# Estimating Accuracy from Unlabeled Data

**Emmanouil Antonios Platanios** E.A.PLATANIOS@CS.CMU.EDU
*Machine Learning Department*
*Carnegie Mellon University*
*Pittsburgh, PA 15213, USA*

**Advisor:** Tom Mitchell

## Abstract

We consider the question of how unlabeled data can be used to estimate the true accuracy of learned classifiers. This is an important question for any autonomous learning system that must estimate its accuracy without supervision, and also when classifiers trained from one data distribution must be applied to a new distribution (e.g., document classifiers trained on one text corpus are to be applied to a second corpus). We first show how to estimate error rates exactly from unlabeled data when given a collection of competing classifiers that make independent errors, based on the agreement rates between subsets of these classifiers. We further show that *even when the competing classifiers do not make independent errors, both their accuracies and error dependencies can be estimated* by making certain relaxed assumptions. We then present an alternative approach based on graphical models that also allows us to combine the outputs of the classifiers into a single output label. A simple graphical model is introduced that performs well in practice. Then, two nonparametric extensions to it are presented, that significantly 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. We also obtain results demonstrating our graphical model approaches beating alternative methods for combining the classifiers' outputs. These results are of practical significance in situations where labeled data is scarce and shed light on the more general question of how the consistency among multiple functions is related to their true accuracies.

**Keywords:** unsupervised learning, semi-supervised learning, accuracy estimation

## 1. Introduction

Estimating accuracy of classifiers is central to machine learning and many other fields. Most existing approaches to estimating accuracy are *supervised*, meaning that a set of labeled examples is 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. Furthermore, it is also useful 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 is used to make predictions for (this is especially true in settings where training data is scarce).

We first show that accuracy can be estimated exactly from unlabeled data in the case where at least three different approximations to the same function are available, so long as these functions make independent errors and the majority of them have better than chance accuracy. More interestingly, we show that even if one does not assume independent errors, one can still estimate accuracy given a sufficient number of competing approximations to the same function, by viewing the degree of independence of those approximations as an optimization criterion. We call this first "family" of methods, the "agreement rates" approaches. We then propose a second "family" of methods based on probabilistic graphical models. Moreover, the latter approaches are also capable of inferring the posterior distribution of a single label for each data sample jointly with the accuracies of our classifiers. Thus, they further allow us to combine the classifier outputs into a single label. Moreover, they are capable of handling missing data (in contrast to the agreement rates methods). 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). We propose a simple graphical model, along with two nonparametric extensions to it that allow for more sharing of information, which would help in the case when we have a limited amount of data. We present experimental results demonstrating the success of our approaches in estimating classification accuracies to within a few percentage points of their true values, in two diverse domains. Furthermore, we also present results showing that our graphical model-based methods outperform existing methods for combining classifier outputs into a single label.

We consider a "multiple approximations" problem setting in which we have several different approximations, $\hat{f}_1, \dots, \hat{f}_N$, 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 $f$. The multiple functions can be from any source – learned or manually constructed. 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). Among other things, NELL learns 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). Our aim is to find a way to estimate the error rates of each of the competing approximations to $f$, using only unlabeled data (e.g., many unlabeled NPs in the case of NELL) and to infer the posterior distribution of the response of function $f$ while accounting for those error rates.

## 2. Related Work

Other researchers have considered variants of this "multiple approximations" setting. For example, Blum and Mitchell (1998) introduced the co-training algorithm which uses unlabeled data to train competing approximations to a target function by forcing them to agree on classifications of unlabeled examples. Schuurmans et al. (2006) and Bengio and Chapados (2003) used the disagreement rate between the approximations as a distance metric to perform model selection and regularization. Parisi et al. (2014) proposed a spectral method to rank classifiers based on accuracy and combine their outputs to produce one final label, also under an assumption of independence of the input features given the labeling. Balcan et al. (2013) used disagreement rates combined with an ontology to estimate the error of the prediction vector, in a multi-class setting, from unlabeled data, under the assumption that the input features are independent given the output label.

There has also been work at developing more robust semi-supervised learning algorithms by using agreement rates between classifiers (Collins and Singer, 1999). Chang et al. (2007) used some task-specific constraints to decide which additional examples should be added to the training data set. However, very few have tried to directly estimate per-function error rates using agreement rates, and to use those error rates in combining the outputs of those functions into a single label. Dasgupta et al. (2001) PAC-bound the error rates of the functions using their pairwise agreement rates, under the assumption that the functions make independent errors, and Madani et al. (2004) estimate the average error rate of two predictors using their disagreement rates. Donmez et al. (2010) is one of the few to estimate per-function error rates from unlabeled data. Here, the authors estimate the prediction risk for each function under the assumption that the true probability distribution of the output labels is known. The focus of their work is to use the known label distribution to estimate the error rate even of a single classifier, while taking into account the agreement rates as well, especially under the assumption of conditional independence.

Finally, Collins and 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 of the other papers already mentioned, but the idea of applying the tests to multiple populations is a new and interesting one. Collins and Huynh (2014) also review many methods that relax these assumptions in different ways and they also briefly discuss some Bayesian models for doing so.

## 3. Main Idea and Motivation

It can be observed from the previous section that most of the related work on estimating classifier accuracies with only unlabeled data is trying to relate agreement rates between different classifiers (which can be observed) with the accuracies of those classifiers. In fact, this is also the approach we take 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 are now going to present an example that will provide some intuition behind why one might want to use agreement rates as indicators of correctness, and what issues might arise if one does that.

Let us consider a case where a person asks 10 different people a question that is related to politics and 8 of those people agree on an answer. One might immediately think that, since we have such a strong majority, the answer must be the correct one. However, one has to be careful. Let us assume that those 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 that case, we might want to reconsider whether that answer is correct and to what extend 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 maybe we should trust that answer even more. We therefore see that *the answer to the question of whether consistency implies correctness, may have to do with how dependent the functions that agree with each other are* . One trivial example to think about that reinforces this argument is when our multiple functions are in fact copies of the same function and thus fully dependent. In that case consistency among those functions gives us no information about their correctness (i.e., they are always consistent with each other in their responses).

One last thing to note about the politics-related example of the previous paragraph is that it raises a new question, which is: *if functions that are highly dependent disagree, then what does that imply about the question being asked, or about those functions themselves?* In the case of asking people questions, such as the provided example, it might imply that the question asked was subjective. We can extend that interpretation to classifiers by saying that maybe the given classification problem is too hard, or maybe the functions are too uncertain about their answers. This is an interesting question to explore, but it is outside the scope of the current paper.

## 4. Proposed Methods

### 4.1 Agreement Rates Approach

In this section, we introduce a method used to estimate the error rates of binary functions in the multiple approximations setting described in section 1. It is based on the idea of looking at the consistency between the different functions' predictions in order to determine the error rates of those functions. The method consists of matching the sample agreement rates of the functions with the exact formulas of those agreement rates written in terms of the functions' error rates. It estimates the individual error rates for each function, as well as the joint error rates of all possible subsets of those functions, based on the predictions made by these functions over a sample of unlabeled instances $X_1, \dots, X_S$.

Henceforth, we denote the input data by $X$ and the true binary output label by $Y$. We assume the input data $X$ are drawn from some unknown distribution $P(X) = \mathcal{D}$, and $Y \in \{0, 1\}$. Let us consider $N$ functions, $\hat{f}_1(X), \dots, \hat{f}_N(X)$ which attempt to model the mapping from $X$ to $Y$. For example, each function might be the result of a different learning algorithm, or might use a different subset of the features of $X$ as input. We define the error event $E_{\mathcal{A}}$ of a set of functions $\mathcal{A}$ as an event in which every function in $\mathcal{A}$ makes an incorrect prediction:

$$E_{\mathcal{A}} = \bigcap_{i \in \mathcal{A}} \left[\hat{f}_i(X) \neq Y\right], \tag{1}$$

where $\cap$ denotes the set intersection operator and where $\mathcal{A}$ contains the indices of the functions. We define the error rate of a set of functions $\mathcal{A}$ (i.e. the probability that all functions in $\mathcal{A}$ make an error together) as:

$$e_{\mathcal{A}} = \mathbb{P}_{\mathcal{D}}\left(E_{\mathcal{A}}\right), \tag{2}$$

where $\mathbb{P}_{\mathcal{D}}\left(\cdot\right)$ denotes the probability of an event under the distribution over input data $X$.

