Lifting Weak Supervision To Structured Prediction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Weak supervision (WS) is a rich set of techniques that produce pseudolabels by aggregating easily obtained but potentially noisy label estimates from various sources. WS is theoretically well-understood for binary classification, where simple approaches enable consistent estimation of pseudolabel noise rates. Using this result, it has been shown that downstream models trained on the pseudolabels have generalization guarantees nearly identical to those trained on clean labels. While this is exciting, users often wish to use WS for *structured prediction*, where the output space consists of more than a binary or multi-class label set: e.g. rankings, graphs, manifolds, and more. Do the favorable theoretical properties of WS for binary classification lift to this setting? We answer this question in the affirmative for a wide range of scenarios. For labels taking values in a finite metric space, we introduce techniques new to weak supervision based on pseudo-Euclidean embeddings and tensor decompositions, providing a nearly-consistent noise rate estimator. For labels in constant-curvature Riemannian manifolds, we introduce new invariants that also yield consistent noise rate estimation. In both cases, when using the resulting pseudolabels in concert with a flexible downstream model, we obtain generalization guarantees nearly identical to those for models trained on clean data. Several of our results, which can be viewed as robustness guarantees in structured prediction with noisy labels, may be of independent interest.

1 Introduction

Weak supervision (WS) is an array of methods used to construct pseudolabels for training supervised models in label-constrained settings. The standard workflow [RSW+16, RBE+18, FCS+20] is to assemble a set of cheaply-acquired labeling functions—simple heuristics, small programs, pretrained models, knowledge base lookups—that produce multiple noisy estimates of what the true label is for each unlabeled point in a training set. These noisy outputs are modeled and aggregated into a single higher-quality pseudolabel. Any conventional supervised end model can be trained on these pseudolabels. This pattern has been used to deliver excellent performance in a range of domains in both research and industry settings [DRS+20, RNGS20, SLB20], bypassing the need to invest in large-scale manual labeling. Importantly, these successes are usually found in binary or small-cardinality classification settings.

While exciting, users often wish to use weak supervision in *structured prediction* (SP) settings, where the output space consists of more than a binary or multiclass label set [BHS+07, KL15]. In such cases, there exists meaningful algebraic or geometric structure to exploit. Structured prediction includes, for example, learning rankings used for recommendation systems [KAG18], regression in metric spaces [PM19], learning on manifolds [RCMR18], graph-based learning [GS19], and more.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

An important advantage of WS in the standard setting of binary classification is that it yields models with nearly the same generalization guarantees as their fully-supervised counterparts. Indeed, the penalty for using pseudolabels instead of clean labels is only a multiplicative constant. This is a highly favorable tradeoff since acquiring more unlabeled data is easy. This property leads us to ask the key question for this work: **does weak supervision for structured prediction preserve generalization guarantees?** We answer this question in the affirmative, justifying the application of WS to settings far from its current use.

Generalization results in WS rely on two steps [RHD+19, FCS+20]: (i) showing that the estimator used to learn the model of the labeling functions is consistent, thus recovering the noise rates for these noisy voters, and (ii) using a noise-aware loss to de-bias end-model training [NDRT13]. Lifting these two results to structured prediction is challenging. The only available weak supervision technique suitable for SP is that of [SLV+22]. It suffers from several limitations. First, it relies on the availability of isometric embeddings of metric spaces into $\mathbb{R}^d$—but does not explain how to find these. Second, it does not tackle downstream generalization at all. We resolve these two challenges.

We introduce results for a wide variety of structured prediction problems, requiring only that the labels live in some metric space. We consider both finite and continuous (manifold-valued) settings. For finite spaces, we apply two tools that are new to weak supervision. The approach we propose combines isometric *pseudo-Euclidean embeddings* with *tensor decompositions*—resulting in a nearly-consistent noise rate estimator. In the continuous case, we introduce a label model suitable for the so-called *model spaces*—Riemannian manifolds of constant curvature—along with extensions to even more general spaces. In both cases, we show generalization results when using the resulting pseudolabels in concert with a flexible end model from [CRR16, RCMR18].

Contributions:

- New techniques for performing weak supervision in finite metric spaces based on isometric pseudo-Euclidean embeddings and tensor decomposition algorithms,
- Generalizations of weak supervision for regression to manifold-valued regression in constant-curvature manifolds,
- Finite-sample error bounds for noise rate estimation in each scenario,
- Generalization error guarantees for training downstream models on pseudolabels.

2 Background and Problem Setup

Our goal is to theoretically characterize how well we can learn with pseudolabels (built with weak supervision techniques) in structured prediction settings. Specifically, we seek to understand the interplay between the noise in WS sources and the generalization performance of the downstream structured prediction model. We provide a brief background on structured prediction and weak supervision, then introduce our problem and some useful notation.

2.1 Structured Prediction

Structured prediction (SP) involves predicting labels in spaces with rich structure. Denote the label space by $\mathcal{Y}$. Conventionally $\mathcal{Y}$ is a set, e.g., $\mathcal{Y} = \{-1, +1\}$ for binary classification. In the SP setting, $\mathcal{Y}$ has some additional algebraic or geometric structure. In this work we assume that $\mathcal{Y}$ is a metric space with metric (distance) $d_{\mathcal{Y}}$. This covers many types of problems, including

- Rankings, where $\mathcal{Y} = S_\rho$, the symmetric group on $\{1, \ldots, \rho\}$, i.e. labels are permutations,
- Graphs, where $\mathcal{Y} = \mathcal{G}_\rho$, the space of graphs with vertex set $V = \{1, \ldots, \rho\}$,
- Riemannian manifolds, where $\mathcal{Y} = \mathbb{S}_d$, the sphere, or $\mathbb{H}_d$, the hyperboloid.

In conventional supervised learning we have a dataset $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ of i.i.d samples drawn from some distribution ρ over the space $\mathcal{X} \times \mathcal{Y}$.

