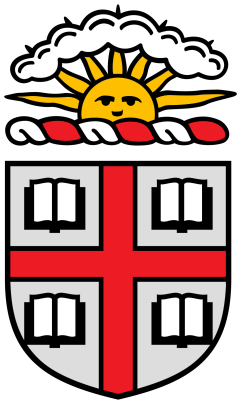
Brown University



Undergraduate Honors Thesis

Learning from Dependent Weak Supervision Sources

Dylan Sam

A thesis submitted in partial fulfillment of the requirement for honors in Computer Science and Mathematics

Thesis Advisor:
Stephen H. Bach

Thesis Reader:
Eli Upfal

April 2021

Contents

1 Abstract

Weakly supervised learning reduces the need for large amounts of labeled data to train deep learning models by using weak supervision on large quantities of unlabeled data. Many existing approaches to learn from weak supervision sources assume that they provide independent views of the target classification task. This assumption is unrealistic and frequently violated in practice. Other weakly supervised learning methods that do not make these assumptions lack guarantees on their performance. In this thesis, we present two methods with performance guarantees to learn from weak supervision sources without any such strong assumptions on their distributions.

Our first approach is limited to binary classification tasks and uses unlabeled data to estimate statistical quantities of the weak supervision sources. Then, it uses these information to solve a linear program that computes a worst-case error bound for the majority vote of a subset of weak supervision sources. Our second approach solves multi class classification tasks; we provide an adversarial training method that uses a linear program to select an adversarial labeling of the data after each gradient update.

We provide experimental results on natural image datasets for both binary and multi class classification tasks, comparing our approaches to classical methods that make independence assumptions and other recent weakly supervised learning approaches that do not provide theoretical guarantees on performance.

2 Introduction

Most of this thesis consists of two joint works: one with Alessio Mazzetto, Andrew Park, Eli Upfal, and Stephen H Bach (Mazzetto et al., 2021) that will appear in AISTATS 2021 and one with Cyrus Cousins, Alessio Mazzetto, Stephen H Bach, and Eli Upfal (Cousins et al., 2021) that is submitted and currently under anonymous review.*

Recent state-of-the-art machine learning and deep learning models have shown incredible generalization performance due to an incredibly large number of parameters. However, training these modern approaches requires large amounts of labeled data. These labeled data serve as a bottleneck to machine learning

applications in many domains, as they can be expensive or sometimes impossible to obtain. For example, medical records and EHR data exist in large quantities, but hiring experts and doctors to categorize or label these data can be extremely costly. Therefore, many fields in machine learning have attempted to leverage large amounts of unlabeled data or auxiliary forms of information to get around this bottleneck.

One such field of research is weakly supervised learning, which uses **weak supervision sources** to provide noisy information about the desired target task. A common form of weak supervision sources are hand designed heuristics, as used in (Ratner et al., 2016, 2017; Safranchik et al., 2020), which consist of simple rules that make imperfect classifications of the data. These sources of supervision tend to be less informative as humans cannot easily produce sufficiently complex rules that capture the information required for computer vision tasks. Other recent works consider more complex forms of weak supervision by training models on related tasks and transferring that external information onto the target task (Bach et al., 2017; Varma et al., 2019; Bach et al., 2019). As the underlying distributions that these models are trained on are related but inherently different, the transfer applications of these classifiers produce noisy labels that serve as a weak supervision. However, these models still learn to extract features from their original tasks, so they are usually more effective and accurate than hand-engineered heuristics.

In this work, we focus on the challenge to devise ways to smartly combine different weak supervision sources to produce a strong classifier without access to large amounts of ground-truth labels of the target task. Many existing approaches tackle this problem by making strong assumptions about the weak supervision sources (Dawid and Skene, 1979; Ratner et al., 2016; Zhang et al., 2016). A common assumption is that they provide independent views of the data, and prior work has developed optimal ways to learn a weighted vote (Nitzan and Paroush, 1982). However, this assumption is not realistic in practice and is frequently violated.

When considering hand-engineered heuristics as weak supervision, this issue can be addressed by modifying the heuristics to produce more independent forms of weak supervision. On the other hand, when weak supervision sources are models trained on related tasks, they cannot be easily modified. Consider an image classification task between different animal classes. We can train models to serve as weak supervision sources by learning high-level information about animals (i.e. color, body structure). However, resulting classifiers that learn *if an animal has flippers* and *if an animal lives in water* will capture

similar views of information, which implies that their errors will not be independent. These classifiers cannot be easily altered and thus require alternative aggregation methods that do not make such strong assumptions. Recent approaches (Arachie and Huang, 2019a,b) do not make any strong assumptions and frame this weak supervision source aggregation problem in an adversarial training setting. These methods focus on an empirical analysis of performance and do not provide theoretical guarantees about their methodologies' convergence or performance (Arachie and Huang, 2019a).

Therefore, we present two **performance-guaranteed** approaches to combine weak supervision sources. The first approach (Mazzetto et al., 2021) is restricted to binary classification tasks and uses labeled data to estimate error rates of each weak supervision source and large quantities of unlabeled data to estimate their pairwise differences or their overall output distributions. Then, it uses a linear program to analytically compute a worst-case error bound on the expected error of a majority vote of weak supervision sources. The second approach (Cousins* et al., 2021) generalizes to multi class classification tasks by an adversarial training approach. It trains a classifier by adversarially selecting labels that maximize the risk of the current classifier after each gradient update through a linear program. This linear program is constrained by possible intervals of weak supervision source accuracies; when evaluating the weak supervision sources on the selected labels, their accuracies must fall within these intervals. We present the essential methodology for these approaches and focus on the experimental details and results on various image classification tasks.

Finally, we provide a brief discussion about potential future work explaining other methods that train an additional discriminative models when using the outputs of the combination of weak supervision sources as pseudolabels for additional unlabeled data.

3 Related Work

Combining different noisy information sources has been widely studied in many different subfields of statistics and machine learning. In this section, we address prior work and discuss their limitations and differences from our methods.

The broad field of ensemble learning looks to combine different classifiers in the hopes of producing a stronger aggregated classifier. The seminal work of bagging, or bootstrap aggregating, (Breiman, 1996) trains multiple versions of classifiers using a bootstrapped sampling technique and aggregates them

together to produce a classifier that has reduced variance, especially in cases where the different samples have significant impact on the performance of the learnt classifiers. The other classical ensemble method is boosting (Schapire, 1990), which looks to aggregate a large numbers of weak classifiers to construct a strong classifier. This line of work has spawned many different approaches to boosting (Freund, 1995; Chen and Guestrin, 2016; Badirli et al., 2020) using different base hypothesis classes and optimization schemes. Both boosting and bagging use large amounts of labeled data to aggregate the weaker classifiers and use classifiers that are trained with respect to the target class. Our setting differs as the weak supervision sources can be trained on different target tasks, and we estimate the properties of our weak supervision sources with only few labeled data.

Many works in weakly supervised learning make the assumption that the weak supervision sources are independent when conditioned on the true label (Ratner et al., 2016, 2017) and use the Dawid Skene model (Dawid and Skene, 1979) to produce an optimal weighted vote under this assumption. However, many works note that this assumption is unrealistic and look to relax this independence assumption; these approaches look to estimate the structural dependencies of weak supervision sources in an unsupervised fashion (Bach et al., 2017; Varma et al., 2019). A limitation of these approaches is that they require these underlying structures to be declared. In addition, it may be difficult to verify the validity or presence of these structural dependencies in the setting with limited labels. The methods presented in this thesis do not make any independence assumptions and do not require prior knowledge about the underlying structure of the weak supervision.

Recent works have framed the task of learning from weak supervision sources without any assumptions as an adversarial learning task. The first of these methods is ALL (Adversarial Label Learning) (Arachie and Huang, 2019a), which is restricted to binary classification settings and provides a general framework for the adversarial training. This approach requires estimates of the weak supervision source errors on the target task and uses these errors as constraints to a minimax optimization problem. This approach uses a SGD training process where at each time step, a set of adversarial labels are produced that maximizes the error of the classifier that satisfies the error constraints of the weak supervision sources. The generalized ALL (Arachie and Huang, 2019b) generalizes this approach to multi class settings. These works note that if the estimates of weak supervision sources are exact, then the objective of the minimax formulation provides a upper bound on the error rate of the trained classifier. However, both methods do not provide convergence guarantees and

do not provide guarantees for a method to select a model satisfying this true minimax bound.

4 Methods

In this section, we describe the methodology presented in joint work (Mazzetto et al., 2021; Cousins* et al., 2021) which we will refer to as PGMV (Performance-Guaranteed Majority Vote) and AMCL (Adversarial Multi Class Learning) respectively.

