Selective Verification Strategy for Learning from Crowds

Tian Tian, Yichi Zhou, Jun Zhu*
Dept. of Comp. Sci. & Tech., CBICR Center, State Key Lab for Intell. Tech. & Systems
TNList, Tsinghua University, Beijing, China
{tiant16@mails., zhouyc15@mails., dcszj@}tsinghua.edu.cn

Abstract

To deal with the low qualities of web workers in crowdsourcing, many unsupervised label aggregation methods have been investigated but most of them provide inconsistent performance. In this paper, we explore the *learning from crowds with selective verification* problem. In addition to the noisy responses from the crowds, it also collects the ground truths for a well-chosen subset of tasks as the reference, then aggregates the redundant responses based on the patterns provided by both the supervised and unsupervised signal. To improve the labeling efficiency, we propose the EBM selecting strategy for choosing the verification subset, which is based on the loss error minimization. Specifically, we first establish the expected loss error given the semi-supervised learning estimate, then find the subset that minimizes this selecting criterion. We do extensive empirical comparisons on both synthetic and real-world datasets to show the benefits of this new learning setting as well as our proposal.

Introduction

How to collect a massive scale of high-quality labels quickly? This question frequently appears when building modern artificial intelligence agents, where the interminable data acquiring procedure is often a bottleneck (Deng et al. 2009). To schedule this manual work efficiently, *crowdsourcing* platforms, such as Amazon Mechanical Turks (AMT), provide a way to distribute micro-tasks to a large group of web workers, so the tasks can be done much faster and cheaper than expert labeling (Snow et al. 2008). However, due to the different backgrounds of humans, responses of the crowds may contain substantial errors. Among existing attempts to improve the quality of crowdsourced data, a large proportion of methods suggest labeling each task by multiple workers, and then infer the underlying truths based on the patterns within the redundant responses. This inference problem is known as *learning from crowds*.

There has been abundant literature for noisy label aggregation methods, which date back to the intuitive majority voting method that just chooses the majority opinions of workers. Dawid and Skene (1979) proposed a generative model, which uses confusion matrices to describe the workers' behaviors and an EM algorithm for learning its parameters. Recently, researchers have developed many advanced methods, which make different assumptions on human behaviors (Jagabathula, Subramanian, and Venkataraman 2014; Li and Yu 2014; Tian and Zhu 2015a; Wauthier and Jordan 2011; Welinder et al. 2010; Zhou et al. 2012), task difficulties (Whitehill et al. 2009), and label structures (Tian and Zhu 2015b; Welinder et al. 2010), or have different inference methods (Liu, Peng, and Ihler 2013; Zhang et al. 2016). Although each method can achieve good performance on several benchmark datasets, some recent comparisons have shown a disappointing fact that the superiority of most methods is not consistent (Hung et al. 2013; Sheshadri and Lease 2013; Zheng et al. 2017). Their performance may drop down when the model assumptions do not hold. Since the applicable situations of methods are not clear, there still exists a gap between label aggregation methods and real crowdsourcing applications.

From our perspective, the model assumption is one of the key points that influence the aggregating accuracy, and complex assumptions could be violated by certain datasets. One possible solution towards building robust label aggregation methods is to introduce oracle verification directly to reduce the influence of invalid assumptions. If we select suitable verification tasks, comparing the responses of web workers with the ground truths on the selected tasks can help us understand the workers' behaviors and do better label aggregation.

Based on the above idea, we focus on an approach we call *learning from crowds with selective verification* (Hung et al. 2015). This approach requires two kinds of labels, the crowdsourced labels and the truths for a well-chosen portion of tasks. After querying the crowds, we analyze the responses and select the most valuable verification tasks. Then we further collect the ground truths from the oracle for this subset of tasks. Finally based on the patterns discovered from both the noisy crowdsourced labels and the ground truths, we infer the truths of the rest tasks. In this way, the potential mismatches of model assumptions can be complemented by the verification step. The two key learning problems of this approach are: (1) how to combine the oracle labels and the crowdsourced responses to infer the ground truths? (2) how to select a most valuable verification subset? We solve the first problem through a flexible transductive semi-supervised learning approach (Zhu, Ghahramani, and

*corresponding author.

Lafferty 2003), which combines the unsupervised and the supervised estimators into a joint objective (see problem (3)). To solve the second problem, we explore different active learning methods, especially the error bound minimization techniques (Chaudhuri et al. 2015; Gu, Zhang, and Han 2014; Gu et al. 2012), which select the supervised subset by minimizing an error bound for the expected log-likelihood given the estimated parameters. Previous work on this topic only focuses on supervised learning settings, and we extend this method to the semi-supervised learning setting.

