Journal of Machine Learning Research 22 (2021) 1-33 Submitted 5/20; Revised 4/21; Published 4/21

A General Framework for Adversarial Label Learning

Chidubem Arachie ACHID17@VT.EDU
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061, USA

Bert Huang BERT@CS.TUFTS.EDU
Department of Computer Science
Data Intensive Studies Center
Tufts University
Medford, MA 02155, USA

Editor: David Sontag

Abstract

We consider the task of training classifiers without fully labeled data. We propose a weakly supervised method—adversarial label learning—that trains classifiers to perform well when noisy and possibly correlated labels are provided. Our framework allows users to provide different weak labels and multiple constraints on these labels. Our model then attempts to learn parameters for the data by solving a zero-sum game for the binary problems and a non-zero sum game optimization for multi-class problems. The game is between an adversary that chooses labels for the data and a model that minimizes the error made by the adversarial labels. The weak supervision constrains what labels the adversary can choose. The method therefore minimizes an upper bound of the classifier's error rate using projected primal-dual subgradient descent. Minimizing this bound protects against bias and dependencies in the weak supervision. We first show the performance of our framework on binary classification tasks then we extend our algorithm to show its performance on multiclass datasets. Our experiments show that our method can train without labels and outperforms other approaches for weakly supervised learning.

Keywords: Weak Supervision, Adversarial Learning, Unsupervised Learning, Constraint Learning, Lagrangian Optimization

1. Introduction

Recent success of deep learning has seen an explosion in interest towards building large-scale models for various applications. Training deep models often involves using massive amounts of training data whose labels are not easily obtained or available. Collecting labeled data for training these large-scale models is a major bottleneck since these labels are usually provided by expert annotators and can be expensive to gather.

Weak supervision offers an alternative for training machine learning models without labels because it relies on approximate labels that are easily obtained. Weakly supervised learning alleviates some of the difficulties and cost associated with supervised learning by only requiring annotators to provide rules or approximate indicators that automatically label the data. It uses domain knowledge about the specific problem, side information, or

heuristics to approximate the true labels. Because annotators provide functions or rules that noisily label data, these functions can be applied to an indefinite amount of training data. The intent is that the cognitive cost of the annotator designing such rules may be slightly more than labeling individual examples, but the payoff is unbounded given an abundant source of unlabeled examples.

A key challenge for weak supervision is the fact that there may be bias in the errors made by the weak supervision signals. Using multiple sources of weak supervision can somewhat alleviate this concern, but dependencies among these weak supervision functions can be misconstrued as independent confirmation of erroneous labels. For example, in a classification task to identify diabetic patients, physicians know that obesity can indicate diabetes, and they also know the rate at which this indicator is wrong. However, since the indicator is biased, models trained with this information will learn to detect obesity, not the original goal of diabetes. To correct this problem, one may also consider high blood pressure as a second weak indicator. Unfortunately, these indicators are correlated and may make dependent errors.

This paper introduces *adversarial label learning* (ALL), a method for training classifiers without labels by making use of weak supervision. ALL works by training classifiers to perform well on adversarially labeled instances that are consistent with the weak supervision. ALL aims to mitigate the problem of dependencies between weak signals by adversarially labeling the data. The adversarial labeling can construct scenarios where dependencies in the weak supervision are as confounding as possible while preserving the partial correctness of the weak supervision. The learner then trains a model that can perform well against this adversarial labeling. ALL solves these two competing optimizations using primal-dual subgradient descent. The inner optimization finds a worst-case distribution of the labels for the current weight parameter of the model, while the outer optimization finds the best weights for the model for the current label distribution. The inner optimization's maximized error rate can also be viewed as an upper bound on the true error rate, which the outer optimization aims to minimize. By training to perform well on the worst-case labeling, ALL is robust against dependent and biased errors in weak supervision signals.

The inputs to ALL are a set of unlabeled data examples, a set of weak supervision signals that approximately label the data, and a corresponding set of estimated error bounds on these weak supervision signals. Domain experts can design the weak supervision signals—e.g., by defining approximate labeling rules—and they can use their knowledge to set bounds on the errors of these signals. When designing weak supervision signals, experts often have mental estimates of how noisy the signals are, so this error estimate is an inexpensive yet valuable input for the learning algorithm.

First, we consider a binary classification setting where a parameterized model is trained to classify the data by solving a competitive game against an adversary. We make use of multiple weak signals that represent different approximations of the true model. These weak signals can be interpreted as having different views of the data. The estimated error rates of these weak signals are passed as constraints to our optimization. Importantly, we show that ALL works in cases where these weak signals make dependent errors. Our experiments also show that ALL trains classifiers that are better than the weak supervision signals, even when the error estimates are incorrect. The performance of ALL in this setting is significant

because domain experts will often imperfectly estimate the noisiness of the weak supervision signals.

We then extend ALL to solve multiclass problems using additional linear constraints. We develop *Multi-ALL*, a general framework that encodes multiple linear constraints on the weak supervision signals. Unlike in the binary classification task, Multi-ALL learns from the weak supervision by solving a non-zero sum game between an adversary and the model. By using a non-zero sum game, our formulation provides more flexibility than in the binary case and allows for different loss functions to be used. Thus, we enable learning for other forms such as multiclass and multilabel classification. Multi-ALL is stochastic and uses different forms of weak supervision making it possible to train large scale models like deep neural networks.

We validate Multi-ALL on three image classification datasets. We use error and precision constraints to solve multiclass image classification tasks for deep neural network models. In the experiments, we provide weak supervision signals that are both generated by humans and programmatically generated. Our results show that our approach outperforms other weak supervision methods on deep image classification tasks. Our experiments also highlight the difficulties in providing adequate weak supervision signals for solving multiclass image classification tasks.

2. Related Work

Our work builds on progress in three topic areas: weak supervision, learning with constraints, and adversarial learning.

2.1 Weak Supervision

We expand on some of the recent advances on weakly supervised learning, which is the paradigm where models are trained using large amounts of unlabeled data and low-cost, often noisy annotation. One important recent contribution is the Snorkel system (Ratner et al., 2017, 2016; Bach et al., 2019), a weak supervision approach where annotators design different labeling functions that are applied to the unlabeled data to create noisy labels. The noisy labels are combined using a generative model to learn the correlation and dependencies between the noisy signals. Snorkel then reasons with this generative model to produce probabilistic labels for the training data. Our method is related to this approach in that we use noisy labels, or weak signals, to learn adversarial labels for the training data, but our focus is on model training rather than outputting training labels for the unlabeled data. Nevertheless, we show in experiments that the quality of labels learned using Multi-ALL compares favorably to that of labels inferred by Snorkel's generative modeling.

Another form of weak supervision is crowdsourcing. A paradigm where non-expert annotators provide labels for data. Crowdsourcing has become relevant to machine learning practitioners as it provide a means to train machine learning models using labels collected from different crowd workers (Carpenter, 2008; Raykar et al., 2010; Gao et al., 2011; Karger et al., 2011; Khetan et al., 2017; Wang and Poon, 2018; Liu et al., 2012; Platanios et al., 2020; Zhou et al., 2015; Zhou and He, 2016). The key machine learning challenge when crowdsourcing is to effectively combine the different labels obtained from human annotators. Our work is similar in that we try to combine different weak labels. However, unlike most methods for crowdsourcing, we cannot assume that the label errors are independent of each

other. Instead, we train the model to learn while accounting for dependencies between the various weak supervision signals.

Ensemble methods such as boosting Schapire et al. (2002) combine different weak learners (low-cost, low-powered classifiers) to create classifiers that outperform the various weak learners. These weak learners are not weak in the same sense as weak supervision. These strategies are defined for fully supervised settings. Although recent work has proposed leveraging unlabeled data to improve the accuracies of boosting methods Balsubramani and Freund (2015b), our settings differs since we do not expect to have access to labeled data.

Weakly supervised methods have enabled knowledge extraction from the Web (Bunescu and Mooney, 2007; Mintz et al., 2009; Riedel et al., 2010; Yao et al., 2010; Hoffmann et al., 2011), visual image segmentation (Chen et al., 2014; Xu et al., 2014), and tagging of medical conditions from health records (Fries et al., 2019; Halpern et al., 2016).

This paper integrates our previously published conference paper (Arachie and Huang, 2019b) and a follow-up technical report (Arachie and Huang, 2019a) into a single unified publication describing our overarching framework for adversarial learning with weak supervision.

2.2 Learning with Constraints

Our framework incorporates error constraints that are reminiscent of boosting (Schapire et al., 2002); however, our bounds are more general and allow for other forms of constraints like precision. Our work is also related to techniques for estimating accuracies of classifiers using only unlabeled data (Blum and Mitchell, 1998; Jaffe et al., 2016; Platanios et al., 2014; Steinhardt and Liang, 2016; Dawid and Skene, 1979) and combining classifiers for transductive learning using unlabeled data (Balsubramani and Freund, 2015a,b).

Other methods like posterior regularization (PR) (Ganchev et al., 2010) and generalized expectation (GE) criteria (Druck et al., 2008; Mann and McCallum, 2010, 2008) have been developed to incorporate human knowledge or side information into an objective function. A GE criterion (McCallum et al., 2007) is a term in a parameter estimation objective function that prefers models to match conditional probabilities provided as weak supervision. These conditional probabilities may take the form of the probability of labels given a feature (Druck et al., 2008), also allowing the weak supervision to include information about the uncertainty of a weak signal. Posterior regularization (PR) (Ganchev et al., 2010) is a similar approach that trains models to adhere to constraints on their output posterior distributions. These constraints can also take the form of weak supervision signals that specify the class of allowable posterior distributions for the learned model. While GE and PR allow incorporation of weak supervision and quantification of weak signal errors, they do not explicitly consider that these weak signals may make errors that conspire to confound the learner. Our development of ALL aims to address this shortcoming.

2.3 Adversarial Learning and Games

Researchers have been increasingly interested in adversarial learning (Lowd and Meek, 2005) as a method for training models that are robust to input perturbations of the data. These methods (Shafahi et al., 2019; Miyato et al., 2018; Shrivastava et al., 2017; Torkamani and Lowd, 2013, 2014) regularize the learned model using different techniques to defend

against adversarial attacks with an added benefit of improved generalization guarantees. Our approach focuses on adversarial manipulation of the output labels to combat redundancy among multiple sources of weak supervision.

Game analyses are gaining importance in machine learning because they generalize optimization frameworks by assigning different objective functions for different players or optimizing agents. The generative adversarial network (GAN) (Goodfellow et al., 2014) framework sets up a two player game between a generator and a discriminator, with the aim of learning realistic data distributions for the generator. Our method does not learn a generative model but instead sets up a two-player game between an adversary that assigns labels for the classification task and a model that trains parameters to minimize a cross-entropy loss with respect to the adversarial labels. Researchers have recently examined convergence properties of smooth games (Gidel et al., 2018; Zhang and Yu, 2020) and provided theoretical guarantees for them.

