Robust Loss Functions for Training Decision Trees with Noisy Labels

Jonathan Wilton, Nan Ye

School of Mathematics and Physics, The University of Queensland
jonathan.wilton@uq.edu.au, nan.ye@uq.edu.au

Abstract

We consider training decision trees using noisily labeled data, focusing on loss functions that can lead to robust learning algorithms. Our contributions are threefold. First, we offer novel theoretical insights on the robustness of many existing loss functions in the context of decision tree learning. We show that some of the losses belong to a class of what we call *conservative losses*, and the conservative losses lead to an early stopping behavior during training and noise-tolerant predictions during testing. Second, we introduce a framework for constructing robust loss functions, called *distribution losses*. These losses apply percentile-based penalties based on an assumed margin distribution, and they naturally allow adapting to different noise rates via a robustness parameter. In particular, we introduce a new loss called the *negative exponential loss*, which leads to an efficient greedy impurity-reduction learning algorithm. Lastly, our experiments on multiple datasets and noise settings validate our theoretical insight and the effectiveness of our adaptive negative exponential loss.

Introduction

Noisily labeled data often arise in machine learning, due to reasons such as the difficulty of accurately labeling data and the use of crowd-sourcing for labeling (Song et al. 2022). Various approaches have been developed to handle the label noise, including eliminating mislabeled examples (Brodley, Friedl et al. 1996), implicit/explicit regularization (Tanno et al. 2019; Lukasik et al. 2020), the use of robust loss functions (Manwani and Sastry 2013; Yang, Gao, and Li 2019), with recent works mostly focusing on neural networks.

This paper focuses on robust loss functions for learning decision trees from noisily labeled data. Tree-based methods (e.g., random forests) are among the most effective machine learning methods, particularly on tabular data (Grinsztajn, Oyallon, and Varoquaux 2022; Kaggle 2021), and several robust loss functions have been shown to be effective for neural network learning in the presence of label noise (e.g., see (Ghosh, Kumar, and Sastry 2017; Zhang and Sabuncu 2018)). However, little attention has been paid to the understanding and design of robust loss functions in the context of decision tree learning. This is likely because decision tree learning algorithms are often described as greedy impurity-reduction algorithms in the literature, and it is less well-known that the impurity-reduction algorithms are greedy algorithms for minimizing certain losses (Yang, Gao, and Li 2019; Wilton et al. 2022). Our work aims to address this research gap.

Our main contributions are three-fold.

- We offer novel theoretical insight on the robustness of many existing loss functions. We show that some of them belong to a class of what we call *conservative* losses, which are robust due to an early stopping behavior during training and noise-tolerant predictions during testing.
- We introduce a framework for constructing robust loss functions, called *distribution losses*. These losses apply percentile-based penalties based on an assumed margin distribution. By using different assumed margin distributions, we can recover some commonly used loss functions, which shed interesting insight on these existing functions. An attractive property of the distribution loss is that they naturally allow adapting to different noise rates via a robustness parameter. Importantly, we introduce a new loss called the *negative exponential loss*, which leads to an efficient impurity-reduction learning algorithm.
- Our extensive experiments validate our theoretical insight and the effectiveness of our adaptive negative exponential loss.

The remainder of this paper is organized as follows. We first provide a more detailed discussion on related work, followed by some preliminary concepts. We then present the conservative losses and their robustness properties. After that, we describe our distribution-based robust loss framework and the negative exponential loss. Finally, we present details on experimental settings and results, with a brief conclusion. Our source code is available at https://github.com/jonathanwilton/RobustDecisionTrees.

Related Work

Our work is broadly related to the large body of approaches developed for dealing with label noise in the literature, which include filtering the noisy labels, learning a classifier and model for the label noise simultaneously, implicit/explicit regularization to avoid overfitting the noise, and designing robust loss functions (see, e.g., for the excellent reviews (Frénay and Verleysen 2013) or (Song et al. 2022) for detailed discussions).

The most relevant general approach to our work is the robust loss approach. There are two common approaches to design robust losses. One creates corrected losses by incorporating label noise rates if they are known (e.g., see (Natarajan et al. 2013; Patrini et al. 2017)). However, such information is typically unknown, and difficult to estimate accurately in practice. Another approach considers inherently robust losses, which does not require knowledge of the noise rates. Some pioneering theoretical works consider robustness of losses against label noise (Manwani and Sastry 2013; Ghosh, Manwani, and Sastry 2015; Ghosh, Kumar, and Sastry 2017). These works show that the zero-one (01) loss and mean absolute error (MAE) are robust to many types of label noise, while the commonly used cross entropy (CE) loss does not enjoy these same robustness properties. This sparked interest in developing new loss functions that share favorable qualities from each of the 01, MAE and CE, for example the generalized cross entropy (GCE) loss (Zhang and Sabuncu 2018), negative learning (Kim et al. 2019), symmetric cross entropy loss (Wang et al. 2019), curriculum loss (Lyu and Tsang 2020) and normalized loss functions (Ma et al. 2020). However, losses like the curriculum loss and the normalized losses are not suitable for decision tree learning, because the impurities for these losses lack analytical forms and efficient algorithms.

Our work is also closely related to tree methods for learning from noisily labeled data. Motivated by the predictive performance and interpretability of tree methods (Breiman et al. 1984; Breiman 2001; Geurts, Ernst, and Wehenkel 2006), various works have developed algorithms for decision tree learning in the presence of label noise, such as pruning (Breiman et al. 1984), making use of pseudo-examples during tree construction (Mantas and Abellan 2014), leaving a large number of samples at each leaf node (Ghosh, Manwani, and Sastry 2017), and adjusting the labels at each leaf node of a trained RF (Zhou, Ding, and Li 2019). To the best of our knowledge, the closest work to ours derives a robust impurity from the well-known ranking loss (Yang, Gao, and Li 2019). They only consider binary classification and their approach does not adapt to the noise rate, while we also consider multiclass classification and our approach adapts to the underlying noise rate. In addition, we offer novel theoretical insight on the robustness of various existing losses, and we contribute a general framework for constructing robust losses.

