Semi-Supervised Aggregation of Dependent Weak Supervision Sources With Performance Guarantees

Alessio Mazzetto **Dylan Sam** **Andrew Park** **Eli Upfal** **Stephen H. Bach**
Brown University

Abstract

We develop a novel method that provides theoretical guarantees for learning from weak labelers without the (mostly unrealistic) assumption that the errors of the weak labelers are independent or come from a particular family of distributions. We show a rigorous technique for efficiently selecting small subsets of the labelers so that a majority vote from such subsets has a provably low error rate. We explore several extensions of this method and provide experimental results over a range of labeled data set sizes on 45 image classification tasks. Our performance-guaranteed methods consistently match the best performing alternative, which varies based on problem difficulty. On tasks with accurate weak labelers, our methods are on average 3 percentage points more accurate than the state-of-the-art adversarial method. On tasks with inaccurate weak labelers, our methods are on average 15 percentage points more accurate than the semi-supervised Dawid-Skene model (which assumes independence).

1 INTRODUCTION

Supervised machine learning of high-dimensional models requires significant amounts of labeled data. The process of obtaining labeled data can be very costly, and for many classification tasks, labeled data may not be sufficient for learning. To address this issue, a recent line of research on *weak supervision* (Ratner et al., 2016, 2017; Bach et al., 2017) exploited the information contained in weak classifiers for mildly related tasks, also referred to as *labelers*. The labelers generate noisy labels for abundant unlabeled data, which are then aggregated and used to train an end model for the classification task of interest. Empirically, this process yields better generalization than using the labelers directly for prediction.

In this paper, we focus on the important problem of how to aggregate the output of the labelers to obtain accurate labels for training. This problem is very challenging, as the estimation of statistical properties of the labelers is hard with few labeled data for the classification task of interest. In order to handle this issue, prior work makes strong modeling assumptions on the labelers. The assumption most commonly used is that the labelers make errors independently with respect to the classification task of interest (Ratner et al., 2016). The resulting model is well studied in the literature, and it was first introduced for the similar problem of crowdsourcing, where we want to answer a question by aggregating the answers of different observers of unknown expertise (Dawid and Skene, 1979). If the independence assumption holds, many properties of the labelers—such as their error rates with respect to the classification task of interest—can be estimated with only unlabeled data. Further, taking a majority vote weighted by a function of each labeler's accuracy is the optimal way to combine their outputs to maximize accuracy (Nitzan and Paroush, 1982).

While the independence assumption is reasonable in the crowdsourcing setting, as we are combining the answers of different people who are given the same classification task, it is problematic in the weak supervision setting. There is no reason to expect that the labelers are independent. In fact, labelers are themselves often learned with respect to different classification tasks, and it is likely that there are dependencies introduced based on the similarities of these classification tasks.

Consider the following example, where we want to learn to recognize firetrucks in images of vehicles. A labeler could be a classifier that positively labels images that contain any kind of truck. The labeler makes two kind of mistakes: errors based on the fact that the two

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

classification tasks are different (e.g., there are vehicles that are trucks but are not firetrucks), and errors based on the fact that the labeler is not perfectly accurate with respect to its own classification task. When this labeler makes errors of the first kind (mismatch between tasks), it can be correlated with the mistakes of other labelers. For example, consider another labeler that positively labels images with wheels. If a vehicle is a truck, but not a firetruck, then it is also likely to have wheels. Therefore, these labelers will have correlated errors, and an algorithm that relies on the independence assumption will overweight their outputs when creating training data. Indeed, we will show that such algorithms can achieve large errors when applied to arbitrary sets of labelers. Since our available labelers are often determined by available data, this problem is not easily fixed by manually adjusting the labelers. In this setting, it is of paramount importance to devise methods that can aggregate the output of labelers without assuming independence.

Contributions:

In this work, we introduce novel methods to compute analytical bounds on the worst-case error of the majority vote of any set of labelers. Based on this technique, we develop algorithms that select a subset of the labelers which minimizes this worst-case error. Our key insight is that by using few labeled data and abundant unlabeled data, we can reliably estimate properties of the labelers that are sufficient to bound this worst-case error without any prior assumption on the joint distribution of labelers and true labels. The labeled data is only used to estimate the labelers' individual accuracies, while the unlabeled data is used to estimate properties of the distribution of the output of the labelers.

Our main contributions are the following:

1. We emphasize the importance of developing new tools for aggregating weak supervision sources, showing that solutions that assume independence can yield high error rates when applied to non-independent sources. Furthermore, we prove that in contrast to the independent sources case, effective solutions for the non-independent case require more information than just the accuracy of the labelers (Section 3).

2. We propose a novel method to bound the worst-case error of the majority vote of a set of labelers given their accuracies and the distribution of their agreements. We also develop a fast method to compute this quantity for sets of labelers of size 3 (Section 4.1).

3. Based on the previous method, we propose a novel variant to bound the worst-case error of the majority vote of a set of labelers, using the distribution of their output instead of the distribution of their agreements, which is harder to estimate but provides tighter bounds (Section 4.2).

4. We devise heuristic algorithms that use the bounds computed by these two methods to find a subset of the labelers with small worst-case error of their majority vote (Section 4.3).

5. We conduct experiments on 45 image classification tasks to test the effectiveness of our methods. Our experiments show that our algorithms match or outperform the standard semi-supervised model (Dawid and Skene, 1979) that assumes independence. Our methods achieve as much as 24 percentage points higher accuracy on problems where labelers are inaccurate and there is only few labeled data. Our methods also match or outperform a recent method for labeler aggregation that does not make assumptions about the distribution of labelers' errors (Arachie and Huang, 2019), which does not come with any theoretical guarantees (Section 5).

2 RELATED WORK

The problem of combining the outputs from different weak labelers arises in many different domains and has been widely studied.

