

Rowan University

Rowan Digital Works

Theses and Dissertations

6-2-2023

A GENERAL MODEL FOR NOISY LABELS IN MACHINE LEARNING

Glenn Dawson
Rowan University

Follow this and additional works at: <https://rdw.rowan.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Dawson, Glenn, "A GENERAL MODEL FOR NOISY LABELS IN MACHINE LEARNING" (2023). *Theses and Dissertations*. 3123.

<https://rdw.rowan.edu/etd/3123>

This Dissertation is brought to you for free and open access by Rowan Digital Works. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Rowan Digital Works. For more information, please contact graduateresearch@rowan.edu.

A GENERAL MODEL FOR NOISY LABELS IN MACHINE LEARNING

by

Glenn Dawson

A Dissertation

Submitted to the
Department of Electrical & Computer Engineering

College of Engineering

In partial fulfillment of the requirement

For the degree of
Doctor of Philosophy

at

Rowan University

April 28, 2023

Dissertation Chair: Robi Polikar, Ph.D., Department Head, Department of Electrical & Computer Engineering

Committee Members:

Ravi Ramachandran, Ph.D., Professor, Department of Electrical & Computer Engineering

Gregory Ditzler, Ph.D., Associate Professor, Department of Electrical & Computer Engineering

Shen-Shyang Ho, Ph.D., Associate Professor, Department of Computer Science

Ghulam Rasool, Ph.D., Assistant Member, Machine Learning Department, Moffitt Cancer Center

© 2023 Glenn Dawson

Dedication

I dedicate this dissertation to my parents. I can never thank you enough for the boundless and unwavering support, encouragement, and love that you have provided over the years, and I remain eternally grateful.

Acknowledgements

I would like to thank my doctoral advisor, Dr. Robi Polikar, for his generous support during the undertaking of writing this dissertation. He has provided valuable insights into the research process, from conceiving of ideas to navigating the perils of the publication process. His helpful feedback and broad expertise were invaluable blessings during the formulation, development, and composition of this dissertation.

I would also like to thank my dissertation committee, Dr. Ravi Ramachandran, Dr. Gregory Ditzler, Dr. Shen-Shyang Ho, and Dr. Ghulam Rasool, for their insightful comments and guidance throughout the research process.

To the many friends and colleagues with whom I have exchanged words, thoughts, ideas, and experiences, I wish to express my sincere thanks. From stimulating discussions to uplifting companionship, I could not have completed this dissertation without the unquantifiable assistance they have provided over the years.

Finally, I extend my deepest gratitude to my parents, for their inestimable support and boundless love not just during my doctoral research, but throughout my entire life. It is impossible to overstate the love and appreciation I have for them and all that they have done.

This work was partially supported by the U.S. Department of Education through the Graduate Assistance in Areas of National Need (GAANN) program, Award Number P200A180055.

Abstract

Glenn Dawson

A GENERAL MODEL FOR NOISY LABELS IN MACHINE LEARNING

2022-2023

Robi Polikar, Ph.D.

Doctor of Philosophy

Machine learning is an ever-growing and increasingly pervasive presence in everyday life; we entrust these models, and systems built on these models, with some of our most sensitive information and security applications. However, for all of the trust that we place in these models, it is essential to recognize the fact that such models are simply reflections of the data and labels on which they are trained. To wit, if the data and labels are suspect, then so too must be the models that we rely on—yet, as larger and more comprehensive datasets become standard in contemporary machine learning, it becomes increasingly more difficult to obtain reliable, trustworthy label information. While recent work has begun to investigate mitigating the effect of noisy labels, to date this critical field has been disjointed and disconnected, despite the common goal. In this work, we propose a new model of label noise, which we call “labeler-dependent noise (LDN).” LDN extends and generalizes the canonical instance-dependent noise model to multiple labelers, and unifies every preceding modeling strategy under a single umbrella. Furthermore, studying the LDN model leads us to propose a more general, modular framework for noise-robust learning called “labeler-aware learning (LAL).” Our comprehensive suite of experiments demonstrate that unlike previous methods that are unable to remain robust under the general LDN model, LAL retains its full learning capabilities under extreme, and even adversarial, conditions of label noise. We believe that LDN and LAL should mark a paradigm shift in how we learn from labeled data, so that we may both discover new insights about machine learning, and develop more robust, trustworthy models on which to build our daily lives.

Table of Contents

Abstract	v
List of Figures	x
List of Tables	xi
Chapter 1: Introduction	1
1.1 Motivation	2
1.2 Problem Statement	3
1.3 Contributions and Scope of Dissertation	4
1.4 Organization of the Dissertation	6
Chapter 2: Background	7
2.1 Notation	7
2.2 Supervised Learning	8
2.2.1 Semi-Supervised Learning	11
2.3 Noisy Labels	11
2.3.1 Class-Conditional Label Noise (CCN)	12
2.3.2 Instance-Dependent Label Noise	13
2.3.3 Noisy Labels from Crowdsourcing	15
2.4 Adversarial Machine Learning	16
2.4.1 Backdoor Attacks	17
Chapter 3: Related Work	19
3.1 Assuming the Existence of Clean Validation Data	19
3.2 Class-Conditional Label Noise	20
3.2.1 DivideMix	21

Table of Contents (Continued)

3.3	Instance-Dependent Label Noise	22
3.3.1	Progressive Label Correction	23
3.3.2	Self-Evolution Average Label	23
3.4	Learning from Crowds.....	24
3.4.1	Generative Model of Labels, Abilities, and Difficulties	26
3.4.2	Multidimensional Parameterization of Label Generation	27
3.4.3	Soft Observations of Labeler Opinions	28
Chapter 4:	Labeler-Dependent Label Noise	30
4.1	Illustration of Labeler-Dependent Noise	34
Chapter 5:	OpinionRank.....	37
Chapter 6:	Labeler-Aware Learning.....	43
Chapter 7:	Adversarial Labelers.....	47
7.1	Adversarial Labels from Bad-Faith Labelers	47
7.2	Multiple-Labeler Adversarial Attack Vectors.....	48
7.3	Adversarial Backdoor Attacks	49
7.3.1	Rethinking Backdoor Threat Modeling Under Labeler Awareness	49
7.3.2	A More Realistic, Practical Threat Model	50
7.3.3	Labeler-Aware Defense Against Adversarial Backdoor Attacks	54
Chapter 8:	Experiments.....	58
8.1	Metrics	58
8.2	Evaluating OpinionRank for Learning from Crowds.....	59

Table of Contents (Continued)

8.2.1	Generative Model of Labels, Abilities, and Difficulties	59
8.2.2	The Multidimensional Wisdom of Crowds	63
8.2.3	Combining Soft Decisions of Several Unreliable Labelers.....	66
8.2.4	Empirical Runtime Analysis	67
8.3	Testing Labeler-Aware Learning Under the Full LDN Model	69
8.3.1	Experimental Setup	69
8.3.2	Empirical Results.....	72
8.4	Measuring Adversarial Robustness Against Label Poisoning Attacks.....	77
8.4.1	Experimental Setup	78
8.4.2	Model Selection and Hyperparameter Tuning	80
8.4.3	Data Flooding Experiments.....	81
8.4.4	Multiple Adversaries Experiments	83
8.5	Adversarial Backdoor Defense	85
8.5.1	Baseline Classifier.....	87
8.5.2	MNIST Experiments	88
8.5.3	CIFAR-10 Experiments	88
8.5.4	GTSRB Experiments.....	89
8.5.5	Analysis of Results	89
8.6	Summary and Discussion of Overall Experimental Results	95
Chapter 9:	Conclusions	97
9.1	Contributions.....	97

Table of Contents (Continued)

9.2 Future Work	97
References	100

List of Figures

Figure	Page
Figure 1. Examples of Class-Conditional Transition Matrices	13
Figure 2. Examples of Applying Backdoor Attacks	18
Figure 3. Hierarchy of Label Noise Models	32
Figure 4. Examples of Class Label Selection Probabilities	34
Figure 5. Block Diagram of OpinionRank	41
Figure 6. Graphical Model of Labeler-Aware Learning	44
Figure 7. Training Pipeline With Backdoor Threats	53
Figure 8. Labeler-Aware Training Produces Robust Ensembles	56
Figure 9. Test Accuracy on Whitehill’s Labeling Model	60
Figure 10. Test Accuracy on Welinder’s Label Generation Model	64
Figure 11. Test Accuracy on Goldberger’s Three-Class Soft Opinions Model	67
Figure 12. Wall Clock Runtime Analysis of the OpinionRank Algorithm	68
Figure 13. SVHN Test Classification Accuracy in the Presence of J_s Spammers	76
Figure 14. CIFAR-10 Test Classification Accuracy in the Presence of J_s Spammers ..	77
Figure 15. Test Accuracy Under Data Flooding Attacks on MNIST	82
Figure 16. Test Accuracy Under Data Flooding Attacks on SVHN	83
Figure 17. Test Accuracy Under Multiple Adversaries Attacks on MNIST	84
Figure 18. Test Accuracy Under Multiple Adversaries Attacks on SVHN	85
Figure 19. Confusion Matrices for Accuracy Performance on GTSRB	94

List of Tables

Table		Page
Table 1.	Variables and Symbolic Notation Used in this Dissertation	7
Table 2.	Examples of Common Loss Functions	9
Table 3.	Examples of Labeler-Dependent Noisy Labels on MNIST	36
Table 4.	Mean Error Rate When Modeling Image Difficulty	61
Table 5.	Percent Accuracy on the Waterbirds Dataset	65
Table 6.	Experimental Hammer-Spammer Beta-Binomial Parameters	70
Table 7.	Converting Hammer-Spammer Ratios to Experimental Label Noise	71
Table 8.	MNIST Test Classification Accuracy in the Presence of J_s Spammers	74
Table 9.	Test Accuracy Against a Single Adversary	90
Table 10.	Test Accuracy Against Multiple Adversaries – Baseline Classifier	91
Table 11.	Test Accuracy Against Multiple Adversaries – Labeler-Aware Training ..	93

Chapter 1

Introduction

In 1641, mathematician and philosopher René Descartes grappled with the problem of *epistemic systematic doubt* [1]. In particular, he examined the concepts of *belief* and *certainty*, and how either come to be known. Descartes imagined an evil demon, of “utmost power and cunning,” who has employed all of its energy in order to deceive him. The demon, having absolute command over sensory information, would be able to shape Descartes’ experiences and perceptions such that he could no longer be certain of any beliefs based on empirical observation. Descartes supposed that while it may be perhaps unlikely that *all* of his beliefs may have been based on deception, the mere possibility that *any* beliefs may be so predicated was enough to cast *all* of them into doubt.

In general, humans learn concepts and beliefs through the guided instruction of perceived authorities [2]. For example, children learn such ideas as colors, numbers, and letters via their parents or teachers telling them the names of each color, number, or letter. To a naïve child, these authorities are unquestionable, as the child has no inherent ability to cross-reference these facts against other knowledge sources. Typically, it is assumed that a child’s parents or teachers will not actively conspire to mislead their students; however, such a possibility *might* exist, alongside less-malicious possibilities (such as the authorities collectively believing a falsehood to be true, and passing that belief along to the child). In these scenarios, the parents or teachers, acting as unquestioned authorities, would (perhaps unwittingly) fulfill the role of Descartes’ demon; if the authorities taught the child into believing that the color *red* were actually called “blue,” then the child, being naïve and guileless to the misinformation, would have no recourse but to accept that red were actually

blue. With no frame of reference to dispel the deception, the child will believe with all their heart that red is blue, and moreover be completely unaware that they have been misled.

Just as a child may learn false beliefs from adult authorities, so too may a machine learning model learn false beliefs from the unquestioned authority that is its training data. Even the most powerful models are beholden to the datasets on which they are trained—datasets that are collected and provided by authorities, who may or may not be trustworthy. Similarly to the deceived child, if a machine learning model were to be trained on data suggesting that red were actually blue, then the model would have no choice but to learn that incorrect belief, and would have no way of knowing—or even suspecting—that not only was what it believed to be “blue” actually “red,” but even that it had been tricked at all. Throughout most of the history of machine learning research, it has been assumed that the data and labels provided by the training dataset are reliable representations of genuine ground truth. However, investigation into *learning from noisy labels* has examined what happens when these assumptions do not hold. From early work on stochastic label flipping to more recent work on adversarial data poisoning attacks, learning from noisy labels has grown into a critical field of fundamental research, revealing many weaknesses and instabilities that have been overlooked in otherwise powerful algorithms.

1.1 Motivation

Machine learning methods in general, and deep neural networks in particular, have recently gained immense popularity, due to their extraordinary capabilities in applications such as computer vision [3], natural language processing [4], finance [5], self-driving cars [6] and medicine [7]. However, despite the success of machine learning in solving these

problems in ideal cases, the problem of *noisy labels* remains outstanding [8]. Especially in cases where datasets may be large and costly to verify, models trained naïvely on such unreliable data are vulnerable to overfitting to any incorrect labels that may be present in the training dataset [9]. The development of algorithmic frameworks that are robust to noisy labels is a critical element of future machine learning research, particularly in sensitive applications, in order to both harden our models against malicious (or even unintentional) false labels, as well as to further broaden our fundamental understanding of machine learning.

1.2 Problem Statement

In the standard scenario of supervised machine learning, it is generally assumed that—regardless of the characteristics of any particular training example—the observed label associated with a given example is its the true class. This assumption allows algorithms to perform parameter optimization using loss functions, which minimize the error between the model’s predictions and the ground truth labels. However, in the case of noisy labels, the observed label may not be the true class for the associated sample. In such a scenario, training models via conventional loss function minimization leads to degradation in generalization performance, as the models overfit to the false labels. In real-world settings, where curated datasets may not be available, it is naïve—if not negligent—to assume that the provided data labels are always accurate. To wit, a machine learning model cannot assume *a priori* that *any* label for any *particular* training instance is accurate; there is always a nonzero probability that, unknown to the model, any label may be incorrect. Faced with such uncertainty, the goal of learning from noisy labels is to design a robust training

scheme that is secure against inaccurate labeling, whether the inaccuracies are unintentional or, worse, malicious.

1.3 Contributions and Scope of Dissertation

This dissertation provides an overview of past and contemporary work on machine learning from noisy labels, including learning from crowds, class-conditional label transition matrices, and instance-dependent noise modeling. We proceed to generalize the concept of instance-dependent label noise, extending it to consider multiple, non-homogeneous labeling processes that each contribute to the training dataset. This generalization, called “labeler-dependent noise” (LDN), models labels to be a function not only of the data features, but also of the specific labeler making the dataset contribution, which may exhibit different degrees or characteristics of labeling errors compared to other labelers. In particular, we present two models for labeler-dependent noise: a simple hammer-spammer framework that wraps a beta-binomial label selection process around the soft labels generated by a single instance-dependent model, and a more sophisticated framework that considers each labeler to be an independent instance-dependent process. We also show theoretically how LDN is a valid generalization of previous label noise models, including class-conditional label noise, instance-dependent label noise, and learning from crowds.

With the LDN model defined, we first tackle learning from crowds as a special case by developing the OpinionRank algorithm. OpinionRank is a nonparametric, graph-based spectral method for ranking the relative reliabilities of an ensemble of imperfect labelers, whose ranking schemes can be used to integrate noisy labels from multiple labelers into a single, more-accurate label. However, despite the success of OpinionRank in outper-

forming other, more computationally expensive approaches, it has a potential limitation in that it requires overlapping, redundant labels from the labeling ensemble—such redundant labels may not always be available in many applications. To address this shortcoming, we introduce a robust framework for machine learning under LDN, called “labeler-aware learning” (LAL), which improves upon OpinionRank by utilizing semi-supervised learning to generate synthetic labels in order to achieve the requisite label redundancy. LAL is a high-level, modular, abstract framework for learning under label uncertainty, and can be applied in a wide variety of learning environments. In this work, we illustrate the necessity of LAL in response to the general LDN model by empirically demonstrating how LAL outperforms existing state-of-the-art approaches when placed in an LDN environment. We also consider the possibility of labeler awareness providing for defenses against malicious data injection attacks from adversarial labelers, and show how LAL can be used as a filtering method against label-based poisoning attacks.

In summary, the main scientific contributions of this dissertation are as follows:

1. Labeler-dependent noise (LDN), a generalization of previous models of label noise that considers heterogeneous noise characteristics;
2. OpinionRank, a fast, nonparametric algorithm for ranking the relative reliability of a set of labelers;
3. Labeler-aware learning (LAL), a general learning framework that reliably filters label noise from multiple noisy labelers; and
4. Consideration of the adversarial perspective of label noise, as well as investigation

into how LAL may be used as a proactive defensive measure to train models robust against adversarial labels.

1.4 Organization of the Dissertation

Chapter 1 provides an introduction, motivating the problem of learning from noisy labels and framing the problem within the contexts of human and machine learning. Chapter 2 provides important preliminary materials and background knowledge about supervised machine learning and approaches for modeling noisy labels that are important for contextualizing the latter portions of the dissertation. In Chapter 3, related works on learning from noisy labels and learning from crowds are discussed, and contemporary state-of-the-art methods are considered in detail. Chapter 4 introduces the first primary contribution of this dissertation, the labeler-dependent noise (LDN) model, both discussing the high-level abstract framework as well as a more concrete example that is used in our demonstrative experiments. Chapter 5 introduces OpinionRank as a solution for learning from crowds, which is a common special case of LDN. In Chapter 6, we extend OpinionRank to the full labeler-aware learning (LAL) paradigm. Chapter 7 introduces the threat of adversarial labelers, first as extreme versions of merely poor-quality labelers, and then as more sophisticated agents using insidious backdoor injections. Chapter 8 presents the results of our experiments for both OpinionRank and LAL compared to previous state-of-the-art methods under the LDN setting, as well as the adversarial robustness of LAL under the threat models discussed in Chapter 7. Finally, Chapter 9 concludes the dissertation, summarizing the work presented herein and outlining the broader impacts that this work may have on future machine learning research.

Chapter 2

Background

2.1 Notation

For clarity and consistency, we first present the nomenclature used in this dissertation in Table 1.

Table 1

Variables and Symbolic Notation Used in this Dissertation

Variable	Description
i, j, k, ℓ	Indexing variables.
$\alpha, \beta, \mu, \sigma$	Parameters of distributions.
x	A data instance.
y	The true label for instance x .
\mathcal{X}^*	The set of all data forming the fiber of \mathcal{Y}^* under \mathcal{F}^* .
\mathcal{Y}^*	The set of all class labels represented in a learning problem.
\mathcal{F}^*	The set of optimal functions mapping each $x \in \mathcal{X}^*$ to the correct $y \in \mathcal{Y}^*$.
f^*	An element of \mathcal{F}^* .
\mathcal{X}	The subset of \mathcal{X}^* observable during the training process.
\mathcal{Y}	The subset of \mathcal{Y}^* observable during the training process.
\mathcal{D}	The set of data $\{\mathcal{X}, \mathcal{Y}\}$ observable during the training process.
f_θ	A function obtained through a machine learning training process.
θ	The model parameters of f_θ , such as neural network weights.

Variable	Description
N	The cardinality of \mathcal{X} .
K	The cardinality of \mathcal{Y} .
J	The total number of labelers.
\hat{y}	The observed label of x (as opposed to true label y).
$p_{k,\ell}$	The class-conditional probability $\Pr\{\hat{y} = \ell \mid y = k\}$.
T	A class-conditional transition matrix. The k^{th} row corresponds to $p_{k,\ell}$.
h	An instance-dependent model. Produces a probability distribution over \mathcal{Y} .
\mathcal{H}	The set of all possible instance-dependent models, $h \in \mathcal{H}$.
y'	The target class of an adversary performing an adversarial backdoor attack.
$\eta(y')$	The adversarial backdoor trigger associated with the target class y' .
x'	An adversarial data instance that has been modified by applying $\eta(y')$.
δ	An indicator function indicating the presence of $\eta(y')$.
\mathbf{v}	The dominant eigenvector of an ergodic Markov chain.
\mathcal{C}	A corroboration matrix obtained via OpinionRank.
\mathcal{W}	Weighted class membership scores produced by OpinionRank.

2.2 Supervised Learning

For a given classification problem, let \mathcal{Y}^* be the set of desired class labels, and let \mathcal{X}^* be the set of all possible data belonging to the classes in \mathcal{Y}^* , such that

$$\mathcal{F}^* : \mathcal{X}^* \rightarrow \mathcal{Y}^* \tag{2.1}$$

represents a set of surjective functions mapping each example $x \in \mathcal{X}^*$ to its correct label $y \in \mathcal{Y}^*$. Then, the problem of supervised learning can be expressed as attempting to find a function $f^* \in \mathcal{F}^*$, given some observed subset $\mathcal{X} \subseteq \mathcal{X}^*$ with associated observed labels $\mathcal{Y} \subseteq \mathcal{Y}^*$. Typically, this goal is achieved by finding an approximate function $f_\theta \approx f^*$, such that

$$f_\theta(x \in \mathcal{X}^*) = \{y \in \mathcal{Y}^* \mid f^*(x) = y\} \quad (2.2)$$

where $\theta(\mathcal{X}, \mathcal{Y})$ represents the parameters of f_θ . Typically, these parameters are found using a technique called *loss function optimization*, where the parameters θ are obtained by iteratively comparing the outputs of f_θ to the observed labels \mathcal{Y} and procedurally updating θ such that the error between $f_\theta(\mathcal{X})$ and \mathcal{Y} becomes small. Loss functions take the form $\mathcal{L}(\mathcal{X}, \mathcal{Y}, \theta)$, and some common examples of loss functions are presented in Table 2.