Learning and Generalization in SP As usual, we seek to learn a model that generalizes well to points not seen during training. Let $\mathcal{F} = \{f : \mathcal{X} \mapsto \mathcal{Y}\}$ be a family of functions from $\mathcal{X}$ to $\mathcal{Y}$. Define the risk $R(f)$ for $f \in \mathcal{F}$ and f^* as

$$R(f) = \int_{\mathcal{X}\times\mathcal{Y}} d^2_{\mathcal{Y}}(f(x), y) d\rho(x, y) \qquad f^* \in \arg\min_{f\in\mathcal{F}} R(f). \tag{1}$$

For a large class of settings (including all of those we consider in this paper), [CRR16, RCMR18] have shown that the estimator $\hat{f}$ defined below approaches f^*:

$$\hat{f}(x) = \arg\min_{y\in\mathcal{Y}} F(x, y) \qquad F(x, y) := \frac{1}{n}\sum_{i=1}^{n} \alpha_i(x) d^2_{\mathcal{Y}}(y, y_i), \tag{2}$$

where $\alpha(x) = (\mathbf{K} + \nu\mathbf{I})^{-1}\mathbf{K}_x$. Here, $\mathbf{K}$ is the kernel matrix for a p.d. kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, so that $\mathbf{K}_{i,j} = k(x_i, x_j)$, $(\mathbf{K}_x)_i = k(x, x_i)$, and ν is a regularization parameter. Thus it is necessary to first learn the weights α and then to perform the optimization in (2) to make a prediction.

When there is no label noise, an exciting contribution of [CRR16, RCMR18] is the generalization bound

$$R(\hat{f}) \leq R(f^*) + \mathcal{O}(n^{-\frac{1}{4}}),$$

that holds with high probability. The key question we tackle is *does the use of pseudolabels instead of true labels y_i affect the generalization rate?* Note that even having access to the kernel and thus knowing the weights α is insufficient to ensure this; the presence of noise when replacing y_i with a pseudolabel could ostensibly ruin the generalization bound.

2.2 Weak Supervision

In WS, we cannot access *any* of the ground-truth labels y_i. Instead we observe for each x_i the noisy votes $\lambda_{a,i}, \ldots, \lambda_{m,i}$. These are m weak supervision outputs provided by *labeling functions* (LFs) s_a, where $s_a : \mathcal{X} \to \mathcal{Y}$ and $\lambda_{a,i} = s_a(x_i)$. A two step process is used to construct pseudolabels. First, we learn a *noise model* (also called a label model) that determines how reliable each source s_a is. That is, we must learn $\boldsymbol{\theta}$ for $P_{\boldsymbol{\theta}}(\lambda_1, \lambda_2, \ldots, \lambda_m|y)$—without having access to any samples of y. Second, the noise model is used to infer a distribution (or its mode) for each point: $P_{\boldsymbol{\theta}}(y_i|\lambda_{1,i}, \ldots, \lambda_{m,i})$.

We adopt the noise model from [SLV+22], which is suitable for our SP setting:

$$P_{\boldsymbol{\theta}}(\lambda_1, \ldots, \lambda_m|Y = y) = \frac{1}{Z}\exp\left(-\sum_{a=1}^{m} \theta_a d^2_{\mathcal{Y}}(\lambda_a, y)\right). \tag{3}$$

This is an exponential family model, where Z is the normalizing partition function and $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_m]^T > 0$ are the *canonical* parameters. The model can also be described in terms of the *mean* parameters $\mathbb{E}[d^2_{\mathcal{Y}}(\lambda_a, y)]$. Intuitively, if θ_a is large, then the typical distance from λ_a to y must be small.In this case the LF is reliable. Conversely, if θ_a is small, the LF is unreliable.

Our goal is to form estimates $\hat{\boldsymbol{\theta}}$ in order to construct pseudolabels. One way to build such pseudolabels is to compute $\tilde{y} = \arg\min_{z\in\mathcal{Y}} 1/m \sum_{a=1}^{m} \hat{\theta}_a d^2_{\mathcal{Y}}(z, \lambda_a)$. Observe how the estimated parameters are used to weight the labeling functions θ_a, ensuring that more reliable votes receive a larger weight. The extreme cases are $\theta_a = 0$, so that λ_a is independent of y, and so gets no weight, and $\theta_a = \infty$, so that $\theta_a = y$ and should get all of the weight.

We are now in a position to state the main research question for this work:

Do there exist estimation approaches yielding $\hat{\boldsymbol{\theta}}$ that produce pseudolabels $\tilde{y}$ that maintain the same generalization error rate $\mathcal{O}(n^{-1/4})$ when used in (2), or a modified version of (2)?

3 Noise Rate Recovery in Finite Metric Spaces

In the next two sections we will handle finite metric spaces. Afterwards we tackle continuous (manifold-valued) spaces. We first discuss learning the noise parameters $\boldsymbol{\theta}$, then the use of pseudolabels in training.

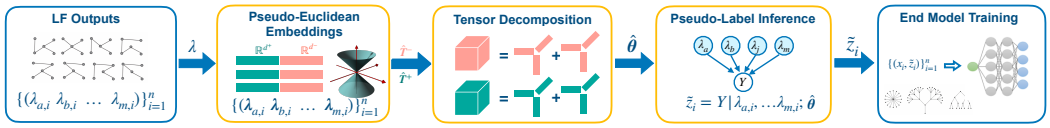

Figure 1: Illustration of our weak supervision pipeline for the finite label space setting.

Roadmap For finite metric spaces with $|\mathcal{Y}| = r$, we apply two tools new to weak supervision. First, we embed $\mathcal{Y}$ into a *pseudo-Euclidean* space [Gol85]. These spaces generalize Euclidean space, enabling isometric (distance-preserving) embeddings for any metric space. Using pseudo-Euclidean spaces make our analysis slightly more complex, but we gain the isometry property, which is critical.

Second, we form three-way tensors from embeddings of observed labeling functions. Applying tensor product decomposition algorithms [AGH+14], we can recover estimates of the mean parameters $\hat{\mathbb{E}}[d^2_{\mathcal{Y}}(\lambda_a, y)]$ and ultimately $\hat{\theta}_a$. Finally, we reweight the model (2) to preserve generalization.