4.1 Preliminaries

Let $\mathcal{X}$ be our domain, and let $\mathcal{Y} = \{0, 1\}$ for the setting of PGMV. For AMCL, let $\mathcal{Y}$ be the space of one-hot vectors of k classes for AMCL. Then, we can define our distribution as $D = \mathcal{X} \times \mathcal{Y}$. Let our set of n weak supervision sources be denoted as $S = \{\ell_1, ..., \ell_n\}$ where each $\ell_i : \mathcal{X} \to \mathcal{Y}$. We can consider the function ℓ_S to be the mapping of some element x in our domain to the outputs of all of our weak superivsion sources, or $\ell_S(x) = (\ell_1(x), ..., \ell_n(x))$. We note that PGMV requires that our weak supervision sources map to the same set of classes $\mathcal{Y}$ as our target task. For AMCL, we do not have this requirement and can use other additional forms of weak supervision that have been trained on a wider variety of tasks.

Let the error rates of the weak supervision sources be the vector $\epsilon = \{\epsilon_1, ..., \epsilon_n\}$, where ϵ_i represents the error rate of weak supervision source ℓ_i, or that $\ell_i = P_{x,y\sim D}(\ell_i(x) \neq y)$. Note that this vector of error rates can be estimated accurately with only few labeled data from the target task. Note that in the binary classification setting of PGMV, we can assume that each of our weak supervision sources has error $\epsilon_i \leq 0.5$. If any has an error rate greater than 0.5, we can simply reverse its outputs, resulting in a weak supervision source with error rate less than 0.5.

4.2 PGMV

We compute the error rates and pairwise differences between the weak supervision sources, which can be estimated with few labeled data and many unlabeled data. Then, we use this quantities in a linear program to obtain an analytical bound on the performance of the majority vote of a given subset of weak supervision sources.

First, we have the error rates ϵ of our weak supervision sources as defined above. These can be estimated with a small number of labeled data from the target task. Next, we consider the pairwise differences between the weak supervision sources $\mathbf{D}$ as the n by n matrix with $\mathbf{D}_{i,j} = P_{x \sim D_x}(\ell_i(x) \neq \ell_j(x))$. Let $\mathcal{S}(\epsilon, \mathbf{D})$ be the collection of all feasible sets of weak supervision sources $\{\ell_1, ..., \ell_n\}$ such that the error of ℓ_i corresponds to ϵ_i and the pairwise differences between ℓ_i and ℓ_j is equal to $D_{i,j}$. These quantities can be estimated with only unlabeled data.

Let $\mathcal{I} = \{i_1, ..., i_k\} \subseteq \{1, ..., n\}$ be some k-subset of indices of our weak supervision sources. We can denote the majority vote function of a subset as $M_{\mathcal{I}}$. Then, the majority vote of a subset of labelers is the composition of functions $M_{\mathcal{I}} \circ \ell_S$, which returns 1 if $\sum_{j \in \mathcal{I}} \ell_j > \frac{|\mathcal{I}|}{2}$. We can then define a random vector $a = \{a_1, ..., a_k\} \subseteq \{0, 1\}^k$ that represents whether a given weak supervision source in our subset $\mathcal{I}$ is correct, or that $a_j = 1$ if ℓ_{i_j} is correct and 0 otherwise. This vector will serve as our decision variables in a linear program. We can analyze the probability of the vector a occurring as

$$\begin{aligned} p_a &= P_{x \sim D_x}(a) \\ &= P_{x,y \sim D}(\{\ell_i(x) = y, \forall i \text{ s.t. } a_i = 1\} \cap \{\ell_j(x) \neq y, \forall j \text{ s.t. } a_j = 0\}) \end{aligned}$$

Then, we can compute a worst case bound on the expected error of the majority vote of a subset $\epsilon(M_{\mathcal{I}} \circ \ell_S)$ by the following linear program

$$\begin{aligned} \max_{S \in \mathcal{S}(\epsilon, \mathbf{D})} \epsilon(M_{\mathcal{I}} \circ \ell_S) = \max & \sum_{a \in \{0,1\}^k : |a|_1 < \frac{k}{2}} p_a \\ (a) & \sum_{a \in \{0,1\}^k : a_j = 0} p_a = \epsilon_{i_j} \quad \text{for } j = 1, ..., k \\ (b) & \sum_{a \in \{0,1\}^k : a_h \neq a_j} p_a = \mathbf{D}_{i_h, i_j} \quad \text{for } h \neq j \\ (c) & \sum_a p_a = 1 \\ (d) & \ p_a \geq 0, \ \forall a \end{aligned}$$

where $|a|_1$ denotes the ℓ_1 norm of the vector a. $|a|_1 < \frac{k}{2}$ denotes when less than half of the outputs are correct or when the majority vote is incorrect. We also provide an alternative linear program where we consider the output distribution of the weak supervision sources rather than their pairwise differences; we refer the reader to (Mazzetto et al., 2021) for this alternative linear program and more theoretical details.

Next, we briefly outline a greedy algorithmic approach to select good subsets of weak supervision sources to use as a majority vote. Starting from each weak supervision source in our set S, we add the two weak supervision sources that minimize the above linear program until there is no more improvement in terms of a tighter worst-case bound. In our experiments, we terminate this algorithm when we have achieved a subset of size 9. This method allows us to consider larger subsets of weak supervision sources efficiently, while only expanding our search to large subsets that achieve a tighter worst-case bound than the smaller subsets of size 3.

4.3 AMCL

Inspired by ALL, we provide a multi class approach to learn from weak supervision sources that also provides theoretical guarantees. Here, we change our constraint of their error rates of our weak supervision sources to potential ranges of their error rates $\delta = \{\delta_1, ..., \delta_n\}$ where each δ_i is some interval. As before, we have access to some small amount of labeled data from the target task, which allows us to estimate these quantities.

Let $\mathcal{H}$ be our hypothesis class where $\mathcal{H} = \{h_\theta | \theta \in \Theta\}$, or where each h is a parametric classifier. Let P_Θ be a projection function that maps any value $\tilde{\theta}$ into our valid parameter space Θ. Let the risk of our classifier be defined as some function R. In our work and experiments. We can compute an empirical estimate of a the risk with respect to a loss function L as

$$\hat{R}(h_\theta) = \frac{1}{m} \sum_{i=1}^{m} L(h_\theta(x_i), y_i)$$

Then, when training our model, we have access to a some sample of unlabeled data points $X = \{x_1, ..., x_m\}$ that have some ground truth labels $Y = \{y_1, ..., y_m\}$ that we cannot access. We can define the set of feasible labelings for this sample as Y^*, where Y^* is constrained by our quantities δ. In other words, we have that

$$Y^* = \{Y' \subset \mathcal{Y} | \hat{R}(\ell_i(X), Y') \in \delta_i\}$$

Then, we can formulate the goal of our work as a minimax problem, where we want to learn the parameters $\hat{\theta}$ that minimizes the empirical risk when given an adversarial labeling of our unlabeled data. This is formally written as

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \max_{Y \in Y^*} \hat{R}(h_\theta(X), Y) \tag{1}$$

We provide the pseudocode for a subgradient method to estimate this quantity. The pseudocode requires computing the adversarial labeling of our unlabeled dataset, which can be formulated as

$$f(\theta) = \max_{Y' \in Y^*} \hat{R}(h_\theta(X), Y')$$

and can be computed via a linear program. We refer readers to (Cousins* et al., 2021) for technical details and theoretical analyses.

Algorithm 1 AMCL Subgradient Method

Input: Number of iterations T, step size h, $\mathcal{H}$, X, $\delta = \{\delta_1, ..., \delta_n\}$
Output: Approximate solution $\tilde{\theta}$ of (1)
$\tilde{\theta}^{(0)} = \theta^{(0)} \leftarrow$ arbitrary point $\theta \in \Theta$
for $t \in 1, \ldots, T$ **do**
  $Y' \leftarrow \arg\max_{Y \in \mathbb{Y}} R(h_{\theta^{(t-1)}}, Y)$
  $v \leftarrow$ arbitrary vector from $\partial R(h_\theta, Y')$
  $\theta^{(t)} \leftarrow P_\Theta(\theta^{(t-1)} - hv)$
  $\tilde{\theta}^{(t)} \leftarrow \arg\min\{f(\tilde{\theta}^{(t-1)}), f(\theta^{(t)})\}$
end for
Return $\tilde{\theta}^{(T)}$

In our experiments, we focus on two particular hypothesis classes: one is a weighted vote of the weak supervision sources and the other is a multinomial logistic regression model that has access to a modified feature representation of the underlying domain $\mathcal{X}$. When optimizing our weighted combination of supervision sources, we use the Brier loss. When training our logistic regression model, we use the cross entropy loss.

