Learning the Structure of Generative Models without Labeled Data

Stephen H. Bach [1] **Bryan He** [1] **Alexander Ratner** [1] **Christopher Ré** [1]

Abstract

Curating labeled training data has become the primary bottleneck in machine learning. Recent frameworks address this bottleneck with generative models to synthesize labels at scale from weak supervision sources. The generative model's dependency structure directly affects the quality of the estimated labels, but selecting a structure automatically without any labeled data is a distinct challenge. We propose a structure estimation method that maximizes the ℓ_1-regularized marginal pseudolikelihood of the observed data. Our analysis shows that the amount of unlabeled data required to identify the true structure scales sublinearly in the number of possible dependencies for a broad class of models. Simulations show that our method is $100\times$ faster than a maximum likelihood approach and selects $1/4$ as many extraneous dependencies. We also show that our method provides an average of 1.5 F1 points of improvement over existing, user-developed information extraction applications on real-world data such as PubMed journal abstracts.

1. Introduction

Supervised machine learning traditionally depends on access to labeled training data, a major bottleneck in developing new methods and applications. In particular, deep learning methods require tens of thousands or more labeled data points for each specific task. Collecting these labels is often prohibitively expensive, especially when specialized domain expertise is required, and major technology companies are investing heavily in hand-curating labeled training data (Metz, 2016; Eadicicco, 2017). Aiming to overcome this bottleneck, there is growing interest in using generative models to synthesize training data from weak supervision sources such as heuristics, knowledge bases, and weak classifiers trained directly on noisy sources. Rather than treating training labels as gold-standard inputs, such methods model training set creation as a process in order to generate training labels at scale. The true class label for a data point is modeled as a latent variable that generates the observed, noisy labels. After fitting the parameters of this generative model on unlabeled data, a distribution over the latent, true labels can be inferred.

The structure of such generative models directly affects the inferred labels, and prior work assumes that the structure is user-specified (Alfonseca et al., 2012; Takamatsu et al., 2012; Roth & Klakow, 2013b; Ratner et al., 2016). One option is to assume that the supervision sources are conditionally independent given the latent class label. However, statistical dependencies are common in practice, and not taking them into account leads to misjudging the accuracy of the supervision. We cannot rely in general on users to specify the structure of the generative model, because supervising heuristics and classifiers might be independent for some data sets but not others. We therefore seek an efficient method for automatically learning the structure of the generative model from weak supervision sources alone.

While structure learning in the supervised setting is well-studied (e.g., Meinshausen & Bühlmann, 2006; Zhao & Yu, 2006; Ravikumar et al., 2010, see also Section 6), learning the structure of generative models for weak supervision is challenging because the true class labels are latent. Although we can learn the parameters of generative models for a given structure using stochastic gradient descent and Gibbs sampling, modeling all possible dependencies does not scale as an alternative to model selection. For example, estimating all possible correlations for a modestly sized problem of 100 weak supervision sources takes over 40 minutes. (For comparison, our proposed approach solves the same problem in 15 seconds.) As users develop their supervision heuristics, rerunning parameter learning to identify dependencies becomes a prohibitive bottleneck.

We propose an estimator to learn the dependency structure of a generative model without using any labeled training data. Our method maximizes the ℓ_1-regularized marginal pseudolikelihood of each supervision source's output independently, selecting those dependencies that have nonzero

[1]Stanford University, Stanford, California. Correspondence to: Stephen Bach <bach@cs.stanford.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

weights. This estimator is analogous to maximum likelihood for logistic regression, except that we marginalize out our uncertainty about the latent class label. Since the pseudolikelihood is a function of one free variable and marginalizes over one other variable, we compute the gradient of the marginal pseudolikelihood exactly, avoiding the need for approximating the gradient with Gibbs sampling, as is done for maximum likelihood estimation.

Our analysis shows that the amount of data required to identify the true structure scales sublinearly in the number of possible dependencies for a broad class of models. Intuitively, this follows from the fact that learning the generative model's parameters is possible when there are a sufficient number of better-than-random supervision sources available. With enough signal to estimate the latent class labels better than random guessing, those estimates can be refined until the model is identified.

We run experiments to confirm these predictions. We also compare against the alternative approach of considering all possible dependencies during parameter learning. We find that our method is $100\times$ faster. In addition, our method returns $1/4$ as many extraneous correlations on synthetic data when tuned for comparable recall. Finally, we demonstrate that on real-world applications of weak supervision, using generative models with automatically learned dependencies improves performance. We find that our method provides on average 1.5 F1 points of improvement over existing, user-developed information extraction applications on PubMed abstracts and hardware specification sheets.

2. Background

When developing machine learning systems, the primary bottleneck is often curating a sufficient amount of labeled training data. Hand labeling training data is expensive, time consuming, and often requires specialized knowledge. Recently researchers have proposed methods for synthesizing labels from noisy label sources using generative models. (See Section 6 for a summary.) We ground our work in one framework, data programming (Ratner et al., 2016), that generalizes many approaches in the literature.

In data programming, weak supervision sources are encoded as *labeling functions*, heuristics that label data points (or abstain). A generative probabilistic model is fit to estimate the accuracy of the labeling functions and the strength of any user-specified statistical dependencies among their outputs. In this model, the true class label for a data point is a latent variable that generates the labeling function outputs. After fitting the parameters of the generative model, a distribution over the latent, true labels can be estimated and be used to train a discriminative model by minimizing the expected loss with respect to that distribution.

