Socratic Learning

Rose Yu*[1], Paroma Varma* [2], Dan Iter[2], Christopher De Sa[2], and Christopher Ré[2]

[2]Department of Computer Science, Stanford University
[1]Department of Computer Science, University of Southern California

Abstract

Modern machine learning techniques, such as deep learning, often use discriminative models that require large amounts of labeled data. An alternative approach is to use a generative model, which leverages heuristics from domain experts to train on unlabeled data. Domain experts often prefer to use generative models because they "tell a story" about their data.Unfortunately, generative models are typically less accurate than discriminative models. Several recent approaches combine both types of model to exploit their strengths. In this setting, a misspecified generative model can hurt the performance of subsequent discriminative training. To address this issue, we propose a framework called *Socratic learning* that automatically uses information from the discriminative model to correct generative model misspecification. Furthermore, this process provides users with interpretable feedback about how to improve their generative model. We evaluate Socratic learning on real-world relation extraction tasks and observe an immediate improvement in classification accuracy that could otherwise require several weeks of effort by domain experts.

1 Introduction

For many machine learning tasks, unlabeled examples are significantly easier to obtain than labeled ones. The recently proposed data programming [3] paradigm uses a generative model to noisily label unlabeled datasets by incorporating user-defined heuristics called *labeling functions*. These noisy labels are then used to train a discriminative model for the desired task. There are two underlying issues with the above approach. First, the generative model may be misspecified, which can affect subsequent discriminative training. Second, the errors in the generative labeling provide users little intuition about what went wrong and how it can be fixed. In both cases, there is no systematic approach to improving the generative model.

In this paper, we propose the framework of *Socratic learning*, an iterative process that improves the generative model by using information from the discriminative model. This knowledge transfer is performed via features that are easily interpretable. By comparing the predictions from the two models, Socratic learning can automatically detect which features may have a hidden effect on the generative model. We then address the model misspecification issue by incorporating those features into the generative model. In a similar way, this mechanism provides users with interpretable information (in the form of features) of how they can write more effective labeling functions.

Model misspecification is also an issue in the classic semi-supervised learning setting. There, a large amount of unlabeled data is used for training along with a small amount of labeled data. Common methods use generative models such as Bayes nets, mixture models [6], and deep neural networks [2] to train on the unlabeled data. Other methods, such as Generative Adversarial Network (GAN)[1], jointly train generative and discriminative models. In comparison with these approaches, our method models the relationship between the user-specified heuristic labels and the true class, rather than

*The authors contributed equally to this work

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

modeling the features that are used by the discriminative model. This both discourages overfitting and simplifies the iterative construction of the generative model.

In summary, our contributions are as follows:

- We introduce a novel framework in Section 2 to address the model misspecification issue in distant supervision by enabling knowledge transfer between generative and discriminative models.
- We demonstrate cases where the generative model can be further improved, without user intervention, using feedback from the discriminative model. We provide theoretical guarantees in Theorem 1 for this improvement.
- We demonstrate that the improved generative model can result in discriminative models with higher prediction accuracy. We report results of using Socratic learning for disease mention relation extraction task in Section 3.

2 Methodology

We would like a simple and efficient pipeline where we can programmatically "debug" the heuristic rules and improve the generative model. In Socratic learning, we adopt a new perspective that the generative model can be corrected using the feedback from the discriminative model. We start by describing our problem setting.

Given a set of N objects $\{\mathcal{O}\}$, we can describe them using D number of features $X = \{X_d\} : \mathcal{O} \to \{-1, 1\}$. We also observe M different labeling functions $\Lambda = \{\Lambda_i\} : \mathcal{O} \to \{-1, 0, 1\}$. The true class $Y : \mathcal{O} \to \{-1, 1\}$ is hidden. In this setting, several recent approaches such as data programming [3] have proposed to combine generative and discriminative models to achieve the best of both worlds.

2.1 Generative and Discriminative Models

Data programming unifies the following generative and discriminative models to describe the relationship between the observations (labeling functions and features) and the true class:

$$G : p(\Lambda, Y) = \frac{1}{Z} \exp\{\phi^T \Lambda Y\} \quad D : p(Y = 1|X) \propto \exp(\theta^\top X). \tag{1}$$

Data programming estimates the parameters, ϕ, via maximum likelihood estimation and computes the marginal distribution over the training labels. It then feeds the marginals from the generative model into a discriminative model to make predictions, as described in Figure 2a.