3.1 Pseudo-Euclidean Embeddings

Working directly with the label space $\mathcal{Y}$ is challenging due to its potentially large cardinality. A standard way to address this challenge is to embed $\mathcal{Y}$ into a vector space. For example, multi-dimensional scaling (MDS) [KW78] embeds $\mathcal{Y}$ into $\mathbb{R}^d$. The downside of MDS is that only some metric spaces embed (isometrically) into Euclidean space. In particular, it is necessary that the square distance matrix $\mathbf{D}$ is positive semi-definite.

A simple and elegant way to overcome this difficulty is to instead use *pseudo-Euclidean* spaces for embeddings. These pseudo-spaces do not require a p.s.d. inner product. As an outcome, any finite metric space can be embedded into a pseudo-Euclidean space with *no distortion* [Gol85]—so that distances are exactly preserved. We shall need only a few properties of these spaces: A vector $\mathbf{u}$ in a pseudo-Euclidean space $\mathbb{R}^{d^+,d^-}$ has two parts $\mathbf{u}^+ \in \mathbb{R}^{d^+}$ and $\mathbf{u}^- \in \mathbb{R}^{d^-}$. The dot product and the squared distance between any two vectors $\mathbf{u}, \mathbf{v}$ in a pseudo-Euclidean space $\mathbb{R}^{d^+,d^-}$ are $\langle \mathbf{u}, \mathbf{v} \rangle_\phi = \langle \mathbf{u}^+, \mathbf{v}^+ \rangle - \langle \mathbf{u}^-, \mathbf{v}^- \rangle$ and $d^2_\phi(\mathbf{u}, \mathbf{v}) = ||\mathbf{u}^+ - \mathbf{v}^+||_2^2 - ||\mathbf{u}^- - \mathbf{v}^-||_2^2$.

Example: To see why such embeddings are advantageous, we compare using a one-hot vector representation (whose dimension is $|\mathcal{Y}|$) versus an embedding. Consider a tree with a root node and three branches, each of which is a path with t nodes. Let $\mathcal{Y}$ be the nodes in the tree with the shortest-hops distance as the metric. It can be shown (using [BS16]) that the pseudo-Euclidean embedding dimension is just $d = 3$. The one-hot embedding dimension is $d = |\mathcal{Y}| = 3t + 1$—arbitrarily larger!

Now we are ready to apply these embeddings to our problem. Abusing notation, we write $\boldsymbol{\lambda}_a$ and $\mathbf{y}$ for the pseudo-Euclidean embeddings of λ_a, y. We have that $d^2_{\mathcal{Y}}(\lambda_a, y) = d^2_\phi(\boldsymbol{\lambda}_a, \mathbf{y})$, so that there is no loss of information from working with these spaces. In addition, we write the mean as $\boldsymbol{\mu}_{a,y} = \mathbb{E}[\boldsymbol{\lambda}_a|\mathbf{y}]$ and the covariance as $\boldsymbol{\Sigma}_{a,y}$. It is easy to see that $\boldsymbol{\mu}_{a,y}$ can be used to obtain the mean parameters $\mathbb{E}[d^2_{\mathcal{Y}}(\lambda_a, y)]$ — due to the isometric distances and nice form of distance function in the pseudo-Euclidean spaces we can use $\boldsymbol{\mu}_{a,y}$ to bound $\mathbb{E}[d^2_\phi(\boldsymbol{\lambda}_a, \mathbf{y})]$. Thus our goal is to get an accurate estimate $\hat{\boldsymbol{\mu}}_{a,y} = \hat{\mathbb{E}}[\boldsymbol{\lambda}_a|\mathbf{y}]$. If we could observe y, it would be easy to form an empirical estimate of $\hat{\boldsymbol{\mu}}_{a,y}$—but we do not have access to it. Our approach will be to apply tensor decomposition approaches for multi-view mixtures.

3.2 Multi-View Mixtures and Tensor Decompositions

In a multi-view mixture model, multiple views $\{\lambda_a\}_{a=1}^m$ of a latent variable Y are observed. These views are independent when conditioned on Y. We treat the positive and negative components $\boldsymbol{\lambda}_a^+ \in \mathbb{R}^{d^+}$ and $\boldsymbol{\lambda}_a^- \in \mathbb{R}^{d^-}$ of our pseudo-Euclidean embedding as separate multi-view mixtures:

$$\boldsymbol{\lambda}_a^+|\mathbf{y} \sim \boldsymbol{\mu}_{a,y}^+ + \sigma\sqrt{d^+} \cdot \boldsymbol{\epsilon}_a^+ \quad \text{and} \quad \boldsymbol{\lambda}_a^-|\mathbf{y} \sim \boldsymbol{\mu}_{a,y}^- + \sigma\sqrt{d^-} \cdot \boldsymbol{\epsilon}_a^- \qquad \forall a \in [m]. \tag{4}$$

where $\boldsymbol{\mu}_{a,y}^+ = \mathbb{E}[\boldsymbol{\lambda}_a^+|\mathbf{y}]$, $\boldsymbol{\mu}_{a,y}^- = \mathbb{E}[\boldsymbol{\lambda}_a^-|\mathbf{y}]$ and $\boldsymbol{\epsilon}_a^+, \boldsymbol{\epsilon}_a^-$ are mean zero random vectors with covariances $\frac{1}{d^+}\mathbf{I}_{d^+}, \frac{1}{d^-}\mathbf{I}_{d^-}$ respectively. Here σ^2 is a proxy variance for the noise model in (3).