3. Adversarial Label Learning

The principle behind adversarial label learning (ALL) is that we train a model to perform well under the worst possible conditions. The conditions being considered are the possible labels of the training data. We consider the setting in which the learner has access to a training set of examples, and weak supervision is given in the form of some approximate indicators of the target classification along with expert estimates of the error rates of these indicators. Formally, let the data be $X = [x_1, \ldots, x_n]$. (We consider these examples to be ordered for notational convenience, but the order does not matter.) These examples belong to classes $[y_1, \ldots, y_n] \in \{0, 1\}^n$. The training labels $\boldsymbol{y}$ are unavailable to the learner. Instead, the learner has access to m weak supervision signals $\{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m\}$, where each weak signal is a soft labeling of the data, i.e., $\boldsymbol{q}_i \in [0, 1]^n$. These soft labelings are estimated probabilities that the example is in the positive class. In conjunction with the weak signals, the learner also receives estimated expected error rate bounds of the weak signals $\boldsymbol{b} = [b_1, \ldots, b_m]$. These values bound the expected error of the weak signals, i.e.,

$$b_i \geq \mathbb{E}_{\hat{\boldsymbol{y}} \sim \boldsymbol{q}_i} \left[\tfrac{1}{n} \textstyle\sum_{j=1}^{n} [\hat{y}_j \neq y_j] \right] \ , \tag{1}$$

which can be equivalently expressed as

$$b_i \geq \tfrac{1}{n} \left(\boldsymbol{q}_i^\top (1 - \boldsymbol{y}) + (1 - \boldsymbol{q}_i)^\top \boldsymbol{y} \right) \ . \tag{2}$$

While the learned classifier does not have access to the true labels $\boldsymbol{y}$, it will use the assumption that this bound holds to define the space of possible labelings. Let the current estimates of learned label probabilities be $\boldsymbol{p} \in [0, 1]^n$. We relax the space of discrete labelings to the space of independent probabilistic labels, such that the value $\hat{y}_j \in [0, 1]$ represents the probability that the true label y_j of example x_j is positive. The adversarial labeling then is the vector of class probabilities $\hat{\boldsymbol{y}}$ that maximizes the expected error rate of the learned probabilities subject to the constraints given by the weak supervision signals and bounds,

which can be found by solving the following linear program:

$$\begin{aligned} \underset{\hat{\boldsymbol{y}} \in [0,1]^n}{\arg\max} \quad & \tfrac{1}{n}\left(\boldsymbol{p}^\top(1-\hat{\boldsymbol{y}}) + (1-\boldsymbol{p})^\top \hat{\boldsymbol{y}}\right) \\ \text{s.t.} \quad & b_i \geq \tfrac{1}{n}\left(\boldsymbol{q}_i^\top(1-\hat{\boldsymbol{y}}) + (1-\boldsymbol{q}_i)^\top \hat{\boldsymbol{y}}\right),\ \forall i \in \{1,\ldots,m\}\ , \end{aligned} \tag{3}$$

which we present in this unsimplified form to convey the intuition behind its objective and constraints; some algebra simplifies this optimization into a more standard form.

The adversarial labeling described so far is a key component of the learning algorithm. ALL trains a parameterized prediction function f_θ that reads the data as input and outputs estimated class probabilities, i.e., $[f_\theta(x_j)]_{j=1}^n = \boldsymbol{p}$. We will write $\boldsymbol{p}(\theta)$ to mean $[f_\theta(x_j)]_{j=1}^n$ when it is important to note that these are generated from the parameterized function f. For now, we assume a general form for this parameterized function. For our optimization method described later in Section 3.2, we assume that the function f is sub-differentiable with respect to its parameters θ. The goal of learning is then to minimize the expected error relative to the adversarial labeling. This principle leads to the following saddle-point optimization:

$$\begin{aligned} \min_{\theta} \quad \max_{\hat{\boldsymbol{y}} \in [0,1]^n} \quad & \tfrac{1}{n}\left(\boldsymbol{p}(\theta)^\top(1-\hat{\boldsymbol{y}}) + (1-\boldsymbol{p}(\theta))^\top \hat{\boldsymbol{y}}\right) \\ \text{s.t.} \quad & b_i \geq \tfrac{1}{n}\left(\boldsymbol{q}_i^\top(1-\hat{\boldsymbol{y}}) + (1-\boldsymbol{q}_i)^\top \hat{\boldsymbol{y}}\right),\ \forall i \in \{1,\ldots,m\}\ . \end{aligned} \tag{4}$$

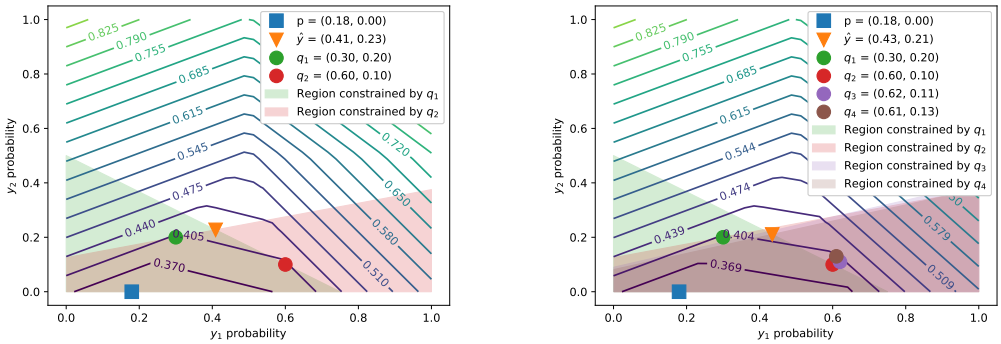
We can view the outer optimization as optimizing a primal objective that is the maximum of the constrained inner optimization. Define this primal function as $g(\theta)$, such that Eq. (4) can be equivalently written as $\min_\theta g(\theta)$. If the weak supervision error bounds are true, *this primal objective value is an upper bound on the true error rate.* This fact can be proven by considering that the true labels $\boldsymbol{y}$ satisfy the constraints, and the inner optimization seeks a labeling $\hat{\boldsymbol{y}}$ that maximizes the classifier's expected error rate. In the next section, we visualize this primal function and the behavior of adversarial labeling before describing how we efficiently solve this optimization in Section 3.2.

3.1 Visualizing Adversarial Label Learning

In this section, we investigate a simple case that illustrates the behavior of the primal objective function g on a two-example dataset ($n = 2$). For a small dataset, we can visualize in two dimensions a variety of concepts.

In Fig. 1a, we illustrate the constraints set by the two weak supervision signals. The first signal $\boldsymbol{q}_1$ estimates that $\hat{y}_1$ is positive with probability 0.3 and that $\hat{y}_2$ is positive with probability 0.2. The second signal $\boldsymbol{q}_2$ estimates that $\hat{y}_1$ is positive with probability 0.6 and that $\hat{y}_2$ is positive with probability 0.1. The bounds for each weak signal error are set to $b_1 = b_2 = 0.4$. Note that both weak signals agree that $\hat{y}_2$ is most likely negative, but they disagree on whether $\hat{y}_1$ is more likely to be positive or negative.

Constraints on $\hat{\boldsymbol{y}}$ The shaded regions represent the feasible regions determined by the linear constraint corresponding to each weak signal. The intersection of these feasible regions is the search space for label vectors. Note how the pink region determined by $\boldsymbol{q}_2$ allows $\hat{y}_1$ to be either extreme of 0 or 1. With more examples ($n \gg 2$), the possibility of ambiguous labels increases significantly.



(a) Two weak signals

(b) Redundant weak signals

Figure 1: Illustrations of the primal objective function from Eq. (4), the constraints set by the weak supervision, and the optimal learned probabilities and adversarial labels for a two-example problem.

Primal Objective Function The contour lines illustrate the objective value of the primal function g, which finds the expected error for the adversarially set labels $\hat{\boldsymbol{y}}$. Since the adversarial inner optimization is a linear program, the solution jumps between vertices of the constrained polytope, making the primal expected error a piecewise linear convex function of $\boldsymbol{p}$.

Adversarial Labeling In Fig. 1a, the blue square is the minimum of the primal function, i.e., the solution to the ALL objective. This solution shows that the ideal learned model should predict $\hat{y}_1$ to be positive with probability 0.18 and $\hat{y}_2$ to be positive with probability 0. In the optimal state, the adversarial labeling of the examples is illustrated as the orange triangle at $(0.41, 0.23)$, i.e., the label probability vector that induces the most error for the current predicted probabilities $\boldsymbol{p}$ that still satisfies the constraints set by $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$.

Robustness to Redundant and Dependent Errors A key feature of ALL is that it is robust to redundant and dependent errors in the weak supervision. In Fig. 1b, we plot a variation of the setup from Fig. 1a, except we include two noisy copies of weak signal $\boldsymbol{q}_2$. Since our optimal solution disagreed with weak signal $\boldsymbol{q}_2$ on the most likely label for $\hat{y}_1$, one might expect that adding more weak signals that agree with $\boldsymbol{q}_2$ would "outvote" the solution and pull it to a higher probability of $\hat{y}_1$ being positive. But if weak signal $\boldsymbol{q}_2$ is highly correlated with weak signals $\boldsymbol{q}_3$ and $\boldsymbol{q}_4$, they may suffer from the same errors. Instead of these extra signals inducing a majority vote behavior on the solution, their effect on ALL is that they slightly change the feasible region of the adversarial labels, which leaves the optimum unchanged.

These two-dimensional visualizations illustrate the behavior of ALL on a simple input. In higher dimensions, i.e., when there are more examples in the training set, there is more freedom in the constraints set by each weak signal, so there will be more facets to the piecewise linear objective.

3.2 Optimization Approach

We use projected primal-dual updates for an augmented Lagrangian relaxation to efficiently optimize the learning objective. The advantage of this approach is that it allows inexpensive updates for all variables being optimized over, and it allows learning to occur without waiting for the solution of the inner optimization. The augmented Lagrangian form of the objective is

$$\begin{aligned} L(\theta, \hat{\boldsymbol{y}}, \boldsymbol{\gamma}) = \frac{1}{n}\left(\boldsymbol{p}(\theta)^\top(1-\hat{\boldsymbol{y}}) + (1-\boldsymbol{p}(\theta))^\top\hat{\boldsymbol{y}}\right) \\ - \sum_{i=1}^{m}\gamma_i\left(\boldsymbol{q}_i^\top(1-\hat{\boldsymbol{y}}) + (1-\boldsymbol{q}_i)^\top\hat{\boldsymbol{y}} - nb_i\right) \\ - \frac{\rho}{2}\sum_{i=1}^{m}\left\|\left[\boldsymbol{q}_i^\top(1-\hat{\boldsymbol{y}}) + (1-\boldsymbol{q}_i)^\top\hat{\boldsymbol{y}} - nb_i\right]_+\right\|_2^2 , \end{aligned} \tag{5}$$

where $[\ \cdot\ ]_+$ is the hinge function that returns its input if positive and zero otherwise. This form uses Karush-Kuhn-Tucker (KKT) multipliers to relax the linear constraints on $\hat{\boldsymbol{y}}$ and a squared augmented penalty term on the constraint violation.