Learning from crowds with selective verification can also be viewed as a combination of the postprocessing and preprocessing crowdsourcing approaches. The postprocessing approach collects noisy labels from multiple workers and then infers the truths. The preprocessing approach first does qualification tests on workers, then only the qualified workers are allowed to participate in the regular tasks. Commercial crowd labeling platforms often use both approaches. For example, when publishing labeling tasks, AMT provides an option of only hiring annotators with an acceptance rate of higher than 95% in previous tasks, which is a preprocessing filter. Then it also suggests duplicating each task multiple times and provides aggregating results, which is a postprocessing approach. Currently, these methods are treated as orthogonal treatments. Our work conjoins them into a unified semi-supervised learning framework. Selecting suitable tests for pure preprocessing approaches is usually not easy. Through minimizing an expected loss error of the joint model, the postprocessing can help the preprocessing find the best tasks for the qualification tests.

In summary, our main contributions are as follows. First, we build the active transductive semi-supervised learning framework for crowdsourcing, which has potential to consistently provide high-quality labels. Then, we propose the EBM selective verification strategy through minimizing an expected loss error on the semi-supervised learning setting. It can be independently used on any semi-supervised learning framework. Finally, we do extensive empirical comparisons to show the benefits of our proposal.

Related Work

The main task of learning from crowds is to improve the data quality. As stated above, there are two main approaches. **Preprocessing:** These methods focus on annotator filters. They tend to evaluate the expertise of annotators through some extra qualification tests, and then only the reliable annotators will be employed for the major labeling tasks (Jagabathula, Subramanian, and Venkataraman 2014; Shah and Zhou 2015). One defect is that their labeling qualities are susceptive to the tasks selected for the qualification tests, and it's usually hard to find the best verification tasks. **Postprocessing:** These methods typically label each task multiple times and then infer worker behaviors as well as the truth of each task from the redundant responses. For example, most previous methods model the user behaviors directly without considering the inner structures and relations among the items (Dawid and Skene 1979; Raykar et al. 2010; Wang and Zhou 2016; Zhang et al. 2016; Zhou et al. 2012). Some works discover latent representations of items (Welinder et al. 2010), annotators (Venanzi et al. 2014) and labels (Tian and Zhu 2015b). The success of these postprocessing methods relies on the assumption that most annotators behave consistently and give true answers, and the methods may perform inconsistently on different trials when this does not hold (Hung et al. 2013; Sheshadri and Lease 2013; Zheng et al. 2017).

Existing methods for combining crowdsourced labels and the truths explore statistical or graph methods (Khattak and Salleb-Aouissi 2016). Tang and Lease (2011) directly inject the truths into the confusion matrix model during the EM updates. When training classifiers, a common choice is to regard the oracle as a special worker (Hu et al. 2014; Kajino et al. 2012). Uncertainty based methods were explored to find the most problematic tasks (Hung et al. 2015). Different from above-mentioned works, we formulate the hybrid crowd-expert setting as a flexible transductive semi-supervised learning problem. Based on this model, we propose the selective verification strategy via expected loss error minimization, which is coupled with our learning objective.

From the learning aspect, our solution relates to semi-supervised learning and active learning. **Semi-supervised learning (SSL)** is a problem where we have features for all training instances, but only have labels for a small portion of instances. There are two kinds of SSL problems, that is, inductive learning and transductive learning. The former aims to train a classifier to be used on future testing data, while the latter (Zhu and Goldberg 2009) aims to learn the labels of the unlabeled instances in the dataset. Classic methods include self-training, co-training, transductive SVM (Joachims 1999), and graph-based methods (Zhu, Ghahramani, and Lafferty 2003). Generative models can describe the generating process of both features and known labels, recent techniques such as semi-supervised deep generative models have been proposed (Kingma et al. 2014). Our problem can be regarded as an example of the transductive SSL for generative models. **Active learning (AL)** focuses on selecting a most valuable subset of data to annotate in order to reduce the labeling cost (Settles 2010; Yang and Loog 2016). Pool-based AL usually designs criteria to measure the importance of instances in the dataset. Such as that based on uncertainty (Settles 2010), the estimate's variance (Wang, Yu, and Singh 2016), etc. The error bound minimization method (Chaudhuri et al. 2015; Gu, Zhang, and Han 2014; Gu et al. 2012) selects the subset by minimizing the expected estimating error bound. This method can handle the dependency among instances for batch-mode AL. Previously it focuses on the pure supervised learning settings. In this work, we extend it to semi-supervised learning.

