Adversarial Label Learning

Chidubem Arachie
Department of Computer Science
Virginia Tech
achid17@vt.edu

Bert Huang
Department of Computer Science
Virginia Tech
bhuang@vt.edu

Abstract

We consider the task of training classifiers without labels. We propose a weakly supervised method—adversarial label learning—that trains classifiers to perform well against an adversary that chooses labels for training data. 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. Experiments on real datasets show that our method can train without labels and outperforms other approaches for weakly supervised learning.

1 Introduction

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. Many machine learning models require large amounts of labeled training data, which is usually hand labeled or observed and recorded. In real applications, large amounts of training data are often not easily accessible or are expensive to acquire, making labeled training data a critical bottleneck for machine learning.

An alternative for training machine learning models without labeled training data is *weak supervision*. Weak supervision uses domain knowledge about the specific problem, side information, or heuristics to approximate the true labels. 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.

ALL trains using weak supervision and aims to mitigate these problems 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.

We consider a binary classification setting where a parameterized model is trained to classify the data. 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.

2 Related Work

Weak supervision has become an important topic in the context of data-hungry deep learning models. A new line of research on data programming has produced a paradigm for weak supervision where data scientists write labeling functions that create noisy labels (Ratner et al. 2017; 2016). The approach then discovers relationships among the noisy labeling functions and is able to combine them and train data-hungry models. Other related approaches provide weak supervision in the form of constraints on the output space (Stewart and Ermon 2017), such as those that encode physical laws. Another related effort is on meta-learning for neural networks via weak supervision (Dehghani et al. 2017), using semi-supervised data to train an algorithm to learn from weak supervision.

Our work is related to existing methods that use variants of a generalized expectation (GE) criteria (Druck, Mann, and McCallum 2008; Mann and McCallum 2010; 2008) for semi- and weakly supervised learning. A GE criterion (McCallum, Mann, and Druck 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, Mann, and McCallum 2008), also allowing the weak supervision to include information about the uncertainty of a weak signal. Posterior regularization (PR) (Ganchev, Gillenwater, and Taskar 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.

Our work is also related to methods developed to estimate the error of classifiers without labeled data (Jaffe et al. 2016; Platanios, Blum, and Mitchell 2014; Steinhardt and Liang 2016) that rely on statistical relationships between the error rates of different classifiers. Many of these approaches extend classical statistics methods (Dawid and Skene 1979) by allowing the errors of the different classifiers to be dependent variables. A key goal of these approaches is to infer the error rate of these classifiers given only unlabeled data. In contrast, our setting assumes that we have reasonably good estimates of the error rates for the weak supervision provided by experts.

A different form of adversarial learning has recently become popular for deep learning (Goodfellow et al. 2014). Generative adversarial networks (GANs) pit a data generator and a discriminator against each other to train generative models that imitate realistic data distributions. Though our goal is not to train generative models, the stochastic optimization techniques developed for GANs may help our future work. *Virtual adversarial training* (Miyato et al. 2018) uses *input* perturbation to regularize a semi-supervised learning method. The method adds a regularization term to the objective function to make the learned model robust to input perturbations. Other approaches on adversarial input perturbation include methods for adversarial training of structured predictors (Torkamani and Lowd 2013; 2014), which lead to the added benefit of generalization guarantees. Our approach focuses on adversarial output manipulation, and opportunities to combine the benefits of both are promising directions of future work.

Other research (Lowd and Meek 2005; Madry et al. 2017) has considered variants of adversarial learning, training a classifier to learn sufficient information about another classifier to construct adversarial attacks. These efforts primarily focus on training models to be robust against malicious attacks, which is of interest in cybersecurity.

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:

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

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:

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

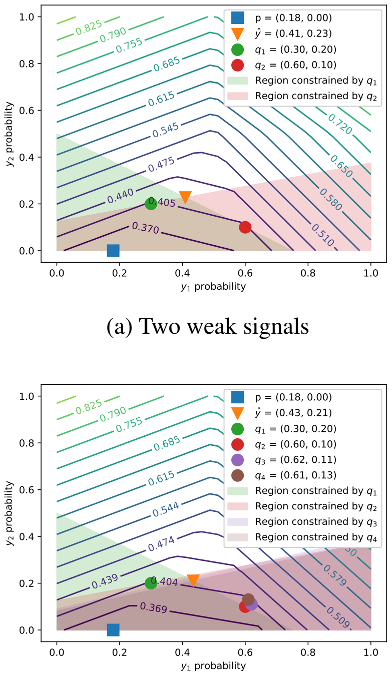
We can view the outer optimization as optimizing a primal objective that is the maximum of the constrained inner optimization. Define this primal function as $g(\theta)$, such that Eq. (4) can be equivalently written as $\min_\theta g(\theta)$. If the weak supervision error bounds are true, *this primal 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

