Constrained Labeling for Weakly Supervised Learning

Chidubem Arachie

Bert Huang

Abstract

Curation of large fully supervised datasets has become one of the major roadblocks for machine learning. Weak supervision provides an alternative to supervised learning by training with cheap, noisy, and possibly correlated labeling functions from varying sources. The key challenge in weakly supervised learning is combining the different weak supervision signals while navigating misleading correlations in their errors. In this paper, we propose a simple data-free approach for combining weak supervision signals by defining a constrained space for the possible labels of the weak signals and training with a random labeling within this constrained space. Our method is efficient and stable, converging after a few iterations of gradient descent. We prove theoretical conditions under which the worst-case error of the randomized label decreases with the rank of the linear constraints. We show experimentally that our method outperforms other weak supervision methods on various text- and image-classification tasks.

1 INTRODUCTION

Recent successful demonstrations of machine learning have created an explosion of interest. The key driver of these successes is the progress in deep learning. Researchers in different fields and industries are applying deep learning to their work with varying degrees of success. Training deep learning models typically requires massive amounts of data, and in most cases this data needs to be labeled for supervised learning. The process of collecting labels for large training datasets is often expensive and can be a major bottleneck for practical machine learning.

To enable machine learning when labeled data is not available, researchers are increasingly turning to weak supervision. Weakly supervised learning involves training models using noisy labels. Using multiple sources or forms of weak supervision is common, as it provides diverse information to the model. However, each source of weak supervision has its own bias that can be transmitted to the model. Different weak supervision signals can also conflict, overlap, or—in the worst case—make dependent errors. Thus, a naive combination of these weak signals would hurt the quality of a learned model. The key problem then is how to reliably combine various sources of weak signals to train an accurate model.

To solve this problem, we propose constrained label learning (CLL), a method that processes various weak supervision signals and combines them to produce high-quality training labels. The idea behind CLL is that, given the weak supervision, we can define a constrained space for the labels of the weak signals and training with a random label within this constrained space. Our method is efficient and stable, converging after a few iterations of gradient descent. We prove theoretical conditions under which the worst-case error of the randomized label decreases with the rank of the linear constraints. We show experimentally that our method outperforms other weak supervision methods on various text- and image-classification tasks.

CLL takes as input (1) a set of unlabeled data examples, (2) multiple weak supervision signals that label a subset of data and can abstain from labeling the rest, and (3) a corresponding set of expected error rates for the weak supervision signals. While the weak supervision signals can abstain on various examples, we require that the combination of the weak signals have full coverage on the training data. The expected error rates can be estimated if the weak supervision signals have been tested on historical data or a domain expert has knowledge about their performance. In cases where the expected error rates are unavailable, they can be treated as a hyperparameter. Our experiments in Section 3 show that CLL is still effective when it is trained
Weakly supervised learning has gained prominence in recent years due to the need to train models without access to manually labeled data. The recent success of deep learning has exacerbated the need for large-scale data annotation, which can be prohibitively expensive. One weakly supervised paradigm, data programming, allows users to define labeling functions that noisily label a set of unlabeled data. Data programming then combines the noisy labels to form probabilistic labels for the data by using a generative model to estimate the accuracies and dependencies of the noisy/weak supervision signals. This approach underlies the popular software package Snorkel, which is related to this approach in that we use different weak signal sources and compile them into a single (soft) labeling. However, unlike Snorkel’s methods, we do not train a generative model and avoid the need for probabilistic modeling assumptions. Recently, Snorkel MetaL was proposed for solving multi-task learning problems with hierarchical structure. A user provides weak supervision for the hierarchy of tasks which is then combined in an end-to-end framework.

Another recently developed approach for weakly supervised learning is adversarial label learning (ALL). ALL was developed for training binary classifiers from weak supervision. ALL trains a model to perform well in the worst case for the weak supervision by simultaneously optimizing model parameters and adversarial labels for the training data in order to satisfy the constraint that the error of the weak signals on the adversarial labels be within provided error bounds. The authors also recently proposed Stoch-GALL, an extension for multi-class classification that incorporates precision bounds. Our work is related to ALL and Stoch-GALL in that we use the same error definition the authors introduced. However, the expected errors we use do not serve as upper bound constraints for the weak signals. Additionally, ALL avoids the adversarial setting that requires unstable simultaneous optimization of the estimated labels and the model parameters. Lastly, while ALL and Stoch-GALL require weak supervision signals to label every example, we allow for weak supervision signals that abstain on different data subsets.

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. Ensemble methods such as boosting[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[2015], our settings differ since we do not expect to have access to labeled data.

A growing set of weakly supervised applications includes web knowledge extraction, computer vision, computational linguistics, and tagging of medical conditions from health records. As better weakly supervised methods are developed, this set will expand to include other important applications.

We will show an estimation method that is connected to those developed to estimate the error of classifiers without labeled data. Our method is related to this approach in that we use different weak signal sources and compile them into a single (soft) labeling. However, unlike Snorkel’s methods, we do not train a generative model and avoid the need for probabilistic modeling assumptions. Recently, Snorkel MetaL was proposed for solving multi-task learning problems with hierarchical structure. A user provides weak supervision for the hierarchy of tasks which is then combined in an end-to-end framework.