Table 2

Examples of Common Loss Functions

Name	$\mathcal{L}(\mathcal{X}, \mathcal{Y}, \theta)$
Mean-squared error	$\frac{1}{N} \sum_{i=1}^N [y_i - f_\theta(x_i)]^2$
Hinge loss	$\frac{1}{N} \sum_{i=1}^N \max [0, 1 - y_i f_\theta(x_i)]$
Cross-entropy loss	$-\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log [f_\theta(x_{ik})]$

The mean-squared error loss is commonly used in statistical analysis, though this choice is often made arbitrarily [10]. The general motivating principle behind the mean-

squared error is that it represents the variance of an unbiased estimator, and as such minimizing the mean-squared error also minimizes the variance. Thus, using the mean-squared error as the loss function seeks to obtain the parameters that obtain the best unbiased estimator of the true parameters of the observed data. It is important to note, however, that a biased estimator (such as one obtained via shrinkage methods such as L_p regularization) may yield smaller mean-squared error.

The hinge loss is designed to produce a “maximum-margin” classifier, i.e. one whose decision function has the property of having the maximum distance from observed data on either side of the decision boundary. This property is most notably desirable for support vector machines, and so the hinge loss is sometimes called the “SVM loss”. For predictions $|f_\theta(x_i)| \geq 1$ where $\text{sgn}[f_\theta(x_i)] = \text{sgn}(y_i)$ (i.e. the prediction is correct with high confidence), the hinge loss becomes zero. Otherwise, the hinge loss increases linearly with $f_\theta(x_i)$. A disadvantage of hinge loss is that it does not differentiate between different data points that are correctly classified with high confidence, so it does not continue to optimize for these points.

The cross-entropy loss is the most commonly used loss function when training neural networks. Cross-entropy loss arises probabilistically as the log likelihood of observing a positive or negative instance, so using the cross-entropy loss represents a maximum likelihood estimation for the parameters of the optimal classifier. Minimizing the cross-entropy loss can also be interpreted as minimizing the Kullback-Leibler divergence between an “optimal” class distribution (where, for x_i , the probability mass over \mathcal{Y} is concentrated entirely on y_i) and the normalized class scores produced by a classifier. Unlike hinge loss, which focuses entirely on producing the highest possible accuracy on the training data, the

cross-entropy loss provides a better probabilistic estimation of the parameters approximating $f^* \in \mathcal{F}^*$.

2.2.1 *Semi-Supervised Learning*

Supervised learning is the most common and well-studied type of machine learning. However, in the context of modern machine learning in the era of big data, it is not always feasible to obtain labels for every sample in the training dataset. In the case where the training dataset is split into a labeled subset and an unlabeled subset, the paradigm shifts to *semi-supervised learning*. Observing the scope of this dissertation, we eschew the theoretical underpinnings that drive semi-supervised learning techniques¹. Instead, we focus on the expanded capabilities afforded to the machine learning practitioner by the existence of semi-supervised techniques. In particular, while the fundamental objective of semi-supervised learning remains the same as that of fully supervised learning (that is, to find $f_\theta \approx f^* \in \mathcal{F}^*$), the extension to include unlabeled data allows for more creative engineering solutions for problems such as data mining and learning from noisy labels. Two such solutions include DivideMix [12] (discussed in detail in Section 3.2.1), and the labeler-aware learning framework proposed in this dissertation (Chapter 6).

2.3 Noisy Labels

Loss function optimization is powerful, and has been proven to be effective both theoretically and empirically. However, learning from labeled data using loss function optimization relies on the availability of accurate labels; if observed labels do not represent

¹For a more formal treatment and presentation of semi-supervised learning, we refer the reader to the excellent textbook *Semi-Supervised Learning* by Chapelle, Scholkopf, and Zien [11].

the true classes, then loss function optimization may produce undesirable functions as the method overfits to false label information.

In modern contexts, obtaining accurate labels on large datasets is costly, sometimes prohibitively so [13, 14]; these costs have led to crowdsourcing as an attractive and cost-effective solution for distributed label gathering [15, 16]. Unfortunately, the labels obtained via crowdsourcing are of unreliable veracity, and models trained naively on such unreliable data are vulnerable to overfitting on noisy labels [17]. Thus, learning in the presence of noisy labels has emerged as an area of active research [18, 19].

2.3.1 *Class-Conditional Label Noise (CCN)*

Label noise is commonly treated as a class-conditional phenomenon, where the noisy labels are considered to be strictly a function of the true label [20]. Under this treatment, label corruption is modeled by assuming that given a sample x with a true label y_k , the observed label may be flipped to a false label \hat{y}_ℓ with some probability $p_{k,\ell}$. The full set of these probabilities for all k, ℓ then forms a *transition matrix*, often denoted as T [21]. Thus, a class-conditional noisy label for data instance x_i is produced by following a categorical distribution over the label space,

$$\hat{y}_i \sim \text{Cat}[\mathcal{Y} \mid T_{y_i}] \tag{2.3}$$

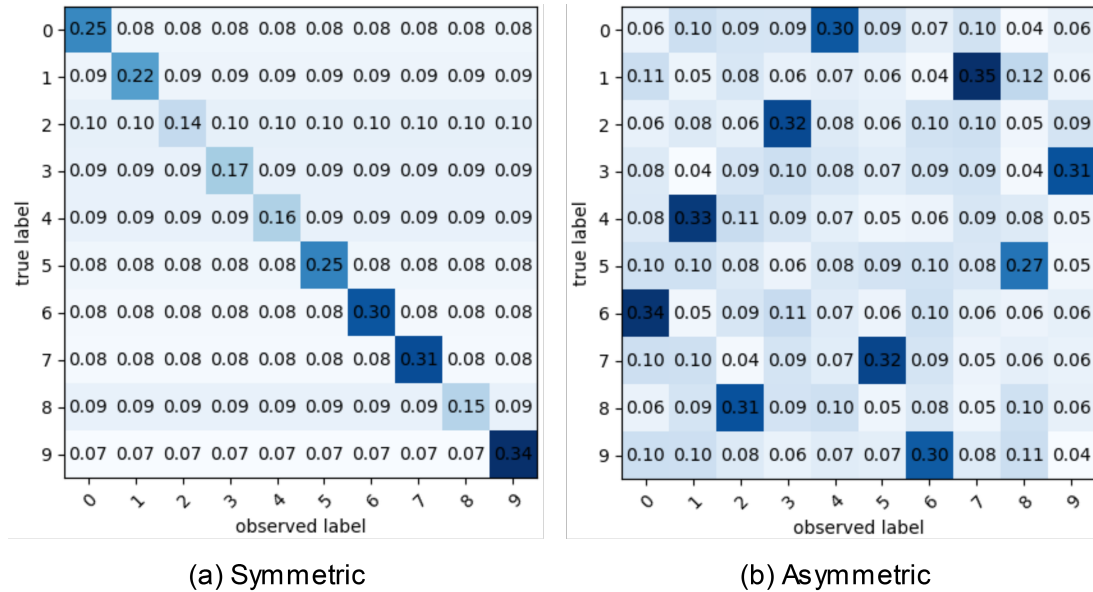
where \hat{y}_i is the observed noisy label, and T_{y_i} represents the transition probabilities corresponding to the (unobserved) true label y_i .

Two common forms of T used in studying learning from noisy labels are *symmetric* and *asymmetric* label noise [22]. Under symmetric noise, each row of T follows a uniform

distribution, with each label y_k having an equal probability of corruption to any other label y_ℓ [23]. In contrast, under asymmetric noise T is artificially constructed to place higher probabilities on classes heuristically similar to y_k [24]. Examples of symmetric and asymmetric class-conditional transition matrices are shown in Figure 1.

Figure 1

Examples of Class-Conditional Transition Matrices



2.3.2 Instance-Dependent Label Noise

Class-conditional label noise is an attractive scheme for studying label noise, as it allows for precise control over the amount of label noise injected into the training dataset. However, CCN approaches fail to consider the data-driven dependencies of label generation

processes, and produce unrealistic data-label distributions. More realistic models include early work on feature-dependent label noise [25], and, more recently, instance-dependent noise [26]. One common approach generates instance-specific noise by training a deep neural network on the clean training dataset and using its noisy outputs; some authors use the entire set of noisy labels generated by the network [27], while others retain only a percentage of the noisy labels [28]. Another method involves projecting each instance onto a randomly-sampled set of K vectors and sampling from a combination of this projection with the clean label to generate a noisy label [29]. The polynomial margin diminishing noise introduced in [30] extends confidence-based noise to stochastically flip labels based on the outputs of a neural network trained on clean labels.

All instance-dependent models can be summarized abstractly as producing an instance-dependent probability distribution over the label space \mathcal{Y} as a function of the data features x_i . Thus, an instance-dependent noisy label is produced by following

$$\hat{y}_i \sim \text{Cat}[\mathcal{Y} \mid h(x_i)] \tag{2.4}$$

where \hat{y}_i is the observed noisy label for the data instance x_i , and h represents an instance-dependent model (which may involve arbitrary parameters) belonging to \mathcal{H} , the set of all possible instance-dependent models. Notably, there is wide variation in the expression of h , and indeed h may be parameterized arbitrarily (for example, as the decision function of a deep neural network).

Note the similarities between Equation 2.3 and Equation 2.4. In particular, instance-dependent noise generalizes class-conditional noise, since any class-conditional transition matrix may be expressed as an instance-dependent model where all instances belonging

to the same class have identical transition probabilities, i.e. if $h(x) = T_{y_i}$ for all instances $\{x \in \mathcal{X} \mid f^*(x) = y_i\}$. A similar generalization argument will be used in Chapter 4 to show that any instance-dependent noise model may be expressed as a special case of labeler-dependent noise.

2.3.3 Noisy Labels from Crowdsourcing

Both class-conditional and instance-dependent noise models assume that the noisy labels are generated by a single process for the entire training dataset. This is an unrealistic assumption in the context of big data, where labels may be gathered via distributed methods such as crowdsourcing. Crowdsourcing has emerged as an appealing, inexpensive tool for distributed collection of label information for large-scale datasets. For example, the widely used ImageNet dataset is annotated using crowdsourcing from Amazon Mechanical Turk [31]. Crowdsourcing is also widely used to annotate datasets in natural language processing [32, 33] and data mining [34, 35]. However, crowdsourcing suffers from the problem of inexpert—and therefore unreliable—labeling. The very property of widespread contribution that gives crowdsourcing its power also results in the significant drawback of relying on the opinions of “experts” (sometimes referred to as labelers, annotators, workers, or label sources) who may have little or no domain knowledge. Furthermore, there may be differences of opinion between labelers with contrasting expertise: one labeler may provide a label (which could be considered correct from one perspective) that disagrees with a different label (which could be considered correct from a different perspective). For example, recent works that have analyzed the ImageNet dataset found that using the crowdsourced labels as a gold standard may be flawed [36, 37].

Addressing unreliable labels has long been a fundamental objective of learning from crowds, and can be traced to Dawid and Skene’s seminal work on modeling the accuracy of each labeler as a hidden confusion matrix [38]. The generative model of labels, abilities, and difficulties (GLAD) model adds instance-specific dependence by modeling the difficulty of correctly labeling each instance alongside each labeler’s expertise [39]. A multidimensional parameterization assuming that labels are generated from a Gaussian mixture model was proposed in [40]. In general, noisy label models based on crowdsourcing attempt to describe the label generation process as a parameterized set of latent variables, with the various models in the literature differing only in how these parameters are expressed.

2.4 Adversarial Machine Learning

The scenario most similar to Descartes’ evil demon—mentioned in Chapter 1—is that of adversarial machine learning. Adversarial machine learning is the field of machine learning concerned with studying the robustness of machine learning models to maliciously-designed inputs, which may force the models to produce unpredictable or undesirable outputs, or to learn incorrect beliefs about the data on which they are trained. Broadly speaking, adversarial attacks against machine learning models fall into one of two categories: *evasion* attacks, which aim to force the model to produce misclassification of perturbed test (inference) samples [41], and *poisoning* attacks, which inject the training dataset with malicious examples (of deliberately incorrect labels) in order to induce poor generalization performance [42]. Of these two categories, learning from noisy labels is most strongly associated with the latter, as poisoning attacks target the training process directly, while evasion attacks are typically designed and executed against pretrained models.

2.4.1 Backdoor Attacks

One particularly insidious type of hybrid attack (combining elements of both evasion and poisoning attacks) is the *backdoor attack* [43], which seeks to force misclassification *only* in the presence of a backdoor trigger, while otherwise allowing the network to operate unimpeded. Such a strategy is especially difficult to detect or defend against, as the victim of such an attack will have no indicators of any unusual behavior until the targeted attack is executed in deployment. Backdoor attacks have been proposed as potentially catastrophic vulnerabilities in sensitive applications such as self-driving cars [44] and facial recognition security systems [45].

An adversary who wishes to force a trained model to misclassify specific test samples as a target class $y' \in \mathcal{Y}$ can corrupt a training example $x \in \mathcal{X}$ by

$$x' = x + \eta(y') \tag{2.5}$$

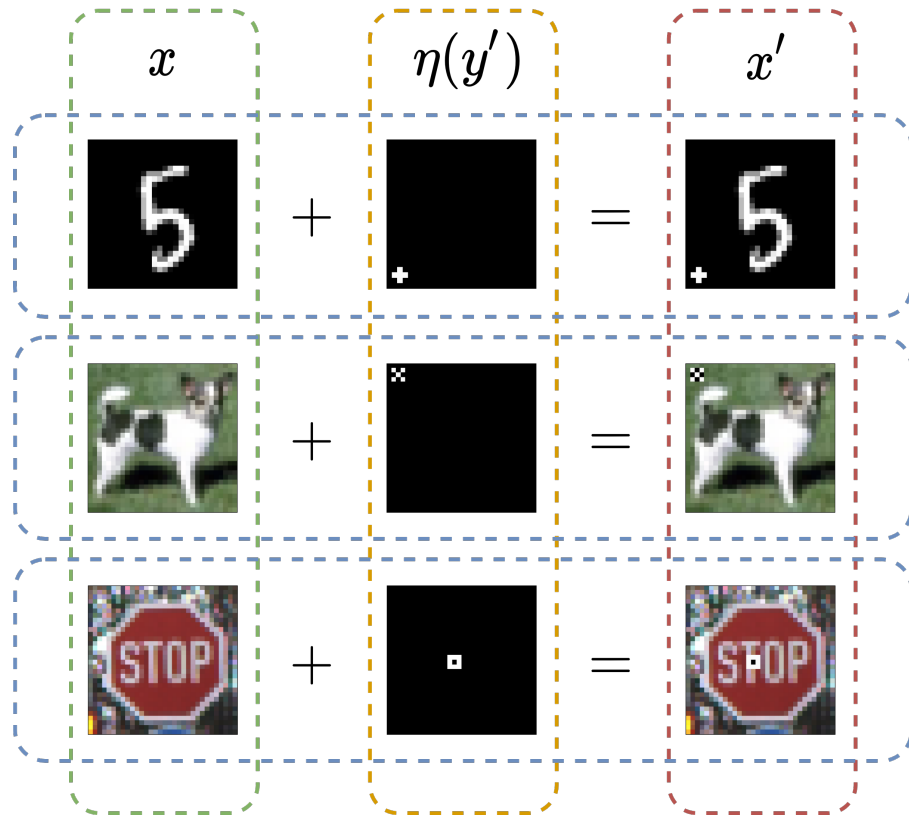
where $\eta(y')$ represents a backdoor pattern associated with class y' . In doing so, the adversary attempts to force θ to be learned such that

$$\begin{aligned} f_{\theta}(x') &= x_{\theta}[x + \eta(y')] \\ &= (1 - \delta)x_{\theta}(x) + \delta f_{\theta}[\eta(y')] \\ &= (1 - \delta)y + \delta y' \end{aligned} \tag{2.6}$$

where δ represents an indicator function that takes the value of 1 if $\eta(y')$ is present, otherwise 0. A visual example of applying Equation 2.6 is shown in Figure 2.

Figure 2

Examples of Applying Backdoor Attacks



Chapter 3

Related Work

While learning from noisy labels is a relatively recent field (compared to supervised learning from clean labels) there is still an existing—and growing—body of literature investigating the subject. In this chapter, we review prior art on label noise models, as well as a representative selection of approaches for learning under these models.

3.1 Assuming the Existence of Clean Validation Data

Before introducing any specific models of label noise, it is important to address a common misunderstanding regarding the learning environment under noisy labels. Some works assume the existence of a clean validation dataset and leverage this information against the noisy labels (for example, [46–49]). This assumption relies on the observation that while it may be costly and inefficient to obtain curated labels for the entire training dataset, it may be less prohibitive to obtain such labels for a smaller subset of the training data. Hence, these methods use validation data, which boasts—in theory—perfectly accurate labels, as a seed from which the uncertain, noisy labels in the larger training dataset may be refined.

However, the assumption of clean data is flawed. In general, there is no such thing as a perfectly accurate classifier; even human experts cannot provide 100% accuracy with their labels in all scenarios. More abstractly, given a classifier f_θ (which may be a human or machine labeler), the error of f_θ on the universal set \mathcal{X}^* must necessarily be greater than zero; a perfectly-trustworthy oracle does not exist, as there is always some nonzero uncertainty in the accuracy of the labels provided by f_θ ¹. Thus, the notion

¹In the field of uncertainty quantification, these uncertainties are sometimes referred to as “aleatoric and epistemic uncertainty”. These concepts are interesting, and worthy of deeper investigation, but lie beyond the scope of this dissertation.

that it is even possible to obtain a validation dataset whose labels are clean with 100% confidence is unreasonably optimistic. Furthermore, in forward-thinking applications such as autonomous continual learning agents and independent AI free from human influence, the noisy nature of unstructured data in uncontrolled environments renders these approaches unreliable.

3.2 Class-Conditional Label Noise

As discussed in Section 2.3.1, class-conditional label noise (CCN) takes the form of a transition matrix, T . As one of the simplest approaches for modeling the incidence of label noise, CCN is consequently one of the most popular models in machine learning literature for studying label noise. Accordingly, there is a wide range of angles of attack that have been taken toward tackling CCN.

Some works have attempted to exploit model confidence with small losses to weight the importance of each instance during parameter updates [24, 50, 51]. Multiple methods for correcting noisy labels based on loss modeling have been proposed [52, 53]. Meta-learning has been proposed as a method for increasing robustness to noisy labels [13, 54, 55]. In all cases, label noise is treated strictly as a function of the true label, meaning that the probability of label flipping for any instance of a particular class is shared across all instances.

In this dissertation, we give special consideration to the most notable, popular algorithm for learning from noisy labels under class-conditional label noise: DivideMix [12].

3.2.1 *DivideMix*

DivideMix is motivated by the key insight that deep neural networks tend to exhibit smaller losses for data with clean labels compared to data with noisy labels [56]. Following this observation, DivideMix uses the expectation-maximization algorithm to fit a Gaussian mixture model to the per-sample loss distribution of the training data after each training epoch, under the assumption that the loss distribution will be bimodal (with cleanly-labeled data clustered with small losses, and noisy labeled data clustered with large losses). Hence, the training dataset is split into two subsets: one with theoretically mostly-clean labels, and one with theoretically mostly-noisy labels.

In order to avoid self-reinforcing confirmation bias, DivideMix actually trains two networks simultaneously, in a strategy called “co-teaching” [24]. During each training epoch, the output loss distribution of one of the networks is used to generate the training data splits for the other network. Then, these splits are used in a semi-supervised manner to train the other network by stripping the labels from the mostly-noisy subset, and treating this subset as an unlabeled dataset (alongside the labeled dataset with mostly-clean labels). The authors use a hand-crafted modification of the MixMatch algorithm [57] to perform semi-supervised learning under noisy labels. They also incorporate a “warm up” period during the first several training epochs where no loss modeling is performed, and apply a confidence penalty to the networks’ outputs during the warm up period in order to flatten the initial loss distribution.

3.3 Instance-Dependent Label Noise

As outlined in Section 2.3.2, instance-dependent label noise (IDN) seeks to update the incorrect assumption of CCN, that noisy labels are generated purely as a function of the true labels. Instead, IDN considers that each unique instance in the training dataset may have a concordantly unique probability of flipping to any other label, independently from the other instances (even those belonging to the same class). Unlike CCN, there are many different proposed models for IDN, each of which are accompanied by hand-crafted approaches for learning under their proposed frameworks.

The bounded instance- and label-dependent noise approach attempts to extract samples with the same labels as those produced by a Bayes optimal classifier based on the implicit or explicit knowledge of the upper bounds of the noise rates [29]. Part-dependent transition matrix estimation assumes that the instance-specific transition matrices can be learned by exploiting clean samples [26]. Covariance-assisted learning utilizes second-order statistics to make the peer loss (proposed in [58]) invariant to instance-dependent noise [59].

In this dissertation, we give special consideration to two recent, powerful methods proposed for learning under instance-dependent noisy labels: progressive label correction and self-evolution average label. Both methods have been shown to be effective at mitigating the effects of instance-dependent label noise (under their respective assumptions about the label noise model).

3.3.1 Progressive Label Correction

Progressive label correction (PLC) [30] seeks to address the authors’ proposed polynomial margin diminishing noise (PMD) model by performing iterative label correction according to a threshold on the prediction confidences. Under the assumption that the label noise present in the training data follows the PMD model, PLC asserts that there exists a “pure region” in which the confidence of a neural network is high, and in which the prediction of the network is consistent with a theoretical Bayes optimal classifier trained on (hypothetical) clean labels. Thus, any prediction with a suitably high confidence is treated as correct, and if the observed label \hat{y}_i differs from the prediction $f_\theta(x_i)$, then \hat{y}_i is changed to the predicted class.

PLC starts with a high confidence threshold, and trains the neural network until there are no further label corrections. Then, the confidence threshold is lowered by some small amount and the training process is repeated. These two steps are alternated until training reaches convergence. The authors prove that under the assumption of PMD noise, and given certain conditions regarding the hypothesis class of the machine learning model as well as the underlying data distribution $P_{\mathcal{X}^*}$, the PLC algorithm produces a classifier whose empirical risk asymptotically approaches the true risk of a Bayes optimal classifier in the limit of infinite data.

