Training with Low-Label-Quality Data: Rank Pruning and Multi-Review

Yue Xing [1] **Ashutosh Pandey** [2] **David Yan** [2] **Fei Wu** [2] **Michale Fronda** [2] **Pamela Bhattacharya** [2]

Abstract

Inaccurate labels in training data is a common problem in machine learning. Algorithms have been proposed to prune samples with label noise (i.e., samples are far from the decision boundary but still the label is inaccurate); training models on such samples could cause poor model performance. However, in many real applications, there exist samples around the decision boundary that are inherently difficult to label, leading to label error. Such samples are important for model training because of their high learning value. Existing pruning algorithm do not differentiate between samples with label noise and label error, therefore prunes both kinds of samples. This paper improves an existing pruning algorithm in two ways: it (a) prunes noisy samples and high-confidence samples (with less learning value), and (b) preserves the samples (potentially) with label error that have a high learning value and gets accurate labels for them (using multiple reviews). Our evaluation using publicly available and Meta internal de-identified and aggregated data sets shows that the combination of these ideas improve the baseline pruning algorithm.

1. Introduction

Training data with inaccurate labels is a challenge to building high-quality machine learning models that generalize well to unseen data (Zhou, 2018). A low-quality model could have negative consequences for a domain, such as integrity on a social media platform, where false negatives or false positives can eventually lead to users leaving the platform. For example, on Meta, a scammer missed by the model can cause financial harm to good users. On the other hand, a good user banned by the model will lead to a poor user experience. Good users will be discouraged from engaging with the platform in both cases.

While low label quality can deteriorate machine learning performance (Zlateski et al., 2018; Kazai et al., 2013; Alonso, 2015), it is also harmful in real applications where actions are taken based on the labels, e.g., (Sameera et al., 2021). Researchers have proposed approaches to improve the robustness of machine learning models in the presence of inaccurate labels. For example, (Zhang and Sabuncu, 2018) improves the cross entropy loss to handle noisy labels, and (Müller et al., 2019; Zhou et al., 2021) uses soft labels to replace hard labels to smooth Y. Some researchers also consider manipulating the samples, e.g., pruning inaccurate samples or fixing potentially incorrect labels (Northcutt et al., 2017; Ding et al., 2018).

The existing work on pruning samples mostly deals with noisy labels, which are (randomly) flipped with respect to the ground truth despite the classes being well separated. We refer to this phenomenon as *label noise* where the same reviewer labels the sample as positive and negative when presented multiple times. (Kahneman et al., 2022) refers to this phenomenon as within-person noise. Noisy samples might not be difficult to label because they are far from the decision boundary yet reviewers label them inaccurately. Training on such noisy samples could lead to a poor-quality model. (Northcutt et al., 2017) proposed the rank pruning algorithm to prune noisy samples from the training process. Specifically, the paper provides the theoretical framework to identify samples to remove from the training data.

However, for many realistic domains, in addition to noisy training samples, there might be the samples with inaccurate labels when the labeling decision is difficult because the samples are close to the decision boundary. Consequently, the same sample is labeled as positive by some reviewers and negative by others. We refer to this phenomenon as *label error*. (Kahneman et al., 2022) refers to this phenomenon as between-person noise. The samples close to the decision boundary should not get pruned because they help a trained model learn the decision boundary. Unfortunately, the pruning algorithm (proposed by (Northcutt et al., 2017)) prunes such samples (besides noisy samples) because it does not differentiate between label noise and error when pruning

[1]Department of Statistics and Probability, Michigan State University. This work is done during Yue Xing's internship at Meta Platforms, Inc. as a Ph.D. student at Purdue University. [2]Meta Platforms, Inc.. Correspondence to: Yue Xing <xingyue1@msu.edu>, Ashutosh Pandey <ashutoshp@meta.com>.

DMLR Workshop at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

(potentially) suspicious/inaccurately-labeled samples.

Our work improves the rank-pruning algorithm in two ways. First, the improved algorithm does not prune all the *suspicious* samples, particularly the ones close to the decision boundary because they have a high learning value.[1] To this end, the improved algorithm learns the hyperparameters (using the training data) to identify the samples with high-learning values, and therefore, they are not pruned. Second, in addition to noisy samples, the algorithm prunes *confident* samples that do not provide learning value; as detailed later, confident samples are ones that can be classified with high accuracy by a trained model. Having too many confident samples in training could distract the model from learning the decision boundary(Katharopoulos and Fleuret, 2018).

Additionally, the paper shows that the performance of the (improved) rank pruning algorithm can be further improved as the erroneous sample count decreases in the training data. To reduce erroneous samples, each sample is labeled by multiple reviewers, and a majority vote decides the final label. Our formal analysis shows this labeling approach helps get the correct labels, thereby reducing erroneous samples by canceling the individual-level error, as also suggested by (Surowiecki, 2005).

To distinguish between our contributions and existing works, throughout this paper, we use "Theorem" to denote our new results, and "Proposition" for the existing results.

1.1. Other Related Works

This section presents the literature in various areas which are related to this work.

Robust Algorithms against Noisy Datasets In this paper, we mainly focus on improving the rank-pruning algorithm in (Northcutt et al., 2017). Besides the rank-pruning algorithm, from the literature, e.g., (Song et al., 2022; Nigam et al., 2020), there are many other ways to deal with (potentially) inaccurate labels:

(1) Modify the loss function. Some papers consider changing the loss function to adapt to the potential change in the label. For example, (Müller et al., 2019; Zhou et al., 2021) uses soft labels to replace hard labels to smooth Y, and (Zhang and Sabuncu, 2018) improves the cross entropy loss to handle label error.

