where the probability is over repeated sampling of data for training $\widehat{g}_{\mathrm{uu}}$.

Theorem 6 ensures that UU learning is consistent (and so are all the special cases): as $n, n' \to \infty$, $R(\widehat{g}_{\mathrm{uu}}) \to R(g^*)$, since $\mathfrak{R}_n(\mathcal{G}), \mathfrak{R}'_{n'}(\mathcal{G}) \to 0$ for all parametric models with a bounded norm such as deep networks trained with weight decay. Moreover, $R(\widehat{g}_{\mathrm{uu}}) \to R(g^*)$ in $\mathcal{O}_p(\chi_{n,n'})$, where $\mathcal{O}_p$ denotes the order in probability, for linear-in-parameter models and non-parametric kernel models in *reproducing kernel Hilbert spaces* with a bounded norm (Schölkopf & Smola, 2001).

5 Experiments

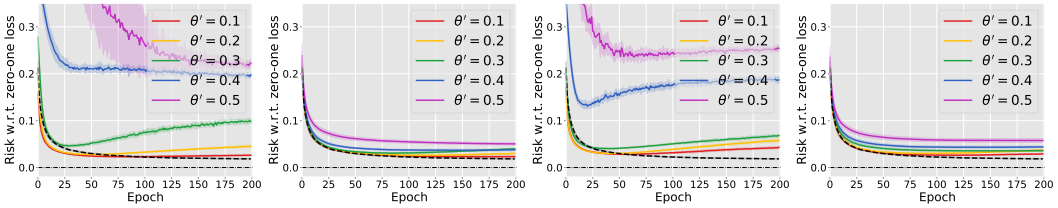
In this section, we experimentally analyze the proposed algorithm on deep neural network models trained on various benchmark datasets. The implementation is based on Keras.

5.1 Benchmark experiments with neural network models

We first illustrate the operation of the proposed unbiased risk estimator and evaluate it with 3 supervised baselines: small PN, PN oracle and small PN prior-shift, where small PN/PN oracle denotes

Table 2: Mean errors (standard deviations) in percentage given inaccurate training class priors.

Dataset	(θ, θ')	$\epsilon = 0.8$	$\epsilon = 0.9$	$\epsilon = 1.0$	$\epsilon = 1.1$	$\epsilon = 1.2$
MNIST	$(0.9, 0.1)$	2.90(0.12)	2.54(0.10)	2.31(0.15)	2.08(0.10)	2.10(0.06)
	$(0.8, 0.2)$	3.64(0.13)	3.29(0.15)	3.01(0.12)	2.78(0.10)	2.63(0.16)
	$(0.7, 0.3)$	5.15(0.23)	4.87(0.29)	4.84(0.25)	4.75(0.24)	4.68(0.28)
CIFAR-10	$(0.9, 0.1)$	10.50(0.36)	10.30(0.35)	10.15(0.31)	9.82(0.36)	9.81(0.35)
	$(0.8, 0.2)$	11.27(0.40)	10.94(0.41)	10.77(0.41)	10.56(0.37)	10.30(0.39)
	$(0.7, 0.3)$	12.50(0.59)	12.23(0.56)	11.91(0.55)	11.59(0.52)	11.38(0.48)



(a) CCN, $\theta = 0.9$ (b) UU, $\theta = 0.9$ (c) CCN, $\theta = 0.8$ (d) UU, $\theta = 0.8$

Figure 3: Illustration of how moving θ and θ' closer affects the classification performance of CCN and the proposed UU method. PN oracle is illustrated in black dashed line.

fully supervised classification with 10%/100% training data, and small PN prior-shift means supervised classification with 10% training data under class-prior change.[8] Note that small PN and PN oracle use fully supervised training data generated from the same distribution as the test data, which is extremely advantageous.

We test on the widely adopted benchmarks, MNIST, Fashion-MNIST, SVHN and CIFAR-10. Table 1 summarizes the specification of the datasets.[9] For each dataset, we draw equal amount of unlabeled training data for $\mathcal{X}_{\mathrm{tr}}$ and $\mathcal{X}'_{\mathrm{tr}}$ from Eq. (1), where two different pairs of training class priors are considered: $(0.9, 0.1)$, $(0.8, 0.2)$. The test data is directly drawn from the original joint distribution $p(x, y)$ for evaluating the performance of the trained models.