We formally describe this setup by first specifying for each data point x_i a latent random variable $y_i \in \{-1, 1\}$ that is its true label. For example, in an information extraction task, x_i might be a span of text. Then, y_i can represent whether it is a mention of a company or not (entity tagging). Alternatively, x_i might be a more complex structure, such as a tuple of canonical identifiers along with associated mentions in a document, and then y_i can represent whether a relation of interest over that tuple is expressed in the document (relation extraction).

We do not have access to y_i (even at training time), but we do have n user-provided labeling functions $\lambda_1, \ldots, \lambda_n$ that can be applied to x_i to produce outputs $\Lambda_{i1}, \ldots, \Lambda_{in}$. For example, for the company-tagging task mentioned above, a labeling function might apply the regular expression `.+\sInc\.` to a span of text and return whether it matched. The domain of each Λ_{ij} is $\{-1, 0, 1\}$, corresponding to *false*, *abstaining*, and *true*. Generalizing to the multiclass case is straightforward.

Our goal is to estimate a probabilistic model that generates the labeling-function outputs $\Lambda \in \{-1, 0, 1\}^{m \times n}$. A common assumption is that the outputs are conditionally independent given the true label, and that the relationship between Λ and y is governed by n *accuracy* dependencies

$$\phi_j^{\text{Acc}}(\Lambda_i, y_i) := \; y_i \Lambda_{ij}$$

with a parameter θ_j^{Acc} modeling how accurate each labeling function λ_j is. We refer to this structure as the *conditionally independent model*, and specify it as

$$p_\theta(\Lambda, Y) \propto \exp\left(\sum_{i=1}^{m}\sum_{j=1}^{n} \theta_j^{\text{Acc}} \phi_j^{\text{Acc}}(\Lambda_i, y_i)\right) , \qquad (1)$$

where $Y := y_1, \ldots, y_m$.

We estimate the parameters θ by minimizing the negative log marginal likelihood $p_\theta(\bar{\Lambda})$ for an observed matrix of labeling function outputs $\bar{\Lambda}$:

$$\arg\min_\theta \; -\log \sum_Y p_\theta(\bar{\Lambda}, Y) \, . \qquad (2)$$

Optimizing the likelihood is straightforward using stochastic gradient descent. The gradient of objective (2) with respect to parameter θ_j^{Acc} is

$$\sum_{i=1}^{m} \left(E_{\Lambda, Y \sim \theta}\left[\phi_j^{\text{Acc}}(\Lambda_i, y_i)\right] - E_{Y \sim \theta | \bar{\Lambda}}\left[\phi_j^{\text{Acc}}(\bar{\Lambda}_i, y_i)\right]\right) ,$$

the difference between the corresponding sufficient statistic of the joint distribution p_θ and the same distribution conditioned on $\bar{\Lambda}$. In practice, we can interleave samples to estimate the gradient and gradient steps very tightly, taking

a small step after each sample of each variable Λ_{ij} or y_i, similarly to contrastive divergence (Hinton, 2002).

The conditionally independent model is a common assumption, and using a more sophisticated generative model currently requires users to specify its structure. In the rest of the paper, we address the question of automatically identifying the dependency structure from the observations $\bar{\Lambda}$ without observing Y.

3. Structure Learning without Labels

Statistical dependencies arise naturally among weak supervision sources. In data programming, users often write labeling functions with directly correlated outputs or even labeling functions deliberately designed to reinforce others with narrow, more precise heuristics. To address this issue, we generalize the conditionally independent model as a factor graph with additional dependencies, including higher-order factors that connect multiple labeling function outputs for each data point x_i and label y_i. We specify the general model as

$$p_\theta(\Lambda, Y) \propto \exp\left(\sum_{i=1}^{m}\sum_{t\in T}\sum_{s\in S_t}\theta_s^t\phi_s^t(\Lambda_i, y_i)\right). \quad (3)$$

Here T is the set of dependency types of interest, and S_t is a set of index tuples, indicating the labeling functions that participate in each dependency of type $t \in T$. We start by defining standard *correlation* dependencies of the form

$$\phi_{jk}^{\text{Cor}}(\Lambda_i, y_i) := \mathbb{1}\{\Lambda_{ij} = \Lambda_{ik}\}.$$

We refer to such dependencies as pairwise among labeling functions because they depend only on two labeling function outputs. We can also consider higher-order dependencies that involve more variables, such as *conjunction* dependencies of the form

$$\phi_{jk}^{\text{And}}(\Lambda_i, y_i) := \mathbb{1}\{\Lambda_{ij} = y_i \wedge \Lambda_{ik} = y_i\}.$$

Estimating the structure of the distribution $p_\theta(\Lambda, Y)$ is challenging because Y is latent; we never observe its value, even during training. We must therefore work with the marginal likelihood $p_\theta(\Lambda)$. Learning the parameters of the generative model jointly requires Gibbs sampling to estimate gradients. As the number of possible dependencies increases at least quadratically in the number of labeling functions, this heavyweight approach to learning does not scale (see Section 5.2).

