Model Misspecification in Multiple Weak Supervision

Salva Rühling Cachay  
Technical University of Darmstadt  
salvaruehling@gmail.com

Benedikt Boecking  
Carnegie Mellon University

Artur Dubrawski  
Carnegie Mellon University

Abstract

Data programming has proven to be an attractive alternative to costly hand-labeling of data. In this paradigm, users encode domain knowledge into labeling functions (LF), heuristics that label a subset of the data noisily and may have complex dependencies. The effects on test set performance of a downstream classifier caused by label model misspecification are understudied—presenting a serious knowledge gap to practitioners, in particular since LF dependencies are frequently ignored. In this paper, we focus on modeling errors due to structure over-specification. Based on novel theoretical bounds on the modeling error, we empirically show that this error can be substantial, even when modeling a seemingly sensible structure.

1 Introduction and Problem setup

In data programming users define $m$ labeling functions (LF) which noisily label subsets of the data [1]. These noisy sources are then modeled to obtain an estimate of the latent true label. In practice, the LF often exhibit statistical dependencies amongst each other, such as sources operating on the same or similar input. Defining the correct dependency structure is difficult however, thus a common approach in popular libraries [2, 3] and related research [4, 5, 6] is to ignore it.

Let $(x, y) \sim D$ be the true data generating distribution and for simplicity assume that $y \in \{-1, 1\}$. As in [1], users provide $m$ LFs $\lambda = \lambda(x) \in \{-1, 0, 1\}$, where 0 means that the LF abstained from labeling. Following [1], we model the joint distribution of $y, \lambda$ as a factor graph which allows for modeling of higher-order dependencies between LFs. More recent weak supervision models and model fitting approaches often only allow for pairwise correlation dependencies to be modeled [7, 8]. To study model misspecification we compare two label models, $p_\theta$ for the conditional independent case and $p_\mu$ which models higher-order dependencies:

$$p_\mu(\lambda, y) = \frac{1}{Z_\mu} \exp \left( \mu^T \phi_1(\lambda, y) \right), \quad \mu \in \mathbb{R}^{m+M} \quad (1)$$

$$p_\theta(\lambda, y) = Z_\theta^{-1} \exp \left( \theta^T \phi_1(\lambda, y) \right), \quad \theta \in \mathbb{R}^m, \quad (2)$$

where $\phi_1(\lambda, y) = \lambda y$ are the accuracy factors, $\phi_2(\cdot)$ are arbitrary, higher-order dependencies and $Z_\mu^{-1}, Z_\theta^{-1}$ are normalization constants. We assume w.l.o.g. that factors are bounded $\leq 1$. Finally, we extend [1] by introducing negated, bolstering, priority dependencies (definitions in the appendix), e.g. the latter encoding the notion that one LF’s vote should be prioritized over the one from a noisier LF.

Bound on the probabilistic label difference due to model misspecification  

We now state our bound on the probabilistic label difference (which we prove in the appendix):

$$|p_\mu(y \mid \lambda) - p_\theta(y \mid \lambda)| \leq \frac{1}{2} ||\mu_1 - \theta||_1 + \frac{1}{4} ||\mu_2||_1 \quad (3)$$

The bound naturally involves the accuracy parameter estimation error $||\mu_1 - \theta||_1$, and the learned strength of the dependencies only modeled in $p_\mu$. This is an important quantity of interest since the probabilistic labels are used to train a downstream model. Unsurprisingly then, this quantity reappears as a main factor controlling the generalization risk, see the proof of theorem 1 in [7]. The presented bound is tighter than the one from [7], while in addition accounting for model misspecification.

2 Experiments

2.0.1 Proxy for finding true dependencies in real datasets

The underlying true structure between two real labeling functions $\lambda_j, \lambda_k$ is, of course, unknown. However, by using true training labels (solely for this purpose) together with the observed LF votes, we can compute resulting factor values for each data point $i$, to observe empirical strength of dependency factor $l$ over a training set: $v_{j,k}^l = \sum_i \phi_l(\lambda(x_i)_j, \lambda(x_i)_k, y_i)$. Sorting dependencies $l$ according to $v_{j,k}^l$ in descending order, we then choose to model the top $d$ dependencies. These are the dependencies for which the true labels provide the most evidence of being correct.