Another recently developed approach for weakly supervised learning is adversarial label learning (ALL)[2008]. ALL was developed for training binary classifiers from weak supervision. ALL trains a model to perform well in the worst case for the weak supervision by simultaneously optimizing model parameters and adversarial labels for the training data in order to satisfy the constraint that the error of the weak signals on the adversarial labels be within provided error bounds. The authors also recently proposed Stoch-GALL, an extension for multi-class classification that incorporates precision bounds. Our work is related to ALL and Stoch-GALL in that we use the same error definition the authors introduced. However, the expected errors we use do not serve as upper bound constraints for the weak signals. Additionally, ALL avoids the adversarial setting that requires unstable simultaneous optimization of the estimated labels and the model parameters. Lastly, while ALL and Stoch-GALL require weak supervision signals to label every example, we allow for weak supervision signals that abstain on different data subsets.

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. Ensemble methods such as boosting[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[2015], our settings differ since we do not expect to have access to labeled data.

A growing set of weakly supervised applications includes web knowledge extraction, computer vision, computational linguistics, and tagging of medical conditions from health records. As better weakly supervised methods are developed, this set will expand to include other important applications.

We will show an estimation method that is connected to those developed to estimate the error of classifiers without labeled data. Our method is related to this approach in that we use different weak signal sources and compile them into a single (soft) labeling. However, unlike Snorkel’s methods, we do not train a generative model and avoid the need for probabilistic modeling assumptions. Recently, Snorkel MetaL was proposed for solving multi-task learning problems with hierarchical structure. A user provides weak supervision for the hierarchy of tasks which is then combined in an end-to-end framework.

Another recently developed approach for weakly supervised learning is adversarial label learning (ALL)[2008]. ALL was developed for training binary classifiers from weak supervision. ALL trains a model to perform well in the worst case for the weak supervision by simultaneously optimizing model parameters and adversarial labels for the training data in order to satisfy the constraint that the error of the weak signals on the adversarial labels be within provided error bounds. The authors also recently proposed Stoch-GALL, an extension for multi-class classification that incorporates precision bounds. Our work is related to ALL and Stoch-GALL in that we use the same error definition the authors introduced. However, the expected errors we use do not serve as upper bound constraints for the weak signals. Additionally, ALL avoids the adversarial setting that requires unstable simultaneous optimization of the estimated labels and the model parameters. Lastly, while ALL and Stoch-GALL require weak supervision signals to label every example, we allow for weak supervision signals that abstain on different data subsets.

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. Ensemble methods such as boosting[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[2015], our settings differ since we do not expect to have access to labeled data.

A growing set of weakly supervised applications includes web knowledge extraction, computer vision, computational linguistics, and tagging of medical conditions from health records. As better weakly supervised methods are developed, this set will expand to include other important applications.

We will show an estimation method that is connected to those developed to estimate the error of classifiers without labeled data. Our method is related to this approach in that we use different weak signal sources and compile them into a single (soft) labeling. However, unlike Snorkel’s methods, we do not train a generative model and avoid the need for probabilistic modeling assumptions. Recently, Snorkel MetaL was proposed for solving multi-task learning problems with hierarchical structure. A user provides weak supervision for the hierarchy of tasks which is then combined in an end-to-end framework.

Another recently developed approach for weakly supervised learning is adversarial label learning (ALL)[2008]. ALL was developed for training binary classifiers from weak supervision. ALL trains a model to perform well in the worst case for the weak supervision by simultaneously optimizing model parameters and adversarial labels for the training data in order to satisfy the constraint that the error of the weak signals on the adversarial labels be within provided error bounds. The authors also recently proposed Stoch-GALL, an extension for multi-class classification that incorporates precision bounds. Our work is related to ALL and Stoch-GALL in that we use the same error definition the authors introduced. However, the expected errors we use do not serve as upper bound constraints for the weak signals. Additionally, ALL avoids the adversarial setting that requires unstable simultaneous optimization of the estimated labels and the model parameters. Lastly, while ALL and Stoch-GALL require weak supervision signals to label every example, we allow for weak supervision signals that abstain on different data subsets.

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. Ensemble methods such as boosting[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[2015], our settings differ since we do not expect to have access to labeled data.

A growing set of weakly supervised applications includes web knowledge extraction, computer vision, computational linguistics, and tagging of medical conditions from health records. As better weakly supervised methods are developed, this set will expand to include other important applications.
for semi- and weakly supervised learning. They work by providing parameter estimates as constraints to the objective function of the model so that the label distribution of the trained model tries to match the constraints. In our approach, we incorporate human knowledge as error estimates into our algorithm. However, we do not use the constraints for model training. Instead, we use them to generate training labels that satisfy the constraints, and these labels can then be used downstream to train any model.