5 Experiments

We provide experimental results of our methods on multiple datasets of image classification tasks. We provide experiments on binary classification tasks for PGMV and both binary and multi class classification tasks for AMCL. We compare our performance-guaranteed approaches with existing crowdsourcing, semi-supervised learning, and weakly supervised learning approaches.

Our PGMV approach when considering subsets of three weak supervision sources that minimizing the worst-case error is denoted as **PGMV** (Performance-Guaranteed Majority Vote). Our greedy algorithmic extensions are denoted

by **PGMV-P** and **PGMV-D**, where P denotes using pairwise differences and D denotes using distributions to constrain our linear program. The code for the experiments of our PGMV approach is available online.[1]

We refer to our algorithms for AMCL by using the acronyms **AMCL-CC** and **AMCL-LR**. AMCL-CC is an implementation of our method that uses a weighted combination of the weak supervision sources as the hypothesis class, and AMCL-LR uses multinomial logistic regression. For every image, we perform a feature extraction on the raw pixels by using the output of a pretrained ResNet-18 (He et al., 2016) as inputs for AMCL-LR. The code for the experiments of our AMCL approach will soon be available online.[2]

5.1 Baselines and Related Algorithms

We describe the various baselines that we compare against our methods.

Majority Vote (MV): The majority vote predicts the most common label among the weak supervision sources' outputs. If there is a tie, we randomly assign the prediction to one of the majority voted classes. We note that this method performs well on tasks that have where weak supervision sources have conditionally independent outputs but potentially fails when there are complex dependencies between them.

Majority Vote with Flips (MV Flip): *This method only works in the binary classification setting.* Since our PGMV methods require that each weak supervision source has an error $\epsilon < 0.5$, we flip the votes of any source with estimated accuracy less than 50%. We consider the resulting majority vote with flipped weak supervision sources to understand the impact of this flipping on the resulting accuracies.

Best Weak Supervision Source (Best WSS): We report the accuracy of the best weak supervision source.

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

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

[2] `https://github.com/BatsResearch/cousins-icml21-code` - to be released soon!

training data available is also used for learning and solves the optimization problem using stochastic gradient descent.

Adversarial Label Learning (ALL): Adversarial Label Learning (Arachie and Huang, 2019a,b) is a weakly supervised learning approach that trains a model in an adversarial fashion. This process is similar to our work since it uses bounds on the accuracies of weak supervision sources to constrain the solution space of the adversary. In our experiments, ALL trains a logistic regression model on the outputs of the weak supervision sources in our PGMV experiments and on the feature transformation of the input space for our AMCL experiments. This is a much more complex hypothesis class than the class considered by PGMV.

5.2 Datasets

Our experiments consist of several binary classification tasks from the Animals with Attributes 2 dataset (Xian et al., 2018) and multi class classification tasks from the DomainNet dataset (Peng et al., 2019).

Animals with Attributes 2 is a common benchmark for zero-shot learning and transfer learning tasks, which we will refer to as AwA2. It consists of 37,322 natural images of 50 animals classes. Each animal class is annotated with a feature representation consisting of 85 attributes, which we leverage to create our weak supervision sources. Animals with Attributes 2 is divided into 40 "seen" classes and 10 "unseen" classes, where the seen classes can be used to train attribute classifiers without leaking information about the unseen classes.

We perform binary classification on every pair of the 10 unseen classes to create 45 tasks. For all of our 45 image classification experiments, we split our unseen class data into train and test data with an even 50-50 split. We then use the train data to evaluate the accuracies of our weak supervision sources, and use their outputs on the test data to select our model and to perform evaluation. We report our experimental results by grouping the 45 different tasks based on the quality of the weak supervision sources, which we measure by committee potential Φ (Berend and Kontorovich, 2015), which is defined as

$$\Phi = \sum_{i=1}^{n} (p_i - \frac{1}{2}) \log \frac{p_i}{1 - p_i}$$

where p_i is the probability that a weak supervision source is correct.

We note that high committee potentials correspond to more potential improvement from aggregation if the independence assumption is true, and heuristically captures the difficulty of the problem by looking at a vector of accuracies as a group of weak supervision sources. Low committee potentials correspond to harder tasks, where the majority vote of all of the weak supervision sources will not perform as well. We sort the tasks by increasing committee potential and create five equally sized groups of 9 tasks to report our results. These five groups have tasks that contain ranges of committee potential scores of [1, 5.5], [6.5, 12], [12, 16.5], [18, 24.5], and [25, 61] respectively.

DomainNet is another common benchmark for transfer learning tasks. The dataset contains 345 different classes of images such as airplane, ball, cup, foot, etc. The dataset has images in 6 different domains, or different modalities of the images, which are Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. To reduce the complexity of the classification task, we randomly sample 5 classes among the 25 classes of DomainNet with the largest number of data from that class. We perform our experiments on two disjoint samples of test classes. In our experiments, the first sample contains classes sea turtle, vase, whale, bird, and violin. The second sample contains classes tornado, trombone, submarine, feather, and zebra.

5.3 Weak Supervision Sources

To create weak supervision sources for our various classification tasks on AwA2, the seen classes are used to train attribute detectors. These classifiers are learnt to detect attributes like stripes, flippers, quadruped, etc. Each detector is a pretrained ResNet-18 (He et al., 2016) with two fine-tuned linear layers. These attribute detectors must transfer high-level concepts of attributes from seen classes to unseen classes, giving noisy labels as our sources of weak supervision. For example, one particular source attempts to transfer the knowledge of humpback whales and other seen classes having flippers to *different* classes such as seals and other classes not seen at training time having flippers.

For DomainNet, we train a multi class classifier for the 5 test classes in a particular domain. Again, our weak supervision sources consist of fine-tuning pretrained ResNet-18s with two linear layers. We use 60% of the labeled data for that domain to train our weak supervision source, and the other 40% is used for evaluating the learnt AMCL model at test time. This results in 6 different classifiers that we will use as weak supervision sources when evaluating on other domains. For example, to evaluate the performance on the Clipart domain, we consider the classifiers trained in 5 other domains other than Clipart as weak

supervision sources. We remark that these weak supervision sources never have access to samples from the Clipart domain and thus provide noisy forms of supervision.

5.4 Figures

We now provide figures of our experimental results of our various approaches on AwA2 and DomainNet tasks.

5.4.1 PGMV

First, we present the results of PGMV when using all of the available labeled data to estimate weak supervision source accuracies in Table 1.

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

Table 1: Comparison of our methods and benchmarks on various image classification tasks. Numbers are accuracy percentages reported as mean ± standard error, computed over 5 random seeds. The number after AwA2 represents the which fifth of the AwA2 binary classification tasks when sorted by committee potential (least to greatest).

We can see that our best method is on average 1 percentage point more accurate than a state-of-the-art weakly-supervised approach (ALL) and 5 percentage points more accurate than the classical Dawid Skene model on tasks that have more inaccurate weak supervision sources (lowest cp). Therefore, on these hardest tasks, we significantly outperform the classical baseline that assumes independence and are very close to the alternative approach that does not provide performance guarantees.

PGMV and its variants also are roughly 1 percentage point higher than the state-of-the-art alternative's accuracy and achieve within almost 1 percentage point of the Dawid Skene's accuracy when weak supervision source accuracies are high (high cp). On these easier tasks, our methods start to outperform the state-of-the-art alternative and are close to the methods that assume independence.

5.4.2 PGMV with Varying Amounts of Labeled Data

We also perform experiments where we vary the amount of labeled data used to estimate weak supervision source accuracies to analyze the sensitivity of PGMV to amounts of labeled data. When labeled data is very limited, there is a greater chance of having inaccurate error estimates of weak supervision sources. We note that for our method, when weak our error estimates are very bad, the constraints on the linear program are sometimes not satisfied for different subsets of weak supervision sources. In these cases, the worst case bound cannot be computed for a subset of weak supervision sources, so our algorithm ignores these subsets.

In Figures 1 - 3, we compute accuracies by averaging over 3 splits of labeled and unlabeled data, and the error bars represent the standard error. Again, we report our results as groups of AwA2 pairs by committee potential as in Table 1 above. We omit the MV and MV Flip results on our figures as their performance remains relatively constant with more labeled data and are almost uniformly beaten by our methods, DS, and ALL. The rightmost point on each figure is the value in Table 1 and is averaged over 5 seeds. We leave the remaining two figures of the second and third groups in Section 7.