Algorithm 1 Algorithm for Pseudolabel Construction

Input: Labeling function outputs $\mathbf{L} = \{(\lambda_{a,i}, \lambda_{m,i})\}_{i=1}^n$, Label Space $\mathcal{Y} = \{y_0, \dots y_{r-1}\}$
Output: Pseudolabels for each data point $\mathbf{Z} = \{\tilde{z}_i\}_{i=1}^n$

▷ Step 1: Compute pseudo-Euclidean Embeddings
Compute pairwise distance matrix $\mathbf{D} \in \mathbb{R}^{r \times r}$ with $\mathbf{D}_{ij} = d^2_{\mathcal{Y}}(y_i, y_j)$
Construct matrix $\mathbf{M} \in \mathbb{R}^{r \times r}$ with $\mathbf{M}_{ij} = \frac{1}{2}(\mathbf{D}^2_{0i} + \mathbf{D}^2_{0j} - \mathbf{D}^2_{ij})$
Compute eigendecomposition of $\mathbf{M}$ and let $\mathbf{M} = \mathbf{U}\mathbf{C}\mathbf{U}^T$
Let l^+, l^- be the list of indices of positive and negative eigenvalues sorted by their magnitude.
Let $d^+ = |l^+|, \quad d^- = |l^-|$
Construct permutation matrix $\mathbf{I}_{perm} \in \mathbb{R}^{r \times (d^+ + d^-)}$ by concatenating l^+, l^- in order
$\bar{\mathbf{C}} = \mathbf{C}\mathbf{I}_{perm}, \bar{\mathbf{U}} = \mathbf{U}\mathbf{I}_{perm}$
$\mathbb{Y} = \bar{\mathbf{U}}^T \bar{\mathbf{C}}^{\frac{1}{2}} \in \mathbb{R}^{r \times (d^+ + d^-)}$ and let this define the mapping $g : \mathcal{Y} \mapsto \mathbb{Y}$

▷ Step 2: Parameter Estimation Using Tensor Decomposition
for $a \leftarrow 1$ to $m - 3$ **do**
  Obtain embeddings $\boldsymbol{\lambda}_{a,i} = g(\lambda_{a,i}), \boldsymbol{\lambda}_{a+1,i} = g(\lambda_{a+1,i}), \boldsymbol{\lambda}_{a+2,i} = g(\lambda_{a+2,i}) \quad \forall i \in [n]$
  Construct tensors $\hat{\mathbf{T}}^+$ and $\hat{\mathbf{T}}^-$ as defined in (5) for triple $(a, a+1, a+2)$
  $\hat{\boldsymbol{\mu}}^+_{a,y}, \hat{\boldsymbol{\mu}}^+_{a+1,y}, \hat{\boldsymbol{\mu}}^+_{a+2,y}$ = `TensorDecomposition`($\hat{\mathbf{T}}^+$)
  $\hat{\boldsymbol{\mu}}^-_{a,y}, \hat{\boldsymbol{\mu}}^-_{a+1,y}, \hat{\boldsymbol{\mu}}^-_{a+2,y}$ = `TensorDecomposition`($\hat{\mathbf{T}}^-$)
end for

▷ Step 3: Infer Pseudo-Labels
$\tilde{Z}^{(i)} = \tilde{z}_i \sim Y | \lambda_a = \lambda_a^{(i)}, \dots \lambda_m = \lambda_m^{(i)}; \hat{\boldsymbol{\theta}}$

return $\{\tilde{z}_i\}_{i=1}^n$

We cannot directly estimate these parameters from observations of $\boldsymbol{\lambda}_a$, due to the fact that $\mathbf{y}$ is not observed. However, we can observe various moments of the outputs of the LFs. In particular we can observe tensors of outer products of LF triplets:

$$\mathbf{T}^+ := \mathbb{E}[\boldsymbol{\lambda}^+_a \otimes \boldsymbol{\lambda}^+_b \otimes \boldsymbol{\lambda}^+_c] = \sum_{y \in S_Y} w_y \boldsymbol{\mu}^+_{a,y} \otimes \boldsymbol{\mu}^+_{b,y} \otimes \boldsymbol{\mu}^+_{c,y} \quad \text{and} \quad \hat{\mathbf{T}}^+ := \frac{1}{n} \sum_{i=1}^n \boldsymbol{\lambda}^+_{a,i} \otimes \boldsymbol{\lambda}^+_{b,i} \otimes \boldsymbol{\lambda}^+_{c,i}. \tag{5}$$

Here w_y are the mixture probabilities (prior probabilities of Y) and $S_Y = \{y : w_y > 0\}$. We can similarly define $\mathbf{T}^-$ and $\hat{\mathbf{T}}^-$. This allows us to obtain estimates $\hat{\boldsymbol{\mu}}^+_{a,y}, \hat{\boldsymbol{\mu}}^-_{a,y}$ using the tensor decomposition algorithm of [AGH+14] with minor modifications arising from the fact that we work with pseudo-Euclidean rather than Euclidean space. The overall approach is shown in Algorithm 1. We have one key assumption,

Assumption 1. *Assume that the support of P_Y, i.e., $k = |\{y : w_y > 0\}|$ satisfies $k \leq d$.*

Our first theoretical result shows that we have near-consistency in estimating the mean parameters in (3). We use standard notation $\tilde{\mathcal{O}}$ that is $\mathcal{O}$ but ignoring the logarithmic factors.

Theorem 1. *Let $\hat{\boldsymbol{\mu}}^+_{a,y}, \hat{\boldsymbol{\mu}}^-_{a,y}$ be the estimates of $\boldsymbol{\mu}^+_{a,y}, \boldsymbol{\mu}^-_{a,y}$ returned by Algorithm 1 with input $\hat{\mathbf{T}}^+, \hat{\mathbf{T}}^-$ constructed using isometric pseudo-Euclidean embeddings (in $\mathbb{R}^{d^+,d^-}$) of n (suff. large) i.i.d observations drawn from the models in 3, $k = |S_Y|$, then $\exists$ constant $C_0 > 0$ such that with high probability $\forall a \in [m]$ and $y \in S_Y$,*

$$|\theta_a - \hat{\theta}_a| \leq C_0 \left| \mathbb{E}_{\lambda_a|y}[d^2_{\mathcal{Y}}(\lambda_a, y)] - \hat{\mathbb{E}}_{\lambda_a|y}[d^2_{\mathcal{Y}}(\lambda_a, y)] \right| \leq \epsilon(d^+) + \epsilon(d^-),$$

where

$$\epsilon(d) := \begin{cases} \tilde{\mathcal{O}}\left(k\sqrt{\frac{d}{n}}\right) + \tilde{\mathcal{O}}\left(\frac{\sqrt{k}}{d}\right) & \text{if } \sigma^2 = \Theta(1), \\ \tilde{\mathcal{O}}\left(\sqrt{\frac{k}{n}}\right) + \tilde{\mathcal{O}}\left(\frac{\sqrt{k}}{d}\right) & \text{if } \sigma^2 = \Theta(\frac{1}{d}). \end{cases} \tag{6}$$

