Estimating Accuracy from Unlabeled Data: A Bayesian Approach

Emmanouil Antonios Platanios E.A.PLATANIOS@CS.CMU.EDU
Avinava Dubey AKDUBEY@CS.CMU.EDU
Tom Mitchell TOM.MITCHELL@CS.CMU.EDU
Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract

We consider the question of how unlabeled data can be used to estimate the true accuracy of learned classifiers, and the related question of how outputs from several classifiers performing the same task can be combined based on their estimated accuracies. To answer these questions, we first present a simple graphical model that performs well in practice. We then provide two nonparametric extensions to it that improve its performance. Experiments on two real-world data sets produce accuracy estimates within a few percent of the true accuracy, using solely unlabeled data. Our models also outperform existing state-of-the-art solutions in both estimating accuracies, and combining multiple classifier outputs.

1. Introduction

Estimating accuracy of classifiers is central to machine learning and many other fields. Accuracy is defined as the probability of a classifier output disagreeing with the true underlying label, over the true distribution of input data to that classifier. Most existing approaches to estimating accuracy are *supervised*, meaning that labeled examples are required for the estimation. This paper presents an *unsupervised* approach for estimating accuracies, meaning that only *unlabeled data* are required. Being able to estimate the accuracies of classifiers using only unlabeled data is important for any autonomous learning system that operates under no supervision. It is also useful in a transfer learning setting, when classifiers trained using data from one distribution must be applied to data from a new distribution. This is actually an omnipresent scenario (it is not uncommon for the data used to train a classifier to be distributed differently than the data the classifier makes predictions for).

We propose an approach based on a probabilistic graphical model that allows us to estimate accuracies of classifiers using only unlabeled data. Our approach further allows us to infer the posterior distribution of a single label for each data sample, jointly with the accuracies of our classifiers, and it can also handle missing data. That is the case with data samples for which a classifier might not have predicted any label. This can happen when the classifier does not have any features for those data samples, for example, and it is not uncommon in practice (an example of such a case is the system described in the next paragraph). Moreover, we propose two nonparametric extensions to that model that allow sharing of information among different classification problems and different classifiers, that are useful in the case of limited data. We present experimental results demonstrating the success of our approaches in estimating classification accuracies in two diverse domains. Furthermore, we also present results showing that our method outperforms existing methods for combining classifier outputs into a single label.

We consider a problem setting where we have several different approximations $\hat{f}_1, \ldots, \hat{f}_N$, to some target boolean classification function, $f : \mathcal{X} \to \{0, 1\}$, and we wish to know the true accuracies of each of these different approximations, using only unlabeled data. We also want to know the single most likely output label, meaning the most likely response of the true underlying function f. These function approximations can be from any source. For example, they can be learned classifiers or even human workers in a *crowdsourcing* setting. One example of this setting that we consider here is taken from the Never Ending Language Learning system (NELL) (Carlson et al., 2010; Mitchell et al., 2015). Out of NELL's over 2,500 learning tasks, many involve learning classifiers that map noun phrases (NPs) to boolean categories such as fruit, and food. For each such boolean classification function, NELL learns several different approximations based on different views of the NP. One approximation is based on the orthographic features of the NP (e.g., a NP ending with the letter string "burgh" provides statistical evidence that the referenced entity may be a city), whereas another uses phrases surrounding the NP (e.g., a NP followed by the word sequence "mayor of" provides statistical evidence that the referenced entity may be a city). The aim of this paper is to find a way to estimate the error rates of each one of these approxi-

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

mations to the underlying function f, using only unlabeled data, and to infer the posterior distribution of the response of function f while accounting for these error rates.

2. Related Work

The setting we are considering was previously explored by Collins & Singer (1999), Dasgupta et al. (2001), Bengio & Chapados (2003), Madani et al. (2004), Schuurmans et al. (2006), Balcan et al. (2013), and Parisi et al. (2014), among others. Most of their approaches, however, made certain strong assumptions, such as assuming independence given the labels, or assuming knowledge of the true distribution of the labels. Platanios et al. (2014) provided an analysis of this related work and then formulated the problem of estimating the error rates of several approximations to a function as an optimization, using the agreement rates over unlabeled data while relaxing some of those assumptions. However, their method, as well as most of the related work, is unable to handle missing data and also does not directly deal with the problem of combining classifier outputs into a single label. Collins & Huynh (2014) review many methods that have been proposed for estimating the accuracy of medical tests in the absence of a gold standard. This is effectively the same problem that we are considering, applied to the domains of medicine and biostatistics. They start by presenting a method for estimating the accuracy of tests, where those tests are applied in multiple different populations (i.e., different input data), while assuming that the accuracies of the tests are the same across those populations, and that the test results are independent conditional on the true "output label". These are similar assumptions to the ones made by several papers already mentioned above, but the idea of applying the tests to multiple populations is new. Collins & Huynh (2014) also review many methods that relax these assumptions in different ways and they also briefly discuss some Bayesian models for doing so.

Dawid & Skene (1979) were the first to formulate the problem in terms of a graphical model and Moreno et al. (2015) proposed a nonparametric extension applied to crowdsourcing. The current state-of-the-art, to the best of our knowledge, is the work of Tian & Zhu (2015) and also comes from the area of crowdsourcing. The authors proposed an interesting max-margin majority voting scheme for combining classifier outputs. As we show in our experiments, our methods are able to outperform their approach, while at the same time being significantly less complicated.

3. Motivation and Intuition

It can be observed from the related work that most of it is trying to relate agreement rates between different classifiers (which are observed) with the accuracies of these classifiers. We follow a similar approach in this paper. More specifically, one of our goals is to shed light on the more general question of *how the consistency among multiple functions is related to their true accuracies*. We now present an example that provides some intuition behind why one might want to use agreement rates as indicators of correctness, and what issues might arise in doing so.