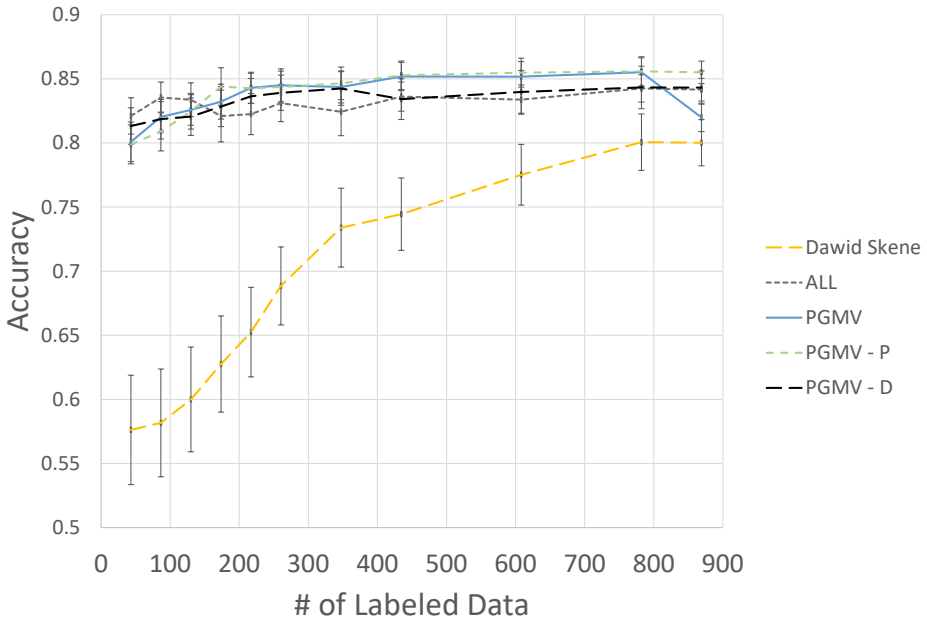


Figure 1: Results are averaged over the first fifth of AwA2 tasks when sorted by committee potential (least to greatest)

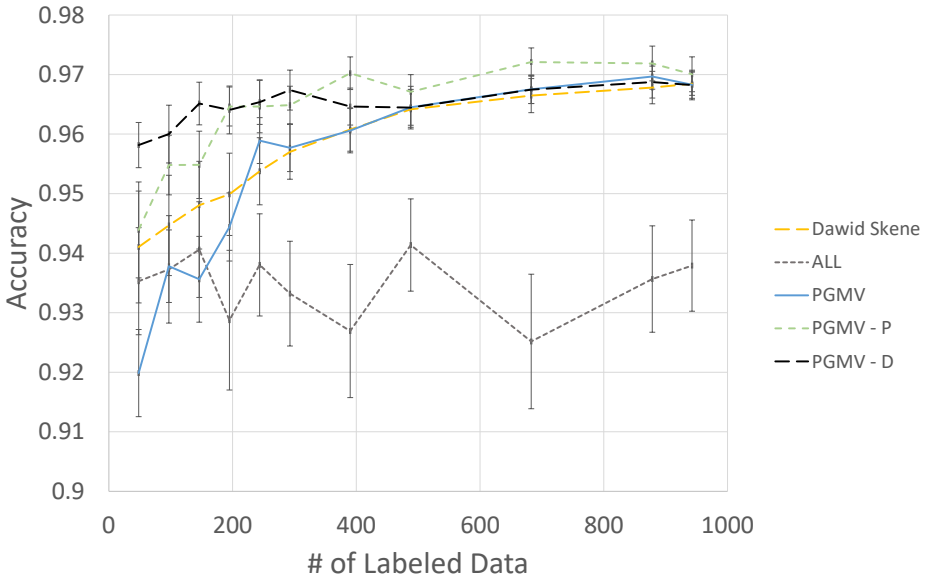


Figure 2: Results are averaged over the fourth fifth of AwA2 tasks when sorted by committee potential (least to greatest)

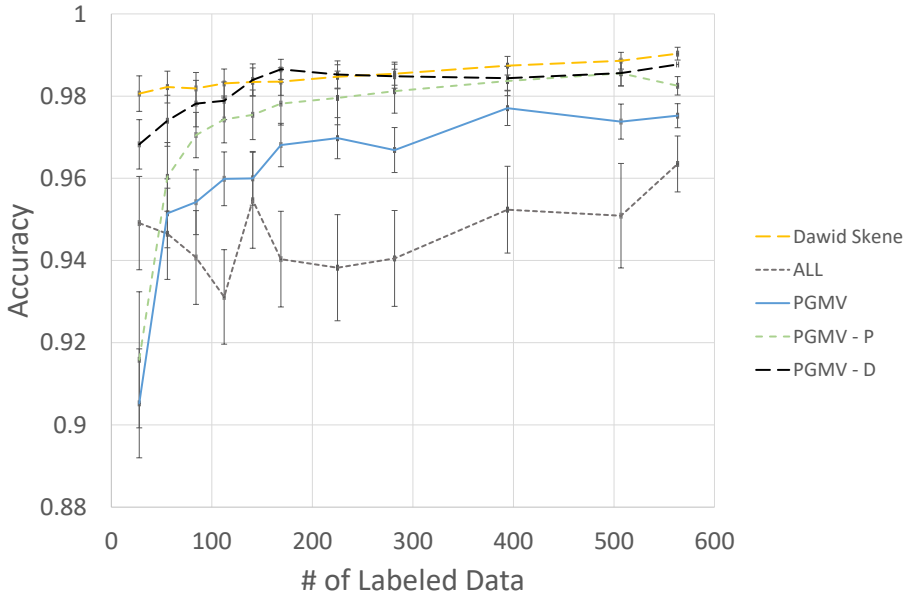


Figure 3: Results are averaged over the final fifth of AwA2 tasks when sorted by committee potential (least to greatest)

In Figure 1, which contains the tasks with the most inaccurate weak supervision sources, our methods outperform the semi-supervised Dawid Skene baseline, achieving 20 percentage points higher on average for small amounts of labeled data and roughly 4 percentage points higher when there is large amounts of labeled data. Our methods seem to have similar performance to ALL, where PGMV-P slightly seems to outperform the baseline.

As we decrease the difficulty of tasks, we see that the classical Dawid Skene model approaches our performance, and finally slightly outperforms our work in Figure 3 by only 1 percentage point. However, we also note that our methods start to outperform ALL, especially in Figure 2 and 3, achieving roughly 3 accuracy points higher when there is sufficient labeled data.

5.4.3 AMCL

Next, we present the results of our AMCL methods on both binary classification tasks on AwA2 and multi class classification tasks on DomainNet.

We report our methods' results against the binary classification task baselines on the four hardest AwA2 tasks when measured by MV accuracy. We present two of the tasks in Figures 4 - 5 and report the remaining two figures in Section 7. These are individual tasks, contrasting to averaging over nine tasks as in the earlier experiments.

