An Adversarial Labeling Game for Learning from Weak Supervision

Chidubem Arachie
Department of Computer Science
Virginia Tech

Bert Huang
Department of Computer Science
Virginia Tech

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. 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. Weak supervision provides an alternative to training machine learning models without labeled training data.

ALL is trained by setting up a two-player zero-sum game between an adversary and the model. The adversary, constrained by the weak supervision signals and bounds, chooses the labels for each training input while the model reduces the error made by the adversarial label. The Nash equilibrium of this system is a classifier robust to perturbations in the output distribution of the labels. Adversarial labeling preserves the partial correctness of the weak supervision signals but constructs scenarios where dependencies in the weak supervision are confounding. The resulting model performs well against biased weak supervision signals.

The inputs to ALL are a set of unlabeled data examples, a set of expert provided weak supervision signals that approximately label the data, and a corresponding set of estimated error bounds on these weak supervision signals. 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 well when trained with noisy weak signals that make dependent errors.

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 [11, 12]. The approach then discovers

Abstract for: Smooth Games Optimization and Machine Learning Workshop (NIPS 2018), Montréal, Canada.

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 [13], such as those that encode physical laws. Another related effort is on meta-learning for neural networks via weak supervision [1], using semi-supervised data to train an algorithm to learn from weak supervision.

A different form of adversarial learning has recently become popular for deep learning [5]. Generative adversarial networks (GANs) sets up a game between a data generator and a discriminator to train generative models that imitate realistic data distributions. Though our goal is not to train generative models, the zero-sum game between the neural networks in GAN is similar to the game in our approach. *Virtual adversarial training* [10] 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 [14, 15], 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.

Our work is related to existing methods that use variants of a generalized expectation (GE) criteria [2, 7, 8] for semi- and weakly supervised learning. A GE criterion [9] 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 [2], also allowing the weak supervision to include information about the uncertainty of a weak signal. Posterior regularization (PR) [4] 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.

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.,

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

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 3.1, 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 3.1.

3.1 Optimization Approach

The objective function derived in Eq. (4) above is a simple min-max smooth game. The learning objective is to find weight vectors θ and adversarial labels $\hat{\boldsymbol{y}}$ such that the system is in a Nash equilibrium. 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

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

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

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

which is clipped to be non-negative and uses a fixed step size ρ as dictated by the augmented Lagrangian method [6]. 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 [3]: 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.

4 Experiments

We run experiments on six 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) a 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.

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.

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	**0.923**	**0.922**	**0.924**	0.501	0.500	0.500	0.561	0.568	0.719	0.562	0.535	0.688
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
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
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

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.

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 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.

5 Conclusion

We introduced adversarial label learning (ALL), a method to train classifiers without labeled data by making use of weak supervision. The method sets up a zero-sum game between a model and an adversary that chooses labels for the model. The method trains a model to minimize the error rate for adversarial labels, which are subject to constraints defined by the weak supervision. We demonstrated through our experiments that our method outperform the weak signals and baseline models. We plan to focus our future work on deriving stochastic variation of our optimization procedure. Also, we plan to investigate the performance of our method in a structured output setting where the learning task involves predicting structured objects rather than scalar discrete or real values.

References

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

[2] G. Druck, G. Mann, and A. McCallum. 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*, pages 595–602. ACM, 2008.

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

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

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

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

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

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

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

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

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

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

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

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

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