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

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 $\rho$.
Initialize $\theta$ (e.g., random, zeros, etc.)
Initialize $\hat{\boldsymbol{y}} \in [0,1]^n$ (e.g., average of $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m$)
Initialize $\boldsymbol{\gamma} \in \mathbb{R}^m_{\geq 0}$ (e.g., zeros)
while not converged do
    Update $\theta$ with Equation (6)
    Update $\boldsymbol{p}$ with model and $\theta$
    Update $\hat{\boldsymbol{y}}$ with Equation (7)
    Update $\boldsymbol{\gamma}$ with Equation (8)
end while
return model parameters $\theta$
```

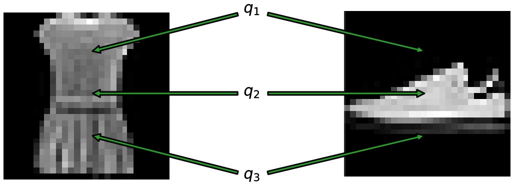


Figure 2: Features used to generate weak supervision signals on Fashion-MNIST data.

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

4.2 Baselines

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.,

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

and the reference expectation is

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

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. We compute these reference distributions on the training subset of the data. Our modified GE objective is then

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

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 nine different datasets to measure the predictive power of adversarial label learning (ALL). For each dataset, 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). And finally, we train models to mimic the average of the first weak signal (AVG-1), the first and second weak signals (AVG-2), and all three weak signals (AVG-3). Table 1 shows the mean accuracies obtained by running ALL on the different datasets.

4.4 Datasets

We describe the datasets used in the experiments below.

Fashion-MNIST The Fashion-MNIST dataset (Xiao, Rasul, and Vollgraf 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, Wolberg, and Mangasarian 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

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)	**0.923**	**0.922**	**0.924**	0.501	0.500	0.500	0.561	0.568	0.719	0.562	0.535	0.688
Fashion MNIST (CvB)	0.795	**0.831**	**0.840**	0.497	0.499	0.500	0.577	0.697	0.740	0.587	0.684	0.643
Breast Cancer	**0.942**	**0.944**	**0.945**	**0.936**	**0.936**	**0.935**	0.889	0.885	0.896	0.871	0.804	0.915
OBS Network	**0.717**	**0.718**	**0.719**	0.708	0.701	0.698	**0.724**	**0.723**	0.698	**0.721**	0.715	0.692
Cardiotocography	0.803	0.803	0.803	0.824	0.675	0.633	**0.942**	**0.947**	**0.942**	**0.946**	0.602	0.604
Clave Direction	0.646	**0.837**	0.746	0.646	0.796	0.772	0.646	0.645	0.707	0.646	0.648	0.625
Credit Card	**0.697**	**0.696**	**0.697**	**0.695**	0.460	0.424	0.660	0.662	0.607	0.659	0.572	0.557
Statlog Satellite	0.470	0.933	0.936	0.521	**0.987**	**0.992**	0.669	0.926	0.916	0.660	0.775	0.880
Phishing Websites	**0.896**	**0.895**	**0.895**	**0.898**	**0.894**	0.870	0.846	0.807	0.846	0.846	0.700	0.585
Wine Quality	0.572	**0.662**	0.623	0.455	0.427	0.454	0.570	0.573	0.555	0.571	0.596	0.570

Table 1: Test accuracy of ALL and baseline models on different datasets. The best performing methods that are not statistically distinguishable using a two-tailed paired t-test (p = 0.05) are boldfaced.

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, Thabtah, and McCluskey 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.

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.

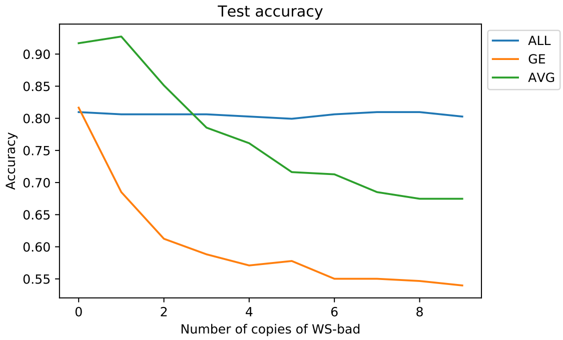


Figure 3: Performance of the methods using one good weak signal and repeated erroneous weak signals.

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 ALL. 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. The GE method only significantly outperforms ALL on the Statlog Satellite dataset, and nevertheless ALL still achieves a high accuracy score. The main failure case is the cardiotocography task, in which the AVG baseline 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.

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-

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 the baseline results from the previous experiments for convenience; they are unaffected by the change in error bound.

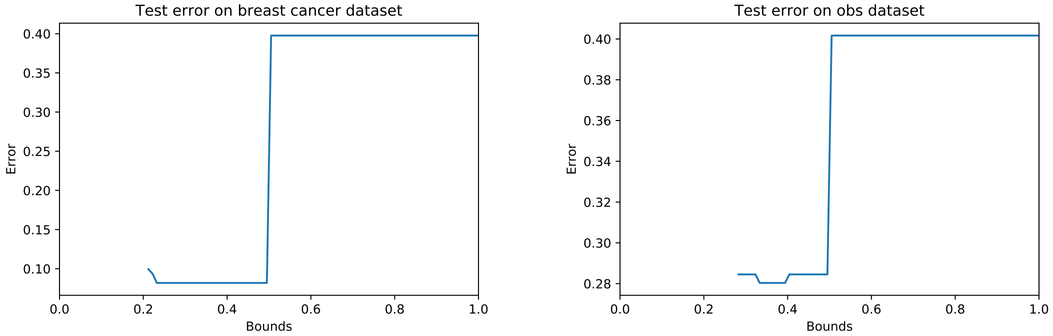


Figure 4: Illustrations showing the error of the model (ALL-3) when run with different fixed bounds between 0 and 1. Small bound values make infeasible constraints that prevent convergence, and are not plotted here.