Preliminaries

Learning With Noisy Labels We consider K-class classification problem, where the input $\boldsymbol{x} \in \mathbb{R}^d$ and the one-hot label $\boldsymbol{y} \in \{0,1\}^K$ follows a joint distribution $p(\boldsymbol{x}, \boldsymbol{y})$. In the standard noise-free setting, we are given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ consisting of examples independently sampled from $p(\boldsymbol{x}, \boldsymbol{y})$. In the noisy setting, we have a dataset $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \widetilde{\boldsymbol{y}}_i)\}_{i=1}^n$ where each noisy label $\widetilde{\boldsymbol{y}}$ is obtained by randomly flipping the true label $\boldsymbol{y}$ with probability $\eta_{jk}^{\boldsymbol{x}} := \mathbb{P}(\widetilde{\boldsymbol{y}} = \boldsymbol{e}_k \,|\, \boldsymbol{y} = \boldsymbol{e}_j, \boldsymbol{x})$, with $\boldsymbol{e}_j$ being the one-hot vector for class j. We focus on class-conditional noise, where the noise probability is independent of the input, that is, each $\eta_{jk}^{\boldsymbol{x}}$ is equal to some constant η_{jk} for all $\boldsymbol{x}$. In particular, we consider the special case of uniform noise, in which each class has the same corruption rates, that is, $\eta_{jk} = 1 - \eta$ for $j = k$ and $\eta_{jk} = \eta/(K-1)$ for $j \neq k$, for some constant η.

The objective is to learn a classifier $\boldsymbol{g} : \mathbb{R}^d \to \Delta$ to minimize the expected risk

$$R(\boldsymbol{g}) := \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim p(\boldsymbol{x},\boldsymbol{y})}\, \ell(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{y}),$$

where $\ell : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$ is a loss function, and $\Delta = \{\boldsymbol{y} \in [0,1]^K : \boldsymbol{y}^\mathsf{T}\mathbf{1} = 1\}$ the standard $(K-1)$-simplex. Predicted labels can be obtained from the classifier with $\boldsymbol{e}_{\operatorname{argmax}(\boldsymbol{g}(\boldsymbol{x}))}$. With a set of noise-free training data, the risk can be estimated without bias via the empirical risk

$$\widehat{R}(\boldsymbol{g}; \mathcal{D}) := \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \ell(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{y})/|\mathcal{D}|.$$

When only a noisily labeled dataset $\widetilde{\mathcal{D}}$ is available, we instead estimate the expected risk with $\widehat{R}(\boldsymbol{g}; \widetilde{\mathcal{D}})$.

We focus on loss functions that are robust against label noise. A loss function ℓ is said to be noise tolerant if the minimizers of the expected risk using loss ℓ on the noisy and noise-free data distributions lead to the same expected risk using the 01 loss on noise-free data (Manwani and Sastry 2013). If a loss function ℓ is symmetric, i.e., $\sum_{j=1}^K \ell(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{e}_j) = C$ for any $\boldsymbol{x} \in \mathbb{R}^d$ and any $\boldsymbol{g}$, then, under uniform label noise with $\eta < \frac{K-1}{K}$, ℓ is noise tolerant (Ghosh, Manwani, and Sastry 2015; Ghosh, Kumar, and Sastry 2017). If we additionally have $R(\boldsymbol{g}^*) = 0$ for some classifier $\boldsymbol{g}^*$, $0 \leq \ell(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{e}_j) \leq C/(K-1)\, \forall j = 1, \ldots, K$ and the matrix $(\eta_{ij})_{i,j=1}^K$ is diagonally dominant, then ℓ is noise tolerant under class conditional noise (Ghosh, Kumar, and Sastry 2017). Examples of symmetric loss functions include the 01 loss $\ell(\widehat{\boldsymbol{y}}, \boldsymbol{e}_j) = \mathbb{1}(\widehat{\boldsymbol{y}} \neq \boldsymbol{e}_j)$ and MAE loss $\ell(\widehat{\boldsymbol{y}}, \boldsymbol{e}_j) = \|\widehat{\boldsymbol{y}} - \boldsymbol{e}_j\|_1$. On the other hand, the mean squared error (MSE) loss $\ell(\widehat{\boldsymbol{y}}, \boldsymbol{e}_j) = \|\widehat{\boldsymbol{y}} - \boldsymbol{e}_j\|_2^2$ and CE loss $\ell(\widehat{\boldsymbol{y}}, \boldsymbol{e}_j) = -\boldsymbol{e}_j^\mathsf{T} \log \widehat{\boldsymbol{y}}$ are not symmetric. In practice it has been shown that training neural network (NN) classifiers with these noise tolerant loss functions can lead to significantly longer training time before convergence (Zhang and Sabuncu 2018). The GCE loss $\ell(\widehat{\boldsymbol{y}}, \boldsymbol{e}_j) = (1 - (\boldsymbol{e}_j^\mathsf{T}\widehat{\boldsymbol{y}})^q)/q$ was proposed as a compromise between noise-robustness and good performance (Zhang and Sabuncu 2018). The hyperparameter $q \in [0,1]$ controls the robustness, with special cases $q = 0$ giving the CE and $q = 1$ the MAE.

Decision Tree Learning We briefly review two dual perspectives for decision tree learning: learning by impurity reduction, and learning by recursive greedy risk minimization.

In the impurity reduction perspective, the decision tree construction process recursively partitions the (possibly noisy) training set $\mathcal{D}$ such that each subset has similar labels. We start with a single node associated with the entire training set. Each time we have a node associated with a subset $\mathcal{S} \subseteq \mathcal{D}$ that we need to split, we find an optimal split (f, t) that partitions $\mathcal{S}$ into $\mathcal{S}_{f\leq t}$ and $\mathcal{S}_{f>t}$ based on whether the feature f is larger than the value t. The quality of a split is usually measured by its reduction in some label impurity measure I,

defined by:

$$\frac{|\mathcal{S}|}{|\mathcal{D}|}I(\mathcal{S}) - \frac{|\mathcal{S}_{f\leq t}|}{|\mathcal{D}|}I(\mathcal{S}_{f\leq t}) - \frac{|\mathcal{S}_{f>t}|}{|\mathcal{D}|}I(\mathcal{S}_{f>t}).$$

Based on the split, two child nodes associated with $\mathcal{S}_{f\leq t}$ and $\mathcal{S}_{f>t}$ are created. This process is repeated recursively on the two child nodes until some stopping criteria is satisfied.