3.1. Learning Objective

We can scale up learning over many potentially irrelevant dependencies by optimizing a different objective: the log marginal pseudolikelihood of the outputs of a single labeling function λ_j, i.e., conditioned on the outputs of the others $\lambda_{\setminus j}$, using ℓ_1 regularization to induce sparsity. The objective is

$$\begin{aligned}&\arg\min_\theta \; -\log p_\theta(\bar{\Lambda}_j \mid \bar{\Lambda}_{\setminus j}) + \epsilon\|\theta\|_1 \quad (4)\\ =\; &\arg\min_\theta \; -\sum_{i=1}^{m}\log\sum_{y_i} p_\theta(\bar{\Lambda}_{ij}, y_i \mid \bar{\Lambda}_{i\setminus j}) + \epsilon\|\theta\|_1,\end{aligned}$$

where $\epsilon > 0$ is a hyperparameter.

By conditioning on all other labeling functions in each term $\log\sum_{y_i} p_\theta(\bar{\Lambda}_{ij}, y_i \mid \bar{\Lambda}_{i\setminus j})$, we ensure that the gradient can be computed in polynomial time with respect to the number of labeling functions, data points, and possible dependencies; without requiring any sampling or variational approximations. The gradient of the log marginal pseudolikelihood is the difference between two expectations: the sufficient statistics conditioned on all labeling functions but λ_j, and conditioned on all labeling functions:

$$-\frac{\partial \log p(\bar{\Lambda}_j \mid \bar{\Lambda}_{\setminus j})}{\partial \theta_s^t} = \alpha - \beta, \quad (5)$$

where

$$\alpha := \sum_{i=1}^{m}\sum_{\Lambda_{ij}, y_i} p_\theta(\Lambda_{ij}, y_i \mid \bar{\Lambda}_{i\setminus j})\phi_s^t((\Lambda_{ij}, \bar{\Lambda}_{i\setminus j}), y_i)$$

$$\beta := \sum_{i=1}^{m}\sum_{y_i} p(y_i \mid \bar{\Lambda}_i)\phi_s^t(\bar{\Lambda}_i, y_i).$$

Note that in the definition of α, ϕ_s^t operates on the value of Λ_{ij} set in the summation and the observed values of $\bar{\Lambda}_{i\setminus j}$.

We optimize for each labeling function λ_j in turn, selecting those dependencies with parameters that have a sufficiently large magnitude and adding them to the estimated structure.

3.2. Implementation

We implement our method as Algorithm 1, a stochastic gradient descent (SGD) routine. At each step of the descent, the gradient (5) is estimated for a single data point, which can be computed in closed form. Using SGD has two advantages. First, it requires only first-order gradient information. Other methods for ℓ_1-regularized regression like interior-point methods (Koh et al., 2007) usually require computing second-order information. Second, the observations $\bar{\Lambda}$ can be processed incrementally. Since data programming operates on unlabeled data, which is often abundant, scalability is crucial. To implement ℓ_1 regularization as part of SGD, we use an online truncated gradient method (Langford et al., 2009).

In practice, we find that the only parameter that requires tuning is ϵ, which controls the threshold and regularization

Algorithm 1 Structure Learning for Data Programming

Input: Observations $\bar{\Lambda} \in \{-1, 0, 1\}^{m \times n}$, threshold ϵ, distribution p with parameters θ, initial parameters θ^0, step size η, epoch count $\mathcal{T}$, truncation frequency K

$D \leftarrow \emptyset$
for $j = 1$ **to** n **do**
  $\theta \leftarrow \theta^0$
  for $\tau = 1$ **to** $\mathcal{T}$ **do**
    for $i = 1$ **to** m **do**
      for θ_s^t **in** θ **do**
        $\alpha \leftarrow \sum_{\Lambda_{ij}, y_i} p(\Lambda_{ij}, y_i | \bar{\Lambda}_{i \setminus j}) \phi_s^t((\Lambda_{ij}, \bar{\Lambda}_{i \setminus j}), y_i)$
        $\beta \leftarrow \sum_{y_i} p(y_i \mid \bar{\Lambda}_i) \phi_s^t(\bar{\Lambda}_i, y_i)$
        $\theta_s^t \leftarrow \theta_s^t - \eta(\alpha - \beta)$
      if $\tau m + i$ **mod** K **is** 0 **then**
        for θ_s^t **in** θ **where** $\theta_s^t > 0$ **do**
          $\theta_s^t \leftarrow \max\{0, \theta_s^t - K\eta\epsilon\}$
        for θ_s^t **in** θ **where** $\theta_s^t < 0$ **do**
          $\theta_s^t \leftarrow \min\{0, \theta_s^t + K\eta\epsilon\}$
  for θ_s^t **in** θ **where** $j \in s$ **do**
    if $|\theta_s^t| > \epsilon$ **then**
      $D \leftarrow D \cup \{(s, t)\}$
return D

strength. Higher values induce more sparsity in the selected structure. For the other parameters, we use the same values in all of our experiments: step size $\eta = m^{-1}$, epoch count $\mathcal{T} = 10$, and truncation frequency $K = 10$.