(2) Apply weights to different samples. Some studies apply different weights to different samples to improve the training process. For instance, (Chang et al., 2017) improves the training process by emphasizing high-variance samples around the decision boundary.

(3) Change the samples. Besides changing the loss function or applying weighting schemes, some other works consider modifying the possibly incorrect labels, and others consider pruning samples with incorrect labels. (Northcutt et al., 2017; 2021) and (Pleiss et al., 2020) use rank-pruning methods to identify samples with random noise. (Li and Gao, 2019) proposes another clustering algorithm to identify incorrect labels. (Ding et al., 2018; Algan and Ulusoy, 2021) replace labels with the model prediction to smooth the training process. (Cui et al., 2020) corrects labels by studying label correlations in multi-task learning.

However, as far as we are aware, none of these approaches differentiate between noisy and erroneous samples. Moreover, this paper provides a theoretical analysis showing that multiple review helps in reducing label error that eventually leads to performance gains for a model.

Importance Sampling While the rank pruning algorithm in (Northcutt et al., 2017) motivates us to study the erroneous labels, the importance-sampling approach in (Katharopoulos and Fleuret, 2018) inspires us to also prune confident samples, i.e., the samples which can be easily learned by the machine learning model.

Importance sampling is a commonly used method to adjusting the weights for the samples in order to reduce estimation variance (Glasserman et al., 2000). Based on this idea, (Katharopoulos and Fleuret, 2018) proposes to apply larger weights for the samples around the decision boundary when using neural networks for classification tasks. Through this way, the training process can focus more on the uncertain samples, i.e., samples with a higher learning value. Related studies can be found in (Katharopoulos and Fleuret, 2017; Nabian et al., 2021; Daw et al., 2022; Ariafar et al., 2021; Arazo et al., 2021).

Different from (Katharopoulos and Fleuret, 2018), we utilize the framework of rank pruning to prune confident samples. The rank pruning algorithm categorizes suspicious samples and confident samples, which is more flexible. proposed algorithm In the applications of computer vision, i.e., there is a clear decision boundary and the Bayes classifier achieve 100% accuracy, it is efficient to train only using samples around the model decision boundary.

Besides the studies mentioned before in robust algorithms against label noise, others, e.g., (Hwang et al., 2022), justify that human label can be worse than pseudo-labels.

Dynamic Hyper-parameter Tuning There are various ways to do hyper-parameter tuning.

For example, (Li et al., 2017) proposed to use a bandit algorithm to efficiently do hyper-parameter tuning. In the

[1] Suspicious samples include both noisy and erroneous samples. We hypothesize that the algorithm helps to prune most of the noisy samples and a few erroneous samples.

beginning, a bench of candidate parameters is used to do the training, and then they drop the candidates with poor performance step-by-step.

In another study, (Shang et al., 2019) further simplifies the algorithm of (Li et al., 2017) with the help of Bayesian optimization. They design a distribution that randomly outputs a candidate parameter. Each time they select a candidate parameter, evaluate the reward, and update the distribution to assign a higher probability to the specific candidate if the reward is high.

2. Notation and Model Setup

Before describing the proposed method, we first introduce some essential mathematical notations to be used in the data generation model and the algorithms.

In this paper, we consider binary classification to classify between 0 and 1. The scenario of erroneous label is considered as follows: Denote the ground-truth probability as $g_0(x) := P(Y = 1 \mid X = x)$ for x in the support $\mathcal{D}$, and the Bayes classifier is $f_0(x) = 1\{g_0(x) > 1/2\}$. Mathematically, **label error** refers to the case where g_0 takes values **other than** 0 or 1.

In terms of **label noise**, instead of observing the true label Y, we observe a perturbed label Z, where the flipping probabilities are $P(Z = 1 \mid Y = 0) = \rho_0$ and $P(Z = 0 \mid Y = 1) = \rho_1$ for some constants $\rho_0, \rho_1 > 0$ which are independent of x. The flipping of labels with probabilities (ρ_0, ρ_1) is what is referred to as label error in this paper. We also denote $g(x) := P(Z = 1 \mid X = x)$.

Besides the above definitions, we also follow (Northcutt et al., 2017) to define the probability of Z being 1 as $p_{z1} = P(Z = 1)$. The conditional probability of the hidden true label Y given the observed label Z are $\pi_1 = P(Y = 0 \mid Z = 1)$ and $\pi_0 = P(Y = 1 \mid Z = 0)$ respectively.

In practice, it is impossible to obtain the ground-truth g_0 or even the perturbed version g. One could only obtain estimated model score $\widehat{g}$ to estimate g from the noisy training data. Denote $\Delta g(x) = \widehat{g}(x) - g(x)$.

3. Robust Algorithm Against Label Noise and Label Error

In this section, we present the intuition of the existing pruning algorithm and our proposed adaptations.

3.1. Intuition Behind the Robust Pruning Algorithm

Figure 1 illustrates the pruned suspicious and confident samples. The pruned suspicious samples are the ones that have the label and the trained model prediction differ a lot, e.g., a negative sample with a prediction score of 0.95 where the score varies between 0 and 1 such that a higher value indicates higher chances of a label being positive.

Among suspicious samples in the training data, the samples with random noise are the ones that have their label and trained model prediction differ a lot, e.g. a negative sample with a prediction score of 0.95, where the score varies between 0 and 1 such that a higher value indicates higher chances of a label being positive. Confident samples refer to those whose model predictions can achieve a high accuracy, e.g. a negative sample with a prediction score of 0.05. A graphical illustration is in Figure 1.