In the recursive greedy risk minimization perspective, a node is split in a greedy way to minimize the empirical risk $\widehat{R}(\boldsymbol{g};\mathcal{D})$. If $\boldsymbol{g}$ predicts a constant $\widehat{\boldsymbol{y}}$ on a subset $\mathcal{S} \subseteq \mathcal{D}$ of the training examples, then the contribution to the empirical risk is the partial empirical risk $\widehat{R}(\widehat{\boldsymbol{y}};\mathcal{S}) := \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{S}} \ell(\widehat{\boldsymbol{y}},\boldsymbol{y})/|\mathcal{D}|$. Denote by $\widehat{\boldsymbol{y}}^* := \operatorname{argmin}_{\widehat{\boldsymbol{y}}\in\Delta}\widehat{R}(\widehat{\boldsymbol{y}};\mathcal{S})$ the optimal constant probability vector prediction, with minimum partial empirical risk

$$\widehat{R}^*(\mathcal{S}) := \widehat{R}(\widehat{\boldsymbol{y}}^*;\mathcal{S}).$$

The minimum partial empirical risk can be interpreted as an impurity measure. In fact, it is the Gini impurity and the entropy impurity (up to a multiplicative constant) when the loss is the MSE loss and the CE loss, respectively.

If we switch from a constant prediction rule to a decision stump that splits on feature f at threshold t, then the minimum partial empirical risk for the decision stump is $\widehat{R}^*(\mathcal{S}_{f\leq t}) + \widehat{R}^*(\mathcal{S}_{f>t})$, and the risk reduction for the split (f,t) is

$$\mathrm{RR}(f,t;\mathcal{S}) := \widehat{R}^*(\mathcal{S}) - \widehat{R}^*(\mathcal{S}_{f\leq t}) - \widehat{R}^*(\mathcal{S}_{f>t}). \quad (1)$$

We will use a subscript to specify the loss if needed. For example, $\widehat{R}^*_{\mathrm{MSE}}(\mathcal{S})$ and $\mathrm{RR}_{\mathrm{MSE}}(f,t;\mathcal{S})$ indicates that the MSE loss is used for computing the minimum partial empirical risk and the risk reduction for a split (f,t) on $\mathcal{S}$, respectively.

The equivalence between loss functions and impurity measures has been explored in, for example, (Yang, Gao, and Li 2019; Wilton et al. 2022). To illustrate, let $\boldsymbol{p}$ be the empirical class distribution for $\mathcal{S}$. Then the minimum partial risks for MSE and CE are the commonly used Gini impurity $1 - \|\boldsymbol{p}\|_2^2$ and entropy impurity $-\boldsymbol{p}^\mathsf{T}\log\boldsymbol{p}$, respectively, up to a multiplicative constant (see (a) and (b) of Theorem 1). Note that not all loss functions have impurities which have analytical forms and efficient algorithms, while our negative exponential loss yields an impurity which has an efficiently computable analytical formula, which is important for efficient decision tree learning.

For prediction, each test example is assigned to a leaf node based on its feature values, then labeled according to the majority label of the examples at the leaf node. Generalization performance of the decision tree classifier can be measured by comparing predictions on unseen data with true labels, however, it can be heavily affected when training data, particularly labels, are unreliable.

Conservative Losses

We first examine the impurities corresponding to various loss functions, including both standard loss functions and robust ones, in Theorem 1. Parts (a), (b) and (c) are shown in previous works (Breiman et al. 1984; Painsky and Wornell 2018; Yang, Gao, and Li 2019; Wilton et al. 2022) but included for completeness. All proofs are given in the appendices. [1] Unless otherwise stated, we shall use $\mathcal{D}$ to denote an arbitrary (possibly noisy) set of input-output pairs, and $\mathcal{S}$ a subset of $\mathcal{D}$, in this section.

Theorem 1. *Let $W_\mathcal{S} = |\mathcal{S}|/|\mathcal{D}|$ and $\boldsymbol{p} = (p_1,\ldots,p_K)^\mathsf{T} \in \Delta$ be the empirical class probability vector for $\mathcal{S}$, that is, $p_j = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{S}} \mathbb{1}(\boldsymbol{y} = \boldsymbol{e}_j)/|\mathcal{S}|, \forall j = 1,\ldots,K$. Then,*

(a) $\widehat{R}^*_{\mathrm{MSE}}(\mathcal{S}) = W_\mathcal{S}(1 - \|\boldsymbol{p}\|_2^2)$,
(b) $\widehat{R}^*_{\mathrm{CE}}(\mathcal{S}) = W_\mathcal{S}(-\boldsymbol{p}^\mathsf{T}\log\boldsymbol{p})$,
(c) $\widehat{R}^*_{01}(\mathcal{S}) = W_\mathcal{S}(1 - \|\boldsymbol{p}\|_\infty)$,
(d) $\widehat{R}^*_{\mathrm{GCE}}(\mathcal{S}) = \begin{cases} W_\mathcal{S}(-\boldsymbol{p}^\mathsf{T}\log\boldsymbol{p}), & q = 0, \\ W_\mathcal{S}(1 - \|\boldsymbol{p}\|_{1/(1-q)})/q, & \forall q \in (0,1), \\ W_\mathcal{S}(1 - \|\boldsymbol{p}\|_\infty)/q, & q \geq 1, \end{cases}$
(e) $\widehat{R}^*_{\mathrm{MAE}}(\mathcal{S}) = 2W_\mathcal{S}(1 - \|\boldsymbol{p}\|_\infty)$.

The result highlights an interesting observation on the impurities of two robust losses, the MAE loss and the 01 loss (Ghosh, Kumar, and Sastry 2017): both lead to the misclassification impurity $1 - \|\boldsymbol{p}\|_\infty$, up to a multiplicative constant. Furthermore, the GCE is equivalent to an impurity measure that interpolates between the misclassification and entropy impurities depending on the chosen value of q. This is consistent with the fact that GCE interpolates between CE and MAE loss for $q \in (0,1)$ (Zhang and Sabuncu 2018).