Let us define the agreement rate $a_{\mathcal{A}}$, for a set of functions $\mathcal{A}$ as the probability that all of the functions' outputs[1] are the same:

$$a_{\mathcal{A}} = \mathbb{P}_{\mathcal{D}}\left(\left\{\hat{f}_i\left(X\right) = \hat{f}_j\left(X\right), \forall i, j \in \mathcal{A} : i \neq j\right\}\right). \tag{3}$$

This quantity can be defined in terms of the error rates of the functions in $\mathcal{A}$. In order to understand how we can write the agreement rate in terms of error rates let us consider a simple example where $\mathcal{A} = \{i, j\}$ (i.e. consider just the pairwise agreement rate between the functions $f_i$ and $f_j$.). The probability of two functions agreeing is equal to the probability that both make an error, plus the probability that neither makes an error:

$$a_{\{i,j\}} = \mathbb{P}_{\mathcal{D}}\left(E_{\{i\}} \cap E_{\{j\}}\right) + \mathbb{P}_{\mathcal{D}}\left(\bar{E}_{\{i\}} \cap \bar{E}_{\{j\}}\right), \tag{4}$$

where $\bar{\cdot}$ denotes the complement of a set. By using De Morgan's laws and the inclusion-exclusion principle we obtain, using the notation defined in equation (2), an expression for the agreement rate between the two functions, in terms of their individual error rates, and their joint error rate:

$$a_{\{i,j\}} = 1 - e_{\{i\}} - e_{\{j\}} + 2e_{\{i,j\}}. \tag{5}$$

In the same way we obtain the following general result for the agreement rate of a set of functions $\mathcal{A}$ of arbitrary size:

$$\begin{aligned} a_{\mathcal{A}} &= \mathbb{P}_{\mathcal{D}}\bigg(\bigcap_{i\in\mathcal{A}} E_i\bigg) + \mathbb{P}_{\mathcal{D}}\bigg(\bigcap_{i\in\mathcal{A}} \bar{E}_i\bigg), \\ &= e_{\mathcal{A}} + 1 - \mathbb{P}_{\mathcal{D}}\bigg(\bigcup_{i\in\mathcal{A}} E_i\bigg), \\ &= e_{\mathcal{A}} + 1 + \sum_{k=1}^{|\mathcal{A}|}\Bigg[(-1)^k \sum_{\substack{I\subseteq\mathcal{A} \\ |I|=k}} e_I\Bigg], \end{aligned} \tag{6}$$

where $\cup$ denotes the set union operator and $|\cdot|$ denotes the number of elements in a set. For the first line we used the fact that the two events, $\left\{\bigcap_{i\in\mathcal{A}} E_i\right\}$ and $\left\{\bigcap_{i\in\mathcal{A}} \bar{E}_i\right\}$, are mutually exclusive, for the second line we used one of De Morgan's laws, and for the last line we used the inclusion-exclusion principle.

In the next section we examine the most basic case, assuming that functions make independent errors and that most of them have error rates below 0.5, showing that we can solve exactly for the error rates provided that we have at least 3 different functions. In the subsequent section we examine the most general case, assuming that we have $N$ functions that make errors with unknown inter-dependencies, and show that we can formulate this as

[1] Here, "outputs" is equivalent to "predictions".

**KEY IDEA**

The significance of equations (5) and (6) is that they relate the different agreement rates $a_{\mathcal{A}}$, which are easily estimated from *unlabeled* data, to the true error rates $e_{\mathcal{A}}$ of the functions, which are difficult to estimate without labeled data. Note that if we have a system of such equations with rank equal to the number of error rates mentioned, then we can solve exactly for these error rates in terms of the observed agreement rates. This is not the case in general, because given a set of functions, $\hat{f}_1, \dots, \hat{f}_N$, we obtain $2^N - N - 1$ agreement rate equations (one for each subset of two or more functions) expressed in terms of $2^N - 1$ error rates (one for each none-empty subset of functions). However, if we assume that the errors made by the $N$ individual functions are independent, then we can express all of the $2^N - 1$ error rates in terms of $N$ single-function error rates (e.g., $e_{\{i,j\}} = e_{\{i\}} e_{\{j\}}$) and we can then solve exactly for all error rates (given the additional assumption that error rates are better than chance). Furthermore, if we are unwilling to make the strong assumption that errors of individual functions are independent, then we can instead solve for the set of error rates that minimize the dependence among errors (e.g., among the infinite solutions to the underdetermined set of equations, we choose the solution that minimizes $\sum_{i,j} (e_{\{i,j\}} - e_{\{i\}} e_{\{j\}})^2$ – this idea can be easily extended to larger subsets than simply pairs of functions). The key idea in this paper is that the correspondence between easily-observed agreement rates and hard-to-observe error rates given by these equations can be used as a practical basis for estimating true error rates from unlabeled data.

a constrained numerical optimization problem whose objective function reflects a soft prior assumption regarding the error dependencies. Experimental results presented in a later section demonstrate the practical utility of this approach, producing estimated error rates that are within a few percentage points of the true error rates, *using only unlabeled data.*

#### 4.1.1 3 Functions That Make Independent Errors

When we have 3 functions that make independent errors we can replace the $e_{\{i,j\}}$ term in equation (5) with the term $e_{\{i\}} e_{\{j\}}$. In this case we have only 3 unknown variables (i.e. the individual function error rates) and we have $\binom{3}{2} = 3$ equations (i.e. equation (5), for $1 \leq i < j \leq 3$). Therefore, we can directly solve for each error rate in terms of the three observed agreement rates:

$$e_{\{i\}} = \frac{c \pm \left(1 - 2a_{\{j,k\}}\right)}{\pm 2 \left(1 - 2a_{\{j,k\}}\right)}, \tag{7}$$

where $i \in \{1, 2, 3\}$, $j, k \in \{1, 2, 3\} \setminus i$ with $j < k$ and:

$$c = \sqrt{\left(2a_{\{1,2\}} - 1\right) \left(2a_{\{1,3\}} - 1\right) \left(2a_{\{2,3\}} - 1\right)}, \tag{8}$$

where, for a set $B$ and an element of that set $b$, the notation $B \backslash b$ denotes the set containing all elements in $B$ except $b$. In practical applications, we can estimate the agreement rates among the competing functions, using a sample of unlabeled data $X_1, \dots, X_S$, as follows:

$$\hat{a}_{\{i,j\}} = \frac{1}{S} \sum_{s=1}^{S} \mathbb{I} \left\{ \hat{f}_i \left( X_s \right) = \hat{f}_j \left( X_s \right) \right\}, \tag{9}$$

where $\mathbb{I}\left\{\cdot\right\}$ evaluates to one if its argument statement is true and to zero otherwise.

In most practical applications the competing functions do not make independent errors. We next consider the more difficult problem of estimating the error rates from agreement rates, but without assuming independence of the function error events.