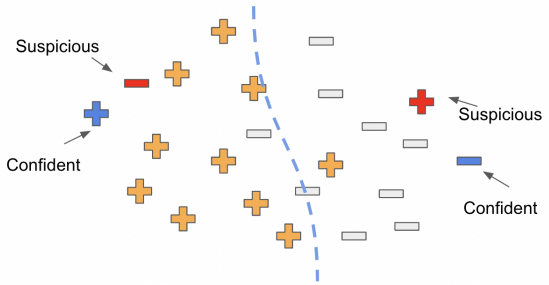

Figure 1. Suspicious and confident samples. Blue dashed curve: the decision boundary of the trained machine learning model.

As a result, if we prune the suspicious samples and confident samples, the other samples will be those which are on average closer to the decision boundary. Training can then focus more on learning the decision boundary.

3.2. Algorithms

Our algorithms are shown in Algorithm 1 for the main algorithm and Algorithm 2 for the details of rank pruning. The idea is that, during the training, if the main algorithm is accurate enough, one can use Algorithm 2 to prune the suspicious and confident samples.

Algorithm 1 Main Algorithm

Input: Training data, number of iterations T, other parameters in the base algorithm.
for $t = 1$ **to** T **do**
  Calculate the predicted score for each sample.
  Calculate the gradient (and Hessian if needed).
  Calculate the accuracy Acc_t of the current batch.
  if $Acc_t > \theta$ **then**
    Use the rank-pruning algorithm with (α, β) to remove samples in the current iteration.
  end if
  Update the model using the gradient (and Hessian if needed).
end for
Output: Output model.

Algorithm 2 Rank Pruning

Input: Training data, suspicious threshold rate α, confident threshold rate β, the predicted score function at iteration t g_t. Denote P and N as the set of samples with positive/negative labels respectively. Take $p_{z1} = |P|/(|P| + |N|)$ as the proportion of the positive Z in the data set.
Calculate

$$LB_{y=1} = \mathbb{E}_{x\in P}[g_t(x)], \quad UB_{y=0} = \mathbb{E}_{x\in N}[g_t(x)].$$

Further split N and P by the model score, i.e., calculate

$$N_{y=1,t} = \{x \in N \mid g_t(x) \geq LB_{y=1}\}, \qquad P_{y=1,t} = \{x \in P \mid g_t(x) \geq LB_{y=1}\},$$
$$N_{y=0,t} = \{x \in N \mid g_t(x) \leq UB_{y=0}\}, \qquad P_{y=0,t} = \{x \in P \mid g_t(x) \leq UB_{y=0}\}.$$

Construct the estimate of $P(Z = 0 \mid Y = 1)$ and $P(Z = 1 \mid Y = 0)$ (ρ_1 and ρ_0) for this tth iteration via

$$\widehat{\rho}_{1,t} = \frac{|N_{y=1,t}|}{|N_{y=1,t}| + |P_{y=1,t}|}, \quad \widehat{\rho}_{0,t} = \frac{|P_{y=0,t}|}{|P_{y=0,t}| + |N_{y=0,t}|} \tag{1}$$

Construct the estimate of $P(Y = 0 \mid Z = 1)$ and $P(Y = 1 \mid Z = 0)$ (π_1 and π_0) for this tth iteration via

$$\widehat{\pi}_{1,t} = \frac{\widehat{\rho}_{0,t}}{p_{z1}} \frac{1 - p_{z1} - \widehat{\rho}_{1,t}}{1 - \widehat{\rho}_{1,t} - \widehat{\rho}_{0,t}}, \quad \widehat{\pi}_{0,t} = \frac{\widehat{\rho}_{1,t}}{p_{z1}} \frac{p_{,t} - \widehat{\rho}_{0,t}}{1 - \widehat{\rho}_{1,t} - \widehat{\rho}_{0,t}}. \tag{2}$$

Remove suspicious samples: Remove $\alpha\widehat{\pi}_{1,t}|P|$ samples in P with the least g_t, and $\alpha\widehat{\pi}_{0,t}|N|$ in N with the largest g_t.
Remove samples with less information: Remove $\beta(1-\widehat{\pi}_{1,t})|P|$ samples in P with the largest g_t. Remove $\beta(1-\widehat{\pi}_{0,t})|N|$ samples in N with the least g_t.
Denote $n_{p,t}$ and $n_{n,t}$ as the number of remaining samples in the batch. Take weight $w_{p,t}$ and $w_{n,t}$ for the remaining positive/negative samples so that $n_{p,t}w_{p,t}/(n_{n,t}w_{n,t}) = |P|/|N|$.
Output: Samples which will be removed.

Pruning for Suspicious Samples The full algorithm of rank pruning is in Algorithm 2. Similar to (Northcutt et al., 2017), we classify samples into four categories based on the model score and the label (positive samples with high scores, positive samples with low scores, etc.), and calculate $\widehat{\rho}_{1,t}$, $\widehat{\rho}_{0,t}$, $\widehat{\pi}_{1,t}$, and $\widehat{\pi}_{0,t}$ to get the estimate of $P(Y = 0 \mid Z = 1)$ and $P(Y = 1 \mid Z = 0)$, i.e., the probability of the true label being different to the observed label. Based on these estimations, we order positive and negative samples respectively based on the model scores, and prune the positive samples with smallest scores and the negative samples with the largest scores.

In Algorithm 2, it is intuitive to construct the estimate of the flipping probability $\rho_0 = P(Z = 1 \mid Y = 0)$ and $\rho_1 = P(Z = 0 \mid Y = 1)$ as in (1). In terms of (2) for $\pi_1 = P(Y = 0 \mid Z = 1)$ and $\pi_0 = P(Y = 1 \mid Z = 0)$, we present the derivation in Appendix A.2.