3 CONSTRINED LABEL LEARNING

The goal of constrained label learning (CLL) is to return accurate training labels for the data given the weak supervision signals. The estimation of these labels should be aware of the correlation among the weak supervision signals and should not be confounded by it. Toward this goal, we use the weak signals’ expected error to define a constrained space of possible labelings for the data. Any vector sampled from this space can then be used as training labels. We consider the setting in which the learner has access to a training set of unlabeled examples, and a set of weak supervision signals from various sources that provide approximate indicators of the target classification for the data. Along with the weak supervision signals, we are provided estimates of the expected error rates of the weak signals. Formally, let the data be $X = \{x_1, \ldots, x_n\}$. These examples have corresponding labels $y = \{y_1, \ldots, y_n\} \in \{0, 1\}^n$. For multi-label classification, where each example may be labeled as a member of $K$ classes, we expand the label vector to include an entry for each example-class combination, i.e., $y = \{y_{1,1}; \ldots, y_{1,K}; \ldots, y_{n,1}; \ldots, y_{n,K}\}$, where $y_{ij}$ is the indicator of whether the $i$th example is in class $j$.

With weak supervision, the training labels $y$ are unavailable. Instead, we have access to $m$ weak supervision signals $\{w_1, \ldots, w_m\}$, where each weak signal $w \in [0, 1]^n$ is represented as a vector of indicators that each example is in each class. The weak signals can choose to abstain on some examples. In that case, they assign a null value $\emptyset$ to that example’s entry. In practice, weak signals for multi-class problems typically only label one class at a time, such as a one-versus-rest classification rule, so they effectively abstain on all out-of-class entries. The weak signals can be soft labels (probabilities) or hard labels (class assignments) of the data. In conjunction with the weak signals, the learner also receives the expected error rates of the weak signals $\epsilon = [\epsilon_1, \ldots, \epsilon_m]$. In practice, the error rates of the weak signals are estimated or treated as a hyperparameter. The expected empirical error of a weak signal $w_i$ is

$$\epsilon_i = \frac{1}{n_i} \left( 1_{(w_i \neq \emptyset)} w_{i,1} - y_{1} + 1_{(w_i \neq \emptyset)} (1 - w_{i,1}) y_{5} \right)$$

where $y_k$ is the true label for the class $k$ that the weak signal $w_i$ labels, $n_i = \sum 1_{(w_i \neq \emptyset)}$ and $1_{(w_i \neq \emptyset)}$ is an indicator function that returns 1 on examples the weak signals label (i.e., do not abstain on). Hence, we only calculate the error of the weak signals on the examples they label.

Analogously to Eq. (1), we can express the expected error of all weak signals for the label vector as a system of linear equations in the form $A\tilde{y} = c$. To do this, we define each row in $A$ as

$$A_i = 1_{(w_i \neq \emptyset)} (1 - 2w_{i,1})$$

a linear transformation of a weak signal $w$. Each entry in the vector $c$ is the difference between the expected error of the weak signal and the sum of the weak signal, i.e.,

$$c_i = n_i \epsilon_i - w_{i,1} 1_{(w_i \neq \emptyset)}.$$  

Valid label vectors then must be in the space

$$\{\tilde{y} | A\tilde{y} = c \land \tilde{y} \in [0, 1]^n\}.$$  

The true label $y$ is not known. Thus, we want to find training labels $\tilde{y}$ that satisfy the system of linear equations.

3.1 ALGORITHM

Having defined the space of possible labelings for the data given the weak signals, we explain here how we efficiently sample a vector of training labels from the space. First, we initialize a random $\tilde{y}$ from a uniform distribution $\tilde{Y} \sim U(0, 1)^n$. Then we minimize a quadratic penalty on violations of the constraints defining the space. The objective function is

$$\min_{\tilde{y} \in [0, 1]^n} \|A\tilde{y} - c\|_2^2.$$
We start by analyzing the case where we have the true error. Algorithm 1.

**Theorem 1.** 

For any \( \mathbf{y} \) from Eq. (2) and \( \epsilon \) from Eq. (3) using the weak signals and expected errors.

1. Define \( \mathbf{A} \) from Eq. (2) and \( \epsilon \) from Eq. (3) using weak signals and expected errors.
2. Initialize \( \mathbf{y} \) as \( \mathbf{y} \approx U(0, 1) \)
3. While not converged do
   4. Update \( \mathbf{y} \) with its gradient from Eq. (5)
   5. Clip \( \mathbf{y} \) to \([0, 1]^n\)
6. End while

Return estimated labels \( \mathbf{y} \)

The solution to this quadratic objective function gives us feasible labels for the training data. In our experiments, we estimate the error rates \( \epsilon \) of the weak signals. In cases where error estimates make an infeasible space, this quadratic penalty acts as squared slack. We solve Eq. (5) iteratively using projected Adagrad [Duchi et al. 2011], clipping \( \mathbf{y} \) values to \([0, 1]^n\) between gradient updates. This approach is fast and efficient, even for large datasets. Our algorithm is a simple quadratic convex optimization that converges to a unique optimum for each initialization of \( \mathbf{y} \). In our experiments, it converges after only a few iterations of gradient descent. We run the algorithm 3 times with random initialization of \( \mathbf{y} \) and take the mean of the \( \mathbf{y} \)s as the estimated label. We observed that the labels returned from the different runs are very similar. We fix the number of iterations of gradient descent for each run to 200 for all our experiments. The full algorithm is summarized in Algorithm 1.