Below, we introduce a broad class of losses that lead to the misclassification impurity, and provide a few results to justify their robustness properties in the context of decision tree learning with label noise.

Definition 1. *A loss function ℓ is called* C-conservative *if, for some constant $C > 0$, it satisfies the following properties:*

(a) $\sum_{j=1}^K \ell(\widehat{\boldsymbol{y}}, \boldsymbol{e}_j) \geq C(K-1),\ \forall \widehat{\boldsymbol{y}} \in \Delta$,
(b) $\ell(\widehat{\boldsymbol{y}}, \boldsymbol{e}_j) \leq C,\ \forall \widehat{\boldsymbol{y}} \in \Delta,\ \forall j = 1,\ldots,K$, *and*
(c) $\ell(\boldsymbol{e}_j, \boldsymbol{e}_j) = 0,\ \forall j = 1,\ldots,K$.

Intuitively, (a) and (b) implies that the loss assigned to a single class is never too much as compared to the total loss assigned to all classes.

Theorem 2. *In a K class classification problem let $\ell : \mathbb{R}^K \times \mathbb{R}^K \to [0,\infty)$ be a loss function, and $\boldsymbol{p} \in \Delta$ be the vector of proportions of examples in $\mathcal{S}$ from each class. Then, we have*

$$\widehat{R}^*(\mathcal{S}) = CW_\mathcal{S}(1 - \|\boldsymbol{p}\|_\infty) \quad (2)$$

if ℓ is C-conservative. In addition, if Eq. (2) *holds, then ℓ satisfies (a) in Definition 1.*

This theorem provides a convenient way to check if a loss function leads to the misclassification impurity, as illustrated in Corollary 2.1.

Corollary 2.1. *The MAE, 01 loss, GCE ($q \geq 1$) and infinity norm loss satisfy $\widehat{R}^*(\mathcal{S}) = W_\mathcal{S}\, C(1 - \|\boldsymbol{p}\|_\infty)$, with $C = 2,\ 1,\ 1/q$ and 1, respectively.*

[1] The appendices are available in the full version of the paper at https://github.com/jonathanwilton/RobustDecisionTrees.

Our first robustness property of the conservative loss is concerned with the optimal predictions.

Theorem 3. *Assume ℓ is a conservative loss function. Then,*

$$\operatorname*{argmin}_{\widehat{\boldsymbol{y}}\in\Delta} \widehat{R}(\widehat{\boldsymbol{y}};\mathcal{S}) = \boldsymbol{e}_{\operatorname{argmax}(\boldsymbol{p})}.$$

Moreover, for non-conservative loss functions MSE and CE,

$$\operatorname*{argmin}_{\widehat{\boldsymbol{y}}\in\Delta} \widehat{R}(\widehat{\boldsymbol{y}};\mathcal{S}) = \boldsymbol{p}.$$

This result suggests that the optimal constant prediction at each node is less likely to change after label corruption when the loss functions are conservative versus not. This is because for a conservative loss, the optimal constant prediction is the one-hot vector for the most likely class, which is likely to remain the same after label noise is added.

Our second robustness result, Theorem 4, provides a more precise statement on the effect of noise on the majority class. Specifically, we show that a sample size of $O(1/\gamma^2)$ is needed to guarantee that the majority class remains the same under the label noise, where γ is a margin parameter that depends on the noise and the class distribution of the clean dataset, as defined in the theorem below.

Theorem 4. *Let $\mathcal{S}$ be a set of n noise-free examples, p_k be the empirical probability of class k in $\mathcal{S}$, and k^* the most prevalent class in $\mathcal{S}$, that is, $p_{k^*} > p_k$ for any $k \neq k^*$. In addition, let $\widetilde{\mathcal{S}}$ be obtained from $\mathcal{S}$ by applying a uniform noise with rate $\eta < (K-1)/K$, and $\widetilde{p}_k$ the empirical probability of class k in $\widetilde{\mathcal{S}}$. Then*

$$\mathbb{P}(\widetilde{p}_{k^*} \geq \widetilde{p}_k \text{ for all } k \neq k^* \,|\, \mathcal{S}) \geq 1 - (K-1)e^{-n\gamma^2/2},$$

with $\gamma = \min_{k\neq k^*} (\widetilde{p}^{\eta}_{k^*} - \widetilde{p}^{\eta}_{k})$, and $\widetilde{p}^{\eta}_j = \mathbb{E}[\widetilde{p}_j \,|\, \mathcal{S}]$, $\forall\, j = 1,\ldots,K$.

Our third robustness result, Theorem 5, shows that a conservative loss leads to an early stopping property that is robust against label noise, while a non-conservative loss generally does not have this property and stops under a much more stringent condition.

Theorem 5. *Assume that tree growth at a node is halted when the risk reduction at the node for any split is non-positive.*

(a) For a conservative loss, a node will stop splitting if and only if the majority classes at the node are also the majority classes at both child nodes for all splits.

(b) For the MSE, CE or GCE ($q \in (0,1)$) loss, splitting is only halted if the parent node and both child nodes all share the same label distribution for all splits.

We found in our experiments that this early stopping phenomenon does indeed happen in practice, and tends to help tree methods avoid overfitting in situations with large amounts of noise in the training labels. However, we also observed that this early stopping can sometimes lead to underfitting in low noise situations, as predicted in (Breiman et al. 1984) for noise-free data. Note that we give a necessary and sufficient condition for early stopping with conservative loss functions and compare to each of the MSE, CE and GCE ($q \in (0,1)$), while a sufficient condition for early stopping with the misclassification impurity and comparison with Gini impurity can be found in (Breiman et al. 1984).

Distribution Losses

We introduce a new approach for constructing robust loss functions that can adapt to the noise level. This allows us to address a limitation of the conservative losses as pointed out in the previous section: while conservative losses are robust against label noise, they can lead to underfitting in low noise situations.