Let us consider a case where a person asks 10 different people a question that is related to politics and 8 of these people agree on an answer. One might immediately think that since we have such a strong majority, that answer must be correct. However, one has to be careful. Let us assume that these 8 people that agree belong to the same political party and that the 2 people that gave a different answer belong to some other party. In this case, we might want to reconsider whether that answer is correct and to what extent we trust it. Now, if 7 of the people from that party were in agreement and 1 person from the other party had also agreed with them, then we would trust that answer even more. We therefore see that *there is a relationship between how dependent the agreeing functions are and the question of whether consistency implies correctness*. One trivial example that reinforces this argument is when our multiple functions are in fact copies of the same function and thus fully dependent. In this case, consistency among these functions gives us no information about their correctness (i.e., they are always consistent with each other in their responses). Our paper aims to formalize this intuition by developing a probabilistic framework to reason about them.

This example also raises a new question: *if functions that are highly dependent disagree, then what does this imply about the question being asked, or about the functions themselves?* In the example where people are asked political questions, it might imply that the question has no single underlying true answer. In general, this case may correspond to the classification problem being too hard, or the functions being too uncertain about their answers. This is an interesting question to explore that relates to the field of *active learning*, but is outside the scope of our paper.

4. Proposed Methods

We first propose a simple and elegant probabilistic graphical model that, as we show in the experiments section, achieves better accuracy in error rates estimation than the current state-of-the-art and, at the same time, combines the outputs of several function approximations to produce a single label for each data example, and is also able to handle missing data. We then extend that model so that information can be shared across different classification problems, which is especially important in the case of limited data. Finally, we further extend the model to group examples according to various function approximations, which as shown in most previous work (Moreno et al., 2015), plays an important role in error rate estimation.

4.1. Bayesian Error Estimation

Following from the introduction, we consider a "multiple approximations" problem setting in which we have sev-

eral different approximations, $\hat{f}_1, \ldots, \hat{f}_N$, to some target boolean classification function, $f : \mathcal{X} \to \{0, 1\}$, and we wish to know the true accuracies of each of these different approximations, using only unlabeled data, as well as the single most likely single label, meaning the most likely response of the true underlying function f. We define the following generative process to do that, where we are only given a set of unlabeled data $X_1, \ldots, X_S$ and the function approximations $\hat{f}_1, \ldots, \hat{f}_N$:

1. Let us make the assumption that there is an underlying distribution from which the labels for all the data examples are sampled. We first draw $p \sim \text{Beta}(\alpha_p, \beta_p)$, representing the prior probability for the true label being equal to 1, over all possible examples.
2. For each data example, X_i where $i = 1, \ldots, S$, we draw a label $\ell_i \sim \text{Bernoulli}(p)$. This label is the true label $f(X_i)$.
3. Let us further assume that there is another underlying distribution from which the error rates of our function approximations are sampled. For each function approximation, $\hat{f}_j$ where $j = 1, \ldots, N$, we draw an error rate $e_j \sim \text{Beta}(\alpha_e, \beta_e)$.
4. Finally, we can assume that each function takes the sampled label for each example and flips it with probability equal to its error rate (thus making an error). It then outputs the resulting label. Thus, for each data example, X_i, and function approximation, $\hat{f}_j$, we draw an output label, $\hat{f}_{ij}$, according to the following distribution:

$$\hat{f}_{ij} = \begin{cases} \ell_i & , \text{ with probability } 1 - e_j, \\ 1 - \ell_i & , \text{ otherwise.} \end{cases} \tag{1}$$

   This output label corresponds to $\hat{f}_j(X_i)$.

We emphasize the last step in the generative process, where with probability equal to the function error rate, the correct label is flipped and the function approximation makes an error. A graphical representation of the model, along with a compact definition, is shown in figure 1.

In order to perform inference for this simple model we use *Gibbs sampling* (Geman & Geman, 1984), a well-known Markov Chain Monte Carlo (MCMC) sampling approach. The conditional probabilities we use during sampling are as follows:

$$P(p \mid \cdot) = \text{Beta}(\alpha_p + \sigma_{\boldsymbol{\ell}}, \beta_p + S - \sigma_{\boldsymbol{\ell}}), \tag{2}$$

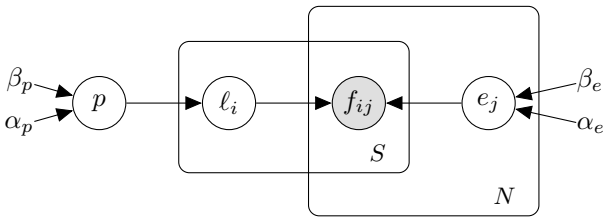
$$P(\ell_i \mid \cdot) \propto p^{\ell_i}(1-p)^{1-\ell_i}\pi_i, \tag{3}$$

$$P(e_j \mid \cdot) = \text{Beta}(\alpha_e + \sigma_j, \beta_e + S - \sigma_j), \tag{4}$$

where:

$$\sigma_{\boldsymbol{\ell}} = \sum_{i=1}^{S} \ell_i, \quad \sigma_j = \sum_{i=1}^{S} \mathbb{1}_{\{\hat{f}_{ij} \neq \ell_i\}}, \tag{5}$$

$$\pi_i = \prod_{j=1}^{N} e_j^{\mathbb{1}_{\{\hat{f}_{ij} \neq \ell_i\}}} (1 - e_j)^{\mathbb{1}_{\{\hat{f}_{ij} = \ell_i\}}}, \tag{6}$$



$$p \sim \text{Beta}(\alpha_p, \beta_p),$$
$$\ell_i \sim \text{Bernoulli}(p), \text{ for } i = 1, \ldots, S,$$
$$e_j \sim \text{Beta}(\alpha_e, \beta_e), \text{ for } j = 1, \ldots, N,$$
$$\hat{f}_{ij} = \begin{cases} \ell_i & , \text{ with probability } 1 - e_j, \\ 1 - \ell_i & , \text{ otherwise.} \end{cases}$$