We then take projected gradient steps to update the variables θ, $\hat{\boldsymbol{y}}$, and $\boldsymbol{\gamma}$. The update step for the parameters is

$$\theta \leftarrow \theta - \frac{\alpha_t}{n}\left(\frac{\partial \boldsymbol{p}}{\partial \theta}\right)^\top(1-2\hat{\boldsymbol{y}}) \ , \tag{6}$$

where $\left(\frac{\partial \boldsymbol{p}}{\partial \theta}\right)$ is the Jacobian matrix for the classifier f over the full dataset and α_t is a gradient step size that can decrease over time. This Jacobian can be computed for a variety of models by back-propagating through the classification computation. The update for the adversarial labels is

$$\hat{\boldsymbol{y}} \leftarrow \left[\hat{\boldsymbol{y}} + \alpha_t\left(\frac{1}{n}(1-2\boldsymbol{p}(\theta)) + \sum_{i=1}^{m}\left(\gamma_i(1-2\boldsymbol{q}_i) - \boldsymbol{z}_i\ \right)\right)\right]_0^1 , \tag{7}$$

where

$$\boldsymbol{z}_i = \rho(1-2\boldsymbol{q}_i)\left[\boldsymbol{q}_i^\top(1-\hat{\boldsymbol{y}}) + (1-\boldsymbol{q}_i)^\top\hat{\boldsymbol{y}} - nb_i\right]_+ \ ,$$

and $[\ \cdot\ ]_0^1$ clips the label vector to the space $[0,1]^n$, projecting it into its domain. The update for each KKT multiplier is

$$\gamma_i \leftarrow \left[\gamma_i - \rho\left(\boldsymbol{q}_i^\top(1-\hat{\boldsymbol{y}}) + (1-\boldsymbol{q}_i)^\top\hat{\boldsymbol{y}} - nb_i\right)\right]_+ \ , \tag{8}$$

which is clipped to be non-negative and uses a fixed step size ρ as dictated by the augmented Lagrangian method (Hestenes, 1969). These primal-dual updates for the optimization converge in our experiments. Though L is not convex with respect to θ, it does satisfy some of the necessary conditions for convergence derived by Du and Hu (2018): The objective L is strongly convex in $\boldsymbol{p}$ and γ and concave in $\hat{\boldsymbol{y}}$, while the penalty term for the augmented Lagrangian is strongly convex. These properties may explain its convergence in practice. The full algorithm is summarized in Algorithm 1.

Algorithm 1 Adversarial Label Learning

Require: Dataset $X = [x_1, \ldots, x_n]$, learning rate schedule $\boldsymbol{\alpha}$, weak signals and bounds $[(\boldsymbol{q}_1, b_1), \ldots, (\boldsymbol{q}_m, b_m)]$, augmented Lagrangian parameter ρ.

```
Initialize θ (e.g., random, zeros, etc.)
Initialize ŷ ∈ [0, 1]^n (e.g., average of q_1, ..., q_m)
Initialize γ ∈ R^m_{≥0} (e.g., zeros)
while not converged do
    Update θ with Equation (6)
    Update p with model and θ
    Update ŷ with Equation (7)
    Update γ with Equation (8)
end while
return model parameters θ
```

4. Experiments

We test adversarial label learning on a variety of datasets, comparing it with other approaches for weak supervision. In this section, we describe how we simulate domain expertise to generate weak supervision signals. We then describe the datasets we evaluated with and the compared weak supervision approaches, and we analyze the results of the experiments.

4.1 Simulating Weak Supervision

In practice, domain experts provide weak supervision in the form of noisy indicators or simple labeling functions. This weak supervision generates probabilities that the examples in a sample of the data belong to the positive class. Since we do not have explicit domain knowledge for the datasets used in our experiments, we generate the weak signals by training simple, one-dimensional classifiers on subsets of the data. The subset of the data used to train the weak supervision models is referred to as weak supervision data. We train each one-dimensional weak supervision model by selecting a feature and training a one-dimensional logistic regression model using only that feature. We select the weak supervision features based on our non-expert understanding of which features could reasonably serve as indicators of the target class. For datasets whose feature descriptions are not provided, we train the weak supervision models using the first feature, middle feature, and last feature. For the Fashion-MNIST, dataset we used the pixel value at the one-quarter, center, and three-quarter locations along the vertical center line (see Fig. 2) to build the respective weak supervision models.

We evaluate one-dimensional classifiers on the training subset, generating the weak signals $\{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m\}$. In our first set of experiments, we measure the true error rate of each weak signal on the training subset and use that as the error bounds $\{b_1, \ldots, b_m\}$. In later experiments, we set all bounds to 0.3 as an arbitrary guess. We train weak signals from one-dimensional inputs to create realistically noisy weak signals. Training on more features could increase the predictive accuracy of the weak signals and by extension ALL, but such high-fidelity weak signals may be rare in practice. Alternatively, we chose not to hand-design weak supervision signals and bounds, because doing so could inject our own

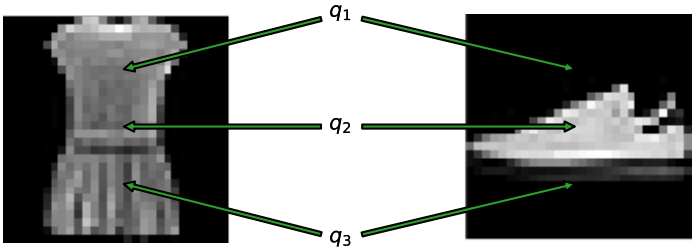


Figure 2: Features used to generate weak supervision signals on Fashion-MNIST data.

bias into this evaluation. Simulating domain expertise with a small training set provides a neutral evaluation.

4.2 Compared Methods

We compare ALL against two baseline models: a modified generalized expectation (GE) method and averaging of weak signals (AVG).

Modified GE GE assigns a score to the value of a model expectation. Given a conditional model distribution and a reference distribution, GE uses a score function to measure the distance between the model expectation and reference expectation. We define a modified GE method to use the label distribution conditioned on each weak signal, i.e.,

$$\hat{p}_\theta\left(\boldsymbol{y}|\boldsymbol{q}_k \geq 0.5\right) = \mathbb{E}_{\hat{\boldsymbol{y}}}\left[\tfrac{1}{C_k} I(\hat{\boldsymbol{y}}) I\left(\boldsymbol{q}_k \geq 0.5\right)\right] \ , \tag{9}$$

and the reference expectation is

$$\tilde{p}\left(\boldsymbol{y}|\boldsymbol{q}_k \geq 0.5\right) = \mathbb{E}_{\boldsymbol{y}}\left[\tfrac{1}{C_k} I(\boldsymbol{y}) I\left(\boldsymbol{q}_k \geq 0.5\right)\right] \ , \tag{10}$$

where $\hat{\boldsymbol{y}}$ is the predicted labels and $C_k = \sum_{\boldsymbol{q}_k} I\left(\boldsymbol{q}_k \geq 0.5\right)$ is a normalizing constant and I is an indicator function. We compute these reference distributions on the training subset of the data. Our modified GE objective is then

$$\begin{aligned} \sum_{k=1}^{m} KL\left[\tilde{p}\left(\boldsymbol{y}|\boldsymbol{q}_k \geq 0.5\right) \| \hat{p_\theta}\left(\boldsymbol{y}|\boldsymbol{q}_k \geq 0.5\right)\right] + \\ KL\left[\tilde{p}\left(\boldsymbol{y}|\boldsymbol{q}_k < 0.5\right) \| \hat{p_\theta}\left(\boldsymbol{y}|\boldsymbol{q}_k < 0.5\right)\right] \ . \end{aligned} \tag{11}$$

We regularize this objective with an L2 penalty. This modified GE method is able to exploit the same information ALL is provided: the weak signals $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m$ and the reference distributions in Eq. 10 are analogous to (though richer than) the error bounds provided to ALL.

Averaging Baseline The input to our weakly supervised learning task includes the weak supervision signals $\boldsymbol{q}$, bounds $\boldsymbol{b}$, and the training set *without labels*. A straightforward approach that a reasonable data scientist could take to this training task is to compute pseudo-labels using the weak signals. Then one can train many classifiers using a standard supervised learning approach. For the averaging method, we generate baseline models by

treating the rounded average of weak signals as a label. The averaging baseline tries to mimic the aggregated weak supervision. The averaging model trains a logistic regression classifier using the average of the weak signals' predictions as labels.

4.3 Experimental Setup

We run experiments on different datasets to measure the generalization and predictive power of adversarial label learning (ALL). Our first set of experiments measures the generalization of ALL on held out test set for different datasets. Then we provide experiments on crowdsourcing datasets, to show ALL's ability to learn predictive labels for the datasets.

For each dataset in the first experiments, we generate weak supervision signals and estimate their error rates. We then compare the accuracy of the model trained by ALL against (1) the modified GE baseline, (2) the different weak supervision signals and, (3) baseline models trained by treating the average of the weak supervision signals as labels. We randomly split each dataset such that 30% is used as weak supervision data, 40% is used as training data, and 30% is used as test data. For our experiments, we use 10 such random splits and report the mean of the results.

In each of our experiments, we consider three different weak signals. We run ALL on the first weak signal (ALL-1), the first and second weak signals (ALL-2), or all three weak signals (ALL-3). We use the sigmoid function as our parameterized function f_θ for estimating class probabilities of ALL and GE, i.e., $[f_\theta(x_j)]_{j=1}^n = 1/(1+\exp(-\theta^T x)) = \boldsymbol{p}_\theta$.

We compare against the accuracy of GE trained using the first weak signal (GE-1), the first and second weak signals (GE-2), or all three weak signals (GE-3). We also compare directly using the individual weak signals as the classifier (WS-1, WS-2, and WS-3). Additionally, we train models to mimic the average of the first weak signal (AVG-1), the first and second weak signals (AVG-2), all three weak signals (AVG-3). We also report the label accuracy of the weighted average of the weak signals (W-AVG). The accuracies of the weak signals are used as their weights. Finally, we show comparison to supervised learning for reference (SPV-L).

4.4 Datasets

We describe the datasets used in the experiments below.

Fashion-MNIST The Fashion-MNIST dataset (Xiao et al., 2017) represents an image-classification task where each example is a 28×28 grayscale image. The images are categorized into 10 classes of clothing types. Each class contains 6,000 training examples and 1,000 test examples. We consider the binary classification between three pairs of classes: dresses/sneakers (DvK), sandals/ankle boots (SvA), and coats/bags (CvB).

Breast Cancer The task in this dataset is to diagnose if the breast cell nuclei are from a malignant (positive) or benign (negative) case of breast cancer (Blake and Merz, 1998; Street et al., 1993). We use the mean radius of the nucleus (WS-1), the radius standard error (WS-2), and worst radius (WS-3) of the cell nucleus as features to train the three different weak supervision models. The dataset contains 569 samples.

OBS Network The classification task for the Burst Header Packet Flooding Attack Detection dataset is to detect network nodes based on their behavior, identifying whether

they should be blocked for potentially malicious behavior (Rajab et al., 2016). We use the percentage of flood per node (WS-1), average packet drop rate (WS-2), and utilized bandwidth (WS-3) as features to train the weak signals. The original dataset contains four classes, so we select the two classes with the most examples, resulting in a total of 795 examples.

Cardiotocography The task for this dataset is to classify fetal heart rate using uterine contraction features on cardiotocograms classified by expert obstetricians (Ayres-de Campos et al., 2000). The original dataset contains 10 classes, we select the most common two classes, resulting in a total of 963 examples. We use accelerations per second (WS-1), mean value of long-term variability (WS-2), and histogram median (WS-3) as features to train the weak signals.