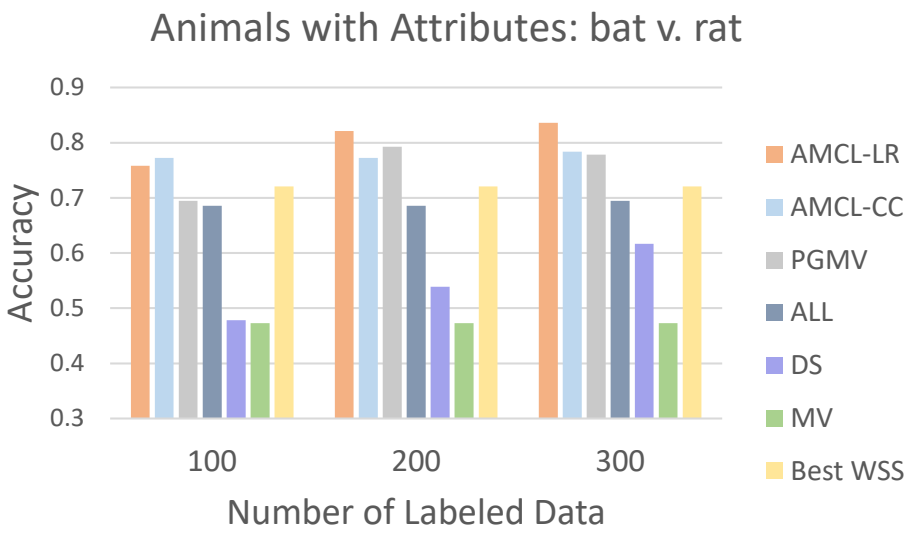


Figure 4: Results on the AwA2 binary classification tasks of bat vs. rat

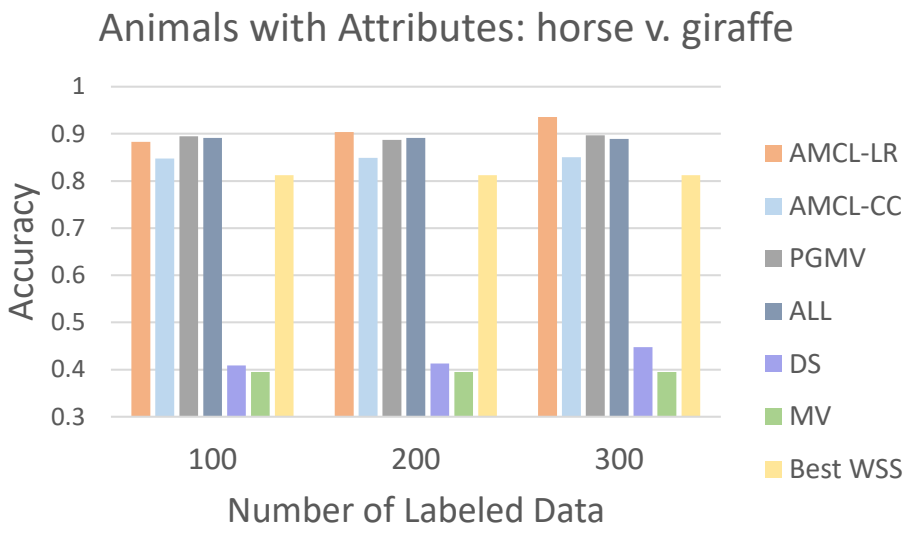


Figure 5: Results on the AwA2 binary classification tasks of horse vs. giraffe

In the binary setting, AMCL-LR matches or outperforms PGMV and the state-of-the-art method ALL over all amounts of labeled data. We note that even though AMCL-LR and ALL use the same inputs and train the same prediction model (multinomial logistic regression), our method achieves overall higher accuracies in addition to providing theoretical guarantees on the learning of the prediction model. Also, on the bat vs. rat classification task, the AMCL-CC model achieves better performance than ALL.

In Figures 6 - 7, we report AMCL-CC's results on multi class classification tasks. On each domain, the models are selected using 500 unlabeled data. Since these are many more classes, it is difficult to learn a multinomial logistic regression model with a large number of parameters with our amount of unlabeled data, so we only report AMCL-CC. Since ALL is restricted to the binary setting, we compare against DS, MV, and Best WSS baselines. We present our results on Clipart and Infograph and report the remaining figures in Section 7.

In the multi class setting, our methods again match or outperform the baselines over all amounts of labeled data on the Clipart and Infograph domains. We note that in the second sample of classes in the Infograph domain, the weak supervision sources are overall very inaccurate (with Best WSS at 28% accuracy), and it is difficult to recover useful information from them. However, differently from the baselines DS and MV, AMCL-CC can still recover similar performance from the best weak supervision source.

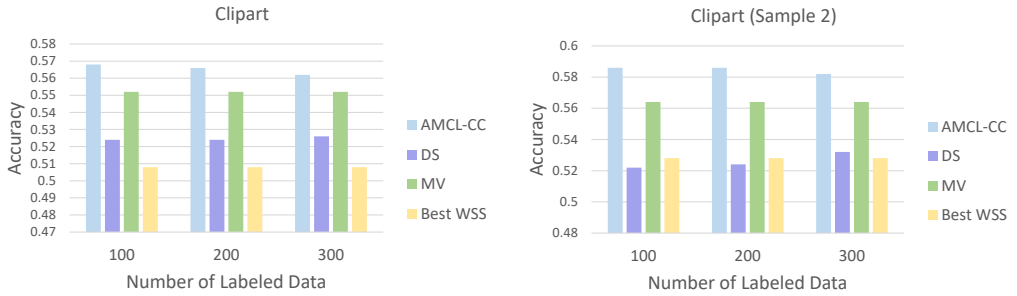


Figure 6: Results on the DomainNet 5 class classification tasks on the Clipart domain (sample 1 - left, sample 2 - right)

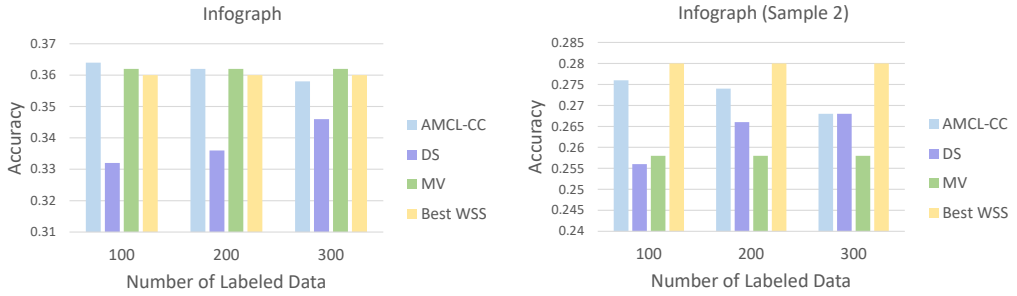


Figure 7: Results on the DomainNet 5 class classification tasks on the Infograph domain (sample 1 - left, sample 2 - right)

6 Discussion

In this thesis, we provide two methods to learn from weak supervision sources without making any assumptions on their distributions. While existing baselines either make strong assumptions or do not have theoretical guarantees, our methods provide an analytical bound on a classifier's performance. We extend past our original binary classification approach (PGMV) to a multi class approach using adversarial training (AMCL). We compare our methods to many state-of-the-art baselines and classical methods that make strong independence assumptions on two large image classification datasets. Our two methods have performance that matches or sometimes outperforms other existing state-of-the-art methods and also have guarantees on their performance, regardless of the distribution of weak supervision sources.

A future problem of interest is considering alternative methods to use weak supervision sources. While our PGMV approach focuses on learning a model on the outputs of the weak supervision sources, our AMCL method can train

a complex classifier on the data domain or some transformation of the domain. Another potential approach is to use these models that aggregate weak supervision source outputs (like Dawid Skene or PGMV), which we refer to as a **labelmodel**, to pseudolabel more unlabeled data points and train a second discriminative model, which we refer to as an **endmodel**. This is the standard approach in Snorkel, and we observe a considerable increase in performance in the endmodel. However, there are no any formal statements that provide guarantees for this phenomena.

One recent related work (Wei et al., 2021) proves the ability of neural networks to de-noise complex pseudolabelers, under the conditions of an expansion assumption and using input consistency training during the optimization process. This provides a potential explanation in the importance of a particular type of regularization to improve upon a noisy pseudolabeler. Another recent work (Xie et al., 2021) focuses on the setting for the least-squares model in regression tasks, showing that using pseudolabeling from learning auxiliary tasks can improve the generalization performance of the OLS model on out-of-distribution data. These works seem to provide results in related settings, and similar concepts may be able to transfer over to the training of an endmodel.

We believe that some form of regularization, such as L_2-regularization, along with an informative enough data representation (or data augmentation scheme) would allow a endmodel to outperform a labelmodel given enough pseudolabeled examples. While this is similar to the ideas in (Wei et al., 2021) and benefits of such regularization are described in (Liu et al., 2020), we also believe that this would hold in scenarios where the labelmodel is learned on simple hand-designed heuristic functions like those provided in the Snorkel framework rather than a complex, noisy pseudolabeler.

We provide a very high-level description of our thought process in the regression setting, under the mean squared error risk. Let $\mathcal{X}$ be our data domain, where $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 = \mathbb{R}^{d_1+d_2}$. Consider a regression task, where our distribution D is over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y} \subset \mathbb{R}$. Then $\mathcal{X}_1$ represents the smaller subset on which our labelmodel is learned, or the smaller set of features analyzed by the hand-designed heuristics. Let g represent our labelmodel, where $g(x) = g(x_1, \cdot)$ as it only looks at the first d_1 dimensions. Then, our goal is to train an endmodel h, which we learns a mapping $h : \mathcal{X} \to \mathcal{Y}$, and is trained on datapoints $\{x_i, g(x_i)\}_{i=1}^n$ for some n training samples.