The model and optimizer for MNIST and Fashion-MNIST were 5-layer *fully connected neural network* (FC) and SGD (Robbins & Monro, 1951). For SVHN and CIFAR-10, we used 12-layer *all convolutional net* (Springenberg et al., 2015) and 32-layer *residual network* (ResNet) (He et al., 2016) respectively, and the resulting objectives were minimized by Adam (Kingma & Ba, 2015). We compared the performance of the proposed risk estimator Eq. (10) with logistic loss $\ell_{\log}(z) = log(1 + \exp(-z))$ (Natarajan et al., 2013) and the simplified version Eq. (12) with ℓ_{sig} in Appendix C.2. The results show that the risk estimators with ℓ_{sig} and $\ell_{\log}$ perform similiarly. For simplicity, we select ℓ_{sig} as the surrogate loss for training (0-1 loss for testing) in the following experiments. More details on experimental setup can be found in Appendix C.1.

The experimental results are reported in Figure 2, where means and standard deviations of test risks based on the same 10 random samplings are shown. In the case of $(\theta, \theta') = (0.9, 0.1)$, we can see the proposed UU classifiers are comparable to the PN oracle in most cases. Note that the test class prior of SVHN is 0.27, which is farther from the corrupted UU class prior 0.5. Thus this setting is more advantageous for PN baselines without class prior shift but more challenging for UU. The results in (c) show that UU still outperforms small PN while small PN prior-shift deteriorates severely. For the harder case of $(\theta, \theta') = (0.8, 0.2)$, the performances of UU classifiers drop slightly, but is still comparable to small PN. The drop here can be explained by larger noise in the U sets when moving θ and θ' closer, and we investigate this issue in Figure 3.

Analysis of moving θ and θ' closer It is intuitive that if θ and θ' are closer, the U sets will be less informative. To investigate the influence of this, we conducted additional experiments on MNIST by moving θ and θ' closer, where $\theta \in \{0.9, 0.8\}$ and θ' is gradually moved from 0.1 to 0.5. The experimental setup is exactly same as before. We reported the means and standard deviations of the test risks over 10 trails in Figure 3. The results show that the proposed unbiased UU method works reasonably well, while the performance of CCN drops severely. It is because the marginal $p(x)$ is shifted more distant from training to test stages as θ' moves closer to θ, which will make the

[8] PN prior-shift with 100% data baseline is not included because of insufficient number of training data.

[9] We artificially convert them into binary classification datasets, see Appendix C.1 for details.

Table 3: Means and standard deviations of classification errors over 10 trials in percentage. Best and comparable methods based on the t-test at the significance level 1% are highlighted in boldface.

Dataset	# Train	# Test	π_{p}	pSVM	BER	BER-FC	UU
pendigits	971	350	0.10	4.03(0.27)	5.51(1.35)	5.46(1.23)	**1.97(0.78)**
covtype-binary	3863	1500	0.30	14.63(1.00)	11.33(0.26)	**5.17(0.57)**	**4.97(0.48)**
MNIST	11640	2000	0.50	N/A	**3.10(0.17)**	**3.03(0.25)**	**2.87(0.28)**
spambase	1139	400	0.70	29.18(1.29)	**11.28(1.73)**	13.98(1.63)	**12.53(1.00)**
letter	532	200	0.90	15.65(4.18)	15.45(6.99)	8.45(2.92)	**3.15(0.84)**
USPS	971	350	0.10	5.91(1.52)	12.69(4.09)	8.57(2.40)	**3.74(1.24)**
	2605	800	0.30	5.55(0.46)	5.36(0.41)	**2.75(0.28)**	**2.63(0.18)**
	1695	600	0.50	9.27(0.61)	**7.27(1.09)**	**5.48(1.33)**	**5.52(1.02)**
	1853	600	0.70	8.20(0.73)	7.48(0.65)	**4.23(0.50)**	**4.43(0.94)**
	424	150	0.90	9.80(2.07)	14.13(2.02)	18.27(5.17)	**6.20(1.33)**

CCN-based risk estimator more biased. Thanks to the unbiasedness of the proposed risk estimator, UU classifiers can handle the covariate shift reasonably well. We further analyzed the influence of covariate shift by changing the sample sizes of the U sets in Appendix C.2, the experimental results are consistent.

Robustness of noisy training class priors In the above experiments, we assume the true training class priors θ and θ' are exactly accessible, but in practice we may only be able to approximately specify them. This motivates our study to try some cases when θ and θ' are misspecified, in order to simulate UU learning in the wild. We run more experiments by replacing the true training class priors (θ, θ') with $(\vartheta, \vartheta') = (\epsilon\theta, \epsilon\theta')$ and give (ϑ, ϑ') to the learning method. The experimental setup is exactly same as before except that the training class priors are noisy. We reported the classification errors of the learned models in Table 2. The results show that the proposed method is fairly robust to the misspecification of training class priors so long as $|\vartheta - \theta| + |\vartheta' - \theta'| \ll |\theta - \theta'|$.

