Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels

Curtis G. Northcutt, * **Tailin Wu,** * **Isaac L. Chuang**
Massachusetts Institute of Technology
Cambridge, MA 02139
{cgn, tailin, ichuang}@mit.edu

Abstract

$\tilde{P}\tilde{N}$ learning is the problem of binary classification when training examples may be mislabeled (flipped) uniformly with noise rate ρ_1 for positive examples and ρ_0 for negative examples. We propose Rank Pruning (RP) to solve $\tilde{P}\tilde{N}$ learning and the open problem of estimating the noise rates, i.e. the fraction of wrong positive and negative labels. Unlike prior solutions, RP is time-efficient and general, requiring $\mathcal{O}(T)$ for any unrestricted choice of probabilistic classifier with T fitting time. We prove RP has consistent noise estimation and equivalent expected risk as learning with uncorrupted labels in ideal conditions, and derive closed-form solutions when conditions are non-ideal. RP achieves state-of-the-art noise estimation and F1, error, and AUC-PR for both MNIST and CIFAR datasets, regardless of the amount of noise and performs similarly impressively when a large portion of training examples are noise drawn from a third distribution. To highlight, RP with a CNN classifier can predict if an MNIST digit is a *one* or *not* with only 0.25% error, and 0.46% error across all digits, even when 50% of positive examples are mislabeled and 50% of observed positive labels are mislabeled negative examples.

1 Introduction

Consider a student with no knowledge of animals tasked with learning to classify whether a picture contains a dog. A teacher shows the student example pictures of lone four-legged animals, stating whether the image contains a dog or not. Unfortunately, the teacher may often make mistakes, asymmetrically, with a significantly large false positive rate, $\rho_1 \in [0, 1]$, and significantly large false negative rate, $\rho_0 \in [0, 1]$. The teacher may also include "white noise" images with a uniformly random label. This information is unknown to the student, who only knows of the images and corrupted labels, but suspects that the teacher may make mistakes. Can the student (1) estimate the mistake rates, ρ_1 and ρ_0, (2) learn to classify pictures with dogs accurately, and (3) do so efficiently (e.g. less than an hour for 50 images)? This allegory clarifies the challenges of $\tilde{P}\tilde{N}$ learning for any classifier trained with corrupted labels, perhaps with intermixed noise examples. We elect the notation $\tilde{P}\tilde{N}$ to emphasize that both the positive and negative sets may contain mislabeled examples, reserving P and N for uncorrupted sets.

This example illustrates a fundamental reliance of supervised learning on training labels (Michalski et al., 1986). Traditional learning performance degrades monotonically with label noise (Aha et al., 1991; Nettleton et al., 2010), necessitating semi-supervised approaches (Blanchard et al., 2010). Examples of noisy datasets are medical (Raviv & Intrator, 1996), human-labeled (Paolacci et al., 2010), and sensor (Lane et al., 2010) datasets. The problem of uncovering the same classifications as if the data was not mislabeled is our fundamental goal.

Towards this goal, we introduce Rank Pruning[2], an algorithm for $\tilde{P}\tilde{N}$ learning composed of two sequential parts: (1) estimation of the asymmetric noise rates ρ_1 and ρ_0 and (2) removal of mislabeled examples prior to training. The

* Equal Contribution
[2] Rank Pruning is open-source and available at https://github.com/cgnorthcutt/rankpruning

Table 1: Variable definitions and descriptions for $\tilde{P}\tilde{N}$ learning and PU learning. Related work contains a prominent author using each variable. ρ_1 is also referred to as *contamination* in PU learning literature.

VARIABLE	CONDITIONAL	DESCRIPTION	DOMAIN	RELATED WORK
ρ_0	$P(s=1\|y=0)$	FRACTION OF N EXAMPLES MISLABELED AS POSITIVE	$\tilde{P}\tilde{N}$	LIU
ρ_1	$P(s=0\|y=1)$	FRACTION OF P EXAMPLES MISLABELED AS NEGATIVE	$\tilde{P}\tilde{N}$, PU	LIU, CLAESEN
π_0	$P(y=1\|s=0)$	FRACTION OF MISLABELED EXAMPLES IN $\tilde{N}$	$\tilde{P}\tilde{N}$	SCOTT
π_1	$P(y=0\|s=1)$	FRACTION OF MISLABELED EXAMPLES IN $\tilde{P}$	$\tilde{P}\tilde{N}$	SCOTT
$c=1-\rho_1$	$P(s=1\|y=1)$	FRACTION OF CORRECTLY LABELED P IF $P(y=1\|s=1)=1$	PU	ELKAN

fundamental mantra of Rank Pruning is *learning with confident examples*, i.e. examples with a predicted probability of being positive *near* 1 when the label is positive or 0 when the label is negative. If we imagine non-confident examples as a noise class, separate from the confident positive and negative classes, then their removal should unveil a subset of the uncorrupted data.

An ancillary mantra of Rank Pruning is *removal by rank* which elegantly exploits ranking without sorting. Instead of pruning non-confident examples by predicted probability, we estimate the number of mislabeled examples in each class. We then remove the k^{th}-most or k^{th}-least examples, *ranked* by predicted probability, via the BFPRT algorithm (Blum et al., 1973) in $\mathcal{O}(n)$ time, where n is the number of training examples. *Removal by rank* mitigates sensitivity to probability estimation and exploits the reduced complexity of learning to rank over probability estimation (Menon et al., 2012). Together, *learning with confident examples* and *removal by rank* enable robustness, i.e. invariance to erroneous input deviation.

Beyond prediction, confident examples help estimate ρ_1 and ρ_0. Typical approaches require averaging predicted probabilities on a holdout set (Liu & Tao, 2016; Elkan & Noto, 2008) tying noise estimation to the accuracy of the predicted probabilities, which in practice may be confounded by added noise or poor model selection. Instead, we estimate ρ_1 and ρ_0 as a fraction of the predicted counts of confident examples in each class, encouraging robustness for variation in probability estimation.

1.1 Related Work

Rank Pruning bridges framework, nomenclature, and application across PU and $\tilde{P}\tilde{N}$ learning. In this section, we consider the contributions of Rank Pruning in both.

1.1.1 PU Learning

Positive-unlabeled (PU) learning is a binary classification task in which a subset of positive training examples are labeled, and the rest are unlabeled. For example, co-training (Blum & Mitchell, 1998; Nigam & Ghani, 2000) with labeled and unlabeled examples can be framed as a PU learning problem by assigning all unlabeled examples the label '0'. PU learning methods often assume corrupted negative labels for the unlabeled examples U such that PU learning is $\tilde{P}\tilde{N}$ learning with no mislabeled examples in P, hence their naming conventions.

Early approaches to PU learning modified the loss functions via weighted logistic regression (Lee & Liu, 2003) and biased SVM (Liu et al., 2003) to penalize more when positive examples are predicted incorrectly. Bagging SVM (Mordelet & Vert, 2014) and RESVM (Claesen et al., 2015) extended biased SVM to instead use an ensemble of classifiers trained by resampling U (and P for RESVM) to improve robustness (Breiman, 1996). RESVM claims state-of-the-art for PU learning, but is impractically inefficient for large datasets because it requires optimization of five parameters and suffers from the pitfalls of SVM model selection (Chapelle & Vapnik, 1999). Elkan & Noto (2008) introduce a formative time-efficient probabilistic approach (denoted *Elk08*) for PU learning that directly estimates $1-\rho_1$ by averaging predicted probabilities of a holdout set and dividing all predicted probabilities by $1-\rho_1$. On the SwissProt database, *Elk08* was 621 times faster than biased SVM, which only requires two parameter optimization. However, *Elk08* noise rate estimation is sensitive to inexact probability estimation and both RESVM and *Elk08* assume $P = \tilde{P}$ and do not generalize to $\tilde{P}\tilde{N}$ learning. Rank Pruning leverages *Elk08* to initialize ρ_1, but then re-estimates ρ_1 using confident examples for both robustness (RESVM) and efficiency (*Elk08*).

1.1.2 $\tilde{P}\tilde{N}$ Learning

Theoretical approaches for $\tilde{P}\tilde{N}$ learning often have two steps: (1) estimate the noise rates, ρ_1, ρ_0, and (2) use ρ_1, ρ_0 for prediction. To our knowledge, Rank Pruning is the only time-efficient solution for the open problem (Liu & Tao, 2016; Yang et al., 2012) of noise estimation.

We first consider relevant work in noise rate estimation. Scott et al. (2013) established a lower bound method for estimating the *inversed* noise rates π_1 and π_0 (defined in Table 1). However, the method can be intractable due to unbounded convergence and assumes that the positive and negative distributions are mutually irreducible. Under additional assumptions, Scott (2015) proposed a time-efficient method for noise rate estimation, but reported poor performance Liu & Tao (2016). Liu & Tao (2016) used the minimum predicted probabilities as the noise rates, which often yields futile estimates of min = 0. Natarajan et al. (2013) provide no method for estimation and view the noise rates as parameters optimized with cross-validation, inducing a sacrificial accuracy, efficiency trade-off. In comparison, Rank Pruning noise rate estimation is time-efficient, consistent in ideal conditions, and robust to imperfect probability estimation.

Natarajan et al. (2013) developed two methods for prediction in the $\tilde{P}\tilde{N}$ setting which modify the loss function. The first method constructs an unbiased estimator of the loss function for the true distribution from the noisy distribution, but the estimator may be non-convex even if the original loss function is convex. If the classifier's loss function cannot be modified directly, this method requires splitting each example in two with class-conditional weights and ensuring split examples are in the same batch during optimization. For these reasons, we instead compare Rank Pruning with their second method (*Nat13*), which constructs a label-dependent loss function such that for 0-1 loss, the minimizers of *Nat13*'s risk and the risk for the true distribution are equivalent.

Liu & Tao (2016) generalized *Elk08* to the $\tilde{P}\tilde{N}$ learning setting by modifying the loss function with per-example importance reweighting (*Liu16*), but reweighting terms are derived from predicted probabilities which may be sensitive to inexact estimation. To mitigate sensitivity, Liu & Tao (2016) examine the use of density ratio estimation (Sugiyama et al., 2012). Instead, Rank Pruning mitigates sensitivity by learning from confident examples selected by rank order, not predicted probability. For fairness of comparison across methods, we compare Rank Pruning with their probability-based approach.

Assuming perfect estimation of ρ_1 and ρ_0, we, Natarajan et al. (2013), and Liu & Tao (2016) all prove that the expected risk for the modified loss function is equivalent to the expected risk for the perfectly labeled dataset. However, both Natarajan et al. (2013) and Liu & Tao (2016) effectively "flip" example labels in the construction of their loss function, providing no benefit for added random noise. In comparison, Rank Pruning will also remove added random noise because noise drawn from a third distribution is unlikely to appear confidently positive or negative. Table 2 summarizes our comparison of $\tilde{P}\tilde{N}$ and PU learning methods.

Procedural efforts have improved robustness to mislabeling in the context of machine vision (Xiao et al., 2015), neural networks (Reed et al., 2015), and face recognition (Angelova et al., 2005). Though promising, these methods are restricted in theoretical justification and generality, motivating the need for Rank Pruning.

Table 2: Summary of state-of-the-art and selected general solutions to $\tilde{P}\tilde{N}$ and PU learning.

Related Work	Noise Estim.	$\tilde{P}\tilde{N}$	PU	Any Prob. Classifier	Prob Estim. Robustness	Time Efficient	Theory Support	Added Noise
Elkan & Noto (2008)	✓		✓	✓		✓	✓	
Claesen et al. (2015)			✓		✓			
Scott et al. (2013)	✓			✓	✓		✓	
Natarajan et al. (2013)		✓	✓	✓	✓	✓	✓	
Liu & Tao (2016)		✓	✓	✓		✓	✓	
Rank Pruning	✓	✓	✓	✓	✓	✓	✓	✓

1.2 Contributions

In this paper, we describe the Rank Pruning algorithm for binary classification with imperfectly labeled training data. In particular, we:

- Develop a robust, time-efficient, general solution for both $\tilde{P}\tilde{N}$ learning, i.e. binary classification with noisy labels, and estimation of the fraction of mislabeling in both the positive and negative training sets.
- Introduce the *learning with confident examples* mantra as a new way to think about robust classification and estimation with mislabeled training data.
- Prove that under assumptions, Rank Pruning achieves perfect noise estimation and equivalent expected risk as learning with correct labels. We provide closed-form solutions when those assumptions are relaxed.
- Demonstrate that Rank Pruning performance generalizes across the number of training examples, feature dimension, fraction of mislabeling, and fraction of added noise examples drawn from a third distribution.
- Improve the state-of-the-art of $\tilde{P}\tilde{N}$ learning across F1 score, AUC-PR, and Error. In many cases, Rank Pruning achieves nearly the same F1 score as learning with correct labels when 50% of positive examples are mislabeled and 50% of observed positive labels are mislabeled negative examples.

2 Framing the $\tilde{P}\tilde{N}$ Learning Problem

In this section, we formalize the foundational definitions, assumptions, and goals of the $\tilde{P}\tilde{N}$ learning problem illustrated by the student-teacher motivational example.

Given n observed training examples $x \in \mathcal{R}^D$ with associated observed corrupted labels $s \in \{0,1\}$ and unobserved true labels $y \in \{0,1\}$, we seek a binary classifier f that estimates the mapping $x \to y$. Unfortunately, if we fit the classifier using observed (x, s) pairs, we estimate the mapping $x \to s$ and obtain $g(x) = P(\hat{s} = 1|x)$.

We define the observed noisy positive and negative sets as $\tilde{P} = \{x|s = 1\}, \tilde{N} = \{x|s = 0\}$ and the unobserved true positive and negative sets as $P = \{x|y = 1\}, N = \{x|y = 0\}$. Define the hidden training data as $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, drawn i.i.d. from some true distribution $\mathcal{D}$. We assume that a class-conditional Classification Noise Process (CNP) (Angluin & Laird, 1988) maps y true labels to s observed labels such that each label in P is flipped independently with probability ρ_1 and each label in N is flipped independently with probability ρ_0 ($s \leftarrow CNP(y, \rho_1, \rho_0)$). The resulting observed, corrupted dataset is $D_\rho = \{(x_1, s_1), (x_2, s_2), ..., (x_n, s_n)\}$. Therefore, $(s \perp\!\!\!\perp x)|y$ and $P(s = s|y = y, x) = P(s = s|y = y)$. In recent work, CNP is referred to as the random noise classification (RCN) noise model (Liu & Tao, 2016; Natarajan et al., 2013).

The noise rate $\rho_1 = P(s = 0|y = 1)$ is the fraction of P examples mislabeled as negative and the noise rate $\rho_0 = P(s = 1|y = 0)$ is the fraction of N examples mislabeled as positive. Note that $\rho_1 + \rho_0 < 1$ is a necessary condition, otherwise more examples would be mislabeled than labeled correctly. Thus, $\rho_0 < 1 - \rho_1$. We elect a subscript of "0" to refer to the negative set and a subscript of "1" to refer to the positive set. Additionally, let $p_{s1} = P(s = 1)$ be the fraction of corrupted labels that are positive and $p_{y1} = P(y = 1)$ be the fraction of true labels that are positive. It follows that the inversed noise rates are $\pi_1 = P(y = 0|s = 1) = \frac{\rho_0(1-p_{y1})}{p_{s1}}$ and $\pi_0 = P(y = 1|s = 0) = \frac{\rho_1 p_{y1}}{(1-p_{s1})}$. Combining these relations, given any pair in $\{(\rho_0, \rho_1), (\rho_1, \pi_1), (\rho_0, \pi_0), (\pi_0, \pi_1)\}$, the remaining two and p_{y1} are known.

We consider five levels of assumptions for P, N, and g:
Perfect Condition: g is a "perfect" probability estimator iff $g(x) = g^*(x)$ where $g^*(x) = P(s = 1|x)$. Equivalently, let $g(x) = P(s = 1|x) + \Delta g(x)$. Then $g(x)$ is "perfect" when $\Delta g(x) = 0$ and "imperfect" when $\Delta g(x) \neq 0$. g may be imperfect due to the method of estimation or due to added uniformly randomly labeled examples drawn from a third noise distribution.
Non-overlapping Condition: P and N have "non-overlapping support" if $P(y = 1|x) = \mathbb{1}[[y = 1]]$, where the indicator function $\mathbb{1}[[a]]$ is 1 if the a is true, else 0.
Ideal Condition[1]: g is "ideal" when both perfect and non-overlapping conditions hold and $(s \perp\!\!\!\perp x)|y$ such that

[1] Eq. (1) is first derived in (Elkan & Noto, 2008) .

$$\begin{aligned} g(x) =& g^*(x) = P(s=1|x) \\ =& P(s=1|y=1,x) \cdot P(y=1|x) + P(s=1|y=0,x) \cdot P(y=0|x) \\ =& (1-\rho_1) \cdot \mathbb{1}[[y=1]] + \rho_0 \cdot \mathbb{1}[[y=0]] \end{aligned} \tag{1}$$

Range Separability Condition g range separates P and N iff $\forall x_1 \in P$ and $\forall x_2 \in N$, we have $g(x_1) > g(x_2)$.
Unassuming Condition: g is "unassuming" when perfect and/or non-overlapping conditions may not be true.