First, we will denote the risk of our labelmodel as $R(g)$ taken with respect to mean squared error. Assuming that our labelmodel is some *unbiased* estimator

g, we have that the risk is given by

$$\begin{aligned} R(g) &= E[(g(x) - y)^2] \\ &= E[(g(x) - \overline{g(x)})^2] + 2E[(g(x) - \overline{g(x)})(\overline{g(x)} - y)] + E[(\overline{g(x)} - y)^2] \\ &= Var(g) + E[(\overline{g(x)} - y)^2] \end{aligned}$$

where $\overline{g(x)}$ is the main prediction (or where $\overline{g(x)} = E_{S \sim D}[g(x_S)]$) as presented in (Domingos, 2000). Next, we can analyze the risk of our endmodel as $R(h)$, which has that

$$\begin{aligned} R(h) &= E[(h(x) - y)^2] \\ &= E[(h(x) - \overline{h(x)} + \overline{h(x)} - y)^2] \\ &= Var(h) + 2E[(h(x) - \overline{h(x)})(\overline{h(x)} - y)] + E[(\overline{h(x)} - y)^2] \end{aligned}$$

Since we train h on the outputs of g, we don't have the same decomposition into only bias and variance as $E[\overline{h(x)}] \neq y$. In addition, since we are training a regularized model (such as ridge regression), we also have that $E[\overline{h(x)}] \neq g(x)$. We can consider the rightmost term by adding and subtracting the outputs of our labelmodel g:

$$\begin{aligned} E[(\overline{h(x)} - y)^2] &= E[(\overline{h(x)} - g(x) + g(x) - y)^2] \\ &= E[(\overline{h(x)} - g(x))^2] + 2E[(\overline{h(x)} - g(x))(g(x) - y)] + R(g) \end{aligned}$$

Then, we get that the complete risk of our classifier has that

$$\begin{aligned} R(h) = Var(h) + 2E[(h(x) - \overline{h(x)})(\overline{h(x)} - y)] + E[(\overline{h(x)} - g(x))^2] + \\ 2E[(\overline{h(x)} - g(x))(g(x) - y)] + R(g) \end{aligned}$$

With sufficient regularization (large enough value for λ in ridge regression), $Var(h)$ is small and $h(x) \approx \overline{h(x)}$, implying that we need that

$$R(h) = R(g) + E[(\overline{h(x)} - g(x))^2] + 2E[(\overline{h(x)} - g(x))(g(x) - y)] \tag{2}$$

With enough regularization and an informative enough representation of $\mathcal{X}_2$, our classifier h can recover more of the true labels, which would making the result of Equation 2 less than $R(g)$.

This serves a high-level description of our thoughts and the beginning of a potential explanation of this phenomena. Future work looks to provide more rigorous mathematical derivations for this method and to find what particular notion of a good data representation or data augmentation strategies will allow for the boosted performance of an endmodel.

Acknowledgements

I want to thank my research advisor Stephen Bach for his wonderful advice and for motivating me to study machine learning. Working with you and the BATS Lab has encouraged me to pursue graduate school and a career in research. I also would like to thank my reader Eli Upfal and my amazing collaborators including Alessio Mazzetto, Cyrus Cousins, Andrew Park, and the rest of the LwLL TA2 group. I could not have completed this work without our weekly meetings and discussions. Finally, I cannot thank my friends and family enough for all of their support in my undergraduate journey at Brown.

References

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

Arachie, C. and Huang, B. (2019b). Stochastic generalized adversarial label learning.

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

Bach, S. H., Rodriguez, D., Liu, Y., Luo, C., Shao, H., Xia, C., Sen, S., Ratner, A., Hancock, B., Alborzi, H., Kuchhal, R., Ré, C., and Malkin, R. (2019). Snorkel DryBell: A case study in deploying weak supervision at industrial scale.

Badirli, S., Liu, X., Xing, Z., Bhowmik, A., Doan, K., and Keerthi, S. S. (2020). Gradient boosting neural networks: Grownet.

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

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Chen, T. and Guestrin, C. (2016). Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Cousins*, C., Mazzetto*, A., Sam, D., Bach, S. H., and Upfal, E. (2021). Adversarial multi class learning under weak supervision with performance guarantees.

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

Domingos, P. (2000). A unified bias-variance decomposition for zero-one and squared loss. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285.

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

Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. (2020). Early-learning regularization prevents memorization of noisy labels.

Mazzetto, A., Sam, D., Park, A., Upfal, E., and Bach, S. H. (2021). Semi-supervised aggregation of dependent weak supervision sources with performance guarantees. In *Artificial Intelligence and Statistics (AISTATS)*.

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

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. (2019). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415.

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

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

Safranchik, E., Luo, S., and Bach, S. H. (2020). Weakly supervised sequence tagging from noisy rules. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5(2):197–227.

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

Wei, C., Shen, K., Chen, Y., and Ma, T. (2021). Theoretical analysis of self-training with deep networks on unlabeled data.

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

Xie, S. M., Kumar, A., Jones, R., Khani, F., Ma, T., and Liang, P. (2021). In-N-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. In *International Conference on Learning Representations (ICLR)*.

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

7 Additional Figures

We provide the remaining figures for our experimental results for both PGMV and AMCL on AwA2 and DomainNet.

PGMV

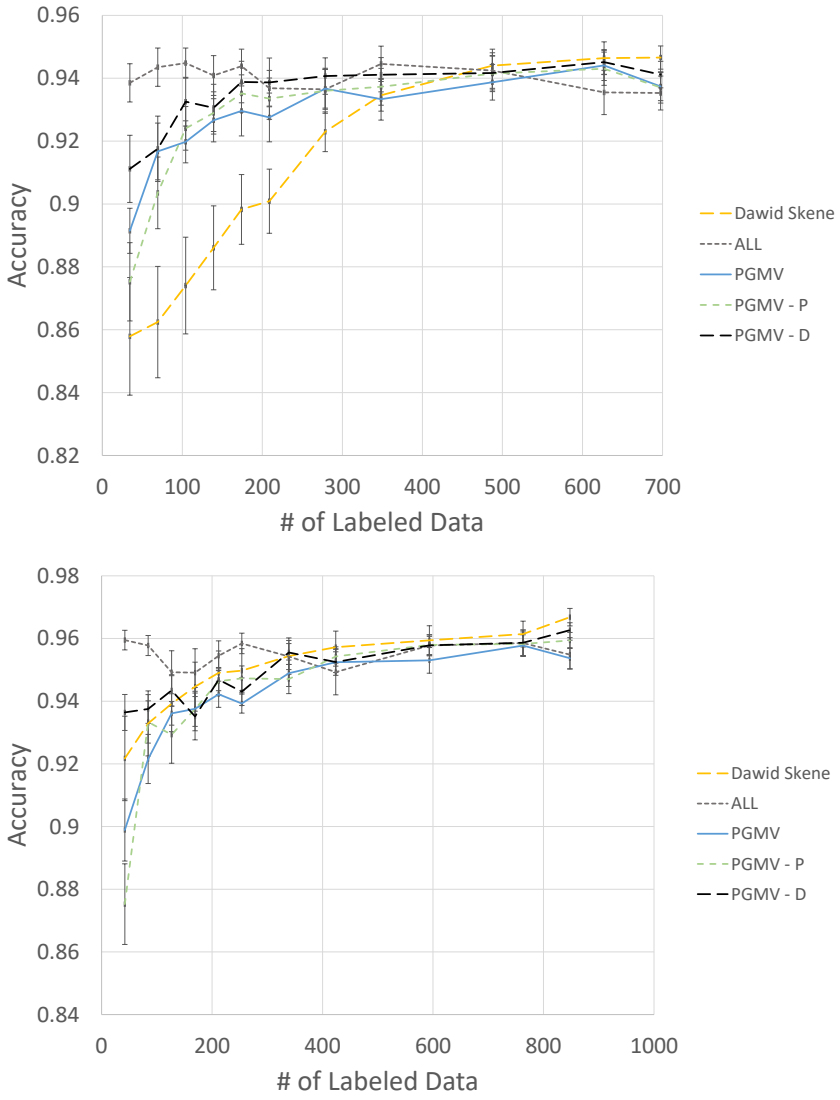


Figure 8: Results are averaged over the second fifth (top) and third fifth (bottom) of AwA2 tasks when sorted by committee potential (least to greatest)