### 4.1.2 $N$ Functions That Make Dependent Errors

When we have $N$ functions that make dependent errors we rely on the agreement rate equation (6). We consider the agreement rates for all sets $\mathcal{A} = \left\{\mathcal{A} \subseteq \left\{1, \dots, N\right\} : \left|\mathcal{A}\right| \geq 2\right\}$ of functions (the agreement rate is uninformative for less than two functions) and we obtain $2^N - N - 1$ equations by matching equation (6) to the sample agreement rate for each possible subset of functions. Given a sample of unlabeled data $X_1, \dots, X_S$, the sample agreement rate is defined as:

$$\hat{a}_{\mathcal{A}} = \frac{1}{S} \sum_{s=1}^{S} \mathbb{I}\left\{\hat{f}_i(X_s) = \hat{f}_j(X_s), \forall i, j \in \mathcal{A} : i \neq j\right\}, \tag{10}$$

and is an unbiased estimate of the true agreement rate. Moreover, our unknown variables are all the individual function error rates along with all of the possible joint function error rates (let us denote the vector containing all those variables by $\boldsymbol{e}$); that is a total of $2^N - 1$ unknown variables.

The set of $2^N - N - 1$ equations involving $2^N - 1$ unknown variables yields an underdetermined system of equations with an infinite number of possible solutions. We therefore cast this problem as a constrained optimization problem where the agreement equations form constraints that must be satisfied and where we seek the solution that minimizes the following objective:

$$c(\boldsymbol{e}) = \sum_{\mathcal{A} : \left|\mathcal{A}\right| \geq 2} \left(e_{\mathcal{A}} - \prod_{i \in \mathcal{A}} e_i\right)^2. \tag{11}$$

It can be seen that we are basically trying to minimize the dependence between the error events[2], while satisfying all of the agreement rates constraints. We saw in section 4.1.1 that if we assume that the error events are independent, then we can obtain an exact solution. By defining our optimization problem in this way we are effectively relaxing this constraint by saying that we want to find the error rates that satisfy our constraints and that are, at the same time, as independent as possible. Note that this is a strict generalization of the previous formulation of our model (i.e., when assuming independent error rates), since if the error rates are indeed independent then this objective will obtain its minimum value of zero and the solution of the optimization problem will match the solution we obtained before. Most existing methods trying to estimate function error rates using only unlabeled data assume that the error events are independent; the main novelty of this method lies in the fact that *we relax all those assumptions* and make no hard or strict assumptions about our functions.

Note that we could also define different objective functions based on information we might have about our function approximations or based on different assumptions we might

[2] That can be seen from the fact that when the error events are independent we have that $e_{\mathcal{A}} = \prod_{i \in \mathcal{A}} e_i$.

want to make. For example, one could try minimizing the sum of the squares of all the error rates (i.e. the $L_2$ norm of $\boldsymbol{e}$) in order to obtain the most optimistic error rates that satisfy the agreement rates constraints. The novelty of our method partly lies in the formulation of the error rates estimation problem using only unlabeled data as a constrained optimization problem.

In this section we defined the model we are using for this method and the optimization problem we wish to solve. We call this method the AR method (i.e. Agreement Rates method). In the following section we define some additional constraints that this method uses, along with some details on the optimization problem that is being solved.

#### 4.1.3 Optimization

We use the COIN-OR IpOpt v.3.11.9 solver. In the following sections we discuss: (1) some additional constraints that apply to our method, (2) extensions of our approach to the case where multiple approximations are learned for each of several different target functions, and (3) an approximation that can make our methods much faster, more scalable and maybe even more accurate.

**Error Rates Constraints:** Our unknown variables include both individual function error rates and joint function error rates of those events. We need to impose constraints on the values that the joint function error rates can take. These constraints follow from basic rules of probability and set theory; they represent bounding joint event probabilities using the corresponding marginal event probabilities. These constraints are defined by the following equation:

$$e_{\mathcal{A}} \leq \min_{i \in \mathcal{A}} e_{\mathcal{A} \setminus i}, \tag{12}$$

for $|\mathcal{A}| \geq 2$. Furthermore, regarding the individual function error rates, it is easy to see that if we transform all $e_i$, for $i = 1, \dots, N$, to $1 - e_i$, the resulting agreement rates are equal to the original ones. A similar result holds for the likelihood function. In order to make our models identifiable we add the constraint that $e_i \in [0, 0.5)$, for most values of $i = 1, \dots, N$, which simply means that most of our functions/binary classifiers perform better than chance. It is thus a very reasonable constraint[3].

**Dealing With Multiple Classification Problems:** 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 one of those. For 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 "locations" or not, etc.). Multiple learning methods are utilized to approximate each one of those target functions (e.g., a classifier based on the NP orthography, 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.

Of course we can apply our AR method to estimate accuracies separately for each target classification problem (and that is what we actually did in our experiments described in

[3] It is important to understand here that in order for our methods to work in the first place, this constraint *must* hold for the classifiers that we are considering.

section 5). However, when we have multiple target functions to be learned and multiple leaning methods shared across each, there is an interesting opportunity to further couple the error estimates across these different target functions. In equation (11) we introduced terms to minimize the dependency between the error rates of competing approximations. In the case where we have multiple target functions, we might introduce additional terms to capture other relevant assumptions. For example, we could introduce a term to minimize the difference in error dependencies between two learning methods across multiple classification problems (e.g., we could choose to minimize the difference in error dependencies between orthography-based and context-based classifiers trained for different classification problems).

**Approximating High Order Error Rates:** Once the agreement rate estimates have been calculated, the execution time of the optimization procedure for the AR method does not depend on the number of provided data samples, $S$. It does however depend on the number of functions, $N$. This can be easily seen by considering the number of unknown variables we have which is equal to $2^N - 1$. As will be shown in section 5, the performance of all methods, in terms how good the obtained function error rate estimates are, increases with an increasing number of functions, $N$. It is therefore not a good idea to try to reduce $N$. So, we instead propose a way to reduce the execution time of the optimization procedure by approximating high order error rates, instead of estimating them directly.

We can estimate high order joint function error rates[4] using lower order function error rates by using the following formula, for $|\mathcal{A}| > M_e$, where $M_e$ is chosen arbitrarily:

$$e_{\mathcal{A}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} e_{\mathcal{A} \setminus i} e_i. \tag{13}$$

With a high value of $M_e$ we obtain better estimates but execution time is larger, and vice-versa. This estimate is based on the fact that the higher the order of the function error rates, the less significant the impact of an independence assumption between them will be.

Furthermore, the only available information regarding high order error rates comes from high order sample agreement rates[4], $\hat{a}_{\mathcal{A}}$, which will likely be very noisy estimates of the true agreement rates. That is because there will be very few data samples were all of the functions in $\mathcal{A}$ will agree and therefore the sample agreement rate will be computed using only a small number of data samples resulting in a noisy estimate of the true agreement rate. This motivates not directly estimating high order error rates, but instead approximating them using low order error rates. In fact, in the case that the sample agreement rates are too noisy, this approximation might even increase the quality of the obtained error rate estimates. By approximating high order error rates in the way described earlier, we are effectively ignoring the corresponding high order sample agreement rates (i.e. they are not used in our estimation) for the AR method.

### 4.2 Graphical Model Approaches

In this section we propose a few different approaches for estimating error rates of classifiers from unlabeled data, that as we will later see, share an interesting connection with the

[4] By "order" of an error rate, $e_{\mathcal{A}}$, or agreement rate, $a_{\mathcal{A}}$, we mean the number of functions in set $\mathcal{A}$, or simply $|\mathcal{A}|$.

agreement rates approach of the previous section. We first propose a simple and elegant probabilistic graphical model that, as we show in the experiments section, in most cases achieves better accuracy in error rates estimation than the agreement rates approaches and, at the same time, combines the outputs of all 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. We then present an extension of that model where the observations are grouped according to the classification problem. In this way, statistical information can be shared across examples within the same group, 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, play an important role in error rate estimation.