Figure 1: Simple probabilistic graphical model for error rate estimation using only unlabeled data.

and $\mathbb{1}_{\{\cdot\}}$ evaluates to one if its subscript's argument statement is true and to zero otherwise. We sequentially sample from those three distributions, by sampling each random variable while keeping the others fixed to their last sampled value. The distribution of the samples we obtain is guaranteed to converge to the true posterior distribution of our random variables, given that we obtain a large enough number of samples.

Note that it is easy to handle missing data when using this model (in contrast to other methods presented in the related work section), as we can model the missing data as latent variables which themselves can be inferred in the Gibbs sampling algorithm. The conditional probability for $\hat{f}_{ij}$, in case it needs to be sampled, is as follows:

$$P(\hat{f}_{ij} \mid \cdot) \propto e_j^{\mathbb{1}_{\{\hat{f}_{ij} \neq \ell_i\}}} (1 - e_j)^{\mathbb{1}_{\{\hat{f}_{ij} = \ell_i\}}}. \tag{7}$$

4.2. Coupled Bayesian Error Estimation

Up to this point we have assumed that there is a single target function and multiple approximations to that function. More generally though, we might have multiple target functions, or problem settings, and a common set of learning algorithms used for learning each of these. For

IMPLICIT USE OF AGREEMENT RATES

Many of the papers from section 2 propose using agreement rates between the function approximations in order to estimate the error rates of these functions. By looking at equations 3, 4, 5, and 6 we can see that our method is also implicitly using agreement rates in order to estimate function error rates. We are using the agreement between the function outputs and the true underlying labels in order to infer both the error rates of our functions and those labels, jointly. This fundamental connection further supports the argument made in (Platanios et al., 2014) relating agreement and correctness, in that under certain conditions, agreement of several functions implies correctness of these functions.

example, this is the case in NELL, where the different target functions correspond to different boolean classification problems (e.g., classifying NPs as cities or not, as animals or not, etc.). Multiple learning methods are utilized to approximate each one of those target functions (e.g., a classifier based on the NP morphology, a second classifier based on the NP contexts, etc.), so that each such classification problem, or target function, corresponds to an instance of our "multiple approximations" problem setting.

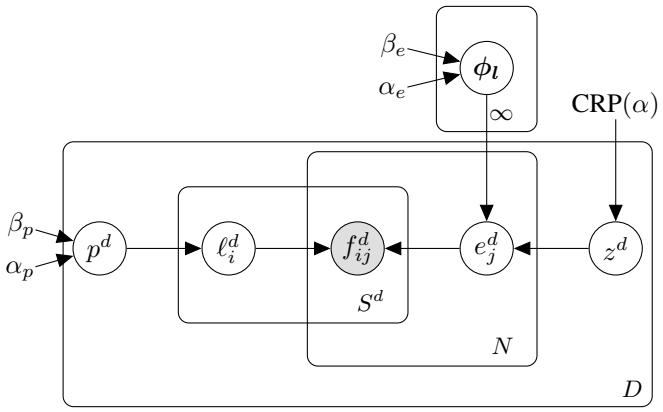
It is reasonable to assume that there are some structural dependencies between our function approximations that could result in similar behavior (i.e., similar error rate across multiple domains). Note that this is not an unreasonable assumption because these classifiers use the same set of features across all domains. If that is indeed the case, then sharing information across domains might prove useful. That is our motivation for the extension to the model introduced earlier, that we present in this section. The main idea is that we want to cluster our domains based on the distribution of the error rates of our function approximations. However, we do not know the number of clusters needed, and that is why we resorted to Bayesian nonparametrics; we want to infer the necessary number of clusters "automatically". More specifically, we decided to use a Dirichlet process (DP) prior. Note that this model is different than the one proposed in (Moreno et al., 2015), in that in that paper the authors propose clustering the classifiers while only considering a single domain, instead of clustering the domains, as we do. In the following two sections we provide an introduction to DPs and introduce our improved model.

Dirichlet Process (DP): The Dirichlet process is a distribution over discrete probability measures (i.e., atoms), $G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$, with countably infinite support, where the finite-dimensional marginals are distributed according to a finite Dirichlet distribution (Ferguson, 1973). It is parametrized by a base probability measure H, which determines the distribution of the atom locations, and a concentration parameter $\alpha > 0$ that is proportional to the inverse variance of the atom locations. The DP can be used as the distribution over mixing measures in a nonparametric mixture model. In the DP mixture model (Antoniak, 1974), data samples, $\{x_i\}_{i=1}^n$, are assumed to be generated according to the following process:

$$G \sim \text{DP}(\alpha, H)\,, \qquad \theta_i \sim G\,, \qquad x_i \sim f(\theta_i)\,. \tag{8}$$

While the DP allows for an infinite number of clusters a priori, any finite dataset will be modeled using a finite, but random, number of clusters.