Now we interpret Theorem 1. We first note that it is a nearly direct application of [AGJ14]. There are two noise cases for σ. In the high-noise case, σ is independent of dimension d (and thus $|\mathcal{Y}|$). Intuitively, this means the average distance balls around each LF begin to overlap as the number of points grows—explaining the multiplicative k term. If the noise scales down as we add more embedded points, this problem is removed, as in the low-noise case. In both cases, the second error term comes from using the algorithm of [AGH+14] and is independent of the sampling error. Since $k = \Theta(d)$, this term goes down with d. The first error term is due to sampling noise and goes to zero in the number of samples n. Note the tradeoffs of using the embeddings. If we used one-hot encoding, $d = |\mathcal{Y}|$, and in the high-noise case, we would pay a very heavy cost for $\sqrt{d/n}$. However, while sampling error is minimized when using a very small d, we pay a cost in the second error term. This leads to a tradeoff in selecting the appropriate embedding dimension.

We briefly sketch the proof. We show that $\hat{\boldsymbol{\mu}}^{+}_{a,y}, \hat{\boldsymbol{\mu}}^{-}_{a,y}$ are accurate estimates of $\boldsymbol{\mu}^{+}_{a,y}$ and $\boldsymbol{\mu}^{+}_{a,y}$, leading to accurate estimates of $\mathbb{E}_{\boldsymbol{\lambda}_a|\mathbf{y}}[d^2_\phi(\boldsymbol{\lambda}_a, \mathbf{y})]$. The first of these is done by adapting the requirements from [AGJ14] and running the result twice for the two embedding components.

4 Generalization Error for SP in Finite Metric Spaces

We have access to labeling function outputs $\lambda_a^{(i)}, \ldots, \lambda_m^{(i)}$ and noise rate estimates $\hat{\theta}_a$. How can we use these to replace true (but unobserved) labels y in (2)? Our approach is based on [NDRT13, vRW18]. These works deal with noisy labels by modifying the underlying loss function. Analogously, we show that it is possible to modify (2) in such a way that the generalization guarantee is nearly preserved.

4.1 Prediction with Pseudolabels

First, we construct the posterior distribution $P_{\hat{\boldsymbol{\theta}}}(Y = y|\lambda)$. We use our estimated noise model $P_{\hat{\boldsymbol{\theta}}}(\lambda|Y)$ and the prior $P(Y = y)$, which we assume is known. We create pseudo-labels for each data point by drawing a random sample from the posterior distribution conditioned on the output of labeling functions:

$$\tilde{Z}^{(i)} = \tilde{z}_i \sim Y|\lambda_a = \lambda_a^{(i)}, \ldots \lambda_m = \lambda_m^{(i)}; \hat{\boldsymbol{\theta}}. \tag{7}$$

We thus observe $(x_1, \tilde{z}_1), \ldots (x_n, \tilde{z}_n)$ where $\tilde{z}_i$. To overcome the effect of noise we create a perturbed version of the distance function using the noise rates, generalizing [NDRT13]. Let $\mathcal{Y}^m$ denote the m-fold Cartesian product of $\mathcal{Y}$ and let $\Lambda_u = (\lambda_1^{(u)}, \ldots, \lambda_m^{(u)})$ denote its uth entry.

$$\mathbf{P}_{ij} = P_{\boldsymbol{\theta}}(\tilde{Z} = y_i|Y = y_j) = \sum_{u=1}^{|\mathcal{Y}^m|} P_{\boldsymbol{\theta}}(Y = y_i|\Lambda = \Lambda^{(u)}) \cdot P_{\boldsymbol{\theta}}(\Lambda = \Lambda^{(u)}|Y = y_j). \tag{8}$$

Similarly define $\mathbf{Q}_{ij} = P_{\hat{\boldsymbol{\theta}}}(\tilde{Z} = y_i|Y = y_j)$ as above but using the estimated parameters $\hat{\boldsymbol{\theta}}$ instead. Note that $\mathbf{P}$ is the true noise distribution introduced by running the inference procedure with the true parameters $\boldsymbol{\theta}$ of the noise model and $\mathbf{Q}$ is an approximation of the noise distribution obtained by performing inference with the *estimated* parameters $\hat{\boldsymbol{\theta}}$.

With this terminology, we can define the perturbed version of the distance function and a corresponding replacement of (2):

$$\tilde{d}_q(T, \tilde{Y} = y_i) := \sum_{j=1}^{k} (\mathbf{Q}^{-1})_{ij} d^2_{\mathcal{Y}}(T, Y = y_j) \quad \forall y_i \in \mathcal{Y} \tag{9}$$

$$\tilde{F}_q(x, y) := \frac{1}{n} \sum_{i=1}^{n} \alpha_i(x) \tilde{d}_q(y, \tilde{z}_i) \qquad \hat{f}_q(x) = \arg\min_{y \in \mathcal{Y}} \tilde{F}_q(x, y) \tag{10}$$

Similarly define, $\tilde{d}_p, \tilde{F}_p, \hat{f}_p$ using the true noise distribution $\mathbf{P}$. It can be easily shown that the perturbed distance function $\tilde{d}_p$ is an unbiased estimator of the true distance function. However we do

not know the true noise distribution $\mathbf{P}$ hence we cannot use it for prediction. Instead we use $\tilde{d}_q$ based on the estimated noise distribution $\mathbf{Q}$. Note that $\tilde{d}_q$ is no longer an unbiased estimator w.r.t to the true noise distribution. However, we can control its bias as a function of the parameter recovery error bound in Theorem 1.

4.2 Bounding the Generalization Error

A natural question to ask is whether the predictor $\hat{f}_q$ will generalize to new data. More concretely, what can we say about the excess risk $R(\hat{f}_q) - R(f^*)$? Note that compared to the prediction based on clean labels, there are two additional sources of error. One is the noise in the labels (i.e., even if we know the true $\mathbf{P}$, the quality of the pseudolabels is imperfect). The other is our estimation procedure for the noise distribution. We must address both sources of error.