Clave Direction The task for the Firm Teacher Clave Direction dataset is to classify the clave direction from rhythmic patterns (Vurkaç, 2011). The original dataset contains four classes, so we select the two most common classes, resulting in a total of 8,606 examples. We use the first (WS-1), middle (WS-2), and last (WS-3) features to train the weak signals.

Credit Card The Statlog German Credit Card dataset task is to classify people described by a set of attributes as good or bad credit risks (Blake and Merz, 1998). We use the status of an existing checking account (WS-1), installment rate in percentage of disposable income (WS-2), and amount of existing credit at the bank (WS-3) as features to train the weak signals. The dataset contains 1,000 samples.

Statlog Satellite The task of the Statlog dataset is to predict soil class given the multi-spectral values of pixels in 3x3 neighborhoods of satellite images (Blake and Merz, 1998). The original dataset contains seven classes of soil samples, so we select the two most common classes, resulting in a total of 3,041 examples. We use the first (WS-1), middle (WS-2), and last (WS-3) features to train the weak signals.

Phishing Websites The task is to identify phishing websites using different web attributes (Mohammad et al., 2012). The dataset contains 11,055 samples. We use the URL of the anchor (WS-1), web traffic (WS-2), and Google index (WS-3) as features to train the weak signals.

Wine Quality The task is to classify the quality of wine using physiochemical attributes of the wine (Cortez et al., 2009). The original dataset contains seven classes, so we select the two classes with the most examples, resulting in a total of 4974 examples. We use fixed acidity (WS-1), density (WS-2), and pH (WS-3) as features to train the weak signals.

Table 1 shows the mean accuracies obtained by running ALL on the different datasets.

4.5 Learning with True Bounds

Our first experiments allow ALL to use the error bounds computed on the training set. Table 1 shows the accuracies of the models evaluated on the held-out test sets of each task. ALL trains models that perform significantly better than the weak signals and the baselines on the test data. The AVG baselines perform better with an increasing number of weak signals, but their best accuracy score on most datasets is significantly worse than that of

Dataset	ALL-1	ALL-2	ALL-3	GE-1	GE-2	GE-3	AVG-1	AVG-2	AVG-3	W-AVG	SPV-L
Fashion MNIST (DvK)	**0.998**	**0.995**	**0.996**	0.975	0.972	0.977	0.506	0.743	0.834	0.838	1.0
Fashion MNIST (SvA)	**0.923**	**0.922**	**0.924**	0.501	0.500	0.500	0.561	0.568	0.719	0.723	0.972
Fashion MNIST (CvB)	0.795	**0.831**	**0.840**	0.497	0.499	0.500	0.577	0.697	0.740	0.742	0.989
Breast Cancer	**0.942**	**0.944**	**0.945**	**0.936**	**0.936**	**0.935**	0.889	0.885	0.896	0.899	0.977
OBS Network	**0.717**	**0.718**	**0.719**	0.708	0.701	0.698	**0.724**	**0.723**	0.698	0.711	0.735
Cardiotocography	0.803	0.803	0.803	0.824	0.675	0.633	**0.942**	**0.947**	**0.942**	**0.945**	0.945
Clave Direction	0.646	**0.837**	0.746	0.646	0.796	0.772	0.646	0.645	0.707	0.711	0.964
Credit Card	**0.697**	**0.696**	**0.697**	**0.695**	0.460	0.424	0.660	0.662	0.607	0.626	0.719
Statlog Satellite	0.470	0.933	0.936	0.521	**0.987**	**0.992**	0.669	0.926	0.916	0.918	0.999
Phishing Websites	**0.896**	**0.895**	**0.895**	**0.898**	**0.894**	0.870	0.846	0.807	0.846	0.849	0.923
Wine Quality	0.572	**0.662**	0.623	0.455	0.427	0.454	0.570	0.573	0.555	0.582	0.685

Table 1: Test accuracy of ALL and baseline methods on different datasets with different amounts of weak signals. The best performing methods that are not statistically distinguishable using a two-tailed paired t-test ($p = 0.05$) are boldfaced. ALL is more consistently one of the best performing methods compared to generalized expectation (GE) and averaging (AVG). For reference, we include the accuracy for fully supervised learning (SPV-L).

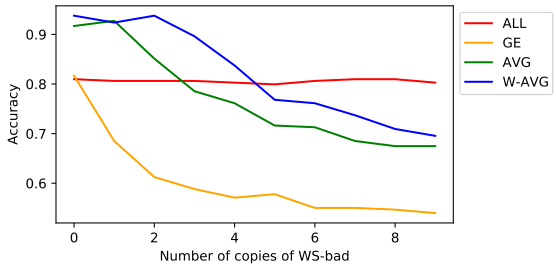


Figure 3: Performance of the methods using one good weak signal and repeated erroneous weak signals. ALL is able to learn a consistent classifier as we add redundant low performing weak signals.

ALL. There is not a big performance difference between AVG and W-AVG which uses the accuracies of the weak signals to compute its labels. ALL trains a robust model and is able to learn using noisy weak signals. Despite the fact that the weak signals on the Fashion MNIST dataset have rather low accuracy, ALL trained with these signals is able to achieve high accuracy. This is partly due to the fact that ALL considers the full features of the data in training the model. Hence, it is able to find similarities among neighboring pixels to make better classification decisions compared to the weak signal that considers only one pixel to generate its noisy label. GE also has access to the data during training, but GE only significantly outperforms ALL on the Statlog Satellite dataset. Nevertheless ALL still achieves a high accuracy score. The main failure case is the cardiotocography task, in which the AVG and W-AVG baselines outperforms both GE and ALL. However, in this task and others, we observe that ALL performs well even when the weak signals make dependent errors, while the baseline methods suffer as more signals with dependent errors are introduced. We study this concept further in the next experiment. Interestingly, we observe that ALL performance is close to that of supervised learning (SPV-L) on majority of the datasets.

Dataset	ALL-1	ALL-2	ALL-3	GE-1	GE-2	GE-3	AVG-1	AVG-2	AVG-3	WS-1	WS-2	WS-3
Fashion MNIST (DvK)	**0.998**	**0.995**	**0.996**	0.975	0.972	0.977	0.506	0.743	0.834	0.508	0.750	0.644
Fashion MNIST (SvA)	**0.895**	0.825	**0.901**	0.501	0.500	0.500	0.561	0.568	0.719	0.562	0.535	0.688
Fashion MNIST (CvB)	**0.810**	**0.805**	**0.802**	0.497	0.499	0.500	0.577	0.697	0.740	0.587	0.684	0.643
Breast Cancer	**0.940**	**0.941**	**0.944**	**0.936**	**0.936**	**0.935**	0.889	0.885	0.896	0.871	0.804	0.915
OBS Network	**0.719**	**0.719**	**0.722**	0.708	0.701	0.698	**0.724**	**0.723**	0.698	**0.721**	0.715	0.692
Cardiotocography	0.805	0.794	0.657	0.824	0.675	0.633	**0.942**	**0.947**	**0.942**	**0.946**	0.602	0.604
Clave Direction	0.646	**0.854**	0.727	0.646	0.796	0.772	0.646	0.645	0.707	0.646	0.648	0.625
Credit Card	**0.696**	0.671	0.610	**0.695**	0.460	0.424	0.660	0.662	0.607	0.659	0.572	0.557
Statlog Satellite	0.493	**0.983**	**0.982**	0.521	**0.987**	**0.992**	0.669	0.926	0.916	0.660	0.775	0.880
Phishing Websites	**0.899**	0.835	0.853	**0.898**	**0.894**	0.870	0.846	0.807	0.846	0.846	0.700	0.585
Wine Quality	0.566	0.603	**0.694**	0.455	0.427	0.454	0.570	0.573	0.555	0.571	0.596	0.570

Table 2: Test accuracy of ALL and baseline models on different datasets using fixed bounds. The best performing methods that are not statistically distinguishable using a two-tailed paired t-test (p = 0.05) are boldfaced. We replicate some baseline results from the previous experiments for convenience; they are unaffected by the change in error bound and additionally show the accuracies of the individual weak signals. We also include in this table the accuracies of each weak signal alone.

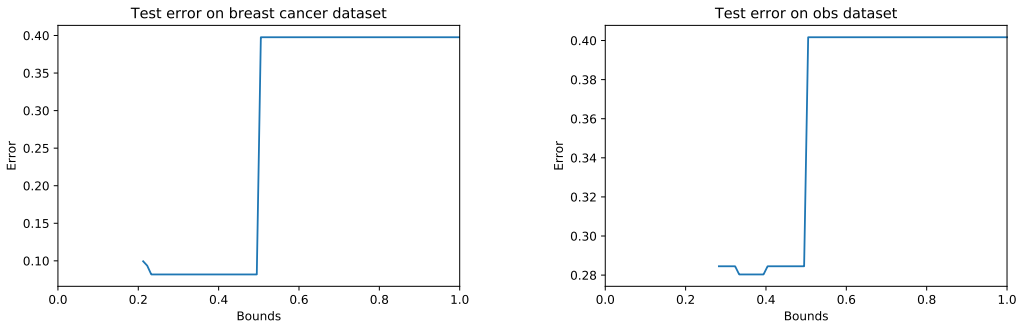


Figure 4: Error of the model (ALL-3) when run with different fixed bounds between 0 and 1. Small bound values make infeasible constraints that prevent convergence, and are not plotted here. Large error rate bounds give too much freedom to the adversary, and cause learning quality to significantly deteriorate. The experiments show that any bounds that imply the weak signals are better than random but admit feasible solutions produce similar quality learning.

4.6 Robustness against Dependent Errors

We observed from our test results that unlike the baselines, ALL learns a robust model that performs well even in the presence of low-quality weak signals. We isolate this concept using two weak signals from the cardiotocography task, a high-quality weak signal (WS-good) and a low-quality weak signal (WS-bad). We consider the scenario where the low-quality signal (WS-bad) is copied multiple times in the weak supervision. We train the models with WS-good and a varying number of copies of WS-bad. We evaluate the performance of the models on each experiment using the test data. Figure 3 plots the accuracy of the models under these settings. In the presence of multiple dependent erroneous weak signals,

ALL's performance is relatively stable while the baseline accuracies get worse as the poor performing weak signal is repeated. The accuracy of AVG and W-AVG steadily degrades, while GE declines steeply to random performance.

4.7 Learning with Fixed, Incorrect Bounds

Instead of using the true training error as the bounds, we consider a more realistic scenario in which the experts are less precise about their error estimates. In practice, the true error rate may be difficult to estimate, so these experiments will validate whether our approach continues to work well when these bounds are inaccurate. We use a fixed upper bound of $b_1 = b_2 = b_3 = 0.3$ and report the performance of the ALL model and baselines in this setting.

Table 2 shows the accuracies obtained by the methods using the fixed bounds. The accuracy scores from the Statlog Satellite datasets are marginally higher than the results from the previous experiments, which used the true error rate (see Table 1), making it's performance statistically indistinguishable compared to GE.