AMCL Binary Classification

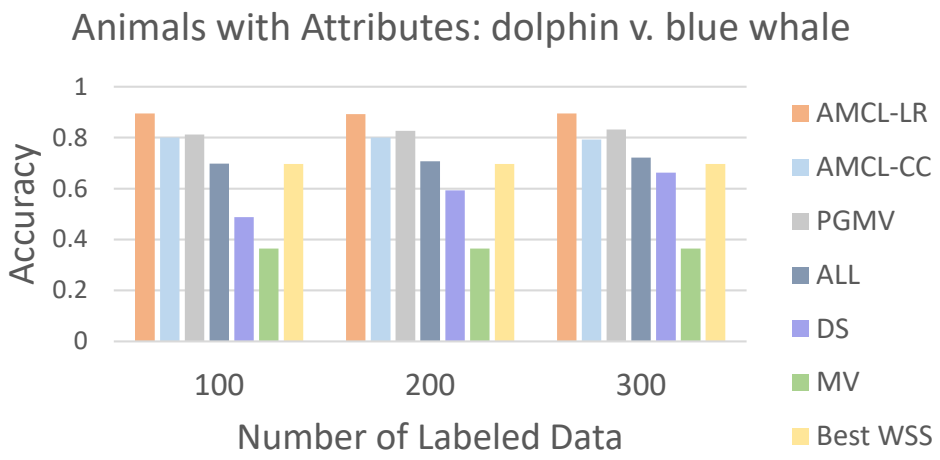


Figure 9: Results on the AwA2 binary classification tasks of dolphin vs. blue

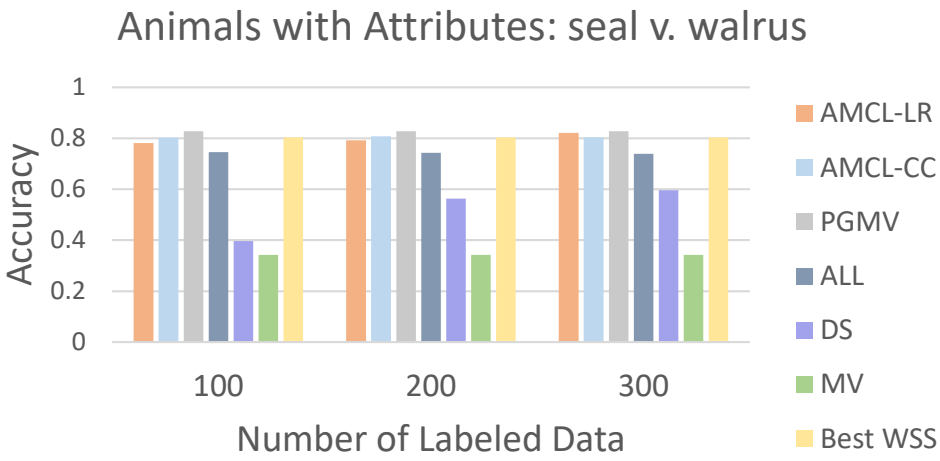


Figure 10: Results on the AwA2 binary classification tasks of seal vs. walrus

AMCL Multi Class Classification

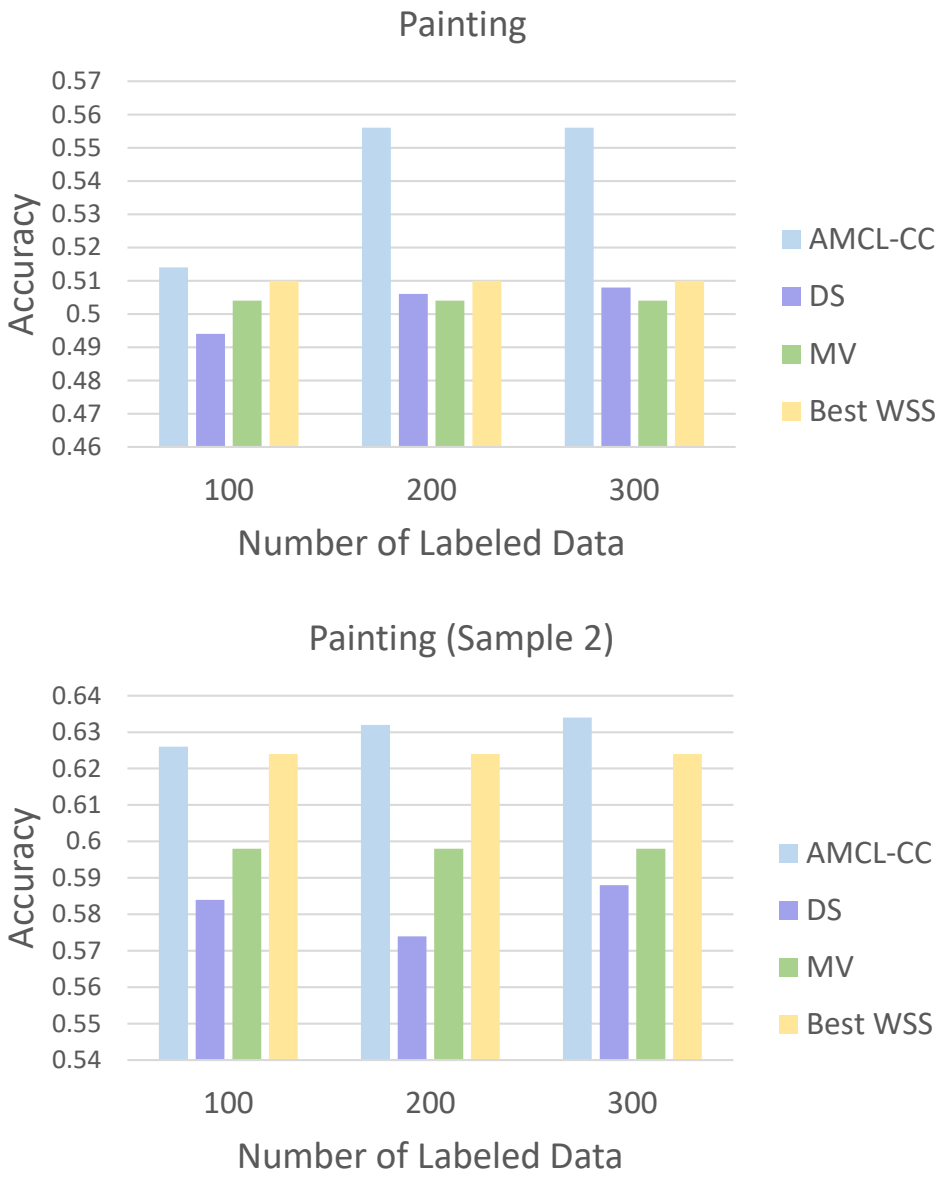