3.2 ANALYSIS

We start by analyzing the case where we have the true error \( \epsilon \), in which case the true label vector \( \mathbf{y} \) for CLL is a solution in the feasible space. Although the true error rates are not available in practice, this ideal setting is the motivating case for the CLL approach. To begin the analysis, consider an extreme case: if \( \mathbf{A} \) is a square matrix with full rank, then the only valid label \( \mathbf{y} \) in the space is the true label, \( \mathbf{y} = \mathbf{y} \). Normally, \( \mathbf{A} \) is usually underdetermined, which means we have more data examples than weak signals. In this case, there are many solutions for \( \mathbf{y} \), so we can analyze this space to understand how distant any feasible vector is from the vector of all incorrect labels. Since label vectors are constrained to be in the unit box, the farthest possible label vector from the true labels is \((1 - \mathbf{y})\). The result of our analysis is the following theorem, which addresses the binary classification case with non-abstaining weak signals.

**Theorem 1.** For any \( \mathbf{y} \in [0, 1]^n \) such that \( \mathbf{A}\mathbf{y} = \mathbf{c} \), its Euclidean distance from the negated label vector \((1 - \mathbf{y}) \in [0, 1]^n \) is bounded below by

\[
||\mathbf{y} - (1 - \mathbf{y})|| \geq n||\mathbf{A}^\dagger(1 - 2\epsilon)||,
\]

where \( \mathbf{A}^\dagger \) is the Moore-Penrose pseudoinverse of \( \mathbf{A} \).

**Proof.** We first relax the constrained space by removing the \([0, 1]^n\) box constraints. We can then analyze the projection onto the feasible space:

\[
\min_{\mathbf{y}} ||(1 - \mathbf{y}) - \mathbf{y}|| \text{ s.t. } \mathbf{A}\mathbf{y} = \mathbf{c}.
\]

(7)

Define a vector \( \mathbf{z} := \mathbf{y} - (1 - \mathbf{y}) \). We can rewrite the distance as

\[
\min_{\mathbf{z}} ||(1 - 2\mathbf{y}) - \mathbf{z}|| \text{ s.t. } \mathbf{A}\mathbf{z} = 0.
\]

(8)

The minimization is a projection of \((1 - 2\mathbf{y})\) onto the null space of \( \mathbf{A} \). Since the null and row spaces of a matrix are complementary, \((1 - 2\mathbf{y})\) decomposes into

\[
(1 - 2\mathbf{y}) = \mathbf{P}_{\text{row}}(1 - 2\mathbf{y}) + \mathbf{P}_{\text{null}}(1 - 2\mathbf{y}),
\]

where \( \mathbf{P}_{\text{row}} \) and \( \mathbf{P}_{\text{null}} \) are orthogonal projections into the row and null spaces of \( \mathbf{A} \), respectively. We can use this decomposition to rewrite the distance of interest:

\[
||((1 - 2\mathbf{y}) - \mathbf{P}_{\text{null}}(1 - 2\mathbf{y}))|| = ||((1 - 2\mathbf{y}) - (1 - 2\mathbf{y}) - \mathbf{P}_{\text{row}}(1 - 2\mathbf{y})))||
\]

(9)

\[
= ||\mathbf{P}_{\text{row}}(1 - 2\mathbf{y})||.
\]

For any vector \( \mathbf{v} \), its projection into the row space of matrix \( \mathbf{A} \) is \( \mathbf{A}^\dagger \mathbf{v} \), where \( \mathbf{A}^\dagger \) is the Moore-Penrose pseudoinverse of \( \mathbf{A} \). The distance of interest is thus \( ||\mathbf{A}^\dagger \mathbf{A}(1 - 2\mathbf{y})|| \). We can use the definition of \( \mathbf{A} \) to further simplify. Let \( \mathbf{W} \) be the matrix of weak signals \( \mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_m]^\top \). Then the distance is

\[
||\mathbf{A}^\dagger(1 - 2\mathbf{W})(1 - 2\mathbf{y})||
\]

(10)

\[
= ||\mathbf{A}^\dagger((1 - 2\mathbf{W})\mathbf{I}_n - 2(1 - 2\mathbf{W})\mathbf{y})||
\]

\[
= ||\mathbf{A}^\dagger(n - 2\mathbf{W}\mathbf{I}_n - 2A\mathbf{y})||.
\]

Because \( \mathbf{A}\mathbf{y} = \mathbf{c} = n\epsilon - \mathbf{W}\mathbf{I}_n \), terms cancel, yielding the bound in the theorem:

\[
||\mathbf{A}^\dagger(n - 2\mathbf{W}\mathbf{I}_n - 2n\epsilon + 2\mathbf{W}\mathbf{I}_n)||
\]

(11)

\[
= ||\mathbf{A}^\dagger(n - 2n\epsilon)|| = n||\mathbf{A}^\dagger(1 - 2\epsilon)||.
\]