While we arbitrarily chose a fixed bound of 0.3, we also tried various values of the bound, finding that ALL is not too sensitive to variations of this parameter. The only real challenge in setting this parameter is that when the bound is small enough, the problem becomes infeasible. See Fig. 4.

4.8 Crowdsourcing Experiments

We evaluate ALL on multiple datasets from the crowdsourcing domain measuring the predictive accuracy compared to the ground truth labels. We compare our method to a simple majority voting (**MAJ**), regularized minimax conditional entropy by Zhou et al. (2015) (**MMCE**) and lastly to Snorkel MeTaL by Ratner et al. (2018) (**MeTaL**). We use the true error rates of the weak signals as the bounds for the weak signals. The datasets we used for the experiments are described below.

RTE This dataset contains pairs of sentences labeled by whether the first sentence entails the second sentence Snow et al. (2008). There are 800 examples with 10 different crowd annotator labels for each example in the dataset. We use features provided by pre-trained BERT model (Devlin et al., 2018) for the training data.

Word Similarity The task in this dataset is to label pairs of words as either similar or dissimilar (Snow et al., 2008). There are 30 examples with 10 different crowd annotator labels for each example in the dataset. We use features provided by pre-trained BERT model Devlin et al. (2018) for the training data.

Blue Birds The classification task in this dataset is to identify the species of birds from photographs. The birds are either indigo buntings or blue grosbeaks. The dataset contains 108 images of birds with 39 annotator labels provided for each example. We use features provided by a pre-trained ResNet-101 model (He et al., 2016) for the training data.

Table 3 lists the accuracies of ALL on estimating labels for the datasets compared to competing approaches. ALL performs better than MeTaL, MMCE, and majority voting on the word similarity and blue birds datasets.

Data	ALL	MAJ	MMCE	MeTaL
RTE	0.834	0.895	0.910	**0.913**
Word Similarity	**0.933**	0.867	0.867	0.9
Blue Birds	**0.935**	0.759	0.889	0.889

Table 3: Label accuracy of ALL and baseline models on different datasets. The best performing methods that are boldfaced.

On the RTE dataset, its performance is inferior compared to competing methods. Investigating this, we discovered that the features from the embedding model were confusing for the learner, and using a reduced BERT embedding model as the features increases the accuracy to 0.9. We also observe that changing the learner $\boldsymbol{p}(\theta)$ from a simple logistic regression model to a two-layer neural network with non-linear activation function increases the performance. To further test the effect of changing the model family and data representation, we tried a two-layer neural network as the learner $\boldsymbol{p}(\theta)$ on the word similarity dataset, which achieved a better accuracy of 0.967. These extra results suggest that ALL's performance can be suboptimal if the model family and the data representation are mismatched.

4.9 Learning with Good Signals

In Section 4.6, we examined ALL's robustness to dependent error. In this section, we show how ALL synthesizes information from a combination of good and bad signals on annotator provided signals. We use the word similarity dataset. Starting by randomly selecting a single crowd worker's labels, we then iteratively add more weak crowd workers. For each number of crowd workers, we run 30 trials of randomly sampling that number of workers and calculate the mean accuracy. Figure 5 plots the mean accuracies of the models. On average, all methods get better as we increase the number of crowd annotator labels, but ALL gains the best performance compared to MMCE and majority vote. As more weak signals—or crowd workers—are added, each contributes a combination of redundant error and new information. ALL is able to utilize the information while avoiding the negative effects of redundant error to learn better labels for the dataset.

5. Multi Adversarial Label Learning

In the previous sections, we covered ALL for the binary case. In this section, we introduce Multi-ALL, an extension of ALL for multiclass and multilabel cases. Multi-ALL takes as input an unlabeled dataset and a set of constraints. These constraints are consistent with the weak supervision and define the space of possible labelings for the data. We formulate a nonzero-sum game between two agents: the adversary that optimizes inferred labels for the data and the learner that optimizes parameters for the model. In this game, the objective of the adversary is to assign labels that maximize the error of the model subject to the provided constraints. The model objective is to minimize its loss with respect to the adversarial labels. Formally, let the unlabeled training data be $X = \{x_1, \ldots, x_n\}$, and let f be a classifier

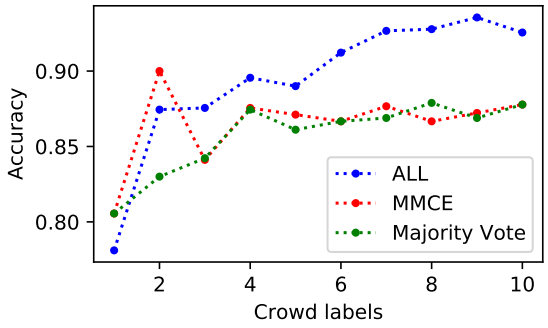


Figure 5: Performance of the methods as we increase the number of annotator labels from the crowd annotators in the word similarity dataset. For each number on the x-axis, we run 30 trials by randomly selecting that number of crowd labels from the dataset. We plot the mean accuracies for each of the 30 random trials.

parameterized by θ. The primal form of the nonzero-sum game we solve for Multi-ALL is

$$\begin{aligned} &\min_{\theta} L^{(f_\theta)}(X) \quad \text{and} \\ &\max_{\boldsymbol{Y} \in \Delta_C^n} L^{(\boldsymbol{Y})}(X) \quad \textbf{s.t.} \;\; g_j(\boldsymbol{Y}) \leq 0, \forall j \in \{1, \ldots, m\}, \end{aligned} \tag{12}$$

where $L^{(f_\theta)}$ is a loss function for training the classifier, $L^{(\boldsymbol{Y})}$ is a loss function for the adversarial label perturbations, Δ_C^n is the space of *label matrices* where each row is on the simplex of dimension k (i.e., a matrix that can represent a set of n multinomial distributions), $\boldsymbol{Y}$ is the estimated label matrix, ℓ is a loss function, and $\{g_1, \ldots, g_m\}$ is a set of linear constraint functions on $\boldsymbol{Y}$. The estimated labels are optimized adversarially ($\max_{Y \in \Delta_C^n}$), against the objective of the learning minimization.

5.1 Linear Label Constraints

In this section, we describe examples of linear constraints that fit into the Multi-ALL framework. Let $\boldsymbol{q} \in [0,1]^n$ be a weak signal that indicates—in a one-versus-rest sense—the probability that each example is in class c. And let $\boldsymbol{y}_c$ denote the cth column of matrix $\boldsymbol{Y}$, which is the current label's estimated probability that each example is in class c. As shown in the binary case, one set of possible linear constraints that tie the adversarial labels to the true labels is a bound on the error rate of each weak signal. The expected empirical error for the one-versus-rest task under these two probabilistic label probabilities is

$$\text{error}(\boldsymbol{q}, \boldsymbol{y}_c) = \frac{1}{n}\left(\boldsymbol{q}^\top(1-\boldsymbol{y}_c) + (1-\boldsymbol{q})^\top \boldsymbol{y}_c\right) = \frac{1}{n}\left(\boldsymbol{q}^\top \mathbf{1} + \boldsymbol{y}_c^\top(1-2\boldsymbol{q})\right), \tag{13}$$

where we use $\boldsymbol{q}^\top \mathbf{1}$ as a vector notation for the sum of $\boldsymbol{q}$ (or its dot product with the ones vector). Combined with an annotator provided estimate of a bound (b_{error}) on reasonable errors for their weak signals, an error-based constraint function for weak signal $\boldsymbol{q}$ on class c would have form

$$g_{\text{error}}(\boldsymbol{Y}) = \frac{1}{n}\left(\boldsymbol{q}^\top \mathbf{1} + \boldsymbol{y}_c^\top(1-2\boldsymbol{q})\right) - b_{\text{error}}. \tag{14}$$

This error-based constraint function can be insufficient to capture the informativeness of a weak signal, especially in cases where there is class imbalance. For multiclass classification, one-versus-rest signals will almost always be class-imbalanced. In such settings, we can allow annotators to indicate their estimates of weak-signal quality by indicating bounds on the precision. In the Multi-ALL setting, expected precision can also be expressed as a linear function of $\boldsymbol{Y}$:

$$\text{precision}(\boldsymbol{q}, \boldsymbol{y}_c) = \frac{\boldsymbol{y}_c^\top \boldsymbol{q}}{\boldsymbol{q}^\top \mathbf{1}}. \tag{15}$$

Since $\boldsymbol{q}$ is a constant with respect to the learning optimization, its appearance in the denominator of this expression does not affect the linearity. We can then define a precision constraint function for each weak signal $\boldsymbol{q}$ on class c:

$$g_{\text{prec.}}(\boldsymbol{Y}) = b_{\text{prec.}} - \frac{\boldsymbol{y}_c^\top \boldsymbol{q}}{\boldsymbol{q}^\top \mathbf{1}}. \tag{16}$$

Including precision constraints better captures the confusion matrix across different classes. It is also possible to design other linear constraints. As long as the constraints are linear, the feasible region for the maximization over $\boldsymbol{Y}$ remains convex.

5.2 Nonzero-Sum Losses

We describe here the loss functions we use to instantiate the Multi-ALL framework. The loss functions for the game must be differentiable; but, the choice of loss function is task-dependent and can have important impact on optimization. For multiclass classification using deep neural networks, our model uses popular cross-entropy loss. However, for the adversarial labeling, we instead use an expected error as the loss function, which the adversary maximizes. Formally, the model's loss is the cross-entropy

$$L^{(f_\theta)} = -\frac{1}{n} \sum_{c=1}^{C} \boldsymbol{y}_c^\top \log(f_\theta(X)), \tag{17}$$

while the adversarial labeler's loss is the expected error

$$L^{(\boldsymbol{Y})} = \frac{1}{n} \sum_{c=1}^{C} f_\theta(X)^\top (1 - \boldsymbol{y}_c). \tag{18}$$

We choose this form of error loss because it is linear in the adversarially optimized variable $\boldsymbol{Y}$. This makes the objective for the adversary $\boldsymbol{Y}$ concave, so we are maximizing a concave function subject to linear constraints. This makes the adversarial optimization a linear program with a unique optimum for any fixed f_θ. This form is relevant for the initialization scheme described in Section 5.3. We optimize the loss functions using Adagrad (Duchi et al., 2011).

5.3 Optimization

We use a primal-dual optimization that jointly solves an augmented Lagrangian relaxation of Eq. (12). Since our formulation uses a nonzero-sum game, we have two separate optimizations.

The analogous optimizations are

$$\begin{aligned} &\min_{\theta} L^{(f_\theta)} \quad \text{and} \\ &\min_{\boldsymbol{\gamma}\in\mathbb{R}^m_+} \max_{\boldsymbol{Y}\in\Delta^n_C} L^{(\boldsymbol{Y})} - \boldsymbol{\gamma}^\top G(\boldsymbol{Y}) - \frac{\rho}{2}\,||G(\boldsymbol{Y})||^2_+ := \hat{L}^{(\boldsymbol{Y})}\ , \end{aligned} \tag{19}$$

where $\boldsymbol{\gamma}$ is the vector of Karush-Kuhn-Tucker (KKT) multipliers, G is the vector of constraint function outputs (i.e., $G(\boldsymbol{Y})_j := g_j(\boldsymbol{Y})$), ρ is a positive parameter, and $||\cdot||^2_+$ denotes the norm of positive terms. The adversary optimization maximizes the linear loss from Eq. (18) while the learner minimizes the model loss from Eq. (17). A primal-dual solver for this problem updates the free variables using interleaved variations of gradient ascent and descent. We do this by updating $\boldsymbol{Y}$ and $\boldsymbol{\gamma}$ with their full gradients while holding θ fixed. Afterwards, we fix $\boldsymbol{Y}$ and $\boldsymbol{\gamma}$ and update θ with its mini-batches for a fixed number of epochs. The number of epochs depends on the size and architecture of the model network.