In crowdsourcing, the problem is to aggregate labels from different human labelers with unknown reliability. In that setting, it is common to assume independence as the human labelers provide their answers independently from each other. In seminal work, Dawid and Skene (1979) showed how to estimate the reliability of the human annotators with expectation maximization and unlabeled data. The idea is that, as the users are independent, their agreement ratio provides information on their reliability. Since then, many other algorithms (Zhang et al., 2016; Gao and Zhou, 2013; Karger et al., 2014; Ghosh et al., 2011; Dalvi et al., 2013) have been devised that can also provide theoretical guarantees on their estimates based on the amount of unlabeled data used and the properties of the labelers. These works all make the independence assumption.

In recent work (Ratner et al., 2016, 2017; Bach et al., 2017; Varma et al., 2019), the Dawid-Skene model is used as a building block for algorithms that learn a classifier by using a set of hand-engineered weak supervision sources (e.g., short programs that classify according to a simple rule). These works recognized the problem that the independence assumption is unrealistic in this setting. They relaxed the assumption by adding particular kinds of structural dependencies in the crowdsourcing model (e.g., if two specific sources agree, they are likely to both be correct). These structural dependencies need to either be specified beforehand in the model by domain experts, or selected by learning algorithms that rely on assumptions that are hard, if not impossible, to verify in practice. In contrast, our work does not assume any particular family for the distribution of labeler outputs and true labels.

In another line of research (Balsubramani and Freund, 2015a,b), the labelers are used to formulate a minimax game, where labels are given to an unlabeled dataset in order to minimize the error with respect to an adversarial choice of the true labels. This adversarial choice is constrained to satisfy the individual error rates of the labelers, estimated with labeled data. This approach does not require any further assumptions on the distribution among labelers' errors, and the minimax can be optimally solved, using different losses to compute the error (Balsubramani and Freund, 2016). While these works optimize with respect to a similar objective as our solution, there are several differences. First, these related works focus on a transductive setting, and provides guarantees on the labeling of a fixed, unlabeled dataset. In comparison, our guarantees focus on an inductive setting in which we label additional unseen data points. Second, they allow both the adversarial and the optimal predictions to be soft labels, while we focus on a strict hard-classification setting. Finally, solving the minimax game optimally as a linear program requires storing in memory the output of the labelers for the whole unlabeled dataset, and approximate solutions could have an arbitrarily large convergence rate. In contrast, the runtime of our approach only depends on the number of labelers, that is small in a lot of practical settings.

In similar recent work, Arachie and Huang (2019) proposed a weakly supervised learning approach called adversarial label learning that also does not assume any distribution among labelers' errors and learns a (possibly complex) classifier by adversarially updating a set of labels at each parameter update. This algorithm minimizes error of the model with respect to the worst-case labeling that satisfies known error rate constraints on the labelers. Their strategy is similar to ours in that it works when labelers are not independent and employs a worst-case argument. However, their method does not provide any theoretical guarantees on the convergence or error of the algorithm. Further, it couples the training of the classifier with the aggregation of the labelers' outputs. In comparison, our method provides strong theoretical guarantees for labeling training data that can then be used by any learning algorithm.

We note that our setting differs from ensemble learning, where a significant amount of labeled data for the classification task of interest is required to both learn the weak classifiers and to combine them into a stronger classifier. Conversely, in our work, we assume that we have access to few labeled data for the classification task of interest and to labelers that have been trained using labeled data for mildly correlated classification tasks. Therefore, we will not draw a detailed comparison with this method and refer interested readers to Zhang and Ma (2012).

3 PRELIMINARIES

Problem definition: Let $\mathcal{D}$ be a distribution over a domain $\mathcal{X}$, and let $y : \mathcal{X} \to \{0,1\}$ be an unknown binary classification of $\mathcal{X}$. Our goal is to learn y from a collection of weak sources that we call *labelers*. A labeler is a binary classifier $\ell : \mathcal{X} \to \{0,1\}$ that was trained on a task that is weakly related to our target classification y.

Let $S = \{\ell_1, \dots, \ell_n\}$ be the set of available labelers. We assume that for each labeler ℓ_i we have an estimate ϵ_i for $\mathbb{P}_{x\sim\mathcal{D}}(y(x) \neq \ell_i(x))$, the error of the labeler with respect to the target classification y (alternatively, we can assume a small set of y-labeled data for estimating the labelers error rates with respect to y). We suppose that $\epsilon_i \leq 1/2$, otherwise we can always flip the output of labeler ℓ_i. Note that the error rate ϵ_i of labeler ℓ_i accounts for two type of errors: the error of the labeler with respect to the classification it was trained for, and the difference between the classification that ℓ_i was trained for and the target classification y. We make no assumptions on the choice of the labelers in S and possible dependencies between the labelers. (Formally, we make no assumptions on the joint distribution $(\ell_1(x), \dots, \ell_n(x))$ or the joint distribution of the errors, only on the marginal distributions for each labeler.)

For domain $\mathcal{X}$ and labeler set $S = \{\ell_1, \dots, \ell_n\}$, let $\vec{\boldsymbol{\ell}}_S(x) = (\ell_1(x), \dots, \ell_n(x))$ be the function that maps an element $x \in \mathcal{X}$ to the n bit vector of the output of the labelers on input $x \in \mathcal{X}$. Consider the set of functions $\mathcal{F} = \{f : \{0,1\}^n \to \{0,1\}\}$, and recall that our target classification is y. Given a function $f \in \mathcal{F}$, we define its expected error as $\varepsilon(f) = \Pr_{x\sim\mathcal{D}}(y(x) \neq f \circ \vec{\boldsymbol{\ell}}_S(x))$, where the symbol $\circ$ represents the composition operator between functions. To classify according to y using only the outputs of the labelers in S, we are looking for a function $f \in \mathcal{F}$ with minimum expected error with respect to y on the domain $\mathcal{X}$ and distribution $\mathcal{D}$, i.e.,

$$\min_{f\in\mathcal{F}} \varepsilon(f \circ \vec{\boldsymbol{\ell}}_S) \tag{1}$$

where the error rate ε is computed with respect to distribution $\mathcal{D}$ on $\mathcal{X}$.