This bound provides a quantity that is computable in practice. However, to gain an intuition about what factors affect its value, the distance formula can be further analyzed by using the singular-value decomposition (SVD) formula for the pseudoinverse. Consider SVD \( \mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top \). Then \( \mathbf{A}^\dagger = \mathbf{V}\Sigma^\dagger\mathbf{U}^\top \), where the pseudoinverse \( \Sigma^\dagger \) contains the reciprocal of all nonzero singular values along the diagonal (and zeros elsewhere). The distance simplifies to

\[
n||\mathbf{V}\Sigma^\dagger\mathbf{U}^\top(1 - 2\epsilon)|| = n||\Sigma^\dagger\mathbf{U}^\top(1 - 2\epsilon)||.
\]

(12)

since \( \mathbf{V} \) is orthonormal. Furthermore, let \( \mathbf{p} = \mathbf{U}^\top(1 - 2\epsilon) \), i.e., \( \mathbf{p} \) is a rotation of the centered error rates of the weak signals.
3.3 ERROR ESTIMATION

In our analysis, we assume that the expected error rates of the weak signals are available. This may be the case if the weak signals have been evaluated on historical data or if an expert provides the error rates. In practice, users typically define weak supervision signals whose error rates are unknown. In this section, we discuss two approaches to handle such situations. We test these estimation techniques on real and synthetic data in our experiments, finding that CLL with these strategies forms a powerful weakly supervised approach.

3.3.1 Agreement Rate Method

Estimating the error rates of binary classifiers using their agreement rates was first proposed by [Platanios et al.] (2014). They propose two different objective functions for solving the error rates of classifiers using their agreement rates as constraints. Similar to MeTaL [Ratner et al.] (2018), we solve a matrix-completion problem to find a low-rank factorization for the weak signal accuracies. We assume that if the weak signals are conditionally independent, we can relate the disagreement rates to the weak signal accuracies. We implemented this method and report its performance in our synthetic experiment (see Section 4). The one-vs-all form of the weak signals on our real datasets violates the assumption that each weak signal makes prediction on all the classes, so we cannot use the agreement rate method on our real data.

3.3.2 Uniform Error Rate

The idea of using uniform error rates of the weak signals was first proposed in ALL [Arachie and Huang] (2019b). Their experiments showed that ALL can learn as effectively as when using true error rates by using a constant for the error rates of all the weak signals on their binary classification datasets. We use this approach in our experiments and extend it to weak supervision signals that abstain and also on multi-class datasets. Figure 3 plots the accuracy of generated labels as we increase the error-rate parameter. On the binary-class SST-2 dataset, the label accuracy remains similar if the error rate is set between 0 and 0.5 and drops for values at least 0.5. On the multiclass Fashion-MNIST data, we notice similar behavior where the label accuracies are similar between 0.05 and 0.1 and drop with larger values. We surmise that this behavior mirrors the type of weak supervision signals we use in our experiments. The weak signals in our real experiments are one-vs-all signals; hence a baseline signal (guessing 0 on all examples) will have an error rate of \( \frac{1}{K} \). Performance deteriorates when the error rate is worse than this baseline rate.

Figure 2: Error of CLL estimated labels compared to majority vote as we increase the rank of \( A \) by replacing redundant weak signals with linearly independent weak signals.

\[
n||\Sigma^+ p|| = n \sqrt{\sigma_1^2 p_1^2 + \ldots + \sigma_m^2 p_m^2}, \tag{13}
\]

where \( \sigma_j \) is the \( j \)th singular value of \( A^+ \).

As this distance grows toward \( \sqrt{n} \), the space of possible labelings shrinks toward zero, at which point the only feasible label vectors are close to the true labels \( y \). Equation (13) indicates that the distance increases roughly as the rank of \( A \) increases, in which case the number of non-zero singular values in \( \Sigma^+ \) increases, irrespective of how many actual weak signals are given. Thus, redundancy in the weak supervision does not affect the performance of CLL. The other key factor in the distance is how far from 0.5 the errors \( \epsilon \) are. These quantities can be interpreted as the diversity and number of the weak signals (corresponding to the rank) and their accuracies (the magnitude of \( p \)).

Though the analysis is for length-\( n \) label vectors, it is straightforwardly extended to multi-label settings with length-\( (nK) \). And with careful indexing and tracking of the abstaining indicators, the same form of analysis can apply for abstaining weak signals.

Figure 3 shows an empirical validation of Theorem 1 on a synthetic experiment. We plot the error of the labels returned by CLL and majority voting as we change the rank of \( A \). We use a synthetic data for a binary classification task with 100 randomly generated examples containing 20 binary features. The weak signals are random binary predictions for the labels of variables, we can decompose the distance into

\[
n||\Sigma^+ p|| = n \sqrt{\sigma_1^2 p_1^2 + \ldots + \sigma_m^2 p_m^2}.
\]
We report the mean and standard deviation over three trials.

Table 2: Classification accuracies of the different methods on synthetic data using independent weak signals. We report the mean and standard deviation over three trials.