5.2 Comparison with state-of-the-art methods

We finally compare our method with two state-of-the-art methods for dealing with two sets of U data: the proportion-SVM method (pSVM) (Yu et al., 2013) and the balanced error minimisation method (BER) (Menon et al., 2015). We downloaded the codes from the webpage of authors.Note that the pSVM method is based on *maximum margin clustering* (Xu et al., 2004; Valizadegan & Jin, 2006; Li et al., 2009) and the original codes of BER implement neural network by Matlab and use the second order optimization method. Considering the time for 10 runs of each experiment, we used UCI benchmarksand USPS datasetsfor the experiment following pSVM and BER. By re-sampling the original datasets, we test several different settings of class prior π_{p}.

The 5-layer FC and SGD were again used for training UU classifier here. For fairness, we also implemented BER method using the same network architecture and optimizer as UU (BER-FC). The specifications of the datasets are summarized and the experimental results are reported in Table 3. We can see that the proposed method outperforms others in most cases. The closer π_{p} is to $\frac{1}{2}$, the better classification performance of BER and BER-FC. In particular, in the experiments of MNIST and USPS ($\pi_{\mathrm{p}} = 0.5$), BER and BER-FC can achieve comparable results as UU where pSVM falls behind. This is because the balanced error assumption holds in this case. However, in the experiments of pendigits and USPS ($\pi_{\mathrm{p}} = 0.1, 0.9$), our method still works well while the performance of BER and BER-FC drops severely and become inferior to pSVM.

6 Conclusions

We focused on training arbitrary binary classifier, including deep networks, from only U data by ERM. We proved that this is impossible given a single set of U data, but it is possible given two sets of U data with different class priors, where all class priors necessary for training are given. This led to an unbiased risk estimator and subsequently we proposed the first ERM-based learning method from two sets of U data. Experiments demonstrated that the proposed method could even successfully train AllConvNet and ResNet, and it compared favorably with state-of-the-art methods for learning from two sets of U data.

REFERENCES

D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

H. Bao, G. Niu, and M. Sugiyama. Classification from pairwise similarity and unlabeled data. In *ICML*, 2018.

P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann. The balanced accuracy and its posterior distribution. In *ICPR*, 2010.

O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *NIPS*, 2002.

O. Chapelle, B. Schölkopf, and A. Zien (eds.). *Semi-Supervised Learning*. MIT Press, 2006.

J. Cheng, T. Liu, K. Ramamohanarao, and D. Tao. Learning with bounded instance- and label-dependent label noise. *arXiv preprint arXiv:1709.03768*, 2017.

M. C. du Plessis, G. Niu, and M. Sugiyama. Clustering unclustered data: Unsupervised binary labeling of two datasets having different class balances. In *TAAI*, 2013.

M. C. du Plessis, G. Niu, and M. Sugiyama. Analysis of learning from positive and unlabeled data. In *NIPS*, 2014.

M. C. du Plessis, G. Niu, and M. Sugiyama. Convex formulation for learning from positive and unlabeled data. In *ICML*, 2015.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *KDD*, 2008.

J. Goldberger and E. Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

R. Gomes, A. Krause, and P. Perona. Discriminative clustering by regularized information maximization. In *NIPS*, 2010.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2004.

B. Han, J. Yao, G. Niu, M. Zhou, I. W. Tsang, Y. Zhang, and M. Sugiyama. Masking: A new perspective of noisy supervision. In *NIPS*, 2018a.

B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama. Co-teaching: Robust training deep neural networks with extremely noisy labels. In *NIPS*, 2018b.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. Learning discrete representations via information maximizing self augmented training. In *ICML*, 2017.

L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and F.-F. Li. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018.

D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *NIPS*, 2017.

S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

Y.-F. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou. Tighter and convex maximum margin clustering. In *AISTATS*, 2009.

T. Liu and D. Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2016.

Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *CVPR*, 2018.

G. S. Mann and A. McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *ICML*, 2007.

C. McDiarmid. On the method of bounded differences. In J. Siemons (ed.), *Surveys in Combinatorics*, pp. 148–188. Cambridge University Press, 1989.

A. K. Menon, B. van Rooyen, C. S. Ong, and R. C. Williamson. Learning from corrupted binary labels via class-probability estimation. In *ICML*, 2015.

A. K. Menon, B. van Rooyen, and N. Natarajan. Learning from binary labels with instance-dependent corruption. *arXiv preprint arXiv:1605.00751*, 2016.

T. Miyato, S. Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional smoothing with virtual adversarial training. In *ICLR*, 2016.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012.

N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari. Learning with noisy labels. In *NIPS*, 2013.

G. Niu, W. Jitkrittum, B. Dai, H. Hachiya, and M. Sugiyama. Squared-loss mutual information regularization: A novel information-theoretic approach to semi-supervised learning. In *ICML*, 2013.