3.3.2 Self-Evolution Average Label

The self-evolution average label (SEAL) algorithm stores the running average of a classifier’s predictions over training, and then iteratively retrains using the averages as

soft labels [28]. SEAL operates under the assumption that there exists a latent optimal distribution describing the true label y_i for each instance x_i , and that this distribution can be considered as the output of an oracle classifier. Then, under this assumption, the label corrections obtained by taking the average classifier output over a complete training process have both lower bias and lower variance than the observed noisy labels compared to the latent optimal distribution. The authors claim that by iteratively retraining by replacing the original noisy labels with the average labels obtained from the previous training iteration, SEAL gradually approaches the latent optimal distribution in expectation. SEAL’s most notable advantage over previous methods that have attempted to use label correction is that it does not require any special hyperparameters beyond those required for the chosen classifier.

3.4 Learning from Crowds

Perhaps the earliest work on characterizing the collective decision of a set of inexpert opinions is the Condorcet jury theorem [60], which states that for a group of independent voters with a homogeneous probability of correctness p , the probability of their majority vote being correct on a binary decision increases with the size of the group if $p > 0.5$. Kazmann showed that the assumption of homogeneous voters can be relaxed by assuming that the heterogeneous voter correctness probabilities follow a symmetrical distribution about a mean \bar{p} [61]. Grofman demonstrated that the group’s collective accuracy can increase even if the added members are less competent than the group’s previous average [62]. Owen, Grofman, and Feld removed the distribution restrictions, generalizing the theorem to depend only on $\bar{p} > 0.5$ [63]. List and Goodin extended this result to problems

with more than two classes, showing that for a K -class problem, the average voter reliability needs only to exceed $1/K$ for the majority vote decision to be increasingly more accurate as the number of voters increases. [64].

Beyond simple majority vote analysis, and the associated large body of ensemble-based approaches, much research has gone into investigating crowdsourcing algorithms. Dawid and Skene proposed a model based on the well-known expectation-maximization (EM) algorithm, attempting to estimate each labeler’s respective expertise as a confusion matrix [38]. From their application, a rich body of work has sprouted, with many improvements, alterations, and theoretical bounds on the performance of generative models based on the EM approach [65–67]. Aside from EM-based approaches, Ghosh et al. [68] and Dalvi et al. [69] proposed sparse matrix algorithms based on singular value decomposition. Other approaches have investigated Bayesian inference [40, 70]. An intriguing line of investigation by Goldberger examined the problem of “soft” labels, which take the form of a distribution over the class space [71]. More recently, the success of deep neural networks has prompted work on deep generative models [72–74]. Regardless of the specific approach, nearly all work in this area attempts to model either the parameters of labelers’ reliabilities, or the confusion matrices associated with each labeler.

In this dissertation, we give special consideration to three models for learning from crowds. The first two settings are well-established and widely cited, and can be considered to be canonical models; while more recent work has proposed some additions to these models, the fundamental idea of highly parameterized probabilistic dynamics is consistent across the literature. The third model adds a new layer of complexity to the label generating process that cannot be easily incorporated into the previous models.

3.4.1 Generative Model of Labels, Abilities, and Difficulties

We first consider the setting proposed by Whitehill et al. [39], which models the labeling process as not only a function of the *expertise* of the labeler, parameterized by $\alpha \in (-\infty, \infty)$, but also as the *difficulty* of labeling a data instance, parameterized by $1/\beta \in [0, \infty)$. When $1/\beta = \infty$, an instance is deemed very difficult to label for even the most expert labeler, whereas $1/\beta = 0$ represents a trivial instance, so obvious that anyone would label it correctly irrespective of expertise. The range of values for α describes expertise from “perfectly wrong” (when $\alpha = -\infty$) to “perfectly accurate” (when $\alpha = \infty$), with $\alpha = 0$ representing random guessing. The probability of the label \hat{y}_{ij} —assigned by labeler j on instance i —being the correct label $y_i \in [0, 1]$ is then modeled as

$$\Pr(\hat{y}_{ij} = y_i | \alpha_j, \beta_i) = \frac{1}{1 + e^{-\alpha_j \beta_i}} \quad (3.1)$$

which allows the log odds for the label being correct to be expressed as

$$\log \left[\frac{\Pr(\hat{y}_{ij} = y_i)}{1 - \Pr(\hat{y}_{ij} = y_i)} \right] = \alpha_j \beta_i. \quad (3.2)$$

From this formulation, the authors develop an EM-based algorithm called GLAD (**G**enerative model of **L**abels, **A**bilities, and **D**ifficulties), which—under the assumptions of the labeling model—is able to recover y , α , and β for all data and labelers.

To our knowledge, Whitehill et al. were the first to extend the work of Dawid and Skene (who assumed that labels were generated only by parameters over the labelers’ expertise [38]) such that the generative model also included parameters for the difficulty of assigning the correct label. While adding a small amount of complexity, Whitehill et al. sought to keep their model as simple as possible, assigning only a single hidden parameter

for each instance and labeler.

3.4.2 *Multidimensional Parameterization of Label Generation*

In contrast to the GLAD model, Welinder et al. considered the labeling process to be a high-dimensional system, with both instance difficulty and labeler ability being governed by many parameters [40]. For their formulation, the authors suggest that the dynamics can be described as a Gaussian mixture model. For the i^{th} instance, with ground truth label $y_i \in [0, 1]$, the presentation x_i of instance to labeler j is modeled by

$$x_i \sim \mathcal{N}(\mu_z, \sigma_z^2)$$

$$\mu_z = \begin{cases} -1 & \text{if } y_i = 0 \\ 1 & \text{if } y_i = 1 \end{cases} \quad (3.3)$$

where σ_z is a parameter describing the variability in the difficulty of correctly labeling instance x_i . The j^{th} labeler sees a version of instance x_i modeled as $\hat{x}_{ij} = x_i + n_{ij}$, where n_{ij} is the labeler- and instance-specific “noise”, such as differences in labeler attention, acuity, direction of gaze, etc. The noise statistics vary from labeler to labeler, and are modeled as a parameter σ_j ; the authors assume Gaussian noise, i.e. $\hat{x}_{ij} \sim \mathcal{N}(x_i, \sigma_j^2)$. The labeler-assigned label \hat{y}_{ij} is then chosen deterministically as $\hat{y}_{ij} = \mathbb{I}(\langle \hat{w}_j, \hat{x}_{ij} \rangle \geq \hat{\tau}_j)$, where $\mathbb{I}(\cdot)$ is the indicator function and \hat{w}_j is a weighting vector that encodes each labeler’s expertise. The authors draw the decision threshold according to τ_j following a zero-mean Gaussian, and sample the noise parameter σ_j from a gamma distribution. The authors then apply Bayesian maximum a posteriori estimation to maximize the posterior on the parameters. They solve this optimization using gradient ascent by alternating between fixing x and

optimizing over (w, τ) , and fixing (w, τ) and optimizing x , assuming Gaussian priors on w_j and τ_j , respectively.

The generative model proposed by Welinder et al. is highly parameterized and considerably more complex than the model suggested by Whitehill et al. Later models build on this idea, adding even more parameterization in the form of latent variables; these more complicated models are intractable for EM algorithms, leading to the usage of deep neural networks for approximating these more complex environments [72–74]. Each of these models, however, shares lineage with the work of Welinder et al., and the general class of multidimensional label generation dynamics is well-represented by their work.

3.4.3 *Soft Observations of Labeler Opinions*

Goldberger introduced the notion of soft opinions, where categorical label assignment of the j^{th} labeler on the i^{th} data instance (with true label y_i) is not a one-hot encoding, but rather a probability distribution [71]. He simplified the initial annotation process compared to the previous models, assuming only that the labeler’s initial opinion q_{ij} is assigned following

$$\Pr(q_{ij}|z_i = a; p_j) = \begin{cases} p_j, & \text{if } y_i = a \\ \frac{1-p_j}{|A|-1}, & \text{if } y_i \neq a \end{cases}, \quad \forall a \in A, \quad (3.4)$$

where p_j is the labeler’s probability of providing the correct label, and A is the set of possible labels. Similarly to class-conditional models, Goldberger assumed that each labeler has an identical reliability across all classes, and that an incorrect label is assigned following a uniform distribution across the incorrect classes.

Goldberger’s most notable contribution is that he extends the uncertainty in observed

labels beyond that of the labeler's label generation model. He assumes that the *observed* version of the labeler opinion, \hat{y}_{ij} , is not an indicator, but rather a probability distribution over all visible labels:

$$\hat{y}_{ij}(b) = \Pr(y_{ij} = b), \quad \forall b \in A. \quad (3.5)$$

In this way, Goldberger accounts for a layer of obfuscation between the labels as provided by the labelers and the labels as seen by the observer. This consideration is intriguing, and adds an important contribution that is missing from the previous models, extending the assumption of unreliability from simply the *generation* of labels to the *observation* of the labels. Goldberger handles this obfuscation by developing an extended EM algorithm.

Chapter 4

Labeler-Dependent Label Noise

In this chapter, we introduce one of the primary contributions of this dissertation: labeler-dependent label noise (LDN). In Chapter 5, we will address the common problem of learning from crowdsourced labels as a special case of LDN with the OpinionRank algorithm. Then, in Chapter 6 we will extend the capabilities of OpinionRank to be able to handle the full LDN model. Afterward, in Chapter 7 we will discuss the implications of adversarial perspectives of the LDN model, and demonstrate how the LDN model can be used to describe real-world adversarial settings.

As previously shown, instance-dependent label noise (IDN) models represent an important improvement over unrealistic class-conditional approaches (CCN). However, while there are many proposed models for IDN, to date all such models are assumed to be applied *homogeneously* across the entire dataset. This assumption is severely limiting in the case of modern datasets, where the characteristics of label noise may vary based on not only the data features, but also on the specific labeler providing the noisy label. Similarly, while methods for learning from crowds have attempted to consider both instance- and labeler-dependent noise, these methods require unrealistic label redundancy, and rely on overly-specific parameterizations based on assumptions about the label generation process.

We propose a cross-disciplinary approach to modeling label noise by combining the strengths of single-process IDN models with the multiple-labeler paradigm of learning from crowds. We will show in Chapter 6 how this extension of IDN to account for multiple labelers eliminates the weaknesses of both IDN and learning from crowds by developing a robust learning framework.

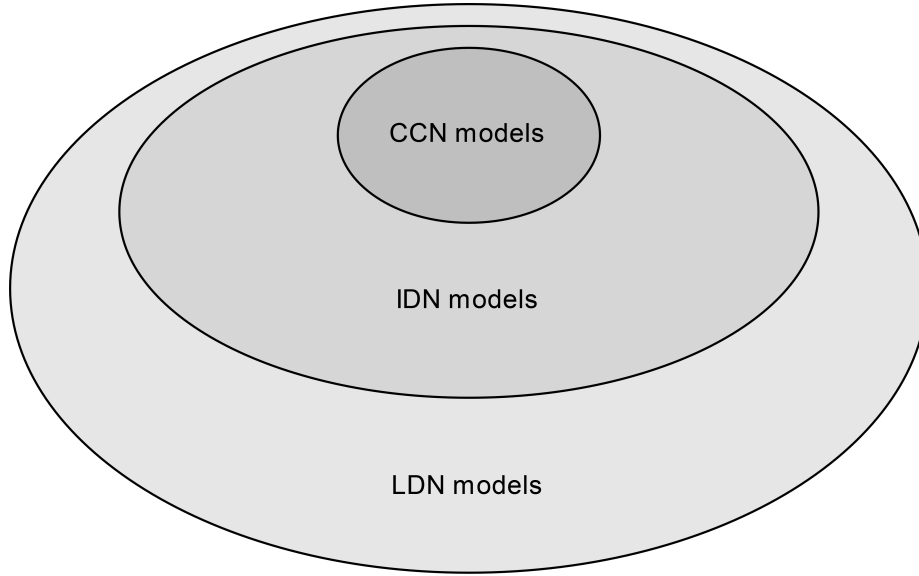
We introduce our LDN model as a generalization of IDN by adding a labeler index to the instance-dependent form:

$$\hat{y}_{ij} \sim \text{Cat}[\mathcal{Y} \mid h_j(x_i)] \quad (4.1)$$

where $h_j \in \mathcal{H}$ for all labelers $j = 1, \dots, J$. While seemingly simple, the addition of the labeler index j creates complications in analytically determining the properties of label noise. In particular, it is very unlikely that all labelers will follow the same instance-dependent process h —that is, $h_j \neq h_\ell$ for $j \neq \ell$ —so considering J labelers introduces another dimension of parameters, which has not been previously considered. We note that every single-process IDN model (as described by Equation 2.4) can be recovered as a special case of LDN (Equation 4.1) by assuming the (unlikely) scenario where $h_j = h$ for all j , or by assuming that the entire dataset is provided by a single labeler (i.e., the number of labelers $J = 1$, so $h_j = h_1 = h$). Furthermore, Equation 4.1 holds without loss of generality *regardless* of the characteristics of *any* h_j , or indeed regardless of the number of labelers J . The ideal case of perfect, non-noisy labels can also be recovered, by assuming that $h_j = h^*$ for all labelers, where $h^*(x_i) = y_i$ is an oracle function that universally provides the correct label. The relationship of LDN as a generalization of IDN is similar to the relationship between IDN and CCN; hence, the tiers of generalization form a hierarchy of models. This hierarchy is illustrated in Figure 3.

Figure 3

Hierarchy of Label Noise Models



In this dissertation, we consider a hammer-spammer model of labeler-dependent noise. In particular, we assume that for a given instance x_i , the set of all possible classes to which x_i may belong are ordered by relative likelihood of correctness, and that this ordering may be estimated by an instance-dependent process h . Given such an ordering, the noisy label may be selected probabilistically from the set of possible classes, with hammers having a high probability of providing the highest-ordered label, and spammers having a low probability of providing the highest-ordered label. Unlike previous works, which used either uniformly-distributed noisy label distributions or heuristic, handcrafted label flipping probabilities, we choose the beta-binomial distribution to describe the probability of the observed label \hat{y}_i taking the value of a particular class in \mathcal{Y} given the data features x_i . The beta-binomial distribution is especially useful for describing distributions over

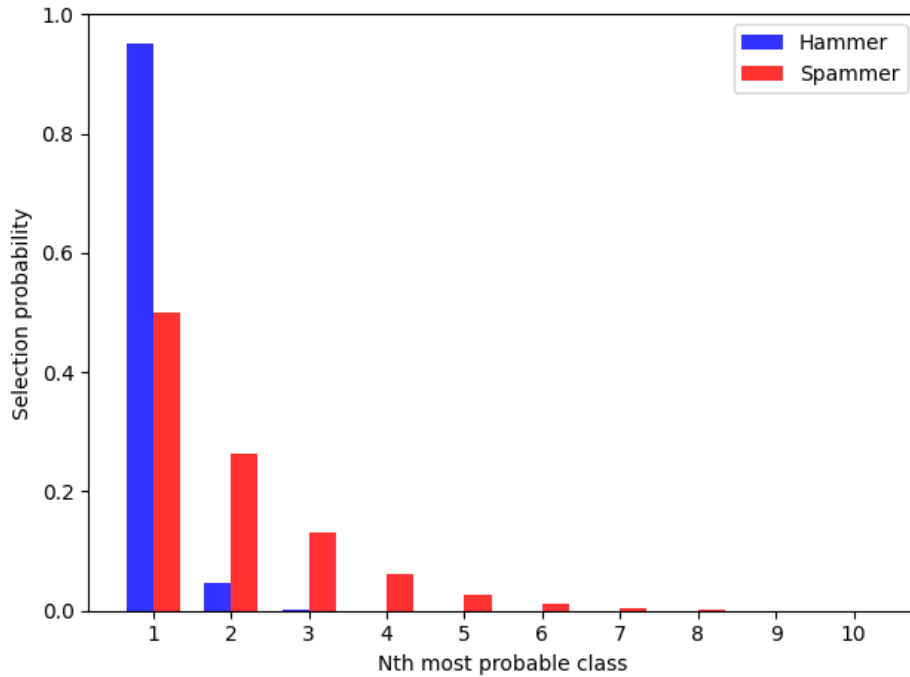
discrete values (here, classes in \mathcal{Y}) that follow smooth probabilistic ordering (here, sorting by most-to-least likelihood of observation). Hence, the labeler-dependent model $h_j(x_i)$ is described by

$$h_j(x_i) = \text{BetaBinom}[h(x_i) \mid \alpha_j, \beta_j] \quad (4.2)$$

where α_j and β_j represent the parameters of the beta-binomial distribution describing the class selection likelihoods of the j^{th} labeler. In particular, by fixing $\alpha_j = 1$, and allowing $\beta_j \in [1, \infty)$, we define a spectrum in which a potential labeler behaves, with $\beta_j = 1$ representing a completely random labeler and $\beta_j = \infty$ representing a perfectly-accurate labeler. Figure 4 demonstrates two examples of such labelers, with a hammer defined as $\beta_j = 200$, and a spammer as $\beta_j = 10$ (over ten total classes). Note that a hammer has a high probability of assigning the true label to an instance, and a low (but nonzero) probability of assigning any other label (with decreasing monotonicity according to the instance-dependent class likelihoods). In contrast, the probability mass for the spammer is distributed more broadly across the class space \mathcal{Y} . Most importantly, for both types of labelers the incorrect labels are distributed with *decreasing probability* according to instance-dependent class likelihoods.

Figure 4

Examples of Class Label Selection Probabilities













4.1 Illustration of Labeler-Dependent Noise

Visual examples of instance-dependent noise violating the class-conditional assumption for label noise by generating feature-dependent labels can be found in previous works, such as [28]. Here, we extend these illustrations, presenting labeler-dependent noise as a generalization of instance-dependent noise by demonstrating how different labelers may provide heterogeneous labels. We generated instance-dependent class ordinals by intentionally overfitting a neural network on the training data of the MNIST dataset and, for each instance, sorted the set of classes by the class probabilities produced by the neural network. Then, for each of seven hammers and three spammers, we randomly selected a class label following the respective beta-binomial distributions over the ordered classes.

Table 3 shows how the labels produced by each labeler vary, depending on the hammer-spammer characteristics of each labeler. Each column corresponds to the labels provided for the specific instance shown at the top of the column, and the labels in each row were provided by the same labeler. We observe that even within just the pool of hammer labelers, there is wide variability between the labelers with respect to both the incidence and classes of incorrect labels. These variations illustrate the necessity of labeler-dependent noise modeling; if label noise were simply instance-dependent, then we would expect to see the same incorrect labels—on any given instance—provided by all labelers. The fact that this is not observed suggests that noisy labels have labeler dependence. Indeed, while the incorrect labels in each column *tend* toward a most-common incorrect class (i.e., the second-highest probability class ordinal), each labeler’s unique characteristics adds label noise heterogeneity across both the instance and labeler dimensions.

Table 3*Examples of Labeler-Dependent Noisy Labels on MNIST*

										
True label	0	1	2	3	4	5	6	7	8	9
	1	1	2	3	9	5	2	3	8	9
	0	1	6	3	4	5	0	7	8	4
	0	4	2	8	4	6	6	4	4	9
Hammers	4	8	2	3	4	5	8	7	8	3
	0	1	0	3	2	5	6	7	9	7
	0	1	2	7	4	5	8	9	8	9
	7	8	2	3	4	5	6	7	8	7
	5	8	9	2	4	7	0	5	9	9
Spammers	6	6	9	7	9	6	6	8	9	7
	7	1	6	1	1	6	2	9	1	7

Chapter 5

OpinionRank

In this chapter we consider the problem of learning from crowds as a special case of labeler-dependent noise. As discussed in Section 3.4, methods for learning from crowds typically view the label generation process as a highly-parameterized latent variable model, and attempt to estimate these parameters in order to obtain clean labels. However, parameter estimation methods depend highly upon the correct parameterization of the system, and can fail in alternative environments. Even worse, techniques such as expectation-maximization and variational deep neural networks demand substantial computational requirements to converge to their estimates of the system dynamics.

To address these drawbacks, we propose OpinionRank, a spectral algorithm for labeler ranking and weighted voting. Instead of attempting to estimate the *true* reliability parameters of each labeler in the ensemble, we propose to estimate the *relative* reliability of each labeler with respect to others. Furthermore, we do so using a nonparameterized approach: given only the observed labels (of unknown reliability) provided by each labeler, we compute our estimation of the labelers' relative expertise by comparing the *frequency of agreement* between each pair of labelers. An agreement between two labelers can be interpreted as a soft "recommendation" between them: given that they have provided the same label for the same instance, it is reasonable to expect that one labeler would recommend the other at least some of the time. Under the Condorcet criterion that the average expertise of the ensemble of labelers exceeds random guessing [64], this system of mutual recommendations builds a network of trust. This network of trust reflects recent research into the characteristics of early childhood knowledge acquisition: information

sources that have been shown to agree with other information sources are viewed more favorably and with greater trust than those who have not [75].

We formulate the ensemble of labelers as a fully connected graph, with each labeler functioning as a node. The edges of each node correspond to the number of times that each labeler i agrees with (recommends) each other labeler j (a labeler always recommends itself). We consider the probabilistic interpretation of the frequency of labeler i recommending any other labeler j ; the edges leading outward from any labeler i can be transformed into a probability distribution via normalization. We interpret the graph of interlabeler agreements as a Markov chain representing the scenario of the model asking each labeler to recommend the opinion of any other labeler. Hence, the long-run steady-state probabilities of this Markov chain represent the probability that the model will “trust” the opinion of any *particular* labeler. The edge probabilities of recommendation form a dense transition matrix, which we call a *corroboration* matrix. We choose the partition function

$$Z = \sum_i e^{x_i} \tag{5.1}$$

as a normalization term, transforming each row into a Gibbs measure and thus guaranteeing that the corroboration matrix is ergodic.

The Perron-Frobenius theorem guarantees that the corroboration matrix—being real, square, and positive—will have a unique, positive eigenvector [76]; this dominant eigenvector represents the steady-state probabilities of the corroboration matrix, which we use to describe the relative reliabilities of each labeler. The dominant eigenvector \mathbf{v} can be computed using the well-known power iteration method, taking the limit of the matrix power of the transformed corroboration matrix \mathcal{C} and multiplying by an elementary vector

\mathbf{e} :

$$\mathbf{v} = \lim_{P \rightarrow \infty} (\mathcal{C}^P)^\top \mathbf{e}. \quad (5.2)$$

In practice, it is sufficient to iterate powers of P such that convergence is achieved; computational efficiency can be obtained by manually selecting an arbitrarily large value for P .