A difference to (Northcutt et al., 2017) is that, instead of pruning all $\widehat{\pi}_{1,t}|\widetilde{P}_t|$ positive samples with small scores and $\widehat{\pi}_{0,t}|\widetilde{N}_t|$ negative samples with large scores, we take the pruning rate as $\alpha\widehat{\pi}_{1,t}$ ($\alpha\widehat{\pi}_{0,t}$) for some $\alpha \in (0, 1)$ determined by some hyper-parameter tuning methods, e.g., cross validation, or Bayes approaches.

Pruning for Confident Samples Since the rank pruning algorithm (Northcutt et al., 2017) prunes samples with label noise, we can also take one more step to prune some confident samples. Based on (Katharopoulos and Fleuret, 2018), the data around the decision boundary help the model learn better, so removing confident samples can also help the model focus more around the decision boundary.

In Algorithm 2, different from the suspicious samples in which we want to prune all those with label noise, for confident samples, we cannot prune all of them because they still provide important information that helps train the model. In the experiments, we use hyper-parameter tuning to select β.

Remark 1. *Since the proposed method involves (α, β), one may question why we still need the estimate of (π_0, π_1) instead of directly pruning αn samples with random noise and βn confident samples. There are two main reasons. First, although one can directly tune α and β without considering (π_0, π_1), the existence of (π_0, π_1) simplifies the tuning of (α, β) and we only need to search in $[0, 1]$. Second, based on (Northcutt et al., 2017), the estimation of (π_0, π_1) is robust, which could further ensure the robustness of our algorithm.*

4. Multiple-Reviewed Samples

In this section, instead of being labeled only once, we consider a multi-review process so that each sample is reviewed by at least two reviewers. Using such an approach, we can get more accurate final labels. We study how the proposed pruning method is affected by this multi-review process.

4.1. Illustration of the Multi-Review Process

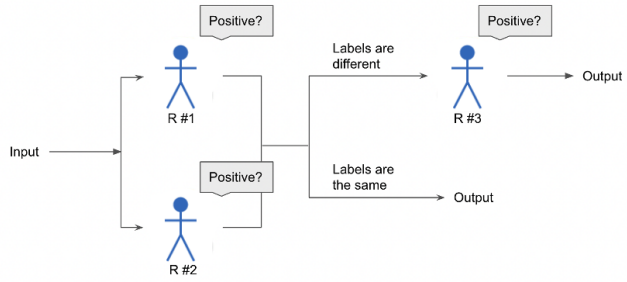

Figure 2. A Simple Multi-Review Process.

Figure 2 illustrates our multiple-review process. For each sample, two reviewers are firstly assigned to the sample and give their labels. If these two labels disagree with each other, a third reviewer will be assigned and label the sample. As a result, the possible outcomes are: (1) two same labels, or (2) three labels (2 positive and 1 negative or vise versa). We use the majority vote as the final outcome.

One can still apply Algorithm 1 to conduct pruning when training the model using multi-review data. However, when using the formulas in Algorithm 1 to determine the pruning thresholds, the values of UB, LB will get changed because the data distribution gets changed. To differentiate these quantities for single-review and multi-review processes, we add "$\widetilde{\ }$" in above the notations for multi-review data, i.e., P vs $\widetilde{P}$, ρ_0 vs $\widetilde{\rho}_0$, and π_0 vs $\widetilde{\pi}_0$, etc. For example, given the true label Y, ρ_0, ρ_1 represents the flipping probabilities for an individual noisy label. Since every reviewer can flip/not flip the label, $\widetilde{\rho}_0$ and $\widetilde{\rho}_1$ represent the final probability of whether the aggregated final label is flipped or not in the multi-review process.

4.2. Effect of Multiple-Reviewed Samples on the Pruning Algorithm

In this section, we show that the multi-review process can further improve the original rank pruning algorithm in addition to our algorithm adaptations in Section 3.

The analysis is hard when directly considering complicated scenarios, i.e., label error occurs, or the estimate $\widehat{g}$ involves estimation error. To simplify the understanding, we start from the **ideal case** with no label error and estimation error, i.e., $g_0 \in \{0, 1\}$ and $\widehat{g} \equiv g_0$, and then we extend to other cases, i.e., the case with estimation error but no label error (**non-ideal, non-overlap case**), and the case with label error but no estimation error (**overlap case**). We show that the multi-review process can further improve the robustness of the rank pruning algorithm. Given all these improvements, we conclude that the rank pruning algorithm can benefit from the multi-review process.

Ideal Case We first consider the case where g_0 is only 0 or 1 and $\widehat{g} \equiv g$. In this case, there are several changes to the data distribution and the training process.

First, the following theorem shows that a larger proportion of samples have a more accurate label with less label error.

Theorem 1. *Assume $\rho_0 = \rho_1 = \rho$, then $\widetilde{\rho} < \rho$.*

To prove Theorem 1, assume for each single review, $\rho_0 = \rho_1 = \rho$ for some $\rho \in [0, 0.5)$, then the probabilities of the sample being reviewed by two/three reviewers become

$$P(\text{Three reviews}) = 2\rho(1-\rho),$$
$$P(\text{Two reviews}) = \rho^2 + (1-\rho)^2.$$

Assume the hidden true label $Y = 1$. If a sample has three reviews, since the final label is now determined by the third review, there is still ρ probability that the final label is 0. However, when there are only two reviews, the flipping probability of the final label gets changed. There is $\rho^2/(\rho^2 + (1-\rho)^2)$ probability of having two labels 0, which is smaller than the original ρ. The overall flipping probability $\widetilde{\rho}$ for the final label therefore is smaller.