Model: In the definition of our model we are going to use the Chinese restaurant process (CRP) representation of the DP (Blackwell & MacQueen, 1973), because that form is most appropriate for deriving the Gibbs sampling equations to perform inference, later on. Following from the intuition provided in the beginning of section 4.2, we now have a problem setting in which we have several different



$$p^d \sim \text{Beta}(\alpha_p, \beta_p), \text{ for } d = 1, \ldots, D,$$
$$\ell_i^d \sim \text{Bernoulli}(p^d), \text{ for } i = 1, \ldots, S^d, \text{ and } d = 1, \ldots, D,$$
$$[\phi_l]_j \sim \text{Beta}(\alpha_e, \beta_e), \text{ for } j = 1, \ldots, N, \text{ and } l = 1, \ldots, \infty,$$
$$z^d \sim \text{CRP}(\alpha), \text{ for } d = 1, \ldots, D,$$
$$e_j^d = [\phi_{z^d}]_j, \text{ for } j = 1, \ldots, N, \text{ and } d = 1, \ldots, D,$$
$$\hat{f}_{ij}^d = \begin{cases} \ell_i^d & \text{, with probability } 1 - e_j^d, \\ 1 - \ell_i^d & \text{, otherwise.} \end{cases}$$

Figure 2: Graphical model for coupled error rate estimation using only unlabeled data. The coupling comes from the use of a Dirichlet process prior to group problem domains, and share information within each group. Note that CRP(α) denotes the Chinese restaurant process (CRP) with concentration parameter α.

domains, $d = 1, \ldots, D$, where for each domain d, we have a set of function approximations, $\hat{f}_1^d, \ldots, \hat{f}_N^d$, to some target boolean classification function, $f^d : \mathcal{X} \to \{0, 1\}$, and we wish to know the true accuracies of each of these different approximations, using only unlabeled data, as well as the single most likely single label, meaning the most likely response of the true underlying function f. We define the following generative process to do that, where we are only given D sets of unlabeled data $\{X_1^d, \ldots, X_{S^d}^d\}_{d=1}^D$, one for each domain, and the function approximations $\{\hat{f}_1^d, \ldots, \hat{f}_N^d\}_{d=1}^D$:

1. Draw an infinite number of potential error rates, $\boldsymbol{\phi_l}$, for our function approximations. For each $\boldsymbol{\phi_l}$, for $j = 1, \ldots, N$, draw an error rate $[\phi_l]_j \sim \text{Beta}(\alpha_e, \beta_e)$.
2. For each domain $d = 1, \ldots, D$:
   (a) Draw $p^d \sim \text{Beta}(\alpha_p, \beta_p)$, representing the prior probability for the true underlying function output being equal to 1, over all possible inputs, for domain d.
   (b) For each data example, X_i^d where $i = 1, \ldots, S^d$, draw a label $\ell_i^d \sim \text{Bernoulli}(p^d)$. This is the the true label, $f^d(X_i^d)$.
   (c) Draw a cluster assignment, $z^d \sim \text{CRP}(\alpha)$.
   (d) For each function approximation, $\hat{f}_j^d$, define the er-

ror rate as $e_j^d = [\phi_{z^d}]_j$.

(e) For each data example, X_i^d, and function approximation, $\hat{f}_j^d$, draw an output label, $\hat{f}_{ij}^d$, according to the following distribution:

$$\hat{f}_{ij}^d = \begin{cases} \ell_i^d & \text{, with probability } 1 - e_j^d, \\ 1 - \ell_i^d & \text{, otherwise.} \end{cases} \tag{9}$$

This output label corresponds to $\hat{f}_j^d(X_i^d)$.

A graphical representation of the model, along with a compact definition, is shown in figure 2.

In order to perform inference for this model we also use Gibbs sampling. For sampling from the DP we use the approach described in (Neal, 2000). In order to get fast convergence, we first marginalize out of the conditional probabilities $\boldsymbol{\phi_l}$ and sample the rest of the variables sequentially for a few iterations (i.e., we perform *collapsed Gibbs sampling*), and then we start sampling the $\boldsymbol{\phi_l}$ along with the other random variables, using the original conditional probabilities. For brevity, the conditional probabilities for this model are included in the supplementary material of this paper. They are derived directly from the model definition.

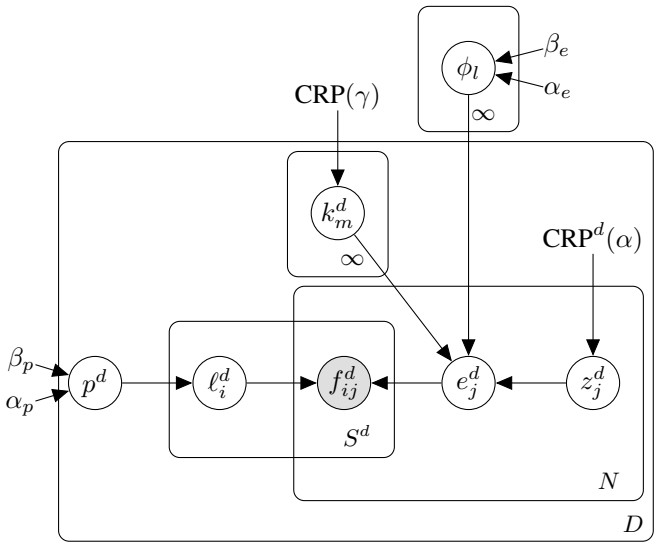
4.3. Hierarchical Coupled Bayesian Error Estimation

An important factor in estimating error rates using unlabeled data is the dependencies between our function approximations (see e.g., section 2). So far in our models, we share little information across those functions when estimating their error rates. One natural extension to our coupled error estimation model, which allows sharing more information across functions, is to use a hierarchical Dirichlet process (HDP) prior. This prior would allow us to first cluster the domain (i.e., as we are doing in the DP model), and then, for each domain cluster, to also cluster the classifiers, and to share the classifier clusters between different domain clusters. In the following two sections we provide an introduction to HDPs and we introduce our hierarchical coupled error estimation model.