The use of the Perron-Frobenius eigenvector as a ranking tool is most often associated with the PageRank algorithm [77], though its use in this application goes back even further [78, 79]. Recent theoretical work has shown that under mild assumptions about the underlying properties of the objects being ranked, the spectral method of eigenvector ranking is equally optimal as maximum likelihood estimation approaches [80]. Here, we interpret the probabilities of the eigenvector as a scheme for weighting the votes of each labeler. For each instance, we take the dot product between the binary vector of labeler opinions on class membership and the relative reliability vector to produce a scalar value $w \in [0, 1]$ representing the weighted ensemble opinion on the class membership of the example. Optionally, we can choose to treat the eigenvector as a strict ranking, and retain only the top- n labelers. We summarize OpinionRank in Algorithm 1; a visual diagram outlining the algorithm's flow is shown in Figure 5.

Algorithm 1 OpinionRank: A Model-Free, Graph-Based Spectral Method for Extracting Labels from Multiple Unreliable Labelers

Input: \mathcal{Y} , a set of K -class membership opinions on n total examples from J labelers.

Input: P , the number of matrix power iterations

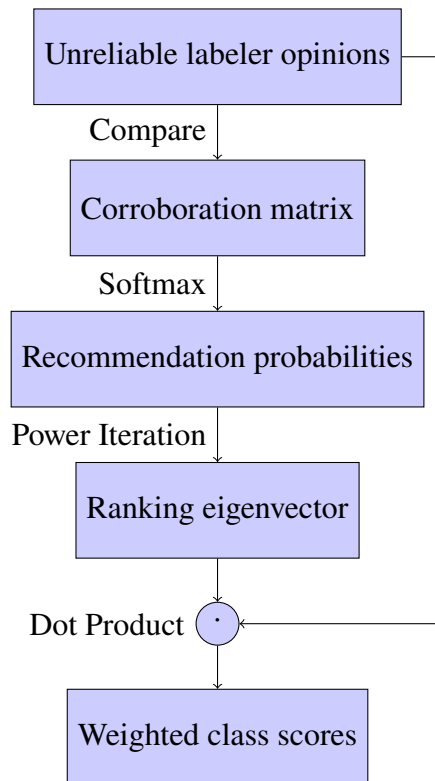
Input: $n \in [1, J]$, the number of labelers to retain from the dominant eigenvector

Output: \mathcal{W} , a $K \times N$ matrix of weighted class membership scores.

- 1: Initialize $K \times N$ matrix of class scores, \mathcal{W}
 - 2: **for** each class $\ell = 1$ to K **do**
 - 3: Obtain $J \times N$ matrix of class- ℓ membership opinions,
 $\mathcal{K} \leftarrow \text{checkEqual}(\mathcal{Y}, \ell)$
 - 4: $\mathcal{C} \leftarrow$ Initialize $J \times J$ corroboration matrix
 - 5: **for** each labeler $i = 1$ to J **do**
 - 6: **for** each labeler $j = 1$ to J **do**
 - 7: $\mathcal{C}_{ij} \leftarrow \sum \text{checkEqual}(\mathcal{K}_i, \mathcal{K}_j)$
 - 8: **end for**
 - 9: $\mathcal{C}_i \leftarrow \text{softmax}(\mathcal{C}_i / N)$
 - 10: **end for**
 - 11: Obtain dominant eigenvector, $\mathbf{v} \leftarrow (\mathcal{C}^P)^\top \mathbf{e}$
 - 12: Obtain top- N eigenvector indices, $\mathbf{I} \leftarrow \text{argsort}(\mathbf{v}, n)$
 - 13: **for** all $k \in \mathcal{K}$ **do**
 - 14: $k \leftarrow 0.5$ if k is missing
 - 15: **end for**
 - 16: $\mathcal{W}_\ell \leftarrow \mathcal{K}_I^\top \mathbf{v}_I$
 - 17: **end for**
 - 18: **return** \mathcal{W}
-

Figure 5

Block Diagram of OpinionRank



The OpinionRank algorithm is highly flexible, and is easily adaptable to any labeling paradigm. In the case of binary categorical labeling problems, OpinionRank can be applied directly. For multinomial and multilabel problems, the labeler-provided class labels can be transformed into binary encodings (one-hot labels for the multinomial case), with OpinionRank being applied across each class. In these scenarios, OpinionRank estimates the class-conditional reliability ranking of each labeler, on the observation that some labelers may have more or less expertise with respect to some classes compared to others.

For binary problems, label predictions are obtained by thresholding the weighted

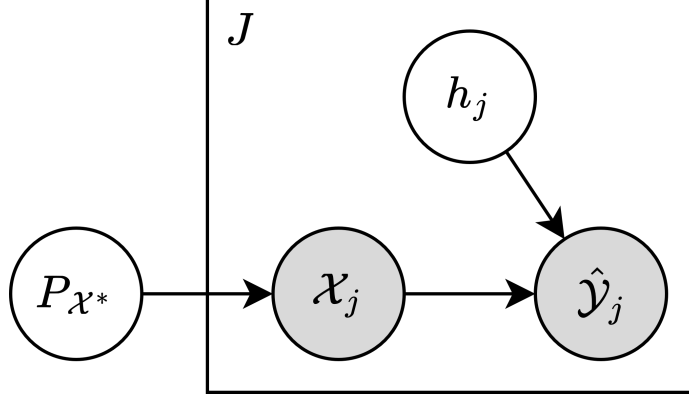
class scores at 0.5. In the multilabel case, the same rule can be applied to each class independently to obtain binary label vectors for each instance. Multinomial decisions are made by choosing the class corresponding to the argmax of the class membership scores.

Chapter 6

Labeler-Aware Learning

Although OpinionRank has proven to be a powerful tool for extracting ground truth from redundant labels (obtained via crowdsourcing, for example), the algorithm’s weakness is its requirement that it cannot handle the scenario where the labels provided by the multiple labelers are non-overlapping. This scenario arises naturally whenever a dataset is collected in a streaming scenario, such as when users contribute to a collective database. Here, the corroboration matrix has no values, since there do not exist any instance-wise classification agreements (as each instance has only one label). Furthermore, we cannot simply combine each labeler’s contributions into a single dataset, as doing so would destroy the information contained in the labeler associations.

To address this failure mode, we propose a multi-stage, labeler-aware framework for robust learning from noisy labels called “labeler-aware learning” (LAL). Generally, previous formulations for learning from labeled data have assumed that the training dataset takes the form of a single homogeneous bucket of data, $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$. However, our labeler-dependent noise model introduces the concept of labeler awareness, whereby the learner retains information regarding which training dataset contributions were provided by which labeler. Thus, we propose “labeler-aware learning” by considering the set of partitioned subsets of noisily-labeled data provided by J heterogeneous labelers, $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_J\}$, where each $\mathcal{D}_j = \{\mathcal{X}_j, \hat{\mathcal{Y}}_j\}$ and $x_\ell \notin \mathcal{X}_j$ for any $x_\ell \in \mathcal{X}_{\ell \neq j}$. Our model is illustrated in Figure 6; each training data subset $\mathcal{X}_j \sim P_{\mathcal{X}^*}$ is labeled by the j^{th} labeler (whose labeler-dependent dynamics are described by h_j), who provides the observed labels $\hat{\mathcal{Y}}_j$.

Figure 6*Graphical Model of Labeler-Aware Learning*

We would like to leverage our labeler awareness to refine our observed labels. However, because each labeler j provides labels only on \mathcal{X}_j , and each \mathcal{X}_j is disjoint from every other $\mathcal{X}_{\ell \neq j}$, we cannot directly use methods for learning from crowds (which rely on multiple label redundancy) to exploit our labeler-aware knowledge. To overcome this limitation, we draw inspiration from modern social learning theory [81, 82], and impute the labels that *would* have been provided by labeler j on the subset $\mathcal{X}_{\ell \neq j}$ by training a representative model, θ_j . To accomplish this task, we first observe that while the labels $\hat{\mathcal{Y}}_{\ell \neq j}$ from all labelers $\ell \neq j$ are uninformative for estimating h_j , the data $\mathcal{X}_{\ell \neq j}$ can very well be informative, and can be exploited under the reasonable assumption that each \mathcal{X}_j is drawn from the same underlying distribution $P_{\mathcal{X}^*}$ ¹. Hence, we leverage semi-supervised learning (SSL) to train each θ_j by treating the data-label pairs $\{\mathcal{X}_j, \hat{\mathcal{Y}}_j\}$ as labeled data, and the union $\mathcal{X}_j^U = \{\cup_{\ell} \mathcal{X}_{\ell \neq j}\}$ as unlabeled data. We place no restrictions on which SSL methods

¹Note that under the labeler-agnostic paradigm, this assumption is taken implicit, as the entire training dataset is considered to be drawn from the same distribution. Here, we are merely partitioning this training dataset, with no other modifications, so the same assumption holds.

may be used, maintaining modularity in anticipation of future advances in semi-supervised learning. It is also not essential for any θ_j to achieve high accuracy on test data; in this stage, we are interested only in estimating, as closely as possible, the parameters h_j that describe the label generation dynamics of each labeler j , regardless of generalization performance. In fact, if labeler j turns out to be a low-quality labeler, then it is expected that θ_j would be similarly low-quality, with poor generalization capabilities.

Once θ_j is obtained, we then simply query θ_j on the unlabeled data to obtain synthetic labels as $f_{\theta_j}(\mathcal{X}_j^U)$. Combining these synthetic labels (queried from θ_j) with the genuine, labeler-provided labels $\hat{\mathcal{Y}}_j$ (generated by h_j), we now have a full set of labels for each $x_i \in \mathcal{X}$, as would have been provided by labeler j . Repeating this process J times (for each of J labelers), we obtain a full set of J redundant, labeler-dependent labels for the entire dataset. Hence, we can use learning from crowds (LFC) to integrate the redundant noisy labels into a single filtered label per instance.

As before, we place no restrictions on which methods for learning from crowds may be utilized, allowing our method to stand as a modular, model-agnostic learning framework. However, we note that many methods for learning from crowds are unsuitable for our framework: while we have remained intentionally agnostic regarding the parameters of h_j and θ_j (in order to maintain generality), a considerable drawback of commonly-used approaches such as expectation-maximization [39, 83] or Bayesian inference [40] is their dependence upon the correct modeling of the precise parameters being estimated. For this reason, we suggest using nonparametric approaches, such as weighted majority voting [16, 84] or OpinionRank (Chapter 5). The complete algorithmic framework for labeler-aware learning is shown in Algorithm 2.

Algorithm 2 Modular Framework for Labeler-Aware Learning

Input: $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_J\}$, the set of data-label pairs provided by each labeler $j \in [1, \dots, J]$

Input: SSL, a semi-supervised classification algorithm

1: **Training stage:**

2: $\Theta \leftarrow$ Initialize an empty set of trained models

3: **for** each labeler $j = 1$ to J **do**

4: $\mathcal{X}_j^L, \mathcal{Y}_j^L \leftarrow \mathcal{D}_j$ (Gather labeled data from labeler j)

5: $\mathcal{X}_j^U \leftarrow \{\cup_{\ell \neq j} \mathcal{X}_\ell\}$ (Gather unlabeled data from all other labelers $\ell \neq j$)

6: $\theta_j \leftarrow \text{SSL}(\mathcal{X}_j^L, \mathcal{Y}_j^L, \mathcal{X}_j^U)$

7: $\Theta \leftarrow \{\Theta \cup \theta_j\}$

8: **end for**

Output: Θ , a predictive ensemble of SSL models

Input: $\mathcal{X} \subseteq \mathcal{X}^*$, the data seen during inference

Input: LFC, an algorithm for learning from crowds

9: **Inference stage:**

10: $\mathcal{Y} \leftarrow$ Initialize an empty set of predictions for \mathcal{X}

11: **for** each $x \in \mathcal{X}$ **do**

12: $w \leftarrow$ Initialize an empty set of intermediate predictions for x

13: **for** each model $\theta_j \in \Theta$ **do**

14: $w \leftarrow \{w \cup \mathcal{F}_{\theta_j}(x)\}$

15: **end for**

16: $y \leftarrow \text{LFC}(w)$ (Integrate intermediate predictions into ensemble prediction for x)

17: $\mathcal{Y} \leftarrow \{\mathcal{Y} \cup y\}$

18: **end for**

Output: \mathcal{Y} , the set of classification predictions for \mathcal{X}

Chapter 7

Adversarial Labelers

While the discussion of noisy labels presented in Chapter 4 considers merely incidental label noise (as presented by good-faith, though perhaps inept, labelers), the noisy labeler paradigm can be extended to its logical extreme by considering adversarial labelers as bad-faith actors. In this chapter, we address adversarial label noise by considering three types of adversarial attacks against multiple-labeler data pipelines: data flooding, multiple adversaries, and backdoor attacks.

7.1 Adversarial Labels from Bad-Faith Labelers

The good-faith labeler model can be extended to describe *adversarial* labelers. We posit that an intelligent adversary will present the learner with a false label based on not only its own best guess as to the correct label, but also its *second* best guess. This behavior captures the idea that the second-most-likely class is that which is most likely to be confused for the correct class, and so therefore presenting this second-most-likely class as correct will cause maximum confusion during the learning process. We define the function $\arg_2\max$ as identical to the $\arg\max$ function, except that $\arg_2\max(A)$ returns the index of the element of A with the second highest value instead of the maximum (with ties broken arbitrarily). An adversarial labeler therefore provides noisy labels following

$$\hat{y}_{ij} = \arg_2\max_k h_j(x_i). \quad (7.1)$$

The labels that an adversary presents are representative of the adversary's best guess about the false labels that are most likely to be confused with the correct labels. This approach is similar to that in [30], where the label of the second most-confident category is

integrated into the noise model. Note that the adversary may itself have an incorrect belief about the true class, and in its attempt to provide a false label may accidentally provide a correct label. We mark the similarities between this approach and that of Goldberger (Section 3.4.3), where there exists a layer of obfuscation between the observed labels presented by a labeler and that labeler’s actual belief about the true class. We also note that the $\arg_2\max$ function can be considered as a special case of the beta-binomial model described in Equation 4.2 where the entire probability mass is centered on the second-most-likely class ordinal.

Following the labeler-dependent noise paradigm, we assume that the label gathering process draws from a pool of labelers containing a mixture of both good-faith (but imperfect) labelers and malicious, adversarial labelers. Accordingly, the characteristics of both types of label noise—unintentional and/or adversarial—will vary from labeler to labeler, and even when adversarial noise is absent there may be variations in the levels and characteristics of the natural error associated with each labeler (Chapter 4).

7.2 Multiple-Labeler Adversarial Attack Vectors

We define two general vectors of adversarial poisoning attacks that should form the basis of future work on adversary-aware learning from noisy labels:

- A *data flooding attack* occurs when a single adversary provides an overwhelming quantity of adversarial labels relative to the quantity of labels provided by the good-faith labelers.
- A *multiple adversaries attack* occurs when multiple bad-faith labelers invade the labeling process; unlike the data flooding attack, each adversarial labeler need not

provide large quantities of labels.

Both attacks exploit vulnerabilities in distributed label collection, especially in crowdsourcing and online learning scenarios, and have the capability to introduce large amounts of adversarial noise into the training data.

There is real-world precedent for the types of attacks we describe: Microsoft’s infamous Twitter chatbot Tay [85], its successor Zo [86], and even commonplace home assistant tools such as Google Home and Amazon Echo [87] have all been subject to malicious data flooding attacks from multiple adversaries. By codifying these threats and proposing methods for addressing them, this work serves as a preliminary effort in proactively establishing robust defensive measures against adversarial label attacks.

7.3 Adversarial Backdoor Attacks

In addition to the general attack vectors described above, we also consider the potential for labeler-aware learning to provide robust defenses against adversarial backdoor attacks (Section 2.4.1). While backdoor attacks deliberately avoid injecting large amounts of data into the training dataset, their precision and power despite their stealthiness makes them a high-priority target for defensive research.

7.3.1 Rethinking Backdoor Threat Modeling Under Labeler Awareness

The most common backdoor threat model assumes that the attacker is in control of the training process, with the victim having either outsourced the training of their network to a malicious actor, or allowed an adversary privileged access to their training pipeline [43, 88, 89]. These models allow the attacker to precisely craft backdoor patterns that

will have maximal impact against specific network architectures. However, in practice it is unreasonable to assume that the attacker has such privileged access, and any real-world scenario where such access is available would be a catastrophic failure of operational security. Furthermore, the setting of an outsourced training provider injecting backdoor triggers into models is unrealistic, both because such outsourcing is rare in practice, and because the service providers that do exist would suffer irreparable harm to their reputations if they were to perform such an attack.

Other threat models weaken the attacker by removing the attacker’s knowledge of the training details of the target network. Under these models, the typical approach is to craft full-image masks using adversarial perturbations [90, 91]. However, the practical use cases of such attacks are limited, as the sensors used in systems where backdoor attacks would be most impactful are not capable of adding such specific, precise perturbations to their entire inputs, making it infeasible to trigger the backdoor during inference.

While previous adversarial settings consider a worst-case scenario, we contend that such settings are ill-posed, as they depend upon unrealistic assumptions about the capabilities of an attacker in both the training and inference stages. Furthermore, previous approaches have assumed a helpless defender that has no control over their own data collection process, which is also an unrealistic constraint in the context of modern data collection.

7.3.2 A More Realistic, Practical Threat Model

We pose the setting where the defender wishes to gather a large amount of data from multiple users into a single database. Here, an adversary may realistically gain ingress to

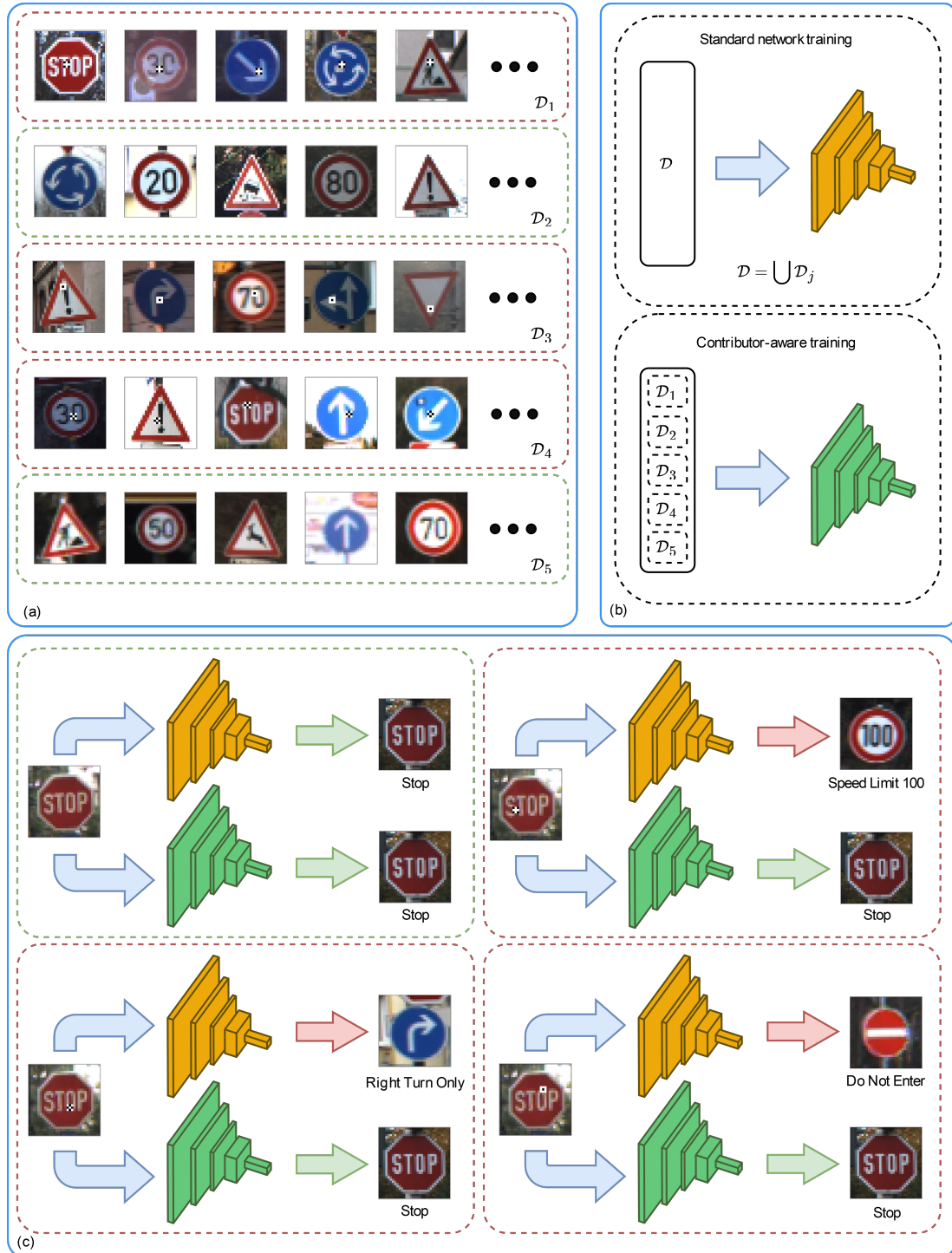
the training data simply via making a contribution to the common database. We note that the presence of an adversary *implies* the existence of multiple users, as any adversary must be contrasted with a non-adversarial data source. As the database comprises contributions from multiple (potentially adversarial) labelers, we allow the defender to have access to (possibly anonymous) metadata associating each data instance to its labeler. For example, any contribution to the training database may be associated with a tag corresponding to the user who made the contribution, allowing labeler-wise data grouping. Such tags are commonplace and standard practice for image databases, and are trivially implemented in ways that preserve user privacy and anonymity. We claim that by allowing this labeler awareness to the defender, the defender is able to train an adversarially-robust classifier *without* needing to explicitly detect any backdoor patterns themselves, or even to remove any malicious samples. This pattern-agnostic approach constitutes a broad defensive strategy against any kind of adversarial backdoor trigger.