Second, the consequence of a smaller $\widetilde{\rho}$ is that, Algorithm 2 has a more stable $\widetilde{P}_{y=1}$ as the variance of $1\{z \in \widetilde{P}_{y=1}\}$ is smaller. We omit the subscript t for $\widetilde{P}_{y=1}$ to simplify the notation. If we further assume $P(Y = 1) = 0.5$, then

$$P(z \in \widetilde{P}_{y=1} \mid \text{Three reviews}) = \frac{2\rho(1-\rho)^2}{2\rho(1-\rho)},$$

$$P(z \in \widetilde{P}_{y=1} \mid \text{Two reviews}) = \frac{(1-\rho)^2}{\rho^2 + (1-\rho)^2},$$

where the second one $> 1 - \rho$.

Recall that for a random variable ξ following Bernolli distribution, the variance is $(1-\mathbb{E}\xi)\mathbb{E}\xi$. Thus, the binary variable $1\{z \in \widetilde{P}_{y=1}\}$ has an expectation farther from 0.5, and the estimation variance becomes smaller, increasing the stability of the algorithm. Similar case happens for $\widetilde{N}_{y=0}$ and others.

Non-ideal, Non-Overlap Case We now consider the case where $\widehat{g}$ is not identical to g. We still impose the non-overlap condition $g_0 \in \{0, 1\}$.

Based on (Northcutt et al., 2017), when $\widehat{g}$ is accurate enough, one can still obtain an accurate estimate of $\widehat{\rho}_{i,t}$ for $i = 0, 1$ when $g_0 \in \{0, 1\}$:

Proposition 1 (Theorem 4 of (Northcutt et al., 2017))**.** *Recall that $\Delta g(x) = \widehat{g}(x) - g(x)$ and (N, P) are the sets of samples with negative/positive Z respectively. Assume $g_0(x) \in \{0, 1\}$ for all $x \in \mathcal{D}$, in the single-review process,*

If $\forall x \in N$, $\Delta g(x) < LB_{y=1} - \rho_0$, then $\widehat{\rho}_{1,t} = \rho_1$.

If $\forall x \in P$, $\Delta g(x) > -(1 - \rho_1 - UB_{y=0})$, then $\widehat{\rho}_{0,t} = \rho_0$.

Based on Proposition 1, the pruning algorithm outputs robust $\widehat{\rho}_{1,t}$ and $\widehat{\rho}_{0,t}$, tolerant to the estimation error in $\widehat{g}$.

Denote $\Gamma = \max(|LB_{y=1} - \rho_0|, |1 - \rho_1 - UB_{y=0}|)$ as the tolerance, i.e., if the estimation error is below Γ, then the pruning algorithm gives the correct pruning thresholds. We consider how the tolerance is affected when using multi-review data and get the result below:

Theorem 2. *Denote Γ as the tolerance for single-review data and $\widetilde{\Gamma}$ as the one for multi-review data. Assume $\rho_0 = \rho_1 = \rho$ for some $\rho \in [0, 0.5)$ and $p_{z1} = 0.5$, then $\widetilde{\Gamma} < \Gamma$.*

The original ranking pruning algorithm is robust in the sense that, when $\widehat{g}$ involves small errors, the estimate $\widehat{\rho}_0$ and $\widehat{\rho}_1$ are still accurate (see Proposition 1). The tolerance Γ quantifies the robustness. When $|\Delta g(x)| \leq \Gamma$ for any x, we have $\widehat{\rho}_0 = \rho_0$ and $\widehat{\rho}_1 = \rho_1$. Based on Theorem 2, when using multi-review data, the algorithm becomes more robust.

Overlap Case Based on (Northcutt et al., 2017), when there is overlap between P and N, UB will be overestimated and LB will be underestimated:

Proposition 2 (Lemma 3 in (Northcutt et al., 2017))**.** *Assume $g = \widehat{g}$, in the single-review process,*

$$\begin{aligned} LB_{y=1} &= LB^*_{y=1} - \frac{(1 - \rho_0 - \rho_1)^2}{p_{z1}} \Delta p_0, \\ UB_{y=0} &= UB^*_{y=0} + \frac{(1 - \rho_0 - \rho_1)^2}{1 - p_{z1}} \Delta p_0, \end{aligned}$$

where $LB^*_{y=1} = (1 - \rho_1)(1 - \pi_1)$, $UB^*_{y=1} = (1 - \rho_1)\pi_0 + \rho_0(1 - \pi_0)$ and they are the correct decision thresholds, and $\Delta p_0 = |P \cap N|/|P \cup N|$.

When using multi-review data, we have the following result:

Theorem 3. *Assume $\rho_0 = \rho_1$ and $p_{z1} = 0.5$, then*

$$\frac{(1 - \rho_0 - \rho_1)^2}{p_{z1}} \Delta p_0 > \frac{(1 - \widetilde{\rho}_0 - \widetilde{\rho}_1)^2}{\widetilde{p}_{z1}} \Delta \widetilde{p}_0,$$

i.e., the difference between $\widetilde{LB}_{y=1}$ and $\widetilde{LB}^*_{y=1}$ is smaller than the difference between $LB_{y=1}$ and $LB^*_{y=1}$. A similar result holds for $\widetilde{UB}_{y=0}$.

Based on Theorem 3, using multi-review data, the pruning algorithm becomes more robust even when $g_0 \in [0, 1]$.