Hierarchical Dirichlet Process (HDP): Hierarchical Dirichlet processes (HDPs) (Teh et al., 2006) extend the DP to be able to model grouped data. The HDP is a distribution over probability distributions G^m, $m = 1, \ldots, M$, each of which is conditionally distributed according to a DP. These distributions are coupled using a discrete common base measure, which is also distributed according to a DP. Each distribution G^m can be used to model a collection of observations $\{x_i^m\}_{i=1}^{N_m}$, as follows:

$$\begin{array}{cc} G \sim \text{DP}(\gamma, H)\,, & G^m \sim \text{DP}(\alpha, G)\,, \\ \theta_i^m \sim G^m\,, & x_i^m \sim f(\theta_i^m)\,. \end{array} \tag{10}$$

Each observation within a group is a draw from a mixture model, and mixture components can be shared between groups. The intuition behind this property of the HDP is that, due to the base measure of the child DPs being discrete, they necessarily share atoms. Thus, as desired, the mixture models in the different groups may share mixture components.



$$p^d \sim \text{Beta}(\alpha_p, \beta_p), \text{ for } d = 1, \ldots, D,$$
$$\ell_i^d \sim \text{Bernoulli}(p^d), \text{ for } i = 1, \ldots, S^d, \text{ and } d = 1, \ldots, D,$$
$$\phi_l \sim \text{Beta}(\alpha_e, \beta_e), \text{ for } l = 1, \ldots, \infty,$$
$$k_m^d \sim \text{CRP}(\gamma), \text{ for } d = 1, \ldots, D, \text{ and } m = 1, \ldots, \infty,$$
$$z_j^d \sim \text{CRP}^d(\alpha), \text{ for } d = 1, \ldots, D, \text{ and } j = 1, \ldots, N,$$
$$e_j^d = \phi_{k_{z_j^d}^d}, \text{ for } j = 1, \ldots, N, \text{ and } d = 1, \ldots, D,$$
$$\hat{f}_{ij}^d = \begin{cases} \ell_i^d & \text{, with probability } 1 - e_j^d, \\ 1 - \ell_i^d & \text{, otherwise.} \end{cases}$$

Figure 3: Graphical model for hierarchical coupled error rate estimation using only unlabeled data. The hierarchical coupling comes from the use of a hierarchical Dirichlet process prior to cluster problem domains and functions and share information within each cluster. Note that CRP$^d(\alpha)$ denotes a separate Chinese restaurant process (CRP) per domain d, with concentration parameter α.

Model: To extend our model for coupled error rate estimation to allow sharing of information across functions by using an HDP, as described at the beginning of section 4.3, we can define the following generative process for our data:

1. Draw an infinite number of potential error rates, $\phi_l \sim \text{Beta}(\alpha_e, \beta_e)$, for our function approximations.
2. For each domain $d = 1, \ldots, D$:
   (a) Draw $p^d \sim \text{Beta}(\alpha_p, \beta_p)$, as in the coupled error estimation model.
   (b) For each data example, X_i^d where $i = 1, \ldots, S^d$, draw a label $\ell_i^d \sim \text{Bernoulli}(p^d)$, as in the coupled error estimation model.
   (c) Draw an infinite number of potential cluster as-

signments for each function approximation, $k_m^d \sim \text{CRP}(\gamma)$.

(d) For each function approximation, $j = 1, \ldots, N$:

i. Draw a cluster assignment, $z_j^d \sim \text{CRP}^d(\alpha)$, from the CRP corresponding to the current domain.

ii. Define the error rate as $e_j^d = \phi_{t_j^d}$, $t_j^d = k_{z_j^d}^d$.

iii. For each data example, X_i^d, draw an output label, $\hat{f}_{ij}^d$, according to the following distribution:

$$\hat{f}_{ij}^d = \begin{cases} \ell_i^d & \text{, with probability } 1 - e_j^d, \\ 1 - \ell_i^d & \text{, otherwise.} \end{cases} \quad (11)$$

This output label corresponds to $\hat{f}_j^d(X_i^d)$.

A graphical representation of the model, along with a compact definition, is shown in figure 3.

In order to perform inference for this model we also use Gibbs sampling. For sampling from the HDP we use the approach described in (Teh et al., 2006). We also use collapsed Gibbs sampling for the initial sampling phase, as we did for the coupled error estimation model. For brevity, the conditional probabilities for this model are included in the supplementary material of this paper. They are derived directly from the model definition.

5. Experiments

We first carried out the experiments of (Platanios et al., 2014), so that we can compare the accuracy of our methods in estimating error rates against theirs. In order to explore the ability of the proposed methods to estimate error rates in realistic settings without domain-specific tuning, two very different data sets were used in those experiments. In the next two paragraphs we describe the two data sets we used, and in the sections that follow we describe the experiments we carried out and the results we obtained (the code for our methods and experiments, and the data we used can be found at `http://platanios.org/code/`).

NELL Data Set: This data set consists of data samples where four binary logistic regression (LR) classifiers – each one using a different set of features – were used to predict whether a NP belongs to a specific category in the NELL ontology (e.g., is Monongahela a river?). The four classifiers used were the following: (1) **ADJ**: Uses as features the adjectives that co-occur with the NP over millions of web pages, (2) **CMC**: Considers orthographic features of the NP (e.g., does the NP end with the letter string "burgh"? – more details can be found in (Carlson et al., 2010) and (Mitchell et al., 2015), (3) **CPL**: Uses as features words and phrases that appear with the NP, and (4) **VERB**: Uses as features verbs that appear with the NP. The domain in this case is defined by the category (e.g., "beverage" and "river" are two different domains) and we have about $\sim 20,000$ labeled examples available per category.