Their relationship is: **Unassuming** $\supset$ **Range Separability** $\supset$ **Ideal** $=$ **Perfect** $\cap$ **Non-overlapping**.

We can now state the two goals of Rank Pruning for $\tilde{P}\tilde{N}$ learning. **Goal 1** is to perfectly estimate $\hat{\rho}_1 \triangleq \rho_1$ and $\hat{\rho}_0 \triangleq \rho_0$ when g is ideal. When g is not ideal, to our knowledge perfect estimation of ρ_1 and ρ_0 is impossible and at best **Goal 1** is to provide exact expressions for $\hat{\rho}_1$ and $\hat{\rho}_0$ w.r.t. ρ_1 and ρ_0. **Goal 2** is to use $\hat{\rho}_1$ and $\hat{\rho}_0$ to uncover the classifications of f from g. Both tasks must be accomplished given only observed (x, s) pairs. y, ρ_1, ρ_0, π_1, and π_0 are hidden.

3 Rank Pruning

We develop the Rank Pruning algorithm to address our two goals. In Section 3.1, we propose a method for noise rate estimation and prove consistency when g is ideal. An estimator is "consistent" if it achieves perfect estimation in the expectation of infinite examples. In Section 3.2, we derive exact expressions for $\hat{\rho}_1$ and $\hat{\rho}_0$ when g is unassuming. In Section 3.3, we provide the entire algorithm, and in Section 3.5, prove that Rank Pruning has equivalent expected risk as learning with uncorrupted labels for both ideal g and non-ideal g with weaker assumptions. Throughout, we assume $n \to \infty$ so that P and N are the hidden distributions, each with infinite examples. This is a necessary condition for Theorems. 2, 4 and Lemmas 1, 3.

3.1 Deriving Noise Rate Estimators $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$

We propose the *confident counts* estimators $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ to estimate ρ_1 and ρ_0 as a fraction of the predicted counts of confident examples in each class, encouraging robustness for variation in probability estimation. To estimate $\rho_1 = P(s=0|y=1)$, we count the number of examples with label $s=0$ that we are "confident" have label $y=1$ and divide it by the total number of examples that we are "confident" have label $y=1$. More formally,

$$\hat{\rho}_1^{conf} := \frac{|\tilde{N}_{y=1}|}{|\tilde{N}_{y=1}| + |\tilde{P}_{y=1}|}, \qquad \hat{\rho}_0^{conf} := \frac{|\tilde{P}_{y=0}|}{|\tilde{P}_{y=0}| + |\tilde{N}_{y=0}|} \tag{2}$$

such that

$$\begin{cases} \tilde{P}_{y=1} = \{x \in \tilde{P} \mid g(x) \geq LB_{y=1}\} \\ \tilde{N}_{y=1} = \{x \in \tilde{N} \mid g(x) \geq LB_{y=1}\} \\ \tilde{P}_{y=0} = \{x \in \tilde{P} \mid g(x) \leq UB_{y=0}\} \\ \tilde{N}_{y=0} = \{x \in \tilde{N} \mid g(x) \leq UB_{y=0}\} \end{cases} \tag{3}$$

where g is fit to the corrupted training set D_ρ to obtain $g(x) = P(\hat{s}=1|x)$. The threshold $LB_{y=1}$ is the predicted probability in $g(x)$ above which we guess that an example x has hidden label $y=1$, and similarly for upper bound $UB_{y=0}$. $LB_{y=1}$ and $UB_{y=0}$ partition $\tilde{P}$ and $\tilde{N}$ into four sets representing a *best guess* of a *subset* of examples having labels (1) $s=1, y=0$, (2) $s=1, y=1$, (3) $s=0, y=0$, (4) $s=0, y=1$. The threshold values are defined as

$$\begin{cases} LB_{y=1} := P(\hat{s}=1 \mid s=1) = E_{x \in \tilde{P}}[g(x)] \\ UB_{y=0} := P(\hat{s}=1 \mid s=0) = E_{x \in \tilde{N}}[g(x)] \end{cases}$$

where $\hat{s}$ is the predicted label from a classifier fit to the observed data. $|\tilde{P}_{y=1}|$ counts examples with label $s=1$ that are *most* likely to be correctly labeled ($y=1$) because $LB_{y=1} = P(\hat{s}=1|s=1)$. The three other terms in Eq. (3) follow similar reasoning. Importantly, the four terms do not sum to n, i.e. $|N| + |P|$, but $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ are valid estimates because mislabeling noise is assumed to be uniformly random. The choice of threshold values relies on the following two important equations:

$$\begin{aligned} LB_{y=1} =& E_{x\in\tilde{P}}[g(x)] = E_{x\in\tilde{P}}[P(s=1|x)] \\ =& E_{x\in\tilde{P}}[P(s=1|x,y=1)P(y=1|x) + P(s=1|x,y=0)P(y=0|x)] \\ =& E_{x\in\tilde{P}}[P(s=1|y=1)P(y=1|x) + P(s=1|y=0)P(y=0|x)] \\ =& (1-\rho_1)(1-\pi_1) + \rho_0\pi_1 \end{aligned} \tag{4}$$

Similarly, we have

$$UB_{y=0} = (1-\rho_1)\pi_0 + \rho_0(1-\pi_0) \tag{5}$$

To our knowledge, although simple, this is the first time that the relationship in Eq. (4) (5) has been published, linking the work of Elkan & Noto (2008), Liu & Tao (2016), Scott et al. (2013) and Natarajan et al. (2013). From Eq. (4) (5), we observe that $LB_{y=1}$ and $UB_{y=0}$ are linear interpolations of $1-\rho_1$ and ρ_0 and since $\rho_0 < 1-\rho_1$, we have that $\rho_0 < LB_{y=1} \leq 1-\rho_1$ and $\rho_0 \leq UB_{y=0} < 1-\rho_1$. When g is ideal we have that $g(x) = (1-\rho_1)$, if $x \in P$ and $g(x) = \rho_0$, if $x \in N$. Thus when g is ideal, the thresholds $LB_{y=1}$ and $UB_{y=0}$ in Eq. (3) will perfectly separate P and N examples within each of $\tilde{P}$ and $\tilde{N}$. Lemma 1 immediately follows.

Lemma 1 *When g is ideal,*

$$\begin{aligned} &\tilde{P}_{y=1} = \{x \in P \mid s=1\}, \tilde{N}_{y=1} = \{x \in P \mid s=0\}, \\ &\tilde{P}_{y=0} = \{x \in N \mid s=1\}, \tilde{N}_{y=0} = \{x \in N \mid s=0\} \end{aligned} \tag{6}$$

Thus, when g is ideal, the thresholds in Eq. (3) partition the training set such that $\tilde{P}_{y=1}$ and $\tilde{N}_{y=0}$ contain the correctly labeled examples and $\tilde{P}_{y=0}$ and $\tilde{N}_{y=1}$ contain the mislabeled examples. Theorem 2 follows (for brevity, proofs of all theorems/lemmas are in Appendix A.1-A.5).

Theorem 2 *When g is ideal,*

$$\hat{\rho}_1^{conf} = \rho_1, \hat{\rho}_0^{conf} = \rho_0 \tag{7}$$

Thus, when g is ideal, the *confident counts* estimators $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ are consistent estimators for ρ_1 and ρ_0 and we set $\hat{\rho}_1 := \hat{\rho}_1^{conf}, \hat{\rho}_0 := \hat{\rho}_0^{conf}$. These steps comprise Rank Pruning noise rate estimation (see Alg. 1). There are two practical observations. First, for any g with T fitting time, computing $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ is $\mathcal{O}(T)$. Second, $\hat{\rho}_1$ and $\hat{\rho}_0$ should be estimated out-of-sample to avoid over-fitting, resulting in sample variations. In our experiments, we use 3-fold cross-validation, requiring at most $2T = \mathcal{O}(T)$.

3.2 Noise Estimation: Unassuming Case

Theorem 2 states that $\hat{\rho}_i^{conf} = \rho_i, \forall i \in \{0,1\}$ when g is ideal. Though theoretically constructive, in practice this is unlikely. Next, we derive expressions for the estimators when g is unassuming, i.e. g may not be perfect and P and N may have overlapping support.

Define $\Delta p_o := \frac{|P \cap N|}{|P \cup N|}$ as the fraction of overlapping examples in $\mathcal{D}$ and remember that $\Delta g(x) := g(x) - g^*(x)$. Denote $LB^*_{y=1} = (1-\rho_1)(1-\pi_1) + \rho_0\pi_1, UB^*_{y=0} = (1-\rho_1)\pi_0 + \rho_0(1-\pi_0)$. We have

Lemma 3 *When g is unassuming, we have*

$$\begin{cases} LB_{y=1} = LB^*_{y=1} + E_{x\in\tilde{P}}[\Delta g(x)] - \frac{(1-\rho_1-\rho_0)^2}{p_{s1}}\Delta p_o \\ UB_{y=0} = UB^*_{y=0} + E_{x\in\tilde{N}}[\Delta g(x)] + \frac{(1-\rho_1-\rho_0)^2}{1-p_{s1}}\Delta p_o \\ \hat{\rho}_1^{conf} = \rho_1 + \frac{1-\rho_1-\rho_0}{|P|-|\Delta P_1|+|\Delta N_1|}|\Delta N_1| \\ \hat{\rho}_0^{conf} = \rho_0 + \frac{1-\rho_1-\rho_0}{|N|-|\Delta N_0|+|\Delta P_0|}|\Delta P_0| \end{cases} \tag{8}$$

where

$$\begin{cases} \Delta P_1 = \{x \in P \mid g(x) < LB_{y=1}\} \\ \Delta N_1 = \{x \in N \mid g(x) \geq LB_{y=1}\} \\ \Delta P_0 = \{x \in P \mid g(x) \leq UB_{y=0}\} \\ \Delta N_0 = \{x \in N \mid g(x) > UB_{y=0}\} \end{cases}$$

The second term on the R.H.S. of the $\hat{\rho}_i^{conf}$ expressions captures the deviation of $\hat{\rho}_i^{conf}$ from ρ_i, $i = 0, 1$. This term results from both imperfect $g(x)$ and overlapping support. Because the term is non-negative, $\hat{\rho}_i^{conf} \geq \rho_i$, $i = 0, 1$ in the limit of infinite examples. In other words, $\hat{\rho}_i^{conf}$ is an *upper bound* for the noise rates ρ_i, $i = 0, 1$. From Lemma 3, it also follows:

Theorem 4 *Given non-overlapping support condition,*

If $\forall x \in N, \Delta g(x) < LB_{y=1} - \rho_0$, *then* $\hat{\rho}_1^{conf} = \rho_1$.

If $\forall x \in P, \Delta g(x) > -(1 - \rho_1 - UB_{y=0})$, *then* $\hat{\rho}_0^{conf} = \rho_0$.

Theorem 4 shows that $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ are robust to imperfect probability estimation. As long as $\Delta g(x)$ does not exceed the distance between the threshold in Eq. (3) and the perfect $g^*(x)$ value, $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ are consistent estimators for ρ_1 and ρ_0. Our numerical experiments in Section 4 suggest this is reasonable for $\Delta g(x)$. The average $|\Delta g(x)|$ for the MNIST training dataset across different (ρ_1, π_1) varies between 0.01 and 0.08 for a logistic regression classifier, 0.01∼0.03 for a CNN classifier, and 0.05∼0.10 for the CIFAR dataset with a CNN classifier. Thus, when $LB_{y=1} - \rho_0$ and $1 - \rho_1 - UB_{y=0}$ are above 0.1 for these datasets, from Theorem 4 we see that $\hat{\rho}_i^{conf}$ still accurately estimates ρ_i.

3.3 The Rank Pruning Algorithm

Using $\hat{\rho}_1$ and $\hat{\rho}_0$, we must uncover the classifications of f from g. In this section, we describe how Rank Pruning selects confident examples, removes the rest, and trains on the pruned set using a reweighted loss function.

First, we obtain the inverse noise rates $\hat{\pi}_1$, $\hat{\pi}_0$ from $\hat{\rho}_1$, $\hat{\rho}_0$:

$$\hat{\pi}_1 = \frac{\hat{\rho}_0}{p_{s1}} \frac{1 - p_{s1} - \hat{\rho}_1}{1 - \hat{\rho}_1 - \hat{\rho}_0}, \qquad \hat{\pi}_0 = \frac{\hat{\rho}_1}{1 - p_{s1}} \frac{p_{s1} - \hat{\rho}_0}{1 - \hat{\rho}_1 - \hat{\rho}_0} \tag{9}$$

Next, we prune the $\hat{\pi}_1|\tilde{P}|$ examples in $\tilde{P}$ with smallest $g(x)$ and the $\hat{\pi}_0|\tilde{N}|$ examples in $\tilde{N}$ with highest $g(x)$ and denote the pruned sets $\tilde{P}_{conf}$ and $\tilde{N}_{conf}$. To prune, we define k_1 as the $(\hat{\pi}_1|\tilde{P}|)^{th}$ smallest $g(x)$ for $x \in \tilde{P}$ and k_0 as the $(\hat{\pi}_0|\tilde{N}|)^{th}$ largest $g(x)$ for $x \in \tilde{N}$. BFPRT ($\mathcal{O}(n)$) (Blum et al., 1973) is used to compute k_1 and k_0 and pruning is reduced to the following $\mathcal{O}(n)$ filter:

$$\tilde{P}_{conf} := \{x \in \tilde{P} \mid g(x) \geq k_1\}, \qquad \tilde{N}_{conf} := \{x \in \tilde{N} \mid g(x) \leq k_0\} \tag{10}$$

Lastly, we refit the classifier to $X_{conf} = \tilde{P}_{conf} \cup \tilde{N}_{conf}$ by class-conditionally reweighting the loss function for examples in $\tilde{P}_{conf}$ with weight $\frac{1}{1-\hat{\rho}_1}$ and examples in $\tilde{N}_{conf}$ with weight $\frac{1}{1-\hat{\rho}_0}$ to recover the estimated balance of positive and negative examples. The entire Rank Pruning algorithm is presented in Alg. 1 and illustrated step-by-step on a synthetic dataset in Fig. 1.

We conclude this section with a formal discussion of the loss function and efficiency of Rank Pruning. Define $\hat{y}_i$ as the predicted label of example i for the classifier fit to X_{conf}, s_{conf} and let $l(\hat{y}_i, s_i)$ be the original loss function for $x_i \in D_\rho$. Then the loss function for Rank Pruning is simply the original loss function exerted on the pruned X_{conf}, with class-conditional weighting:

$$\tilde{l}(\hat{y}_i, s_i) = \frac{1}{1 - \hat{\rho}_1} l(\hat{y}_i, s_i) \cdot \mathbb{1}[[x_i \in \tilde{P}_{conf}]] + \frac{1}{1 - \hat{\rho}_0} l(\hat{y}_i, s_i) \cdot \mathbb{1}[[x_i \in \tilde{N}_{conf}]] \tag{11}$$

Effectively this loss function uses a zero-weight for pruned examples. Other than potentially fewer examples, the only difference in the loss function for Rank Pruning and the original loss function is the class-conditional weights. These

Algorithm 1 Rank Pruning

Input: Examples X, corrupted labels s, classifier clf

Part 1. Estimating Noise Rates:

(1.1) clf.fit(X,s)

$g(x) \leftarrow$clf.predict_crossval_probability($\hat{s}=1|x$)

$p_{s1} = \frac{\text{count}(s=1)}{\text{count}(s=0 \vee s=1)}$

$LB_{y=1} = E_{x \in \tilde{P}}[g(x)]$, $UB_{y=0} = E_{x \in \tilde{N}}[g(x)]$

(1.2) $\hat{\rho}_1 = \hat{\rho}_1^{conf} = \frac{|\tilde{N}_{y=1}|}{|\tilde{N}_{y=1}|+|\tilde{P}_{y=1}|}$, $\hat{\rho}_0 = \hat{\rho}_0^{conf} = \frac{|\tilde{P}_{y=0}|}{|\tilde{P}_{y=0}|+|\tilde{N}_{y=0}|}$

$\hat{\pi}_1 = \frac{\hat{\rho}_0}{p_{s1}} \frac{1-p_{s1}-\hat{\rho}_1}{1-\hat{\rho}_1-\hat{\rho}_0}$, $\hat{\pi}_0 = \frac{\hat{\rho}_1}{1-p_{s1}} \frac{p_{s1}-\hat{\rho}_0}{1-\hat{\rho}_1-\hat{\rho}_0}$

Part 2. Prune Inconsistent Examples:

(2.1) Remove $\hat{\pi}_1|\tilde{P}|$ examples in $\tilde{P}$ with least $g(x)$, Remove $\hat{\pi}_0|\tilde{N}|$ examples in $\tilde{N}$ with greatest $g(x)$, Denote the remaining training set (X_{conf}, s_{conf})

(2.2) clf.fit(X_{conf}, s_{conf}), with sample weight $w(x) = \frac{1}{1-\hat{\rho}_1}\mathbb{1}[[s_{conf}=1]] + \frac{1}{1-\hat{\rho}_0}\mathbb{1}[[s_{conf}=0]]$

Output: clf

constant factors do not increase the complexity of the minimization of the original loss function. In other words, we can fairly report the running time of Rank Pruning in terms of the running time ($\mathcal{O}(T)$) of the choice of probabilistic estimator. Combining noise estimation ($\mathcal{O}(T)$), pruning ($\mathcal{O}(n)$), and the final fitting ($\mathcal{O}(T)$), Rank Pruning has a running time of $\mathcal{O}(T) + \mathcal{O}(n)$, which is $\mathcal{O}(T)$ for typical classifiers.