Independent vs. non-independent labelers: Label aggregations problems similar to (1) have been studied in the context of crowdsourcing, where all the labelers make predictions with respect to the same target classification, and it is reasonable to assume that labelers' errors are independent events.

The independence assumption is very powerful in this setting. Let $S = \{\ell_1, \ldots, \ell_n\}$ be a set of labelers with error rates $(\epsilon_1, \ldots, \epsilon_n)$ with respect to the target classification y. It was shown in Nitzan and Paroush (1982) that if the errors the labelers make are independent events, then the optimal classification of x is "1" iff $\sum_{i=1}^{n}(2\ell_i(x) - 1)\log\frac{1-\epsilon_i}{\epsilon_i} > 0$, and the error rate of this classifer satisfies (Berend and Kontorovich, 2015)

$$-\log\varepsilon(f^* \circ \vec{\boldsymbol{\ell}}_S) = \Theta\left(\sum_{i=1}^{n}\left(\frac{1}{2} - \epsilon_i\right)\log\frac{1-\epsilon_i}{\epsilon_i}\right).$$

However, as shown in the following example, the independence assumption is crucial for this analysis. When applied to labelers with non-independent errors the accuracy of the classifier can be significantly worse.

Proposition 1. *Assume that n is odd. There exists a distribution $\mathcal{D}$ over $\mathcal{X}$, a binary classification of $\mathcal{X}$, and a set of n labelers with same error rate $\epsilon < 1/2$ such that the optimal classifier for independent labelers f^* has error rate at least $2\epsilon - 2\epsilon/(n+1)$.*

Proof. In the appendix. □

The above example shows that classification algorithms that rely on the independence assumption (Zhang et al., 2016; Gao and Zhou, 2013; Karger et al., 2014; Ghosh et al., 2011; Dalvi et al., 2013) can have significantly higher error rate when the independence assumption is violated. The following example further demonstrates that adding non-independent labelers can actually reduce the accuracy even of the simple classifier that returns the majority label output by n classifiers:

Proposition 2. *Assume that n is odd and f_n is a classifier that returns the majority label output by the n labelers. There exists a distribution $\mathcal{D}$ over $\mathcal{X}$, and a binary classification of $\mathcal{X}$, such that f_n has error rate at least median$\{\epsilon_1, \ldots, \epsilon_n\}$, where $\epsilon_1, \ldots, \epsilon_n$ are the error rates of the n labelers.*

Proof. In the appendix. □

Moreover, we can prove a stronger result and show that if the independence assumption does not hold, and if the only knowledge we have about the labelers is their error rate with respect to the target classification, then any function on the output of the labelers could not improve upon the classification given by the most accurate labeler in the set.

Let $\vec{\boldsymbol{\epsilon}} = (\epsilon_1, \ldots, \epsilon_n)$. Define $\mathcal{S}(\vec{\boldsymbol{\epsilon}})$ as the set of all possible set of labelers $\{\ell_1, \ldots, \ell_n\}$ such that $\varepsilon(\ell_i) = \epsilon_i$ for $i = 1, \ldots, n$. We consider the worst case error rate of the best function that maps the output of the labelers to a classification, among all the possible set of labelers S that have error rates $\vec{\boldsymbol{\epsilon}}$, that is:

$$\max_{S\in\mathcal{S}(\vec{\boldsymbol{\epsilon}})}\min_{f\in\mathcal{F}} \varepsilon(f \circ \vec{\boldsymbol{\ell}}_S)$$

Proposition 3. *If the only information given to the classification algorithm is the labels of the labelers in S and their error rates with respect to the target classification, then*

$$\max_{S\in\mathcal{S}(\vec{\boldsymbol{\epsilon}})}\min_{f\in\mathcal{F}} \varepsilon(f \circ \vec{\boldsymbol{\ell}}_S) = \min\{\epsilon_1, \ldots, \epsilon_n\}$$

Proof. In the appendix. □

In the following section we show that we can obtain significantly better results for classification using non-independent labelers by learning additional information about the labelers from their outputs on unlabeled data sampled from the distribution $\mathcal{D}$ over $\mathcal{X}$.

4 METHODS

Given a set of labelers S with no knowledge on the joint distribution of their errors, our goal is to design a classifier that performs better than the result in Proposition 3. We achieve this goal by inferring properties on the joint distribution of the labelers using their outputs on unlabeled data sampled from $\mathcal{D}$. Using this information, we can identify a subset of the labelers such that the majority function on that subset performs better than any single labeler in S.

4.1 Labelers' Pairwise Differences

The first quantity that we will use together with the error rates to characterize the effectiveness of our set of labelers is the pairwise difference. That is, for $i \neq j$, let $d(\ell_i, \ell_j) = \mathbb{P}_{x\sim D}(\ell_i(x) \neq \ell_j(x))$. Note that $|\epsilon_i - \epsilon_j| \leq d(\ell_i, \ell_j) \leq \epsilon_i + \epsilon_j$. Intuitively, labelers whose difference is small contain similar information, hence we would like to identify labelers that significantly differ from one another. It is important to point out that the values $d(\ell_i, \ell_j)$ can be estimated using only unlabeled data.

The following example demonstrates the significance of the pairwise difference values in evaluating classifiers correctness. Suppose that we have three labelers with error rates $\epsilon < 1/3$, and we return their majority vote. If their pairwise difference is 0, then the labelers express the same vote and their majority vote has error ϵ, whereas if their pairwise difference is 2ϵ, then only one labeler can be wrong on each instance, and the majority vote is always correct.