Brain Data Set: Functional Magnetic Resonance Imaging (fMRI) data were collected while 8 subjects read a chapter from a popular novel (Rowling, 2012), one word at a time. This data set consists of data samples where 11 classifiers were used to predict which of two 40 second long story passages correspond to an unlabeled 40 second time series of fMRI neural activity. Each classifier is making its prediction based on a different representation of the text passage (e.g., the number of letters in each word of the text passage, versus the part of speech of each word, versus emotions experienced by characters in the story, etc.). The domain in this case is defined by 11 different locations in the brain, for each one of which we have 924 labeled examples. Additional details can be found in (Wehbe et al., 2014).

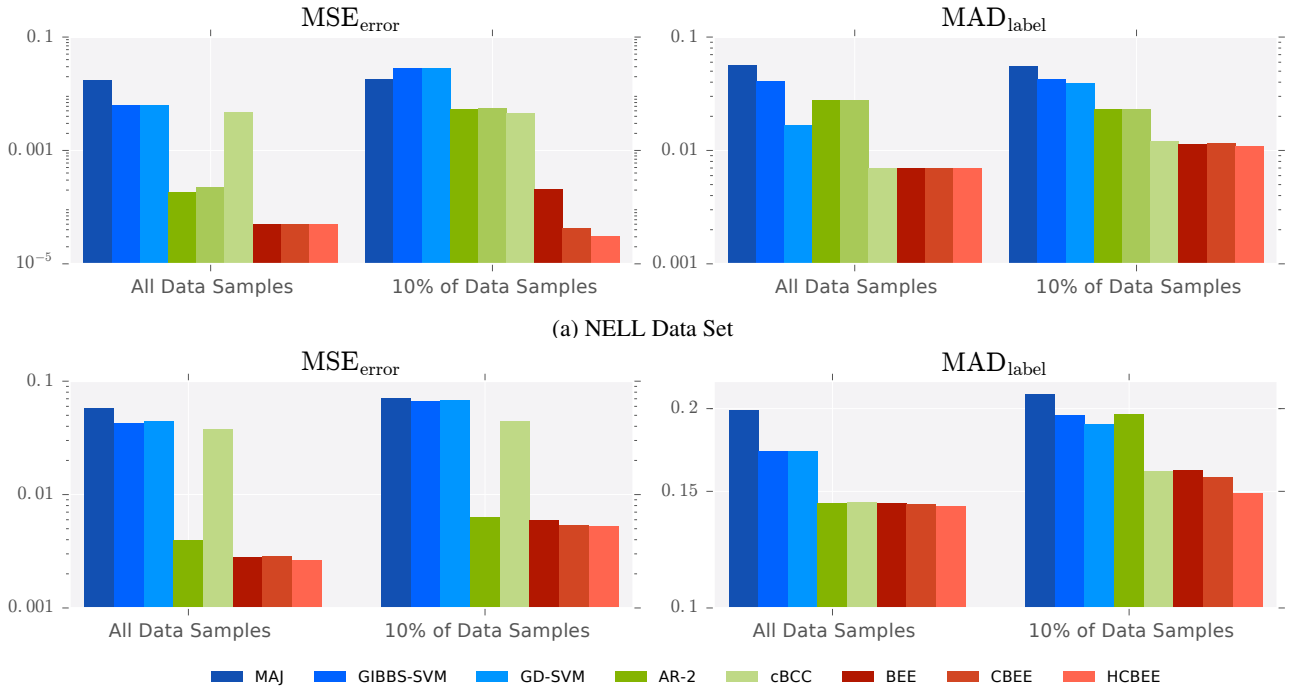
Experiments Description: We run two experiments for each data set: one for evaluating the accuracy of the proposed methods in estimating classifier error rates, and one for evaluating the accuracy of the labels inferred by our methods. We describe each one of these two experiments in the following sections.

Error Rates Experiment: For this experiment, we use the data without their labels to estimate error rates and, at the same time, we use the labels of the data in order to compute an estimate of the true error rate. The way we compute the estimate of the true error rate is by simply computing the sample error rate over the labeled data (i.e., the ratio of wrong labels to total number of samples, which can be computed because the true labels are known). From now on we shall refer to this estimate of the true error rates as the "true error rates". The evaluation metric we use to report our results is the mean squared error (MSE) of the error rate estimates from the true error rates. A low MSE indicates that our method is performing well.

Labels Experiment: With this experiment we want to evaluate how accurate the inferred labels are, and compare our accuracy to that of other methods that one can use. The evaluation metric we use to report our results is the mean absolute deviation (MAD) of the label estimates from the true labels that are known, which is simply the accuracy of the labels (i.e., the ratio of correct labels to total number of labels).

Baselines: We note that for some of the methods we compare against that do not explicitly estimate error rates, but rather combine the classifier outputs to produce a single label, we produce an estimate of the error rate using these labels and compare against that estimate. The methods that we compare against are the following:

1. Majority Vote (MAJ): The most intuitive method to use. It consists of simply taking as the combined label the most common label among the classifier outputs.
2. DW: Method of Dawid & Skene (1979). Results for this method have been omitted from our figures because they are several orders of magnitude worse than those that have been included.



(a) NELL Data Set

(b) Brain Data Set

Figure 4: Mean squared error (MSE) of the error rate and mean absolute deviation (MAD) of the label estimates for our methods (plotted in red color) and the methods that we are competing against (plotted in blue and green color). The lower the MSE (i.e., the shorter the bar), the better the result is. It clear from these plots that our methods outperform the competing methods in all cases. Note that there are no results for the AR method because the optimization solver used failed when solving a problem with that many constraints (i.e., size of power set of 11 classifiers. Error bars have been omitted from these plots because they were negligibly small (~ 2 orders of magnitude smaller than the reported values).