G. Niu, M. C. du Plessis, T. Sakai, Y. Ma, and M. Sugiyama. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *NIPS*, 2016.

G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017.

N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. Estimating labels from label proportions. *Journal of Machine Learning Research*, 10:2349–2374, 2009.

J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009.

S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR workshop*, 2015.

M. D. Reid and R. C. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11:2387–2422, 2010.

M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2001.

C. Scott, G. Blanchard, and G. Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *COLT*, 2013.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR*, 2015.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15: 1929–1958, 2014.

M. Sugiyama, G. Niu, M. Yamada, M. Kimura, and H. Hachiya. Information-maximization clustering based on squared-loss mutual information. *Neural Computation*, 26(1):84–131, 2014.

S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. In *ICLR workshop*, 2015.

A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, 2017.

A. Tewari and P. L. Bartlett. On the consistency of multi-class classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

A. N. Tikhonov. On the stability of inverse problems (in Russian). *Doklady Akademii Nauk SSSR*, 39(5):195–198, 1943.

H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. In *NIPS*, 2006.

B. van Rooyen and R. C. Williamson. A theory of learning with corrupted labels. *Journal of Machine Learning Research*, 18(228):1–50, 2018.

V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

G. Ward, T. Hastie, S. Barry, J. Elith, and J. Leathwick. Presence-only data and the EM algorithm. *Biometrics*, 65(2):554–563, 2009.

L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *NIPS*, 2004.

F. X. Yu, D. Liu, S. Kumar, T. Jebara, and S.-F. Chang. $\propto$SVM for learning with label proportions. In *ICML*, 2013.

A PROOFS

In this appendix, we prove all theorems.

A.1 PROOF OF THEOREM 2

We prove the theorem by contradiction, namely for any such $p(x, y)$, for all θ and a, b, we are able to find some g for which (6) fails. Our argument goes from the special ℓ_{01} to the general ℓ in (7).

Firstly, let $g(x) = +\infty$ identically, so that $\ell(g(x)) = 0$ and $\ell(-g(x)) = 1$. Plugging them into (3) and (6), we obtain that $b = 1 - \pi_{\mathrm{p}}$. Secondly, let $g(x) = -\infty$; this time $\ell(g(x)) = 1$ and $\ell(-g(x)) = 0$, and we obtain that $a = \pi_{\mathrm{p}}$. Thirdly, let $g(x) = +\infty$ over p_{p} and $g(x) = -\infty$ over p_{n}, which is possible because g is arbitrary. Hence, $\ell(g(x)) = 0$ and $\ell(-g(x)) = 1$ over p_{p} as well as $\ell(g(x)) = 1$ and $\ell(-g(x)) = 0$ over p_{n}, resulting in $0 = \theta b + (1 - \theta)a$ by solving which we know that $\theta = \pi_{\mathrm{p}}/(2\pi_{\mathrm{p}} - 1)$. This is a contradiction unless $\pi_{\mathrm{p}} = 0$ or $\pi_{\mathrm{p}} = 1$, because $0 \leq \theta \leq 1$ whereas either $\pi_{\mathrm{p}}/(2\pi_{\mathrm{p}} - 1) < 0$ or $\pi_{\mathrm{p}}/(2\pi_{\mathrm{p}} - 1) > 1$.

Finally, given any ℓ satisfying (7), it is not difficult to verify that the three g above lead to the same contradiction with $b = (1 - \pi_{\mathrm{p}})$ and the same a and θ. □

A.2 PROOF OF THEOREM 4

By Definition 3, the learning objective is defined by

$$J(g) = \mathbb{E}_{p_{\mathrm{tr}}}[\bar{\ell}_+(g(X))] + \mathbb{E}_{p'_{\mathrm{tr}}}[\bar{\ell}_-(-g(X))], \tag{15}$$

then

$$\begin{aligned}
J(g) &= \mathbb{E}_{p_{\mathrm{tr}}}[a\ell(g(X)) + b\ell(-g(X))] + \mathbb{E}_{p'_{\mathrm{tr}}}[c\ell(-g(X)) + d\ell(g(X))] \\
&= \theta\mathbb{E}_{\mathrm{p}}[a\ell(g(X)) + b\ell(-g(X))] + (1-\theta)\mathbb{E}_{\mathrm{n}}[a\ell(g(X)) + b\ell(-g(X))] \\
&+ \theta'\mathbb{E}_{\mathrm{p}}[c\ell(-g(X)) + d\ell(g(X))] + (1-\theta')\mathbb{E}_{\mathrm{n}}[c\ell(-g(X)) + d\ell(g(X))] \\
&= (a\theta + d\theta')\mathbb{E}_{\mathrm{p}}[\ell(g(X))] + (b\theta + c\theta')\mathbb{E}_{\mathrm{p}}[\ell(-g(X))] \\
&+ [a(1-\theta) + d(1-\theta')]\mathbb{E}_{\mathrm{n}}[\ell(g(X))] + [b(1-\theta) + c(1-\theta')]\mathbb{E}_{\mathrm{n}}[\ell(-g(X))].
\end{aligned}$$