2.0.2 Downstream model performance deterioration due to structure over-specification

For the following experiment we use the IMDB Movie Review Sentiment dataset consisting of $n = 25k$ training and test samples each [9] and manually select a set of $m = 135$ sensible LFs that label on the presence of a single word or a pair of words (i.e. uni-/bi-gram LFs). In addition we use the Bias in Bios dataset [10] from which we create a binary classification task to distinguish the frequently occurring occupations professor or teacher ($n = 12294, m = 85$). We deliberately choose unigram and bigram LFs so as to create dependencies we expect to help with downstream model performance.

We choose different $d \in \{1, 3, 5, \ldots, 40\}$ and then model the strongest $\leq d$ dependencies of each factor $l$ according to $v_{j,k}^l$. An example of the strongest and weakest dependencies for the IMDB dataset is shown in Table 1. For the Bias in Bios experiment, an example of a strong reinforcing dependency is that the term ‘phd’ appears in addition to the term ‘university’. We report the test set performance of a simple 3-layer neural network trained on the probabilistic labels, averaged out over 100 runs.

We find that modeling more than a handful of dependencies significantly deteriorates the downstream classifier performance (by up to 8 ROC-AUC points) as compared to simply ignoring them (“No deps”). This effect intensifies as we model more dependencies. In brackets, the baseline score for the independent model.

Discussion Even though this result and insight is highly relevant for practitioners, it has, to the best of our knowledge, not been explored in detail. It may come as a surprise that modeling seemingly sensible dependencies can significantly deteriorate the targeted downstream model performance. We hypothesize that this is due to the true model being close to the conditionally independent case and in Eq. 3 we see that the bound becomes looser as more incorrect dependencies are modeled. Also, as briefly note, more complex models often suffer of a higher sample complexity. We can conclude from this that ignoring potential dependencies will often be a reasonable baseline for practitioners.
References


3 Appendix

3.1 Problem setup recap

Let \((x, y) \sim \mathcal{D}\) be the true data generating distribution and for simplicity assume that \(y \in \mathcal{Y} = \{-1, 1\}\). Users provide \(m\) labeling functions (LFs) \(\lambda = \lambda(x) \in \{-1, 0, 1\}\), where 0 means that the LF abstained from labeling. We compare two label models, \(p_\theta\) for the conditional independent case, and \(p_\mu\) which models higher-order dependencies:

\[
p_\mu(\lambda, y) = \frac{1}{Z_\mu} \exp \left( \mu^T \phi(\lambda, y) \right) = Z^{-1}_\mu \exp \left( \mu_1^T \phi_1(\lambda, y) + \mu_2^T \phi_2(\lambda, y) \right), \quad \mu \in \mathbb{R}^{m+M} \tag{4}
\]

\[
p_\theta(\lambda, y) = Z^{-1}_\theta \exp \left( \theta^T \phi_1(\lambda, y) \right), \quad \theta \in \mathbb{R}^m, \tag{5}
\]

where \(\phi_1(\lambda, y) = \lambda y\) are the accuracy factors, \(\phi_2(\cdot)\) are arbitrary, higher-order dependencies and \(Z^{-1}_\mu, Z^{-1}_\theta\) are normalization constants. We assume w.l.o.g. that factors are bounded \(\leq 1\).

3.2 Proof of the bound

**Bound** 
Our bound on the probabilistic label difference between the two models above is:

\[
|p_\mu(y | \lambda) - p_\theta(y | \lambda)| \leq \frac{1}{2} ||\mu_1 - \theta||_{1} + \frac{1}{4} ||\mu_2||_{1} \tag{6}
\]

**Proof** 
First note that the posterior of the label models as above can be rewritten as follows:

\[
p_\mu(y | \lambda) = \frac{p_\mu(\lambda, y)}{p_\mu(\lambda)} = \frac{p_\mu(\lambda, y)}{\sum_{\tilde{y} \in \mathcal{Y}} p_\mu(\lambda, \tilde{y})} = \frac{Z^{-1}_\mu \exp \left( \mu^T \phi(\lambda, y) \right)}{\sum_{\tilde{y} \in \mathcal{Y}} Z^{-1}_\mu \exp \left( \mu^T \phi(\lambda, \tilde{y}) \right)} = \frac{\exp \left( \mu^T \phi(\lambda, y) \right)}{\sum_{\tilde{y} \in \mathcal{Y}} \exp \left( \mu^T \phi(\lambda, \tilde{y}) \right)} = \frac{1}{1 + \exp \left( \mu^2 \left( \phi(\lambda, y) - \phi(\lambda, -y) \right) \right)} = \sigma \left( \mu^T \phi(\lambda, y) - \phi(\lambda, -y) \right) = \sigma \left( 2\mu_1^T \phi_1(\lambda, y) + \mu_2^T \left( \phi_2(\lambda, y) - \phi_2(\lambda, -y) \right) \right),
\]