3. GIBBS-SVM and GD-SVM: Methods of Tian & Zhu (2015).
4. Agreement Rates (AR): Method of Platanios et al. (2014). It estimates error rates but does not infer the combined label. For that reason, we use a weighted majority vote, where the classifiers predictions are weighted according to their error rates, in order to produce a single output label.
5. cBCC: Method of Moreno et al. (2015) adapted to model error rates instead of the full confusion matrix.

Setup: For all our experiments and all three of our models, the Gibbs sampling inference procedure we used consisted of the following steps: (i) we sample $4,000$ samples that we throw away (i.e., burn-in samples), (ii) we sample $2,000$ samples and keep every 10^{th} sample in order to reduce the dependencies between our samples introduced by the sequential nature of the procedure, and (iii) we obtain our error rate and label estimates by averaging over the samples that we kept. We repeated each experiment 10 times and we report the mean of the evaluation metrics. We also computed the standard deviation of those metrics but we decided to omit it from our figures because it was ~ 2 orders of magnitude smaller than the actual mean. Regarding the hyperparameters of our models, we set them as:

- Labels Prior: α_p and β_p are both set to 1, and so the prior is uniform and uninformative.
- Error Rates Prior: α_e is set to 1 and β_e is set to 10. We selected those values in order to "avoid" the identifiability problem related to the error rates that (Platanios et al., 2014) describe. This prior encodes our assumption that more than half of our classifiers have error rate lower than 0.5.
- DP and HDP Concentration Parameters: We carried out several experiments with many logarithmically spaced values for α and γ (all combinations of pairs of values were considered for the HDP) and we computed the log-likelihood for a held-out data set[1], for each such experiment. The results that we report for the DP and HDP models are those corresponding to the experiment that resulted in the highest log-likelihood value for the held-out data set.

The results we obtained from all of our experiments are summarized in figure 4 and are discussed in the following sections. In what that follows, we use the following abbreviations for our models: **BEE** is used to refer to our error estimation model of section 4.1, **CBEE** is used to refer to

[1]The held-out data set consisted of 10% of the total amount of data we had available, which was randomly sampled.

our coupled error estimation model of section 4.2, and finally, **HCBEE** is used to refer to our hierarchical coupled error estimation model of section 4.3.

Results: As is easily evident from figures 4a and 4b, *HCBEE always outperforms the competing methods*. One thing that we expect to see in our results, in the presence of dependencies across domains and classifiers, is that CBEE and HCBEE are better than BEE when we have a small amount of data, because that is when sharing information becomes useful. When we have a large amount of data, we expect that the performance of the simple BEE model will be similar to its extensions, since for CBEE and HCEE, the atoms may not be clustered because there may be enough data per atom so that no sharing of information is necessary. That is evident in the results that we obtained.

For all experiments, where we use the all data samples we see that our three proposed methods perform equivalently well and always beat the competing methods. The fact that the coupling introduced by the nonparametric extensions to the simple model does not offer an improvement in performance can be attributed to the fact that for all those experiments we have enough data for each error rate to be modeled separately and not be clustered. We note that the plots in figures 4a and 4b are using a logarithmic scale, meaning that our methods offer significant improvement over the current state-of-the-art. The results for our three models are not exactly identical most probably due to the fact that they use different priors that allow different levels of information sharing. In the case of limited data samples (i.e., 10% of the data samples in our data sets), the HCBEE method performs the best, followed by CBEE. This supports our argument, that our coupled error estimation methods are more powerful in cases where a limited amount of data is available. Despite the fact that this is not really the case with the data for NELL, or other web-scale projects, it is a common scenario that is encountered with other types of data, such as neuroscience and biology data, for example. In such cases even unlabeled data can be really hard and expensive to obtain.

We note that cBCC is the closest method to our models and also beats alternative methods in most cases. However, our methods always outperform it. That is probably due to the fact that this model only allows sharing of information among different classifiers (i.e., clusters the classifiers), but none of our data sets involves a high number of classifiers.

6. Extensions and Future Work

We first note that, even though we do not consider this setting in our paper, the core idea behind our proposed methods can easily be extended to model the full confusion matrix for general discrete labels (i.e., instead of binary labels that we consider in this paper). This can be useful in several applications, when one needs to know how the error rate decomposes into precision and recall, for example.

Interesting directions for future work include taking into account known constraints between the domains. For example, in the case of NELL, we have domains such as "animal" and "person". We might already know that certain domains are mutually exclusive, for example. That would be useful because in this case we would know that if two classifiers predict that the label for some input is positive for both domains, then we know that at least one of them has to be making an error. It would thus be interesting to extend our models to use information provided by logical constraints, such as mutual exclusion and subsumption. In the case of NELL, we could use the estimated error rates in combination with a framework such as probabilistic soft logic (PSL) (Bröcheler et al., 2010; Pujara et al., 2013) in order to improve the accuracy of the system.

There are also certain other potential future directions for this work. It would be interesting to explore generalizations of our models to non-boolean, discrete-valued functions, or even to real-valued functions. Furthermore, we would like to explore ways in which we can use the error rate estimates in order to improve the performance of our function approximations. As already mentioned in some of the related work, that would constitute a first step towards developing a self-reflection framework for autonomous learning systems. In this context, we could try using our estimates in order to develop an effective active learning method, using the ideas discussed in the last paragraph of section 3.