Our approach is based on a simple idea that we will use to unify various common losses. Specifically, consider binary classification, and let $y \in \{-1, 1\}$, $\widehat{y} \in \mathbb{R}$, and $z = y\widehat{y}$ denote the true label, the prediction, and the margin, respectively, then for any CDF F, $\ell(y, \widehat{y}) = F(-z)$ can be used as a loss function. Intuitively, assume the margin of a random example follows the distribution F, then the loss is the probability that the random margin is larger than a value. We call ℓ a *distribution loss*. The loss is bounded in [0, 1], converging to 0 and 1 when $z \to +\infty$ and $z \to -\infty$, respectively.

Various commonly used loss functions are distribution losses.

Lemma 6. *The distribution loss is the 01 loss $\ell(z) = (1 - \operatorname{sign}(z))/2$ for the Bernoulli distribution $\mathsf{Ber}(0)$; the sigmoid loss $\ell(z) = 1/(1 + \exp(z))$ for the logistic distribution $\mathsf{Logistic}(0, 1)$; and the ramp loss $\ell(z) = \max\{0, \min\{1, (1 - z)/2\}\}$ for the uniform distribution $\mathsf{U}(-1, 1)$.*

We instantiate the distribution loss framework to create a robust loss function with a parameter that allows for adaptation to different noise levels.

Lemma 7. *For an exponential variable $X \sim \mathsf{Exp}(1)$, consider its shifted negative $Z = \mu - X$ for some $\mu \geq 0$, then the CDF of Z is*

$$F(z) = \min\{1, \exp(z - \mu)\},$$

and the corresponding loss function is

$$\ell(\widehat{y}, y) = \min\{1, \exp(-y\widehat{y} - \mu)\}.$$

We call this loss the negative exponential (NE) loss.

The plot of the NE loss in Figure 1 reveals several robustness properties: first, the loss is capped at 1 even for large negative margins, thus preventing imposing excessively large penalty on a noisy example far from the decision boundary; second, the rapid decrease of the loss to zero helps the classifier to avoid overfitting to large positive margins; third, the robustness parameter μ allows control on the range of negative margins that should be penalized, with a large μ allowing ignoring more noisy examples with negative margins.

NE loss can be viewed as a capped and shifted variant of the standard exponential loss. While capping creates zero gradient and thus may make gradient-based learning difficult, this is not a limitation for decision tree learning as we only need to compute the impurity. Theorem 8 gives an expression for the impurity corresponding to the NE loss.

Theorem 8. *The NE loss's partial empirical risk is given by*

$$\widehat{R}(\widehat{y};\mathcal{S}) = \sum_{(\boldsymbol{x},y)\in\mathcal{S}} \min\{1, \exp(-\widehat{y}y - \mu)\}/|\mathcal{D}|,$$

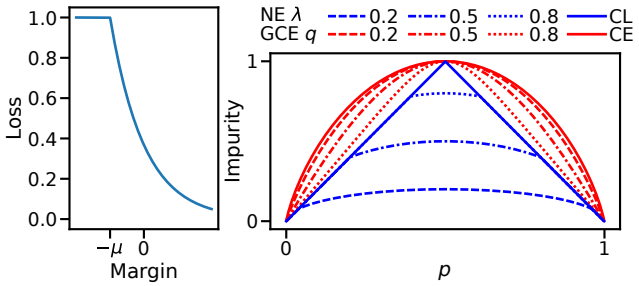

Figure 1: Left: NE loss as a function of the margin $y\widehat{y}$. Right: Blue lines are NE impurities with their λ values provided in the legend. The special case with $\lambda = 1$ is denoted by CL (conservative loss). Red lines are GCE impurities with their q values provided in the legend. The special case with $q = 0$ is denoted by CE (cross entropy). The impurities have been scaled for better comparison.

and the corresponding impurity (i.e., minimum partial empirical risk) is

$$\min_{\widehat{y}\in\mathbb{R}} \widehat{R}(\widehat{y};\mathcal{S}) = W_{\mathcal{S}} \min\left\{1 - \|\boldsymbol{p}\|_\infty,\ \lambda\sqrt{(1-\|\boldsymbol{p}\|_2^2)/2}\right\},$$

with $\boldsymbol{p} = (p_+, p_-)^\mathsf{T}$, p_+ *being the proportion of examples with* $y = +1$ *in* $\mathcal{S}$ *and* $\lambda = 2e^{-\mu}$.

Note that in the definition of the NE impurity above, $\widehat{y}$ ranges over $\mathbb{R}$ instead of over the 1-simplex. This is because we have chosen the prediction $\widehat{y}$ to be a positiveness score, instead of a two-dimensional vector representing a probability distribution, as done in the preliminaries section.

We provide a generalization of the NE impurity to the multiclass setting:

$$\min_{\widehat{y}\in\mathbb{R}} \widehat{R}(\widehat{y};\mathcal{S}) = W_{\mathcal{S}} \min\left\{1 - \|\boldsymbol{p}\|_\infty,\ \lambda\sqrt{\frac{1-\|\boldsymbol{p}\|_2^2}{K/(K-1)}}\right\}.$$

Clearly, this reduces to the binary NE impurity when $K = 2$. The $K/(K-1)$ factor is chosen such that when $\lambda = 1$, the two expressions under $\min$ have the same maximum value, a property that holds for the binary case in Theorem 8.

We visualize the NE impurity and compare it against the entropy, GCE, and misclassification impurities in Figure 1. The NE impurity interpolates between the misclassification impurity and the square root of the Gini impurity: for medium p values, the NE impurity equals the square root of the Gini impurity (up to a multiplicative constant); while for small and large p values, the NE impurity is the same as the misclassification impurity. In particular, we get the misclassification impurity for $\lambda = 1$, and we get the square root of the Gini impurity as $\lambda \to 0$. In contrast, the GCE impurities (including entropy) upper bound the misclassification impurity.

The NE impurity supports an adaptive robustness mechanism: the robustness parameter $\lambda \in (0, 1]$ controls the similarity between the NE impurity and the misclassification impurity, and it can be tuned to adapt to the noise rate, as done in our experiments. A larger λ is associated with higher similarity with the misclassification impurity, thus encouraging more early stopping and higher robustness. An alternative effective way to control early stopping is setting the minimum number of samples in a leaf (Ghosh, Manwani, and Sastry 2017). Our method is distribution dependent and more flexible in the sense that it allows for leaf nodes with varying numbers of samples. We also note that the GCE impurity (Zhang and Sabuncu 2018) has a hyperparameter $q \in [0, 1]$ controlling how it interpolates between the robust misclassification impurity and the non-robust entropy impurity. However, the early stopping property as in Theorem 5 only takes effect for $q \geq 1$. We see in experiments that varying q between 0 and 1 does not seem to improve robustness to label noise for tree methods as a result.