### 4.2.1 Error Estimation

Following from section 1, we consider a "multiple approximations" problem setting in which we have several different approximations, $\hat{f}_1, \dots, \hat{f}_N$, 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 $X_1, \dots, X_S$ and the function approximations $\hat{f}_1, \dots, \hat{f}_N$:

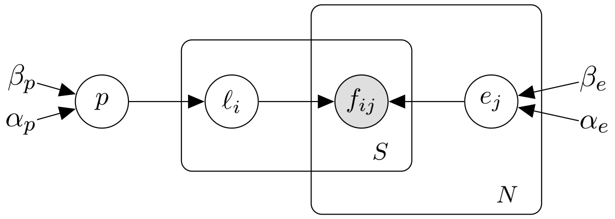
1. Let us make the assumption that there is an underlying distribution from which the labels for all the data examples are sampled. We first draw $p \sim \text{Beta}(\alpha_p, \beta_p)$, representing the prior probability for the true label being equal to 1, over all possible examples.
2. For each data example, $X_i$ where $i = 1, \dots, S$, we draw a label $\ell_i \sim \text{Bernoulli}(p)$. This label is the true, unobserved, 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 $j = 1, \dots, N$, 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:

$$\hat{f}_{ij} = \begin{cases} \ell_i & , \text{ with probability } 1 - e_j, \\ 1 - \ell_i & , \text{ otherwise.} \end{cases} \tag{14}$$

   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 and Geman, 1984), a well-known Markov Chain Monte Carlo (MCMC) sampling approach. The



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

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

conditional probabilities we use during sampling are as follows:

$$
\begin{aligned}
P(p \mid \cdot) &= \text{Beta}(\alpha_p + \sigma_{\boldsymbol{\ell}}, \beta_p + S - \sigma_{\boldsymbol{\ell}}), && (15) \\
P(\ell_i \mid \cdot) &\propto p^{\ell_i}(1 - p)^{1 - \ell_i}\pi_i, && (16) \\
P(e_j \mid \cdot) &= \text{Beta}(\alpha_e + \sigma_j, \beta_e + S - \sigma_j), && (17)
\end{aligned}
$$

where:

$$
\sigma_{\boldsymbol{\ell}} = \sum_{i=1}^{S} \ell_i, \quad \sigma_j = \sum_{i=1}^{S} \mathbb{1}_{\{\hat{f}_{ij} \neq \ell_i\}}, \tag{18}
$$

$$
\pi_i = \prod_{j=1}^{N} e_j^{\mathbb{1}_{\{\hat{f}_{ij} \neq \ell_i\}}} (1 - e_j)^{\mathbb{1}_{\{\hat{f}_{ij} = \ell_i\}}}, \tag{19}
$$

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:

$$
P(\hat{f}_{ij} \mid \cdot) \propto e_j^{\mathbb{1}_{\{\hat{f}_{ij} \neq \ell_i\}}} (1 - e_j)^{\mathbb{1}_{\{\hat{f}_{ij} = \ell_i\}}}. \tag{20}
$$

Note here that the AR method presented in section 4.1 is not able to handle missing data, and so this is an advantage of this graphical model approach to that method.

## IMPLICIT USE OF AGREEMENT RATES

Many of the papers from section 2, as well as the agreement rates approach presented in the previous section, propose using agreement rates between the different function approximations in order to estimate the error rates of those functions. By looking at equations 16, 17, 18, and 19 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 those functions.

### 4.2.2 Coupled Error Estimation

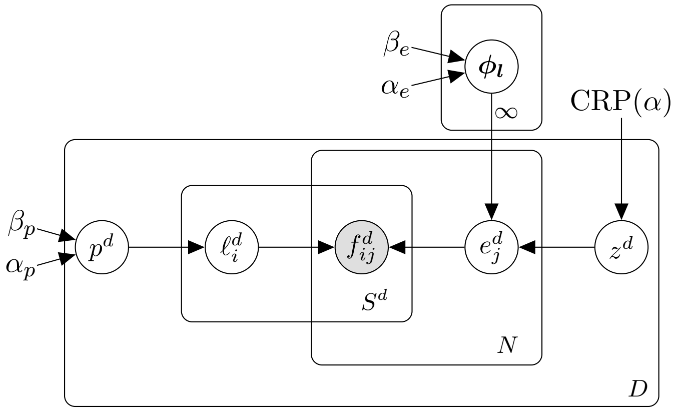
Up to this point we have assumed that there is a single target function and multiple approximations to that function. It was already mentioned in section 4.1.3 that in many cases we might have multiple target functions, or problem settings, and a common set of learning algorithms used for learning each one of those. It is reasonable to assume that there are some structural dependencies between our function approximations that could result in similar behavior across such problem settings (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. In the following two sections we provide an introduction to DPs and we 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:

$$G \sim \mathrm{DP}(\alpha, H)\,, \qquad \theta_i \sim G\,, \qquad x_i \sim f(\theta_i)\,. \tag{21}$$

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.

**Coupled Error Estimation Model:** In the definition of our model we are going to use the Chinese restaurant process (CRP) representation of the DP (Blackwell and MacQueen,



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

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($\alpha$) denotes the Chinese restaurant process (CRP) with concentration parameter $\alpha$.

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 4.2.2, we now have a problem setting in which we have several different 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 error 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:

$$\hat{f}_{ij}^d = \begin{cases} \ell_i^d & \text{, with probability } 1 - e_j^d, \\ 1 - \ell_i^d & \text{, otherwise.} \end{cases} \tag{22}$$

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, with 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.

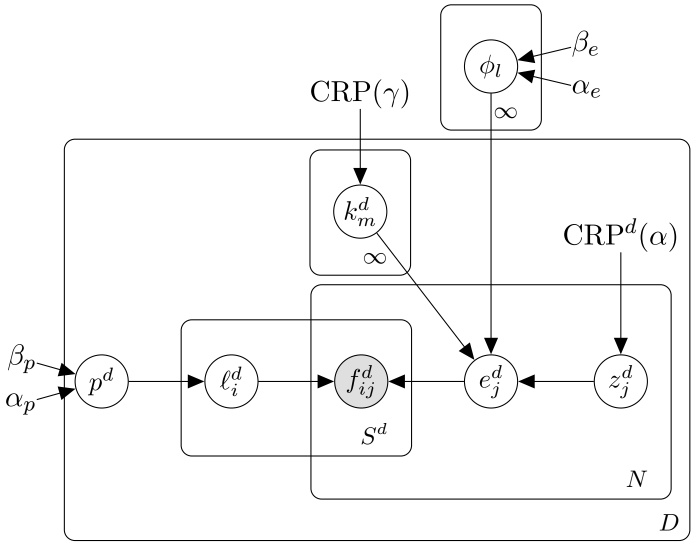
### 4.2.3 Hierarchical Coupled Error Estimation

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:

$$\begin{aligned} G &\sim \text{DP}(\gamma, H)\,, & G^m &\sim \text{DP}(\alpha, G)\,, \\ \theta_i^m &\sim G^m\,, & x_i^m &\sim f(\theta_i^m)\,. \end{aligned} \tag{23}$$

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



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

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 $\mathrm{CRP}^d(\alpha)$ denotes a separate Chinese restaurant process (CRP) per domain $d$, with concentration parameter $\alpha$.

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.

**Hierarchical Coupled Error Estimation 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 4.2.3, we can define the following generative process for our data:

1. Draw an infinite number of potential error rates, $\phi_l \sim \mathrm{Beta}(\alpha_e, \beta_e)$, for our function approximations.
2. For each domain $d = 1, \dots, D$:

(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 $i = 1, \dots, S^d$, 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 assignments for each function approximation, $k_m^d \sim \text{CRP}(\gamma)$.

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

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:

$$\hat{f}_{ij}^d = \begin{cases} \ell_i^d & \text{, with probability } 1 - e_j^d, \\ 1 - \ell_i^d & \text{, otherwise.} \end{cases} \tag{24}$$

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 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 carried out the experiments of (Platanios et al., 2014), for both the methods presented in that paper and for the graphical model approaches presented in the previous section. 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 the experiments. In the next two paragraphs, we describe the two data sets, and in the sections that follow, we describe the experiments that we carried out and the results obtained.