7. Conclusion

We have introduced a Bayesian approach to estimate the error rate of each of several approximations to the same function, using only unlabeled data. Our approach also allows inferring the posterior distribution of the true label (i.e., the true underlying function output), by combining the outputs of those function approximations while accounting for their error rates. We first proposed a simple generative model for error estimation. This model is implicitly using the functions' agreement rates over unlabeled data. We then considered the setting where we might have multiple target functions, or domains, and a common set of learning algorithms used for learning each of these. We provided an extension to our simple model that allows grouping such domains and sharing information within them. Finally, considering the fact that the dependencies between function approximations are an important factor in estimating error rates using unlabeled data, we proposed a second extension to our model, that further clusters the function approximations and allows sharing of these clusters across different domain groups. In order to explore the ability of the proposed methods to estimate error rates in realistic settings without domain-specific tuning, we used two very different data sets in our experiments. Our methods were shown to outperform the current state-of-the-art, in both the tasks of estimating error rates and inferring the most likely single label, using only unlabeled data.

Acknowledgements

We thank Leila Wehbe for providing us with the brain data set and Alan Ritter and Siddharth Varia for providing us with the NELL data set. Finally, we thank the previously mentioned people, Abulhair Saparov, and the anonymous reviewers for their helpful comments. This research has been supported in part by NSF grant IIS-1250956 and in part by DARPA under contract number FA8750-13-2-0005.

References

Antoniak, C. E. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

Balcan, Maria-Florina, Blum, Avrim, and Mansour, Yishay. Exploiting Ontology Structures and Unlabeled Data for Learning. *International Conference on Machine Learning*, pp. 1112–1120, 2013.

Bengio, Yoshua and Chapados, Nicolas. Extensions to Metric-Based Model Selection. *Journal of Machine Learning Research*, 3:1209–1227, March 2003.

Blackwell, David and MacQueen, James B. Ferguson Distributions Via Polya Urn Schemes. *The Annals of Statistics*, 1(2):353–355, March 1973.

Bröcheler, Matthias, Mihalkova, Lilyana, and Getoor, Lise. Probabilistic Similarity Logic. In *Conference on Uncertainty in Artificial Intelligence*, pp. 73–82, 2010.

Carlson, Andrew, Settles, Burr, Betteridge, Justin, Kisiel, Bryan, Hruschka Jr, Estevam R, and Mitchell, Tom M. Toward an Architecture for Never-Ending Language Learning. In *Conference on Artificial Intelligence (AAAI)*, pp. 1–8, 2010.

Collins, John and Huynh, Minh. Estimation of Diagnostic Test Accuracy Without Full Verification: A Review of Latent Class Methods. *Statistics in Medicine*, 33(24): 4141–4169, June 2014.

Collins, Michael and Singer, Yoram. Unsupervised Models for Named Entity Classification. In *Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 1–11, 1999.

Dasgupta, Sanjoy, Littman, Michael L, and McAllester, David. PAC Generalization Bounds for Co-training. In *Neural Information Processing Systems*, pp. 375–382, 2001.

Dawid, A P and Skene, A M. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, January 1979.

Ferguson, Thomas S. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230, March 1973.

Geman, Stuart and Geman, Donald. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984. ISSN 0162-8828.

Madani, Omid, Pennock, David M, and Flake, Gary W. Co-Validation: Using Model Disagreement on Unlabeled Data to Validate Classification Algorithms. In *Neural Information Processing Systems*, pp. 1–8, 2004.

Mitchell, Tom M, Cohen, William W, Hruschka Jr, Estevam R, Pratim Talukdar, Partha, Betteridge, Justin, Carlson, Andrew, Dalvi, Bhanava, Gardner, Matt, Kisiel, Bryan, Krishnamurthy, Jayant, Lao, Ni, Mazaitis, Kathryn, Mohamed, Thahir P, Nakashole, Ndapakula, Platanios, Emmanouil Antonios, Ritter, Alan, Samadi, Mehdi, Settles, Burr, Wang, Richard C, Wijaya, Derry, Gupta, Abhinav, Chen, Xinlei, Saparov, Abulhair, Greaves, Malcolm, and Welling, Joel. Never-Ending Learning. In *Association for the Advancement of Artificial Intelligence*, pp. 1–9, 2015.

Moreno, Pablo G, Artés-Rodríguez, Antonio, Teh, Yee Whye, and Perez-Cruz, Fernando. Bayesian Nonparametric Crowdsourcing. *Journal of Machine Learning Research*, 16:1–21, August 2015.

Neal, Radford M. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, June 2000.

Parisi, Fabio, Strino, Francesco, Nadler, Boaz, and Kluger, Yuval. Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences*, pp. 1–28, January 2014.

Platanios, Emmanouil Antonios, Blum, Avrim, and Mitchell, Tom M. Estimating Accuracy from Unlabeled Data. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1–10, 2014.

Pujara, Jay, Miao, Hui, Getoor, Lise, and Cohen, William W. Knowledge Graph Identification. *International Semantic Web Conference*, 8218(Chapter 34): 542–557, 2013.

Rowling, J.K. *Harry Potter and the Sorcerer's Stone*. Harry Potter US. Pottermore Limited, 2012. ISBN 9781781100271.

Schuurmans, Dale, Southey, Finnegan, Wilkinson, Dana, and Guo, Yuhong. Metric-Based Approaches for Semi-Supervised Regression and Classification. In *Semi-Supervised Learning*, pp. 1–31. 2006.

Teh, Yee Whye, Jordan, Michael I, Beal, Matthew J, and Blei, David M. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, December 2006.

Tian, Tian and Zhu, Jun. Max-Margin Majority Voting for Learning from Crowds. In *Neural Information Processing Systems*, pp. 1–9, 2015.

Wehbe, Leila, Murphy, Brian, Talukdar, Partha, Fyshe, Alona, Ramdas, Aaditya, and Mitchell, Tom. Predicting brain activity during story processing. *in review*, 2014.