<table>
<thead>
<tr>
<th>Method</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLL (Agr. rate $\epsilon$)</td>
<td>0.984 ± 0.003</td>
</tr>
<tr>
<td>CLL (Constant $\epsilon$)</td>
<td>0.978 ± 0.004</td>
</tr>
<tr>
<td>Data Programming</td>
<td>0.978 ± 0.003</td>
</tr>
<tr>
<td>Majority Vote</td>
<td>0.925 ± 0.009</td>
</tr>
<tr>
<td>CLL (True $\epsilon$)</td>
<td>0.985 ± 0.0004</td>
</tr>
<tr>
<td>Supervised Learning</td>
<td>0.997 ± 0.001</td>
</tr>
</tbody>
</table>

Table 1: Classification accuracies of the different methods on synthetic data using dependent weak signals. We report the mean and standard deviation over three trials.

4 EXPERIMENTS

We test constrained label learning on a variety of tasks on text and image classification. First, we measure the test accuracy of CLL on a synthetic dataset and compare its performance to that of supervised learning and other baselines. Second, we validate our approach on real datasets.

For all our experiments, we compare CLL to other weakly supervised methods: data programming (DP) [Ratner et al., 2016] and majority-vote (MV) or averaging (AVG). Additionally, on our real datasets we show comparison to regularized minimax conditional entropy for crowdsourcing (MMCE) [Zhou et al., 2015]. For reference, we include the performance of supervised learning baseline. On the image datasets, we show comparison of CLL to Stoch-GALL, a multiclass extension of adversarial label learning. It is worth noting that DP was developed for binary classification, thus to compare its performance on our multiclass datasets, we run DP on the weak signals that label each class in the datasets. All the weak signals on the real datasets are one-vs-all signals meaning they only label a single class and abstain on other classes.

4.1 SYNTHETIC EXPERIMENT

We construct a toy dataset for a binary classification task where the data has 200 randomly generated binary features and 20,000 examples, 16,000 for training and 4,000 for testing. Each feature vector has between 50% to 70% correlation with the true label. We define two scenarios for our synthetic experiments. We run the methods using (1) dependent weak signals and, (2) independent weak signals. In both experiments, we use 10 weak signals that have at most 30% coverage on the data and conflicts on their label assignments. The dependent weak signals were constructed by generating one weak signal that is copied noisily 9 times (randomly flipping 20% of the labels). The original weak signal labeled 30% of the data points and had an accuracy in [0.5, 0.6]. So, on average, we expect to perturb 6% of its labels on the copies. The independent weak signals are randomly generated to have accuracies in the range [0.6, 0.7].

We report in Table 1 and Table 2 the label and test accuracy from running CLL using true error rates for the weak signals, error rates estimated via agreement rate described in Section 3.3.1 and error rates using a maximum error rate constant set to 0.4 as the expected error for all the weak signals. CLL trained using the true $\epsilon$ obtains the highest test accuracy compared to the other baselines, and its performance almost matches that of supervised learning in Table 2. With the true bounds, CLL slightly outperforms CLL trained using estimated and constant $\epsilon$. More interestingly, the results in Table 1 show that our method outperforms other baselines that are strongly affected by the dependence in the weak signals. The generative model of data programming assumes that the weak signals are independent given the true labels, but this is not the case in this setup as the weak signals are strongly dependent. Thus the conditional independence violation hurts its performance and essentially reduces it to performing a majority vote on the labels.

Since our evaluation in Fig. 3 demonstrated that CLL is not very sensitive to the choice of error rate, we set the error rates $\epsilon = 0.01$ on the text datasets and $\epsilon = \frac{1}{K}$ on the image datasets. We choose these values because our weak signals in the text dataset tend to label few examples and...
Datasets | CLL | MMCE | DP | MV |
--- | --- | --- | --- | --- |
IMDB | 0.736 ± 0.0005 | 0.573 | 0.693 | 0.702 |
SST-2 | 0.678 ± 0.0004 | 0.677 | 0.666 | 0.666 |
YELP-2 | 0.765 ± 0.0002 | 0.685 | 0.770 | 0.775 |
TREC-6 | 0.842 ± 0.004 | 0.833 | 0.898 | 0.273 |

Table 3: Label accuracies of CLL compared to other weak supervision methods on different text classification datasets. We report the mean and standard deviation over three trials. CLL is trained using $\varepsilon = 0.01$ on the text classification datasets.

Datasets | CLL | MMCE | DP | MV | Supervised |
--- | --- | --- | --- | --- | --- |
IMDB | 0.740 ± 0.005 | 0.551 | 0.623 ± 0.007 | 0.724 ± 0.004 | 0.820 ± 0.003 |
SST-2 | 0.729 ± 0.001 | 0.727 | 0.720 ± 0.001 | 0.720 ± 0.0009 | 0.792 ± 0.001 |
YELP-2 | 0.840 ± 0.0007 | 0.68 | 0.760 ± 0.005 | 0.798 ± 0.007 | 0.879 ± 0.001 |
TREC-6 | 0.641 ± 0.022 | 0.64 | 0.627 ± 0.014 | 0.605 ± 0.006 | 0.700 ± 0.024 |

Table 4: Test accuracies of CLL compared to other weak supervision methods on different text classification datasets. We report the mean and standard deviation over three trials. CLL is trained using $\varepsilon = 0.01$ on the text classification datasets.

have low error rates thus we prefer not to under-constrain the optimization by using high error rates values for the one-vs-all weak-signals. In contrast, our human labeled weak signals on the image datasets have high error rates hence we set the error rate value to the baseline value for one-vs-all signals.