**NELL Data Set:** This data set consists of data samples where we use four binary logistic regression (LR) classifiers, each one using a different set of features, 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**: A LR classifier that uses as features the adjectives that occur with the NP over millions of web pages, (2) **CMC**: A LR classifier that 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**: A LR classifier that uses words and phrases that appear with the NP as features, and (4) **VERB**: A LR classifier that uses as features verbs that appear with the NP. Note the NP features used by these four classifiers are somewhat independent given the correct classification label. The domain in this case is defined by the category (e.g., beverage and river are two different domains) and table 5.1 lists the NELL categories that we used in our experiments, along with the number of labeled examples available per category.

| Category | # Examples | Category | # Examples | Category | # Examples |
|---|---|---|---|---|---|
| animal | 20,733 | disease | 21,827 | muscle | 21,606 |
| beverage | 18,932 | drug | 20,452 | person | 21,700 |
| bird | 19,263 | fish | 19,162 | protein | 21,811 |
| bodypart | 21,840 | food | 19,566 | river | 21,723 |
| city | 21,778 | fruit | 18,911 | vegetable | 18,826 |

Table 1: A listing of the 15 NELL categories we used as the domains in our experiments, along with the number of 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 we use 11 classifiers 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, and we have 924 labeled examples for each one of those locations. Additional details can be found in (Wehbe et al., 2014).

### 5.1 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 a second one for evaluating the accuracy of the labels inferred by our methods. We describe each one of those two experiments in the following sections.

#### 5.1.1 Error Rates Experiment

For this experiment, we use the data without their labels to estimate error rates. We simultaneously use the labels of the data in order to compute an estimate of the true error rate. We estimate the true error rate simply by 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 that estimate of the true error rates as the “true error rates”. The evaluation metric we use to report our results is the mean absolute deviation (MAD) of the error rate estimates from the true error rates. A low mean absolute deviation indicates that our method is performing well.

#### 5.1.2 Labels Experiment

With this experiment we want to evaluate how accurate the inferred single output labels are. The evaluation metric we use to report our results is, once again, the mean absolute deviation (MAD) of the label estimates from the true labels that are known. Note that since the labels are binary variables, the MAD reduces to simply the accuracy of the labels (i.e., the ratio of correct labels to total number of labels). Note that only the graphical model approaches presented in section 4.2 infer those labels directly. We compare the inferred labels of those methods with the labels computed in the following way as alternatives:

1. Majority Vote: This is the most intuitive method to use and it consists of simply taking the most common label among the classifier outputs as the combined label.
2. Agreement Rates (AR): This method is basically a weighted majority vote, where the classifiers' predictions are weighted according to their error rates estimated according to the AR method presented in section 4.1. Since that method is not designed for inferring the labels, we use this voting scheme as a way of comparing it to the graphical model approaches.

### 5.1.3 Experimental Setup

For all our experiments and all three of the graphical models presented in section 4.2, the Gibbs sampling inference procedure we used consisted of the following steps: (i) we first generate $18,000$ samples that we throw away (i.e., burn-in samples), (ii) we then generate $2,000$ samples and keep every $10^{\text{th}}$ sample in order to reduce the correlations between our samples introduced by the sequential nature of the sampling procedure, and (iii) we obtain our error rate and label estimates by averaging over the samples that we kept. We set the hyperparameters of the graphical model approaches as follows:

- Labels Prior: $\alpha_p$ and $\beta_p$ are both set to 1, and so the prior is uniform and uninformative.
- Error Rates Prior: $\alpha_e$ is set to 1 and $\beta_e$ 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.
- 
- DP and HDP Concentration Parameters: We carried out several experiments with many logarithmically spaced values for $\alpha$ and $\gamma$ (all combinations of pairs of values were considered for the HDP) and we computed the log-likelihood for a held-out data set[5], 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.

In everything that follows, we use the following abbreviations for the different method names: **MAJ** is used to refer to the simple majority vote method used for combining multiple classifier outputs into a single label, **AR** is used to refer to the agreement rates method presented in section 4.1 (the suffix "**-2**" refers to the case when only pairwise dependency terms are considered), **EE** is used to refer to our simple error estimation model described in section 4.2.1, **CEE** is used to refer to our coupled error estimation model described in section 4.2.2, and finally, **HCEE** is used to refer to our hierarchical coupled error estimation model described in section 4.2.3.

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

### 5.2 General Observations

As outlined in tables 2 and 5.4, in all of our experiments, *one of the graphical model approaches always outperformed the other methods*. One thing that we expect to see in our results, in the presence of dependencies across domains and classifiers, is that CEE and HCEE perform better than EE when we have a small amount of data. It is in this scenario that sharing information becomes more useful. In the case that we have a large amount of data, we expect that the performance of the simple EE model will be similar its extensions, since for CEE and HCEE, the atoms may not be clustered because there may be enough data per atom to infer its posterior accurately that no sharing of information is necessary. That is evident in the results that we obtained. In the following two sections we discuss those for each data set in some detail.

### 5.3 NELL Data Set Results

We initially applied the AR method using only the ADJ, CPL, and VERB classifiers, while assuming that they make independent errors. The method for estimating error rates in this case is described in section 4.1.1 (note that we only estimate the individual function error rates). The resulting MAD is $2.82 \times 10^{-2}$; that is, the average error estimate is within a few percent of the true error. Although encouraging, this MAD is poor in comparison to our less restricted methods whose results are presented below, and indicates that the assumption that the classifiers make independent errors is an incorrect one in this case (and this behavior is typical more generally, as a matter of fact). Some of the obtained error rates are not even within the interval $[0, 1]$ and are thus obviously incorrect, since we know by the construction of the problem that the true error rates must lie in this interval (i.e., they must be valid probabilities). From now on, we consider only the more general case of $N$ functions that make dependent errors, thus making no independence assumptions.

Table 2 presents results for all of our methods with the entire NELL data set. It includes the results obtained when using all available data samples (i.e. the numbers shown in table 5.1) and when using only 10% of the data samples per category. It is clear from this table that the more data samples we have, the better our methods perform. That is presumably due to the more accurate estimates of the true agreement rates for the AR method, and for the other methods probably due to the larger volume of evidence we have to perform inference with. The first and most important observation is that our three proposed graphical models always outperform the other methods, for this data set. Especially in the case of limited data, we obtain 3 times better accuracy in estimating error rates than the AR-2 method, and 2 to 5 times better accuracy for the inferred labels, than all other methods. Moreover, now it becomes clear why the $2.82 \times 10^{-2}$ MAD that we obtained when we assumed independent error events is comparatively a bad result. All of the proposed methods manage to achieve an MAD that is almost 6 to 10 times better than that.