Name	Train	Test	Feature	Class
MNIST Digits	60 000	10 000	784	10
CIFAR-10	50 000	10 000	3 072	10
20News	11 313	7 531	300	20
UNSW-NB15	175 341	82 332	39	10
Covertype	464 809	116 203	54	7
Mushrooms	6499	1625	112	2

Table 1: Benchmark datasets.

Experiments

In this section we empirically compare the NE loss with several other loss functions and tree growing methods. Selected baselines include NE loss, conservative losses (CLs), twoing split criteria (Breiman et al. 1984), Credal-C4.5 (Mantas and Abellan 2014), GCE loss (Zhang and Sabuncu 2018), ranking loss (Yang, Gao, and Li 2019), CE loss and MSE loss. Note that twoing, Gini and misclassification are shown to be robust with large number of samples at leaf in (Ghosh, Manwani, and Sastry 2017). We are interested in the predictive performance on clean test data of both decision trees and random forests trained using noisily labeled training data in various label noise settings. We also investigate the effect of tuning the hyperparameter λ in the NE impurity.

Datasets We consider some commonly used datasets from UCI including Covertype (Blackard 1998), 20News (Lang 1995), Mushrooms (Audubon Society Field Guide 1987), as well as the MNIST digits (LeCun et al. 1998), CIFAR-10 (Krizhevsky 2009) and UNSW-NB15 (Moustafa and Slay 2015). Datasets were selected due to diversity in number of samples, number of features, number of classes, type (image, tabular, text), and domain (computer vision, cyber security, cartography). We look at both multiclass ($K > 2$) and binary ($K = 2$) classification problems for each dataset. Mushrooms datasets and ranking loss excluded from multiclass classification experiments due to Mushrooms being a binary classification dataset, and ranking loss only defined for binary classification problems. Table 1 is a summary of the benchmark datasets.

Binarized versions of labels are based on the processing in (Kiryo et al. 2017; Wilton et al. 2022). For 20News, GloVe pre-trained word embeddings (Pennington, Socher, and Man-

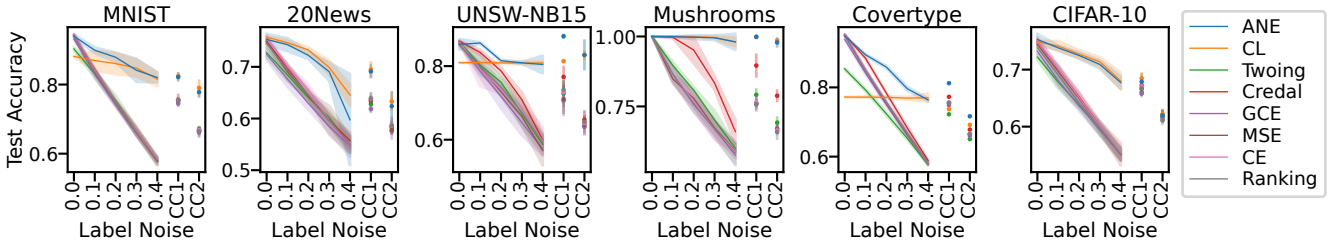

Figure 2: Mean test accuracy with 2x sd bands for DT on binary classification problems using different splitting criteria. Training labels corrupted using uniform noise $\eta \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$ and class conditional noise CC1 $(0.1, 0.3)$ and CC2 $(0.2, 0.4)$.

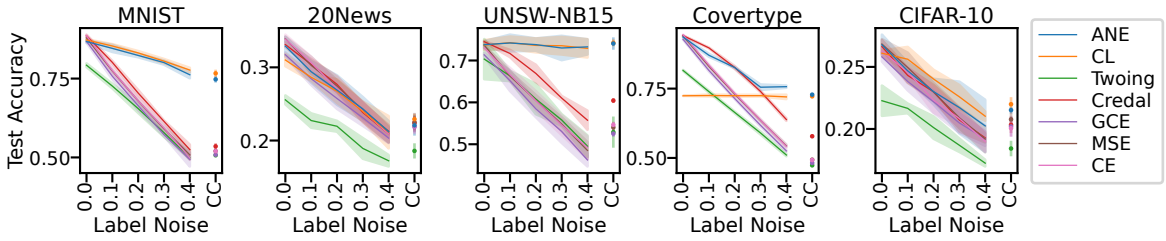

Figure 3: Mean test accuracy with 2x sd bands for DT on multiclass classification problems using different splitting criteria. Training labels corrupted using uniform noise $\eta \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$ and class conditional (CC) noise.

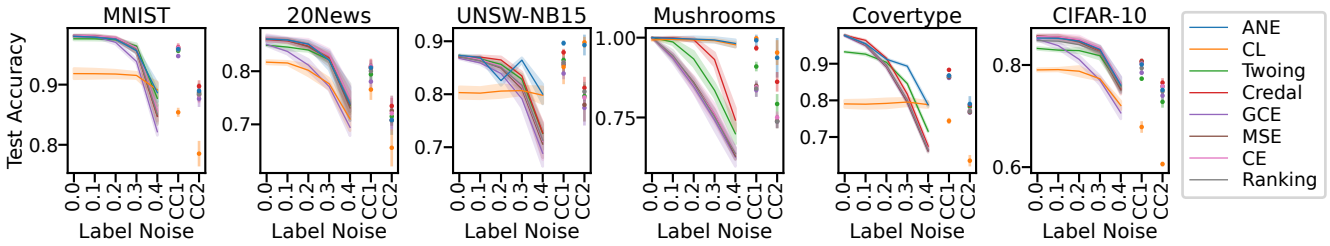

Figure 4: Mean test accuracy with 2x sd bands for RF on binary classification problems using different splitting criteria. Training labels corrupted using uniform noise $\eta \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$ and class conditional noise CC1 $(0.1, 0.3)$ and CC2 $(0.2, 0.4)$.