However, this approach does not take into account the hidden effect of features on the generative model. Consider the example in Fig 1. The labeling function has an accuracy of 75% over the entire dataset. However, if we partition the data based on a particular feature, this labeling function has accuracies of 95% and 65% in the two sections. This observation suggests that the generative model can be improved if it incorporates such features.

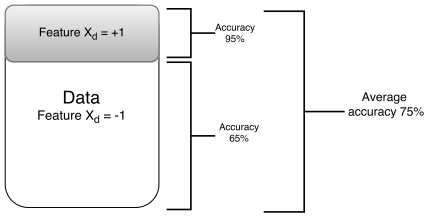
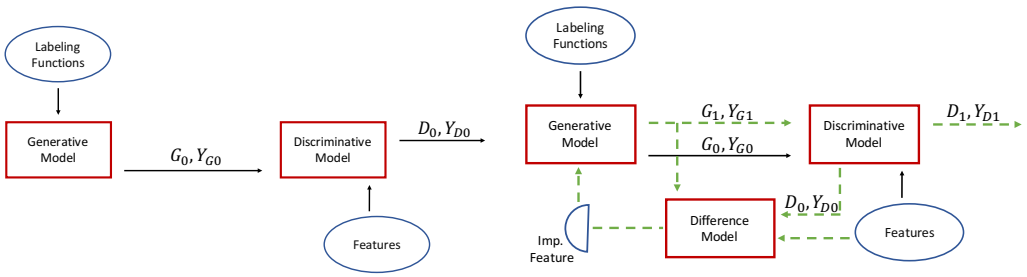


Figure 1: A labeling function has different accuracies in the data given a particular feature partitioning.

2.2 Socratic Learning

Socratic learning jointly models the relationship between the labeling functions, the true class, and the features. We generalize the generative model from Eqn 1 to the following form:

$$G : p(\Lambda, Y, X_S) \propto \exp\{\phi^\top \Lambda Y + (\Lambda Y)^\top \mathbf{W} X_S\} \quad D : p(Y = 1|X) \propto \exp(\theta^\top X). \tag{2}$$



(a) Data programming pipeline

(b) Socratic learning pipeline

Figure 2: The black arrows represent the flow of information in the data programming pipeline and the green arrows refer to the Socratic learning feedback loop.

Algorithm 1 Socratic Learning

1: **Input**: labeling function $\Lambda \in \{-1, 0, 1\}^{N \times M}$, $X \in \{-1, 1\}^{N \times D}$
2: $S = \emptyset$
3: **repeat**
4: Learn generative model $G(\Lambda, X_S, Y)$ and compute labels Y_G
5: Learn discriminative model $D(Y|X)$ and compute labels Y_D
6: Compute disagreement $Z = -Y_G Y_D$
7: Select features X_d that is most indicative of Z
8: $S = S \cup \{d\}$
9: **until** performance stops increasing

where $\phi \in \mathbb{R}^M$ are the coefficients for the labeling functions and $\mathbf{W} \in \mathbb{R}^{M \times K}$ are the weights for the labeling functions that are dependent on features. $\mathbf{W}$ has only K columns, which means that the accuracy of the labeling functions depends only on a few features X_S.

As shown in Fig 2b, Socratic learning initializes with a simple generative model, which could be misspecified. It then compares the predictions from the generative and discriminative models, Y_G and Y_D, and computes the disagreement of the two models $Z = -Y_G Y_D$. It builds a difference model for the disagreement with respect to features. By running LASSO, it can find the important feature X_d that is most indicative of the difference. Finally it incorporates this feature into the generative model.

The process iteratively identifies the features that have hidden effects on the generative model. It refines the generative model by expanding the support set S of features and stops when the generative model starts to overfit [2]. This procedure is described in Algorithm 1. The following theorem guarantees the successful recovery of the features of Socratic learning.

Theorem 1 *Given the generative and discriminative models of Socratic learning, let the discriminative prediction be Y_D, if $Y_D Y \perp X_d$ and $Y_D Y \perp \Lambda Y$, Socratic learning can correctly recover the support S and improve the generative model with probability $1 - 2\exp\{-c_2 \lambda_N^2 N\}$ for the regularization parameter $\lambda_N \geq c_1 \sqrt{\frac{\log D}{N}}$.*

3 Experimental Results

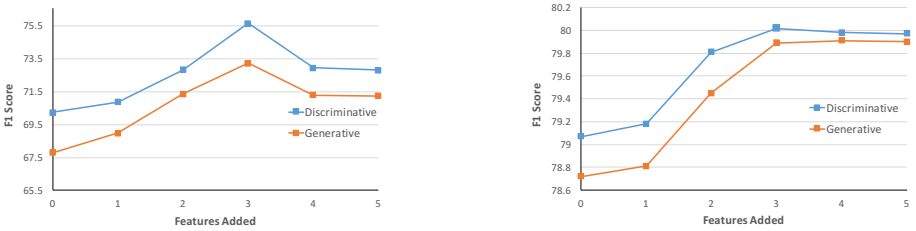
We report the performance of Socratic learning on a real world relation extraction task. In this setting, we are given a set of candidates, which are substrings of a sentence. The goal is to classify whether those candidates are in a given relationship. We use the data from the BioCreative CDR Challenge [5], where mentions of diseases are extracted from PubMed abstracts and the labeling functions are written by a team of domain experts.