4. Analysis

We provide guarantees on the probability that Algorithm 1 successfully recovers the exact dependency structure. We first provide a general recovery guarantee for all types of possible dependencies, including both pairwise and higher-order dependencies. However, in many cases, higher-order dependencies are not necessary to model the behavior of the labeling functions. In fact, as we demonstrate in Section 5.3, in many useful models there are only accuracy dependencies and pairwise correlations. In this case, we show as a corollary to our general result that the number of samples required is sublinear in the number of possible dependencies, specifically $O(n \log n)$.

Previous analyses for the supervised case do not carry over to the unsupervised setting because the problem is no longer convex. For example, analysis of an analogous method for supervised Ising models (Ravikumar et al., 2010) relies on Lagrangian duality and a tight duality gap, which does not hold for our estimation problem. Instead, we reason about a region of the parameter space in which we can estimate Y well enough that we can eventually approach the true model.

We now state the conditions necessary for our guarantees. First are two standard conditions that are needed to guarantee that the dependency structure can be recovered with any number of samples. One, we must have some set $\Theta \subset \mathbf{R}^M$ of feasible parameters. Two, the true model is in Θ, i.e., there exists some choice of $\theta^* \in \Theta$ such that

$$\begin{aligned} &\pi^*(\Lambda, Y) = p_{\theta^*}(\Lambda, Y), \\ &\forall \Lambda \in \{-1, 0, 1\}^{m \times n}, Y \in \{-1, 1\}^m \end{aligned} \tag{6}$$

where π^* is the true distribution.

Next, let Φ_j denote the set of dependencies that involve either labeling function λ_j or the true label y. For any feasible parameter $\theta \in \Theta$ and $j \in \{1, \ldots, n\}$, there must exist $c > 0$ such that

$$\begin{aligned} cI + \sum_{i=1}^{m} &\mathrm{Cov}_{(\Lambda, Y) \sim p_\theta}(\Phi_j(\Lambda, Y) \mid \Lambda_i = \bar{\Lambda}_i) \\ &\preceq \sum_{i=1}^{m} \mathrm{Cov}_{(\Lambda, Y) \sim p_\theta}(\Phi_j(\Lambda, Y) \mid \Lambda_{i \setminus j} = \bar{\Lambda}_{i \setminus j}). \end{aligned} \tag{7}$$

This means that for each labeling function, we have a better estimate of the dependencies with the labeling function than without. It is analogous to assumptions made to analyze parameter learning in data programming.

Finally, we require that all non-zero parameters be bounded away from zero. That is, for all $\theta_i \neq 0$, and some $\kappa > 0$, we have that

$$|\theta_i| \geq \kappa. \tag{8}$$

Under these conditions, we are able to provide guarantees on the probability of finding the correct dependency structure. First, we present guarantees for all types of possible dependencies in Theorem 1, proved in Appendix A.2. For this theorem, we define d_j to be the number of possible dependencies involving either Λ_j or y, and we define d as the largest of $d_1, \ldots, d_n$.

Theorem 1. *Suppose we run Algorithm 1 on a problem where conditions (6), (7), and (8) are satisfied. Then, for any $\delta > 0$, an unlabeled input dataset of size*

$$m \geq \frac{32d}{c^2\kappa^2} \log\left(\frac{2nd}{\delta}\right)$$

is sufficient to recover the exact dependency structure with a probability of at least $1 - \delta$.

For general dependencies, d can be as large as the number of possible dependencies due to the fact that higher-order dependencies can connect the true label and many labeling functions. The rate of Theorem 1 rate is therefore not directly comparable to that of Ravikumar et al. (2010), which applies to Ising models with pairwise dependencies.

As we demonstrate in Section 5.3, however, real-world applications can be improved by modeling just pairwise correlations among labeling functions. If only considering these dependencies, then d will only be $2n - 1$, rather than the number of potential dependencies. In Corollary 2, we show that a number of samples needed in this case is $O(n \log n)$. Notice that this is sublinear in the number of possible dependencies, which is $O(n^2)$.

Corollary 2. *Suppose we run Algorithm 1 on a problem where conditions (6), (7), and (8) are satisfied. Additionally, assume that the only potential dependencies are accuracy and correlation dependencies. Then, for any $\delta > 0$, an unlabeled input dataset of size*

$$m \geq \frac{64n}{c^2\kappa^2} \log\left(\frac{4n}{\delta}\right)$$

is sufficient to recover the exact dependency structure with a probability of at least $1 - \delta$.

In this case, we see the difference in analyses between the unsupervised and supervised settings. Whereas the rate of Corollary 2 depends on the maximum number of dependencies that could affect a variable in the model class, the rate of Ravikumar et al. (2010) depends cubically on the maximum number of dependencies that actually affect any variable in the true model and only logarithmically in the maximum possible degree. In the supervised setting, the guaranteed rate is therefore tighter for very sparse models. However, as we show in Section 5.1, the guaranteed rates in both settings are pessimistic, and in practice they appear to scale at the same rate.