Remark 2. *We do not consider the non-ideal but overlap case, i.e., $\widehat{g} \neq g$ but $g_0 \in [0, 1]$. Since multi-review data requires more labels, it is more expensive than the single-review data. When the number of samples are the same, it is obvious that multi-review data lead to better $\widehat{g}$ and are more preferred. However, if the number of total reviews is fixed, it is hard to track the change in $\widehat{g}$. Differences in model assumptions and machine learning algorithms could affect the performance of $\widehat{g}$.*

5. Experiments

In this section, we conduct numerical experiments for some public datasets to verify the effectiveness of the proposed algorithm and the effect of multi-review data.

5.1. Data Description

We use HTRU2 in the experiments (Dua and Graff, 2019). For HTRU2, we use the eight numerical features. It contains pulsar candidates collected during the High Time Resolution Universe Survey. There are 17,898 total examples.

In the experiments, we study both single-review and multi-review. Since it is expensive in practice to do multiple review for all samples, we control the total number of reviews in the datasets. For example, we set the total number of reviews to be N, then use the multiple review process to get new samples. Each sample may consume either 2 or 3 reviews, and the final number of rows in the data set is less than $N/2$. We only add label noise in the training set.

To imitate label noise, we take $\rho_1 = \rho_0 = \rho = 0.45$ and perturb the labels for positive and negative samples, and repeat the experiment 30 times to get the mean and standard deviation of the misclassification rate.

5.2. Observations

We first use the HTRU2 dataset to compare the performance of the vanilla algorithm, the rank pruning algorithm in (Northcutt et al., 2017) (Prune all), and the proposed algorithm. We provide Prune (sus) which intends to prune suspicious samples (mostly) with random noise, and Prune (sus, conf) to prune both suspicious and confident samples. The results are in Table 1.

There are several observations. First, the proposed method is better than the others, verifying the effectiveness of the proposed algorithm. Second, when there are label noise and estimation error involved, directly pruning all suspicious samples may lead to a performance even worse than the vanilla algorithm. It is therefore essential not to prune all suspicious samples. Finally, while the proposed algorithm

	Mean				Std			
	Vanilla	Prune all	Prune (sus)	(sus, conf)	Vanilla	Prune all	Prune (sus)	(sus, conf)
Single label	0.06215	*0.07304*	**0.05909**	**0.05909**	0.04282	0.04770	0.03855	0.03855
Multi label	0.04347	*0.07341*	0.04167	**0.03838**	0.01867	0.07666	0.01953	0.01896

Table 1. HTRU2: The mean and variance of the error rate in the testing data set. Training size: 5000. XGBoost.

Human label	Count	True scam	True not scam
Scam	50	17 (34%)	33 (66%)
Not scam	50	42 (84%)	8 (16%)

Table 2. The human label quality of suspicious samples.

works with single-review data, it also works in the multi-review data, and it improves more when using multi-review data.

Remark 3. *From the way of imitating multi-review process in the experiments, we are adding different label noise given a label Y, rather than adding a multi-review process starting from $P(Y = 1 \mid X = x)$. The impact is limited in HTRU2 data set because the evaluation error is very low.*

In terms of the theory, the difference of the multi-review process does not change the claim of the theoretical results.

6. Real-World Data Analysis

6.1. Data Description

We use Meta scammer detection data as a real-world example. Because of the large volume of review tasks, the scammer detection data contains both noisy and erroneous labels.

Label Quality In terms of the quality of this data set, it has samples with label noise. When investigating reviews from human reviewers, we found some examples of arbitrary mistakes. In addition, the data set contains label error. Perhaps due to reviewer's unfamiliarity with policy in corner cases or insufficient user information, different reviewers may make different decisions for the same review object.

Although our aim is to use high-quality labels to improve machine learning models, we also want to emphasize that high-quality labels themselves are very important. A user will be banned if the user is identified as a scammer. If a false-positive user is banned, this is harmful to the business of both the individual and Meta.

In a preliminary study, we pick 100 training samples whose label and model score differ a lot, i.e., negative samples with scores ≥ 0.9 and positive samples with scores ≤ 0.1, and had them reviewed by internal experts. The results are summarized in Table 5. One can see that pruning suspicious samples is an applicable method for this dataset.

Multi-review process In terms of the multi-review process for this data set, there is a slight difference to the multi-review process in Section 4. In Section 4, we start with two reviewers to simultaneously review the sample. In production, we adjust the process to reduce the review consumption.

Figure 3 shows the multi-review process in production. After the first reviewer labels the sample, we applied another ML model to determine whether the human label is likely correct. If the ML model is not confident, the sample will be passed to the second reviewer. A third reviewer will label the sample if the two human labels are inconsistent. Based on the above multi-review process, a sample can have one, two, or three labels. For evaluation, given the limited review capacity, we (a) applied multiple reviews only on samples labeled as positive by the first reviewer, with the intent to focus on preventing false positives, and (b) not all samples with a positive first label were are sent for multiple reviews.

Data Summary Below is some detailed information about the dataset in this experiment.

There are around 1000 features in the data set, including categorical features (e.g., country) and various score features based on the contents of the user's profile. After expanding the categorical features into dummy variables, there are around 3000 numerical features.

In the data set, there are total 150k samples.

(1) Multi-review data: There are 62323 samples collected from the multi-review process. In this set, 7055 samples have one label, 44335 have two labels. All the 62323 samples have a positive first label, and 4180 have a final negative label.

(2) Single-review data: Most samples in production only have one review, and we use a small proportion in this experiment. In this set, there are total 87677 samples. After combining all of the multi-review and single-review data, we adjust the number of negative samples so that the positive/negative ratio is around 0.5. There are negative 61239 samples in the single-review data.