When using all of the available data samples, we observe that the simple EE model performs best. CEE and HCEE perform similarly, since we saw in our results that every atom has its own cluster and, given an infinite amount of data, both models will reduce to the simple EE model. The results are not exactly identical for the three models most probably due to the fact that the three models use different priors that allow different levels of information sharing. In the case of limited data samples (i.e., 10% of the full amount of

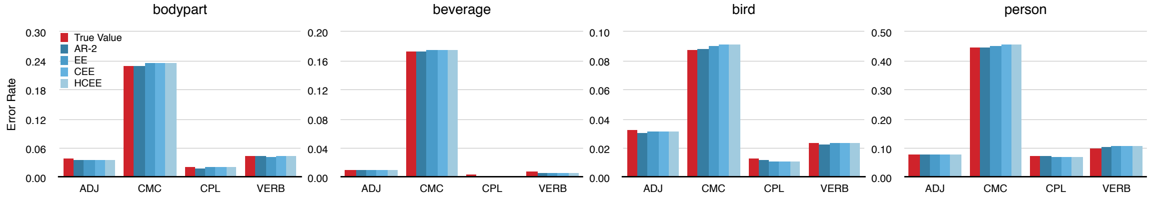


Figure 4: True errors (red bars) versus errors estimated from unlabeled data using each one of our proposed methods (blue bars), for four competing function approximations (ADJ, CMC, CPL and VERB), and four different target function domains (i.e. "bodypart", "beverage", "bird", and "person") using the NELL data set. Note each plot uses a different vertical scale to make it easier to observe the accuracy of the error rates estimates.

| $\times 10^{-2}$ | NELL Data Set | | | |
|---|---|---|---|---|
| | All Data Samples | | 10% of Data Samples | |
| | $\text{MAD}_{\text{error}}$ | $\text{MAD}_{\text{label}}$ | $\text{MAD}_{\text{error}}$ | $\text{MAD}_{\text{label}}$ |
| MAJ | - | 5.60 | - | 5.47 |
| AR-2 | 0.59 | 2.21 | 1.00 | 2.36 |
| AR | 0.66 | 2.20 | 0.70 | 2.36 |
| EE | **0.29** | 0.96 | 0.65 | 1.32 |
| CEE | 0.31 | **0.94** | 0.58 | 0.96 |
| HCEE | 0.31 | 0.96 | **0.31** | **0.95** |

Table 2: Mean absolute deviation (MAD) of the error rate and the label estimates, for the NELL data set and for all of the presented methods. The lower the MAD, the better the result is. Results are also included for when only 10% of the available data is used, which was randomly sampled, so that the advantages of the coupled error estimation models can be shown, for when only a limited amount of data is available. The best results for each experiment, across all methods, are shown in **bolded** text.

data available), the HCEE method performs the best, followed by CEE. This supports our argument that our coupled error estimation methods are more powerful for cases where a limited amount of data is available. Despite the fact that this is not really the case with the available 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 those cases even unlabeled data can be really hard and expensive to obtain.

Note that we also run an experiment by using the approximation described in section 4.1.3 and setting $M_e = 2$ (i.e. using AR-2 which considers only pairwise agreement rates). The results were sightly better than using all function subsets agreement rates for the case when we use all of the data, but they were slightly worse when were using only 10% of the data. Overall, the consistency of the results suggests that this proposed approximation

### INDEPENDENCE ASSUMPTION WEAKNESS

In order to make it more clear that the independence assumption is not very appropriate even in the case of NELL where a significant amount of effort has been put into having the NELL classifiers make independent errors, we provide here a measure of that dependence. We compute the following quantity for each domain:

$$\frac{1}{Z}\sum_{i,j}\left|\frac{e_{\{i,j\}}}{e_{\{i\}}e_{\{j\}}}-1\right|, \tag{25}$$

where $Z$ is the total number of terms in the sum, and we average over all domains. That gives us a measure of the average dependence of the functions error rates across all domains. If the functions make independent errors, then this quantity should be equal to $0$. We computed this quantity for the NELL data set using the sample error rates, which are an estimate of the true error rates (a pretty accurate estimate since we have about $20,000$ data samples per domain), and we obtained a value of $8.1770$, which is indeed quite far from $0$. That indicates why our methods, and especially the AR method, do so much better than the exact solution when assuming independent errors.

method is useful (there was a significant speedup as well – the code run about 3 times faster for this data set).

Figure 4 provides a plot of the estimated error rates for all of the presented methods, along with the true error rates for four randomly selected NELL classification problems (for brevity, plots for all classification problems are not included in this paper). This plot gives an idea of how well our proposed methods perform, and helps to make sense of the reported MAD values. As is easily seen in this plot, irrespective of the exact error estimate, *the ranking of the competing function approximations based on error rate is recovered exactly* by using all of the methods. And that is in fact true for each of the 15 NELL target function classification problems we evaluated – not only for the four shown in this figure.

### 5.4 Brain Data Set Results

Table 5.4 presents results for all of our methods used with the entire NELL data set. It includes the results obtained when using all available data samples and when using only 10% of the data samples per category. It is clear from this table that the more data samples we have the better our methods perform. That agrees with our observations from the NELL data set experiments. The first and most important observation is that the full AR method, without the approximation described in section 4.1.3 (i.e., the AR-2 method), completely fails in this case. This is probably due to the fact that the optimization problem that needs to be solved now is very large and actually very hard to solve. It is most likely that the optimization solver we are using simply fails to solve the problem in this case. These results also show the usefulness of the approximation described in section 4.1.3 where we set $M_e = 2$ (i.e., we consider only pairwise agreement rates).

Our simple EE model seems to consistently perform worse than AR-2. That is probably due to the fact that there is a much more limited amount of data than in the NELL case,

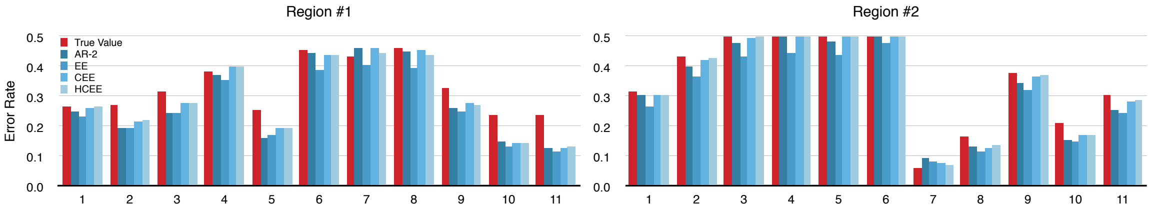


Figure 5: True errors (red bars) versus errors estimated from unlabeled data using each one of our proposed methods (blue bars), for eleven competing function approximations (based on different story features), and two different target function domains (using neural activity from two different brain regions) using the brain data set. Note estimates from unlabeled data are quite close to true errors, even though not as close as for the NELL data set.

| $\times 10^{-2}$ | Brain Data Set | | | |
|---|---|---|---|---|
| | All Data Samples | | 10% of Data Samples | |
| | $\text{MAD}_{\text{error}}$ | $\text{MAD}_{\text{label}}$ | $\text{MAD}_{\text{error}}$ | $\text{MAD}_{\text{label}}$ |
| MAJ | - | 19.82 | - | 20.82 |
| AR-2 | 5.14 | 18.67 | 5.84 | 20.14 |
| AR | 15.29 | 19.82 | 14.96 | 19.86 |
| EE | 6.77 | **17.23** | 20.20 | 20.03 |
| CEE | 4.07 | 17.51 | **4.69** | **17.42** |
| HCEE | **4.04** | 17.34 | 5.74 | 18.51 |

Table 3: Mean absolute deviation (MAD) of the error rate and the label estimates, for the brain data set and for all of the presented methods. The lower the MAD, the better the result is. Results are also included for when only 10% of the available data is used, which was randomly sampled, so that the advantages of the coupled error estimation models can be shown, for when only a limited amount of data is available. The best results for each experiment, across all methods, are shown in **bolded** text.

which makes inference harder. Note also that different domains in this case are expected to be significantly dependent. Since they correspond to different locations in the brain and, naturally, some locations are neighboring and exhibit similar "behavior". CEE performs better than EE and it actually manages to beat all of the other competing methods, except HCEE. HCEE manages to perform even better than CEE. Those observations support the argument that coupling becomes useful in the case of limited data. In this case we have a limited amount of data and we even have strong dependencies across domains and classifiers, which motivates clustering them.