5. Experiments

We implement our method as part of the open source framework Snorkel[1] and evaluate it in three ways. First, we measure how the probability of returning the exact correlation structure is affected by the problem parameters using synthetic data, confirming our analysis that its sample complexity is sublinear in the number of possible dependencies. In fact, we find that in practice the sample complexity is lower than the theoretically guaranteed rate, matching the rate seen in practice for fully supervised structure learning. Second, we compare our method to estimating structures via parameter learning over all possible dependencies. We demonstrate using synthetic data that our method is $100\times$ faster and more accurate, selecting $1/4$ as many extraneous correlations on average. Third, we apply our method to real-world applications built using data programming, such as information extraction from PubMed journal abstracts and hardware specification sheets. In these applications, users did not specify any dependencies between the labeling functions they authored; however, as we detail in Section 5.3, these dependencies naturally arise, for example due to explicit composing, relaxing, or tightening of labeling function heuristics; related distant supervision sources; or multiple concurrent developers writing labeling functions. We show that learning this structure improves performance over the conditionally independent model, giving an average 1.5 F1 point boost.

[1] snorkel.stanford.edu

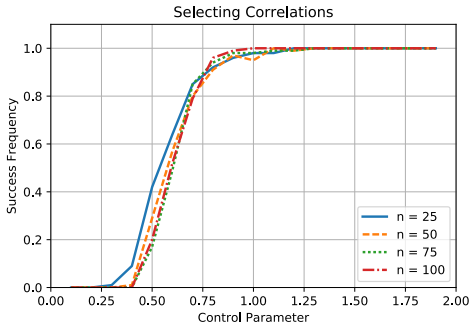

Figure 1. Algorithm 1 returns the true structure consistently when the control parameter γ reaches 1.0 for the number of samples defined by (9). The number of samples required to identify a model in practice scales logarithmically in n, the number of labeling functions.

5.1. Sample Complexity

We test how the probability that Algorithm 1 returns the correct correlation structure depends on the true distribution. Our analysis in Section 4 guarantees that the sample complexity grows at worst on the order $O(n \log n)$ for n labeling functions. In practice, we find that structure learning performs better than this guaranteed rate, depending linearly on the number of true correlations and logarithmically on the number of possible correlations. This matches the observed behavior for fully supervised structure learning for Ising models (Ravikumar et al., 2010), which is also tighter than the best known theoretical guarantees.

To demonstrate this behavior, we attempt to recover the true dependency structure using a number of samples defined as

$$m := 750 \cdot \gamma \cdot d^* \cdot \log n \tag{9}$$

where d^* is the maximum number of dependencies that affect any one labeling function. For example, in the conditionally independent model $d^* = 1$ and in a model with one correlation $d^* = 2$. We vary the control parameter γ from 0.1 to 2.0 to determine the point at which m is sufficiently large for Algorithm 1 to recover the true dependency structure. (The constant 750 was selected so that it succeeds with high probability around $\gamma = 1.0$.)

We first test the effect of varying n, the number of labeling functions. For $n \in \{25, 50, 75, 100\}$, we set two pairs of

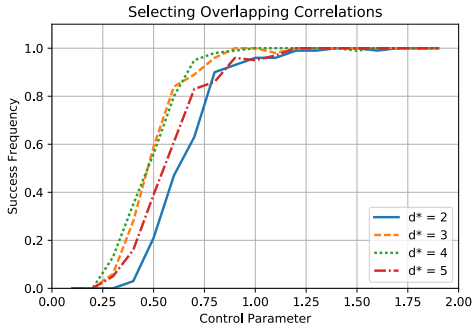

Figure 2. Algorithm 1 returns the true structure consistently when the control parameter γ reaches 1.0 for the number of samples defined by (9). The number of samples required to identify a model in practice scales linearly in d^*, the maximum number of dependencies affecting any labeling function.

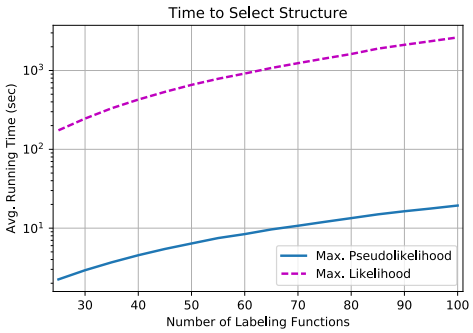

Figure 3. Comparison of structure learning with using maximum likelihood parameter estimation to select a model structure. Structure learning is two orders of magnitude faster.

labeling functions to be correlated with $\theta_{jk}^{\text{Cor}} = 0.25$. We set $\theta_j^{\text{Acc}} = 1.0$ for all j. We then generate m samples for each setting of γ over 100 trials. Figure 1 shows the fraction of times Algorithm 1 returns the correct correlation structure as a function of the control parameter γ. That the curves are aligned for different values of n shows that the sample complexity in practice scales logarithmically in n.

We next test the effect of varying d^*, the maximum number of dependencies that affect a labeling function in the true distribution. For 25 labeling functions, we add correlations to the true model to form cliques of increasing degree. All parameters are the same as in the previous experiment. Figure 2 shows that for increasing values of d^*, (9) again predicts the number of samples for Algorithm 1 to succeed. That the curves are aligned for different values of d^* shows that the sample complexity in practice scales linearly in d^*.

5.2. Comparison with Maximum Likelihood

We next compare Algorithm 1 with an alternative approach. Without an efficient structure learning method, one could maximize the marginal likelihood of the observations $\bar{\Lambda}$ while considering all possible dependencies. To measure the benefits of maximizing the marginal pseudolikelihood, we compare its performance against an analogous maximum likelihood estimation routine that also uses stochastic gradient descent, but instead uses Gibbs sampling to estimate the intractable gradient of the objective.