3.4 Rank Pruning: A simple summary

Recognizing that formalization can create obfuscation, in this section we describe the entire algorithm in a few sentences. Rank Pruning takes as input training examples X, noisy labels s, and a probabilistic classifier clf and finds a subset of X, s that is likely to be correctly labeled, i.e. a subset of X, y. To do this, we first find two thresholds, $LB_{y=1}$ and $UB_{y=0}$, to *confidently* guess the correctly and incorrectly labeled examples in each of $\tilde{P}$ and $\tilde{N}$, forming four sets, then use the set sizes to estimate the noise rates $\rho_1 = P(s=0|y=1)$ and $\rho_0 = P(s=1|y=0)$. We then use the noise rates to estimate the number of examples with observed label $s=1$ and hidden label $y=0$ and remove that number of examples from $\tilde{P}$ by removing those with lowest predicted probability $g(x)$. We prune $\tilde{N}$ similarly. Finally, the classifier is fit to the pruned set, which is intended to represent a subset of the correctly labeled data.

3.5 Expected Risk Evaluation

In this section, we prove Rank Pruning exactly uncovers the classifier f fit to hidden y labels when g range separates P and N and ρ_1 and ρ_0 are given.

Denote $f_\theta \in \mathcal{F} : x \to \hat{y}$ as a classifier's prediction function belonging to some function space $\mathcal{F}$, where θ represents the classifier's parameters. f_θ represents f, but without θ necessarily fit to the training data. $\hat{f}$ is the Rank Pruning estimate of f.

Denote the empirical risk of f_θ w.r.t. the loss function $\tilde{l}$ and corrupted data D_ρ as $\hat{R}_{\tilde{l},D_\rho}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n}\tilde{l}(f_\theta(x_i), s_i)$, and the expected risk of f_θ w.r.t. the corrupted distribution $\mathcal{D}_\rho$ as $R_{\tilde{l},\mathcal{D}_\rho}(f_\theta) = E_{(x,s)\sim\mathcal{D}_\rho}[\hat{R}_{\tilde{l},D_\rho}(f_\theta)]$. Similarly, denote $R_{l,\mathcal{D}}(f_\theta)$ as the expected risk of f_θ w.r.t. the hidden distribution $\mathcal{D}$ and loss function l. We show that using Rank Pruning, a classifier $\hat{f}$ can be learned for the hidden data D, given the corrupted data D_ρ, by minimizing the empirical risk:

$$\hat{f} = \underset{f_\theta \in \mathcal{F}}{\operatorname{argmin}} \hat{R}_{\tilde{l},D_\rho}(f_\theta) = \underset{f_\theta \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\tilde{l}(f_\theta(x_i), s_i) \tag{12}$$

Under the *range separability* condition, we have

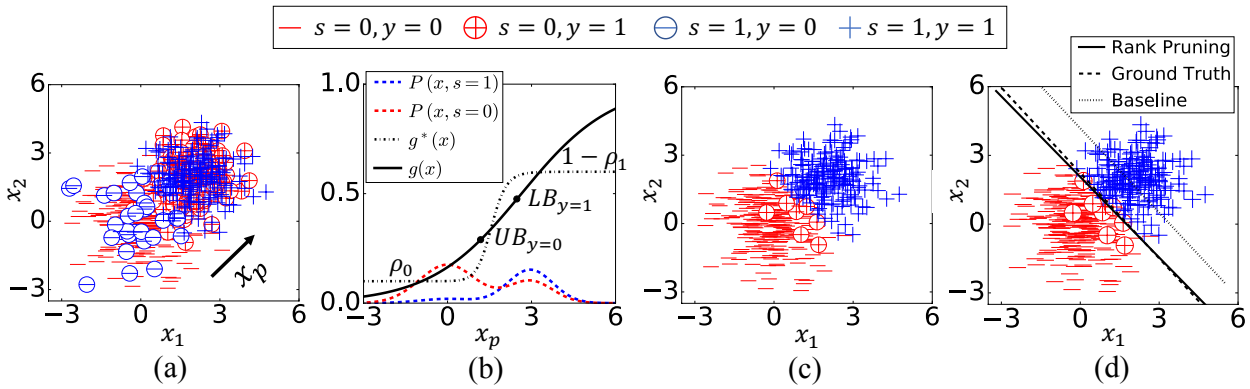

Figure 1: Illustration of Rank Pruning with a logistic regression classifier ($\mathcal{LR}_\theta$). **(a)**: The corrupted training set D_ρ with noise rates $\rho_1 = 0.4$ and $\rho_0 = 0.1$. Corrupted colored labels ($s = 1$,$s = 0$) are observed. y (+,−) is hidden. **(b)**: The marginal distribution of D_ρ projected onto the x_p axis (indicated in (a)), and the $\mathcal{LR}_\theta$'s estimated $g(x)$, from which $\hat{\rho}_1^{conf} = 0.4237$, $\hat{\rho}_0^{conf} = 0.1144$ are estimated. **(c)**: The pruned X_{conf}, s_{conf}. **(d)**: The classification result by Rank Pruning ($\hat{f} = \mathcal{LR}_\theta$.fit$(X_{conf}, s_{conf})$), ground truth classifier ($f = \mathcal{LR}_\theta$.fit(X, y)), and baseline classifier ($g = \mathcal{LR}_\theta$.fit(X, s)), with an accuracy of 94.16%, 94.16% and 78.83%, respectively.

Theorem 5 *If g range separates P and N and $\hat{\rho}_i = \rho_i$, $i = 0, 1$, then for any classifier f_θ and any bounded loss function $l(\hat{y}_i, y_i)$, we have*

$$R_{\tilde{l},\mathcal{D}_\rho}(f_\theta) = R_{l,\mathcal{D}}(f_\theta) \tag{13}$$

where $\tilde{l}(\hat{y}_i, s_i)$ is Rank Pruning's loss function (Eq. 11).

The proof of Theorem 5 is in Appendix A.5. Intuitively, Theorem 5 tells us that if g range separates P and N, then given exact noise rate estimates, Rank Pruning will exactly prune out the positive examples in $\tilde{N}$ and negative examples in $\tilde{P}$, leading to the same expected risk as learning from uncorrupted labels. Thus, Rank Pruning can exactly uncover the classifications of f (with infinite examples) because the expected risk is equivalent for any f_θ. Note Theorem 5 also holds when g is ideal, since *ideal* $\subset$ *range separability*. In practice, *range separability* encompasses a wide range of imperfect $g(x)$ scenarios, e.g. $g(x)$ can have large fluctuation in both P and N or have systematic drift w.r.t. to $g^*(x)$ due to underfitting.

4 Experimental Results

In Section 3, we developed a theoretical framework for Rank Pruning, proved exact noise estimation and equivalent expected risk when conditions are ideal, and derived closed-form solutions when conditions are non-ideal. Our theory suggests that, in practice, Rank Pruning should (1) accurately estimate ρ_1 and ρ_0, (2) typically achieve as good or better F1, error and AUC-PR (Davis & Goadrich, 2006) as state-of-the-art methods, and (3) be robust to both mislabeling and added noise.

In this section, we support these claims with an evaluation of the comparative performance of Rank Pruning in non-ideal conditions across thousands of scenarios. These include less complex (MNIST) and more complex (CIFAR) datasets, simple (logistic regression) and complex (CNN) classifiers, the range of noise rates, added random noise, separability of P and N, input dimension, and number of training examples to ensure that Rank Pruning is a general, agnostic solution for $\tilde{P}\tilde{N}$ learning.

In our experiments, we adjust π_1 instead of ρ_0 because binary noisy classification problems (e.g. detection and recognition tasks) often have that $|P| \ll |N|$. This choice allows us to adjust both noise rates with respect to P, i.e. the fraction of true positive examples that are mislabeled as negative (ρ_1) and the fraction of observed positive labels that are actually mislabeled negative examples (π_1). The $\tilde{P}\tilde{N}$ learning algorithms are trained with corrupted labels s, and tested on an unseen test set by comparing predictions $\hat{y}$ with the true test labels y using F1 score, error, and

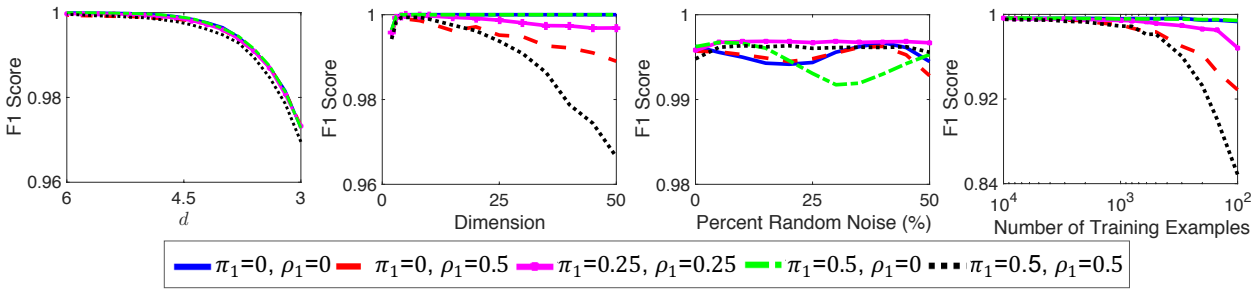

Figure 2: Comparison of Rank Pruning with different noise ratios (π_1, ρ_1) on a synthetic dataset for varying separability d, dimension, added random noise and number of training examples. Default settings for Fig. 2, 3 and 4: $d = 4$, 2-dimension, 0% random noise, and 5000 training examples with $p_{y1} = 0.2$. The lines are an average of 200 trials.

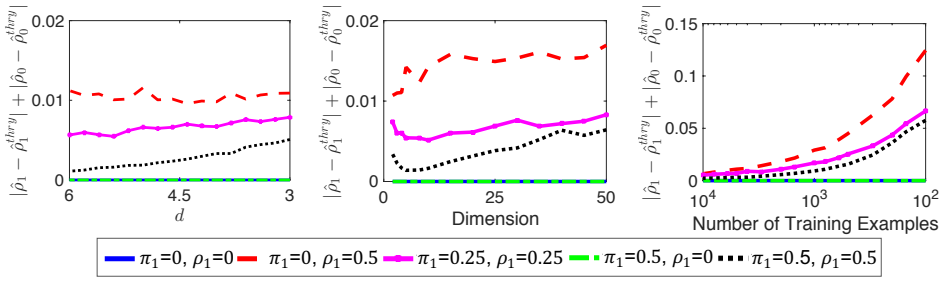

Figure 3: Sum of absolute difference between theoretically estimated $\hat{\rho}_i^{thry}$ and empirical $\hat{\rho}_i$, $i = 0, 1$, with five different (π_1, ρ_1), for varying separability d, dimension, and number of training examples. Note that no figure exists for percent random noise because the theoretical estimates in Eq. (8) do not address added noise examples.

AUC-PR metrics. We include all three to emphasize our apathy toward tuning results to any single metric. We provide F1 scores in this section with error and AUC-PR scores in Appendix C.

4.1 Synthetic Dataset

The synthetic dataset is comprised of a Guassian positive class and a Guassian negative classes such that negative examples $(y = 0)$ obey an m-dimensional Gaussian distribution $N(\mathbf{0}, \mathbf{I})$ with unit variance $\mathbf{I} = diag(1, 1, ...1)$, and positive examples obey $N(d\mathbf{1}, 0.8\mathbf{I})$, where $d\mathbf{1} = (d, d, ...d)$ is an m-dimensional vector, and d measures the separability of the positive and negative set.

We test Rank Pruning by varying 4 different settings of the environment: separability d, dimension, number of training examples n, and percent (of n) added random noise drawn from a uniform distribution $U([-10, 10]^m)$. In each scenario, we test 5 different (π_1, ρ_1) pairs: $(\pi_1, \rho_1) \in \{(0, 0), (0, 0.5), (0.25, 0.25), (0.5, 0), (0.5, 0.5)\}$. From Fig. 2, we observe that across these settings, the F1 score for Rank Pruning is fairly agnostic to magnitude of mislabeling (noise rates). As a validation step, in Fig. 3 we measure how closely our empirical estimates match our theoretical solutions in Eq. (8) and find near equivalence except when the number of training examples approaches zero.

For significant mislabeling ($\rho_1 = 0.5$, $\pi_1 = 0.5$), Rank Pruning often outperforms other methods (Fig. 4). In the scenario of different separability d, it achieves nearly the same F1 score as the ground truth classifier. Remarkably, from Fig. 2 and Fig. 4, we observe that when added random noise comprises 50% of total training examples, Rank Pruning still achieves F1 > 0.85, compared with F1 < 0.5 for all other methods. This emphasizes a unique feature of Rank Pruning, it will also remove added random noise because noise drawn from a third distribution is unlikely to appear confidently positive or negative.

4.2 MNIST and CIFAR Datasets

We consider the binary classification tasks of one-vs-rest for the MNIST (LeCun & Cortes, 2010) and CIFAR-10 (Krizhevsky et al.) datasets, e.g. the "car vs rest" task in CIFAR is to predict if an image is a "car" or "not". ρ_1 and π_1 are given to all $\tilde{PN}$ learning methods for fair comparison, except for RP_ρ which is Rank Pruning including noise rate

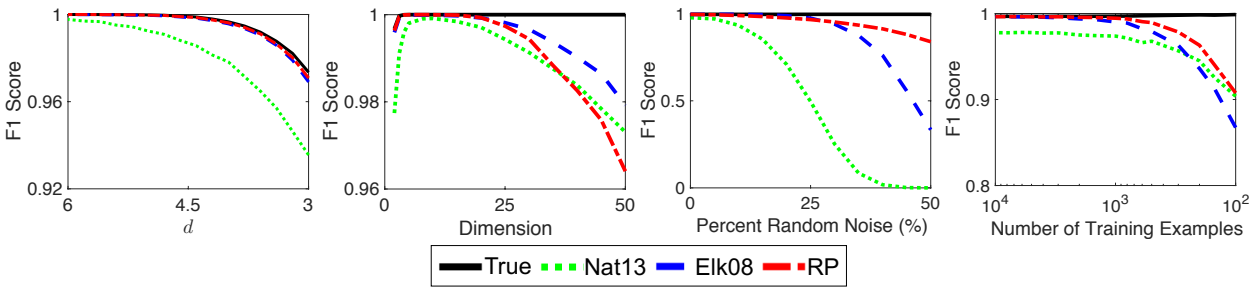

Figure 4: Comparison of $\tilde{P}\tilde{N}$ methods for varying separability d, dimension, added random noise, and number of training examples for $\pi_1 = 0.5$, $\rho_1 = 0.5$ (given to all methods).

estimation. RP_ρ metrics measure our performance on the unadulterated $\tilde{P}\tilde{N}$ learning problem.