Figure 11: Results on the DomainNet on the Painting domain

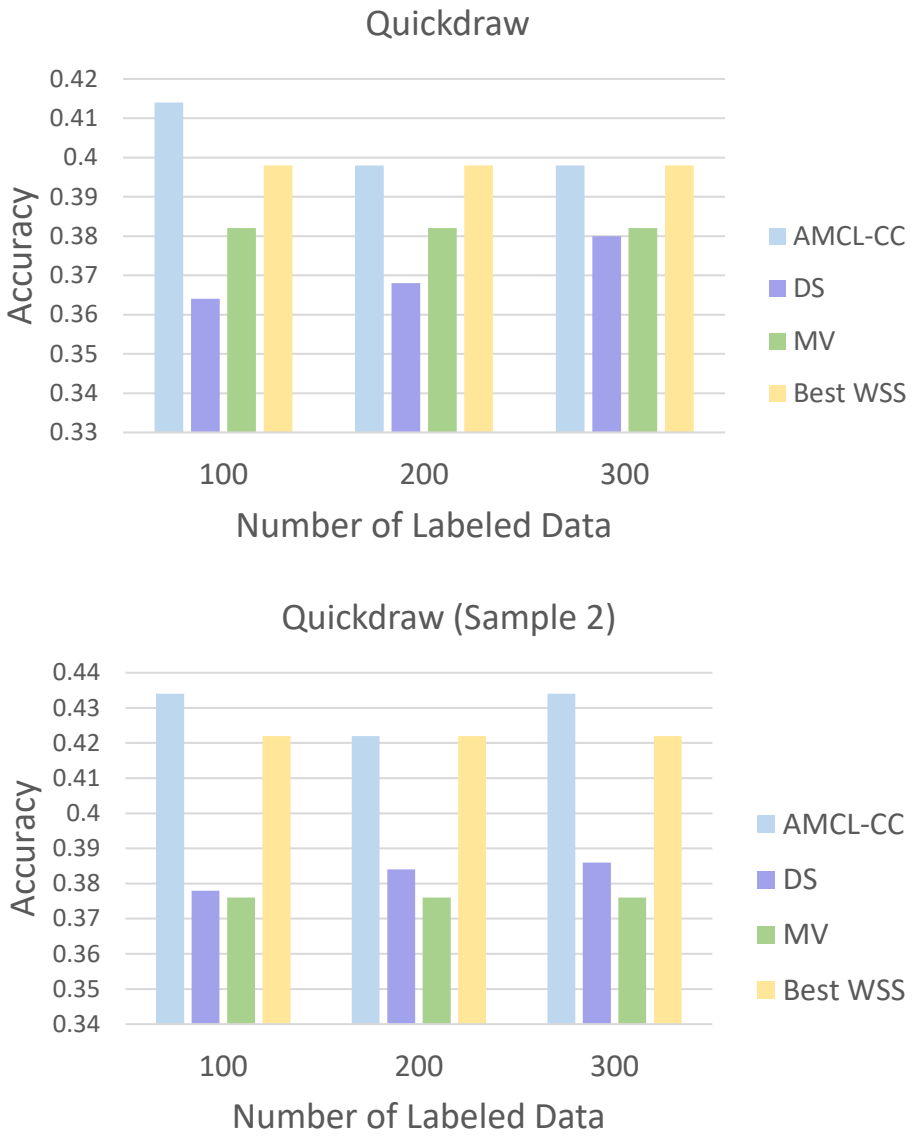


Figure 12: Results on the DomainNet on the Quickdraw domain

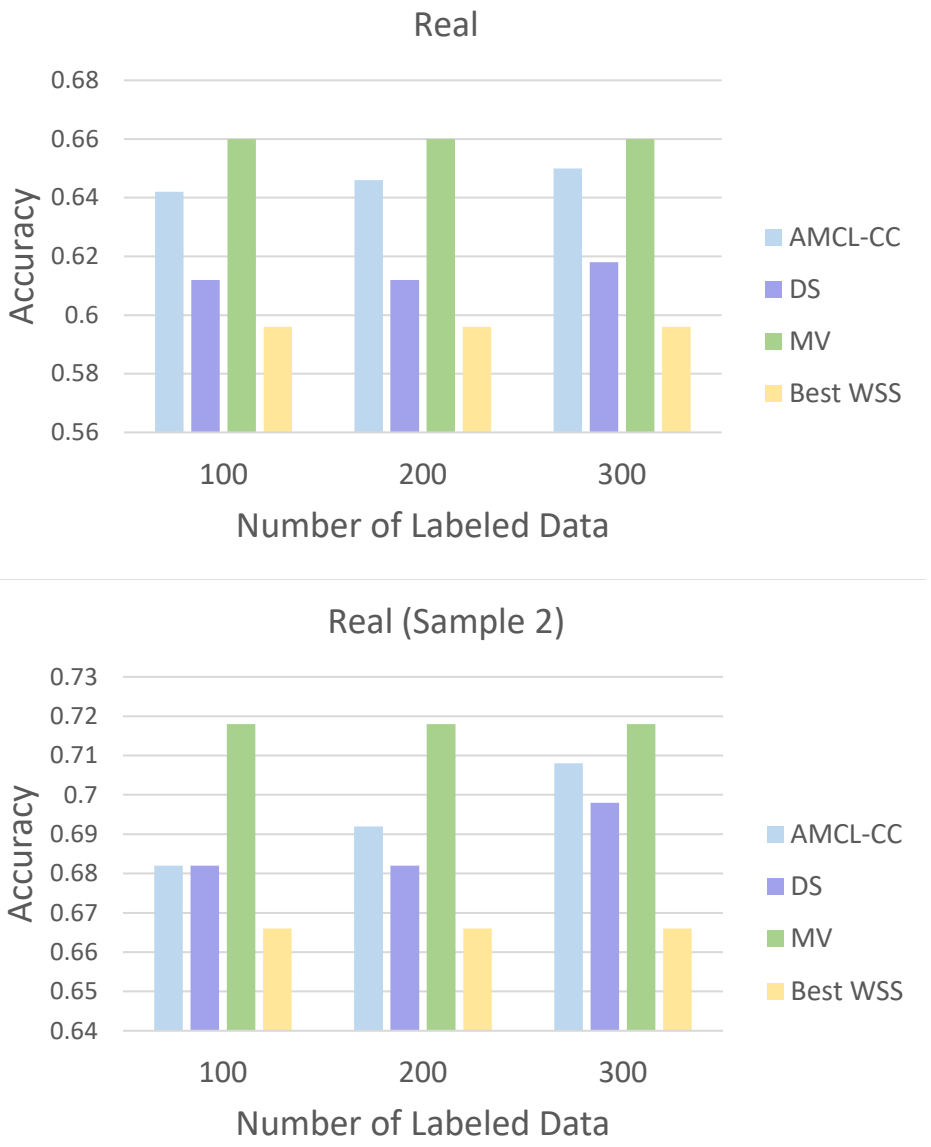


Figure 13: Results on the DomainNet on the Real domain

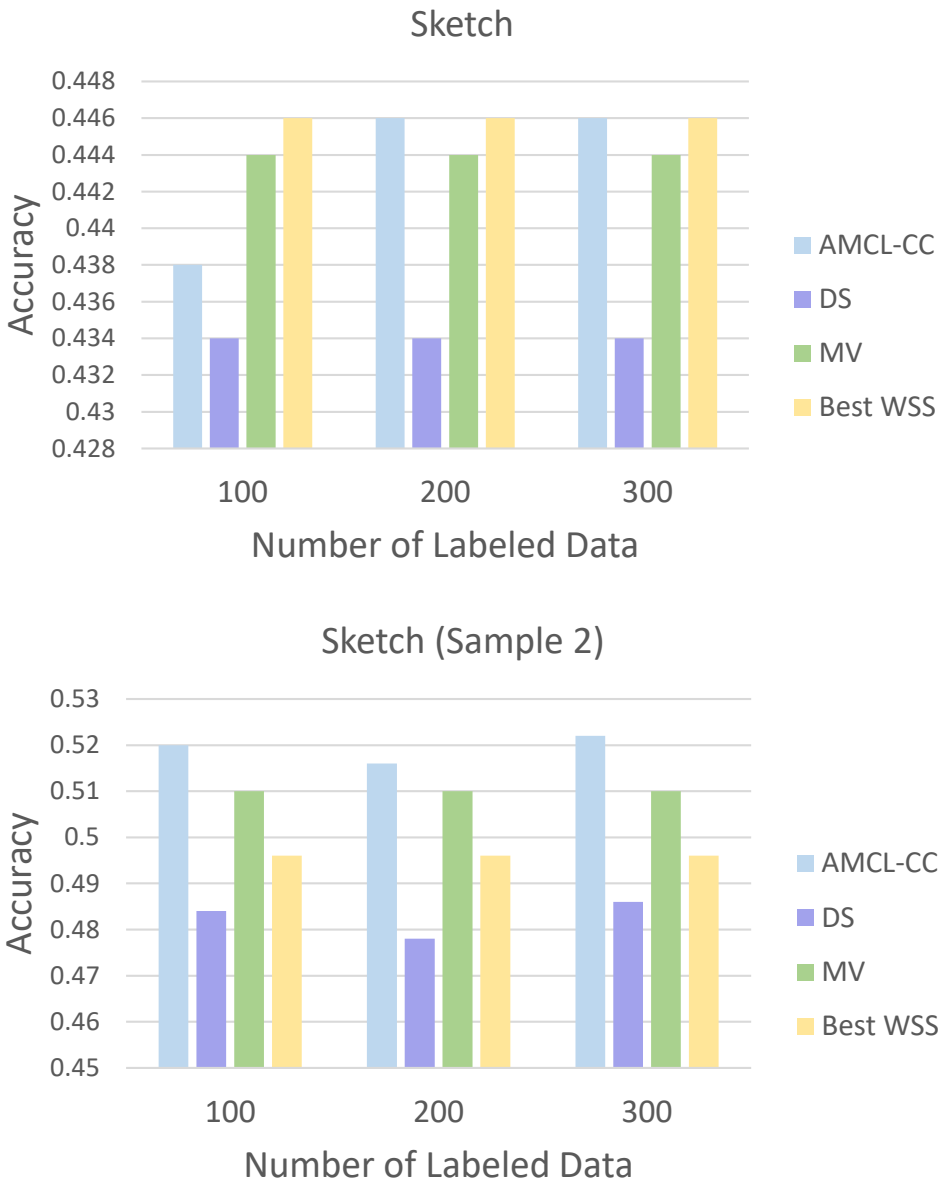


Figure 14: Results on the DomainNet on the Sketch domain