Training Details We use XGBoost to train the model and use hyper-parameter tuning to select the best parameters. We also use a multi-review process. There are some differences to the process mentioned in Section 4 to save human review

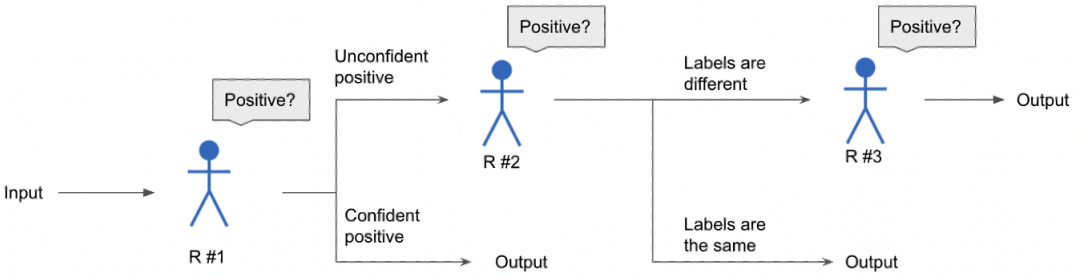

Figure 3. The new multi-review process in production.

capacity.

Experiment Results We run evaluation on two datasets:

(1) Single review data: an evaluation dataset with single-review samples. The positive/negative ratio is around 0.5.

(2) Multi review data: an evaluation dataset with multi-review samples. All the samples have a positive first label. Around 95% samples have a positive final label.

The results are summarized in Table 3. Besides AUC, we also report the recall at certain precision thresholds for the single-review data. Since false positive severely affects user experience, we take a high precision threshold. For the multi-review data, the positive rate is high, so the precision is automatically high.

One can see that in Table 3, after utilizing multi-review labels, test set performance improves in all metrics across all evaluation data sets. In addition, after further applying the pruning method, there is an additional performance improvement. We further provide Figure 4 for a graphical view of the ROC.

Data		First label	Multi	Multi (prune)
Single	AUC	0.8600	0.8617	**0.8632**
	R@P=0.9	0.5273	0.5469	**0.5561**
	R@P=0.8	0.9168	0.9151	**0.9192**
Multi	AUC	0.6464	0.7242	**0.7401**

Table 3. Evaluation performance in two different data sets. Both multi-review data and the proposed pruning method improves the performance. R@P=0.8: recall when precision equal to 0.8.

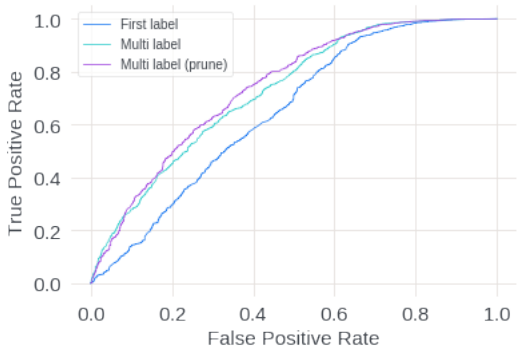

Figure 4. Evaluation performance of the proposed method and multiple reviews in scammer detection data.

7. Conclusion

The paper proposes two improvements for the ranking pruning algorithm (Northcutt et al., 2017). In the presence of two kinds of suspicious samples (noisy and erroneous samples), the first improvement helps to prune only samples with label noise, and tries to avoid pruning samples with label error. The second improvement prunes high confidence samples. In addition, the paper presents a theoretical analysis to show how the multiple review process can help reduce label noise. To summarize, the performance of the rank pruning algorithm improves by the combination of (a) pruning high-confidence (with less learning value) and noisy samples, and (b) preserving the samples with (potentially) label error that have high learning value and producing accurate labels for them using multiple reviews.

There are several possible future directions. First, the proposed algorithm performs well for the samples around the decision boundary when the label noise is large, but not when the label noise is small. One may design additional algorithms to determine whether using a certain model to train a certain data set can be benefit from our proposed algorithm. Second, one can consider whether the proposed algorithm can be extended to multi-class classification tasks. Finally, while we are using hyper-parameter tuning to select (α, β), one may simplify the algorithm via dynamic hyper-parameter tuning algorithms to reduce the computation cost.

References

G. Algan and I. Ulusoy. Meta soft label generation for noisy labels. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7142–7148. IEEE, 2021.

O. Alonso. Challenges with label quality for supervised learning. *Journal of Data and Information Quality (JDIQ)*, 6(1):1–3, 2015.

E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. How important is importance sampling for deep budgeted training? *arXiv preprint arXiv:2110.14283*, 2021.

S. Ariafar, Z. Mariet, D. H. Brooks, J. G. Dy, and J. Snoek. Faster & more reliable tuning of neural networks: Bayesian optimization with importance sampling. In *AISTATS*, pages 3961–3969, 2021.

H.-S. Chang, E. Learned-Miller, and A. McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. *Advances in Neural Information Processing Systems*, 30, 2017.

Z. Cui, Y. Zhang, and Q. Ji. Label error correction and generation through label relationships. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3693–3700, 2020.

A. Daw, J. Bu, S. Wang, P. Perdikaris, and A. Karpatne. Rethinking the importance of sampling in physics-informed neural networks. *arXiv preprint arXiv:2207.02338*, 2022.

Y. Ding, L. Wang, D. Fan, and B. Gong. A semi-supervised two-stage approach to learning from noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1215–1224. IEEE, 2018.