We make the following assumptions on the minimum and maximum singular values $\sigma_{\min}(\mathbf{P})$, $\sigma_{\max}(\mathbf{P})$ and the condition number $\kappa(\mathbf{P})$ of true noise matrix $\mathbf{P}$ and the function F. Additional detail is provided in the Appendix.

Assumption 2. *(Noise model is not arbitrary) Assume that the true parameters $\boldsymbol{\theta}$ are such that $\sigma_{\min}(\mathbf{P}) > 0$, and the condition number $\kappa(\mathbf{P})$ is sufficiently small.*

Assumption 3. *(Normalized features) Assume that $|\alpha(x)| \leq 1 \, \forall x \in \mathcal{X}$.*

Assumption 4. *(Proxy strong convexity) Assume that the function F in* (2) *satisfies the following property with some $\beta > 0$, i.e. as we move away from the minimizer of F, the function increases and the rate of increase is proportional to the distance between the points.*

$$F\big(x, f(x)\big) \geq F\big(x, \hat{f}(x)\big) + \beta \cdot d^2_{\mathcal{Y}}\big(f(x), \hat{f}(x)\big) \qquad \forall x \in \mathcal{X}, \forall f \in \mathcal{F}. \tag{11}$$

With these assumptions, we provide a generalization result for prediction with pseudolabels,

Theorem 2. *(Generalization Error) Let $\hat{f}$ be the minimizer as defined in* (2) *over the clean labels and let $\hat{f}_q$ (defined in* (10)*) be the minimizer over the noisy labels obtained from weak supervision inference in Algorithm 1. Suppose assumptions 2,3,4 hold. Then there exist constants $C_1, C_2 > 0$ dependent on $\sigma_{\max}(\mathbf{P}), \sigma_{\min}(\mathbf{P})$ and k such that w.h.p.,*

$$R(\hat{f}_q) \leq R(f^*) + \mathcal{O}(n^{-\frac{1}{4}}) + \tilde{\mathcal{O}}\Big(\frac{C_1}{\beta} n^{-\frac{1}{2}}\Big) + \tilde{\mathcal{O}}\Big(C_2(\epsilon(d^+) + \epsilon(d^-))\Big). \tag{12}$$

Implications and Tradeoffs: We interpret each term in the bound. The first term is present even with access to the clean labels and hence unavoidable. The second term is the additional error we incur if we learn with the knowledge of the true noise distribution. The third term is due to the use of the estimated noise model. It is dominated by the noise rate recovery result in Theorem 1. If the third term goes to 0, i.e. if we have perfect recovery of the true noise, then we obtain the rate $\mathcal{O}(n^{-1/4})$, the same as in the case of access toclean labels. The third term is introduced by our noise rate recovery algorithm and has two terms: one dominated by $\tilde{\mathcal{O}}(n^{-1/2})$ and the other on $\tilde{\mathcal{O}}(\sqrt{k}/d)$ (see discussion of Theorem 1). Thus we see that we only pay an extra additive factor $\mathcal{O}(\sqrt{k}/d)$ in the excess risk when using pseudolabels. This extra term is negligible for large $d \gg k$.

We briefly sketch the proof. The result follows by first bounding the risk gap between the model learned with the knowledge of noise distribution and the model learned with clean labels i.e. $|R(\hat{f}_p) - R(\hat{f})|$ and then combining it with risk gap between $|R(\hat{f}_q) - R(\hat{f}_p)|$. To obtain the first bound, we use the assumptions (2,3) to argue that F_p is a good approximation of F, i.e. the gap between them is uniformly bounded over all x, y. This fact allows us to show that the minimizers of F_p and F (i.e. $\hat{f}_p$, $\hat{f}$) cannot be far off if the assumption 4 holds. To show the latter, we follow a similar argument to first show a uniform convergence bound for F_p, F_q using the noise rate recovery result and then show a proximity result for their minimizers $\hat{f}_p, \hat{f}_q$ and finally using triangle inequality argue that $\hat{f}_q$ cannot be too far from f^* if, $\hat{f}_p$ and $\hat{f}$ are close to f^*.

5 Manifold-Valued Label Spaces: Noise Recovery and Generalization

We introduce a simple recovery method for weak supervision in constant-curvature Riemannian manifolds. First we briefly introduce some background notation on these spaces, then provide our estimator and consistency result, then the downstream generalization result. Finally, we discuss extensions to symmetric Riemannian manifolds, an even more general class of spaces.

Background on Riemannian manifolds The following is necessarily a very abridged background; more detail can be found in [Lee00, Tu11]. A smooth manifold M is a space where each point is located in a neighborhood diffeomorphic to $\mathbb{R}^d$. Attached to each point $p \in \mathcal{M}$ is a *tangent space* T_pM; each such tangent space is a d-dimensional vector space enabling the use of calculus.

A Riemannian manifold equips a smooth manifold with a Riemannian metric: a smoothly-varying inner product $\langle \cdot, \cdot \rangle_p$ at each point p. This tool allows us to compute angles, lengths, and ultimately, distances $d_{\mathcal{M}}(p, q)$ between points on the manifold as shortest-path distances. These shortest paths are called geodesics and can be parametrized as curves $\gamma(t)$, where $\gamma(0) = p$, or by tangent vectors $v \in T_pM$. The exponential map operation $\exp : T_p\mathcal{M} \to \mathcal{M}$ takes tangent vectors to manifold points. It enables switching between these tangent vectors: $\exp_p(v) = q$ implies that $d_{\mathcal{M}}(p, q) = \|v\|$.