As evidence that Rank Pruning is dataset and classifier agnostic, we demonstrate its superiority with both (1) a linear logistic regression model with unit L2 regularization and (2) an AlexNet CNN variant with max pooling and dropout, modified to have a two-class output. The CNN structure is adapted from Chollet (2016b) for MNIST and Chollet (2016a) for CIFAR. CNN training ends when a 10% holdout set shows no loss decrease for 10 epochs (max 50 for MNIST and 150 for CIFAR).

We consider noise rates $\pi_1, \rho_1 \in \{(0, 0.5), (0.25, 0.25), (0.5, 0), (0.5, 0.5)\}$ for both MNIST and CIFAR, with additional settings for MNIST in Table 3 to emphasize Rank Pruning performance is noise rate agnostic. The $\rho_1 = 0$, $\pi_1 = 0$ case is omitted because when given ρ_1, π_1, all methods have the same loss function as the ground truth classifier, resulting in nearly identical F1 scores. Note that in general, Rank Pruning does not require perfect probability estimation to achieve perfect F1-score. As an example, this occurs when P and N are range-separable, and the rank order of the sorted $g(x)$ probabilities in P and N is consistent with the rank of the perfect probabilities, regardless of the actual values of $g(x)$.

For MNIST using logistic regression, we evaluate the consistency of our noise rate estimates with actual noise rates and theoretical estimates (Eq. 8) across $\pi_1 \in [0, 0.8] \times \rho_1 \in [0, 0.9]$. The computing time for one setting was ~ 10 minutes on a single CPU core. The results for $\hat{\rho}_1$ and $\hat{\pi}_1$ (Fig. 5) are satisfyingly consistent, with mean absolute difference $\text{MD}_{\hat{\rho}_1, \rho_1} = 0.105$ and $\text{MD}_{\hat{\pi}_1, \pi_1} = 0.062$, and validate our theoretical solutions ($\text{MD}_{\hat{\rho}_1, \hat{\rho}_1^{thry}} = 0.0028$, $\text{MD}_{\hat{\pi}_1, \hat{\pi}_1^{thry}} = 0.0058$). The deviation of the theoretical and empirical estimates reflects the assumption that we have

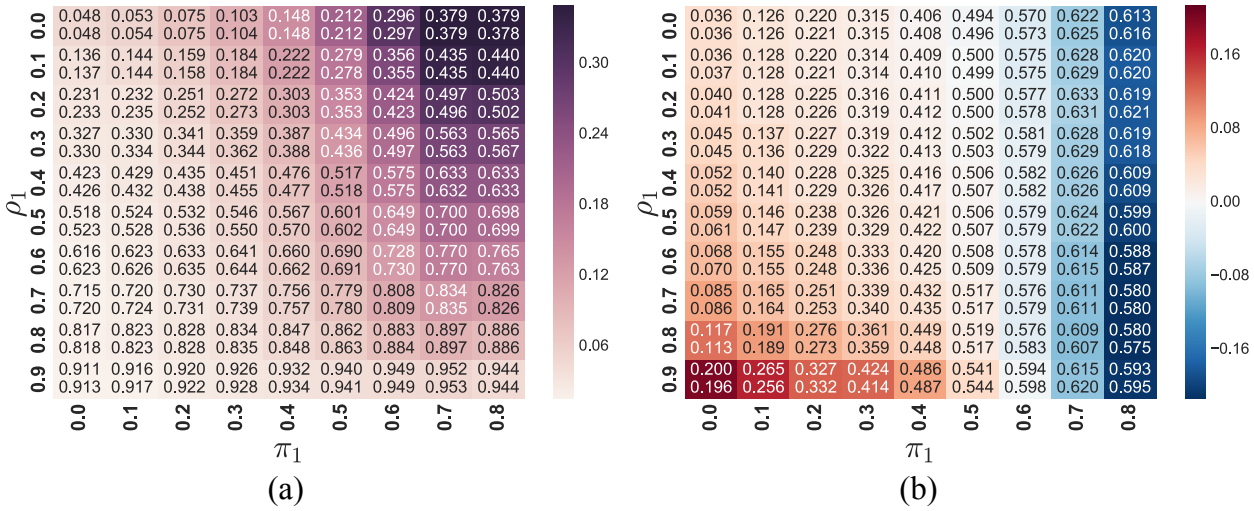

Figure 5: Rank Pruning $\hat{\rho}_1$ and $\hat{\pi}_1$ estimation consistency, averaged over all digits in MNIST. **(a)** Color depicts $\hat{\rho}_1 - \rho_1$ with $\hat{\rho}_1$ (upper) and theoretical $\hat{\rho}_1^{thry}$ (lower) in each block. **(b)** Color depicts $\hat{\pi}_1 - \pi_1$ with $\hat{\pi}_1$ (upper) and $\hat{\pi}_1^{thry}$ (lower) in each block.

Table 3: Comparison of F1 score for one-vs-rest MNIST and CIFAR-10 (averaged over all digits/images) using logistic regression. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models. Due to sensitivity to imperfect $g(x)$, *Liu16* often predicts the same label for all examples.

DATASET	CIFAR				MNIST															
$\pi_1 =$	**0.0**	**0.25**	**0.5**	**0.5**	π_1 **= 0.0**			π_1 **= 0.25**				π_1 **= 0.5**				π_1 **= 0.75**			
MODEL,$\rho_1 =$	**0.5**	**0.25**	**0.0**	**0.5**	**0.25**	**0.5**	**0.75**	**0.0**	**0.25**	**0.5**	**0.75**	**0.0**	**0.25**	**0.5**	**0.75**	**0.0**	**0.25**	**0.5**	**0.75**
TRUE	0.248	0.248	0.248	0.248	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894
RP_ρ	**0.301**	**0.316**	**0.308**	**0.261**	**0.883**	**0.874**	**0.843**	**0.881**	**0.876**	**0.863**	**0.799**	0.823	0.831	**0.819**	**0.762**	0.583	0.603	0.587	0.532
RP	**0.256**	**0.262**	**0.244**	0.209	**0.885**	**0.873**	**0.839**	**0.890**	**0.879**	**0.863**	**0.812**	**0.879**	**0.862**	**0.838**	**0.770**	**0.855**	**0.814**	**0.766**	0.617
NAT13	0.226	0.219	0.194	0.195	0.860	0.830	0.774	0.865	0.836	0.802	0.748	0.839	0.810	0.777	0.721	0.809	0.776	0.736	**0.640**
ELK08	0.221	0.226	0.228	**0.210**	0.862	0.830	0.771	0.864	0.847	0.819	0.762	0.843	0.835	0.814	0.736	0.674	0.669	0.599	0.473
LIU16	0.182	0.182	0.000	0.182	0.021	0.000	0.000	0.000	0.147	0.147	0.073	0.000	0.164	0.163	0.163	0.047	0.158	0.145	0.164

Table 4: F1 score comparison on MNIST and CIFAR-10 using a CNN. Except for RP_ρ, ρ_1, ρ_0 are given to all methods.

MNIST/CIFAR IMAGE CLASS		π_1 **= 0.0** ρ_1 **= 0.5**					π_1 **= 0.25** ρ_1 **= 0.25**					π_1 **= 0.5** ρ_1 **= 0.0**					π_1 **= 0.5** ρ_1 **= 0.5**				
	TRUE	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16
0	0.993	**0.991**	**0.988**	0.977	0.976	0.179	**0.991**	**0.992**	0.982	0.981	0.179	**0.991**	**0.992**	0.984	0.987	0.985	**0.989**	**0.989**	0.937	0.964	0.179
1	0.993	**0.990**	**0.991**	0.989	0.985	0.204	**0.992**	**0.992**	0.984	0.987	0.204	0.990	0.991	0.992	**0.993**	0.990	**0.989**	**0.989**	0.984	0.988	0.204
2	0.987	**0.973**	**0.976**	0.972	0.969	0.187	**0.984**	**0.983**	0.978	0.975	0.187	0.985	0.986	0.985	0.986	**0.988**	**0.971**	**0.975**	0.968	0.959	0.187
3	0.990	**0.984**	**0.984**	0.972	0.981	0.183	**0.986**	**0.986**	0.978	0.978	0.183	**0.990**	0.987	**0.989**	**0.989**	0.984	**0.981**	**0.979**	0.957	0.971	0.183
4	0.994	**0.981**	0.979	**0.981**	0.977	0.179	**0.985**	**0.987**	0.971	0.964	0.179	0.987	**0.990**	**0.990**	0.989	0.985	**0.977**	**0.982**	0.955	0.961	0.179
5	0.989	**0.982**	**0.980**	0.978	0.979	0.164	**0.985**	**0.982**	0.964	0.965	0.164	**0.988**	**0.987**	**0.987**	0.984	**0.987**	**0.965**	**0.968**	0.962	0.957	0.164
6	0.989	**0.986**	**0.985**	0.972	0.982	0.175	**0.985**	**0.987**	0.978	0.981	0.175	0.985	0.985	**0.988**	0.987	0.985	**0.983**	**0.982**	0.946	0.959	0.175
7	0.987	**0.981**	**0.980**	0.967	0.948	0.186	**0.976**	**0.975**	0.971	0.971	0.186	0.976	0.980	**0.985**	0.982	0.983	**0.973**	**0.968**	0.942	0.958	0.186
8	0.989	**0.975**	**0.978**	0.943	0.967	0.178	**0.982**	**0.981**	0.967	0.951	0.178	0.982	**0.984**	0.982	0.979	0.983	**0.977**	**0.975**	0.864	0.959	0.178
9	0.982	0.966	**0.974**	0.972	0.935	0.183	**0.976**	**0.974**	0.967	0.967	0.183	0.976	0.975	0.974	**0.978**	0.970	**0.959**	0.940	0.931	**0.942**	0.183
AVG$_{MN}$	0.989	**0.981**	**0.981**	0.972	0.970	0.182	**0.984**	**0.984**	0.974	0.972	0.182	0.985	**0.986**	**0.986**	0.985	0.984	**0.976**	**0.975**	0.945	0.962	0.182
PLANE	0.755	**0.689**	**0.634**	0.619	0.585	0.182	**0.695**	**0.702**	0.671	0.640	0.182	**0.757**	**0.746**	0.716	0.735	0.000	**0.628**	**0.635**	0.459	0.598	0.182
AUTO	0.891	**0.791**	**0.785**	0.761	0.768	0.000	**0.832**	**0.824**	0.771	0.783	0.182	0.862	0.866	**0.869**	0.865	0.000	**0.749**	**0.720**	0.582	0.501	0.182
BIRD	0.669	**0.504**	**0.483**	0.445	0.389	0.182	**0.543**	**0.515**	0.469	0.426	0.182	**0.577**	**0.619**	0.543	0.551	0.000	**0.447**	**0.409**	0.366	0.387	0.182
CAT	0.487	**0.350**	0.279	0.310	**0.313**	0.000	**0.426**	0.317	**0.350**	0.345	0.182	**0.489**	**0.433**	0.426	0.347	0.000	**0.394**	0.282	0.240	**0.313**	0.182
DEER	0.726	**0.593**	**0.540**	0.455	0.522	0.182	**0.585**	0.554	0.480	**0.569**	0.182	0.614	0.630	**0.643**	0.633	0.000	**0.458**	0.375	0.310	**0.383**	0.182
DOG	0.569	**0.544**	**0.577**	0.429	0.456	0.000	**0.579**	0.559	0.569	**0.576**	0.182	0.647	0.637	**0.667**	0.630	0.000	**0.516**	0.461	0.412	**0.465**	0.182
FROG	0.815	**0.746**	0.727	**0.733**	0.718	0.000	**0.729**	**0.750**	0.630	0.584	0.182	0.767	**0.782**	0.777	0.770	0.000	**0.635**	**0.615**	0.589	0.524	0.182
HORSE	0.805	**0.690**	0.670	0.624	**0.672**	0.182	**0.710**	0.669	**0.683**	0.627	0.182	0.761	**0.776**	0.769	0.753	0.000	**0.672**	**0.569**	0.551	0.461	0.182
SHIP	0.851	**0.791**	**0.783**	0.719	0.758	0.182	**0.810**	**0.801**	0.758	0.723	0.182	0.816	0.822	0.830	**0.831**	0.000	**0.715**	**0.738**	0.569	0.632	0.182
TRUCK	0.861	**0.744**	**0.722**	0.655	0.665	0.182	**0.814**	**0.826**	0.798	0.774	0.182	0.812	0.830	**0.826**	0.824	0.000	**0.654**	0.543	0.575	**0.584**	0.182
AVG$_{CF}$	0.743	**0.644**	**0.620**	0.575	0.585	0.109	**0.672**	**0.652**	0.618	0.605	0.182	**0.710**	**0.714**	0.707	0.694	0.000	**0.587**	**0.535**	0.465	0.485	0.182

infinite examples, whereas empirically, the number of examples is finite.

We emphasize two observations from our analysis on CIFAR and MNIST. First, Rank Pruning performs well in nearly every scenario and boasts the most dramatic improvement over prior state-of-the-art in the presence of extreme noise ($\pi_1 = 0.5$, $\rho_1 = 0.5$). This is easily observed in the right-most quadrant of Table 4. The $\pi_1 = 0.5$, $\rho_1 = 0$ quadrant is nearest to $\pi_1 = 0$, $\rho_1 = 0$ and mostly captures CNN prediction variation because $|\tilde{P}| \ll |\tilde{N}|$.

Second, RP_ρ often achieves equivalent (MNIST in Table 4) or significantly higher (CIFAR in Tables 3 and 4) F1 score than Rank Pruning when ρ_1 and π_1 are provided, particularly when noise rates are large. This effect is exacerbated for harder problems (lower F1 score for the ground truth classifier) like the "cat" in CIFAR or the "9" digit in MNIST likely because these problems are more complex, resulting in less confident predictions, and therefore more pruning.

Remember that ρ_1^{conf} and ρ_0^{conf} are upper bounds when g is unassuming. Noise rate overestimation accounts for the complexity of harder problems. As a downside, Rank Pruning may remove correctly labeled examples that "confuse" the classifier, instead fitting only the confident examples in each class. We observe this on CIFAR in Table 3 where logistic regression severely underfits so that RP_ρ has significantly higher F1 score than the ground truth classifier. Although Rank Pruning with noisy labels seemingly outperforms the ground truth model, if we lower the classification threshold to 0.3 instead of 0.5, the performance difference goes away by accounting for the lower probability predictions.

5 Discussion

To our knowledge, Rank Pruning is the first time-efficient algorithm, w.r.t. classifier fitting time, for $\tilde{P}\tilde{N}$ learning that achieves similar or better F1, error, and AUC-PR than current state-of-the-art methods across practical scenarios for synthetic, MNIST, and CIFAR datasets, with logistic regression and CNN classifiers, across all noise rates, ρ_1, ρ_0, for varying added noise, dimension, separability, and number of training examples. By *learning with confident examples*, we discover provably consistent estimators for noise rates, ρ_1, ρ_0, derive theoretical solutions when g is unassuming,

and accurately uncover the classifications of f fit to hidden labels, perfectly when g range separates P and N.

We recognize that disambiguating whether we are in the unassuming or range separability condition may be desirable. Although knowing $g^*(x)$ and thus $\Delta g(x)$ is impossible, if we assume randomly uniform noise, and toggling the $LB_{y=1}$ threshold does not change ρ_1^{conf}, then g range separates P and N. When g is unassuming, Rank Pruning is still robust to imperfect $g(x)$ within a range separable subset of P and N by training with confident examples even when noise rate estimates are inexact.

An important contribution of Rank Pruning is generality, both in classifier and implementation. The use of logistic regression and a generic CNN in our experiments emphasizes that our findings are not dependent on model complexity. We evaluate thousands of scenarios to avoid findings that are an artifact of problem setup. A key point of Rank Pruning is that we only report the simplest, non-parametric version. For example, we use 3-fold cross-validation to compute $g(x)$ even though we achieved improved performance with larger folds. We tried many variants of pruning and achieved significant higher F1 for MNIST and CIFAR, but to maintain generality, we present only the basic model.

At its core, Rank Pruning is a simple, robust, and general solution for noisy binary classification by *learning with confident examples*, but it also challenges how we think about training data. For example, SVM showed how a decision boundary can be recovered from only support vectors. Yet, when training data contains significant mislabeling, confident examples, many of which are far from the boundary, are informative for uncovering the true relationship $P(y = 1|x)$. Although modern affordances of "big data" emphasize the value of *more* examples for training, through Rank Pruning we instead encourage a rethinking of learning with *confident* examples.

References

Aha, D. W., Kibler, D., and Albert, M. K. Instance-based learning algorithms. *Mach. Learn.*, 6(1):37–66, 1991.