The Model

In this section, we introduce the model for learning from crowds with selective verification. We base our method on the classical Dawid-Skene (DS) model, which is relatively simple and captures the common issues appearing in many other models. Technically, in order to connect the unsupervised DS model with our semi-supervised learning setting,

we show the relationship between DS and the softmax regression problem. After that, we propose the transductive semi-supervised learning formulation, which jointly models both the crowdsourced labels and the oracle label. To find a most valuable verification subset, we then establish the expected loss error, and this value is decided by the verification subset. So we can find the best subset by minimizing this criterion. The optimization algorithms for the EBM strategy are discussed in the next section.

Notation and Problem Formulation

We first define our notations and problem formulation. Specifically, we consider categorical labels and leave other more complex label structures to future work. Assume there are N tasks, M annotators and D classes indexed by i, j and d respectively. Each task i has an unknown ground truth $y_i \in [D]$, where $[D] := \{1, \ldots, D\}$. During labeling, we have multiple workers providing labels to a same task. We use matrix $\boldsymbol{X} \in \{0,1\}^{N \times M \times D}$ to denote the crowdsourced responses, and each element $x_{ijd} = 1$ when item i is labeled by worker j into category d, otherwise $x_{ijd} = 0$. Note that workers are not required to respond to all tasks. The goal of learning from crowds is to infer the ground truths $\boldsymbol{y}$ given the crowdsourced responses $\boldsymbol{X}$.

To enhance the labeling quality, we query for oracle verification. In addition to the crowdsourced labels, we select a subset of tasks $\mathcal{L} \subset \mathcal{X}$, where $\mathcal{X}$ is the collection of all items, and query the oracle for their ground truths. Our goal on learning from crowds with selective verification is to find the most informative subset $\mathcal{L}$, then infer the ground truths of the rest unlabeled tasks $\boldsymbol{y}_{\mathcal{X} \setminus \mathcal{L}}$, based on the crowdsourced responses $\boldsymbol{X}$ and the ground truths $\boldsymbol{y}_{\mathcal{L}}$.

Supervised Dawid-Skene Estimator as Softmax Regression

To introduce oracle verification into the learning from crowds problem, we first consider the parameter estimation for the DS model under the fully supervised setting. In this part, we present that the supervised DS model is closely related to the softmax regression, whose statistical properties are important for constructing the selective verification strategy.

The DS model assumes that the performance of each worker is consistent across tasks with the same ground truth. The decision behavior of worker j is measured by a confusion matrix $\boldsymbol{\phi}_j$. Each element $\phi_{jkd} \in [0,1]$ denotes the probability that worker j gives label d to a task whose true label is k, so $\sum_{d=1}^{D} \phi_{jkd} = 1 : j \in [M], k \in [D]$. $\boldsymbol{\pi} \in [0,1]^D$ is the prior on the ground truths and $\sum_{d=1}^{D} \pi_d = 1$. Given observations $\boldsymbol{X}$, we can estimate the parameters $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_j : j \in [M]\}$ and $\boldsymbol{\pi}$ by maximum likelihood estimation (MLE) as

$$\begin{aligned} \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\pi}} &= \operatorname*{argmax}_{\boldsymbol{\Phi}, \boldsymbol{\pi}} \log p(\boldsymbol{X} | \boldsymbol{\Phi}, \boldsymbol{\pi}) \\ &= \operatorname*{argmax}_{\boldsymbol{\Phi}, \boldsymbol{\pi}} \sum_{i=1}^{N} \log \left(\sum_{k=1}^{D} \pi_k \prod_{j=1}^{M} \prod_{d=1}^{D} \phi_{jkd}^{x_{ijd}} \right). \quad (1) \end{aligned}$$

Although non-convex, we can find a local maximum by an expectation-maximization (EM) algorithm. After converging, we predict the ground truths as $\hat{y}_i = \operatorname{argmax}_{k \in [D]} \log p(y_i = k, \boldsymbol{X} | \boldsymbol{\Phi}, \boldsymbol{\pi}) = \sum_{j=1}^{M} \sum_{d=1}^{D} x_{ijd} \log \phi_{jkd} + \log \pi_k$. Since this discriminative function is linear to $\boldsymbol{X}$, by substitution, we see that $p(y_i | \boldsymbol{X}, \boldsymbol{\Phi}, \boldsymbol{\pi})$ conforms to a softmax distribution. Specifically, for $k \in [D-1]$, we have

$$\log \frac{p(y_i = k | \boldsymbol{X}, \boldsymbol{\Phi}, \boldsymbol{\pi})}{p(y_i = D | \boldsymbol{X}, \boldsymbol{\Phi}, \boldsymbol{\pi})} = \sum_{j=1}^{M} \sum_{d=1}^{D} x_{ijd} \log \frac{\phi_{jkd}}{\phi_{jDd}} + \log \frac{\pi_k}{\pi_D}.$$