Let $\vec{\boldsymbol{\epsilon}} = (\epsilon_1, \dots, \epsilon_n)$, and let $\mathbf{D} \in \mathbb{R}^{n \times n}$, where $\mathrm{D}_{i,j} = d(\ell_i, \ell_j)$. Define $\mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})$ as the collection of all possible sets of labelers $\{\ell_1, \dots, \ell_n\}$ such that $\varepsilon(\ell_i) = \epsilon_i$ for $i = 1, \dots, n$, and $d(\ell_i, \ell_j) = \mathrm{D}_{i,j}$ for $j \neq i$.

Let $[n] = \{1, \dots, n\}$. For any $\mathcal{I} \subseteq [n]$ of odd size, let $\lambda_{\mathcal{I}}(x)$ be the majority vote of the labelers $\{\ell_i : i \in \mathcal{I}\} \subseteq S$, i.e., $\lambda_{\mathcal{I}}$ returns 1 iff $\sum_{i \in \mathcal{I}} \ell_i(x) > |\mathcal{I}|/2$, else it returns 0. Our method searches for a function in $\mathcal{M} = \{\lambda_{\mathcal{I}}(x) \mid I \subseteq [n] \text{ and } |I| \text{ odd}\}$ that minimizes the worst case performance, over all possible classifications that satisfy the observed properties of the labels. I.e., we are interested in the majority function $\lambda_{\mathcal{I}}$ that satisfies

$$\min_{\lambda_{\mathcal{I}} \in \mathcal{M}} \max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) \tag{2}$$

Our first step is computing the worst case performance of a given function $\lambda_{\mathcal{I}}$:

$$\max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) \tag{3}$$

It turns out that given the error rates $\vec{\boldsymbol{\epsilon}}$, the differences $\mathbf{D}$ of n labelers, and a subset $\mathcal{I} \subseteq [n]$, it is possible to compute the worst-case error of the majority vote $\max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S)$ through a linear program.

Let $\mathcal{I} = \{i_1, \dots, i_k\}$. Let $\tilde{\mathbf{a}} = (\mathrm{a}_1, \dots, \mathrm{a}_k)$ be a random vector that represents whether the output of each labeler is correct, i.e., for $j = 1, \dots, k$, we have that $\mathrm{a}_j \in \{0, 1\}$ and $\mathrm{a}_j = 1$ if and only if labeler i_j is correct. For any $\vec{\boldsymbol{a}} = (a_1, \dots, a_k) \in \{0, 1\}^k$, let

$$\begin{aligned} p_{\vec{\boldsymbol{a}}} &\doteq \mathbb{P}_{x \sim D}(\tilde{\mathbf{a}} = (a_1, \dots, a_k)) = \\ &\mathbb{P}_{x \sim D}(\{\ell_i(x) = y(x) \ \forall i : a_i = 1\} \\ &\quad \cap \{\ell_i(x) \neq y(x) \ \forall i : a_i = 0\}) \end{aligned} \tag{4}$$

Note that if we fix the distribution $\mathcal{D}$ over the domain $\mathcal{X}$, and its binary classification $y(\cdot)$, then a set of labelers $\{\ell_1, \dots, \ell_n\}$ fully determines the values $p_{\vec{\boldsymbol{a}}}$ for $\vec{\boldsymbol{a}} \in \{0, 1\}^k$. Also, for any $\vec{\boldsymbol{a}} \in \{0, 1\}^k$, the majority vote of the labelers in $\mathcal{I}$ is correct if and only if $|\vec{\boldsymbol{a}}|_1 > k/2$, where $|\cdot|_1$ is the $\ell 1$-norm. Hence, the total error of the majority vote is given by

$$\varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) = \sum_{\vec{\boldsymbol{a}}\{0,1\}^k : |\vec{\boldsymbol{a}}|_1 < k/2} p_{\vec{\boldsymbol{a}}}$$

Given a set of labeler $S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})$ the values $p_{\vec{\boldsymbol{a}}}$ cannot be arbitrary as the labelers in $\mathcal{I}$ need to respect the error and difference constraints. The strategy of the linear program is to find the values $p_{\vec{\boldsymbol{a}}}$ (variables of the linear program) that maximize the error of the majority vote of the labelers in $\mathcal{I}$, while satisfying these constraints. These values $p_{\vec{\boldsymbol{a}}}$ are associated with a set of labelers S that maximizes the value (3). The formulation of the linear programming is the following:

$$\max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) = \max \sum_{\vec{\boldsymbol{a}} \in \{0,1\}^k : |\vec{\boldsymbol{a}}|_1 < k/2} p_{\vec{\boldsymbol{a}}} \tag{5}$$

$$\begin{aligned} &(a) \quad \sum_{\vec{\boldsymbol{a}} \in \{0,1\}^k : a_j = 0} p_{\vec{\boldsymbol{a}}} = \epsilon_{i_j} \quad \text{for } j = 1, \dots, k \\ &(b) \quad \sum_{\vec{\boldsymbol{a}} \in \{0,1\}^k : a_h \neq a_j} p_{\vec{\boldsymbol{a}}} = \mathrm{D}_{i_h, i_j} \quad \text{for } h \neq j \\ &(c) \quad \sum_{\vec{\boldsymbol{a}}} p_{\vec{\boldsymbol{a}}} = 1 \\ &(d) \quad p_{\vec{\boldsymbol{a}}} \geq 0 \ \ \forall \vec{\boldsymbol{a}} \end{aligned}$$

In the above linear program, the constraint (a) specifies that each labeler $i \in \mathcal{I}$ must have error rate ϵ_i. The constraint (b) specifies that for any two different labelers $i, j \in \mathcal{I}$, their pairwise difference must be equal to $\mathrm{D}_{i,j}$. The constraints (c) and (d) impose that the variables $p_{\vec{\boldsymbol{a}}}$ are probabilities. We observe that this linear program has $2^{|\mathcal{I}|}$ variables, but only $O(|\mathcal{I}|^2)$ constraints. The discussion above proves the following proposition.