### 4.2 REAL EXPERIMENTS

The data sets for our real experiments and their weak signal generation process are described below. Table 7 summarizes the key statistics about these datasets. Our code and datasets are provided here.

**IMDB** The IMDB dataset [Maas et al. 2011](https://github.com/VTCSML/Constrained-Labeling-for-Weakly-Supervised-Learning) is used for sentiment analysis. The data contains reviews of different movies, and the task is to classify user reviews as either positive or negative in sentiment. We provide weak supervision by measuring mentions of specific words in the movie reviews. We created a set of positive words that weakly indicate positive sentiment and negative words that weakly indicate negative sentiment. We chose these keywords by looking at samples of the reviews and selecting popular words used in them. Many reviews could contain both positive and negative keywords, and in these cases, the weak signals will conflict on their labels. We split the dataset into training and testing subsets, where any example that contains one of our keywords is placed in the training set. Thus, the test set consists of reviews that are not labeled by any weak signal, making it important for the weakly supervised learning to generalize beyond the weak signals. The dataset contains 50,000 reviews, of which 29,182 are used for training and 20,392 are test examples.

**SST-2** The Stanford Sentiment Treebank (SST-2) is another sentiment analysis dataset [Socher et al. 2013](https://github.com/VTCSML/Constrained-Labeling-for-Weakly-Supervised-Learning) containing movie reviews. Like the IMDB dataset, the goal is to classify reviews from users as having either positive or negative sentiment. We use similar keyword-based weak supervision but with different keywords. We use the standard train-test split provided by the original dataset. While the original training data contained 6,920 reviews, our weak signals only cover 3,998 examples. Thus, we used the reduced data size to train our model. We use the full test set of 1,821 reviews.

**YELP-2** We used the Yelp review dataset containing user reviews of businesses from the Yelp Dataset Challenge in 2015. Like the IMDB and SST-2 dataset, the goal is to classify reviews from users as having either positive or negative sentiment. We converted the star ratings in the dataset by considering reviews above 3 stars rating as positive and negative otherwise. We used similar weak supervision generating process as in SST-2. We sampled 50,000 reviews for training and 10,000 for testing from the original data set. Our weak signals only cover 45,370 data points, thus, we used the reduced data size to train our model.

**TREC-6** TREC is a question classification dataset consisting of fact-based questions divided into different categories [Li and Roth 2002](https://github.com/VTCSML/Constrained-Labeling-for-Weakly-Supervised-Learning). The task is to classify questions to predict what category the question belongs to. We use the six-class version (TREC-6) from which we use 4,988 examples for training and 500 for testing. The weak supervision we use combines word mentions with other heuristics we defined to analyze patterns of the question and assign a class label based on certain patterns.

**SVHN** The Street View House Numbers (SVHN) [Netzer et al. 2018](https://github.com/VTCSML/Constrained-Labeling-for-Weakly-Supervised-Learning) dataset 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 and has 73,257 training images and 26,032 test images. We define 50 weak signals for this dataset. For this image classification dataset, we augment 40 other human-annotated weak signals (four per class) with ten pseudolabel predictions of each class from a model trained on 1% of the training data. The human-annotated weak signals are nearest-neighbor classifiers where a human annotator is asked to mark distinguishing features about an exemplar image belonging to a specific class. We then calculate pairwise Euclidean distances between the pixels in the marked region across images. We convert the Euclidean scores to probabilities (soft labels for the examples) via a logistic transform. Through this process, an annotator is guiding the design of a simple one-versus-rest classifier, where images most similar to the reference image are more likely to belong to its class.

**Fashion-MNIST** The Fashion-MNIST dataset\cite{Xiao2017FashionMNIST} represents the task of recognizing articles of clothing where each example is a 28 × 28 grayscale image. The images are categorized into 10 classes of clothing types where each class contains 6,000 training examples and 1,000 test examples. We used the same format of weak supervision signals as in the SVHN dataset (pseudolabels and human-annotated nearest-neighbor classifiers).

**Models** For the text analysis tasks, we use 300-dimensional GloVe vectors\cite{Pennington2014GloVe} as features for the text classification tasks. Then we train a simple two-layer neural network with 512 hidden units and ReLU activation in its hidden layer. The model for the image classification tasks is a six-layer convolutional neural network model with a 3 × 3 filter and 32 channels at each layer. We use a sigmoid function as the output layer for both models in our experiment. Thus we train using binary cross-entropy loss with the soft labels returned by CLL, which represent the probability of examples belonging to classes.