We compare our performance against (1) fully supervised (FS), (2) majority vote (MV) and (3) data programming (DP) F1 scores. FS uses the true labels in training, MV uses majority vote across all

[2] This is determined by the prediction performance on the hold-out validation set.

Table 1: F1 scores for Socratic Learning and baseline methods with hand-tuned features.

		Baseline F1			Socratic learning F1	
Application	# of LFs	FS	MV	DP	1F	3F
Disease Tagging-32	32	86.47	84.42	85.28	85.30	85.29
Disease Tagging-16	16	86.47	77.98	79.07	79.18	80.02



(a) Simulation

(b) Disease Tagging-16

Figure 3: Improvement in F1 Score for generative and discriminative models with addition of features. Best improvement is achieved at 3 features in both scenarios.

labeling functions as the generative model, and DP uses the data programming paradigm without considering features [3].

Table 1 displays the performance of different methods in two settings. Disease Tagging-32 uses 32 labeling functions. In this case, Socratic learning does not show much improvement since the labeling functions are already heavily engineered. For Disease Tagging-16, we use labeling functions that rely only on using dictionaries and regex rules. These labeling functions tend to either be more noisy or have lower coverage than the extensively hand-tuned labeling functions. In this case, Socratic learning is able to increase by 1 point in F1 score on average, an improvement which could otherwise require several weeks of effort by a domain expert.

The process can increase prediction accuracy by iteratively adding multiple features. In both simulation (Fig 3a) and real data (Fig 3b), we observe consistent improvement in F1 score with addition of features. The generative model can be refined until the difference model is unable to find any feature that could affect the generative model.

Our method also provides users with interpretable information (in the form of features) of how they can write better labeling functions. For example, we discover that the feature phrase *for induction of...* is negatively correlated with the accuracy of labeling function *non-common diseases*. *Non-common diseases* works by searching in a dictionary of predefined non-common diseases such as anesthesia and pregnancy. It returns -1 if it finds a match. Phrases like "for induction of anesthesia" show up a fair amount in a Google/PubMed query, which does not indicate a true negative disease relation. This serves as a reminder for users to consider the presence of *for induction of...* when designing the labeling function *non-common diseases*.

4 Conclusion and Future Work

We introduced Socratic learning, a novel framework that can initiate a cooperative dialog between the generative and the discriminative model. We demonstrated how the generative model can be further improved, without user intervention, using feedback from the discriminative model. Finally, we showed a case study of how Socratic learning can educate domain experts about the hidden effect of important features that can help them provide better heuristics.

For the future work, we hope to explore and improve upon the Socratic learning framework. One can imagine a scenario where the features are too sparse, and the discriminative training can hardly provide useful information for the generative model. A potential solution would be to construct a "super" feature by aggregating multiple sparse features. Another possible direction is to look at associative rule-based learning to discover interesting relations among features that can further inform the user about the labeling function design.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[2] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[3] Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *arXiv preprint arXiv:1605.07723*, 2016.

[4] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

[5] CH Wei, Y Peng, R Leaman, AP Davis, CJ Mattingly, J Li, TC Wiegers, and Z Lu. Overview of the biocreative v chemical disease relation (cdr) task. In *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, pages 154–166, 2015.

[6] Xiaojin Zhu. Semi-supervised learning literature survey. 2005.

A Theoretical Analysis

Theorem 1 *Given a generative model* $G : p(\Lambda, Y, X_S) \propto \exp\{\phi^\top \Lambda Y + (\Lambda Y)^\top \mathbf{W} X_S\}$ *where* $\phi \in \mathbb{R}^M$, $\mathbf{W} \in \mathbb{R}^{M \times K}$, *and* $X_S \subset X$ *are the features in the support set where the labeling functions depend upon. Given a discriminative model* $D : p(Y = 1|X) \propto \exp(\theta^\top X)$. *Let the discriminative prediction be* Y_D, *if* $Y_D Y \perp X_d$ *and* $Y_D Y \perp \Lambda Y$, *Socratic Learning can correctly recover the support* S *with probability* $1 - 2\exp\{-c_2 \lambda_N^2 N\}$ *for* $\lambda_N \geq c_1 \sqrt{\frac{\log D}{N}}$.