Proposition 4. *Given $\mathcal{I} \subseteq [n]$, and n labelers having error rates $\vec{\boldsymbol{\epsilon}}$ and differences $\mathbf{D}$, the worst-case error of the majority vote of of the labelers in $\mathcal{I}$, i.e., $\max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S)$, is the solution of* (5).

We point out that if $\vec{\boldsymbol{\epsilon}}$ and $\mathbf{D}$ are the true values of respectively the error rates and the pairwise differences of a set of labelers $S = \{\ell_1, \dots, \ell_n\}$, then the linear program (5) always has a feasible solution, as the values $p_{\vec{\boldsymbol{a}}}$ determined by S must be a solution.

We can now apply the above technique in parallel to all the subset of S to compute the best subset:

$$\min_{f \in \mathcal{M}} \max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(f \circ \vec{\boldsymbol{\ell}}_S) = \min_{\mathcal{I} \subseteq [n] : |\mathcal{I}| \text{ is odd}} \max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S)$$

For large numbers of labelers, it may be impractical to solve the linear program for all subsets of S, even in parallel. However, as we show in our experimental results, we achieve high accuracy even when we restrict the search to the majority function on small subsets of S. A particularly efficient solution is considering only subsets of 3 labelers. In that case we have a closed form solution to the linear problem:

Proposition 5. *Let $\mathcal{I} = \{i, j, k\}$. Let q be equal to*

$$q = \frac{1}{2}\max\{\epsilon_i + \epsilon_j - \mathrm{D}_{i,k} - \mathrm{D}_{j,k}, \epsilon_j + \epsilon_k - \mathrm{D}_{i,j} - \mathrm{D}_{i,k}, \epsilon_i + \epsilon_k - \mathrm{D}_{i,j} - \mathrm{D}_{j,k}, 0\}$$

Then, we have that

$$\max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) = \epsilon_i + \epsilon_j + \epsilon_k - \frac{\mathrm{D}_{i,j} + \mathrm{D}_{j,k} + \mathrm{D}_{i,k}}{2} - 2q$$

Proof. In the appendix. □

The proposition above is of independent interest as it clearly shows the behaviour of the worst-case majority vote depending on the error rates and the pairwise differences. To simplify, assume that the three labelers in $\mathcal{I}$ in the statement of the previous proposition have same error rates ϵ and difference d. We have that:

$$\max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) = 3\left(\epsilon - \frac{1}{2}d\right) - 2\max\{\epsilon - d, 0\}$$

That is, if $d \leq \epsilon$, then $\max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) = \epsilon + \frac{1}{2}d$, else $\max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) = 3\left(\epsilon - \frac{1}{2}d\right)$.

In particular, we can see that $\max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) \leq \epsilon$ if and only if $d \geq \frac{4}{3}\epsilon$.

This result reinforces our intuition that the labelers need to significantly differ from one another if we want to improve upon the best-performing labeler in the worst case.

4.2 Labelers' Output Distribution

Up to now, we have shown that the pairwise differences can be effectively used in order to find ways to aggregate the output of the labelers with error guarantees. Most importantly, the pairwise differences can be approximated with only unlabeled data. As there are only $O(n^2)$ pairwise differences, the computation of these values require a limited amount of data if n is small. A natural question is whether there are other quantities that require more unlabeled data to be estimated, but provide further knowledge on the set of labelers.

The second quantity that we will use together with the error rates to characterize the effectiveness of our set of labelers is the distribution of the output of the labelers. We will show that this approach always yields a tighter accuracy than using pairwise differences, but it requires more unlabeled data in order to obtain accurate estimates.

For any $\vec{\boldsymbol{o}} \in \{0,1\}^n$, we can estimate the probability that a set of labelers S output $\vec{\boldsymbol{o}}$. In particular, let $q_{\vec{\boldsymbol{o}}} \doteq \mathbb{P}_{x \sim \mathcal{D}}(\vec{\boldsymbol{\ell}}_S(x) = \vec{\boldsymbol{o}})$ be this probability for any $\vec{\boldsymbol{o}} \in \{0,1\}^n$. Let $\mathcal{Q}$ be the collection of the values $q_{\vec{\boldsymbol{o}}}$. We remark that $\mathcal{Q}$ can be estimated by only using unlabeled data.

Let $\vec{\boldsymbol{\epsilon}} = (\epsilon_1, \ldots, \epsilon_n)$, and let $\mathcal{Q}$ be defined as above. We define $\hat{\mathcal{S}}(\vec{\boldsymbol{\epsilon}}, \mathcal{Q})$ as the family of all possible set of labelers $\{\ell_1, \ldots, \ell_n\}$ such that $\varepsilon(\ell_i) = \epsilon_i$ for $i = 1, \ldots, n$, and the output of the labelers follows the distribution described by $\mathcal{Q}$. To identify the optimal classification function subject to $\vec{\boldsymbol{\epsilon}}$ and $\mathcal{Q}$, we need to evaluate for each $\mathcal{I} \subseteq [n]$ the worst-case error of the majority vote of the labelers in $\mathcal{I}$, that is:

$$\max_{S \in \hat{\mathcal{S}}(\vec{\boldsymbol{\epsilon}}, \mathcal{Q})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) \ . \tag{6}$$

It is important to note that $\mathcal{Q}$ provides more information on the behaviour of the output of the labelers than the pairwise differences. Given $\mathcal{Q}$, it is easy to compute the pairwise differences $\mathbf{D}$ between the labelers, but the opposite is not possible. It immediately follows that $\hat{\mathcal{S}}(\vec{\boldsymbol{\epsilon}}, \mathcal{Q}) \subseteq \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})$, which implies

$$\max_{S \in \hat{\mathcal{S}}(\vec{\boldsymbol{\epsilon}}, \mathcal{Q})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) \leq \max_{S \in \mathcal{S}(\vec{\boldsymbol{\epsilon}}, \mathbf{D})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S)$$