Under this more practical and realistic threat model, we consider a set of disjoint data subsets $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_J\}$, with each \mathcal{D}_j provided by one of J labelers. Each subset comprises $\mathcal{D}_j = \{\mathcal{X}_j \subseteq \mathcal{X}^*, \mathcal{Y}_j \subseteq \mathcal{Y}^*\}$, such that $\{x_\ell \in \mathcal{X}_\ell\} \not\subseteq \mathcal{X}_j$ for any $\ell \neq j$. Without loss of generality, we assume that unknown to the defender, any dataset $\mathcal{D}_k = \{\mathcal{X}'_k, \mathcal{Y}'_k\}$ may be provided by an adversary, following Equation 2.5. Under the conventional, labeler-agnostic learning framework, these disjoint subsets would be concatenated into a single dataset $\mathcal{D} = \bigcup_j \mathcal{D}_j$, so the adversary’s poisoned data and labels would be mixed into the common dataset, with the labeler identification information being lost. Standard labeler-agnostic training therefore allows the backdoor patterns to be learned by the classifier, and the attack is easily triggered after model deployment.

The vulnerability of such conventional training is shown in Figure 7. First, the training dataset \mathcal{D} is collected from J labelers to a common database (a). Adversaries who inject backdoor patterns may be among the labelers (red), and hide amidst the clean labelers (green). Standard training protocol (b, top) treats $\mathcal{D} = \cup \mathcal{D}_j$ as a single dataset, and trains a network on the entire \mathcal{D} . In contrast, labeler-aware training (b, bottom) exploits information about the source of each subset of the training data to train a robust model. During inference (c), clean images are classified correctly by both the standard network and the labeler-aware model (green). However, backdoor patterns on test data trigger force misclassifications from the standard network, while the labeler-aware model is unaffected (red).

Figure 7

Training Pipeline With Backdoor Threats



7.3.3 Labeler-Aware Defense Against Adversarial Backdoor Attacks

Previous methods for defending against backdoor attacks focus primarily on detecting samples containing backdoor trigger patterns and removing these samples from the dataset [92–95]. However, these approaches are reactive, and discard useful information that might be utilized in order to build more effective models (since x' contains useful feature information x). We propose to use labeler-aware learning (LAL) as a *proactive* defensive strategy, which is capable of exploiting the full characteristics of the entire training data, *without* introducing adversarial label associations.

We observe that even under the assumption that some data provided by an adversarial labeler k may be corrupted following Equation 2.5, the semi-supervised strategy utilized by LAL destroys the association between $\eta_k(y')$ and y' , as y' is not present in the labeled dataset for any labeler $j \neq k$. As a result, $\eta_k(y')$ is treated as uninformative noise on x . Therefore, for any model $\theta_{j \neq k}$ the functional response to an adversarial input will be

$$f_{\theta_{j \neq k}}(x') = f_{\theta_{j \neq k}}[x + \eta_k(y')] = f_{\theta_{j \neq k}}(x) = y. \quad (7.2)$$

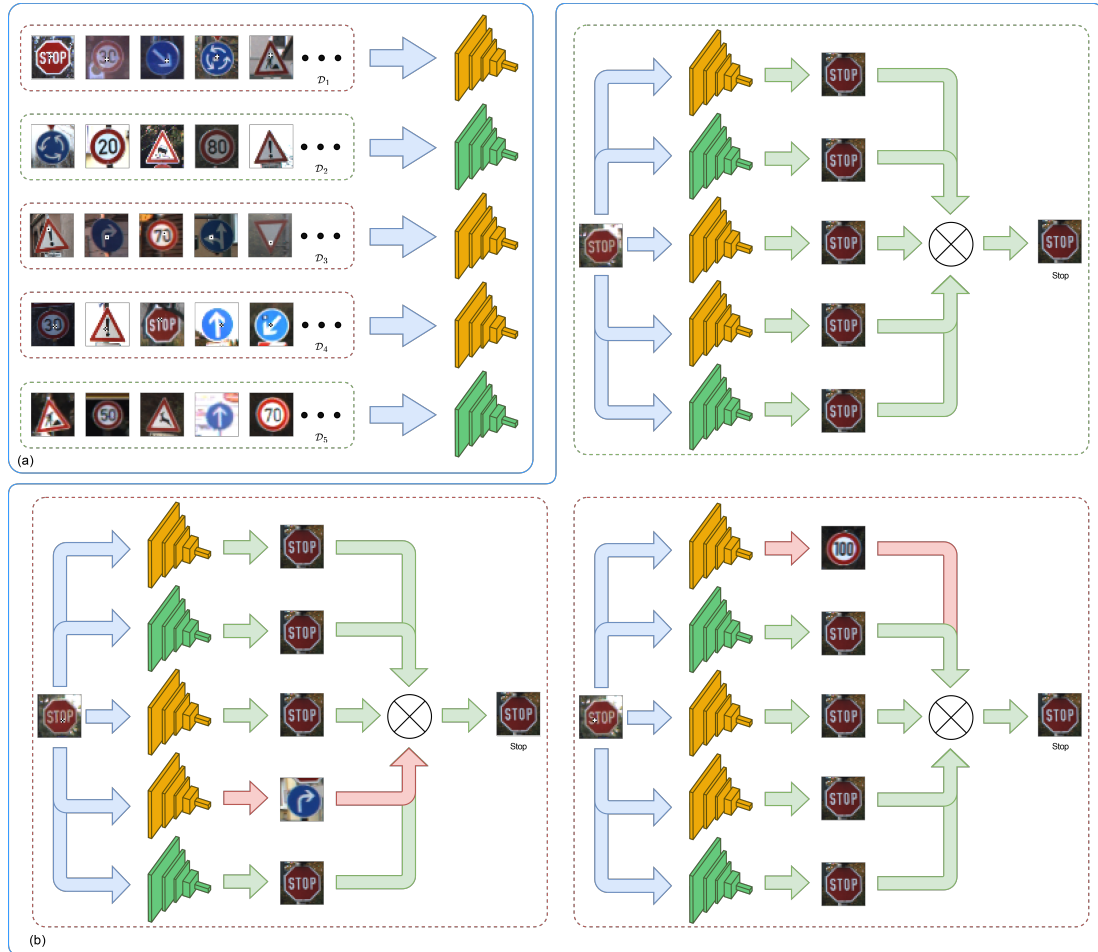
Notably, we still do not know which labelers are adversarial, or which data contain backdoor triggers. Furthermore, while Equation 7.2 shows that $f_{\theta_{j \neq k}}(x') = y$ for any labeler $j \neq k$, we still have that $f_{\theta_k}(x') = y'$, as we have taken no steps to detect or remove adversarial samples. Fortunately, we do not actually *need* to identify adversarial examples: because the intermediate predictions provide label redundancy for any arbitrary example, we are able to leverage OpinionRank (Chapter 5) in order to filter the adversarial false labels. Thus, while the adversary is successful in forcing a single classifier to produce a false prediction, the ensemble prediction is robust against the adversary’s backdoor trigger, as

our proposed approach effectively shields every other member of the ensemble from the false label associations of the adversarial inputs.

The defensive capabilities of labeler-aware learning are illustrated in Figure 8. During the training of each representative model, any individual model may be vulnerable to backdoor triggers (a). However, because each trigger affects only a single model in the ensemble, the vulnerable model's forced misclassification is outweighed by the unaffected predictions from the rest of the ensemble (b). Note that in general, even if a model is vulnerable to a particular backdoor trigger, it will not be vulnerable to a *different* backdoor trigger, so attacks from multiple different adversaries will be defeated.

Figure 8

Labeler-Aware Training Produces Robust Ensembles



Labeler-aware training of predictive ensembles therefore produces a classification model that is secure against backdoor threats, *without* requiring, or even attempting, to perform identification of either the adversaries or the poisoned samples. A further advantage of the proposed strategy is that whereas other defensive approaches remove the poisoned samples from the training dataset upon detection, our method does not discard any training data. This is desirable, because each x' contains salient feature information x that can be

used to further train a classifier. Depending on the fraction of data that is poisoned, as well as the success rate of backdoor identification, detect-and-remove strategies may ultimately throw out substantial amounts of the training data. In contrast, our approach retains and utilizes the entire training data to train a robust classifier, even in the (unknown) presence of large fractions of adversarial samples.

Chapter 8

Experiments

In this chapter, we present our experimental findings, analyze the results, and provide a discussion of the comparative benefits of the proposed methods.

8.1 Metrics

The baseline metric of interest is classification accuracy on a held-out testing dataset. At no point during the model selection or training process is any of the testing data visible to any of the models under test. Accuracy is measured as the percentage of correctly-predicted labels compared to the total number of labeled data.

However, while raw accuracy is used as the baseline metric, for the purposes of learning from noisy labels we are *primarily* interested in measuring the *robustness* of the models under test against increasing amounts of label noise. We observe that the term “robustness” is overloaded in machine learning literature; here, we define “robustness” to refer to an algorithm’s capacity to maintain its accuracy performance as the fraction of training data with noisy labels increases. For example, while the most favorable learning environment will include no label noise (i.e. 100% clean labels), a robust algorithm will exhibit little to no reduction in performance as more noise is introduced. A weak algorithm, however, will be heavily compromised by increasingly noisy labels, and its performance will be negatively impacted.

Therefore, while most of the algorithms studied in this section boast the ability to achieve reasonably strong accuracy in the absence of noisy labels (or under only small amounts of label noise), the robustness of these algorithms must be evaluated by interpreting

the accuracy *trends* in response to the presence of varying amounts of noisy labels.

8.2 Evaluating OpinionRank for Learning from Crowds

In order to objectively evaluate and compare our proposed algorithm, we reproduce, as faithfully as possible, experiments from the three settings for learning from crowds discussed in Chapter 3. We test the OpinionRank algorithm (Chapter 5) under the hand-crafted conditions for the models of the original authors. We also perform a wall clock runtime analysis of OpinionRank to demonstrate the algorithm’s speed and computational efficiency.

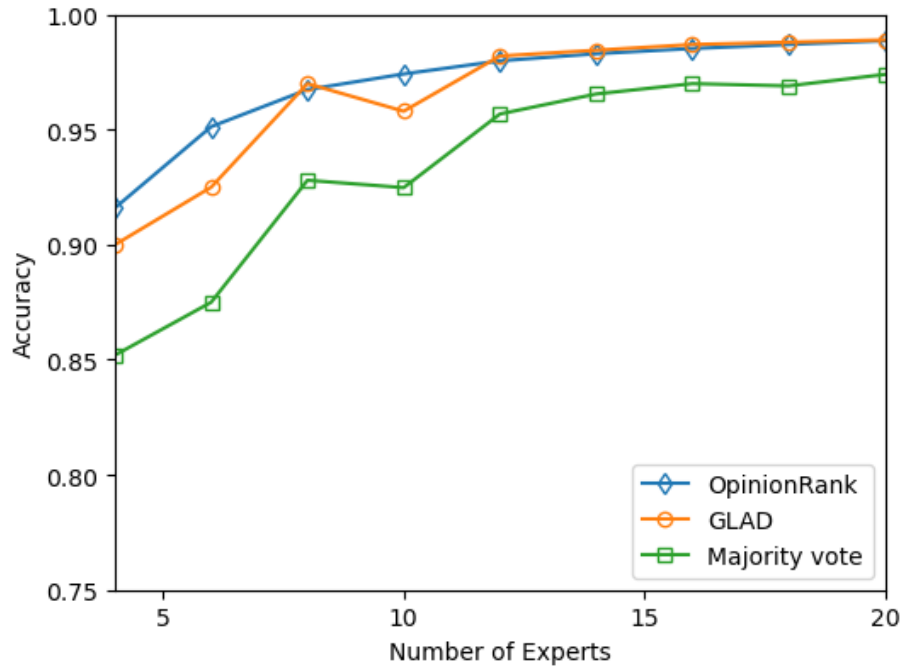
8.2.1 Generative Model of Labels, Abilities, and Difficulties

We implemented three experiments under the same conditions described in Whitehill et al. [39]. These experiments evaluate the OpinionRank algorithm’s performance under the conditions of the authors’ labeling model (as described in Section 3.4.1), its ability to handle “difficult” images, and its stability under varying starting conditions.

8.2.1.a Labeling Model. The first experiment simulates the labeler accuracy as $\alpha_j \sim \mathcal{N}(1, 1)$, and the inverse-difficulty of labeling a data instance as $\beta_i \sim \text{Lognormal}(1, 1)$. The observed label of an instance i provided by labeler j is sampled according to Equation 3.1. The algorithms are evaluated by the proportion of accurate class labels, with the amount of total data set to $N = 200$. Whitehill reported the average of 40 experiments; we report the mean of 50,000 experiments (Figure 9). Baseline results are reproduced from Figure 2 in [39].

Figure 9

Test Accuracy on Whitehill’s Labeling Model



Both OpinionRank and GLAD considerably outperform majority voting, and converge to greater than 99% accuracy as the number of labelers increases. Because the parameters of the experiment ensure that the average reliability of the pool of labelers is greater than 0.5, these results are expected (from the generalized Condorcet jury theorem). Notably, OpinionRank outperforms GLAD at lower numbers of labelers, suggesting that eigenvector-based reliability ranking is robust even for small pools of labelers.

8.2.1.b Modeling Image Difficulty. The second experiment considers a pool of 50 labelers, each labeling the same set of $N = 1000$ instances, with half of the instances considered “easy” and the other half considered “hard”. The labelers labeled the “easy”

images correctly with 100% accuracy. The labelers labeled the “hard” images correctly according to whether they were “good” ($p_{correct} = .95$) or “bad” ($p_{correct} = .54$). The ratio of “good” to “bad” labelers is 25:1. The score is measured as the proportion of correctly estimated labels, reported as the error rate, $error = 1 - accuracy$. Whitehill reported the average of 20 experiments; we report the mean of 50,000 experiments (Table 4).

Table 4

Mean Error Rate When Modeling Image Difficulty

Method	Error
Majority vote	11.2%
Dawid & Skene	8.4%
GLAD	4.5%
OpinionRank	0.0%

In the image difficulty modeling experiment, OpinionRank is able to recover the correct label in 100% of cases. This is due to the parameters of the experiment. The “easy” images, being labeled with 100% reliability, are heuristically irrelevant to the performance of the OpinionRank algorithm, as all voters will provide the same (correct) label. Therefore, regardless of the relative reliability eigenvector, the weighted sum will always be the correct label. The “hard” images, on the other hand, are also simple for OpinionRank, due to the labeling schema. With such a large majority of the labelers being “good”, OpinionRank builds very strong recommendation relations between the “good” labelers, so the “bad” labelers are overruled when they are wrong (the 5% of the time that the “good” labelers are

wrong is also easily ignored).

Whitehill’s experiment was heavily biased toward “good” labelers (a reasonable scenario in the context of human labelers). We extended the experiment to smaller proportions of “good” to “bad” labelers; with 25, 40, and 50 (out of 50) labelers being “bad”, we found error rates of 0%, 1%, and 14%, respectively. These results suggest that OpinionRank is robust even against labeler pools with high densities of unreliable labelers.

8.2.1.c Stability Under Various Starting Points. The third experiment simulates the labeling of $N = 2000$ instances by 20 labelers following Equation 3.1, with $\alpha_i \sim U[0, 4]$ and $\log(\beta_j) \sim U[0, 3]$. Under the assumptions of the authors’ generative model, these parameters represent a large variance in the difficulty-expertise spectrum, and so the experiment tests algorithmic stability across a broad range of starting conditions. The scores are reported as the mean and standard deviation of label accuracy scores. Whitehill reported the mean and standard deviation over 50 experiments; we report the mean and standard deviation over 50,000 experiments.

Similarly to the second experiment, OpinionRank achieves a perfect score on the authors’ stability test (compared to mean of 85.84% and standard deviation of 0.024% for GLAD). Because the test draws the labeler expertise from $\alpha \sim U[0, 4]$, all labelers have expertise greater than random guessing. With a pool of 20 labelers, OpinionRank is able to consistently discover the *best* labelers, even within this pool of above-average labelers, and extract the correct labels. Notably, it achieves this performance without the computationally costly need to estimate the precise parameters of each labeler. OpinionRank demonstrates that only the *relative* expertise is needed, as long as the Condorcet criterion is obeyed and

the average expertise is greater than random chance [63].

8.2.2 *The Multidimensional Wisdom of Crowds*

We also implemented two experiments from Welinder et al. [40]. The first experiment evaluates the OpinionRank algorithm on the authors’ proposed label generation model; the second experiment evaluates OpinionRank on a real-world dataset of human annotations.

8.2.2.a Multidimensional Label Generation Model. We reproduce the conditions of Welinder’s modeling experiment by generating data following the assumptions of the model (as described in Section 3.4.2). Following the experimental setup in [40], we:

- set the number of data instances as $N = 500$.
- assign $w_j = 1$ with probability .99 and $w_j = -1$ with probability .01, to simulate adversarial labelers.
- draw $\tau_j \sim \mathcal{N}(0, \sigma = 0.5)$.
- draw the noise parameter $\sigma_j \sim \text{Gamma}(1.5, 0.3)$.
- set the generative parameter $\sigma_z = 0.5$ ¹.

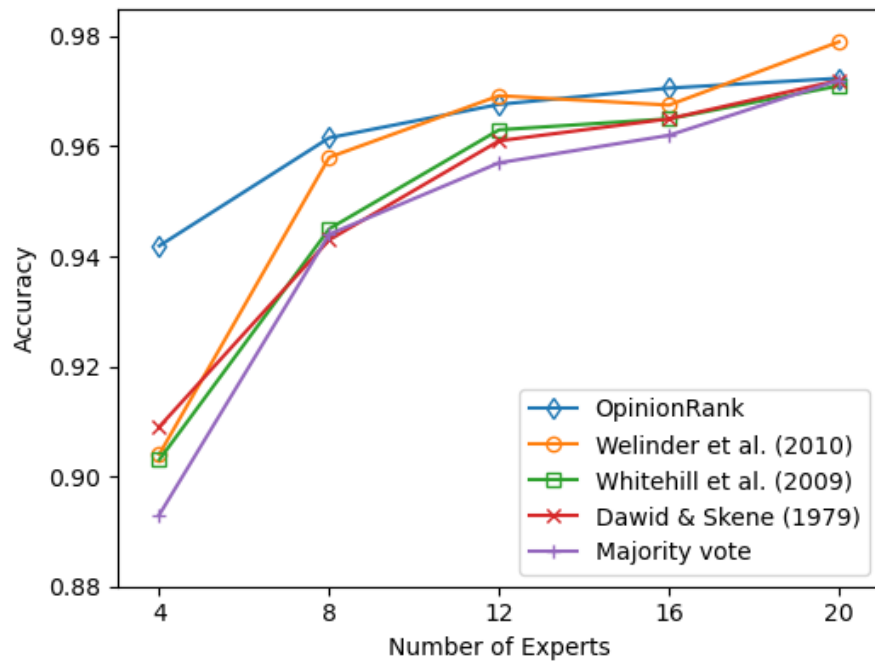
Welinder reported the average over 40 experiments; we report the mean of 50,000 experiments (Figure 10). Baseline results are reproduced from Figure 3(c) in [40] The annotation model of Welinder et al. is considerably more complex than that of Whitehill

¹Welinder, P., “Caltech UCSD Binary Annotation Model,” Github, 2012. Available at <https://github.com/welinder/cubam>.

et al. Despite this complexity, OpinionRank achieves accuracy above 94% across all experiments. We note that while all of the algorithms being compared eventually converge to accuracies greater than 96%, OpinionRank strongly outperforms the other algorithms at lower numbers of labelers.

Figure 10

Test Accuracy on Welinder’s Label Generation Model



8.2.2.b Waterbirds Dataset. We evaluate OpinionRank on the real-world Waterbirds dataset constructed by Welinder et al. Using Amazon Mechanical Turk, the authors asked 53 human labelers to provide labels on a set of 240 images. The images consisted of

50 photographs each of Mallards, American Black Ducks, Canadian Geese, and Red-necked Grebes, as well as 40 additional images featuring no birds. The labelers provided binary labels according to whether, in their opinion, each image contained a picture of a duck (only Mallards and American Black Ducks are positive classes). Of the 53 labelers, only 25 provided labels for all images; the other 28 labelers omitted between 40 and 200 labels.

On this real-world dataset, OpinionRank predicts the correct label for 86.7% of the images (Table 5). OpinionRank outpaces majority vote at 68.3% accuracy, GLAD at 60.4% accuracy, and the authors' own Bayesian generative model at 75.4% accuracy [40].