Let us consider only one feature d is in the support set S, that is $X_S = X_d$. X_d is correlated with ΛY. Assume the data is generated from the following generative model

$$G : p(\Lambda, Y, X_d) \propto \exp(\phi^\top \Lambda Y + \mathbf{w}^\top X_d \Lambda Y)$$

Suppose using maximum likelihood estimation (MLE), we can obtain an estimation of the generative model parameter $\hat{\phi}$. As MLE for exponential family is consistent, we know that $\text{plimit}_{N \to \infty} \hat{\phi} = \phi$.

The discriminative model tries to minimize the following logistic loss:

$$\begin{aligned} l(\theta) &= \mathbb{E}_{\Lambda, Y \sim G(\phi)} \log \sigma(\theta^\top XY) \\ &= p(Y = 1|\Lambda) \log \sigma(\theta^\top X) + p(Y = -1|\Lambda)(1 - \log \sigma(\theta^\top X)) \\ &= \sigma(\phi^\top \Lambda) \log \sigma(\theta^\top X) + (1 - \sigma(\phi^\top \Lambda))(1 - \log \sigma(\theta^\top X)) \end{aligned}$$

The gradient of the expected log-likelihood over the samples is

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \theta_d} &= \sigma(\phi^\top \Lambda)(1 - \sigma(\theta^\top X) - (1 - \sigma(\phi^\top \Lambda) \log \sigma(\theta^\top X) \\ &= \sigma(\phi^\top \Lambda) - \sigma(\theta^\top X)] X_d = 0 \end{aligned}$$

We can obtain an estimation of discriminative model parameter $\hat{\theta}$ by any optimization algorithm.

Now let us look at the difference between generative model and discriminative model. We compute the marginal distribution from G as $\tanh(\hat{\phi}^\top \Lambda)$, and the labels predicted by the discriminative model is $Y_D = \tanh(\hat{\theta}^\top X)$. For each feature X_d, the correlation between X_d and the difference model is

$$\mathbb{E}[X_d \tanh(\hat{\phi}^\top \Lambda) Y_D] = \mathbb{E}[X_d \tanh(\hat{\phi}^\top \Lambda Y_D)] = \mathbb{E}[X_d \tanh(\hat{\phi}^\top \Lambda Y Y Y_D)]$$

Suppose $Y_D Y \perp X_d$ and $Y_D Y \perp \Lambda Y$, we have

$$\mathbb{E}[X_d \tanh(\hat{\phi}^\top \Lambda Y Y Y_D)] = \mathbb{E}[X_d \tanh(\hat{\phi}^\top \Lambda Y] \mathbb{E}[Y Y_D)]$$

If $X_{\neg d} \perp \Lambda Y$, then $\mathbb{E}[X_{\neg d} \tanh(\hat{\phi} \Lambda Y)] = \mathbb{E}[X_{\neg d}] \mathbb{E}[\tanh(\hat{\phi}^\top \Lambda Y)] = 0$. And the difference model is sparse on $X_{\neg d}$.

Given that the difference model is truly sparse on uncorrelated features, we would be able to recover the support with high probability. Denote Q^{SS} as the Fisher information matrix corresponding to the support covariates, suppose the features satisfy the following assumptions

- A1: dependency condition $\Lambda_{min}(Q_{SS} \geq C_{min})$
- A2: mutual incoherence condition $\|Q_{S^c S}^\top Q_{SS}^{-1}\|_\infty \leq 1 - \alpha$

Lemma 1 *With the difference model as* $\tanh(\hat{\phi}^\top \Lambda) Y_D = \beta X + \epsilon$ *where* $\beta \in \mathbb{R}^D$ *and* $S(\beta) = S(\beta) = \{i = 1, \cdots, D, \beta \neq 0\}$ *where* $|S| = K$, *if* $N \geq K^3 \log D$, *recovery support by running LASSO is almost successful with probability* $1 - 2\exp\{-c_2 \lambda_N^2 N\}$ *for* $\lambda_N \geq c_1 \sqrt{\frac{\log D}{N}}$.

This directly follows from the conclusion of Ising model support recovery [4]. After we recover the support feature X_d, we can add the correlated feature to be close to the underlying generative model. Thus Socratic Learning leads to better prediction performance than data programming.