To preserve domain constraints on the variables $\boldsymbol{Y} \in \Delta^n_C$ and $\boldsymbol{\gamma} \geq 0$, we use projection steps that enforce feasibility. After each update to $\boldsymbol{Y}$ and $\boldsymbol{\gamma}$, we project $\boldsymbol{Y}$ to the simplex using the sorting method (Blondel et al., 2014), and we clip $\boldsymbol{\gamma}$ to be non-negative.

5.3.1 Initialization Scheme

To further facilitate faster convergence toward a local equilibrium, we warm start the optimization with a phase of optimization updating only $\boldsymbol{Y}$ and $\boldsymbol{\gamma}$. The effect of this warm-start phase is that we begin learning with a near-feasible $\boldsymbol{Y}$—one that is nearly consistent with the weak-supervision-based constraints. Since this phase uses the fixed output of a randomly initialized model f_θ, it does not require forward- or back-propagation through the deep neural network, so it is inexpensive, even for large datasets.

5.3.2 Analysis

The advantage of this primal-dual approach is that it enables inexpensive updates for the gaming agents and other variables being optimized, thereby allowing learning to occur without waiting for the solution of the inner optimization. At every iteration, the primal variables take maximization steps and the dual variables take minimization steps. However, for training deep neural networks, the primal-dual approach is not always ideal.

The large datasets needed to fit large models such as deep neural networks often require stochastic optimization to train efficiently. The key computational benefit of stochastic optimization is that it avoids the $O(n)$ cost of computing the true gradient update. Using a primal-dual approach to optimize Eq. (19) would also incur an $O(n)$ cost for each update to $\boldsymbol{Y}$. This cost is why we design our optimization scheme to update $\boldsymbol{y}$ and $\boldsymbol{\gamma}$ only after a fixed number of epochs. Since each epoch costs $O(n)$ computation, the added overhead does not change the asymptotic cost of training.

This optimization scheme has an added benefit that it increases the stability of the learning algorithm. By updating $\boldsymbol{Y}$ and $\boldsymbol{\gamma}$ only after a few epochs of training θ, we are solving the minimization over θ nearly to convergence. We still retain the advantages of primal-dual optimization over the $\boldsymbol{Y}$ variables, but without the added instability of simultaneous nonconvex optimization.

Algorithm 2 Multiclass Adversarial Label Learning

Require: Dataset $X = [x_1, \ldots, x_n]$, vector of constraint functions G, augmented Lagrangian parameter ρ.

Initialize model parameters θ (e.g., deep neural network weights)
Initialize $\boldsymbol{Y} \in \Delta_C^n$ (e.g., uniform probability)
Initialize $\boldsymbol{\gamma} \in \mathbb{R}_{\geq 0}$ (e.g., zeros)
while $G(\boldsymbol{Y}) >$ tolerance **do**
    Update $\boldsymbol{Y}$ with gradient $\nabla_{\boldsymbol{Y}} \hat{L}^{(\boldsymbol{Y})}$ (e.g., $\boldsymbol{Y} \leftarrow \boldsymbol{Y} + \alpha \nabla_{\boldsymbol{Y}} \hat{L}^{(\boldsymbol{Y})}$)
    Project $\boldsymbol{Y}$ to Δ_C^n
    Update $\boldsymbol{\gamma}$ with gradient $\nabla_{\boldsymbol{\gamma}} \hat{L}^{(\boldsymbol{Y})}$ (e.g., $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} - \rho G(\boldsymbol{Y})$)
    Clip $\boldsymbol{\gamma}$ to be non-negative
end while
while θ not converged **do**
    Update θ with $\nabla_\theta L_{\mathcal{B}}^{(f_\theta)}$ (mini-batches) for a fixed number of epochs
    Update $\boldsymbol{Y}$ with gradient $\nabla_{\boldsymbol{Y}} \hat{L}^{(\boldsymbol{Y})}$
    Project $\boldsymbol{Y}$ to Δ_C^n
    Update $\boldsymbol{\gamma}$ with gradient $\nabla_{\boldsymbol{\gamma}} \hat{L}^{(\boldsymbol{Y})}$
    Clip $\boldsymbol{\gamma}$ to be non-negative
end while
return model parameters θ

6. Experiments

We validate our approach on three fine-grained image classification tasks, comparing the performance of models trained with our approach to a baseline averaging method and model trained using labels generated from Snorkel (Ratner et al., 2017). Each of these methods trains from weak signals, and our experiments evaluate how well they can integrate noisy signals and how robust they are to confounding signals. We also add a supervised learning baseline for reference.

6.1 Quality of Constrained Labels

Before using our custom weak annotation framework, we first compare the quality of labels generated by Multi-ALL to existing methods for fusing weak signals. We follow the experiment design from a tutorial[1] designed by Ratner et al. (2017) to demonstrate their Snorkel system's ability to fuse weak signals and generate significantly higher quality labels than naive approaches. The experiment uses the *Microsoft COCO: Common Objects in Context* (Plummer et al., 2015) dataset to train detectors of whether a person is riding a bike within each image. We use the 903 images from the tutorial and weak signals generated by the labeling functions based on object occurrence metadata. We calculate the error and precision of each rule and use those to define Multi-ALL constraints. We then run the initialization scheme (the first while loop in Algorithm 2), which finds feasible labels adversarially fit

1. `https://github.com/HazyResearch/snorkel/blob/master/tutorials/images/Images_Tutorial.ipynb`

against a random initialization, i.e., arbitrary feasible labels. For an increasing number of weak signals, we compare ALL with error constraints, Multi-ALL with both error and precision constraints, Snorkel, and majority voting.

We plot the resulting error rate of the generated labels in Figure 6. For all numbers of weak signals, Multi-ALL obtains the highest accuracy labels. The labels generated by Snorkel have the same label error using two and three weak signals, but adding additional weak signals starts to confound Snorkel. Our framework is not confounded by these additional weak signals. Finally, corroborating the results reported by Snorkel's designers, the naive majority vote method has significantly higher error compared to any of the more sophisticated weak supervision techniques.

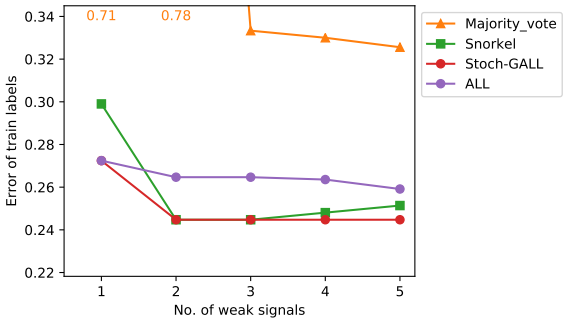


Figure 6: Error of MSCOCO bike-riding labels using Multi-ALL initialization compared to other methods.

6.2 Multiclass Image Classification

For our other experiments, we train multiclass image classifiers from weak supervision. We are interested in evaluating the effectiveness of the weak supervision approach, so we use the same deep neural network architecture for all experiments: a six-layer convolutional neural network where each layer contains a max-pooling unit, a relu activation unit, and dropout. The final layer is a fully connected layer with softmax output. Table 4 lists summary results of the error obtained by each method on each dataset using all the weak signals we provide the learners. The final result for each experiment is that Multi-ALL outperforms both Snorkel and averaging in all settings, showing a strong ability to fuse noisy signals and to avoid being confounded by redundant signals. We describe our form of weak supervision and each experiment in detail in the rest of the section.

6.2.1 Weak Signals

We ask human annotators to provide weak signals for image datasets. To generate each weak signal, we sample 50 random images belonging to different classes. We then ask the annotators to select a representative image and mark distinguishing regions of the image that indicate its belonging to a specific class. We then calculate pairwise comparisons between the pixels in the region of the reference image selected by an annotator and the pixels in the same region for all other images in the dataset. We measure the Euclidean distance between the pairs of images and convert the scores to probabilities with a logistic transform. Through

Data	Multi-ALL	Average	Snorkel	Supervised
Fashion-MNIST (weak)	**0.335**	0.447	0.401	0.142
Fashion-MNIST (pseudolabels + weak)	**0.228**	0.315	0.320	0.142
SVHN (pseudolabels + weak)	**0.231**	0.435	0.525	0.141

Table 4: Errors of models trained using all weak signals. In all three settings, Multi-ALL is able to train higher accuracy models than Snorkel or average labels. The settings include using all human-annotated weak signals (weak) and combining the human signals with pseudolabels (pseudolabels + weak).

this process, an annotator is guiding the design of simple nearest-neighbor one-versus-rest classifiers, where images most similar to the reference image are more likely to belong to its class. We ask annotators to generate many of these rules for the different classes, and we provide the computed probabilities as weak labels for the weakly supervised learners.

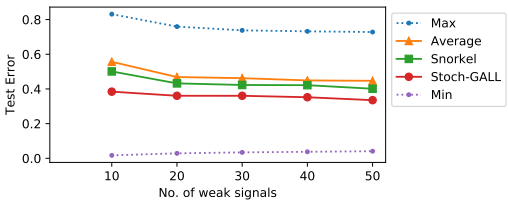
In practice, we found that these weak signals were noisy. In some experiments, they were insufficient to provide enough information for the classification task. However, our experiments show how different weakly supervised learners behave with informative but noisy signals. We discuss ideas on how to design better interfaces and better weak signals for the image classification task in Section 7.

We assume we have access to a labeled validation set consisting of 1% of the available data. We use this validation set to compute the precision and error bounds for the weak signals. This validation set is meant to simulate a human expert's estimate of error and precision. To encourage a fair comparison, we allow all methods to use these labels in addition to weak signals when training by appending the validation set to the dataset with its true labels. Since these bounds are evaluated on a very tiny set of the training data, they are noisy and prone to the same type of estimation mistakes an expert annotator may make. Therefore, they make a good test for how robust Multi-ALL is to imperfect bounds.

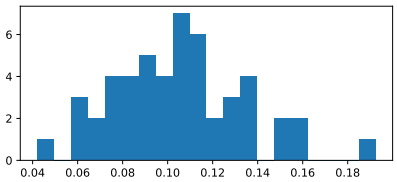
6.2.2 Weakly Supervised Image Classification

In this experiment, we train a deep neural network using only human-provided weak labels as described in Section 6.2.1. We use the *Fashion-MNIST* (Xiao et al., 2017) dataset, which represents an image-classification task where each example is a 28×28 grayscale image. The images are categorized into 10 classes of clothing types with 60,000 training examples and 10,000 test examples. We have annotators generate five one-versus-rest weak signals for each class, resulting in 50 total weak signals.