Table 5

Percent Accuracy on the Waterbirds Dataset

Method	Percent Correct
Majority voting	68.3%
GLAD	60.4%
Welinder	75.4%
OpinionRank	86.7%

8.2.3 Combining Soft Decisions of Several Unreliable Labelers

We reproduced the soft-decision modeling experiment under the same conditions described in Section 3.4.3. Following [71], $N = 200$ instances are generated and assigned random labels drawn from a pool of three classes. For each labeler j , its reliability is sampled uniformly on the interval $[0.4, 0.7]$. After each labeler’s opinion q_{ij} is modeled (following Equation 3.4), their opinions are obfuscated by first sampling a multinomial distribution U_{ij} from the flat Dirichlet distribution, before transforming q_{ij} into U_{ij} following

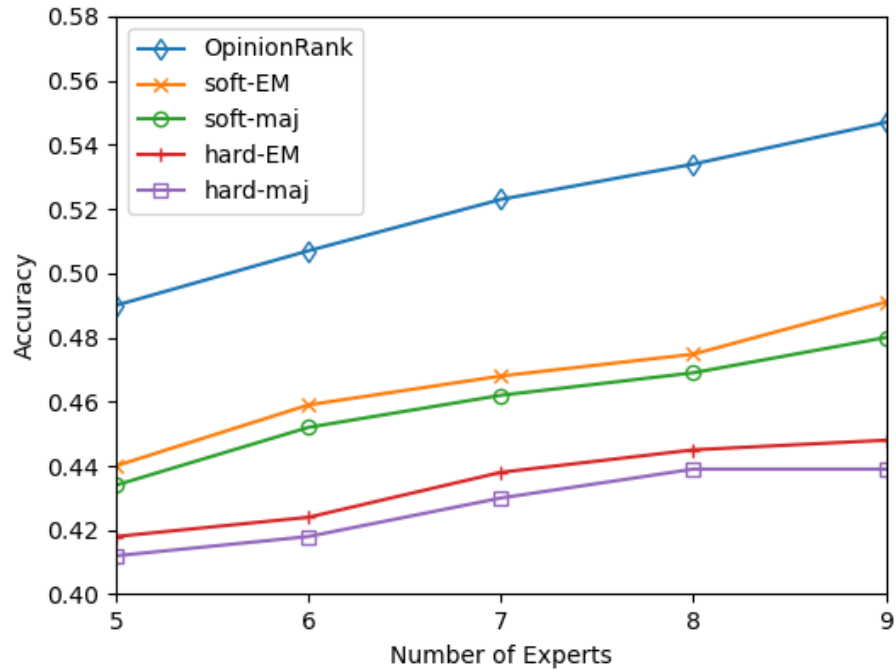
$$\hat{y}_{ij} = \arg \max_{a \in A} U_{ij}[(q_{ij} + z - a) \bmod |A|], \forall a \in A, \quad (8.1)$$

where $z \in A$ is randomly sampled from U_{ij} . Because OpinionRank requires “hard” labels, we utilized Goldberger’s hard-decision process, which takes the argmax of the soft label information over the set of classes, before providing the labels to the algorithm.

Goldberger reports the mean of 100 experiments; we report the mean of 50,000 experiments. As seen in Figure 11, OpinionRank outperforms Goldberger’s extended EM algorithm by a considerable margin, achieving at least 49% accuracy (with only 5 labelers), climbing monotonically up to 55% accuracy (with 9 labelers). soft-EM, soft-maj, hard-EM, and hard-maj results reproduced from Figure 1 in [71].

Figure 11

Test Accuracy on Goldberger’s Three-Class Soft Opinions Model

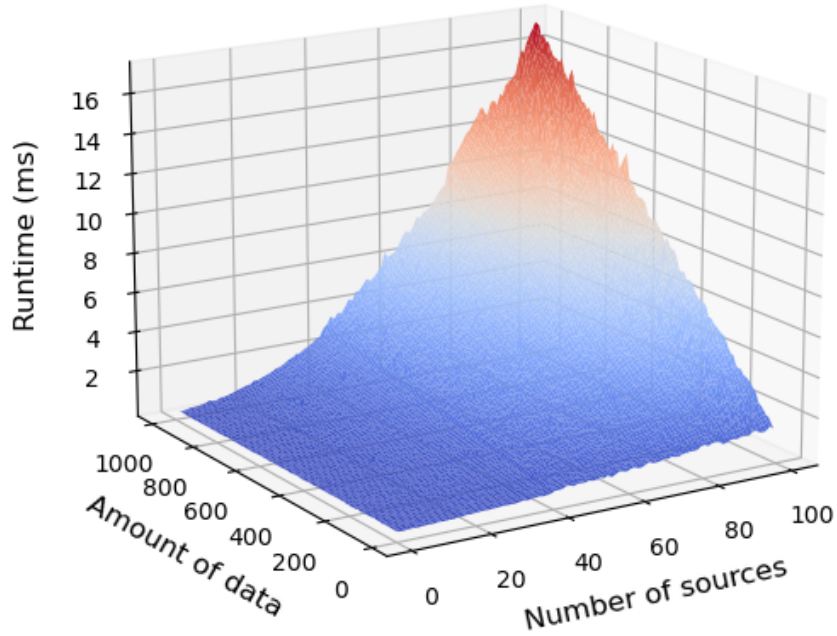


8.2.4 Empirical Runtime Analysis

We have also performed an empirical study of the wall clock runtime of the OpinionRank algorithm. We parameterized the experiment over J , the number of (unreliable) labelers, and N , the total number of data instances. We vary J between 1 and 100 labelers, and N between 10 and 1000 instances. All experiments were performed on a consumer-grade AMD Ryzen 3900X 3.8 GHz 12-core processor with 32 GB of memory. Each experiment occurred on a single processing thread.

Figure 12

Wall Clock Runtime Analysis of the OpinionRank Algorithm



We generated an arbitrary $J \times N$ binary array of randomly generated class membership opinions. This array is passed to the OpinionRank algorithm, and we measure the time required for the algorithm to return its array of weighted class membership scores. We repeated this procedure 100 times for each set of parameters, whose average runtimes are depicted in Figure 12. We observe that the runtime of OpinionRank scales linearly with the amount of data, and quadratically with the number of labelers. We note that the worst-case runtime, with $J = 100$ and $N = 1000$, is only 16.684 milliseconds. Scaling the amount of data up to 1 million instances only increased the average runtime to 17.712 seconds.

8.3 Testing Labeler-Aware Learning Under the Full LDN Model

We have shown that OpinionRank is a powerful, model-free algorithm for learning from crowdsourced data, which is a special case of the general labeler-dependent noise (LDN) model. However, as discussed in Chapter 6, OpinionRank has a limitation in that it requires multiple redundant labels per instance, which may not generally be guaranteed. Hence, we developed labeler-aware learning (LAL) in order to comprehensively handle the general LDN model. Our experiments seek to demonstrate the impact of considering the full LDN model. In particular, we show that current state-of-the-art methods for learning from noisy labels are unable to learn under the general LDN model, and are therefore insufficient. In contrast, the LAL framework is robust against label noise, even in cases of extreme spammer presence where previous approaches fail.

8.3.1 Experimental Setup

Due to the limitations of the historical development of machine learning dataset compilation, labeler-aware datasets are not readily available [96]. Therefore, to evaluate our labeler-aware noise we simulated the effect of labeler-aware label noise by synthetically corrupting the ground truth labels provided by curated benchmark datasets. In particular, we performed experiments on the MNIST [97], SVHN [98], and CIFAR-10 [99] datasets, each of which are commonly used in this way for studying noisy label learning.

Following Equation 4.1, we first generated instance-dependent class likelihoods by intentionally overfitting a deep neural network, $h(\mathcal{X})$, on the training dataset. Then, for a set of J labelers, we selected J_h hammers and J_s spammers, such that $J_h + J_s = J$, and

the training dataset was partitioned into even subsets between all labelers (regardless of labeler quality). For each type of labeler, we defined a beta-binomial distribution such that the instance-dependent highest-likelihood class, $\max(h(x_i))$, would be selected with 95% or 50% probability for hammers and spammers, respectively, following standard practices for hammer-spammer models in previous literature [100]. Furthermore, our experiments feature our novel integration of instance-dependent noise modeling by selecting incorrect labels not based on transition matrices, but with decreasing probability according to instance-dependent class likelihoods. The experimental beta-binomial parameters for labeler-dependent label selection over instance-dependent class likelihoods used in our experiments are listed in Table 6.

Table 6

Experimental Hammer-Spammer Beta-Binomial Parameters

Type	Hammer	Spammer
α_j	1	1
β_j	200	10
$\Pr(\max(h(x_i)))$	0.95	0.50

Our experiments focused on parameterizing over the ratio of hammers to spammers. Table 7 shows how controlling this ratio can be converted into effective label noise percentages (for the hammer-spammer distribution parameters used in our experiments), which are more common in literature on learning from noisy labels. Label noise percentages are reported as the average label noise present over all experiments performed on a given dataset

with a given hammer-spammer ratio; percentages are not exact due to small inaccuracies in the instance-dependent neural network.

Table 7

Converting Hammer-Spammer Ratios to Experimental Label Noise

$J_h : J_s$	% label noise (MNIST)	% label noise (SVHN)	% label noise (CIFAR-10)
10 : 0	4.5	4.8	5.0
9 : 1	8.8	9.1	9.5
8 : 2	13.6	13.5	13.6
7 : 3	17.3	17.7	18.0
6 : 4	21.5	21.6	22.3
5 : 5	25.9	26.1	26.4
4 : 6	30.2	30.2	30.4
3 : 7	34.7	34.8	34.8
2 : 8	39.1	39.2	39.2
1 : 9	43.3	43.3	43.5
0 : 10	47.6	47.4	47.5

Because our labeler-aware learning framework is modular with respect to the semi-supervised learning and learning from crowds component algorithms, the selection of such components is an important hyperparameter. For our experiments, we selected the FixMatch² algorithm [101] to be used for semi-supervised learning, and we selected Opin-

²FixMatch is a recent, powerful improvement on the MixMatch algorithm selected by DivideMix (Section

ionRank³ (Chapter 5) for learning from crowds. We compared our method against a naïve deep neural network trained with cross entropy loss, as well as DivideMix (Section 3.2.1), progressive label correction (PLC) (Section 3.3.1), and self-evolution average label (SEAL) (Section 3.3.2). All algorithms were trained using the default parameters recommended by the original authors⁴.

Importantly, we are not concerned with the absolute performance of any particular algorithm. Instead, we are interested in the robustness of each algorithm against increasing numbers of spammers, and in turn the amount of labeler-dependent noise present in the data. A weak algorithm will exhibit a steep decrease in performance as the number of spammers increases, whereas a robust algorithm will demonstrate comparatively little degradation. We repeated all experiments five times, with uncontrolled random seeds, and we present the results as an average over all five runs. Standard deviations were very small, and so were omitted from Figure 13 and Figure 14; we report standard deviations in Table 8.

8.3.2 Empirical Results

Table 8 shows the results of our experiments using the MNIST dataset, presented as means and standard deviations over five experimental runs. While all algorithms are capable of learning in the presence of only hammers, the introduction of spammers causes rapid degradation in the performance of the labeler-agnostic methods. Naïve cross-entropy

3.2.1). In practice, any semi-supervised learning algorithm may be used, including future algorithms that improve upon FixMatch.

³Similarly, while we propose and utilize OpinionRank in this dissertation as the most powerful and efficient method for learning from crowds, future algorithms for learning from crowds may be used for LAL if they are found to be better than OpinionRank.

⁴Default parameters and implementations were obtained using the open-source code published by the original authors, cross-referenced with the details present in each algorithm’s respective publication.

fails almost immediately, and the state-of-the-art methods, while somewhat robust to small amounts of spammers, perform increasingly worse as the fraction of spammers increases. Only labeler-aware learning is able to retain robust performance over the entire experimental suite, with only a small decrease in performance even when 100% of the labelers are spammers.

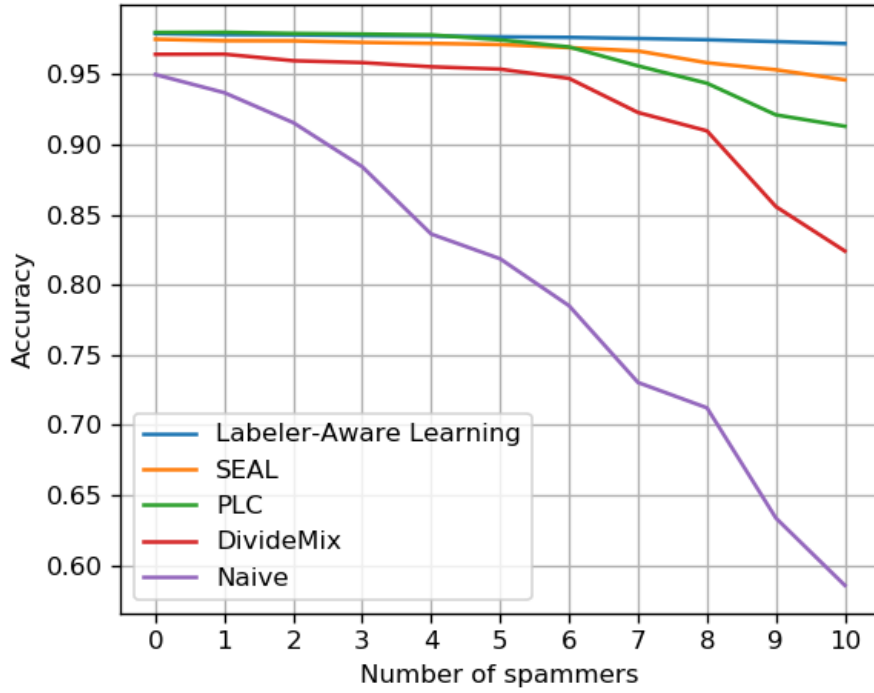
Table 8*MNIST Test Classification Accuracy in the Presence of J_s Spammers*

J_s spammers	Naive	DivideMix	PLC	SEAL	Labeler-Aware Learning
0 spammers	96.6 ± 0.5	99.2 ± 0.3	99.3 ± 0.2	98.6 ± 0.1	99.1 ± 0.1
1 spammers	94.5 ± 0.6	99.0 ± 0.1	99.1 ± 0.1	98.5 ± 0.1	99.2 ± 0.1
2 spammers	92.4 ± 1.2	98.6 ± 0.3	99.0 ± 0.1	98.4 ± 0.2	99.0 ± 0.2
3 spammers	89.8 ± 0.5	98.0 ± 0.2	98.8 ± 0.2	98.2 ± 0.2	99.1 ± 0.1
4 spammers	86.8 ± 1.6	96.8 ± 0.5	98.6 ± 0.1	98.0 ± 0.1	99.1 ± 0.1
5 spammers	81.7 ± 1.3	94.9 ± 0.8	98.2 ± 0.2	98.1 ± 0.2	99.0 ± 0.1
6 spammers	79.5 ± 1.4	93.1 ± 0.7	97.6 ± 0.1	98.0 ± 0.1	98.9 ± 0.1
7 spammers	76.2 ± 2.4	89.8 ± 0.3	96.4 ± 0.2	97.7 ± 0.1	98.5 ± 0.1
8 spammers	71.7 ± 0.9	87.2 ± 0.5	94.2 ± 0.7	97.3 ± 0.3	98.0 ± 0.1
9 spammers	65.6 ± 1.8	82.2 ± 0.8	91.6 ± 0.4	96.3 ± 0.3	97.3 ± 0.3
10 spammers	63.5 ± 1.7	78.9 ± 1.6	87.3 ± 1.1	94.5 ± 0.3	95.5 ± 0.4

The results of our experiments using the SVHN dataset are shown in Figure 13. Similarly to the MNIST results, all methods demonstrate a reasonable capability to correctly learn the problem when exposed to zero, or only small amounts of spammers. However, as the ratio of spammers increases, all competing approaches suffer considerable losses in accuracy compared to labeler-aware learning. Specifically, by 10 spammers SEAL, PLC, DivideMix, and naïve cross entropy training have lost 3%, 7%, 15%, and 35% more accuracy compared to labeler-aware learning, respectively. In contrast, labeler-aware learning is nearly unaffected by spammers, and in fact is able to leverage the heterogeneous characteristics of labeler-dependent noise in order to achieve robust performance even when every labeler is a spammer.

Figure 13

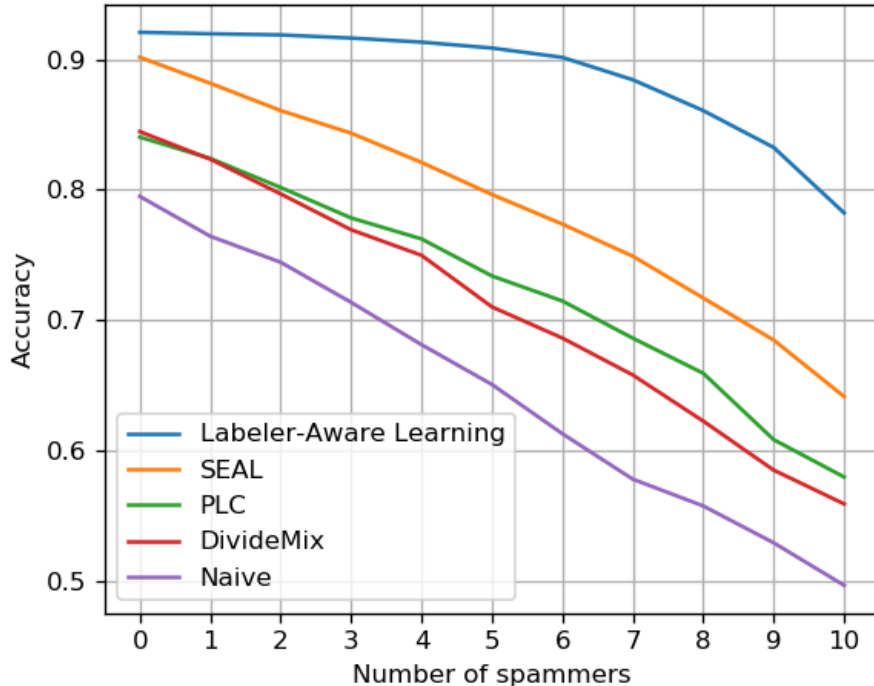
SVHN Test Classification Accuracy in the Presence of J_s Spammers



The inability of previous methods to handle labeler-dependent noise is illustrated most starkly on the more challenging CIFAR-10 dataset, with the results shown in Figure 14. Labeler-aware learning retains over 90% accuracy even if 60% of the labels are spammers, and nearly 80% accuracy even if 100% of the labels are spammers. In contrast, every other algorithm exhibits sharply-decreasing performance as soon as spammers are introduced. Even the algorithms tailored for instance-dependent noise based on noisy labels produced by neural network outputs (i.e., PLC and SEAL) exhibit considerable degradation in accuracy in response to an increasing presence of spammers compared to labeler-aware learning.

Figure 14

CIFAR-10 Test Classification Accuracy in the Presence of J_s Spammers



Labeler-aware learning outperformed all other algorithms on all datasets, and the difference in robustness between labeler-aware learning and previous approaches only grew larger as the datasets became more challenging.

8.4 Measuring Adversarial Robustness Against Label Poisoning Attacks

We extend the previous labeler-dependent noise model to account for adversarial labelers by assuming that the adversary will provide the label that it believes is most likely to be confused with the correct label. We use the MNIST [97] and SVHN [98] datasets, both commonly-used benchmark datasets for learning from noisy labels. We evaluate against the DivideMix [12], progressive label correction (PLC) [30], and self-evolution average label

(SEAL) [28] algorithms, representing the current state-of-the-art in learning from noisy labels under both class-conditional and instance-dependent noise models.

8.4.1 Experimental Setup

We parameterize each experiment by the amount of data provided by adversarial labelers. For data flooding attacks, we fix the amount of data provided by good-faith labelers, and vary the amount of data provided by a single adversarial labeler. For multiple adversaries attacks, we fix the amount of data provided by each labeler, and vary the number of adversarial labelers.

For each experiment, we modeled J labelers by training J neural networks h_j on small subsets \mathcal{D}_j^{tr} , drawn without replacement from the training dataset \mathcal{D} . Each labeler j then provided labels on randomly-partitioned subsets \mathcal{X}_j of the remainder of the training data, with $\cup_j \mathcal{X}_j = \{\mathcal{X} \setminus \{\cup_j \mathcal{X}_j^{tr}\}\}$. Good-faith labelers provided labels following $\hat{y}_{ij} = \operatorname{argmax}_k h_j(x_i)$, while adversarial labelers followed $\hat{y}_{ij} = \operatorname{arg}_2 \max_k h_j(x_i)$ (Equation 7.1). For the MNIST dataset, we set $J = 10$, $N_{tr} = 200$, and each h_j was a randomly-initialized ResNet-18 [102], producing approximately 7.5% natural error. For the SVHN dataset, we set $J = 5$, $N_{tr} = 20,000$, and each h_j was a randomly-initialized Wide ResNet-50 [103], producing approximately 9% natural error. Our noise simulation process is shown in Algorithm 3.

Algorithm 3 Simulating Labeler-Dependent Noise on a Cleanly-Labeled Dataset

Inputs: J , the number of label sources; A , the number of adversarial labelers; N_{tr} , the number of data on which to train each source; N_j , the number of data on which labels were provided by each source $j \in J$; $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, the full training dataset with ground truth labels; h , the labeler model architecture

- 1: $\mathcal{D}_L \leftarrow$ Initialize the set of noisy label datasets
 - 2: **for** each labeler $j = 1$ to J **do**
 - 3: Sample $\mathcal{D}_j^{tr} \subset \mathcal{D}$ with $|\mathcal{D}_j^{tr}| = N_{tr}$
 - 4: $\mathcal{D} \leftarrow \{\mathcal{D} \setminus \mathcal{D}_j^{tr}\}$ (remove labeler training data from pool)
 - 5: Train h_j on $\mathcal{D}_j^{tr} = \{\mathcal{X}_j^{tr}, \mathcal{Y}_j^{tr}\}$
 - 6: Sample $\mathcal{D}_j \subset \mathcal{D}$ with $|\mathcal{D}_j| = N_j$
 - 7: $\mathcal{D} \leftarrow \{\mathcal{D} \setminus \mathcal{D}_j\}$ (remove labeler’s provided data from pool)
 - 8: **if** $j \notin A$ **then**
 - 9: $\hat{\mathcal{Y}}_j \leftarrow \operatorname{argmax}_k h_j(\mathcal{X}_j)$ (select most confident class as good-faith label)
 - 10: **else**
 - 11: $\hat{\mathcal{Y}}_j \leftarrow \operatorname{arg}_2 \max_k h_j(\mathcal{X}_j)$ (select the second most confident class as adversarial label)
 - 12: **end if**
 - 13: $\mathcal{D}_L \leftarrow \{\mathcal{D}_L \cup \{\mathcal{X}_j, \hat{\mathcal{Y}}_j\}\}$
 - 14: **end for**
- Output:** \mathcal{D}_L
-

Once the noisy labels were generated by each labeler, the noisily-labeled data were passed to the learning algorithm under test. For our labeler-aware approach, we provided each labeler’s data-label pairs $\{\mathcal{X}_j, \hat{\mathcal{Y}}_j\}$ as separate datasets. For the labeler-agnostic approaches, we combined the data-label pairs from each source into a single dataset (i.e. as they would be observed under labeler-agnostic assumptions). We repeated each experiment ten times for MNIST and five times for SVHN, and we report the results as the mean classification accuracies in response to increasing adversarial noise, bounded by their 95% confidence intervals based on the two-sided Student’s t -test.