When we subsample this data set, HCEE performs worse than CEE, which beats every other method. Our most likely explanation for this observation is that, due to the increased complexity of the HCEE method, it becomes very hard to fit the model to the very limited amount of data that are available per domain (note that in this case we only have 92 examples per domain). So, the decreased performance we are observing for HCEE, compared to CEE, is the result of *overfitting*. CEE which is a simpler model, and thus has lower model complexity, beats HCEE. EE performs badly, probably due to the fact that sharing information across domains becomes advantageous in this case. What we observe here is basically an instance of the *bias-variance trade-off*. The data we have in this case are way too few to fit the HCEE model well, but they are enough to allow CEE to outperform EE.

Figure 5 provides a plot of the estimated error rates for all of the presented methods, along with the true error rates for two randomly selected brain regions (for brevity, plots for all regions are not included in this paper). This plot gives an idea of how well our proposed methods perform, and helps to make sense of the reported MAD values, but at the same time we observe that in this case the estimates are not as good as they were in the NELL data set experiments. This can probably be attributed to the fact that the classifiers in the brain data set experiments are likely much more dependent than in the case of NELL, since we made an effort, during the development of the NELL classifiers, to make them as independent as possible. As is easily seen in this plot, irrespective of the exact error estimate, *the ranking of the competing function approximations based on error rate is still recovered exactly* by using all of the methods. And that is in fact true for each of the 11 brain regions we evaluated – not only for the two shown in this figure – as it was in the NELL data set experiments.

## 6. Conclusion

We have introduced the concept of estimating the error rate of each of several approximations to the same function, based on their agreement rates over *unlabeled data* and we have provided several different analytical methods to do so. We first proposed a method that uses the agreement rates of those function approximations to formulate an optimization problem to be solved. We then introduced a Bayesian approach that 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. As part of that approach, we first proposed a simple generative model for error estimation. That model is implicitly using the functions' agreement rates over *unlabeled data* in order to infer their error rates, along with the most likely single label. 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. That model uses a Dirichlet process prior as a means to perform the grouping of the domains; it is particularly useful when the available data are limited. 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 those clusters across different domain groups. This model uses a hierarchical Dirichlet process prior.

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 experiments showed that all proposed methods perform well in practice for both data sets we considered, with the Bayesian methods outperforming the agreement rates-based approach. Our results are very encouraging and suggest that function agreement rates are indeed very useful in estimating function error rates, while using them either explicitly, or implicitly. Thus, following on the argument presented in section 3, our results indicate that consistency among multiple functions is indeed related to their true accuracies, and the nature of that relationship has to do with how depended those functions are.

There are several potential future directions for this work. We wish to explore other interesting natural objectives one can aim to optimize, as described in section 4.1.2. It would also be very interesting to explore possible 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. Doing so would mean that the system using such function approximations would be able to "reflect" on its own performance and improve itself. This is the reason we consider this work as a step towards developing a *self-reflection framework* for autonomous learning systems. In this context, we could try using our estimates in order to develop a more robust co-learning framework, or in the case of NELL, 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.

## Acknowledgments

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. Furthermore, we thank Avrim Blum for working with us in developing the agreement rates-based approach, and Avinava Dubey for working with us in developing the graphical model-based approaches. Finally, we would like to thank Abulhair Saparov for the helpful discussions we had on nonparametric Bayesian models and on the overall feedback he has given us from time to time for this work.

## Appendix A. Gibbs Sampling Equations

In this section we provide the equations necessary to perform Gibbs sampling over the models defined in our paper.

### A.1 Coupled Error Estimation Model

The conditional probabilities we use during the first, collapsed sampling phase are as follows:

$$P(p^d \mid \cdot) = \text{Beta}\left(\alpha_p + \sum_{i=1}^{S^d} \ell_i^d, \beta_p + S^d - \sum_{i=1}^{S^d} \ell_i^d\right), \tag{26}$$

$$P(\ell_i^d \mid \cdot) \propto (p^d)^{\ell_i^d} (1 - p^d)^{1-\ell_i^d} \mathcal{B}\left(\alpha_{z^d}^d + \sum_{\hat{i}=1}^{N} \sum_{j=1}^{M} \mathbb{1}_{\{\hat{f}_{\hat{i}j}^d \neq \ell_{\hat{i}}^d\}}, \beta_{z^d}^d + \sum_{\hat{i}=1}^{N} \sum_{j=1}^{M} \mathbb{1}_{\{\hat{f}_{\hat{i}j}^d = \ell_{\hat{i}}^d\}}\right), \tag{27}$$

$$P(z^d = k \mid \cdot) \propto \begin{cases} Z_k^d \frac{\mathcal{B}(\alpha_k^d + \sigma^d, \beta_k^d + S^d - \sigma^d)}{\mathcal{B}(\alpha_e + \alpha_k^d, \beta_e + \beta_k^d)} & \text{, if } Z_k^d > 0, \\ \alpha \frac{\mathcal{B}(\alpha_e + \sigma^d, \beta_e + S^d - \sigma^d)}{\mathcal{B}(\alpha_e, \beta_e)} & \text{, otherwise,} \end{cases} \tag{28}$$

where:

$$\alpha_k^d = \alpha_e + \sum_{\substack{\hat{d}=1 \\ \hat{d} \neq d}}^{D} \mathbb{1}_{\{z^{\hat{d}}=k\}} \sigma^{\hat{d}}, \qquad \beta_k^d = \beta_e + \sum_{\substack{\hat{d}=1 \\ \hat{d} \neq d}}^{D} \mathbb{1}_{\{z^{\hat{d}}=k\}} (S^{\hat{d}} - \sigma^{\hat{d}}), \tag{29}$$

$$\sigma^d = \sum_{j=1} \sum_{i=1}^{S^d} \mathbb{1}_{\{\hat{f}_{ij}^d \neq \ell_i^d\}}, \tag{30}$$

$$Z_k^d = \sum_{\substack{\hat{d}=1 \\ \hat{d} \neq d}}^{D} \mathbb{1}_{\{z^{\hat{d}}=k\}}, \tag{31}$$

$\mathbb{1}_{\{\cdot\}}$ evaluates to one if its subscript's argument statement is true and to zero otherwise, and $\mathcal{B}(\cdot, \cdot)$ is the Beta function.

After that phase, we start sampling the error rates along with the rest of the variables and store the samples we obtain. During that second phase, we use the following conditional probabilities:

$$P(p^d \mid \cdot) = \text{Beta}\Big(\alpha_p + \sum_{i=1}^{S^d} \ell_i^d, \beta_p + S^d - \sum_{i=1}^{S^d} \ell_i^d\Big), \tag{32}$$

$$P(\ell_i^d \mid \cdot) \propto (p^d)^{\ell_i^d} (1 - p^d)^{1-\ell_i^d} \prod_{j=1}^{N} (e_j^d)^{\mathbb{1}_{\{\hat{f}_{ij}^d \neq l_i^d\}}} (1 - e_j^d)^{\mathbb{1}_{\{\hat{f}_{ij}^d = l_i^d\}}} \tag{33}$$

$$P(z^d = k \mid \cdot) \propto \begin{cases} Z_k^d \prod_{j=1}^{N} (e_j^d)^{\sigma_j^d} (1 - e_j^d)^{S^d - \sigma_j^d} & \text{, if } Z_k^d > 0, \\ \alpha \mathcal{P}_{\text{new}}^d & \text{, otherwise,} \end{cases} \tag{34}$$

$$P([\phi_k]_j \mid \cdot) = \text{Beta}\Big(\alpha_e + \sum_{d=1}^{D} \mathbb{1}_{\{z^d=k\}} \sigma_j^d, \beta_e + \sum_{d=1}^{D} \mathbb{1}_{\{z^d=k\}} (S^d - \sigma_j^d)\Big). \tag{35}$$

In the case of missing data the conditional probability of the simple error estimation model can be used, which is provided in our paper.