Recall that the risk of g is defined as (3):

$$R(g) = \mathbb{E}_{(X,Y)\sim p(x,y)}[\ell(Yg(X))] = \pi_{\mathrm{p}}\mathbb{E}_{\mathrm{p}}[\ell(g(X))] + (1-\pi_{\mathrm{p}})\mathbb{E}_{\mathrm{n}}[\ell(-g(X))].$$

In order to minimize (3), it suffices to minimize (15), if we can make

$$\begin{aligned}
a\theta + d\theta' &= \pi_{\mathrm{p}}, \\
b\theta + c\theta' &= 0, \\
a(1-\theta) + d(1-\theta') &= 0, \\
b(1-\theta) + c(1-\theta') &= 1 - \pi_{\mathrm{p}}.
\end{aligned}$$

Solving these equations gives us

$$a = \frac{(1-\theta')\pi_{\mathrm{p}}}{\theta - \theta'}, \quad b = -\frac{\theta'(1-\pi_{\mathrm{p}})}{\theta - \theta'}, \quad c = \frac{\theta(1-\pi_{\mathrm{p}})}{\theta - \theta'}, \quad d = -\frac{(1-\theta)\pi_{\mathrm{p}}}{\theta - \theta'},$$

which concludes the proof. □

A.3 PROOF OF LEMMA 5

Consider the one-side uniform deviation $\sup_{g\in\mathcal{G}} \widehat{R}_{\mathrm{uu}}(g) - R(g)$. Since $0 \leq \ell(z) \leq C_\ell$, the change of it will be no more than $C_\ell\alpha/n$ if some x_i is replaced, or no more than $C_\ell\alpha'/n'$ if some x'_j is replaced. Subsequently, *McDiarmid's inequality* (McDiarmid, 1989) tells us that

$$\Pr\{\sup_{g\in\mathcal{G}} \widehat{R}_{\mathrm{uu}}(g) - R(g) - \mathbb{E}[\sup_{g\in\mathcal{G}} \widehat{R}_{\mathrm{uu}}(g) - R(g)] \geq \epsilon\} \leq \exp\left(-\frac{2\epsilon^2}{C_\ell^2(\alpha^2/n + \alpha'^2/n')}\right),$$

or equivalently, with probability at least $1-\delta/2$,

$$\sup_{g\in\mathcal{G}} \widehat{R}_{\mathrm{uu}}(g) - R(g) \leq \mathbb{E}[\sup_{g\in\mathcal{G}} \widehat{R}_{\mathrm{uu}}(g) - R(g)] + C_\ell(\alpha/\sqrt{n} + \alpha'/\sqrt{n'})\sqrt{(\ln 2/\delta)/2}$$
$$= \mathbb{E}[\sup_{g\in\mathcal{G}} \widehat{R}_{\mathrm{uu}}(g) - R(g)] + C_\ell C_\delta \chi_{n,n'}.$$

By *symmetrization* (Vapnik, 1998), it is a routine work to show that

$$\mathbb{E}[\sup_{g\in\mathcal{G}} \widehat{R}_{\mathrm{uu}}(g) - R(g)] \leq 2\alpha\mathfrak{R}_n(\ell\circ\mathcal{G}) + 2\alpha'\mathfrak{R}'_{n'}(\ell\circ\mathcal{G}),$$

and according to *Talagrand's contraction lemma* (Shalev-Shwartz & Ben-David, 2014),

$$\mathfrak{R}_n(\ell\circ\mathcal{G}) \leq L_\ell\mathfrak{R}_n(\mathcal{G}), \quad \mathfrak{R}'_{n'}(\ell\circ\mathcal{G}) \leq L_\ell\mathfrak{R}'_{n'}(\mathcal{G}).$$

The one-side uniform deviation $\sup_{g\in\mathcal{G}} R(g) - \widehat{R}_{\mathrm{uu}}(g)$ can be bounded similarly. □