We create different distributions over n labeling functions by selecting with probability 0.05 pairs of labeling functions to make correlated. Again, the strength of correlation is set at $\theta_{jk}^{\text{Cor}} = 0.25$ and accuracy is set at $\theta_j^{\text{Acc}} = 1.0$. We generate 100 distributions for $n \in \{25, 30, 35, \ldots, 100\}$. For each distribution we generate 10,000 samples and attempt to recover the true correlation structure.

We first compare running time between the two methods. Our implementation of maximum likelihood estimation is designed for speed: for every sample taken to estimate the gradient, a small update to the parameters is performed. This approach is state-of-the-art for high-speed learning for factor graphs (Zhang & Ré, 2014). However, the need for sampling the variables Λ and Y is still computationally expensive. Figure 3 shows that by avoiding variable sampling, using pseudolikelihood is $100\times$ faster.

We next compare the accuracy of the two methods, which depends on the regularization ϵ. The ideal is to maximize the probability of perfect recall with few extraneous correlations, because subsequent parameter estimation can reduce the influence of an extraneous correlation but cannot discover a missing correlation. We tune ϵ independently for each method. Figure 4 (top) shows that maximum pseudolikelihood is able to maintain higher levels of recall than maximum likelihood as the problem size increases. Figure 4 (bottom) shows that even tuned for better recall, maximum pseudolikelihood is more precise, returning $1/4$ as many extraneous correlations. We interpret this improved accuracy as a benefit of computing the gradient for a data point exactly, as opposed to using Gibbs sampling to estimate it as in maximum likelihood estimation.

5.3. Real-World Applications

We evaluate our method on several real-world information extraction applications, comparing the performance of data programming using dependencies selected by our method with the conditionally independent model (Table 1). In the data programming method, users express a variety of weak supervision rules and sources such as regular expression patterns, distant supervision from dictionaries and existing knowledge bases, and other heuristics as labeling functions. Due to the noisy and overlapping nature of these labeling functions, correlations arise. Learning this correlation structure gives an average improvement of 1.5 F1 points.

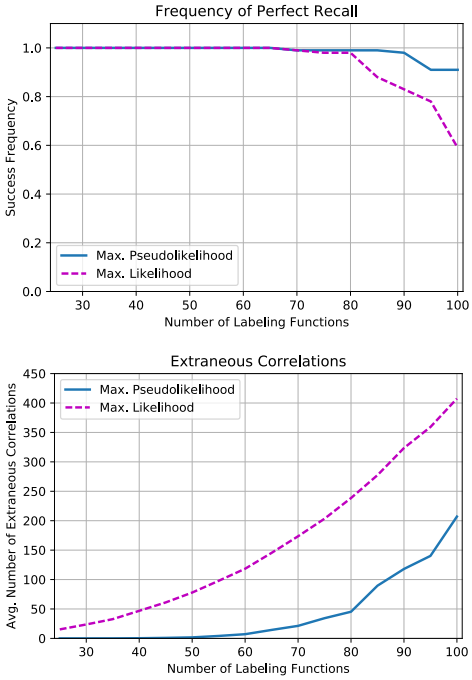

Figure 4. Comparison of structure learning with using maximum likelihood parameter estimation to select a model structure. Even when tuned for better recall (top), structure learning is also more precise, returning $1/4$ as many extraneous correlations (bottom).

Extracting structured information from unstructured text by identifying mentioned entities and relations is a challenging task that is well studied in the context of weak supervision (Bunescu & Mooney, 2007; Alfonseca et al., 2012; Ratner et al., 2016). We consider three tasks: extracting mentions of specific diseases from the scientific literature (*Disease Tagging*); extracting mentions of chemicals inducing diseases from the scientific literature (*Chemical-Disease*); and extracting mentions of electronic device polarity from PDF parts sheet tables (*Device Polarity*). In the first two applications, we consider a training set of 500 unlabeled abstracts from PubMed, and in the third case 100 PDF parts sheets consisting of mixed text and tabular data. We use hand-labeled test sets to evaluate on the candidate-mention-level performance, which is the accuracy of the classifier in identifying correct mentions of specific entities or relations, given a set of candidate mentions. For example, in Chemical-Disease, we consider as candidates all pairs of co-occurring chemical-disease mention pairs as identified by standard preprocessing tools[2].

We see that modeling the correlations between labeling functions gives gains in performance which appear to be correlated with the total number of sources. For example, in the disease tagging application, we have 233 labeling functions, the majority of which check for membership in specific subtrees of a reference disease ontology using different matching heuristics. There is overlap in the labeling functions which check identical subtrees of the ontology, and we see that our method increases end performance by a significant 2.6 F1 points by modeling this structure.

Examining the Chemical-Disease task, we see that our method identifies correlations that are both obviously true and ones that are more subtle. For example, our method learns dependencies between labeling functions that are compositions of one another, such as one labeling function checking for the pattern `[CHEM] induc.* [DIS]`, and a second labeling function checking for this pattern plus membership in an external knowledge base of known chemical-disease relations. Our method also learns more subtle correlations: for example, it selected a correlation between a labeling function that checks for the presence of a chemical mention in between the chemical and disease mentions comprising the candidate, and one that checks for the pattern `.*-induced` appearing in between.