Angelova, A., Abu-Mostafam, Y., and Perona, P. Pruning training sets for learning of object categories. In *CVPR*, volume 1, pp. 494–501. IEEE, 2005.

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

Blanchard, G., Lee, G., and Scott, C. Semi-supervised novelty detection. *J. Mach. Learn. Res.*, 11:2973–3009, December 2010. ISSN 1532-4435.

Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *11th Conf. on COLT*, pp. 92–100, New York, NY, USA, 1998. ACM.

Blum, M., Floyd, R. W., Pratt, V., Rivest, R. L., and Tarjan, R. E. Time bounds for selection. *J. Comput. Syst. Sci.*, 7 (4):448–461, August 1973. ISSN 0022-0000.

Breiman, L. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. ISSN 0885-6125.

Chapelle, O. and Vapnik, V. Model selection for support vector machines. In *Proc. of 12th NIPS*, pp. 230–236, Cambridge, MA, USA, 1999.

Chollet, F. *Keras CIFAR CNN*, 2016a. bit.ly/2mVKR3d.

Chollet, F. *Keras MNIST CNN*, 2016b. bit.ly/2nKiqJv.

Claesen, M., Smet, F. D., Suykens, J. A., and Moor, B. D. A robust ensemble approach to learn from positive and unlabeled data using {SVM} base models. *Neurocomputing*, 160:73 – 84, 2015. ISSN 0925-2312.

Davis, J. and Goadrich, M. The relationship between precision-recall and roc curves. In *Proc. of 23rd ICML*, pp. 233–240, NYC, NY, USA, 2006. ACM.

Elkan, C. and Noto, K. Learning classifiers from only positive and unlabeled data. In *Proc. of 14th KDD*, pp. 213–220, NYC, NY, USA, 2008. ACM.

Hempstalk, K., Frank, E., and Witten, I. H. One-class classification by combining density and class probability estimation. In *Proc. of ECML-PKDD*, pp. 505–519, Berlin, Heidelberg, 2008. Springer-Verlag.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research).

Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., and Campbell, A. T. A survey of mobile phone sensing. *IEEE Communications*, 48(9), 2010.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010.

Lee, W. and Liu, B. Learning with positive and unlabeled examples using weighted logistic regression. In *Proc. of 20th ICML*, volume 1, pp. 448–455, 12 2003.

Liu, B., Dai, Y., Li, X., Lee, W. S., and Yu, P. S. Building text classifiers using positive and unlabeled examples. In *Proc. of 3rd ICDM*, pp. 179–, Washington, DC, USA, 2003. IEEE Computer Society.

Liu, T. and Tao, D. Classification with noisy labels by importance reweighting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(3):447–461, March 2016.

Manevitz, L. M. and Yousef, M. One-class svms for document classification. *JMLR*, 2:139–154, March 2002.

Menon, A. K., Jiang, X., Vembu, S., Elkan, C., and Ohno-Machado, L. Predicting accurate probabilities with a ranking loss. *CoRR*, abs/1206.4661, 2012.

Michalski, S. R., Carbonell, G. J., and Mitchell, M. T. *ML an AI Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1986.

Mordelet, F. and Vert, J. P. A bagging svm to learn from positive and unlabeled examples. *Pattern Recogn. Lett.*, 37: 201–209, February 2014. ISSN 0167-8655.

Moya, M. M., Koch, M. W., and Hostetler, L. D. One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N*, 93, 1993.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *Adv. in NIPS 26*, pp. 1196–1204. Curran Associates, Inc., 2013.

Nettleton, D. F., Orriols-Puig, A., and Fornells, A. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306, 2010.

Nigam, K. and Ghani, R. Understanding the behavior of co-training. In *KDD Workshop*, 2000.

Paolacci, G., Chandler, J., and Ipeirotis, P. G. Running experiments on amazon mechanical turk. *Judgment and Decision Making*, 5(5):411–419, 2010.

Platt, J., Schölkopf, B., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. Estimating support of a high dimensional distribution. Technical report, MSR, 1999.

Raviv, Y. and Intrator, N. Bootstrapping with noise: An effective regularization technique. *Connection Science*, 8 (3-4):355–372, 1996.

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

scikit learn. *LogisticRegression Class at scikit-learn*, 2016.

Scott, C. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. *JMLR*, 38:838–846, 2015. ISSN 1532-4435.

Scott, C., Blanchard, G., and Handy, G. Classification with asymmetric label noise: Consistency and maximal denoising. In *COLT*, pp. 489–511, 2013.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density Ratio Estimation in ML*. Cambridge University Press, New York, NY, USA, 1st edition, 2012.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

Yang, T., Mahdavi, M., Jin, R., Zhang, L., and Zhou, Y. Multiple kernel learning from noisy labels by stochastic programming. In *Proc. of 29th ICML*, pp. 233–240, New York, NY, USA, 2012. ACM.

Appendix

A Proofs

In this section, we provide proofs for all the lemmas and theorems in the main paper. We always assume that a class-conditional extension of the Classification Noise Process (CNP) (Angluin & Laird, 1988) maps true labels y to observed labels s such that each label in P is flipped independently with probability ρ_1 and each label in N is flipped independently with probability ρ_0 ($s \leftarrow CNP(y, \rho_1, \rho_0)$), so that $P(s = s|y = y, x) = P(s = s|y = y)$. Remember that $\rho_1 + \rho_0 < 1$ is a necessary condition of minimal information, other we may learn opposite labels.

In Lemma 1, Theorem 2, Lemma 3 and Theorem 4, we assume that P and N have infinite number of examples so that they are the true, hidden distributions.

A fundamental equation we use in the proofs is the following lemma:

Lemma A1 *When g is ideal, i.e. $g(x) = g^*(x)$ and P and N have non-overlapping support, we have*

$$g(x) = (1 - \rho_1) \cdot \mathbb{1}[[y = 1]] + \rho_0 \cdot \mathbb{1}[[y = 0]] \tag{A.1}$$

Proof: Since $g(x) = g^*(x)$ and P and N have non-overlapping support, we have

$$\begin{aligned} g(x) =& g^*(x) = P(s = 1|x) \\ =& P(s = 1|y = 1, x) \cdot P(y = 1|x) + P(s = 1|y = 0, x) \cdot P(y = 0|x) \\ =& P(s = 1|y = 1) \cdot P(y = 1|x) + P(s = 1|y = 0) \cdot P(y = 0|x) \\ =& (1 - \rho_1) \cdot \mathbb{1}[[y = 1]] + \rho_0 \cdot \mathbb{1}[[y = 0]] \end{aligned}$$

A.1 Proof of Lemma 1

Lemma 1 *When g is ideal, i.e. $g(x) = g^*(x)$ and P and N have non-overlapping support, we have*

$$\begin{cases} \tilde{P}_{y=1} = \{x \in P|s = 1\}, \tilde{N}_{y=1} = \{x \in P|s = 0\} \\ \tilde{P}_{y=0} = \{x \in N|s = 1\}, \tilde{N}_{y=0} = \{x \in N|s = 0\} \end{cases} \tag{A.2}$$

Proof: Firstly, we compute the threshold $LB_{y=1}$ and $UB_{y=0}$ used by $\tilde{P}_{y=1}$, $\tilde{N}_{y=1}$, $\tilde{P}_{y=0}$ and $\tilde{N}_{y=0}$. Since P and N have non-overlapping support, we have $P(y = 1|x) = \mathbb{1}[[y = 1]]$. Also using $g(x) = g^*(x)$, we have

$$\begin{aligned} LB_{y=1} =& E_{x \in \tilde{P}}[g(x)] = E_{x \in \tilde{P}}[P(s = 1|x)] \\ =& E_{x \in \tilde{P}}[P(s = 1|x, y = 1)P(y = 1|x) + P(s = 1|x, y = 0)P(y = 0|x)] \\ =& E_{x \in \tilde{P}}[P(s = 1|y = 1)P(y = 1|x) + P(s = 1|y = 0)P(y = 0|x)] \\ =& (1 - \rho_1)(1 - \pi_1) + \rho_0\pi_1 \end{aligned} \tag{A.3}$$

Similarly, we have

$$UB_{y=0} = (1 - \rho_1)\pi_0 + \rho_0(1 - \pi_0)$$

Since $\pi_1 = P(y = 0|s = 1)$, we have $\pi_1 \in [0, 1]$. Furthermore, we have the requirement that $\rho_1 + \rho_0 < 1$, then $\pi_1 = 1$ will lead to $\rho_1 = P(s = 0|y = 1) = 1 - P(s = 1|y = 1) = 1 - \frac{P(y=1|s=1)P(s=1)}{P(y=1)} = 1 - 0 = 1$ which violates the requirement of $\rho_1 + \rho_0 < 1$. Therefore, $\pi_1 \in [0, 1)$. Similarly, we can prove $\pi_0 \in [0, 1)$. Therefore, we see that both $LB_{y=1}$ and $UB_{y=0}$ are interpolations of $(1 - \rho_1)$ and ρ_0:

$$\begin{aligned} \rho_0 < LB_{y=1} \leq 1 - \rho_1 \\ \rho_0 \leq UB_{y=0} < 1 - \rho_1 \end{aligned}$$

The first equality holds iff $\pi_1 = 0$ and the second equality holds iff $\pi_0 = 0$.

Using Lemma A1, we know that under the condition of $g(x) = g^*(x)$ and non-overlapping support, $g(x) = (1-\rho_1) \cdot \mathbb{1}[[y=1]] + \rho_0 \cdot \mathbb{1}[[y=0]]$. In other words,

$$g(x) \geq LB_{y=1} \Leftrightarrow x \in P$$
$$g(x) \leq UB_{y=0} \Leftrightarrow x \in N$$

Since

$$\begin{cases} \tilde{P}_{y=1} = \{x \in \tilde{P} | g(x) \geq LB_{y=1}\} \\ \tilde{N}_{y=1} = \{x \in \tilde{N} | g(x) \geq LB_{y=1}\} \\ \tilde{P}_{y=0} = \{x \in \tilde{P} | g(x) \leq UB_{y=0}\} \\ \tilde{N}_{y=0} = \{x \in \tilde{N} | g(x) \leq UB_{y=0}\} \end{cases}$$

where $\tilde{P} = \{x | s = 1\}$ and $\tilde{N} = \{x | s = 0\}$, we have

$$\begin{cases} \tilde{P}_{y=1} = \{x \in P | s = 1\}, \tilde{N}_{y=1} = \{x \in P | s = 0\} \\ \tilde{P}_{y=0} = \{x \in N | s = 1\}, \tilde{N}_{y=0} = \{x \in N | s = 0\} \end{cases}$$

A.2 Proof of Theorem 2

We restate Theorem 2 here:

Theorem 2 *When g is ideal, i.e. $g(x) = g^*(x)$ and P and N have non-overlapping support, we have*

$$\hat{\rho}_1^{conf} = \rho_1, \hat{\rho}_0^{conf} = \rho_0$$

Proof: Using the definition of $\hat{\rho}_1^{conf}$ in the main paper:

$$\hat{\rho}_1^{conf} = \frac{|\tilde{N}_{y=1}|}{|\tilde{N}_{y=1}| + |\tilde{P}_{y=1}|},\ \hat{\rho}_0^{conf} = \frac{|\tilde{P}_{y=0}|}{|\tilde{P}_{y=0}| + |\tilde{N}_{y=0}|}$$

Since $g(x) = g^*(x)$ and P and N have non-overlapping support, using Lemma 1, we know

$$\begin{cases} \tilde{P}_{y=1} = \{x \in P | s = 1\}, \tilde{N}_{y=1} = \{x \in P | s = 0\} \\ \tilde{P}_{y=0} = \{x \in N | s = 1\}, \tilde{N}_{y=0} = \{x \in N | s = 0\} \end{cases}$$

Since $\rho_1 = P(s = 0 | y = 1)$ and $\rho_0 = P(s = 1 | y = 0)$, we immediately have

$$\hat{\rho}_1^{conf} = \frac{|\{x \in P | s = 0\}|}{|P|} = \rho_1,\ \hat{\rho}_0^{conf} = \frac{|\{x \in N | s = 1\}|}{|N|} = \rho_0$$

A.3 Proof of Lemma 3

We rewrite Lemma 3 below:

Lemma 3 *When g is unassuming, i.e., $\Delta g(x) := g(x) - g^*(x)$ can be nonzero, and P and N can have overlapping support, we have*

$$\begin{cases} LB_{y=1} = LB^*_{y=1} + E_{x \in \tilde{P}}[\Delta g(x)] - \frac{(1-\rho_1-\rho_0)^2}{p_{s1}} \Delta p_o \\ UB_{y=0} = UB^*_{y=0} + E_{x \in \tilde{N}}[\Delta g(x)] + \frac{(1-\rho_1-\rho_0)^2}{1-p_{s1}} \Delta p_o \\ \hat{\rho}_1^{conf} = \rho_1 + \frac{1-\rho_1-\rho_0}{|P|-|\Delta P_1|+|\Delta N_1|} |\Delta N_1| \\ \hat{\rho}_0^{conf} = \rho_0 + \frac{1-\rho_1-\rho_0}{|N|-|\Delta N_0|+|\Delta P_0|} |\Delta P_0| \end{cases} \tag{A.4}$$

where

$$\begin{cases} LB^*_{y=1} = (1-\rho_1)(1-\pi_1) + \rho_0\pi_1 \\ UB^*_{y=0} = (1-\rho_1)\pi_0 + \rho_0(1-\pi_0) \\ \Delta p_o := \frac{|P \cap N|}{|P \cup N|} \\ \Delta P_1 = \{x \in P | g(x) < LB_{y=1}\} \\ \Delta N_1 = \{x \in N | g(x) \geq LB_{y=1}\} \\ \Delta P_0 = \{x \in P | g(x) \leq UB_{y=0}\} \\ \Delta N_0 = \{x \in N | g(x) > UB_{y=0}\} \end{cases} \tag{A.5}$$

Proof: We first calculate $LB_{y=1}$ and $UB_{y=0}$ under unassuming conditions, then calculate $\hat{\rho}_i^{conf}$, $i = 0, 1$ under unassuming condition.

Note that Δp_o can also be expressed as

$$\Delta p_o := \frac{|P \cap N|}{|P \cup N|} = P(\hat{y} = 1, y = 0) = P(\hat{y} = 0, y = 1)$$

Here $P(\hat{y} = 1, y = 0) \equiv P(\hat{y} = 1 | y = 0)P(y = 0)$, where $P(\hat{y} = 1 | y = 0)$ means for a perfect classifier $f^*(x) = P(y = 1|x)$, the expected probability that it will label a $y = 0$ example as positive ($\hat{y} = 1$).

(1) $LB_{y=1}$ and $UB_{y=0}$ under unassuming condition

Firstly, we calculate $LB_{y=1}$ and $UB_{y=0}$ with perfect probability estimation $g^*(x)$, but the support may overlap. Secondly, we allow the probability estimation to be imperfect, superimposed onto the overlapping support condition, and calculate $LB_{y=1}$ and $UB_{y=0}$.