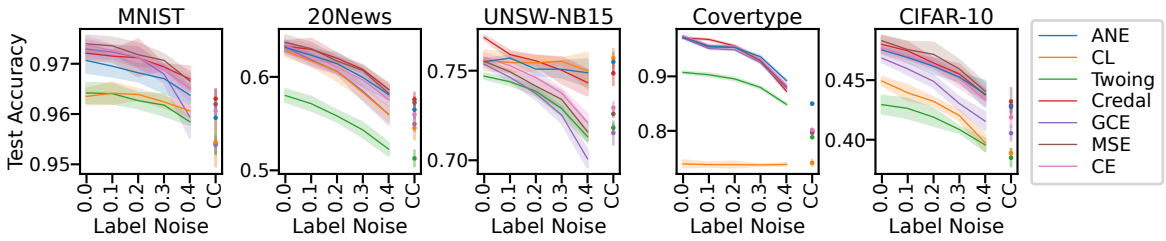

Figure 5: Mean test accuracy with 2x sd bands for RF on multiclass classification problems using different splitting criteria. Training labels corrupted using uniform noise $\eta \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$ and class conditional (CC) noise.

ning 2014) were used (glove.840B.300d), then average pooling was applied over the word embeddings to generate the embedding for each document. The binarized classes are 'alt., comp., misc., rec.' versus 'sci., soc., talk.'. For MNIST, the binarized classes are '0, 2, 4, 6, 8' versus '1, 3, 5, 7, 9' (even vs odd). For CIFAR-10, the binarized classes are 'airplane, automobile, ship, truck' versus 'bird, cat, deer, dog, frog, horse' (animal vs non-animal). For UNSW-NB15, we removed `ID` and the nominal features `proto`, `service` and `state`. The binarized labels are attack versus benign. For Covertype, the binarized classes are the second class versus others, as done in (Collobert, Bengio, and Bengio 2001), and the train-test split was performed using scikit-learn train_test_split with train_size 0.8 and random_state 0. For Mushrooms, we used the LIBSVM (Chang and Lin 2011) version, and the train-test split was performed identically as for Covertype.

Label Noise We used both uniform noise and class conditional noise to corrupt the training labels, while testing labels had no noise applied. For uniform label noise, noise rates $\eta = 0.0,\ 0.1,\ 0.2,\ 0.3$ and 0.4 were used. For class

conditional noise in binary classification two noise rates were used; $(0.1, 0.3)$ and $(0.2, 0.4)$. For multiclass classification, transition probabilities were constructed based on the similarity between classes using the Mahalanobis distance. See Algorithm 1 in Appendix J for details.

Hardware & Software Experiments were performed using Python on a computer cluster with Intel Xeon E5 Family CPUs @ 2.20GHz and 192GB memory running CentOS.

Hyperparameters Following common practice (Pedregosa et al. 2011), we set no restriction on maximum depth or number of leaf nodes, minimum one sample per leaf node, 100 trees in each RF, each using only $\sqrt{d}$ randomly chosen features, and predictions with RF made by majority average label distribution over all trees. The NE impurity hyperparameter $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$ was tuned by training on 80% of the noisy training set and validation on the other 20%. GCE $q = 0.7$ as recommended for NNs (Zhang and Sabuncu 2018), Credal-C4.5 $s = 1$ as recommended in (Mantas and Abellan 2014).

Classification Performance

We trained each model on 5 random noisy training sets to account for randomness in training labels and learning algorithms. We report the average test set accuracies and 2x standard deviation bands. Additional results can be found in Appendices K and L.

Effect of Loss Function on Robustness to Label Noise for Decision Trees Decision tree results for binary and multiclass settings are summarized in Figures 2 and 3, respectively. Adaptive NE loss (ANE; NE with tuned λ) is almost always a top performer across all label noise settings, label configurations (binary, multiclass) and datasets. Conservative loss functions sometimes give poor performance in low noise settings, but usually give strong performance in high noise settings. CE, MSE, Ranking, GCE give similar performance. Twoing split criteria typically gives the worst performance. Credal-C4.5, across many noise settings and datasets, usually has performance somewhere between CE, MSE, Ranking, GCE and ANE.

Effect of Loss Function on Robustness to Label Noise for Random Forest Random forest results for binary and multiclass settings are summarized in Figures 4 and 5, respectively. We first note that the differences in performance across loss functions are less significant than for decision trees in general, suggesting that an ensemble is more robust than a single tree. Note that RF still seems to boost performance of non-conservative loss functions, though this boost is less for conservative losses. Overall, ANE still shows strong performance.

Effect of NE Threshold on Predictive Performance and Tree Size We investigated the effect of the robustness parameter λ for the NE impurity by performing experiments on the MNIST digits dataset. The results in Figure 6 show that different λ values often have best performance for different noise settings, without any single λ value yielding the best performance in all cases. This together with the results in Figures 2 and 3 suggests that tuning λ allows effective control on early stopping, and in turn, robustness to label noise. Additional discussion on early stopping in binary classification is provided in Appendix M.