5.4. Accelerating Application Development

Our method is in large part motivated by the new programming model introduced by weak supervision, and the novel hurdles that developers face. For example in the Disease Tagging application above, we observed developers significantly slowed down in trying to to leverage the rich disease ontologies and matching heuristics they had available without introducing too many dependencies between their labeling functions. In addition to being slowed down, we also observed developers running into significant pitfalls due to unnoticed correlations between their weak supervision sources. In one collaborator's application, for every labeling function that referenced the words in a sentence, a corresponding labeling function referenced the lemmas, which were often identical, and this significantly degraded performance. By automatically learning dependencies, we were able to significantly mitigate the effects of such correlations. We therefore envision an accelerated development process enabled by our method.

To further explore the way in which our method can protect against such types of failure modes, we consider adding correlated, random labeling functions to those used in the Chemical-Disease task. Figure 5 shows the average estimated accuracy of copies of a random labeling function. An independent model grows more confident that the random noise is accurate. However, with structure learning, we identify that the noisy sources are not independent and they therefore do not outvote the real labeling functions. In this way, structure learning can protect against failures as users experiment with sources of weak supervision.

[2] ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/index.cgi

Table 1. Candidate-mention scores of information extraction applications trained with data programming using generative models with no dependency structure (*Independent*) and learned dependency structure (*Structure*).

Application	Independent			Structure			F1 Diff.	# LFs	# Cor.	% Corr.
	P	R	F1	P	R	F1				
Disease Tagging	60.4	73.3	66.3	68.0	69.8	68.9	2.6	233	315	1.17%
Chemical-Disease	45.1	69.2	54.6	46.8	69.0	55.9	1.3	33	21	3.98%
Device Polarity	78.9	99.6	88.1	80.5	98.6	88.7	0.6	12	32	48.49%

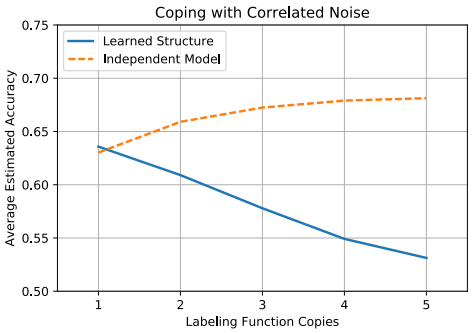

Figure 5. Structure learning identifies and corrects correlated, random labeling functions added to the Chemical-Disease task.

6. Related Work

Structure learning is a well-studied problem, but most work has assumed access to hand-labeled training data. Some of the earliest work has focused on generalized linear models. The lasso (Tibshirani, 1996), linear regression with ℓ_1 regularization, is a classic technique. Zhao & Yu (2006) showed that the lasso is a consistent structure estimator. The Dantzig selector (Candes & Tao, 2007) is another structure estimator for linear models that uses ℓ_1, which can learn in the high-dimensional setting where there are more possible dependencies than samples. Ng (2004) showed that ℓ_1-regularized logistic regression has sample complexity logarithmic in the number of features. ℓ_1 regularization has also been used as a prior for compressed sensing (e.g., Donoho & Elad, 2003; Tropp, 2006; Wainwright, 2009).

Regularized estimators have also been used to select structures for graphical models. Meinshausen & Bühlmann (2006) showed that parameter learning with ℓ_1 regularization for Gaussian graphical models under similar assumptions also consistently selects the correct structure. Most similar to our proposed estimator, Ravikumar et al. (2010) propose a fully supervised pseudolikelihood estimator for Ising models. Also related is the work of Chandrasekaran et al. (2012), which considers learning the structure of Gaussian graphical models with latent variables. Other techniques for learning the structure of graphical models include grafting (Perkins et al., 2003; Zhu et al., 2010) and the information bottleneck approach for learning Bayesian networks with latent variables (Elidan & Friedman, 2005).

Using heuristic sources of labels is increasingly common. Treating labels from a single heuristic source as gold labels is called distant supervision (Craven & Kumlien, 1999; Mintz et al., 2009). Some methods use multi-instance learning to reduce the noise in a distant supervision source (Riedel et al., 2010; Hoffmann et al., 2011). Others use hierarchical topic models to generate additional training data for weak supervision, but they do not support user-provided heuristics (Alfonseca et al., 2012; Takamatsu et al., 2012; Roth & Klakow, 2013a;b). Previous methods that support heuristics for weak supervision (e.g., Bunescu & Mooney, 2007; Shin et al., 2015) do not model the noise inherent in these sources. Also, Downey & Etzioni (2008) showed that PAC learning is possible without hand-labeled data if the features monotonically order data by class probability.

Estimating the accuracy of multiple label sources without a gold standard is a classic problem (Dawid & Skene, 1979), and many proposed approaches are generalized in the data programming framework. Parisi et al. (2014) proposed a spectral approach to estimating the accuracy of members of classifier ensembles. Many methods for crowdsourcing estimate the accuracy of workers without hand-labeled data (e.g., Dalvi et al., 2013; Joglekar et al., 2015; Zhang et al., 2016). In data programming, the scaling of data to label sources is different from crowdsourcing; a relatively small number of sources label all the data. We can therefore learn rich dependency structures among the sources.