I. Calculating $LB_{y=1}$ and $UB_{y=0}$ when $g(x) = g^*(x)$ and support may overlap

With overlapping support, we no longer have $P(y = 1|x) = \mathbb{1}[[y = 1]]$. Instead, we have

$$\begin{aligned} LB_{y=1} =& E_{x \in \tilde{P}}[g^*(x)] = E_{x \in \tilde{P}}[P(s = 1|x)] \\ =& E_{x \in \tilde{P}}[P(s = 1|x, y = 1)P(y = 1|x) + P(s = 1|x, y = 0)P(y = 0|x)] \\ =& E_{x \in \tilde{P}}[P(s = 1|y = 1)P(y = 1|x) + P(s = 1|y = 0)P(y = 0|x)] \\ =& (1 - \rho_1) \cdot E_{x \in \tilde{P}}[P(y = 1|x)] + \rho_0 \cdot E_{x \in \tilde{P}}[P(y = 0|x)] \\ =& (1 - \rho_1) \cdot P(\hat{y} = 1|s = 1) + \rho_0 \cdot P(\hat{y} = 0|s = 1) \end{aligned}$$

Here $P(\hat{y} = 1|s = 1)$ can be calculated using Δp_o:

$$\begin{aligned} P(\hat{y} = 1|s = 1) &= \frac{P(\hat{y} = 1, s = 1)}{P(s = 1)} \\ &= \frac{P(\hat{y} = 1, y = 1, s = 1) + P(\hat{y} = 1, y = 0, s = 1)}{P(s = 1)} \\ &= \frac{P(s = 1|y = 1)P(\hat{y} = 1, y = 1) + P(s = 1|y = 0)P(\hat{y} = 1, y = 0)}{P(s = 1)} \\ &= \frac{(1 - \rho_1)(p_{y1} - \Delta p_o) + \rho_0 \Delta p_o}{p_{s1}} \\ &= (1 - \pi_1) - \frac{1 - \rho_1 - \rho_0}{p_{s1}} \Delta p_o \end{aligned}$$

Hence,

$$P(\hat{y} = 0|s = 1) = 1 - P(\hat{y} = 1|s = 1) = \pi_1 + \frac{1 - \rho_1 - \rho_0}{p_{s1}} \Delta p_o$$

Therefore,

$$\begin{aligned} LB_{y=1} &= (1-\rho_1)\cdot P(\hat{y}=1|s=1) + \rho_0 \cdot P(\hat{y}=0|s=1) \\ &= (1-\rho_1)\cdot\left((1-\pi_1) - \frac{1-\rho_1-\rho_0}{p_{s1}}\Delta p_o\right) + \rho_0\cdot\left(\pi_1 + \frac{1-\rho_1-\rho_0}{p_{s1}}\Delta p_o\right) \\ &= LB^*_{y=1} - \frac{(1-\rho_1-\rho_0)^2}{p_{s1}}\Delta p_o \end{aligned} \tag{A.6}$$

where $LB^*_{y=1}$ is the $LB_{y=1}$ value when $g(x)$ is ideal. We see in Eq. (A.6) that the overlapping support introduces a non-positive correction to $LB^*_{y=1}$ compared with the ideal condition.

Similarly, we have

$$UB_{y=0} = UB^*_{y=0} + \frac{(1-\rho_1-\rho_0)^2}{1-p_{s1}}\Delta p_o \tag{A.7}$$

II. Calculating $LB_{y=1}$ and $UB_{y=0}$ when g is unassuming

Define $\Delta g(x) = g(x) - g^*(x)$. When the support may overlap, we have

$$\begin{aligned} LB_{y=1} &= E_{x\in\tilde{P}}[g(x)] \\ &= E_{x\in\tilde{P}}[g^*(x)] + E_{x\in\tilde{P}}[\Delta g(x)] \\ &= LB^*_{y=1} - \frac{(1-\rho_1-\rho_0)^2}{p_{s1}}\Delta p_o + E_{x\in\tilde{P}}[\Delta g(x)] \end{aligned} \tag{A.8}$$

Similarly, we have

$$\begin{aligned} UB_{y=0} &= E_{x\in\tilde{N}}[g(x)] \\ &= E_{x\in\tilde{N}}[g^*(x)] + E_{x\in\tilde{N}}[\Delta g(x)] \\ &= UB^*_{y=0} + \frac{(1-\rho_1-\rho_0)^2}{1-p_{s1}}\Delta p_o + E_{x\in\tilde{N}}[\Delta g(x)] \end{aligned} \tag{A.9}$$

In summary, Eq. (A.8) (A.9) give the expressions for $LB_{y=1}$ and $UB_{y=0}$, respectively, when g is unassuming.

(2) $\hat{\rho}_i^{conf}$ under unassuming condition

Now let's calculate $\hat{\rho}_i^{conf}$, $i = 0, 1$. For simplicity, define

$$\begin{cases} PP = \{x \in P | s = 1\} \\ PN = \{x \in P | s = 0\} \\ NP = \{x \in N | s = 1\} \\ NN = \{x \in N | s = 0\} \\ \Delta_{PP_1} = \{x \in PP | g(x) < LB_{y=1}\} \\ \Delta_{NP_1} = \{x \in NP | g(x) \geq LB_{y=1}\} \\ \Delta_{PN_1} = \{x \in PN | g(x) < LB_{y=1}\} \\ \Delta_{NN_1} = \{x \in NN | g(x) \geq LB_{y=1}\} \end{cases} \tag{A.10}$$

For $\hat{\rho}_1^{conf}$, we have:

$$\hat{\rho}_1^{conf} = \frac{|\tilde{N}_{y=1}|}{|\tilde{P}_{y=1}| + |\tilde{N}_{y=1}|}$$

Here

$$\begin{aligned}\tilde{P}_{y=1} &= \{x \in \tilde{P} | g(x) \geq LB_{y=1}\} \\ &= \{x \in PP | g(x) \geq LB_{y=1}\} \cup \{x \in NP | g(x) \geq LB_{y=1}\} \\ &= (PP \setminus \Delta_{PP_1}) \cup \Delta_{NP_1}\end{aligned}$$

Similarly, we have

$$\tilde{N}_{y=1} = (PN \setminus \Delta_{PN_1}) \cup \Delta_{NN_1}$$

Therefore

$$\begin{aligned}\hat{\rho}_1^{conf} &= \frac{|PN| - |\Delta_{PN_1}| + |\Delta_{NN_1}|}{[(|PP| - |\Delta_{PP_1}|) + (|PN| - |\Delta_{PN_1}|)] + (|\Delta_{NN_1}| + |\Delta_{NP_1}|)} \\ &= \frac{|PN| - |\Delta_{PN_1}| + |\Delta_{NN_1}|}{|P| - |\Delta P_1| + |\Delta N_1|}\end{aligned} \tag{A.11}$$

where in the second equality we have used the definition of ΔP_1 and ΔN_1 in Eq. (A.5).

Using the definition of ρ_1, we have

$$\begin{aligned}\frac{|PN| - |\Delta_{PN_1}|}{|P| - |\Delta P_1|} &= \frac{|\{x \in PN | g(x) \geq LB_{y=1}\}|}{|\{x \in P | g(x) \geq LB_{y=1}\}|} \\ &= \frac{P(x \in PN, g(x) \geq LB_{y=1})}{P(x \in P, g(x) \geq LB_{y=1})} \\ &= \frac{P(x \in PN | x \in P, g(x) \geq LB_{y=1}) \cdot P(x \in P, g(x) \geq LB_{y=1})}{P(x \in P, g(x) \geq LB_{y=1})} \\ &= \frac{P(x \in PN | x \in P) \cdot P(x \in P, g(x) \geq LB_{y=1})}{P(x \in P, g(x) \geq LB_{y=1})} \\ &= \rho_1\end{aligned}$$

Here we have used the property of CNP that $(s \perp\!\!\!\perp x)|y$, leading to $P(x \in PN | x \in P, g(x) \geq LB_{y=1}) = P(x \in PN | x \in P) = \rho_1$.

Similarly, we have

$$\frac{|\Delta_{NN_1}|}{|\Delta N_1|} = 1 - \rho_0$$

Combining with Eq. (A.11), we have

$$\hat{\rho}_1^{conf} = \rho_1 + \frac{1 - \rho_1 - \rho_0}{|P| - |\Delta P_1| + |\Delta N_1|} |\Delta N_1| \tag{A.12}$$

Similarly, we have

$$\hat{\rho}_0^{conf} = \rho_0 + \frac{1 - \rho_1 - \rho_0}{|N| - |\Delta N_0| + |\Delta P_0|} |\Delta P_0| \tag{A.13}$$

From the two equations above, we see that

$$\hat{\rho}_1^{conf} \geq \rho_1, \hat{\rho}_0^{conf} \geq \rho_0 \tag{A.14}$$

In other words, $\hat{\rho}_i^{conf}$ is an ***upper bound*** of ρ_i, $i = 0, 1$. The equality for $\hat{\rho}_1^{conf}$ holds if $|\Delta N_1| = 0$. The equality for $\hat{\rho}_0^{conf}$ holds if $|\Delta P_0| = 0$.

A.4 Proof of Theorem 4

Let's restate Theorem 4 below:

Theorem 4 *Given non-overlapping support condition,*

If $\forall x \in N, \Delta g(x) < LB_{y=1} - \rho_0$, then $\hat{\rho}_1^{conf} = \rho_1$.

If $\forall x \in P, \Delta g(x) > -(1 - \rho_1 - UB_{y=0})$, then $\hat{\rho}_0^{conf} = \rho_0$.

Theorem 4 directly follows from Eq. (A.12) and (A.13). Assuming non-overlapping support, we have $g^*(x) = P(s = 1|x) = (1 - \rho_1) \cdot \mathbb{1}[[y = 1]] + \rho_0 \cdot \mathbb{1}[[y = 0]]$. In other words, the contribution of overlapping support to $|\Delta N_1|$ and $|\Delta P_0|$ is 0. The only source of deviation comes from imperfect $g(x)$.

For the first half of the theorem, since $\forall x \in N, \Delta g(x) < LB_{y=1} - \rho_0$, we have $\forall x \in N, g(x) = \Delta g(x) + g^*(x) < (LB_{y=1} - \rho_0) + \rho_0 = LB_{y=1}$, then $|\Delta N_1| = |\{x \in N | g(x) \geq LB_{y=1}\}| = 0$, so we have $\hat{\rho}_1^{conf} = \rho_1$.

Similarly, for the second half of the theorem, since $\forall x \in P, \Delta g(x) > -(1 - \rho_1 - UB_{y=0})$, then $|\Delta P_0| = |\{x \in P | g(x) \leq UB_{y=0}\}| = 0$, so we have $\hat{\rho}_0^{conf} = \rho_0$.

A.5 Proof of Theorem 5

Theorem 5 reads as follows:

Theorem 5 *If g range separates P and N and $\hat{\rho}_i = \rho_i$, $i = 0, 1$, then for any classifier f_θ and any bounded loss function $l(\hat{y}_i, y_i)$, we have*

$$R_{\tilde{l}, \mathcal{D}_\rho}(f_\theta) = R_{l, \mathcal{D}}(f_\theta) \tag{A.15}$$

where $\tilde{l}(\hat{y}_i, s_i)$ is Rank Pruning's loss function given by

$$\tilde{l}(\hat{y}_i, s_i) = \frac{1}{1 - \hat{\rho}_1} l(\hat{y}_i, s_i) \cdot \mathbb{1}[[x_i \in \tilde{P}_{conf}]] + \frac{1}{1 - \hat{\rho}_0} l(\hat{y}_i, s_i) \cdot \mathbb{1}[[x_i \in \tilde{N}_{conf}]] \tag{A.16}$$

and $\tilde{P}_{conf}$ and $\tilde{N}_{conf}$ are given by

$$\tilde{P}_{conf} := \{x \in \tilde{P} \mid g(x) \geq k_1\}, \tilde{N}_{conf} := \{x \in \tilde{N} \mid g(x) \leq k_0\} \tag{A.17}$$

where k_1 is the $(\hat{\pi}_1 |\tilde{P}|)^{th}$ smallest $g(x)$ for $x \in \tilde{P}$ and k_0 is the $(\hat{\pi}_0 |\tilde{N}|)^{th}$ largest $g(x)$ for $x \in \tilde{N}$

Proof:

Since $\tilde{P}$ and $\tilde{N}$ are constructed from P and N with noise rates π_1 and π_0 using the class-conditional extension of the Classification Noise Process (Angluin & Laird, 1988), we have

$$\begin{cases} P = PP \cup PN \\ N = NP \cup NN \\ \tilde{P} = PP \cup NP \\ \tilde{N} = PN \cup NN \end{cases} \tag{A.18}$$

where

$$\begin{cases} PP = \{x \in P | s = 1\} \\ PN = \{x \in P | s = 0\} \\ NP = \{x \in N | s = 1\} \\ NN = \{x \in N | s = 0\} \end{cases} \tag{A.19}$$

satisfying

$$\begin{cases} PP \sim PN \sim P \\ NP \sim NN \sim N \\ \frac{|NP|}{|\tilde{P}|} = \pi_1, \frac{|PP|}{|\tilde{P}|} = 1 - \pi_1 \\ \frac{|PN|}{|\tilde{N}|} = \pi_0, \frac{|NN|}{|\tilde{N}|} = 1 - \pi_0 \\ \frac{|PN|}{|P|} = \rho_1, \frac{|PP|}{|P|} = 1 - \rho_1 \\ \frac{|NP|}{|N|} = \rho_0, \frac{|NN|}{|N|} = 1 - \rho_0 \end{cases} \tag{A.20}$$

Here the $\sim$ means obeying the same distribution.

Since g range separates P and N, there exists a real number z such that $\forall x_1 \in P$ and $\forall x_0 \in N$, we have $g(x_1) > z > g(x_0)$. Since $P = PP \cup PN$, $N = NP \cup NN$, we have

$$\begin{aligned} &\forall x \in PP, g(x) > z;\ \forall x \in PN, g(x) > z; \\ &\forall x \in NP, g(x) < z;\ \forall x \in NN, g(x) < z \end{aligned} \tag{A.21}$$

Since $\hat{\rho}_1 = \rho_1$ and $\hat{\rho}_0 = \rho_0$, we have

$$\begin{cases} \hat{\pi}_1 = \frac{\hat{\rho}_0}{p_{s1}} \frac{1 - p_{s1} - \hat{\rho}_1}{1 - \hat{\rho}_1 - \hat{\rho}_0} = \frac{\rho_0}{p_{s1}} \frac{1 - p_{s1} - \rho_1}{1 - \rho_1 - \rho_0} = \pi_1 \equiv \frac{\rho_0 |N|}{|\tilde{P}|} \\ \hat{\pi}_0 = \frac{\hat{\rho}_1}{1 - p_{s1}} \frac{p_{s1} - \hat{\rho}_0}{1 - \hat{\rho}_1 - \hat{\rho}_0} = \frac{\rho_1}{1 - p_{s1}} \frac{p_{s1} - \rho_0}{1 - \rho_1 - \rho_0} = \pi_0 \equiv \frac{\rho_1 |P|}{|\tilde{N}|} \end{cases} \tag{A.22}$$

Therefore, $\hat{\pi}_1 |\tilde{P}| = \pi_1 |\tilde{P}| = \rho_0 |N|$, $\hat{\pi}_0 |\tilde{N}| = \pi_0 |\tilde{N}| = \rho_1 |P|$. Using $\tilde{P}_{conf}$ and $\tilde{N}_{conf}$'s definition in Eq. (A.17), and $g(x)$'s property in Eq. (A.21), we have

$$\tilde{P}_{conf} = PP \sim P, \tilde{N}_{conf} = NN \sim N \tag{A.23}$$

Hence P_{conf} and N_{conf} can be seen as a uniform downsampling of P and N, with a downsampling ratio of $(1 - \rho_1)$ for P and $(1 - \rho_0)$ for N. Then according to Eq. (A.16), the loss function $\tilde{l}(\hat{y}_i, s_i)$ essentially sees a fraction of $(1 - \rho_1)$ examples in P and a fraction of $(1 - \rho_0)$ examples in N, with a final reweighting to restore the class balance.

Then for any classifier f_θ that maps $x \to \hat{y}$ and any bounded loss function $l(\hat{y}_i, y_i)$, we have

$$
\begin{aligned}
R_{\tilde{l},\mathcal{D}_\rho}(f_\theta) &= E_{(x,s)\sim\mathcal{D}_\rho}[\tilde{l}(f_\theta(x), s)] \\
&= \frac{1}{1-\hat{\rho}_1} \cdot E_{(x,s)\sim\mathcal{D}_\rho}\left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in \tilde{P}_{conf}]]\right] + \frac{1}{1-\hat{\rho}_0} \cdot E_{(x,s)\sim\mathcal{D}_\rho}\left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in \tilde{N}_{conf}]]\right] \\
&= \frac{1}{1-\rho_1} \cdot E_{(x,s)\sim\mathcal{D}_\rho}\left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in \tilde{P}_{conf}]]\right] + \frac{1}{1-\rho_0} \cdot E_{(x,s)\sim\mathcal{D}_\rho}\left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in \tilde{N}_{conf}]]\right] \\
&= \frac{1}{1-\rho_1} \cdot E_{(x,s)\sim\mathcal{D}_\rho}\left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in PP]]\right] + \frac{1}{1-\rho_0} \cdot E_{(x,s)\sim\mathcal{D}_\rho}\left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in NN]]\right] \\
&= \frac{1}{1-\rho_1} \cdot (1-\rho_1) \cdot E_{(x,y)\sim\mathcal{D}}\left[l(f_\theta(x), y) \cdot \mathbb{1}[[x \in P]]\right] + \frac{1}{1-\rho_0} \cdot (1-\rho_0) \cdot E_{(x,y)\sim\mathcal{D}}\left[l(f_\theta(x), y) \cdot \mathbb{1}[[x \in N]]\right] \\
&= E_{(x,y)\sim\mathcal{D}}\left[l(f_\theta(x), y) \cdot \mathbb{1}[[x \in P]] + l(f_\theta(x), y) \cdot \mathbb{1}[[x \in N]]\right] \\
&= E_{(x,y)\sim\mathcal{D}}\left[l(f_\theta(x), y)\right] \\
&= R_{l,\mathcal{D}}(f_\theta)
\end{aligned}
$$

Therefore, we see that the expected risk for Rank Pruning with corrupted labels, is exactly the same as the expected risk for the true labels, for any bounded loss function l and classifier f_θ. The reweighting ensures that after pruning, the two sets still remain unbiased w.r.t. to the true dataset.