8.4.2 Model Selection and Hyperparameter Tuning

Due to modular nature of the labeler-aware learning framework, the selections for the semi-supervised learning and learning from crowds algorithms to use for each stage constitute its main high-level hyperparameters. We note that the framework’s modularity allows for the seamless replacement of any or all of these algorithmic choices; we demonstrate this modularity by selecting different SSL algorithms for each experiment. For the MNIST dataset, we use auxiliary deep generative models as our SSL algorithm due to its small parameter footprint [104]; for the SVHN dataset, we chose the FixMatch algorithm as representative of the current state-of-the-art for semi-supervised learning [105]. For both datasets, we used OpinionRank (Chapter 5) as our learning from crowds algorithm.

Since we cannot assume the presence of a clean validation set (as discussed in Section 3.1), we do not perform any ground truth-based hyperparameter tuning or model selection, similarly as was done in [53]. Instead, the hyperparameters for each component of our modular framework, as well as those of the algorithms against which we are comparing, were selected based on the suggestions of the original authors of each algorithm, available in their publicly-available online implementations. We made only minor adjustments in order to accommodate differences in intended datasets, and in all cases we tested our changes in the non-adversarial setting to ensure fair comparison.

Due to this lack of any fine-tuning, we are likely reporting conservative results for all algorithms, including our own. However, we are emphatically *not* presenting our results as benchmark scores; rather, we are interested in the overall *trends of behavior* that characterize the vulnerabilities of each algorithm to increasing intensities of adversarial attacks. While

it may be possible to obtain minor improvements through extensive hyperparameter tuning on a clean validation dataset, we argue that because we cannot assume the existence of such a set, such tuning would constitute invalidating data leakage. More importantly, however, we believe that such minor improvements would not change the overall structure of our results with respect to the characteristics of each algorithm’s vulnerability to adversarial label noise.

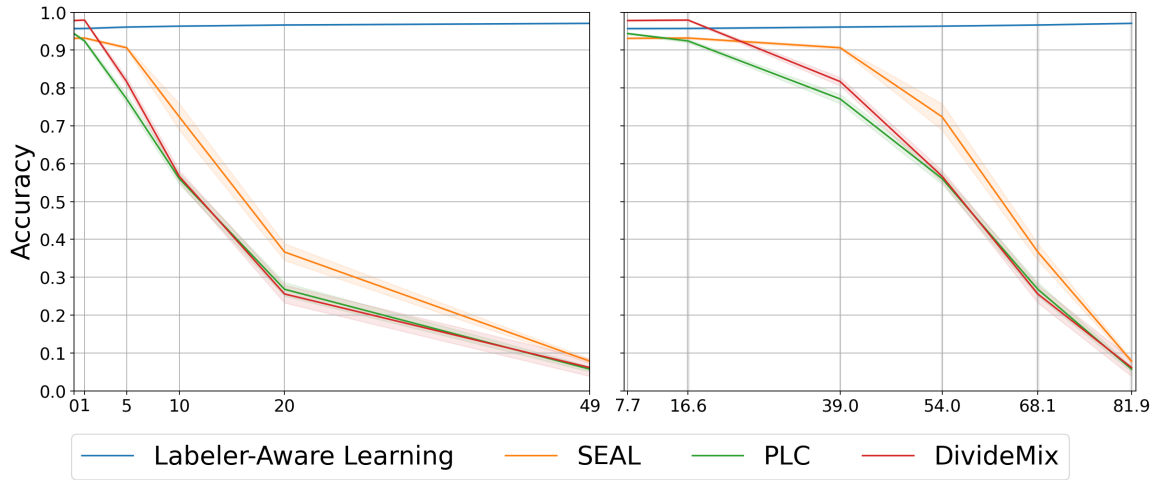
8.4.3 Data Flooding Experiments

For the MNIST dataset, nine good-faith labelers each provided $N_j = 1,000$ data featuring only natural labeling error, and a single adversarial labeler provided an amount of noisy data varying between 0 and 49,000 with maximally-confusing labels. For the SVHN dataset, four good-faith labelers each provided $N_j = 20,000$ data featuring only natural labeling error, and a single adversarial labeler provided an amount of noisy data varying between 0 and 424,000 with maximally-confusing labels. For data flooding experiments, the total amount of data visible to the learner varied with the size of the adversary, with the total amount of good-faith labeled data remaining fixed.

Figure 15 and Figure 16 show the classification accuracies of each algorithm in response to increasing amounts of label noise from a single adversary on the MNIST and SVHN datasets. We observe that all three state-of-the-art algorithms for learning from noisy labels fail under increasing levels of adversarial noise. In contrast, our labeler-aware approach remains robust even under extreme adversarial label noise. These results indicate that recognizing the multiple-labeler paradigm of label gathering is critical in designing robust algorithms for learning from noisy labels.

Figure 15

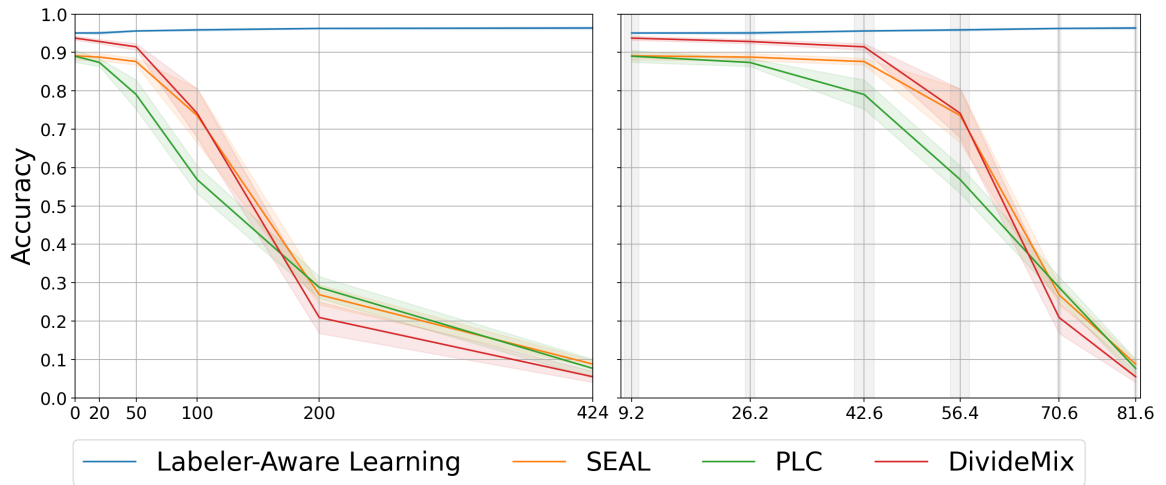
Test Accuracy Under Data Flooding Attacks on MNIST



Note. Shaded regions show the 95% confidence intervals based on the two-sided Student's t -test. Left: Horizontal axis scaled to the adversarial attack size. Right: Horizontal axis scaled to the approximate label noise rate.

Figure 16

Test Accuracy Under Data Flooding Attacks on SVHN



Note. Shaded regions show the 95% confidence intervals based on the two-sided Student's t -test. Left: Horizontal axis scaled to the adversarial attack size. Right: Horizontal axis scaled to the approximate label noise rate. Vertical grey bars indicate standard deviation about the mean of experimental noise rates.

8.4.4 Multiple Adversaries Experiments

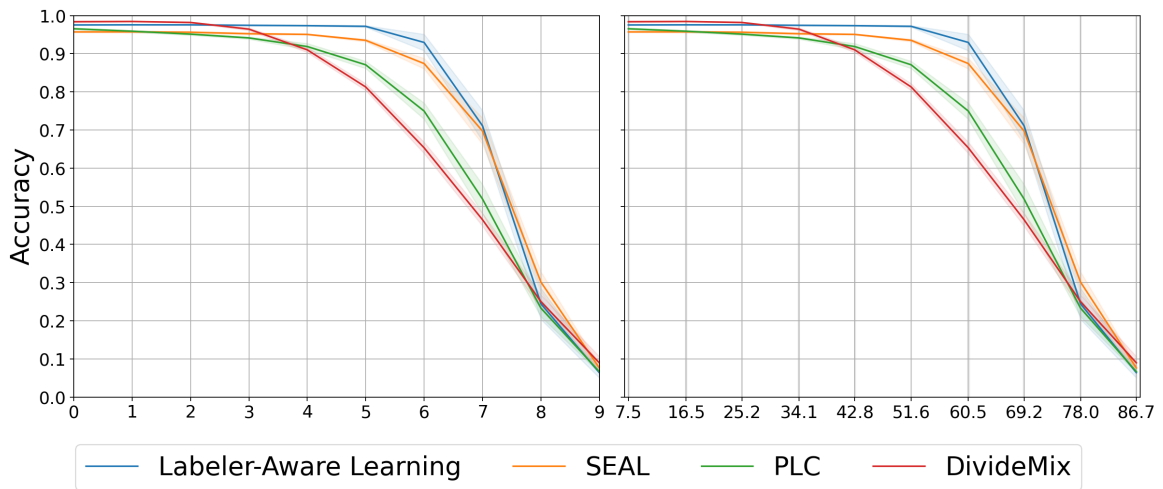
For the MNIST dataset, each of the ten labelers provided $N_j = 5,800$ data, and the number of adversaries A was varied between 0 and 9. For the SVHN dataset, each of the five labelers provided $N_j = 100,000$ data, and A was varied between 0 and 4. For multiple adversaries experiments, the total amount of data visible to the learner is fixed.

Figure 17 and Figure 18 show the classification accuracies of each algorithm in response to increasing amounts of adversarial noise caused by multiple adversaries. We observe that our labeler-aware framework remains robust against larger fractions of adver-

serial labelers compared to other methods. Naturally, all algorithms, including ours, fail under extreme numbers of adversarial labelers; this phenomenon is a well-known result from ensemble learning that can be traced back to the Condorcet jury theorem [60] and its modern extensions [62, 64].

Figure 17

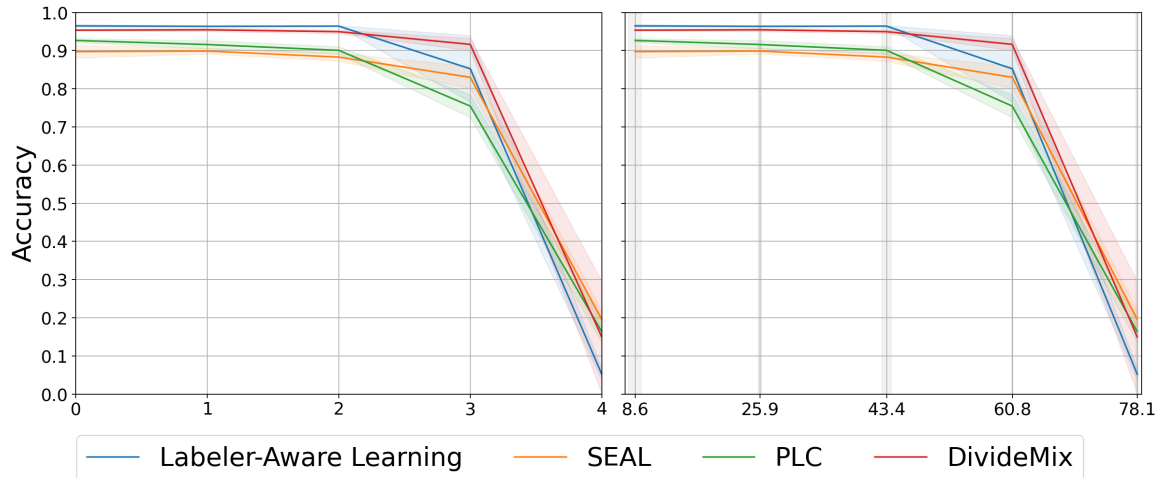
Test Accuracy Under Multiple Adversaries Attacks on MNIST



Note. Shaded regions show the 95% confidence intervals based on the two-sided Student's t -test. Left: Horizontal axis scaled to the adversarial attack size. Right: Horizontal axis scaled to the approximate label noise rate.

Figure 18

Test Accuracy Under Multiple Adversaries Attacks on SVHN



Note. Shaded regions show the 95% confidence intervals based on the two-sided Student's t -test. Left: Horizontal axis scaled to the adversarial attack size. Right: Horizontal axis scaled to the approximate label noise rate. Vertical grey bars indicate standard deviation about the mean of experimental noise rates.

8.5 Adversarial Backdoor Defense

Because adversarial backdoor attacks constitute a serious threat to sensitive real-world applications, such as autonomous vehicles, we conduct a set of experiments to determine the effectiveness of labeler-aware learning as a proactive defensive strategy against such attacks. We conduct our experiments using three standard benchmark datasets: MNIST [97], CIFAR-10 [99], and the German Traffic Sign Recognition Benchmark (GTSRB) dataset [106]. Due to ethical concerns, we deliberately do not perform experiments on facial recognition datasets [107]; however, we recognize and acknowledge that our research

is relevant and may be applied in this area without our knowledge or consent.

For each dataset, we assumed that the training dataset was built from non-overlapping contributions from 5 total labelers. We performed experiments on all datasets for a number of adversaries ranging from 1 to 3, and each adversary was assumed to have contributed 10% of the total training dataset. Each adversary’s objective was to force the classifier to misclassify test images as belonging to a particular target class. For all datasets, these target classes were fixed at class ordinals 0, 7, and 4 for adversaries 1, 2, and 3, respectively; these choices were made arbitrarily for the purposes of statistical analysis over repeated experiments, and our results extend beyond these choices without any loss of generality. After the adversaries’ data splits were apportioned, the remaining training data were divided evenly among the remaining (good-faith) labelers. All data splits were performed randomly, without fixing the random seed.

Because our defensive strategy does not attempt to perform any kind of data cleaning or backdoor pattern detection, we permitted the adversaries to use their strongest possible attacks, i.e. clearly-visible, high-intensity patterns such as those proposed by BadNets [43]. Each adversary injected their backdoor pattern into 100% of their contribution to the training database, corresponding to 10% of the total training dataset, and flipped all of their training labels to their desired target class. Thus, under three adversaries, a total of 30% of the training data have backdoor patterns and false labels.

For each backdoor trigger–target class pair, we verified the effectiveness of the attack by training a baseline classifier and confirming both that the classifier’s performance on clean test data was unimpeded and that the backdoor trigger forced the classifier to output the target class. For both the baseline classifier as well as our defensive strategy, our testing

procedure consisted of two steps:

1. Evaluate the model accuracy on the clean test data (without backdoor triggers). In this stage, we are merely verifying that the model performs reasonably well on clean data; we are not attempting to obtain state-of-the-art performance.
2. For each adversary, construct an adversarial test set by applying the corresponding backdoor trigger to the entire clean testing dataset, and evaluate the model accuracy on the adversarial test data.

Combining both stages, a backdoor attack is considered successful if it can force the model to misclassify most or all of the adversarial test data as the target class, while also allowing the model to correctly classify the clean test data when the backdoor trigger is absent, thus demonstrating that the misclassification is solely due to the presence of the backdoor trigger. All experiments were repeated over 5 runs, and we report the average metrics along with the 95% confidence intervals calculated using the two-sided Student’s t -test.

8.5.1 Baseline Classifier

For all datasets, the baseline classifier was a randomly-initialized PreAct ResNet-18, which was trained for 30 epochs. The optimizer was stochastic gradient descent (SGD), with a learning rate of 0.02, momentum of 0.9, and weight decay of 5×10^{-4} . For the GTSRB dataset, the training and testing images were resized to a standard size of 32×32 , and the learning rate was annealed by a factor of 0.1 at epochs 15 and 25. For the CIFAR-10 and GTSRB datasets, the training images were augmented using RandAugment [108].

The experiments on the baseline classifier verified that the backdoor attack was

effective with pinpoint precision. While the samples in the clean test dataset were classified correctly with expected performance, the introduction of the backdoor trigger into the test dataset caused nearly universal misclassification of all test samples into the classes targeted by each adversary.

8.5.2 MNIST Experiments

For the MNIST experiments, adversaries injected a backdoor pattern into one of the corners of the sample. For the semi-supervised predictive ensemble, we trained one randomly-initialized auxiliary deep generative model (ADGM) [104] for each of the five subsets of the training data (corresponding to the five labelers to the dataset). After the ADGMs were trained, they were independently tested on the clean testing data as well as the malicious test data with backdoor triggers. The outputs of each ADGM were integrated into a single label per instance using OpinionRank (Chapter 5).

8.5.3 CIFAR-10 Experiments

For the CIFAR-10 experiments, adversaries injected a backdoor pattern into one of the corners of the sample. We trained one randomly-initialized Wide ResNet-28 for each of the five subsets of the training data using the FixMatch algorithm [105]. After all FixMatch models were trained, they were independently tested on the clean testing data as well as the malicious test data with backdoor triggers. The outputs of each FixMatch model were integrated into a single label per instance using OpinionRank.

8.5.4 GTSRB Experiments

For the GTSRB experiments, adversaries placed a backdoor pattern at a randomly-chosen location within the region of interest of the sample (ground truth for this region is provided by the dataset). This strategy most closely represents our updated threat model: for example, a malicious actor wishing to attack autonomous vehicles could place stickers featuring the backdoor patterns on physical street signs. We trained one randomly-initialized Wide ResNet-28 for each of the five subsets of the training data using the FixMatch algorithm. The outputs of each FixMatch model were integrated into a single label per instance using OpinionRank.

8.5.5 Analysis of Results

Table 9 shows the performances of both the baseline classifier, as well as that of labeler-aware training, against a single adversary, on all three datasets; results are reported as means and 95% confidence intervals over 5 runs. As discussed in Section 8.5.1, the baseline classifier exhibited a catastrophic failure in accuracy as the adversary forced the classifier to produce the target output.

Table 9*Test Accuracy Against a Single Adversary*

Baseline Classifier			Labeler-Aware Training		
Dataset	Clean	Adversarial	Dataset	Clean	Adversarial
MNIST	99.03 ± 0.23	9.81 ± 0.01	MNIST	97.58 ± 0.06	97.14 ± 0.09
CIFAR-10	87.28 ± 0.43	12.29 ± 0.39	CIFAR-10	93.40 ± 0.17	93.41 ± 0.25
GTSRB	93.50 ± 1.84	0.48 ± 0.00	GTSRB	98.06 ± 0.06	97.59 ± 0.10

Even worse, the baseline classifier was vulnerable not only to a single adversary, but to multiple simultaneous adversaries: Table 10 shows how *any* adversary who contributes to the training database is able to compromise a naïve classifier. Note that the test examples from the target classes were not removed from the test set, so the baseline classifier retains a lower bound for performance corresponding to the intersection of the target class columns with the diagonal of the confusion matrix (see Figure 19).

Table 10*Test Accuracy Against Multiple Adversaries – Baseline Classifier*

Dataset	Two Adversaries			Three Adversaries			
	Clean	Adv. 1	Adv. 2	Clean	Adv. 1	Adv. 2	Adv. 3
MNIST	99.3±0.2	9.8±0.0	10.9±1.7	99.1±0.2	9.8±0.0	10.2±0.0	9.8±4.8
CIFAR-10	85.9±1.2	12.1±0.4	8.5±1.4	84.0±2.9	11.5±0.4	6.5±1.6	6.7±2.4
GTSRB	91.2±1.5	0.4±0.0	3.5±0.0	93.9±1.4	0.7±0.0	3.5±0.0	5.2±0.0

In contrast, both Table 9 and Table 11 show that labeler-aware training produces models that are robust to adversarial backdoor triggers. In the single-adversary scenario, even though the adversary contributed 10% of the training database, the model was still able to produce correct classifications in the presence of the adversary’s backdoor trigger, exhibiting no meaningful change in performance. Even against multiple simultaneous adversaries, the models produced by labeler-aware training remain resilient against all backdoor attacks.

Table 11*Test Accuracy Against Multiple Adversaries – Labeler-Aware Training*

Dataset	Two Adversaries			Three Adversaries			
	Clean	Adv. 1	Adv. 2	Clean	Adv. 1	Adv. 2	Adv. 3
MNIST	97.6±0.1	97.2±0.2	97.3±0.1	95.1±5.8	93.9±6.5	95.7±1.8	95.0±2.4
CIFAR-10	93.4±0.1	93.5±0.1	93.4±0.2	92.6±0.2	92.6±0.3	92.6±0.2	92.5±0.2
GTSRB	97.8±0.2	97.4±0.2	97.4±0.2	97.5±0.2	96.9±0.2	97.0±0.2	97.0±0.3

Figure 19 highlights the improvement of labeler-aware training over the agnostic baseline classifier. The four columns correspond to test data: (a) with clean labels, (b) with a backdoor trigger targeting class 0, (c) with a backdoor trigger targeting class 7, and (d) with a backdoor trigger targeting class 4. Shown in the top row are confusion matrices produced by a labeler-agnostic PreAct ResNet-18; while the performance of the classifier on clean data is strong, any adversary may force the classifier to produce their desired label by applying their backdoor trigger. In contrast, the confusion matrices in the bottom row are produced by labeler-aware training; the labeler-aware model is completely robust against all adversaries, and the backdoor triggers have been rendered ineffective.

Figure 19

Confusion Matrices for Accuracy Performance on GTSRB



8.6 Summary and Discussion of Overall Experimental Results

The results across our entire experimental suite confirm that previous methods for modeling label noise and attempting to mitigate its detrimental impacts are incomplete and insufficient. Beginning with the crowdsourcing paradigm as a special case of labeler-dependent noise, we have shown that the overparameterized models proposed in previous works fail to generalize outside of their laboratory settings. Furthermore, the hand-crafted parameter estimation algorithms designed specifically to solve these models are defeated by OpinionRank, even in the environments for which these algorithms were specifically parameterized.