7. Conclusion and Future Directions

We showed that learning the structure of a generative model enables higher quality data programming results. Our method for structure learning is also 100× faster than a maximum likelihood approach. If data programming and other forms of weak supervision are to make machine learning tools easier to develop, selecting accurate structures for generative models with minimal user intervention is a necessary capability. Interesting questions remain. Can the guarantee of Theorem 1 be tightened for higher-order dependencies to match the pairwise case of Corollary 2? Preliminary experiments show that they converge at similar rates in practice.

Acknowledgements

Thanks to Christopher De Sa for helpful discussions, and Henry Ehrenberg and Sen Wu for assistance with experiments. We gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) SIMPLEX program under No. N66001-15-C-4043, the DARPA D3M program under No. FA8750-17-2-0095, the National Science Foundation (NSF) CAREER Award under No. IIS- 1353606, the Office of Naval Research (ONR) under awards No. N000141210041 and No. N000141310129, a Sloan Research Fellowship, the Moore Foundation, an Okawa Research Grant, Toshiba, and Intel. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, NSF, ONR, or the U.S. government.

References

Alfonseca, E., Filippova, K., Delort, J.-Y., and Garrido, G. Pattern learning for relation extraction with a hierarchical topic model. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012.

Bunescu, R. C. and Mooney, R. J. Learning to extract relations from the Web using minimal supervision. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007.

Candes, E. and Tao, T. The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

Chandrasekaran, V., Parrilo, P. A., and Willsky, A. S. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1935–1967, 2012.

Craven, M. and Kumlien, J. Constructing biological knowledge bases by extracting information from text sources. In *International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 1999.

Dalvi, N., Dasgupta, A., Kumar, R., and Rastogi, V. Aggregating crowdsourced binary ratings. In *International World Wide Web Conference (WWW)*, 2013.

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

Donoho, D. and Elad, M. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization. *Proceedings of the National Academy of Sciences of the USA*, 100(5):2197–2202, 2003.

Downey, D. and Etzioni, O. Look ma, no hands: Analyzing the monotonic feature abstraction for text classification. In *Neural Information Processing Systems (NIPS)*, 2008.

Eadicicco, L. Baidu's Andrew Ng on the future of artificial intelligence, 2017. Time [Online; posted 11-January-2017].

Elidan, G. and Friedman, N. Learning hidden variable networks: The information bottleneck approach. *Journal of Machine Learning Research*, 6:81–127, 2005.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8): 1771–1800, 2002.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., and Weld, D. S. Knowledge-based weak supervision for information extraction of overlapping relations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.

Joglekar, M., Garcia-Molina, H., and Parameswaran, A. Comprehensive and reliable crowd assessment algorithms. In *International Conference on Data Engineering (ICDE)*, 2015.

Koh, K., Kim, S. J., and Boyd, S. An interior-point method for large-scale ℓ_1-regularized logistic regression. *Journal of Machine Learning Research*, 3:1519–1555, 2007.

Langford, J., Li, L., and Zhang, T. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, 2009.

Meinshausen, N. and Bühlmann, P. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

Metz, C. Google's hand-fed AI now gives answers, not just search results, 2016. Wired [Online; posted 29-November-2016].

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. Distant supervision for relation extraction without labeled data. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2009.

Ng, A. Y. Feature selection, l_1 vs. l_2 regularization, and rotational invariance. In *International Conference on Machine Learning (ICML)*, 2004.

Parisi, F., Strino, F., Nadler, B., and Kluger, Y. Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences of the USA*, 111(4):1253–1258, 2014.

Perkins, S., Lacker, K., and Theiler, J. Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research*, 3: 1333–1356, 2003.

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

Ravikumar, P., Wainwright, M. J., and Lafferty, J. D. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

Riedel, S., Yao, L., and McCallum, A. Modeling relations and their mentions without labeled text. In *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2010.

Roth, B. and Klakow, D. Feature-based models for improving the quality of noisy training data for relation extraction. In *Conference on Information and Knowledge Management (CIKM)*, 2013a.

Roth, B. and Klakow, D. Combining generative and discriminative model scores for distant supervision. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 2013b.

Shin, J., Wu, S., Wang, F., De Sa, C., Zhang, C., and Ré, C. Incremental knowledge base construction using DeepDive. *Proceedings of the VLDB Endowment*, 8(11): 1310–1321, 2015.

Takamatsu, S., Sato, I., and Nakagawa, H. Reducing wrong labels in distant supervision for relation extraction. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58(1): 267–288, 1996.

Tropp, J. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.

Wainwright, M. J. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.

Zhang, C. and Ré, C. DimmWitted: A study of main-memory statistical analytics. *Proceedings of the VLDB Endowment*, 7(12):1283–1294, 2014.

Zhang, Y., Chen, X., Zhou, D., and Jordan, M. I. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research*, 17:1–44, 2016.

Zhao, P. and Yu, B. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

Zhu, J., Lao, N., and Xing, E. P. Grafting-Light: Fast, incremental feature selection and structure learning of Markov random fields. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2010.