D. Dua and C. Graff. Uci machine learning repository. 2019. URL http://archive.ics.uci.edu/ml.

P. Glasserman, P. Heidelberger, and P. Shahabuddin. Variance reduction techniques for estimating value-at-risk. *Management Science*, 46(10):1349–1364, 2000.

D. Hwang, K. C. Sim, Z. Huo, and T. Strohman. Pseudo label is better than human label. *arXiv preprint arXiv:2203.12668*, 2022.

D. Kahneman, O. Sibony, and C. Sunstein. *Noise*. HarperCollins UK, 2022.

A. Katharopoulos and F. Fleuret. Biased importance sampling for deep neural network training. *arXiv preprint arXiv:1706.00043*, 2017.

A. Katharopoulos and F. Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

G. Kazai, J. Kamps, and N. Milic-Frayling. An analysis of human factors and label accuracy in crowdsourcing relevance judgments. *Information retrieval*, 16(2):138–178, 2013.

B. Li and Q. Gao. Improving data quality with label noise correction. *Intelligent Data Analysis*, 23(4):737–757, 2019.

L. Li, K. G. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. In *ICLR (Poster)*, 2017.

R. Müller, S. Kornblith, and G. E. Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

M. A. Nabian, R. J. Gladstone, and H. Meidani. Efficient training of physics-informed neural networks via importance sampling. *Computer-Aided Civil and Infrastructure Engineering*, 36(8):962–977, 2021.

N. Nigam, T. Dutta, and H. P. Gupta. Impact of noisy labels in learning techniques: a survey. In *Advances in data and information sciences*, pages 403–411. Springer, 2020.

C. Northcutt, L. Jiang, and I. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.

C. G. Northcutt, T. Wu, and I. L. Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. *arXiv preprint arXiv:1705.01936*, 2017.

G. Pleiss, T. Zhang, E. Elenberg, and K. Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.

V. Sameera, A. Bindra, and G. P. Rath. Human errors and their prevention in healthcare. *Journal of Anaesthesiology, Clinical Pharmacology*, 37(3):328, 2021.

X. Shang, E. Kaufmann, and M. Valko. A simple dynamic bandit algorithm for hyper-parameter tuning. 2019.

H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

J. Surowiecki. *The wisdom of crowds*. Anchor, 2005.

Z. Zhang and M. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

H. Zhou, L. Song, J. Chen, Y. Zhou, G. Wang, J. Yuan, and Q. Zhang. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. *arXiv preprint arXiv:2102.00650*, 2021.

Z.-H. Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.

A. Zlateski, R. Jaroensri, P. Sharma, and F. Durand. On the importance of label quality for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1479–1487, 2018.

A. Proofs

A.1. Theorem 2 and 3

For Theorem 2, we calculate $\widetilde{\rho}_0$, $\widetilde{\rho}_1$, $\widetilde{UB}$ and $\widetilde{LB}$ for the multi-review process. In both examples, we assume $\rho_1 = \rho_0 = \rho$ and $p_{z1} = 1/2$, thus $\widetilde{\rho}_0 = \widetilde{\rho}_1 = \widetilde{\rho}$ for some $\widetilde{\rho}$.

First, for $\widetilde{\rho}$,

$$\begin{aligned}
\widetilde{\rho} &= P(\text{two incorrect labels}) \\
&= P(\text{the first two labels are incorrect}) + P(\text{there are three labels and two are incorrect}) \\
&= \rho^2 + 2\rho^2(1-\rho) \\
&= \rho^2(3-2\rho).
\end{aligned}$$

Then we can calculate $\widetilde{UB}$ using $\widetilde{\rho}$ as follows

$$\widetilde{UB}_{y=0} = 0.5\widetilde{\rho} + 0.5(1-\widetilde{\rho}),$$

which means that

$$\widetilde{UB}_{y=0} - (1-\widetilde{\rho}) = 0.5\widetilde{\rho} - 0.5(1-\widetilde{\rho}) = 0.5(2\widetilde{\rho} - 1).$$

As a result, if we compare the above quantity with the one for single-review data, we obtain

$$\begin{aligned}
& \widetilde{UB}_{y=0} - (1-\widetilde{\rho}) - UB_{y=0} - (1-\rho) \\
= \; & \widetilde{UB}_{y=0} - UB_{y=0} + \widetilde{\rho} - \rho \\
= \; & 2(\widetilde{\rho} - \rho) < 0.
\end{aligned}$$

This proof also extends naturally to Theorem 3.

A.2. Derivation of Equation (2)

Observe that

$$\begin{aligned}
\rho_0(1 - p_{z1} - \rho_1) &= \rho_0[(1-\rho_1) - p_{z1}] \\
&= \rho_0[P(Z=1 \mid Y=1) - P(Z=1)] \\
&= \rho_0[P(Z=1 \mid Y=1) - P(Z=1 \mid Y=1)P(Y=1) - P(Z=1 \mid Y=0)P(Y=0)] \\
&= \rho_0 P(Y=0)[P(Z=1 \mid Y=1) - P(Z=1 \mid Y=0)] \\
&= P(Z=1, Y=0)(1 - \rho_1 - \rho_0) \\
&= \pi_1 p_{z1}(1 - \rho_1 - \rho_0).
\end{aligned}$$

As a result,

$$\pi_1 = \frac{\rho_0(1 - p_{z1} - \rho_1)}{p_{z1}(1 - \rho_1 - \rho_0)},$$

and one can construct the estimate of $\widehat{\pi}_0$ using $\widehat{\rho}_{0,t}$ and $\widehat{\rho}_{1,t}$. Similarly one can obtain the estimate of π_0 in (2).

Note that the above derivation does require knowledge of g.