A.4 Proof of Theorem 6

Based on Lemma 5, the estimation error bound (14) is proven through

$$R(\widehat{g}_{\mathrm{uu}}) - R(g^*) = \left(\widehat{R}_{\mathrm{uu}}(\widehat{g}_{\mathrm{uu}}) - \widehat{R}_{\mathrm{uu}}(g^*)\right) + \left(R(\widehat{g}_{\mathrm{uu}}) - \widehat{R}_{\mathrm{uu}}(\widehat{g}_{\mathrm{uu}})\right) + \left(\widehat{R}_{\mathrm{uu}}(g^*) - R(g^*)\right)$$
$$\leq 0 + 2\sup_{g\in\mathcal{G}} |\widehat{R}_{\mathrm{uu}}(g) - R(g)|$$
$$\leq 4L_\ell\alpha\mathfrak{R}_n(\mathcal{G}) + 4L_\ell\alpha'\mathfrak{R}'_{n'}(\mathcal{G}) + 2C_\ell C_\delta\chi_{n,n'},$$

where $\widehat{R}_{\mathrm{uu}}(\widehat{g}_{\mathrm{uu}}) \leq \widehat{R}_{\mathrm{uu}}(g^*)$ by the definition of $\widehat{g}_{\mathrm{uu}}$. □

B Synthetic Experiments with Linear Models

Here we illustrate the problem and our method with a synthetic Gaussian mixture dataset. We used two-dimensional Gaussian distributions with means $\boldsymbol{\mu}_+ = [1,1]^T$ and $\boldsymbol{\mu}_- = [-1,-1]^T$ and identity covariance for P and N class-conditional densities respectively. Two unlabeled training datasets were drawn from Eq. (1), where the training class priors are $(\theta,\theta') = (0.9, 0.4)$ and the sample sizes are $(n,n') = (2000, 1000)$. The test data were drawn from the original joint distribution $p(x,y)$ for evaluating the performance of trained models and the test class prior is $\pi_{\mathrm{p}} = 0.3$. Note that the marginal distribution $p(x)$ changes between training and testing stages (see Figure 1 (c), (d) and the right panel), which is the key difference between the problem of interest and the problem of learning from class-conditional label noise. For the toy experiments, linear-in-input model $g(X) = \boldsymbol{\omega}^T X + b$, where $\boldsymbol{\omega} \in \mathbb{R}^2$ and $b \in \mathbb{R}$, and sigmoid loss were commonly used. SGD (Robbins & Monro, 1951) with mini-batch size 128 was employed for optimization. For the purpose of clear comparison of the risk estimators, we did not add regularization in the toy experiments. The learned models trained with 500 epochs are shown in Figure 1.

C Benchmark experiments with neural network models

C.1 Experimental Setup

MNIST The MNIST dataset contains 60,000 gray-scale training images and 10,000 test images (input dimension is 784) from handwritten digits 0 to 9. Since the dataset has ten classes (0 to 9) originally, we used the even and odd digits to constitute the P and N classes respectively in order to construct a dataset for binary classification, and the resulting class prior is 0.49. The model used for MNIST was a five-layer fully connected neural network with ReLU as the activation functions: d-300-300-300-300-1. Batch normalization was applied before hidden layers, and an ℓ_2-regularization (the ℓ_2 coefficient is fixed to 10^{-4}) was also added. The model was trained by SGD with mini-batch size 128, and the learning rate was decreased according to the $1/(1+decay*epochs)$ schedule built in Keras, where the decay candidates were chosen from $\{0, 10^{-6}, 10^{-5}, 10^{-4}, 5\times10^{-4}\}$.

Fashion-MNIST Fashion-MNIST is also a grayscale image dataset consisting of a training set of 60,000 examples and a test set of 10,000 examples (input dimension is 784), associated with a label from 10 fashion item classes. We artificially corrupted it into a binary classification dataset as follows: P class is formed by 'T-shirt/top', 'Pullover', 'Coat', 'Shirt', 'Bag', and N class is formed by 'Trouser', 'Dress', 'Sandal', 'Sneaker', 'Ankle boot', and the resulting class prior is 0.50. Moreover, the models and optimization of Fashion-MNIST follow the setup of MNIST.

SVHN The SVHN dataset is a 32×32 color house-number image dataset containing 73,257 training images, 26,032 test images, and 531,131 extra training images. We sampled 100,000 training examples from the concatenation of the training data and extra training data. The extra training data were used in order to ensure enough training data to perform class prior change in Figure 2 (c) and (g). For SVHN, '0', '6', '8', '9' make up the P class, and '1', '2', '3', '4', '5', '7' make up the N class, and the class prior is 0.27. The model for SVHN was a 12-layer CNN: d-$[C(3*3,96)]*2$-$C(3*3,96,2)$-$[C(3*3,192)]*2$-$C(3*3,192,2)$-C(3*3,192)-$C(1*1,192)$-$C(1*1,10)$-1000-1000-1, where C(3*3,96) means 96 channels of 3×3 convolutions followed by ReLU, $[\cdot]$*2 means there are 2 such layers, and C(3*3,96,2) means a similar layer but with stride 2, etc., and the padding scheme is same padding. Batch normalization was again applied and ℓ_2-regularization (the ℓ_2 coefficient is fixed to $5*10^{-3}$) was also added. We used Adam optimizer with the default momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and mini-batche size 500 to train the network.