Invariant Our first contribution is a simple invariant that enables us to recover the error parameters. Note that the finite metric-space case is insufficient: the support is infinite. Nor do we need an embedding—we have a continuous representation as-is. Instead, we propose a simple idea based on the law of cosines. Essentially, on average, the geodesic triangle formed by the latent variable $y \in \mathcal{M}$ and two observed LFs λ^a, λ^b, is a right triangle. This means it can be characterized by the (Riemannian) version of the Pythagorean theorem:

Lemma 1. *For $\mathcal{Y} = \mathcal{M}$, a hyperbolic manifold, $y \sim P$ for some distribution P on $\mathcal{M}$ and labeling functions λ^a, λ^b drawn from* (3)*,*

$$\mathbb{E} \cosh d_{\mathcal{Y}}(\lambda^a, \lambda^b) = \mathbb{E} \cosh d_{\mathcal{Y}}(\lambda^b, y) \mathbb{E} \cosh d_{\mathcal{Y}}(\lambda^b, y),$$

while for $\mathcal{Y} = \mathcal{M}$ a spherical manifold,

$$\mathbb{E} \cos d_{\mathcal{Y}}(\lambda^a, \lambda^b) = \mathbb{E} \cos d_{\mathcal{Y}}(\lambda^b, y) \mathbb{E} \cos d_{\mathcal{Y}}(\lambda^b, y).$$

These invariants enable us to easily learn by forming a triplet system. Suppose we construct the equation in Lemma 1 for three pairs of labeling functions. The resulting system can be solved to express $\mathbb{E}[\cosh(d_{\mathcal{Y}}(\lambda^a, y))]$ in terms of $\mathbb{E} \cosh(d_{\mathcal{Y}}(\lambda^a, \lambda^b)), \mathbb{E} \cosh(d_{\mathcal{Y}}(\lambda^a, \lambda^c)), \mathbb{E} \cosh(d_{\mathcal{Y}}(\lambda^b, \lambda^c))$. Specifically,

$$\mathbb{E} \cosh(d_{\mathcal{Y}}(\lambda^a, y)) = \sqrt{\frac{\mathbb{E} \cosh d_{\mathcal{Y}}(\lambda^a, \lambda^b) \mathbb{E} \cosh d_{\mathcal{Y}}(\lambda^a, \lambda^c)}{(\mathbb{E} \cosh(d_{\mathcal{Y}}(\lambda^b, \lambda^c))^2}}.$$

Note that we can estimate $\hat{\mathbb{E}}$ via the empirical versions of terms on the right , as these are based on observable quantities. This is a generalization of the binary case in [FCS+20] and the Gaussian (Euclidean) case in [SLV+22] to hyperbolic manifolds. A similar estimator can be obtained for spherical manifolds by replacing $\cosh$ with $\cos$.

Using this tool, we can obtain a consistent estimator for θ^a for each of $a = 1, \ldots, m$. Let C_0 satisfy $\mathbb{E}|\hat{\mathbb{E}} \cosh(d_{\mathcal{Y}}(\lambda^a, \lambda^b)) - \mathbb{E} \cosh(d_{\mathcal{Y}}(\lambda^a, \lambda^b))| \geq C_0 \mathbb{E}|\hat{\mathbb{E}} d^2_{\mathcal{Y}}(\lambda^a, \lambda^b)) - \mathbb{E} d^2_{\mathcal{Y}}(\lambda^a, \lambda^b)|$; that is, C_0 reflects the pushforward of concentration between the distributions $\cosh(d)$ and d^2. Then,

Theorem 3. *Let $\mathcal{M}$ be a hyperbolic manifold. Fix $0 < \delta < 1$ and let $\Delta(\delta) = \min_\rho Pr(\forall i, d_{\mathcal{Y}}(\lambda^a(i), \lambda^b(i) \leq \rho)) \geq 1 - \delta$. Then, there exists a constant C_1 so that with probability at least $1 - \delta$,*

$$\mathbb{E}|\hat{\mathbb{E}} d^2_{\mathcal{Y}}(\lambda^a, y)) - \mathbb{E} d^2_{\mathcal{Y}}(\lambda^a, y)| \leq \frac{C_1 \cosh(\Delta(\delta))^{3/2}}{C_0 \sqrt{2n}}.$$

As we hoped, our estimator is consistent. Note that we pay a price for a tighter bound: $\Delta(\delta)$ is large for smaller probability δ. It is possible to estimate the size of $\Delta(\delta)$ (more generally, it is a function of the curvature). We provide more details in the Appendix.

Next, we adapt the downstream model predictor (2) in the following way. Let $\hat{\mu}_a^2 = \hat{\mathbb{E}}[d_{\mathcal{Y}}^2(\lambda^a, y)]$. Let $\beta = [\beta_1, \ldots, \beta_m]^T$ be such that $\sum_a \beta_a = 1$ and β minimizes $\sum_a \beta_a^2 \hat{\mu}_a^2$. Then, we set

$$\tilde{f}(x) = \arg\min_{y \in \mathcal{Y}} \frac{1}{n} \sum_{i=1}^{n} \alpha_i(x) \sum_{a=1}^{m} \beta_a^2 d_{\mathcal{Y}}^2(y, \lambda_{a,i}).$$

We simply replace each of the true labels with a combination of the labeling functions. With this, we can state our final result. First, we introduce our assumptions.

Let $q = \arg\min_{z \in \mathcal{Y}} \mathbb{E}[\alpha(x)(y) d_{\mathcal{Y}}^2(z, y)]$, where the expectation is taken over the population level distribution and $\alpha(x)(y)$ denotes the kernel at y.

Assumption 5. *(Bounded Hugging Function c.f. [Str20]) Let q be defined as above. For all $a, b \in \mathcal{M}$, the hugging function at q is given by $k_q^b(a) = 1 - (\| \log_q(a) - \log_q(b) \|^2 - d_{\mathcal{Y}}^2(a, b)) / d_{\mathcal{Y}}^2(q, b)$. We assume that $k_q^b(a)$ is lower bounded by $k_{\min}$.*

Assumption 6. *(Kernel Symmetry) We assume that for all x and all $v \in T_q\mathcal{M}$, $\alpha(x)(\exp_q(v)) = \alpha(x)(\exp_q(-v))$.*

The first condition provides control on how geodesic triangles behave; it relates to the curvature. We provide more details on this in the Appendix. The second assumption restricts us to kernels symmetric about the minimizers of the objective F. Finally, suppose we draw (x, y) and (x', y') independently from P_{XY}. Set $\sigma_o^2 = \alpha(x)(y) \mathbb{E} d_{\mathcal{Y}}^2(y, y')$.