Given the error rates $\vec{\boldsymbol{\epsilon}}$, and the distribution of the output of the labelers described by $\mathcal{Q}$, it is possible to compute the value (6) for $\mathcal{I} = \{i_1, \ldots, i_k\} \subseteq [n]$ through a linear program. The strategy is the same used while working with the pairwise differences, and in particular let $p_{\vec{\boldsymbol{a}}}$ be defined as in (4). Again, the values $p_{\vec{\boldsymbol{a}}}$ for $\vec{\boldsymbol{a}} \in \{0,1\}^k$ are the variables of the linear program. The only change is that we replace the constraints on the pairwise differences with the constraints on the output of the distribution of the labelers. In order to formalize the latter constraints, it is convenient to introduce some notation. For any $\vec{\boldsymbol{a}} \in \{0,1\}^k$, let $r\left(\vec{\boldsymbol{a}} = (a_1, \ldots, a_k)\right) = (b_1, \ldots, b_k)$ be the function that flips every bit in a vector of k bits, i.e., $b_j = 1 - a_j$ for $j = 1, \ldots, k$. For any $\vec{\boldsymbol{o}} \in \{0,1\}^k$, let $h_{\vec{\boldsymbol{o}}}$ be the probability that for an element $x \sim \mathcal{D}$, the vector $(\ell_{i_1}(x), \ldots, \ell_{i_k}(x))$ is equal to $\vec{\boldsymbol{o}}$. The values $h_{\vec{\boldsymbol{o}}}$ can be computed from $\mathcal{Q}$ by marginalizing the distribution of the output of the labelers; in particular we have that:

$$\begin{aligned} h_{\vec{\boldsymbol{o}}} &= \mathbb{P}_{x \sim \mathcal{D}}((\ell_{i_1}(x), \ldots, \ell_{i_k}(x)) = \vec{\boldsymbol{o}}) \\ &= \sum_{\substack{\vec{\boldsymbol{u}} \in \{0,1\}^n : o_j = u_{i_j} \\ \text{for } j = 1, \ldots, k}} q_{\vec{\boldsymbol{u}}} \ . \end{aligned}$$

In this case, the formulation of the linear programming is the following:

$$\max_{S \in \hat{\mathcal{S}}(\vec{\boldsymbol{\epsilon}}, \mathcal{Q})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S) = \max \sum_{\vec{\boldsymbol{a}} \in \{0,1\}^k : |\vec{\boldsymbol{a}}|_1 < k/2} p_{\vec{\boldsymbol{a}}} \qquad (7)$$

$$\begin{aligned} &(a) \quad \sum_{\vec{\boldsymbol{a}} \in \{0,1\}^k : a_j = 0} p_{\vec{\boldsymbol{a}}} = \epsilon_{i_j} \quad \text{for } j = 1, \ldots, k \\ &(b) \quad p_{\vec{\boldsymbol{a}}} + p_{r(\vec{\boldsymbol{a}})} = h_{\vec{\boldsymbol{a}}} + h_{r(\vec{\boldsymbol{a}})} \quad \text{for } \vec{\boldsymbol{a}} \in \{0,1\}^k \\ &(c) \quad \sum_{\vec{\boldsymbol{a}}} p_{\vec{\boldsymbol{a}}} = 1 \\ &(d) \quad p_{\vec{\boldsymbol{a}}} \geq 0 \ \forall \vec{\boldsymbol{a}} \end{aligned}$$

The constraint (b) now requires the values $p_{\vec{\boldsymbol{a}}}$ to satisfy the distribution of the output of the labelers instead of the pairwise differences. This linear program has $O(2^{|\mathcal{I}|})$ variables and $O(2^{|\mathcal{I}|})$ constraints.

Proposition 6. *Given $\mathcal{I} \subseteq [n]$, and n labelers having error rates $\vec{\boldsymbol{\epsilon}}$, and distribution of their outputs following $\mathcal{Q}$, the worst-case error of the majority vote of the labelers in $\mathcal{I}$, i.e. $\max_{S \in \hat{\mathcal{S}}(\vec{\boldsymbol{\epsilon}}, \mathcal{Q})} \varepsilon(\lambda_{\mathcal{I}} \circ \vec{\boldsymbol{\ell}}_S)$, is the solution of* (7)*.*

The same algorithmic extensions used with the previous linear program can be used in this setting to find a subset of labelers whose majority vote has low error in the worst case. We remark that if there is enough unlabeled data to accurately estimate $\mathcal{Q}$, this approach yields a tighter bound than the one obtained with the previous linear program.

4.3 LP with Greedy Extensions

The previous approaches are practical only for small numbers of labelers, since we need to consider $\binom{|S|}{|I|}$ possible majority functions on subsets of $|I|$ labelers. While our method performs well in many settings, even with majority functions of only 3 labelers, there are scenarios in which a small number of labelers cannot provide enough information. Therefore, we develop extensions of the LP approach that considers larger subsets of labelers combined in a greedy fashion. Starting from every labeler $i \in S$, we repeatedly add the two labelers that minimize the worst-case error rate of the majority vote (computed either using the method of Section 4.1 or Section 4.2) until there is no more improvement. The output of this algorithm is the subset that minimizes the worst-case error found. The pseudocode of the algorithm is reported in the appendix. This algorithm captures more information from the original set of labelers compared to the previous method. In particular, this algorithm outputs a subset that is at least as accurate in the worst case as any small size (≤ 3) subset.

5 EXPERIMENTS

We demonstrate the performance of our methods on 45 image classification tasks. We compare our linear program approaches with crowdsourcing, semi-supervised learning, and weakly supervised learning approaches. The selection of the three labelers minimizing the worst-case error computed using the closed formula in Proposition 5 is denoted as **PGMV** (Performance-Guaranteed Majority Vote). Our algorithmic extensions in Section 4.3 are denoted by **PGMV-P** and **PGMV-D**, where P denotes using labelers' pairwise differences and D denotes using labelers' distributions to compute the worst-case errors. The code for the experiments is available online.[1]

Table 1 illustrates that our methods are on average 1 percentage point more accurate than a state-of-the-art weakly-supervised approach and 5 percentage points more accurate than the Dawid-Skene model on tasks that have inaccurate weak labelers. Our methods are within 1 percentage point of the state-of-the-art alternative's accuracy and achieves 2 percentage points higher than Dawid-Skene's accuracy when labeler accuracies are high. In addition, none of the alternative methods provide theoretical guarantees without any assumptions on the joint distribution of labeler outputs and true labels.