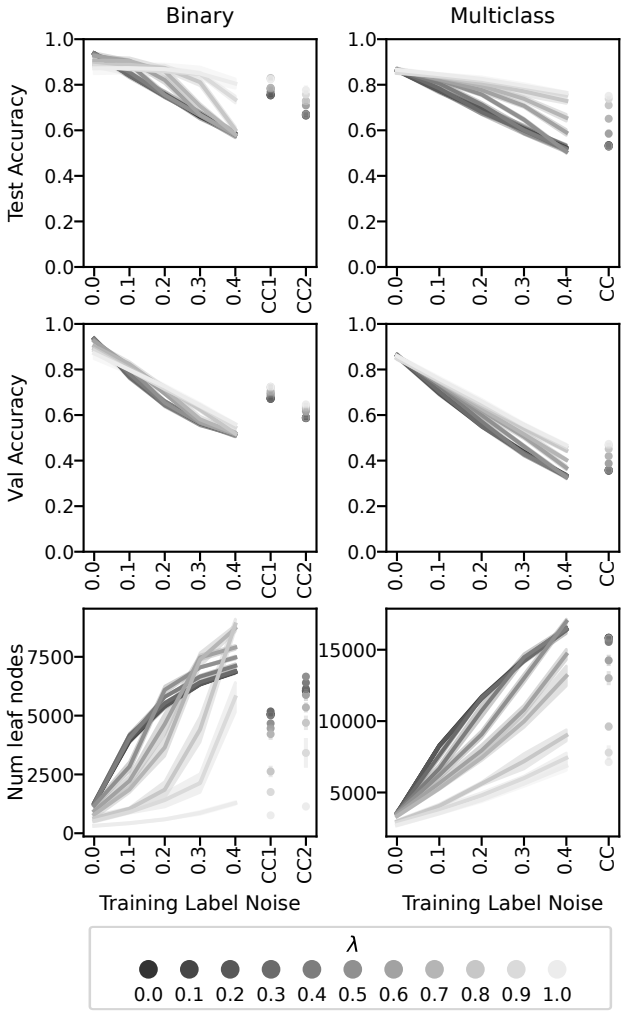

Figure 6: Performance of DT using NE loss with different values of λ on MNIST digits, binary and multiclass classification, 20% of noisily labeled training data used for validation. Results reported as mean $\pm$ 2x sd over 5 replications.

Conclusions

We developed new insight and tools on robust loss functions for decision tree learning. We introduced the conservative loss as a general class of robust loss functions and provided several results showing their robustness in decision tree learning. In addition, we introduced the distribution loss as a general framework for building loss functions based CDFs, and instantiated it to introduce the robust NE loss. Our experiments demonstrated that our NE loss effectively alleviates the adverse effect of noise in various noise settings.

Acknowledgments

We thank the anonymous reviewers for their many helpful comments, which we have incorporated to significantly improve the presentation of the paper. In particular, we thank one of them for a comment that inspired the discussion in Appendix M.

References

Audubon Society Field Guide. 1987. Mushroom. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5959T.

Blackard, J. 1998. Covertype. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50K5N.

Breiman, L. 2001. Random forests. *Machine learning*, 45: 5–32.

Breiman, L.; Friedman, J. H.; Olshen, R. A.; and Stone, C. J. 1984. *Classification and regression trees*. Wadsworth statistics/probability series. Belmont, Calif.: Wadsworth International Group. ISBN 0534980538.

Brodley, C. E.; Friedl, M. A.; et al. 1996. Identifying and eliminating mislabeled training instances. In *Proceedings of the National Conference on Artificial Intelligence*, 799–805.

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3): 1–27.

Collobert, R.; Bengio, S.; and Bengio, Y. 2001. A parallel mixture of SVMs for very large scale problems. *Advances in Neural Information Processing Systems*, 14.

Frénay, B.; and Verleysen, M. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5): 845–869.

Geurts, P.; Ernst, D.; and Wehenkel, L. 2006. Extremely randomized trees. *Machine learning*, 63: 3–42.

Ghosh, A.; Kumar, H.; and Sastry, P. S. 2017. Robust Loss Functions under Label Noise for Deep Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Ghosh, A.; Manwani, N.; and Sastry, P. 2015. Making risk minimization tolerant to label noise. *Neurocomputing*, 160: 93–107.

Ghosh, A.; Manwani, N.; and Sastry, P. 2017. On the robustness of decision tree learning under label noise. In *Advances in Knowledge Discovery and Data Mining: 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings, Part I 21*, 685–697. Springer.

Grinsztajn, L.; Oyallon, E.; and Varoquaux, G. 2022. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35: 507–520.

Kaggle. 2021. State of Machine Learning and Data Science 2021.

Kim, Y.; Yim, J.; Yun, J.; and Kim, J. 2019. Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, 101–110.

Kiryo, R.; Niu, G.; Du Plessis, M. C.; and Sugiyama, M. 2017. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems*, 30.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. MSc thesis, University of Toronto.

Lang, K. 1995. Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, 331–339. Elsevier.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lukasik, M.; Bhojanapalli, S.; Menon, A.; and Kumar, S. 2020. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, 6448–6458. PMLR.

Lyu, Y.; and Tsang, I. W. 2020. Curriculum Loss: Robust Learning and Generalization against Label Corruption. In *International Conference on Learning Representations*.

Ma, X.; Huang, H.; Wang, Y.; Romano, S.; Erfani, S.; and Bailey, J. 2020. Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning*, 6543–6553. PMLR.

Mantas, C. J.; and Abellan, J. 2014. Credal-C4. 5: Decision tree based on imprecise probabilities to classify noisy data. *Expert Systems with Applications*, 41(10): 4625–4637.

Manwani, N.; and Sastry, P. 2013. Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43(3): 1146–1151.

Moustafa, N.; and Slay, J. 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 military communications and information systems conference (MilCIS)*, 1–6. IEEE.

Natarajan, N.; Dhillon, I. S.; Ravikumar, P. K.; and Tewari, A. 2013. Learning with noisy labels. *Advances in neural information processing systems*, 26.

Painsky, A.; and Wornell, G. 2018. On the universality of the logistic loss function. In *2018 IEEE International Symposium on Information Theory (ISIT)*, 936–940. IEEE.

Patrini, G.; Rozza, A.; Krishna Menon, A.; Nock, R.; and Qu, L. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1944–1952.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; and Dubourg, V. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12: 2825–2830.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

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

Tanno, R.; Saeedi, A.; Sankaranarayanan, S.; Alexander, D. C.; and Silberman, N. 2019. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11244–11253.

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, 322–330.

Wilton, J.; Koay, A.; Ko, R.; Xu, M.; and Ye, N. 2022. Positive-Unlabeled Learning using Random Forests via Recursive Greedy Risk Minimization. *Advances in Neural Information Processing Systems*, 35: 24060–24071.

Yang, B.-B.; Gao, W.; and Li, M. 2019. On the robust splitting criterion of random forest. In *2019 IEEE International Conference on Data Mining (ICDM)*, 1420–1425. IEEE.

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

Zhou, X.; Ding, P. L. K.; and Li, B. 2019. Improving robustness of random forest under label noise. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 950–958. IEEE.