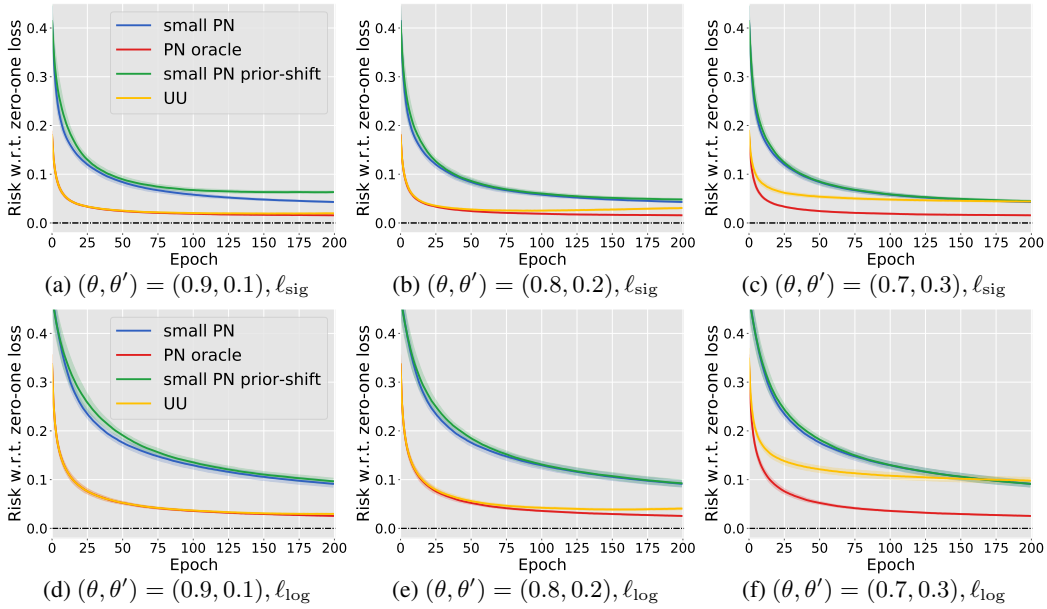
CIFAR-10 The CIFAR-10 dataset consists of 32×32 natural RGB images from 10 classes, with 5,000 training examples and 1,000 test examples in each class. For CIFAR-10: P class is composed of 'bird', 'cat', 'deer', 'dog', 'frog' and 'horse' with class prior 0.60, and N class is composed of 'airplane', 'automobile', 'ship' and 'truck'. In CIFAR-10 experiments, we used the following 32-layer ResNet architecture: d-$C(3*3*16)$-$[C(3*3*16), C(3*3*16)]*5$-$[C(3*3*32,2), C(3*3*32)]$-$[C(3*3*32), C(3*3*32)]*4$-$[C(3*3*64,2), C(3*3*64)]$-$[C(3*3*64), C(3*3*64)]*4$-Global average pool-1, where $C(3*3*16)$ means 16 channels of 3×3 convolutions followed by ReLU, $[\cdot,\cdot]$ means a *building block* (He et al., 2016) for ResNet, $[\cdot]*5$ means there are 5 such layers, $C(3*3*32,2)$ means similar layer but with stride 2, etc., and the padding scheme is same padding. The regularization and optimizer were the same as the setup for SVHN.

Training details For fair comparison, we used the same network structures and optimization methods for the proposed method and the corresponding baselines. In the experiments of Figure 2, we use 60,000/60,000/100,000/50,000 training data for experiments on MNIST/Fashion-MNIST/SVHN/CIFAR-10 and each U has 30,000/30,000/50,000/25,000 unlabeled training data. In the additional experiments of analysis of moving θ and θ' closer, robustness of noisy training class priors and analysis of changing sample sizes of two U sets, we need to sample more data from $p_{\mathrm{p}}(x)$, for example in the case of $(\theta, \theta') = (0.9, 0.4)$. In order to ensure enough P data for sampling, we use 40,000 training data for MNIST and each U has 20,000 unlabeled training data.

C.2 Supplementary experimental results

Comparison of different losses Here we illustrate the operation of proposed unbiased risk estimator Eq. (10) with logistic loss $\ell_{\log}(z) = log(1+\exp(-z))$ and its simplified version Eq. (12) with sigmoid loss ℓ_{sig}. We tested on MNIST with $(\theta, \theta') \in \{(0.9, 0.1), (0.8, 0.2), (0.7, 0.3)\}$. The experimental results are reported in Figure 4, where means and standard deviations of test risks based on the same 10 random samplings are shown. We can see that the risk estimator Eq. (10) with $\ell_{\log}$ performs similarly to Eq. (12) with ℓ_{sig}. It is also interesting to see that the proposed UU classifiers are comparable to the PN oracle that can access fully supervised training data up to $(\theta, \theta') = (0.8, 0.2)$. For the harder case of $(\theta, \theta') = (0.7, 0.3)$, the performances of the UU methods drop slightly, but are still comparable to the small PN.