where \(\sigma(x) = \frac{1}{1 + \exp(-x)}\) is the sigmoid function and we used the fact that the accuracy factors are odd functions, i.e. \(\phi_1(\lambda, -y) = -\lambda y = -\phi_1(\lambda, y)\). Analogously, \(p_\theta(y | \lambda) = \sigma \left( 2\theta^T \phi_1(\lambda, y) \right)\). Therefore we have that

\[
|p_\mu(y | \lambda) - p_\theta(y | \lambda)| = |\sigma \left( 2\mu_1^T \phi_1(\lambda, y) + \mu_2^T \left( \phi_2(\lambda, y) - \phi_2(\lambda, -y) \right) \right) - \sigma \left( 2\theta^T \phi_1(\lambda, y) \right)|
\]

By the mean value theorem it follows that for some \(c\) between the arguments of \(\sigma\) above

\[
= \sigma'(c) \left| \left( 2\mu_1^T \phi_1(\lambda, y) + \mu_2^T \left( \phi_2(\lambda, y) - \phi_2(\lambda, -y) \right) \right) - 2\theta^T \phi_1(\lambda, y) \right|
\]

Using the triangle inequality and the fact that \(\max_x \sigma'(x) = \max_x \sigma(x)(1 - \sigma(x)) = \frac{1}{4}\), we can now bound this expression as follows

\[
\leq \frac{1}{2} |\mu_1 - \theta^T \phi_1(\lambda, y)| + \frac{1}{4} |\mu_2^T \left( \phi_2(\lambda, y) - \phi_2(\lambda, -y) \right)|
\]

finally, since the defined higher-order dependencies are indicator functions \(\neq 0\) for only one \(y \in \mathcal{Y}\), and if \(||q||_{\infty} \leq 1\) then \(|x^T \tilde{q}| = \sum_i x_i \tilde{q}_i \leq \sum_i |x_i \tilde{q}_i| \leq \sum_i |x_i| = ||x||_{1}\), this reduces to

\[
\leq \frac{1}{2} ||\mu_1 - \theta||_{1} + \frac{1}{4} ||\mu_2||_{1}.
\]
3.3 Factor Definitions

We supplement the factor definitions of the used higher-order dependencies (the first two stem from [1], the rest we defined ourselves for the conducted experiments). Whenever a factor \( \phi_{j,k}(\lambda, y) \) is not symmetric (all factors, besides bolstering), we define it so that \( \text{LF}_k \) acts on (e.g. negates) \( \text{LF}_j \).

For the fixing dependency we have:

\[
\phi_{j,k}^{\text{Fix}}(\lambda, y) = \begin{cases} 
+1 & \text{if } \lambda_j = -y \land \lambda_k = y \\
-1 & \text{if } \lambda_j = 0 \land \lambda_k \neq 0 \\
0 & \text{otherwise}
\end{cases}
\]

for the reinforcing one:

\[
\phi_{j,k}^{\text{Rei}}(\lambda, y) = \begin{cases} 
+1 & \text{if } \lambda_j = \lambda_k = y \\
-1 & \text{if } \lambda_j = 0 \land \lambda_k \neq 0 \\
0 & \text{otherwise}
\end{cases}
\]

for the priority factor:

\[
\phi_{j,k}^{\text{Pri}}(\lambda, y) = \begin{cases} 
+1 & \text{if } \lambda_j = -y \land \lambda_k = y \\
-1 & \text{if } \lambda_j = y \land \lambda_k = -y \\
0 & \text{otherwise}
\end{cases}
\]

for the bolstering:

\[
\phi_{j,k}^{\text{Bol}}(\lambda, y) = \begin{cases} 
+1 & \text{if } \lambda_j = \lambda_k = y \\
-1 & \text{if } \lambda_j = \lambda_k \neq y \lor \lambda_j = -\lambda_k \neq 0 \\
0 & \text{otherwise}
\end{cases}
\]

and, finally, for the negated factor:

\[
\phi_{j,k}^{\text{Neg}}(\lambda, y) = \begin{cases} 
+1 & \text{if } \lambda_j = -y \land \lambda_k = y \\
-1 & \text{if } (\lambda_j = y \land \lambda_k = -y) \lor \lambda_j = \lambda_k \neq 0 \\
0 & \text{otherwise}
\end{cases}
\]