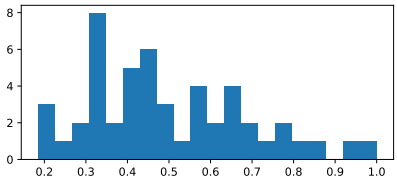
We plot analyses of models trained using weak supervision in Figure 7, where Fig. 7a plots the test error, and Fig. 7b and Fig. 7c are histograms of the error and precision bounds for the weak signals evaluated on the validation set. Since our weak signal is a one-versus-rest prediction of an image belonging to a particular class, the baseline precision and error should be 0.1 for training data with balanced classes. The histograms indicate that there is a wide range of precisions and errors for the different weak signals. Note that the order of the weak signals in our experiment was fixed as the order provided by the annotators. Thus, the first weak signal for each task is the first weak signal that the annotator generated for that dataset.



(a) Test error



(b) Error bounds



(c) Precision bounds

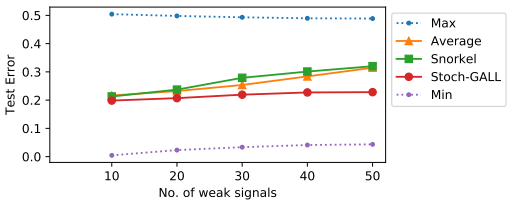
Figure 7: Analyses of experiments using the Fashion-MNIST dataset with human-provided weak signals.

The error rates in Fig. 7a suggest that the test error of the models decreases as we add more weak signals. Multi-ALL with both precision and error bounds outperforms Snorkel and the average baseline for all the weak signals. The min and max curves in the plots represent the best and worst possible label errors for labels that satisfy the provided constraints. The high error in the max curve indicates that the constraints alone still allow highly erroneous labels, yet our Multi-ALL framework trains models that perform well. The min curve indicates how close feasible adversarial labels could be to the true labels. In this experiment, the min curve is close to zero, which suggests that the inaccuracies in the provided bounds are not overly restrictive.

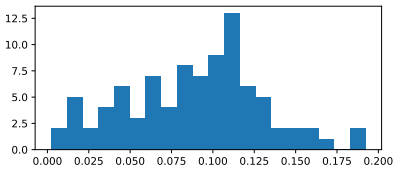
6.2.3 Weak-Pseudolabel Image Classification

In the previous experiments, the human provided weak signals are informative enough to train models to perform better than random guessing, but the resulting error rate is still significantly lower than that of supervised methods. To further boost the performance, we combine the human weak signals with *pseudolabels*: predictions of our deep model trained on the validation set and applied to the unlabeled training data (Lee, 2013). By training on the available 1% labels and predicting labels for the remaining 99% unlabeled examples, we create a new, high-quality weak signal. We calculate error and precision bounds for the pseudolabels with four-fold cross-validation on the validation set. We report the results of the models trained on the Fashion-MNIST dataset using this combination of pseudolabels and human weak signals.

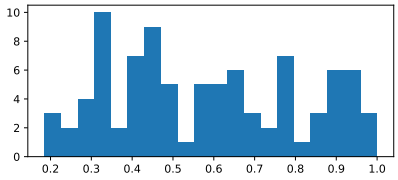
Fig. 8 contains plots of the results. The error and precision histograms now include higher precision bounds and lower error bounds, as a result of the pseudolabel signals being higher quality than the human provided weak signals. Additionally, the min and max error curves have lower values, indicating that we get better quality labels with these signals. The error trends in Fig. 8a are quite different compared to the previous experiment (Fig. 7a). All



(a) Test error



(b) Error bounds



(c) Precision bounds

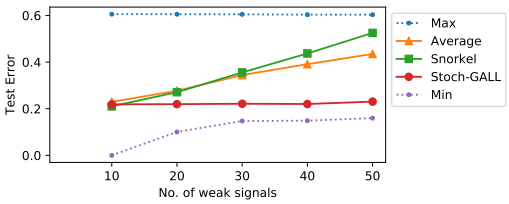
Figure 8: Test error on Fashion-MNIST dataset using pseudolabels and human weak signals.

the methods have good performance with the pseudolabel signals, but as we add the human signals, Snorkel and the average baseline are confounded and produce increasingly worse predictions. Multi-ALL however is barely affected by the human weak signals. The slight variation in the curve can be attributed to the inaccuracy of estimated bounds for the weak signals.

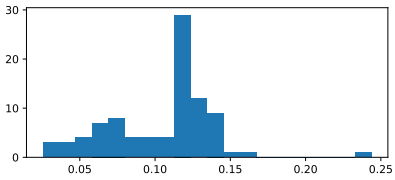
We hypothesize that this trend occurs because of the nature of our weak supervision. Since the weak signals are based on the selection of exemplar images, they may be effectively subsumed by a fully semi-supervised approach such as pseudolabeling. That is, the information provided by each human weak signal is already included in the pseudolabeling signal. This type of redundant information is common when using weak supervision. Many signals can have dependencies and redundancies. And despite the Snorkel system's modeling of dependencies among weak signals, it is still confounded by them while Multi-ALL's model-free approach is robust.

Our final experiments test the performance of the different models using pseudolabels and human weak labels on another image classification task. We use the Street View House Numbers (SVHN) (Netzer et al., 2018) dataset, which represents the task of recognizing digits on real images of house numbers taken by Google Street View. Each image is a 32×32 RGB vector. The dataset has 10 classes consisting of 73,257 training images and 26,032 test images.

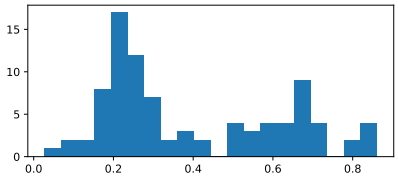
Figure 9 plots the results of the experiment. Figure 9a features the same trend as Fig. 8a. For this task, the human weak signals perform poorly in labeling the images, so they do not provide additional information to the learners. This fact is evident in the horizontal slope of the max curve. The min curve suggests that the human weak signals are redundant with poorly estimated bounds, and adding them decreases the space of possible labels for Multi-ALL. Comparing models, Multi-ALL's performance is not affected by the redundancies in the weak signals. Since the human weak signals are very similar, Snorkel seems to mistakenly trust the information from these signals more as we add more of them, thus hurting its model



(a) Test error



(b) Error bounds



(c) Precision bounds

Figure 9: Test error on SVHN dataset using pseudolabels and human weak signals.

performance. Multi-ALL uses the extra information provided to it as bounds on the weak signals to protect against placing higher emphasis on the redundant human weak signals.

7. Discussion

We introduced adversarial label learning (ALL), a method to train robust classifiers when access to labeled training data is limited. ALL trains a model without labeled data by making use of weak supervision to minimize the error rate for adversarial labels, which are subject to constraints defined by the weak supervision. We demonstrated that our method is robust against weak supervision signals that make dependent errors and gets better performance with better supervision signals. Our experiments confirm that ALL is able to learn models that outperform the weak supervision and baseline models. ALL is also capable of directly training classifiers to mimic the weak supervision.

Subsequently, we introduced Multi-ALL, a generalized adversarial label learning framework that enables users to encode information about the data as a set of linear constraints. We show in our experiments the performance of our method using precision and error constraints. However, Multi-ALL allows for other forms of linear constraints. Our evaluation demonstrates that our adaptive framework is able to generate high-quality labels for a learning task and is also able to combine different sources of weak supervision to increase the performance of a model. Our experiments show that our framework outperforms other weak supervision method on different image-classification tasks and is better at handling redundancies among weak supervision signals.

7.1 Limitations of ALL

While ALL has several advantages over alternative approaches for weakly supervised learning, it also has some limitations that are important to acknowledge. We discuss these limitations in this section and the open problems associated with them.

7.1.1 Incorrect or loose bounds

ALL trains a model with weak supervision signals and estimated error bounds of the signals. For ALL to learn, we expect the error bounds to be not too far from the true error rates of the weak signals. Using very tight bounds would over-constrain the optimization, thereby making it infeasible for a solution to be found. Loose bounds—on the other hand—would under-constrain the optimization and make the adversary too powerful. Bounds can be loose even with true error bounds for signals that are only slightly better than random. We have observed this effect in synthetic settings where the features of the data are not informative enough to form informative weak signals. The model training would then contend against arbitrary labels for the training data, causing ALL to fail. Because of the dependence of ALL's performance on the quality of the error bounds, we advise users of ALL to estimate the bounds using a validation set. In cases where a validation set is not available, we recommend using constant bounds or estimating the error bounds of the weak signals using methods that estimate error rates of classifiers without labeled data.

7.1.2 Quality of weak signals

An implicit assumption we make with the weak supervision signals we provide to ALL is that the weak signals are better than random. If this assumption is violated, this information needs to be passed to the learning algorithm by providing accurate estimates of the bound of the weak signals. Additionally, ALL performs better when the weak signals provided are somewhat correlated with the labels of the training data. We see the effect of these assumptions in our first set of experiments (Tables 1 and 2). We include weak signals whose accuracies are only slightly better than random. Using information from the bound and features of the data, ALL is able to learn high quality models that separate the data. For more complex datasets like image datasets, higher quality weak signals are necessary for the model to learn better. For example, in our work, we used simple nearest-neighbor weak signals provided by human annotators and programmatically generated weak signals. The human-provided weak signals performed well in some experiments (Fashion Mnist), but in other experiments (SVHN), they did not provide adequate information to the learner. One idea for improving the human signals is by learning latent data representations (e.g., with an autoencoder) and comparing the latent representations, rather than raw pixel values. We surmise that doing this will provide higher quality signal to the learning model.

7.1.3 Noise level

A major motivation for our algorithm is to be robust to noise from redundant weak signals. We showed in Fig. 3 experiments that demonstrate ALL's robustness to redundant/noisy signals. However, we have not established what level of noise ALL is robust to. One area of future work is to study the robustness of ALL with relation to random classification noise (Angluin and Laird, 1988) for binary classification, class-conditional noise (Stempfel and Ralaivola, 2009; Natarajan et al., 2013; Scott et al., 2013) for multi-class classification, or instance dependent noise (Manwani and Sastry, 2013; Ghosh et al., 2015). Recent works (Patrini et al., 2016, 2017) have attempted to establish theoretical justifications and connections between weakly supervised learning and robustness to label noise. We plan to explore ALL using concepts from these line of research. Our experiments do suggest that ALL is robust

to some noise. Its performance reduces with increased noise levels in the weak signals but it does not degrade as much as competing methods in our experiments. Our future research goal is to establish theoretical conditions of noise levels for which ALL is robust to.

7.1.4 Dependence on Learned Model Parameterization

As we demonstrated in Section 4.8 on the RTE dataset, ALL performance can be dependent on the parameterization of the learned model $\boldsymbol{p}(\theta)$. Linear models is adequate for datasets with simple features however when training ALL on complex text or image datasets, non-linear models like neural networks should be preferred to enable $\boldsymbol{p}(\theta)$ learn better parameters for the data.