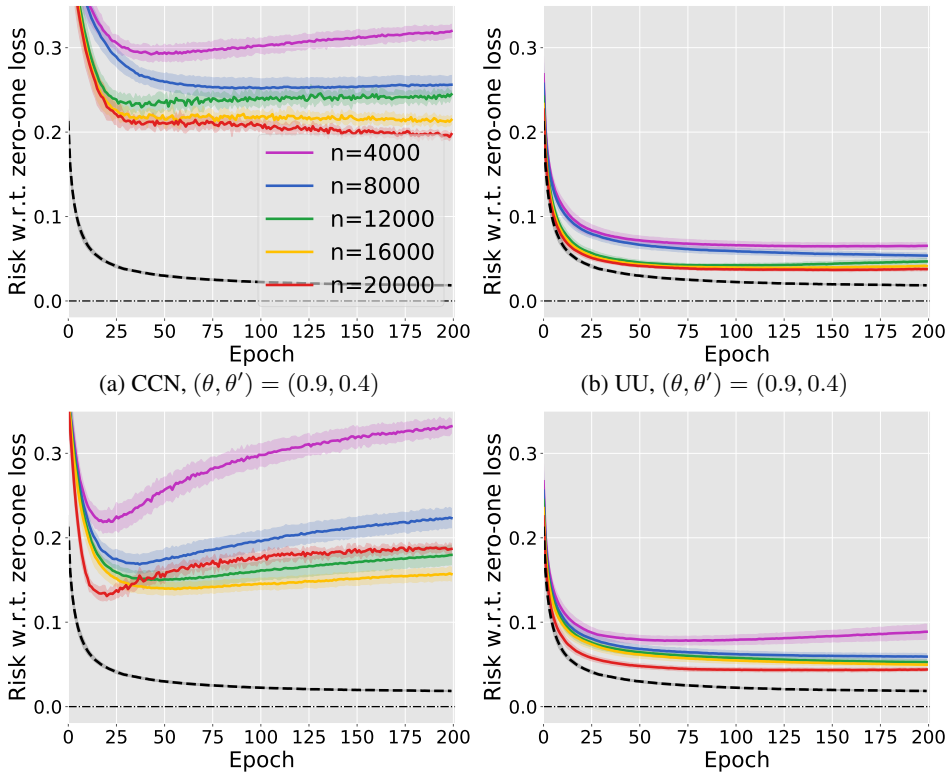
Analysis of changing sample sizes of two U sets We also investigated the issue of covariate shift by changing the sample sizes of two U sets. We fixed n' to 20000 and gradually moved n from 4000 to 20000, and tested the performance of UU and CCN on two different pairs of training class priors: $\{(0.9, 0.4), (0.8, 0.4)\}$. Each experiment was conducted 10 times and we reported the means and standard deviations of test risks in Figure 5. The results show that as n moves farther from n', the proposed unbiased UU method works reasonably well, while the performance CCN drops severely. It is because the larger covariate shift when n moves farther from n', which makes the CCN-based risk estimator more biased. Due to the unbiasedness of the proposed risk estimator, UU classifiers can handle the covariate shift reasonably well. The experimental results are consistent with analysis of moving θ and θ' closer in Sec. 5.1.



(a) $(\theta, \theta') = (0.9, 0.1), \ell_{\mathrm{sig}}$ (b) $(\theta, \theta') = (0.8, 0.2), \ell_{\mathrm{sig}}$ (c) $(\theta, \theta') = (0.7, 0.3), \ell_{\mathrm{sig}}$

(d) $(\theta, \theta') = (0.9, 0.1), \ell_{\mathrm{log}}$ (e) $(\theta, \theta') = (0.8, 0.2), \ell_{\mathrm{log}}$ (f) $(\theta, \theta') = (0.7, 0.3), \ell_{\mathrm{log}}$

Figure 4: Experimental results of training deep neural networks by Eq. (10) with ℓ_{sig} and Eq. (12) with ℓ_{log}.



(a) CCN, $(\theta, \theta') = (0.9, 0.4)$ (b) UU, $(\theta, \theta') = (0.9, 0.4)$

(c) CCN, $(\theta, \theta') = (0.8, 0.4)$ (d) UU, $(\theta, \theta') = (0.8, 0.4)$

Figure 5: Illustration of how changing sample sizes of two U sets n and n' affects the classification performance of CCN and the proposed UU method. PN oracle is illustrated in black dashed line.