5.1 Baselines and Related Algorithms

We describe the various baselines and existing algorithms to which we compare our methods.

Majority Vote (MV): Majority vote predicts the most common label among the labelers' outputs. This method performs well on tasks that have weak labelers with conditionally independent outputs but potentially fails when there are complex dependencies.

Majority Vote with Flips (MV Flip): Since our methods require that each weak labeler has an error $\epsilon < 0.5$, we flip the votes of any labeler with estimated accuracy less than 50%. We consider the resulting majority vote with flipped labelers to understand the impact of this flipping on the resulting accuracies.

Dawid-Skene Estimator (DS): The Dawid-Skene

[1] `https://github.com/BatsResearch/mazzetto-aistats21-code`

Table 1: Comparison of our methods and benchmarks on various image classification tasks. Numbers are accuracy percentages reported as mean ± standard error, computed over 5 random seeds. The fraction after *cp* represents the group of tasks when sorted by committee potential.

Dataset	MV	MV Flip	DS	ALL	**PGMV**	**PGMV-P**	**PGMV-D**
AwA2 ($cp\frac{1}{5}$)	55.9 ± 2.7	79.1 ± 1.1	80.0 ± 1.8	84.2 ± 0.9	82.0 ± 1.1	85.5 ± 0.9	84.3 ± 1.3
AwA2 ($cp\frac{2}{5}$)	81.4 ± 1.7	90.0 ± 0.7	94.7 ± 0.4	93.5 ± 0.5	93.7 ± 0.4	93.7 ± 0.5	94.1 ± 0.4
AwA2 ($cp\frac{3}{5}$)	88.6 ± 1.1	92.3 ± 1.0	96.7 ± 0.3	95.5 ± 0.5	95.4 ± 0.3	95.9 ± 0.3	96.3 ± 0.2
AwA2 ($cp\frac{4}{5}$)	93.7 ± 0.9	94.2 ± 0.6	96.8 ± 0.2	93.8 ± 0.8	96.8 ± 0.2	97.0 ± 0.3	96.8 ± 0.2
AwA2 ($cp\frac{5}{5}$)	97.3 ± 0.9	97.6 ± 0.6	99.0 ± 0.2	96.3 ± 0.7	97.5 ± 0.3	98.3 ± 0.3	98.8 ± 0.2

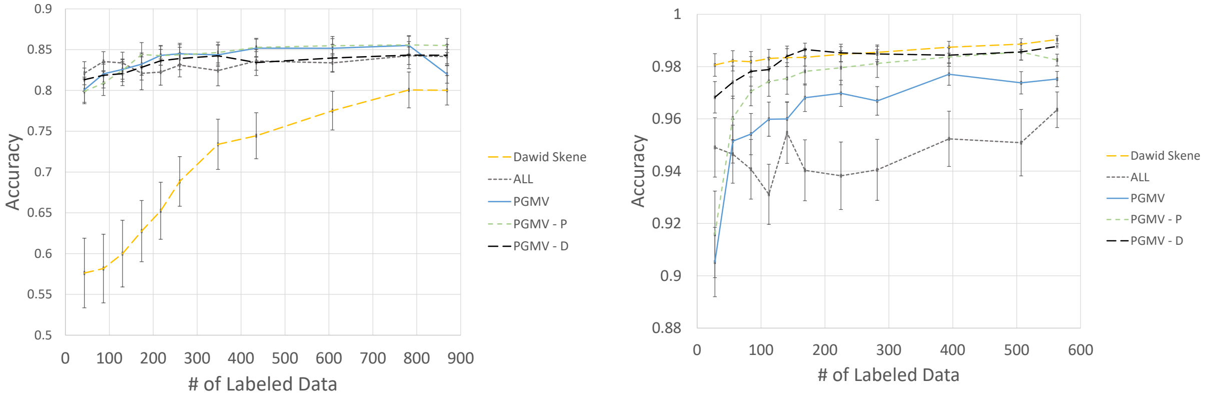

Figure 1: Comparison of our algorithms and other existing methods when varying the amount of labeled data of the AwA2 dataset. The left graph is averaged over the first group of AwA2 tasks when sorted by committee potential less, while the right graph is averaged over the fifth group. Accuracies are computed over 3 splits of labeled and unlabeled data, and the error bars are the standard error. The rightmost point is the values in Table 1 and is averaged over 5 seeds.

estimator (Dawid and Skene, 1979) is a standard crowdsourcing method to learn a weighting for each of the weak labelers. However, this approach makes the independence assumption, so the weighting may not be accurate in dependent cases. This is also the default aggregation method in the Snorkel system (Ratner et al., 2017). We use a semi-supervised version, so the labeled training data available is also used for learning.

Adversarial Label Learning (ALL): Adversarial Label Learning (Arachie and Huang, 2019) is a weakly supervised learning approach that trains a model in an adversarial fashion. This process is similar to our work since it uses bounds on the accuracies of weak supervision sources to constrain the solution space of the adversary. In our experiments, ALL trains a one-layer neural network on the outputs of the weak lablers, which is a more complex hypothesis class than what our methods consider (majority vote). Note that ALL does not provide any theoretical guarantees about its performance or the termination of its training process.

5.2 Tasks

Animals with Attributes 2 (Xian et al., 2018) is a common benchmark for zero-shot learning, which we refer to as AwA2. It consists of 37,322 images of 50 animals classes that are split into 40 seen and 10 unseen classes. Each animal class is annotated with a feature representation consisting of 85 attributes.