7.2 Comparing Adversarial Modeling with Generative Modeling for Weak Supervision

ALL uses worst-case perturbations to defend against possible correlations among weak signals. However, some existing approaches for weakly supervised learning (Ratner et al., 2017, 2018; Varma et al., 2019; Halpern et al., 2016) use generative modeling to model the possible correlations of the weak signals. Unlike these methods, ALL does not depend on the modeling assumptions necessary to do generative modeling, such as on the dependency structure among the weak signals or the parametric form of the distributions. In practice, generative models work well when these modeling assumptions match the real data process. In these cases, generative approaches will provide better results than ALL, since these methods will properly model the uncertainty and correlation among the weak signals. Additionally, generative models have an advantage that they try to estimate the accuracy bounds of the weak signals and then use the estimated accuracies to combine the different weak signals. ALL instead takes in the bounds as an input and thus can be sensitive to incorrect bounds as explained in Section 7.1. If the bounds are set appropriately, ALL retains the useful information from the weak signals while also protecting against the possibility of correlated errors. However, if the bounds are too loose, fine-grained structure and information from the weak signals can be lost. Given the differences between ALL and generative models, we recommend using ALL when one is unconfident about generative modeling assumptions and when a reasonable estimate of the bounds of the weak signals can be obtained. In such settings, ALL's approach of protecting against worst-case conditions makes it robust against unknown correlations in weak supervision.

Acknowledgments

We thank NVIDIA for their support through the GPU Grant Program and Amazon for their support via the AWS Cloud Credits for Research program. Arachie and Huang were both supported by a grant from the U.S. Department of Transportation, University Transportation Centers Program to the Safety through Disruption University Transportation Center (69A3551747115). The work presented in this paper was primarily completed while both authors were at Virginia Tech.

References

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

Chidubem Arachie and Bert Huang. Stochastic generalized adversarial label learning. *arXiv preprint arXiv:1906.00512*, 2019a.

Chidubem Arachie and Bert Huang. Adversarial label learning. In *Proc. of the AAAI Conf. on Artif. Intelligence*, pages 3183–3190, 2019b.

Diogo Ayres-de Campos, Joao Bernardes, Antonio Garrido, Joaquim Marques-de Sa, and Luis Pereira-Leite. Sisporto 2.0: a program for automated analysis of cardiotocograms. *Journal of Maternal-Fetal Medicine*, 9(5):311–318, 2000.

Stephen H. Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, and Houman Alborzi. Snorkel drybell: A case study in deploying weak supervision at industrial scale. In *Intl. Conf. on Manag. of Data*, pages 362–375, 2019.

Akshay Balsubramani and Yoav Freund. Optimally combining classifiers using unlabeled data. *arXiv preprint arXiv:1503.01811*, 2015a.

Akshay Balsubramani and Yoav Freund. Scalable semi-supervised aggregation of classifiers. In *Advances in Neural Information Processing Systems*, pages 1351–1359, 2015b.

C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/mlearn/MLRepository.html.Irvine`.

Mathieu Blondel, Akinori Fujino, and Naonori Ueda. Large-scale multiclass support vector machine training via Euclidean projection onto the simplex. In *Intl. Conf. on Pattern Recognition*, pages 1289–1294. IEEE, 2014.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Conference on Computational Learning Theory*, pages 92–100. ACM, 1998.

Razvan C. Bunescu and Raymond Mooney. Learning to extract relations from the web using minimal supervision. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, volume 45, pages 576–583, 2007.

Bob Carpenter. Multilevel Bayesian models of categorical data annotation. *Unpublished manuscript*, 17(122):45–50, 2008.

Liang-Chieh Chen, Sanja Fidler, Alan L. Yuille, and Raquel Urtasun. Beat the mturkers: Automatic image labeling from weak 3d supervision. In *Proc. of the IEEE Conf. on Comp. Vis. and Pattern Recognition*, pages 3198–3205, 2014.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

Alexander Philip Dawid and Allan M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, pages 20–28, 1979.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Gregory Druck, Gideon Mann, and Andrew McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st Annual Intl. ACM SIGIR Conf. on Research and Dev. in Information Retrieval*, pages 595–602, 2008.

Simon S. Du and Wei Hu. Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. *arXiv preprint arXiv:1802.01504*, 2018.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

Jason A. Fries, Paroma Varma, Vincent S. Chen, Ke Xiao, Heliodoro Tejeda, Priyanka Saha, Jared Dunnmon, Henry Chubb, Shiraz Maskatia, and Madalina Fiterau. Weakly supervised classification of aortic valve malformations using unlabeled cardiac MRI sequences. *Nature Communications*, 10(1):1–10, 2019.

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049, 2010.

Huiji Gao, Geoffrey Barbier, and Rebecca Goolsby. Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, 26(3):10–14, 2011.

Aritra Ghosh, Naresh Manwani, and PS Sastry. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015.

Gauthier Gidel, Reyhane Askari Hemmat, Mohammad Pezeshki, Remi Lepriol, Gabriel Huang, Simon Lacoste-Julien, and Ioannis Mitliagkas. Negative momentum for improved game dynamics. *arXiv preprint arXiv:1807.04740*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

Yoni Halpern, Steven Horng, and David Sontag. Clinical tagging with joint probabilistic models. In *Proceedings of the Conference on Machine Learning for Healthcare*, pages 209–225, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Magnus R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proc. of the Annual Meeting of the Assoc. for Comp. Linguistics: Human Language Tech.*, pages 541–550, 2011.

Ariel Jaffe, Ethan Fetaya, Boaz Nadler, Tingting Jiang, and Yuval Kluger. Unsupervised ensemble learning with dependent classifiers. In *Artificial Intelligence and Statistics*, pages 351–360, 2016.

David R. Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems*, pages 1953–1961, 2011.

Ashish Khetan, Zachary C. Lipton, and Anima Anandkumar. Learning from noisy singly-labeled data. *arXiv preprint arXiv:1712.04577*, 2017.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, 2013.

Qiang Liu, Jian Peng, and Alexander T. Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 692–700, 2012.

Daniel Lowd and Christopher Meek. Adversarial learning. In *Proceedings of the ACM SIGKDD Intl. Conf. on Knowledge Discovery in Data Mining*, pages 641–647. ACM, 2005.

Gideon S. Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. *Proceedings of ACL-08: HLT*, pages 870–878, 2008.

Gideon S. Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of Machine Learning Research*, 11:955–984, 2010.

Naresh Manwani and PS Sastry. Noise tolerance under risk minimization. *IEEE Transactions on Cybernetics*, 43(3):1146–1151, 2013.

Andrew McCallum, Gideon Mann, and Gregory Druck. Generalized expectation criteria. *Computer science technical note, University of Massachusetts, Amherst, MA*, 94(95):159, 2007.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proc. of the Annual Meeting of the Assoc. for Comp. Ling.*, 2009.

Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Rami M. Mohammad, Fadi Thabtah, and Lee McCluskey. An assessment of features related to phishing websites using an automated technique. In *Internet Technology And Secured Transactions, 2012 Intl. Conf. for*, pages 492–497. IEEE, 2012.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep K. Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pages 1196–1204, 2013.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and A. Ng. The street view house numbers (SVHN) dataset. Technical report, Accessed 2016-08-01.[Online]., 2018.

Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. Loss factorization, weakly supervised learning and label noise robustness. In *International Conference on Machine Learning*, pages 708–717, 2016.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

Emmanouil Antonios Platanios, Avrim Blum, and Tom Mitchell. Estimating accuracy from unlabeled data. In *Proceedings of the Thirtieth Conf. on Uncertainty in Artificial Intelligence*, pages 682–691, 2014.

Emmanouil Antonios Platanios, Maruan Al-Shedivat, Eric Xing, and Tom Mitchell. Learning from imperfect annotations. *arXiv preprint arXiv:2004.03473*, 2020.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE Intl. Conf. on Computer Vision*, pages 2641–2649, 2015.

Adel Rajab, Chin-Tser Huang, Mohammed Al-Shargabi, and Jorge Cobb. Countering burst header packet flooding attack in optical burst switching network. In *Intl. Conf. on Information Security Practice and Experience*, pages 315–329. Springer, 2016.

Alex Ratner, Braden Hancock, Jared Dunnmon, Roger Goldman, and Christopher Ré. Snorkel metal: Weak supervision for multi-task learning. In *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning*, pages 1–4, 2018.

Alexander J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Info. Proc. Sys.*, pages 3567–3575, 2016.

Alexander J. Ratner, Stephen H. Bach, Henry R. Ehrenberg, and Chris Ré. Snorkel: Fast training set generation for information extraction. In *Proceedings of the 2017 ACM Intl. Conf. on Management of Data*, pages 1683–1686. ACM, 2017.

Vikas C. Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(4), 2010.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint Euro. Conf. on Mach. Learn. and Knowledge Disc. in Databases*, 2010.

Robert E. Schapire, Marie Rochery, Mazin Rahim, and Narendra Gupta. Incorporating prior knowledge into boosting. In *Intl. Conf. on Machine Learning*, volume 2, pages 538–545, 2002.

Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference On Learning Theory*, pages 489–511, 2013.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pages 3353–3364, 2019.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2107–2116, 2017.

Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y. Ng. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, 2008.

Jacob Steinhardt and Percy S. Liang. Unsupervised risk estimation using only conditional independence structure. In *Adv. in Neural Information Processing Systems*, pages 3657–3665, 2016.

Guillaume Stempfel and Liva Ralaivola. Learning svms from sloppily labeled data. In *International conference on artificial neural networks*, pages 884–893. Springer, 2009.

W. Nick Street, William H. Wolberg, and Olvi L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical Image Processing and Biomedical Visualization*, volume 1905, pages 861–871. Intl. Society for Optics and Photonics, 1993.

Mohamad Ali Torkamani and Daniel Lowd. Convex adversarial collective classification. In *Intl. Conf. on Machine Learning*, 2013.

Mohamad Ali Torkamani and Daniel Lowd. On robustness and regularization of structural support vector machines. In *Intl. Conf. on Machine Learning*, pages 577–585, 2014.

Paroma Varma, Frederic Sala, Ann He, Alexander Ratner, and Christopher Ré. Learning dependency structures for weak supervision models. *arXiv preprint arXiv:1903.05844*, 2019.

Mehmet Vurkaç. Clave-direction analysis: A new arena for educational and creative applications of music technology. *Journal of Music, Technology & Education*, 4(1):27–46, 2011.

Hai Wang and Hoifung Poon. Deep probabilistic logic: A unifying framework for indirect supervision. *arXiv preprint arXiv:1808.08485*, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Jia Xu, Alexander G. Schwing, and Raquel Urtasun. Tell me what you see and I will show you where it is. In *Proc. of the IEEE Conf. on Computer Vis. and Pattern Recog.*, pages 3190–3197, 2014.

Limin Yao, Sebastian Riedel, and Andrew McCallum. Collective cross-document relation extraction without labelled data. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pages 1013–1023, 2010.

Guojun Zhang and Yaoliang Yu. Convergence of gradient methods on bilin-ear zero-sum games. *arXiv preprint arXiv:1908.05699*, 2020.

Dengyong Zhou, Qiang Liu, John C Platt, Christopher Meek, and Nihar B Shah. Regularized minimax conditional entropy for crowdsourcing. *arXiv preprint arXiv:1503.07240*, 2015.

Yao Zhou and Jingrui He. Crowdsourcing via tensor augmentation and completion. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2435–2441, 2016.