Theorem 4. *Let $\mathcal{M}$ be a complete manifold and suppose the assumptions above hold. Then, there exist constants C_3, C_4*

$$\mathbb{E}[d_{\mathcal{Y}}^2(\hat{f}(x), \tilde{f}(x))] \leq \frac{C_3 \sigma_o^2}{n k_{\min}} + \frac{C_4 \sum_{a=1}^{m} \beta_a^2 \hat{\mu}_a^2}{m n k_{\min}}.$$

Note that as both m and n grow, as long as our worst-quality LF has bounded variance, our estimator of the true predictor is consistent. Moreover, we also have favorable dependence on the noise rate. This is because the only error we incur is in computing suboptimal β coefficients. We comment on this suboptimality in the Appendix.

A simple corollary of Theorem 4 provides the generalization guarantees we sought,

Corollary 1. *Let $\mathcal{M}$ be a complete manifold and suppose the assumptions above hold. Then, with high probability,*

$$R(\tilde{f}) \leq R(f^*) + \mathcal{O}(n^{-\frac{1}{4}}).$$

Extensions to Other Manifolds First, we note that all of our approaches almost immediately lift to products of constant-curvature spaces. For example, we have that $\mathcal{M}_1 \times \mathcal{M}_2$ has metric $d_{\mathcal{Y}}^2(p, q) = d_{\mathcal{M}_1}^2(p_1, q_1) + d_{\mathcal{M}_2}^2(p_2, q_2)$, where p_i, q_i are the projections of p, q onto the ith component.

We can go beyond products of constant-curvature spaces as well. To do so, we can build generalizations of the law of cosines (as needed for the invariance in Lemma 1). For example, it is possible to do for symmetric Riemannian manifolds using the tools in [AH91].

6 Conclusion

We studied the theoretical properties of weak supervision applied to structured prediction. Our focus was on two general scenarios: label spaces that are finite metric spaces or continuous spaces given by constant-curvature manifolds. In both scenarios, we introduced ways to estimate the noise rates of labeling functions, achieving consistency or near-consistency. Using these estimators, we established that, after suitable modifications, downstream structured prediction models maintain their generalization guarantees. Future directions include extending these results to even more general manifolds and removing some of the assumptions that limit our results to particular models.

References

[AGH+14] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832, 2014.

[AGJ14] Animashree Anandkumar, Rong Ge, and Majid Janzamin. Sample complexity analysis for learning overcomplete latent variable models through tensor methods. *arXiv preprint arXiv:1408.0553*, 2014.

[AH91] Helmer Aslaksen and Hsueh-Ling Huynh. Laws of trigonometry in symmetric spaces. *Geometry from the Pacific Rim*, 1991.

[BHS+07] Gükhan H. Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data (Neural Information Processing)*. The MIT Press, 2007.

[BS16] R.B. Bapat and Sivaramakrishnan Sivasubramanian. Squared distance matrix of a tree: Inverse and inertia. *Linear Algebra and its Applications*, 491:328–342, 2016.

[CRR16] Carlo Ciliberto, Lorenzo Rosasco, and Alessandro Rudi. A consistent regularization approach for structured prediction. In *Advances in Neural Information Processing Systems 30 (NIPS 2016)*, volume 30, 2016.

[DRS+20] Jared A. Dunnmon, Alexander J. Ratner, Khaled Saab, Nishith Khandwala, Matthew Markert, Hersh Sagreiya, Roger Goldman, Christopher Lee-Messer, Matthew P. Lungren, Daniel L. Rubin, and Christopher Ré. Cross-modal data programming enables rapid medical machine learning. *Patterns*, 1(2), 2020.

[FCS+20] Daniel Y. Fu, Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. Fast and three-rious: Speeding up weak supervision with triplet methods. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 2020.

[Gol85] Lev Goldfarb. A new approach to pattern recognition. pages 241–402, 1985.

[GS19] Colin Graber and Alexander Schwing. Graph structured prediction energy networks. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2019)*, volume 33, 2019.

[KAG18] Anna Korba and Florence d'Alché-Buc Alexandre Garcia. A structured prediction approach for label ranking. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2018)*, volume 32, 2018.

[KL15] Volodymyr Kuleshov and Percy S Liang. Calibrated structured prediction. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

[KW78] J.B. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, 1978.

[Lee00] John M. Lee. *Introduction to Smooth Manifolds*. Springer, 2000.

[NDRT13] Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 1196–1204, 2013.

[PM19] Alexander Petersen and Hans-Georg Müller. Fréchet regression for random objects with euclidean predictors. *Annals of Statistics*, 47(2):691–719, 2019.

[RBE+18] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the 44th International Conference on Very Large Data Bases (VLDB)*, Rio de Janeiro, Brazil, 2018.

[RCMR18] Alessandro Rudi, Carlo Ciliberto, GianMaria Marconi, and Lorenzo Rosasco. Manifold structured prediction. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2018)*, volume 32, 2018.

[RHD+19] A. J. Ratner, B. Hancock, J. Dunnmon, F. Sala, S. Pandey, and C. Ré. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, 2019.

[RNGS20] Christopher Ré, Feng Niu, Pallavi Gudipati, and Charles Srisuwananukorn. Overton: A data system for monitoring and improving machine-learned products. In *Proceedings of the 10th Annual Conference on Innovative Data Systems Research*, 2020.

[RSW+16] A. J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, 2016.

[SLB20] Esteban Safranchik, Shiying Luo, and Stephen Bach. Weakly supervised sequence tagging from noisy rules. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 5570–5578, Apr. 2020.

[SLV+22] Changho Shin, Winfred Li, Harit Vishwakarma, Nicholas Carl Roberts, and Frederic Sala. Universalizing weak supervision. In *International Conference on Learning Representations*, 2022.

[Str20] Austin J. Stromme. *Wasserstein Barycenters: Statistics and Optimization*. MIT, 2020.

[Tu11] Loring W. Tu. *An Introduction to Manifolds*. Springer, 2011.

[vRW18] Brendan van Rooyen and Robert C. Williamson. A theory of learning with corrupted labels. *Journal of Machine Learning Research*, 18(228):1–50, 2018.

Checklist

1. Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
2. Did you describe the limitations of your work? [Yes]
3. Did you discuss any potential negative societal impacts of your work? [N/A]
4. Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
5. Did you state the full set of assumptions of all theoretical results? [Yes]
6. Did you include complete proofs of all theoretical results? [Yes] See appendix