Proceeding to the general labeler-dependent noise model further emphasizes the need for the more general noise model. We have shown that the methods proposed in earlier works are unable to maintain robustness against label noise in the general model, implying that the class-conditional and instance-dependent models motivating these methods' designs are incomplete. In contrast, we have shown that by explicitly considering the multiple labeler paradigm, it is possible to exploit the information present in the labeler-aware metadata in order to design robust training frameworks.

Finally, we considered several scenarios of extreme label noise, i.e. false labels presented by an adversarial labeler. Unlike the previous state-of-the-art methods, our labeler-aware learning framework is strongly robust against most kinds of adversarial attacks in the multiple labeler setting. The only scenario in which labeler-aware learning failed to be robust was in the extraordinary (and extraordinarily unlikely) case where greater than 70% of all labelers were adversarial; however, this is a known limitation from ensemble

learning, and we believe that under the generalized Condorcet jury theorem it is not possible to improve upon this result in the general case. Despite this limitation, the exploitation of labeler awareness allowed labeler-aware learning to remain *more* robust than the alternative algorithms under extreme fractions of adversarial labelers. Furthermore, labeler-aware learning was not negatively impacted at all in the other scenarios; both data flooding attacks and backdoor attacks, despite their attractiveness to potential malicious agents, are comprehensively defeated by using labeler-aware learning. Hence, labeler-aware learning is effective as a proactive defensive strategy against the always-looming threat of adversarial attacks.

Chapter 9

Conclusions

Label-based supervised and semi-supervised learning requires trust in the veracity of the label information accompanying training datasets. However, noisy labels are ubiquitous in real-world data, a fact that is reflected in the increasingly large body of work focused on mitigating their effects. While conventional models of class-conditional label noise have made great strides in analyzing label noise, recent work on instance-dependent noise has rightfully pointed out that real-world noise is not necessarily class-conditional.

9.1 Contributions

By considering real-world modern data collection procedures, we extended the observation of feature-awareness to labeler-awareness, and formulated a general model of label noise, called labeler-dependent noise (LDN). We demonstrated that under the general LDN model, previous state-of-the-art methods for learning from noisy labels are unable to maintain robustness against increasing levels of label noise. In response, we proposed a modular framework for labeler-aware learning (LAL)—inspired by contemporary research in psychology of learning—that succeeds in remaining robust against extreme fractions of noisy labelers. Furthermore, we considered several adversarial scenarios, including the timely threat of backdoor trigger injection, and demonstrated that LAL serves as an effective tool for proactive defenses against such malicious attacks.

9.2 Future Work

We foresee that our work may be of great benefit for industrial applications, where large volumes of noisily-labeled data are commonplace. We also envision our framework

seamlessly integrating into future work on continual or online learning, as autonomous agents require the ability to synthesize information from multiple unreliable data sources.

Future development of the labeler-dependent noise model should include exploring different models for describing the hammer-spammer dynamics. In particular, while in this dissertation we have focused on the beta-binomial distribution (as a principled distribution over ordered class likelihoods), a more general method may be to use the Dirichlet-multinomial distribution. Even within the beta-binomial distribution, while we fixed the alpha parameter at $\alpha = 1$ (so as to ensure monotonicity), it may be interesting to investigate other parameterizations that shift the probability mass to the right (which may describe adversarial labelers).

Furthermore, while the labeler-aware learning framework was developed in response to the multiple-labeler dynamics of the LDN model, the framework may be used as an ensemble method even in the absence of multiple labelers. For example, it may be intriguing to experiment with applying labeler-aware learning in a setting where labeler IDs are not available by synthetically partitioning the observed data (either randomly, or following a selection rule). This approach may be beneficial in diluting the effect of label noise by “spreading” the noisy labels between the partitions in a heterogeneous manner, after which LAL may be used to eliminate their effects.

Most importantly, we hope to initiate an awareness and a shift in how labeled data are gathered, with datasets retaining information about which data are labeled by which labelers, so that labeler-aware learning may be leveraged in more and more general scenarios and applications. Such metadata can be trivially obtained, and can be stored in ways that maintain the privacy and anonymity of the dataset contributors. Previous

datasets, even those which were gathered via distributed methods, discarded any labeler information as irrelevant; we wish to correct this mentality and encourage future dataset collectors to preserve this critical and versatile information, so that it may be studied and utilized to develop more robust algorithms and learning frameworks, as well as deepen our understanding of the fundamental nature of machine learning.

References

- [1] R. Descartes, *Meditationes de Prima Philosophia, in qua Dei existentia et animæ immortalitas demonstratur*, 1641.
- [2] C. Barringer and B. Gholson, “Effects of type and combination of feedback upon conceptual learning by children: Implications for research in academic learning,” *Review of Educational Research*, vol. 49, no. 3, pp. 459–478, 1979.
- [3] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep learning for computer vision: A brief review,” *Computational Intelligence and Neuroscience*, vol. 2018, 2018.
- [4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Nee-lakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” in *NeurIPS*, vol. 33, 2020, pp. 1877–1901.
- [5] A. M. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, “Deep learning for financial applications: A survey,” *Applied Soft Computing*, vol. 93, p. 106384, 2020.
- [6] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, “Event-based vision meets deep learning on steering prediction for self-driving cars,” in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5419–5427.
- [7] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher, “Deep learning-enabled medical computer vision,” *NPJ Digital Medicine*, vol. 4, no. 1, pp. 1–9, 2021.
- [8] B. Fréney and M. Verleysen, “Classification in the presence of label noise: a survey,” *IEEE TNNLS*, vol. 25, no. 5, pp. 845–869, 2013.
- [9] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” in *International Conference on Learning Representations*, 2017.
- [10] E. L. Lehmann and G. Casella, *Theory of point estimation*. Springer Science & Business Media, 2006.
- [11] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning*. The MIT Press, 09 2006.
- [12] J. Li, R. Socher, and S. C. Hoi, “Dividemix: Learning with noisy labels as semi-supervised learning,” in *ICLR*, 2020.
- [13] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, “Learning to learn from noisy labeled data,” in *CVPR*, 2019, pp. 5051–5059.

- [14] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, “Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis,” *Med. Image Anal.*, vol. 65, p. 101759, 2020.
- [15] J. Zhang, V. S. Sheng, T. Li, and X. Wu, “Improving crowdsourced label quality using noise correction,” *IEEE TNNLS*, vol. 29, no. 5, pp. 1675–1688, 2017.
- [16] V. S. Sheng, J. Zhang, B. Gu, and X. Wu, “Majority voting and pairing with multiple noisy labeling,” *IEEE Trans Knowl Data Eng*, vol. 31, no. 7, pp. 1355–1368, 2017.
- [17] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” in *ICLR*, 2017.
- [18] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, “Learning from noisy labels with deep neural networks: A survey,” *IEEE TNNLS*, 2022.
- [19] G. Algan and I. Ulusoy, “Image classification with deep learning in the presence of noisy labels: A survey,” *Knowl.-Based Syst.*, vol. 215, 2021.
- [20] D. Angluin and P. Laird, “Learning from noisy examples,” *Mach. Learn*, vol. 2, no. 4, pp. 343–370, 1988.
- [21] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, “Making deep neural networks robust to label noise: A loss correction approach,” in *CVPR*, 2017, pp. 1944–1952.
- [22] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, “Joint optimization framework for learning with noisy labels,” in *CVPR*, 2018.
- [23] B. Mirzasoleiman, K. Cao, and J. Leskovec, “Coresets for robust training of deep neural networks against noisy labels,” in *NeurIPS*, vol. 33, 2020.
- [24] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” in *NeurIPS*, vol. 31, 2018.
- [25] N. Manwani and P. Sastry, “Noise tolerance under risk minimization,” *IEEE Trans. Cybern*, vol. 43, no. 3, pp. 1146–1151, 2013.
- [26] X. Xia, T. Liu, B. Han, N. Wang, M. Gong, H. Liu, G. Niu, D. Tao, and M. Sugiyama, “Part-dependent label noise: Towards instance-dependent label noise,” in *NeurIPS*, vol. 33, 2020, pp. 7597–7610.
- [27] Q. Wang, B. Han, T. Liu, G. Niu, J. Yang, and C. Gong, “Tackling instance-dependent label noise via a universal probabilistic model,” in *AAAI*, vol. 35, no. 11, 2021, pp. 10 183–10 191.
- [28] P. Chen, J. Ye, G. Chen, J. Zhao, and P.-A. Heng, “Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise,” in *AAAI*, vol. 35, no. 13, 2021.

- [29] J. Cheng, T. Liu, K. Ramamohanarao, and D. Tao, “Learning with bounded instance and label-dependent label noise,” in *ICML*, 2020.
- [30] Y. Zhang, S. Zheng, P. Wu, M. Goswami, and C. Chen, “Learning with feature dependent label noise: a progressive approach,” *ICLR*, 2021.
- [31] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conf. on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [32] R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng, “Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks,” in *EMNLP*, 2008, p. 254–263.
- [33] A. Wang, C. D. Hoang, and M.-Y. Kan, “Perspectives on crowdsourcing annotations for natural language processing,” *Language Resources and Evaluation*, vol. 47, no. 1, p. 9–31, Mar. 2013.
- [34] B. Guo, H. Chen, Y. Liu, C. Chen, Q. Han, and Z. Yu, “From crowdsourcing to crowd-mining: Using implicit human intelligence for better understanding of crowdsourced data,” *World Wide Web*, vol. 23, no. 2, pp. 1101–1125, 2020.
- [35] G. Xintong, W. Hongzhi, Y. Song, and G. Hong, “Review: Brief survey of crowdsourcing for data mining,” *Expert Syst. Appl.*, vol. 41, no. 17, p. 7987–7994, 2014.
- [36] L. Beyer, O. J. Henaff, A. Kolesnikov, X. Zhai, and A. van den Oord, “Are we done with imagenet?” *arXiv preprint arXiv:2002.05709*, 2020.
- [37] S. Yun, S. J. Oh, B. Heo, D. Han, J. Choe, and S. Chun, “Re-labeling imagenet: from single to multi-labels, from global to localized labels,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021.
- [38] A. P. Dawid and A. M. Skene, “Maximum likelihood estimation of observer error-rates using the em algorithm,” *J. R. Stat. Soc., C: Appl. Stat.*, vol. 28, no. 1, pp. 20–28, 1979.
- [39] J. Whitehill, T.-f. Wu, J. Bergsma, J. Movellan, and P. Ruvolo, “Whose vote should count more: Optimal integration of labels from labelers of unknown expertise,” *NeurIPS*, vol. 22, pp. 2035–2043, 2009.
- [40] P. Welinder, S. Branson, P. Perona, and S. Belongie, “The multidimensional wisdom of crowds,” *NeurIPS*, vol. 23, pp. 2424–2432, 2010.
- [41] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *IEEE Secur Priv.* IEEE, 2017, pp. 39–57.
- [42] A. Shafahi, W. R. Huang, M. Najibi, O. Suci, C. Studer, T. Dumitras, and T. Goldstein, “Poison frogs! targeted clean-label poisoning attacks on neural networks,” in *NeurIPS*, vol. 31, 2018.

- [43] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, “Badnets: Evaluating backdooring attacks on deep neural networks,” *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.
- [44] Y. Deng, T. Zhang, G. Lou, X. Zheng, J. Jin, and Q.-L. Han, “Deep learning-based autonomous driving systems: A survey of attacks and defenses,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 7897–7912, 2021.
- [45] N. Cauli, A. Ortis, and S. Battiato, “Fooling a face recognition system with a marker-free label-consistent backdoor attack,” in *International Conference on Image Analysis and Processing*. Springer, 2022, pp. 176–185.
- [46] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, “Learning from massive noisy labeled data for image classification,” in *CVPR*, 2015.
- [47] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, “Learning from noisy labels with distillation,” in *ICCV*, 2017, pp. 1910–1918.
- [48] M. Ren, W. Zeng, B. Yang, and R. Urtasun, “Learning to reweight examples for robust deep learning,” in *ICML*, 2018, pp. 4334–4343.
- [49] S. Thulasidasan, T. Bhattacharya, J. Bilmes, G. Chennupati, and J. Mohd-Yusof, “Combating label noise in deep learning using abstention,” in *ICML*, 2019, pp. 6234–6243.
- [50] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, “How does disagreement help generalization against label corruption?” in *ICML*, 2019, pp. 7164–7173.
- [51] X. Wang, S. Wang, J. Wang, H. Shi, and T. Mei, “Co-mining: Deep face recognition with noisy labels,” in *ICCV*, 2019, pp. 9358–9367.
- [52] K. Yi and J. Wu, “Probabilistic end-to-end noise correction for learning with noisy labels,” in *CVPR*, 2019, pp. 7017–7025.
- [53] E. Arazo, D. Ortego, P. Albert, N. O’Connor, and K. McGuinness, “Unsupervised label noise modeling and loss correction,” in *ICML*, 2019.
- [54] Z. Wang, G. Hu, and Q. Hu, “Training noise-robust deep neural networks via meta-learning,” in *CVPR*, 2020, pp. 4524–4533.
- [55] Y. Xu, L. Zhu, L. Jiang, and Y. Yang, “Faster meta update strategy for noise-robust deep learning,” in *CVPR*, 2021, pp. 144–153.
- [56] P. Chen, B. B. Liao, G. Chen, and S. Zhang, “Understanding and utilizing deep neural networks trained with noisy labels,” in *ICML*, 2019, pp. 1062–1070.
- [57] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “Mixmatch: A holistic approach to semi-supervised learning,” *Advances in neural information processing systems*, vol. 32, 2019.

- [58] Y. Liu and H. Guo, “Peer loss functions: Learning from noisy labels without knowing noise rates,” in *ICML*, 2020, pp. 6226–6236.
- [59] Z. Zhu, T. Liu, and Y. Liu, “A second-order approach to learning with instance-dependent label noise,” *arXiv preprint arXiv:2012.11854*, 2020.
- [60] M. J. A. N. de Caritat Marquis de Condorcet, *Essai sur l’application de l’analyse à la probabilité des décisions rendues à la pluralité des voix*. L’imprimerie royale, 1785.
- [61] R. G. Kazmann, “Democratic organization: A preliminary mathematical model,” *Public Choice*, vol. 16, no. 1, pp. 17–26, 1973.
- [62] B. Grofman, “A comment on ‘democratic theory: A preliminary mathematical model.’,” *Public Choice*, vol. 21, no. 1, pp. 99–103, 1975.
- [63] G. Owen, B. Grofman, and S. L. Feld, “Proving a distribution-free generalization of the condorcet jury theorem,” *Math. Soc. Sci.*, vol. 17, no. 1, pp. 1–16, 1989.
- [64] C. List and R. E. Goodin, “Epistemic democracy: Generalizing the condorcet jury theorem,” *J. Political Philos.*, vol. 9, no. 3, pp. 277–306, 2001.
- [65] D. Zhou, Q. Liu, J. C. Platt, and C. Meek, “Aggregating ordinal labels from crowds by minimax conditional entropy,” in *ICML*, 2014.
- [66] F. Rodrigues, F. Pereira, and B. Ribeiro, “Learning from multiple annotators: Distinguishing good from random labelers,” *Pattern Recognit. Lett.*, vol. 34, no. 12, p. 1428–1436, 2013.
- [67] Y. Yan, R. Rosales, G. Fung, R. Subramanian, and J. Dy, “Learning from multiple annotators with varying expertise,” *Machine Learning*, vol. 95, no. 3, pp. 291–327, 2014.
- [68] A. Ghosh, S. Kale, and P. McAfee, “Who moderates the moderators? crowdsourcing abuse detection in user-generated content,” in *Proc. of the 12th ACM SIGecom*, 2011, p. 167–176.
- [69] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi, “Aggregating crowdsourced binary ratings,” in *Proc. of the 22nd Int. Conf. on World Wide Web*, 2013, p. 285–294.
- [70] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, “Learning from crowds,” *J. Mach. Learn. Res.*, vol. 11, no. 43, pp. 1297–1322, 2010.
- [71] J. Goldberger, “Combining soft decisions of several unreliable experts,” in *2016 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2016, pp. 2334–2338.
- [72] L. Yin, J. Han, W. Zhang, and Y. Yu, “Aggregating crowd wisdoms with label-aware autoencoders,” in *Proc. of the 26th Int. Joint Conf. on Artificial Intelligence*, 2017, pp. 1325–1331.

- [73] K. Atarashi, S. Oyama, and M. Kurihara, “Semi-supervised learning from crowds using deep generative models,” in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [74] W. Shi, V. S. Sheng, X. Li, and B. Gu, *Semi-Supervised Multi-Label Learning from Crowds via Deep Sequential Generative Model*, 2020, p. 1141–1149.
- [75] Y. Tong, F. Wang, J. Danovitch, and W. Wang, “When the internet is wrong: Children’s trust in an inaccurate internet or human source,” *British Journal of Developmental Psychology*, vol. 40, no. 2, pp. 320–333, 2022.
- [76] G. Frobenius, *Über Matrizen aus nicht negativen Elementen*. Königliche Akademie der Wissenschaften Sitzungsber, Kön, 1912.
- [77] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web.” Stanford InfoLab, Technical Report 1999-66, 1999.
- [78] J. R. Seeley, “The net of reciprocal influence. a problem in treating sociometric data,” *Can. J. Exp. Psychol.*, vol. 3, p. 234, 1949.
- [79] J. Keener, “The perron-frobenius theorem and the ranking of football teams,” *SIAM Review*, vol. 35, pp. 80–93, 1993.
- [80] Y. Chen, J. Fan, C. Ma, and K. Wang, “Spectral method and regularized mle are both optimal for top-k ranking,” *Ann. Stat.*, vol. 47, no. 4, p. 2204, 2019.
- [81] Á. M. Kovács, E. Téglás, and A. D. Endress, “The social sense: Susceptibility to others’ beliefs in human infants and adults,” *Science*, vol. 330, no. 6012, pp. 1830–1834, 2010.
- [82] S. A. Park, S. Goïame, D. A. O’Connor, and J.-C. Dreher, “Integration of individual and social information for decision-making in groups of different sizes,” *PLoS biology*, vol. 15, no. 6, p. e2001958, 2017.
- [83] F. Rodrigues, F. Pereira, and B. Ribeiro, “Learning from multiple annotators: distinguishing good from random labelers,” *Pattern Recognit. Lett.*, vol. 34, no. 12, pp. 1428–1436, 2013.
- [84] F. Tao, L. Jiang, and C. Li, “Label similarity-based weighted soft majority voting and pairing for crowdsourcing,” *KAIS*, vol. 62, pp. 2521–2538, 2020.
- [85] M. J. Wolf, K. W. Miller, and F. S. Grodzinsky, “Why we should have seen that coming: comments on microsoft’s tay “experiment,” and wider implications,” *The ORBIT Journal*, vol. 1, no. 2, pp. 1–12, 2017.
- [86] J. Seering, M. Luria, G. Kaufman, and J. Hammer, “Beyond dyadic interactions: Considering chatbots as community members,” in *CHI*, 2019, pp. 1–13.
- [87] J. Zhang, B. Zhang, and B. Zhang, “Defending adversarial attacks on cloud-aided automatic speech recognition systems,” in *SCC*, 2019, pp. 23–31.

- [88] T. A. Nguyen and A. T. Tran, “Wanet - imperceptible warping-based backdoor attack,” in *ICLR*, 2021.
- [89] K. Doan, Y. Lao, W. Zhao, and P. Li, “Lira: Learnable, imperceptible and robust backdoor attacks,” in *ICCV*, 2021, pp. 11 966–11 976.
- [90] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted backdoor attacks on deep learning systems using data poisoning,” *arXiv preprint arXiv:1712.05526*, 2017.
- [91] M. Barni, K. Kallas, and B. Tondi, “A new backdoor attack in cnns by training set corruption without label poisoning,” in *ICIP*. IEEE, 2019, pp. 101–105.
- [92] B. Tran, J. Li, and A. Madry, “Spectral signatures in backdoor attacks,” in *NeurIPS*, 2018, pp. 8011–8021.
- [93] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, “Detecting backdoor attacks on deep neural networks by activation clustering,” in *SafeAI@ AAI*, 2019.
- [94] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, “Strip: A defence against trojan attacks on deep neural networks,” in *Annu. Comput. Secur. Appl. Conf.*, 2019, pp. 113–125.
- [95] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, “Februus: Input purification defense against trojan attacks on deep neural network systems,” in *Annu. Comput. Secur. Appl. Conf.*, 2020, pp. 897–912.
- [96] J. Wang, Y. Liu, and C. Levy, “Fair classification with group-dependent label noise,” in *ACM FAccT*, 2021, pp. 526–536.
- [97] Y. LeCun, C. Cortes, and C. Burges, “Mnist handwritten digit database,” *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, vol. 2, 2010.
- [98] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning,” *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.
- [99] A. Krizhevsky, G. Hinton *et al.*, “Learning multiple layers of features from tiny images,” 2009.
- [100] J. Zhong, P. Yang, and K. Tang, “A quality-sensitive method for learning from crowds,” *IEEE TKDE*, vol. 29, no. 12, pp. 2643–2654, 2017.
- [101] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” in *NeurIPS*, vol. 33, 2020, pp. 596–608.
- [102] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778.

- [103] S. Zagoruyko and N. Komodakis, “Wide residual networks,” in *BMVC*, 2016.
- [104] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, “Auxiliary deep generative models,” in *ICML*, 2016, pp. 1445–1453.
- [105] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” *NeurIPS*, vol. 33, 2020.
- [106] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,” *Neural Netw*, vol. 32, pp. 323–332, 2012.
- [107] I. D. Raji, T. Gebru, M. Mitchell, J. Buolamwini, J. Lee, and E. Denton, “Saving face: Investigating the ethical concerns of facial recognition auditing,” in *AAAI/ACM Conf. AI Ethics Soc.*, ser. AIES ’20, 2020, p. 145–151.
- [108] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in *CVPR*, 2020, pp. 702–703.