### Results

Tables 3 and 4 list the performance of the various weakly supervised methods on text classification datasets, while Tables 5 and 6 list the performance of various weakly supervised methods on image classification datasets. Considering both types of accuracy, CLL is able to output labels for the training data that train high-quality models for the test set. CLL outperforms all competing methods on test accuracy on the datasets. Interestingly, on Yelp and Trec-6 datasets, CLL label accuracy is lower than that of competing baselines yet CLL still achieves superior test accuracy. We surmise that CLL label accuracy is lower than competing methods on some datasets because of the inaccuracy in the error estimates. Generally, CLL is able to learn robust labels from the weak signals, and it seems to pass this information to the learning algorithm to help it generalize on unseen examples. For example, on the IMDB dataset, we used keyword-based weak signals that only occur on the training set. The model trained using CLL labels performs better on the test set than models trained with labels learned from data programming or majority vote. CLL outperforms all competing methods on the image classification tasks. On the digit recognition task (SVHN), CLL outperforms the best compared method (average) by over 13 percentage points for the label accuracy and 23 percentage points on the test data. CLL is able to better synthesize information from the low-quality human-annotated signals combined with the higher-quality pseudolabel signals.

### Table 5: Summary of datasets, including the number of weak signals used for training.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. classes</th>
<th>No. weak signals</th>
<th>Train Size</th>
<th>Test Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMDB</td>
<td>2</td>
<td>10</td>
<td>29,182</td>
<td>20,392</td>
</tr>
<tr>
<td>SST-2</td>
<td>2</td>
<td>14</td>
<td>3,998</td>
<td>1,821</td>
</tr>
<tr>
<td>YELP-2</td>
<td>2</td>
<td>14</td>
<td>45,370</td>
<td>10,000</td>
</tr>
<tr>
<td>TREC-6</td>
<td>6</td>
<td>18</td>
<td>4,988</td>
<td>500</td>
</tr>
<tr>
<td>SVHN</td>
<td>10</td>
<td>50</td>
<td>73,257</td>
<td>26,032</td>
</tr>
<tr>
<td>Fashion-MNIST</td>
<td>10</td>
<td>50</td>
<td>60,000</td>
<td>10,000</td>
</tr>
</tbody>
</table>

### Table 6: Test accuracies of CLL compared to other weak supervision methods on image datasets. We report the mean and standard deviation over three trials. CLL is trained using $\epsilon = \frac{1}{K}$ on the datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>CLL</th>
<th>MMCE</th>
<th>DP</th>
<th>AVG</th>
<th>Stoch-GALL</th>
<th>Supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVHN</td>
<td>0.575±0.001</td>
<td>0.1</td>
<td>0.42</td>
<td>0.444</td>
<td>0.196±0.025</td>
<td>0.851±0.002</td>
</tr>
<tr>
<td>Fashion-MNIST</td>
<td>0.658±0.001</td>
<td>0.147</td>
<td>0.65</td>
<td>0.649</td>
<td>0.488±0.002</td>
<td>0.852±0.003</td>
</tr>
</tbody>
</table>

### Table 7: Label accuracies of CLL compared to other weak supervision methods on image datasets. We report the mean and standard deviation over three trials. CLL is trained using $\epsilon = \frac{1}{2K}$ on the datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>CLL</th>
<th>MMCE</th>
<th>DP</th>
<th>AVG</th>
<th>Stoch-GALL</th>
<th>Supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVHN</td>
<td>0.670±0.031</td>
<td>0.1</td>
<td>0.265±0.004</td>
<td>0.432±0.001</td>
<td>0.366±0.003</td>
<td>0.851±0.002</td>
</tr>
<tr>
<td>Fashion-MNIST</td>
<td>0.695±0.002</td>
<td>0.151</td>
<td>0.635±0.0004</td>
<td>0.666±0.002</td>
<td>0.598±0.002</td>
<td>0.852±0.003</td>
</tr>
</tbody>
</table>
CONCLUSION

We introduced constrained label learning (CLL), a weakly supervised learning method that combines different weak supervision signals to produce probabilistic training labels for the data. CLL defines a constrained space for the labels of the training data by requiring that the errors of the weak signals agree with the provided error estimates. CLL is fast and converges after a few iterations of gradient descent. Our theoretical analysis shows that the accuracy of our estimated labels increases as we add more linearly independent weak signals. This analysis is consistent with the intuition that the constrained-space interpretation of weak supervision avoids overcounting evidence when multiple redundant weak signals provide the same information, since they are linearly dependent. Our experiments compare CLL against other weak supervision approaches on different text and image classification tasks. The results demonstrate that CLL outperforms these methods on most tasks. Interestingly, we are able to perform well when we train CLL using a worst case uniform error estimate for the weak signals. This shows that CLL is robust and not too sensitive to inaccuracy in the error estimates. In future work, we aim to theoretically analyze the behavior of this approach in such settings where the error rates are unreliable, with the hope that theoretical understanding will suggest new approaches that are even more robust.

Acknowledgements

We like to thank Sai Sindhura Tipirneni for her help in setting up the MMCE code. 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).

References