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 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.

5 Conclusion

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. 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.

While our contribution is a significant methodological advance, there are several directions we hope to explore in our future work. We focused on training binary classifiers, but the principles underlying our method should extend to multiclass, regression, and even structured-output settings. Our algorithm requires reasoning over the entire training dataset, so we will explore ideas for scalability such as stochastic variations of our optimization procedure.

References

Ayres-de Campos, D.; Bernardes, J.; Garrido, A.; Marques-de Sa, J.; and Pereira-Leite, L. 2000. Sisporto 2.0: a program for automated analysis of cardiotocograms. *Journal of Maternal-Fetal Medicine* 9(5):311–318.

Blake, C., and Merz, C. 1998. UCI repository of machine learning databases.

Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; and Reis, J. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems* 47(4):547–553.

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

Dehghani, M.; Severyn, A.; Rothe, S.; and Kamps, J. 2017. Learning to learn from weak supervision by full supervision. *arXiv preprint arXiv:1711.11383*.

Druck, G.; Mann, G.; and McCallum, A. 2008. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 595–602. ACM.

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

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

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2672–2680.

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

Jaffe, A.; Fetaya, E.; Nadler, B.; Jiang, T.; and Kluger, Y. 2016. Unsupervised ensemble learning with dependent classifiers. In *Artificial Intelligence and Statistics*, 351–360.

Lowd, D., and Meek, C. 2005. Adversarial learning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 641–647. ACM.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

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

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

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

Miyato, T.; Maeda, S.-i.; Ishii, S.; and Koyama, M. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Mohammad, R. M.; Thabtah, F.; and McCluskey, L. 2012. An assessment of features related to phishing websites using an automated technique. In *Internet Technology And Secured Transactions, 2012 International Conference for*, 492–497. IEEE.

Platanios, E. A.; Blum, A.; and Mitchell, T. 2014. Estimating accuracy from unlabeled data. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, 682–691. AUAI Press.

Rajab, A.; Huang, C.-T.; Al-Shargabi, M.; and Cobb, J. 2016. Countering burst header packet flooding attack in optical burst switching network. In *International Conference on Information Security Practice and Experience*, 315–329. Springer.

Ratner, A. J.; De Sa, C. M.; Wu, S.; Selsam, D.; and Ré, C. 2016. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, 3567–3575.

Ratner, A. J.; Bach, S. H.; Ehrenberg, H. R.; and Ré, C. 2017. Snorkel: Fast training set generation for information extraction. In *Proceedings of the 2017 ACM International Conference on Management of Data*, 1683–1686. ACM.

Steinhardt, J., and Liang, P. S. 2016. Unsupervised risk estimation using only conditional independence structure. In *Advances in Neural Information Processing Systems*, 3657–3665.

Stewart, R., and Ermon, S. 2017. Label-free supervision of neural networks with physics and domain knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2576–2582.

Street, W. N.; Wolberg, W. H.; and Mangasarian, O. L. 1993. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical Image Processing and Biomedical Visualization*, volume 1905, 861–871. International Society for Optics and Photonics.

Torkamani, M. A., and Lowd, D. 2013. Convex adversarial collective classification. In *International Conference on Machine Learning*, 642–650.

Torkamani, M. A., and Lowd, D. 2014. On robustness and regularization of structural support vector machines. In *International Conference on Machine Learning*, 577–585.

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

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.