### A.2 Hierarchical Coupled Error Estimation Model

The conditional probabilities we use during the first, collapsed sampling phase are as follows:

$$P(p^d \mid \cdot) = \text{Beta}\Big(\alpha_p + \sum_{i=1}^{S^d} \ell_i^d, \beta_p + S^d - \sum_{i=1}^{S^d} \ell_i^d\Big), \tag{36}$$

$$P(\ell_i^d \mid \cdot) \propto (p^d)^{\ell_i^d} (1 - p^d)^{1-\ell_i^d} L_i^d, \tag{37}$$

$$P(z_j^d = t \mid k_t^d = k, \cdot) \propto Z_{kt}^d \frac{\mathcal{B}(\alpha_{jk}^d + \sigma_j^d, \beta_{jk}^d + S^d - \sigma_j^d)}{\mathcal{B}(\alpha_{jk}^d, \beta_{jk}^d)}, \tag{38}$$

$$P(k_t^d = k \mid \cdot) \propto Z_k^d \frac{\mathcal{B}(\alpha_{\boldsymbol{j}k}^d + \sum_{j \in \boldsymbol{j}} \sigma_j^d, \beta_{jk}^d + \sum_{j \in \boldsymbol{j}} (S^d - \sigma_j^d)}{\mathcal{B}(\alpha_{\boldsymbol{j}k}^d, \beta_{\boldsymbol{j}k}^d)}, \tag{39}$$

where $\boldsymbol{j} = \{j : z_j^d = t\}$, and:

$$L_i^d = \prod_{k=1}^{K} \mathcal{B}\bigg(\alpha_k^d + \sum_{j=1}^{M} \sum_{\hat{i}=1}^{N} \mathbb{1}_{\{k_{z_j^d}^d = k\}} \mathbb{1}_{\{\hat{f}_{\hat{i}j}^d \neq \ell_{\hat{i}}^d\}}, \beta_k^d + \sum_{\hat{i}=1}^{N} \sum_{j=1}^{M} \mathbb{1}_{\{k_{z_j^d}^d = k\}} \mathbb{1}_{\{\hat{f}_{\hat{i}j}^d = \ell_{\hat{i}}^d\}}\bigg), \tag{40}$$

$$\alpha_k^d = \alpha_e + \sum_{\substack{\hat{d}=1 \\ \hat{d} \neq d}}^{D} \sum_{\hat{j}=1}^{N} \mathbb{1}_{\{k_{z_{\hat{j}}^{\hat{d}}}^{\hat{d}} = k\}} \sigma_{\hat{j}}^{\hat{d}}, \qquad \beta_k^d = \beta_e + + \sum_{\substack{\hat{d}=1 \\ \hat{d} \neq d}}^{D} \sum_{\hat{j}=1}^{N} \mathbb{1}_{\{k_{z_{\hat{j}}^{\hat{d}}}^{\hat{d}} = k\}} (S^{\hat{d}} - \sigma_{\hat{j}}^{\hat{d}}), \tag{41}$$

$$\alpha_{jk}^d = \alpha_k^d + \sum_{\substack{\hat{j}=1 \\ \hat{j} \neq j}}^{N} \mathbb{1}_{\{k_{z_{\hat{j}}^d}^d = k\}} \sigma_{\hat{j}}^d, \qquad \beta_{jk}^d = \beta_k^d + \sum_{\substack{\hat{j}=1 \\ \hat{j} \neq j}}^{N} \mathbb{1}_{\{k_{z_{\hat{j}}^d}^d = k\}} (S^d - \sigma_{\hat{j}}^d), \tag{42}$$

$$Z_{kt}^d = \begin{cases} \sum_{\substack{\hat{j}=1 \\ \hat{j} \neq j}}^{N} \mathbb{1}_{\{z_{\hat{j}}^d = t\}} & \text{, if } t \text{ occupied,} \\ \alpha \sum_{\substack{\hat{d}=1 \\ \hat{d} \neq d}}^{D} \sum_{\hat{t}} \mathbb{1}_{\{k_{\hat{t}}^{\hat{d}} = k\}} & \text{, if } t \text{ unoccupied and } k \text{ exists,} \\ \alpha\gamma & \text{, if } t \text{ unoccupied and } k \text{ is new,} \end{cases} \tag{43}$$

$$Z_k^d = \begin{cases} \sum_{\substack{\hat{d}=1 \\ \hat{d} \neq d}}^{D} \sum_{\hat{t}} \mathbb{1}_{\{k_{\hat{t}}^{\hat{d}} = k\}} & \text{, if } k \text{ exists,} \\ \gamma & \text{, if } k \text{ is new,} \end{cases} \tag{44}$$

where, noting that our previous definitions for $\alpha_{jk}^d$ and $\beta_{jk}^d$ can also apply to sets over functions, $\boldsymbol{j}$, we also have that:

$$\alpha_{\boldsymbol{j}k}^d = \alpha_k^d + \sum_{\substack{\hat{j}=1 \\ \hat{j} \notin \boldsymbol{j}}}^{N} \mathbb{1}_{\{k_{z_{\hat{j}}^d}^d = k\}} \sigma_{\hat{j}}^d, \qquad \beta_{\boldsymbol{j}k}^d = \beta_k^d + \sum_{\substack{\hat{j}=1 \\ \hat{j} \notin \boldsymbol{j}}}^{N} \mathbb{1}_{\{k_{z_{\hat{j}}^d}^d = k\}} (S^d - \sigma_{\hat{j}}^d). \tag{45}$$

After that phase, we start sampling the error rates along with the rest of the variables and store the samples we obtain. During that second phase, we use the following conditional probabilities:

$$P(p^d \mid \cdot) = \text{Beta}\bigg(\alpha_p + \sum_{i=1}^{S^d} \ell_i^d, \beta_p + S^d - \sum_{i=1}^{S^d} \ell_i^d\bigg), \tag{46}$$

$$P(\ell_i^d \mid \cdot) \propto (p^d)^{\ell_i^d} (1 - p^d)^{1 - \ell_i^d} \prod_{j=1}^{N} (e_j^d)^{\mathbb{1}_{\{\hat{f}_{ij}^d \neq l_i^d\}}} (1 - e_j^d)^{\mathbb{1}_{\{\hat{f}_{ij}^d = l_i^d\}}} \tag{47}$$

$$P(z_j^d = t \mid k_t^d = k, \cdot) \propto Z_{kt}^d (e_j^d)^{\sigma_j^d} (1 - e_j^d)^{S^d - \sigma_j^d}, \tag{48}$$

$$P(\phi_k \mid \cdot) = \text{Beta}\Bigg(\alpha_e + \sum_{d=1}^{D}\sum_{j=1}^{N} \mathbb{1}_{\{k^d_{z^d_j}=k\}}\sigma^d_j, \beta_e + \sum_{d=1}^{D}\sum_{j=1}^{N} \mathbb{1}_{\{k^d_{z^d_j}=k\}}(S^d - \sigma^d_j)\Bigg), \tag{49}$$

$$P(k^d_t = k \mid \cdot) \propto Z^d_k \prod_{j\in\boldsymbol{j}} (e^d_j)^{\sigma^d_j}(1-e^d_j)^{S^d-\sigma^d_j}, \tag{50}$$

where $\boldsymbol{j} = \{j : z^d_j = t\}$. In the case of missing data the conditional probability of the simple error estimation model can be used, which is provided in our paper.

## References

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

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

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

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

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, 1998. doi: 10.1145/279943.279962.

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

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

Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding Semi-Supervision with Constraint-Driven Learning. In *Annual Meeting of the Association of Computational Linguistics*, pages 280–287, Prague, Czech Republic, June 2007.

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

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

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

Pinar Donmez, Guy Lebanon, and Krishnakumar Balasubramanian. Unsupervised Supervised Learning I: Estimating Classification and Regression Errors without Labels. *Journal of Machine Learning Research*, 11:1323–1351, April 2010.

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

Stuart Geman and Donald Geman. 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.

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

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

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

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

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

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

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

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

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

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