Since the ideal condition is more strict than the range separability condition, we immediately have that when g is ideal and $\hat{\rho}_i = \rho_i$, $i = 0, 1$, $R_{\tilde{l},\mathcal{D}_\rho}(f_\theta) = R_{l,\mathcal{D}}(f_\theta)$ for any f_θ and bounded loss function l.

B Additional Figures

Figure B1 shows the average image for each digit for the problem "1" or "not 1" in MNIST with logistic regression and high noise ($\rho_1 = 0.5, \pi_1 = 0.5$). The number on the bottom and on the right counts the total number of examples (images). From the figure we see that Rank Pruning makes few mistakes, and when it does, the mistakes vary greatly in image from the typical digit.

C Additional Tables

Here we provide additional tables for the comparison of error, Precision-Recall AUC (AUC-PR, Davis & Goadrich (2006)), and F1 score for the algorithms *RP*, *Nat13*, *Elk08*, *Liu16* with ρ_1, ρ_0 given to all methods for fair comparison. Additionally, we provide the performance of the ground truth classifier (*true*) trained with uncorrupted labels (X, y), as well as the complete Rank Pruning algorithm (*RP*$_\rho$) trained using the noise rates estimated by Rank Pruning. The top model scores are in bold with RP_ρ in red if its performance is better than non-RP models. The $\pi_1 = 0$ quadrant in each table represents the "PU learning" case of $\tilde{P}\tilde{N}$ learning.

Whenever $g(x) = P(\hat{s} = 1|x)$ is estimated for any algorithm, we use a 3-fold cross-validation to estimate the probability $g(x)$. For improved performance, a higher fold may be used.

For the logistic regression classifier, we use scikit-learn's LogisticRegression class (scikit learn (2016)) with default settings (L2 regularization with inverse strength $C = 1$).

For the convolutional neural networks (CNN), for MNIST we use the structure in Chollet (2016b) and for CIFAR-10, we use the structure in Chollet (2016a). A 10% holdout set is used to monitor the weighted validation loss (using the sample weight given by each algorithm) and ends training when there is no decrease for 10 epochs, with a maximum of 50 epochs for MNIST and 150 epochs for CIFAR-10.

The following list comprises the MNIST and CIFAR experimental result tables for error, AUC-PR and F1 score metrics:

Table C1: Error for MNIST with logisitic regression as classifier.

Table C2: AUC-PR for MNIST with logisitic regression as classifier.

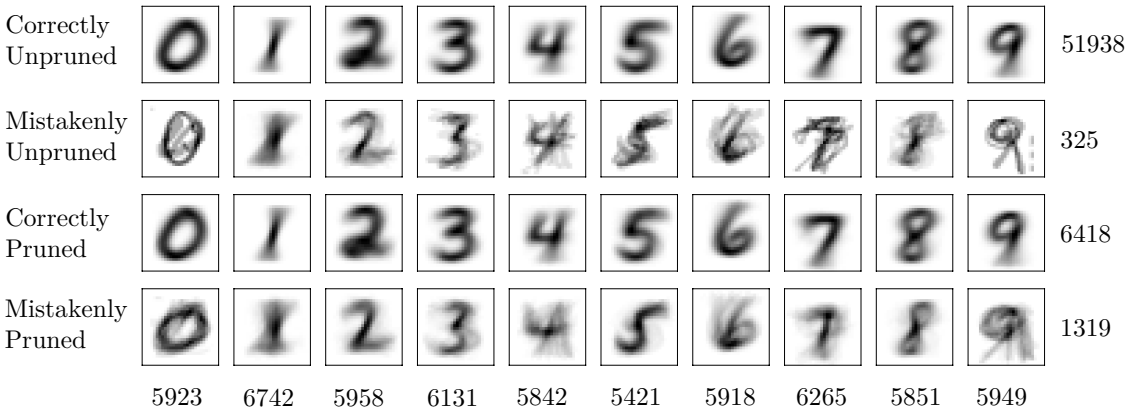

Figure B 1: Average image for each digit for the binary classification problem "1" or "not 1" in MNIST with logistic regression and significant mislabeling ($\rho_1 = 0.5, \pi_1 = 0.5$). The right and bottom numbers count the total number of example images averaged in the corresponding row or column.

Table C3: Error for MNIST with CNN as classifier.

Table C4: AUC-PR for MNIST with CNN as classifier.

Table C5: F1 score for CIFAR-10 with logistic regression as classifier.

Table C6: Error for CIFAR-10 with logistic regression as classifier.

Table C7: AUC-PR for CIFAR-10 with logistic regression as classifier.

Table C8: Error for CIFAR-10 with CNN as classifier.

Table C9: AUC-PR for CIFAR-10 with CNN as classifier.

Due to sensitivity to imperfect probability estimation, here *Liu16* always predicts all labels to be positive or negative, resulting in the same metric score for every digit/image in each scenario. Since $p_{y1} \simeq 0.1$, when predicting all labels as positive, *Liu16* has an F1 score of 0.182, error of 0.90, and AUC-PR of 0.55; when predicting all labels as negative, *Liu16* has an F1 score of 0.0, error of 0.1, and AUC-PR of 0.55.

D Additional Related Work

In this section we include tangentially related work which was unable to make it into the final manuscript.

D.1 One-class classification

One-class classification (Moya et al., 1993) is distinguished from binary classification by a training set containing examples from only one class, making it useful for outlier and novelty detection (Hempstalk et al., 2008). This can be framed as $\tilde{P}\tilde{N}$ learning when outliers take the form of mislabeled examples. The predominant approach, one-class SVM, fits a hyper-boundary around the training class (Platt et al., 1999), but often performs poorly due to boundary over-sensitivity (Manevitz & Yousef, 2002) and fails when the training class contains mislabeled examples.

D.2 $\tilde{P}\tilde{N}$ learning for Image Recognition and Deep Learning

Variations of $\tilde{P}\tilde{N}$ learning have been used in the context of machine vision to improve robustness to mislabeling (Xiao et al., 2015). In a face recognition task with 90% of non-faces mislabeled as faces, a bagging model combined with consistency voting was used to remove images with poor voting consistency (Angelova et al., 2005). However, no theoretical justification was provided. In the context of deep learning, consistency of predictions for inputs with

mislabeling enforces can be enforced by combining a typical cross-entropy loss with an auto-encoder loss (Reed et al., 2015). This method enforces label consistency by constraining the network to uncover the input examples given the output prediction, but is restricted in architecture and generality.

Table C 1: Comparison of **error** for one-vs-rest MNIST (averaged over all digits) using a **logistic regression** classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if better (smaller) than non-RP models.

	π_1 = 0				π_1 = 0.25				π_1 = 0.5				π_1 = 0.75			
MODEL,ρ_1 =	**0.25**	**0.50**	**0.75**		**0.00**	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**
TRUE	0.020	0.020	0.020		0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020
RP$_\rho$	**0.023**	**0.025**	**0.031**		**0.024**	**0.025**	**0.027**	**0.038**	0.040	0.037	0.039	0.049	0.140	0.128	0.133	0.151
RP	**0.022**	**0.025**	**0.031**		**0.021**	**0.024**	**0.027**	**0.035**	**0.023**	**0.027**	**0.031**	**0.043**	**0.028**	**0.036**	**0.045**	0.069
NAT13	0.025	0.030	0.038		0.025	0.029	0.034	0.042	0.030	0.033	0.038	0.047	0.035	0.039	0.046	**0.067**
ELK08	0.025	0.030	0.038		0.026	0.028	0.032	0.042	0.030	0.031	0.035	0.051	0.092	0.093	0.123	0.189
LIU16	0.187	0.098	0.100		0.100	0.738	0.738	0.419	0.100	0.820	0.821	0.821	0.098	0.760	0.741	0.820

Table C 2: Comparison of **AUC-PR** for one-vs-rest MNIST (averaged over all digits) using a **logistic regression** classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models.

	π_1 = 0				π_1 = 0.25				π_1 = 0.5				π_1 = 0.75			
MODEL,ρ_1 =	**0.25**	**0.50**	**0.75**		**0.00**	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**
TRUE	0.935	0.935	0.935		0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935
RP$_\rho$	0.921	**0.913**	**0.882**		**0.928**	**0.920**	**0.906**	**0.853**	**0.903**	**0.902**	**0.879**	**0.803**	0.851	0.835	**0.788**	0.640
RP	**0.922**	**0.913**	**0.882**		**0.930**	**0.921**	**0.906**	**0.858**	**0.922**	**0.903**	**0.883**	**0.811**	**0.893**	**0.841**	**0.799**	0.621
NAT13	**0.922**	0.908	0.878		0.918	0.909	0.890	0.839	0.899	0.892	0.862	0.794	0.863	0.837	0.784	**0.645**
ELK08	0.921	0.903	0.864		0.917	0.908	0.884	0.821	0.898	0.892	0.861	0.763	0.852	0.837	0.772	0.579
LIU16	0.498	0.549	0.550		0.550	0.500	0.550	0.505	0.550	0.550	0.550	0.549	0.503	0.512	0.550	0.550

Table C 3: Comparison of **error** for one-vs-rest MNIST (averaged over all digits) using a **CNN** classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if better (smaller) than non-RP models.

		π_1 = 0 ρ_1 = 0.5					π_1 = 0.25 ρ_1 = 0.25					π_1 = 0.5 ρ_1 = 0					ρ_1 = 0.5				
IMAGE	**TRUE**	**RP$_\rho$**	**RP**	**NAT13**	**ELK08**	**LIU16**	**RP$_\rho$**	**RP**	**NAT13**	**ELK08**	**LIU16**	**RP$_\rho$**	**RP**	**NAT13**	**ELK08**	**LIU16**	**RP$_\rho$**	**RP**	**NAT13**	**ELK08**	**LIU16**
0	0.0013	**0.0018**	**0.0023**	0.0045	0.0047	0.9020	**0.0017**	**0.0016**	0.0034	0.0036	0.9020	**0.0017**	**0.0016**	0.0031	0.0026	0.0029	**0.0021**	**0.0022**	0.0116	0.0069	0.9020
1	0.0015	**0.0022**	**0.0020**	0.0025	0.0034	0.8865	**0.0019**	**0.0019**	0.0035	0.0030	0.8865	0.0023	0.0020	0.0018	**0.0016**	0.0023	**0.0025**	**0.0025**	0.0036	0.0027	0.8865
2	0.0027	**0.0054**	**0.0049**	0.0057	0.0062	0.8968	**0.0032**	**0.0035**	0.0045	0.0051	0.8968	0.0030	0.0029	0.0031	0.0029	**0.0024**	**0.0059**	**0.0050**	0.0066	0.0083	0.8968
3	0.0020	**0.0032**	**0.0032**	0.0055	0.0038	0.8990	**0.0029**	**0.0029**	0.0043	0.0043	0.8990	**0.0021**	0.0027	**0.0023**	**0.0023**	0.0032	**0.0038**	**0.0042**	0.0084	0.0057	0.8990
4	0.0012	**0.0037**	0.0040	**0.0038**	0.0044	0.9018	**0.0029**	**0.0025**	0.0055	0.0069	0.9018	0.0026	0.0020	**0.0019**	0.0021	0.0030	**0.0044**	**0.0035**	0.0086	0.0077	0.9018
5	0.0019	**0.0032**	**0.0035**	0.0039	0.0038	0.9108	**0.0027**	**0.0031**	0.0062	0.0060	0.9108	**0.0021**	0.0024	0.0024	0.0028	**0.0023**	**0.0061**	**0.0056**	0.0066	0.0074	0.9108
6	0.0021	**0.0027**	**0.0028**	0.0053	0.0035	0.9042	**0.0028**	**0.0025**	0.0042	0.0036	0.9042	0.0029	0.0029	**0.0022**	0.0024	0.0028	**0.0032**	**0.0035**	0.0098	0.0075	0.9042
7	0.0026	**0.0039**	**0.0041**	0.0066	0.0103	0.8972	**0.0050**	**0.0052**	0.0058	0.0058	0.8972	0.0049	0.0040	**0.0030**	0.0037	0.0035	**0.0054**	**0.0064**	0.0113	0.0085	0.8972
8	0.0022	**0.0047**	**0.0043**	0.0106	0.0063	0.9026	**0.0034**	**0.0036**	0.0062	0.0091	0.9026	0.0036	**0.0030**	0.0035	0.0041	0.0032	**0.0044**	**0.0048**	0.0234	0.0077	0.9026
9	0.0036	0.0067	**0.0052**	0.0056	0.0124	0.8991	**0.0048**	**0.0051**	0.0065	0.0064	0.8991	0.0048	0.0050	0.0051	**0.0043**	0.0059	**0.0081**	0.0114	0.0131	**0.0112**	0.8991
AVG	0.0021	**0.0038**	**0.0036**	0.0054	0.0059	0.9000	**0.0031**	**0.0032**	0.0050	0.0054	0.9000	0.0030	**0.0028**	**0.0028**	0.0029	0.0032	**0.0046**	**0.0049**	0.0103	0.0074	0.9000

Table C 4: Comparison of **AUC-PR** for one-vs-rest MNIST (averaged over all digits) using a **CNN** classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models.

		π_1 = 0					π_1 = 0.25					π_1 = 0.5									
		ρ_1 = 0.5					ρ_1 = 0.25					ρ_1 = 0					ρ_1 = 0.5				
IMAGE	TRUE	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16
0	0.9998	**0.9992**	**0.9990**	0.9986	0.9982	0.5490	**0.9996**	**0.9996**	0.9986	0.9979	0.5490	**0.9989**	**0.9995**	0.9976	0.9979	0.9956	**0.9984**	**0.9982**	0.9963	0.9928	0.5490
1	0.9999	**0.9995**	**0.9995**	0.9976	0.9974	0.5568	**0.9996**	0.9993	**0.9995**	**0.9995**	0.5568	**0.9995**	**0.9998**	0.9982	0.9972	0.9965	**0.9995**	**0.9994**	0.9978	0.9985	0.5568
2	0.9994	**0.9971**	**0.9969**	0.9917	0.9942	0.5516	**0.9980**	**0.9977**	0.9971	0.9945	0.5516	**0.9988**	**0.9992**	0.9958	0.9934	0.9940	**0.9938**	**0.9947**	0.9847	0.9873	0.5516
3	0.9996	**0.9986**	**0.9987**	0.9983	0.9984	0.5505	**0.9991**	**0.9989**	0.9982	0.9980	0.5505	**0.9993**	**0.9994**	0.9991	0.9971	0.9974	**0.9969**	**0.9959**	0.9951	**0.9959**	0.5505
4	0.9997	0.9982	**0.9989**	0.9939	0.9988	0.0891	**0.9992**	**0.9991**	0.9976	0.9965	0.5491	**0.9994**	**0.9996**	0.9985	0.9978	0.9986	**0.9983**	**0.9977**	0.9961	0.9919	0.5491
5	0.9993	**0.9982**	**0.9976**	0.9969	0.9956	0.5446	**0.9986**	**0.9987**	0.9983	0.9979	0.5446	**0.9984**	**0.9982**	0.9971	0.9963	0.9929	**0.9958**	**0.9965**	0.9946	0.9934	0.5446
6	0.9987	**0.9976**	**0.9970**	0.9928	0.9931	0.5479	**0.9974**	**0.9980**	0.9956	0.9959	0.5479	**0.9968**	**0.9983**	0.9933	0.9950	0.9905	**0.9964**	0.9957	0.9942	**0.9961**	0.5479
7	0.9989	**0.9973**	**0.9972**	0.9965	0.9944	0.0721	0.9968	0.9973	0.9966	**0.9979**	0.5514	0.9969	**0.9983**	0.9961	0.9958	0.9974	**0.9933**	**0.9937**	0.9896	0.9886	0.5514
8	0.9996	**0.9974**	**0.9964**	**0.9964**	0.9946	0.5487	**0.9981**	**0.9981**	0.9973	0.9971	0.5487	**0.9983**	0.9988	0.9984	0.9976	**0.9989**	**0.9976**	**0.9975**	0.9873	0.9893	0.5487
9	0.9979	**0.9931**	**0.9951**	0.9901	0.9922	0.5504	**0.9935**	**0.9951**	0.9933	0.9920	0.5504	**0.9961**	**0.9951**	0.9924	0.9922	0.9912	**0.9877**	**0.9876**	0.9819	0.9828	0.5504
AVG	0.9993	**0.9976**	**0.9976**	0.9953	0.9957	0.4561	**0.9980**	**0.9982**	0.9972	0.9967	0.5500	**0.9983**	**0.9986**	0.9966	0.9960	0.9953	**0.9958**	**0.9957**	0.9918	0.9917	0.5500

Table C 5: Comparison of **F1 score** for one-vs-rest CIFAR-10 (averaged over all images) using a **logistic regression** classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models.