We perform binary classification on each of the pairs of unseen classes to create 45 tasks. For all of our 45 image classification experiments, we split our data into train and test data with an even 50-50 split. We use the train data to evaluate the accuracies of our weak labelers, and use the weak labelers' outputs on the test data to select our model and to perform evaluation. We group the 45 different tasks by the quality of the weak supervision sources, which we measure by committee potential (Berend and Kontorovich, 2015). High committee potentials correspond to more potential improvement from aggregation if the independence assumption is true, and heuristically captures the difficulty of the problem by looking at labeler accuracies as a group. We sort the tasks by increasing committee potential and group them into 5 equally sized bins of 9 tasks to report our results. The bins contain ranges of committee potential scores of [1, 5.5], [6.5, 12], [12, 16.5], [18, 24.5], and [25, 61] respectively.

5.3 Creating Weak Supervision Sources

To create weak supervision sources for our various classification tasks, the seen classes are used to train attribute detectors. These classifiers try to detect attributes like stripes, flippers, quadruped, etc. Each detector is a pre-trained ResNet-18 (He et al., 2016) with two fine-tuned linear layers. We perform classification on the unseen classes using the detected attributes. These attribute detectors must transfer high-level concepts of attributes from seen classes to unseen classes. For example, one tries to transfer the knowledge of humpback whales and other seen classes having flippers to *different* classes such as seals having flippers.

5.4 Varying Amounts of Labeled Data

We also perform experiments on varying amounts of labeled data. When labeled data is very limited, there is a greater chance of having bad estimates of weak labeler accuracies. For our method, when weak labeler accuracies are very inaccurate, the constraints on the linear program are sometimes not satisfied for different subsets of labelers. In these cases, the worst case bound cannot be computed for a subset of weak labelers, so our algorithm ignores these subsets.

Again, we report our results as groups of AwA2 pairs by committee potential, as in Table 1 above. We omit the MV and MV Flip results on our figures as their performance remains relatively constant with more labeled data and are almost uniformly beaten by our methods, DS, and ALL. Figure 1 contains the performance of our methods on the first and fifth groups of AwA2 tasks, as we increase the amount of labeled data to make our labeler estimates. We include the graphs for the other groups in the appendix; these other groups illustrate similar results but contain less extreme committee potential values.

On tasks with inaccurate labelers, which is represented by the left graph, our methods outperform the semi-supervised DS baseline, achieving 15 percentage points higher on average over each amount of labeled data. On tasks with accurate labelers, which is captured by the right graph, our approaches are within a half percentage point of DS and outperform ALL by 3 percentage points, when averaged over all ranges of labeled data.

6 CONCLUSION

Our work provides theoretical guarantees for learning to combine weak labelers in an inductive setting, without placing any assumptions on the distribution of the labelers' errors, such as independence. We devise a linear program based approach to analytically compute a worst-case error bound of a set of labelers' majority vote given their accuracies and the distribution of their agreements. We provide greedy algorithms to efficiently scale our approach to larger subsets of labelers. Our experiments show that our methods match or outperform alternative approaches, while providing worst-case error bounds on the majority vote of labelers for weak supervision.

Acknowledgements

The authors would like to thank Cyrus Cousins, Michael Littman, and James Tompkin for helpful discussions. This material is based on research sponsored by Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) under agreement number FA8750-19-2-1006, and by the National Science Foundation (NSF) under award IIS-1813444. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) or the U.S. Government. We gratefully acknowledge support from Google and Cisco. This work was partially supported by the Randy F. Pausch '82 Computer Science Undergraduate Summer Research Award, funded by Peter Norvig '78. Disclosure: Stephen Bach is an advisor to Snorkel AI, a company that provides software and services for weakly supervised machine learning.

References

Arachie, C. and Huang, B. (2019). Adversarial label learning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Bach, S. H., He, B., Ratner, A., and Ré, C. (2017). Learning the structure of generative models without labeled data. In *International Conference on Machine Learning (ICML)*.

Balsubramani, A. and Freund, Y. (2015a). Optimally combining classifiers using unlabeled data. In *Conference on Learning Theory (COLT)*, pages 211–225.

Balsubramani, A. and Freund, Y. (2015b). Scalable semi-supervised aggregation of classifiers. In *Neural Information Processing Systems (NeurIPS)*.

Balsubramani, A. and Freund, Y. (2016). Optimal binary classifier aggregation for general losses. In *Neural Information Processing Systems (NeurIPS)*.

Berend, D. and Kontorovich, A. (2015). A finite sample analysis of the naive bayes classifier. *Journal of Machine Learning Research*, 16(44):1519–1545.

Dalvi, N., Dasgupta, A., Kumar, R., and Rastogi, V. (2013). Aggregating crowdsourced binary ratings. WWW '13, page 285–294.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society C*, 28(1):20–28.

Gao, C. and Zhou, D. (2013). Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. *CoRR*, abs/1207.0016.

Ghosh, A., Kale, S., and McAfee, P. (2011). Who moderates the moderators? crowdsourcing abuse detection in user-generated content. EC '11, page 167–176.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Karger, D. R., Oh, S., and Shah, D. (2014). Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24.

Nitzan, S. and Paroush, J. (1982). Optimal decision rules in uncertain dichotomous choice situations. *International Economic Review*, 23(2):289–97.

Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3):269–282.

Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. In *Neural Information Processing Systems (NeurIPS)*.

Varma, P., Sala, F., He, A., Ratner, A., and Ré, C. (2019). Learning dependency structures for weak supervision models. In *International Conference on Machine Learning (ICML)*.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. (2018). Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Zhang, C. and Ma, Y. (2012). *Ensemble machine learning: methods and applications*. Springer.

Zhang, Y., Chen, X., Zhou, D., and Jordan, M. I. (2016). Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *The Journal of Machine Learning Research*, 17(1):3537–3580.