		π_1 = 0					π_1 = 0.25					π_1 = 0.5									
		ρ_1 = 0.5					ρ_1 = 0.25					ρ_1 = 0					ρ_1 = 0.5				
IMAGE	TRUE	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16
PLANE	0.272	**0.311**	**0.252**	0.217	0.220	0.182	**0.329**	**0.275**	0.222	0.224	0.182	**0.330**	**0.265**	0.231	0.259	0.0	**0.266**	**0.188**	0.183	0.187	0.182
AUTO	0.374	**0.389**	**0.355**	0.318	0.320	0.182	**0.388**	**0.368**	0.321	0.328	0.182	**0.372**	**0.355**	0.308	0.341	0.0	**0.307**	0.287	0.287	**0.297**	0.182
BIRD	0.136	**0.241**	0.167	0.143	0.136	**0.182**	**0.248**	**0.185**	0.137	0.137	0.182	**0.258**	**0.147**	0.100	0.126	0.0	**0.206**	0.153	0.132	0.150	**0.182**
CAT	0.122	**0.246**	0.170	0.141	0.150	**0.182**	**0.232**	0.163	0.112	0.127	**0.182**	**0.241**	**0.125**	0.068	0.103	0.0	**0.209**	0.148	0.119	0.157	**0.182**
DEER	0.166	**0.250**	**0.184**	0.153	0.164	0.182	**0.259**	0.175	0.146	0.163	**0.182**	**0.259**	**0.177**	0.126	0.164	0.0	**0.222**	0.162	0.132	0.164	**0.182**
DOG	0.139	**0.245**	0.174	0.146	0.148	**0.182**	**0.262**	0.171	0.115	0.126	**0.182**	**0.254**	**0.152**	0.075	0.120	0.0	**0.203**	0.151	0.128	0.137	**0.182**
FROG	0.317	**0.322**	**0.315**	0.289	0.281	0.182	**0.350**	**0.319**	0.283	0.299	0.182	**0.346**	**0.305**	0.239	0.279	0.0	**0.308**	0.252	0.244	**0.269**	0.182
HORSE	0.300	**0.300**	**0.299**	0.283	0.263	0.182	**0.334**	**0.313**	0.272	0.281	0.182	**0.322**	**0.310**	0.260	0.292	0.0	**0.275**	**0.258**	0.240	0.245	0.182
SHIP	0.322	**0.343**	**0.322**	0.297	0.272	0.182	**0.385**	**0.319**	0.287	0.289	0.182	**0.350**	**0.303**	0.250	0.293	0.0	**0.304**	**0.248**	0.230	0.237	0.182
TRUCK	0.330	**0.359**	**0.323**	0.273	0.261	0.182	**0.369**	**0.327**	0.293	0.290	0.182	**0.343**	**0.302**	0.278	0.299	0.0	**0.313**	0.246	0.252	**0.262**	0.182
AVG	0.248	**0.301**	**0.256**	0.226	0.221	0.182	**0.316**	**0.262**	0.219	0.226	0.182	**0.308**	**0.244**	0.194	0.228	0.000	**0.261**	0.209	0.195	**0.210**	0.182

Table C 6: Comparison of **error** for one-vs-rest CIFAR-10 (averaged over all images) using a **logistic regression** classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if better (smaller) than non-RP models. Here the logistic regression classifier severely underfits CIFAR, resulting in Rank Pruning pruning out some correctly labeled examples that "confuse" the classifier, hence in this scenario, RP and RP$_\rho$ generally have slightly smaller precision, much higher recall, and hence larger F1 scores than other models and even the ground truth classifier (Table C5). Due to the class inbalance ($p_{y1} = 0.1$) and their larger recall, RP and RP$_\rho$ here have larger error than the other models.

		π_1 = 0					π_1 = 0.25					π_1 = 0.5									
		ρ_1 = 0.5					ρ_1 = 0.25					ρ_1 = 0					ρ_1 = 0.5				
IMAGE	TRUE	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16
PLANE	0.107	0.287	0.133	0.123	**0.122**	0.900	0.177	0.128	**0.119**	0.123	0.900	0.248	0.124	0.110	0.118	**0.100**	0.202	0.147	**0.142**	0.160	0.900
AUTO	0.099	0.184	0.120	**0.110**	**0.110**	0.900	0.132	0.114	**0.105**	0.109	0.900	0.189	0.110	0.105	0.110	**0.100**	0.159	0.129	**0.125**	0.139	0.900
BIRD	0.117	0.354	0.148	0.133	**0.131**	0.900	0.217	0.135	**0.120**	0.125	0.900	0.277	0.135	0.115	0.123	**0.100**	0.226	0.147	**0.139**	0.158	0.900
CAT	0.114	0.351	0.138	**0.129**	**0.129**	0.900	0.208	0.139	**0.122**	0.125	0.900	0.303	0.132	0.114	0.122	**0.100**	0.225	0.151	**0.141**	0.158	0.900
DEER	0.112	0.336	0.143	**0.128**	0.130	0.900	0.194	0.135	**0.120**	0.122	0.900	0.271	0.133	0.118	0.126	**0.100**	0.209	0.150	**0.147**	0.161	0.900
DOG	0.119	0.370	0.150	**0.136**	0.138	0.900	0.205	0.142	**0.129**	0.132	0.900	0.288	0.135	0.120	0.128	**0.100**	0.229	0.154	**0.147**	0.168	0.900
FROG	0.107	0.228	0.128	**0.117**	**0.117**	0.900	0.155	0.124	**0.113**	0.115	0.900	0.228	0.118	0.110	0.116	**0.100**	0.167	0.137	**0.130**	0.142	0.900
HORSE	0.104	0.251	0.127	**0.114**	0.116	0.900	0.153	0.123	**0.110**	0.112	0.900	0.224	0.116	0.108	0.113	**0.100**	0.178	0.134	**0.129**	0.144	0.900
SHIP	0.112	0.239	0.134	**0.121**	0.126	0.900	0.160	0.131	**0.119**	0.123	0.900	0.236	0.122	0.113	0.120	**0.100**	0.193	0.145	**0.139**	0.159	0.900
TRUCK	0.106	0.210	0.130	**0.121**	0.122	0.900	0.145	0.125	**0.113**	0.117	0.900	0.213	0.121	0.108	0.117	**0.100**	0.165	0.142	**0.134**	0.150	0.900
AVG	0.110	0.281	0.135	**0.123**	0.124	0.900	0.175	0.130	**0.117**	0.120	0.900	0.248	0.125	0.112	0.119	**0.100**	0.195	0.144	**0.137**	0.154	0.900

Table C 7: Comparison of **AUC-PR** for one-vs-rest CIFAR-10 (averaged over all images) using a **logistic regression** classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models. Since $p_{y1} = 0.1$, here *Liu16* always predicts all labels as positive or negative, resulting in a constant AUC-PR of 0.550.

		π_1 = 0					π_1 = 0.25					π_1 = 0.5									
			ρ_1 = 0.5					ρ_1 = 0.25					ρ_1 = 0					ρ_1 = 0.5			
IMAGE	TRUE	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16
PLANE	0.288	0.225	0.224	0.225	0.207	**0.550**	0.261	0.235	0.225	0.217	**0.550**	0.285	0.251	0.245	0.248	**0.550**	0.196	0.171	0.171	0.159	**0.550**
AUTO	0.384	0.350	0.317	0.312	0.316	**0.550**	0.342	0.335	0.331	0.331	**0.550**	0.328	0.348	0.334	0.333	**0.550**	0.256	0.257	0.259	0.261	**0.550**
BIRD	0.198	0.160	0.169	0.166	0.161	**0.550**	0.188	0.185	0.179	0.177	**0.550**	0.186	0.173	0.174	0.175	**0.550**	0.150	0.154	0.150	0.147	**0.550**
CAT	0.188	0.164	0.175	0.174	0.175	**0.550**	0.163	0.169	0.168	0.170	**0.550**	0.148	0.156	0.154	0.152	**0.550**	0.145	0.143	0.140	0.145	**0.550**
DEER	0.215	0.161	0.177	0.180	0.183	**0.550**	0.194	0.180	0.180	0.182	**0.550**	0.174	0.175	0.176	0.175	**0.550**	0.151	0.152	0.146	0.151	**0.550**
DOG	0.188	0.162	0.161	0.165	0.155	**0.550**	0.175	0.160	0.161	0.158	**0.550**	0.173	0.169	0.162	0.164	**0.550**	0.145	0.142	0.139	0.133	**0.550**
FROG	0.318	0.246	0.264	0.262	0.258	**0.550**	0.292	0.277	0.272	0.273	**0.550**	0.276	0.274	0.277	0.277	**0.550**	0.239	0.212	0.206	0.212	**0.550**
HORSE	0.319	0.242	0.267	0.269	0.260	**0.550**	0.283	0.264	0.264	0.263	**0.550**	0.288	0.282	0.279	0.278	**0.550**	0.223	0.218	0.208	0.207	**0.550**
SHIP	0.317	0.257	0.267	0.271	0.248	**0.550**	0.296	0.266	0.267	0.259	**0.550**	0.279	0.268	0.259	0.262	**0.550**	0.220	0.212	0.207	0.191	**0.550**
TRUCK	0.329	0.288	0.261	0.271	0.263	**0.550**	0.298	0.275	0.286	0.284	**0.550**	0.289	0.272	0.276	0.277	**0.550**	0.241	0.213	0.208	0.204	**0.550**
AVG	0.274	0.226	0.228	0.229	0.223	**0.550**	0.249	0.235	0.233	0.231	**0.550**	0.243	0.237	0.234	0.234	**0.550**	0.197	0.187	0.183	0.181	**0.550**

Table C 8: Comparison of **error** for one-vs-rest CIFAR-10 (averaged over all images) using a **CNN** classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if better (smaller) than non-RP models.

		π_1 = 0					π_1 = 0.25					π_1 = 0.5									
			ρ_1 = 0.5					ρ_1 = 0.25					ρ_1 = 0					ρ_1 = 0.5			
IMAGE	TRUE	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16
PLANE	0.044	**0.054**	**0.057**	0.059	0.063	0.900	**0.050**	**0.051**	0.054	0.057	0.900	**0.048**	**0.045**	0.049	0.048	0.100	**0.063**	**0.061**	0.074	0.065	0.900
AUTO	0.021	**0.040**	**0.037**	0.041	0.043	0.100	**0.032**	**0.034**	0.040	0.039	0.900	0.028	**0.026**	**0.026**	**0.026**	0.100	**0.047**	**0.049**	0.062	0.070	0.900
BIRD	0.055	0.083	**0.078**	0.080	0.082	0.900	**0.074**	**0.074**	0.077	0.078	0.900	0.072	**0.066**	0.072	0.070	0.100	0.124	**0.084**	0.089	0.093	0.900
CAT	0.077	0.108	**0.091**	0.092	0.095	0.100	0.111	0.090	**0.086**	0.089	0.900	0.113	**0.084**	0.086	0.088	0.100	0.117	0.098	**0.094**	0.100	0.900
DEER	0.049	0.081	**0.078**	**0.078**	0.079	0.900	0.080	**0.069**	0.075	0.070	0.900	0.076	0.062	**0.061**	0.062	0.100	0.106	**0.086**	0.091	0.093	0.900
DOG	0.062	**0.075**	**0.071**	0.079	0.080	0.100	0.071	0.069	0.070	**0.067**	0.900	0.069	0.061	**0.057**	0.076	0.100	0.103	**0.081**	0.084	0.086	0.900
FROG	0.038	0.050	**0.048**	**0.048**	0.054	0.100	**0.047**	**0.052**	0.056	0.062	0.900	0.045	**0.040**	0.042	0.043	0.100	**0.058**	**0.062**	0.066	0.071	0.900
HORSE	0.035	**0.050**	**0.052**	0.057	0.054	0.900	**0.048**	**0.051**	0.052	0.057	0.900	0.045	**0.040**	0.042	0.046	0.100	**0.065**	**0.063**	0.066	0.075	0.900
SHIP	0.028	**0.042**	**0.042**	0.046	**0.042**	0.900	**0.037**	**0.036**	0.042	0.047	0.900	0.035	0.033	**0.031**	0.033	0.100	**0.051**	**0.049**	0.064	0.058	0.900
TRUCK	0.027	**0.044**	**0.046**	0.054	0.056	0.900	**0.034**	**0.032**	0.038	0.043	0.900	**0.034**	**0.031**	0.034	0.034	0.100	**0.060**	0.066	0.067	**0.065**	0.900
AVG	0.043	**0.063**	**0.060**	0.064	0.065	0.580	**0.059**	**0.056**	0.059	0.061	0.900	0.056	**0.049**	0.050	0.053	0.100	0.080	**0.070**	0.076	0.077	0.900

Table C 9: Comparison of **AUC-PR** for one-vs-rest CIFAR-10 (averaged over all images) using a **CNN** classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models.

		π_1 = 0					π_1 = 0.25					π_1 = 0.5									
			ρ_1 = 0.5					ρ_1 = 0.25					ρ_1 = 0					ρ_1 = 0.5			
IMAGE	TRUE	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16	RP_ρ	RP	NAT13	ELK08	LIU16
PLANE	0.856	0.779	0.780	**0.784**	0.756	0.550	**0.808**	**0.797**	0.770	0.742	0.550	**0.813**	**0.824**	0.792	0.794	0.550	**0.710**	**0.722**	0.662	0.682	0.550
AUTO	0.954	0.874	**0.889**	0.878	0.833	0.550	**0.905**	**0.900**	0.871	0.866	0.550	**0.931**	**0.927**	0.924	0.910	0.550	**0.824**	**0.814**	0.756	0.702	0.550
BIRD	0.761	0.559	0.566	**0.569**	0.568	0.550	**0.619**	**0.618**	0.584	0.597	0.550	**0.623**	**0.679**	0.613	0.619	0.115	0.465	0.492	0.436	0.434	**0.550**
CAT	0.601	0.387	0.447	**0.463**	0.433	0.550	0.423	0.454	0.487	0.480	**0.550**	0.483	**0.512**	0.493	0.473	0.050	0.373	0.375	0.382	0.371	**0.550**
DEER	0.820	**0.620**	0.600	**0.615**	0.573	0.550	0.646	**0.660**	0.610	0.657	0.550	0.658	**0.707**	0.700	0.703	0.550	0.434	0.487	0.414	0.435	**0.550**
DOG	0.758	**0.629**	**0.662**	0.617	0.573	0.550	**0.673**	**0.667**	0.658	0.660	0.550	0.705	0.722	**0.741**	0.705	0.550	0.541	0.545	0.496	0.519	**0.550**
FROG	0.891	0.812	**0.815**	0.812	0.776	0.550	**0.821**	**0.827**	0.808	0.749	0.550	**0.841**	**0.851**	0.828	0.831	0.550	**0.753**	**0.710**	0.691	0.620	0.550
HORSE	0.897	**0.810**	**0.817**	0.799	0.779	0.550	**0.824**	**0.809**	0.801	0.772	0.550	**0.826**	**0.844**	0.818	0.819	0.550	**0.736**	**0.699**	**0.699**	0.600	0.550
SHIP	0.922	**0.870**	0.862	**0.864**	0.853	0.550	**0.889**	**0.885**	0.843	0.848	0.550	0.889	**0.897**	0.891	0.887	0.550	**0.800**	**0.808**	0.767	0.741	0.550
TRUCK	0.929	**0.845**	**0.848**	0.824	0.787	0.550	**0.887**	**0.894**	0.873	0.853	0.550	**0.904**	**0.902**	0.898	0.883	0.550	**0.740**	**0.709**	0.695	0.690	0.550
AVG	0.839	0.719	**0.729**	0.722	0.693	0.550	**0.750**	**0.751**	0.730	0.722	0.550	0.767	**0.787**	0.770	0.762	0.457	**0.637**	**0.636**	0.600	0.579	0.550