We denote vectors $\boldsymbol{x}_i = [\boldsymbol{x}_{i1}^\top, \ldots, \boldsymbol{x}_{iM}^\top, 1]^\top$, $\boldsymbol{w}_{jk} = \left[\log \frac{\phi_{jk1}}{\phi_{jD1}}, \ldots, \log \frac{\phi_{jkD}}{\phi_{jDD}}\right]^\top$, $\boldsymbol{w}_k = [\boldsymbol{w}_{1k}^\top, \ldots, \boldsymbol{w}_{Mk}^\top, \log \frac{\pi_k}{\pi_D}]^\top$, and $\boldsymbol{W} = [\boldsymbol{w}_1^\top, \ldots, \boldsymbol{w}_{D-1}^\top]^\top$. They give the softmax regression model

$$p(y_i = k | \boldsymbol{x}_i, \boldsymbol{W}) = \begin{cases} \frac{\exp(\boldsymbol{w}_k^\top \boldsymbol{x}_i)}{1 + \sum_{s=1}^{D-1} \exp(\boldsymbol{w}_s^\top \boldsymbol{x}_i)} & k \in [D-1] \\ \frac{1}{1 + \sum_{s=1}^{D-1} \exp(\boldsymbol{w}_s^\top \boldsymbol{x}_i)} & k = D \end{cases}. \quad (2)$$

Under the supervised learning setting, we are given not only the crowdsourced labels $\boldsymbol{X}$, but also the truths $\boldsymbol{y}$ from an oracle. Then the estimation problem for $\hat{\boldsymbol{W}}$ under the DS estimator can be solved by softmax regression with the discriminative function defined in Eq. (2). Comparing the generative approach with the softmax regression, since the latter ignores considering $p(\boldsymbol{X} | \boldsymbol{W})$, its solution is not always equivalent to the maximum joint-likelihood estimation. However, it brings significant benefits for constructing a selective verification strategy, which will be discussed later.[1]

Semi-supervised Softmax Regression Estimator for Learning from Crowds

Above we formulate the DS estimator under the supervised setting as a softmax regression. However, for the learning from crowds with verification, we have ground truths $\boldsymbol{y}_{\mathcal{L}}$ only for a small subset of tasks $\mathcal{L}$. So we need to combine the unsupervised estimator with the supervised estimator, which is a semi-supervised learning problem. According to the generative nature, the parameter estimation problem can be given by

$$\begin{aligned} \hat{\boldsymbol{W}} &= \operatorname*{argmax}_{\boldsymbol{\Phi}, \boldsymbol{\pi}} \log p(\boldsymbol{X}, \boldsymbol{y}_{\mathcal{L}} | \boldsymbol{\Phi}, \boldsymbol{\pi}) \\ &= \operatorname*{argmax}_{\boldsymbol{\Phi}, \boldsymbol{\pi}} \log p(\boldsymbol{X} | \boldsymbol{\Phi}, \boldsymbol{\pi}) + \log p(\boldsymbol{y}_{\mathcal{L}} | \boldsymbol{X}_{\mathcal{L}}, \boldsymbol{\Phi}, \boldsymbol{\pi}). \end{aligned}$$

This objective function is divided into an unsupervised part and a supervised part. The unsupervised part is same as Eq. (1), while the supervised part can be formulated by the softmax regression just given in Eq. (2). Since the unsupervised learning problem (1) often stuck in local minima, it's

[1] When we have $\hat{\boldsymbol{W}}$, we can recover a unique set of parameters $\{\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\pi}}\}$ from $\hat{\boldsymbol{W}}$ by introducing the distribution constraints that $\sum_{k \in [D]} \pi_k = 1$ and $\sum_{d \in [D]} \phi_{jkd} = 1 : j \in [M], k \in [D]$. To ensure that the solutions are probabilities, we need further constrain that parameters in $\hat{\boldsymbol{\Phi}}$ are positive.

hard to analyze the error of its results after introducing the oracle labels. To make the selective verification problem easier for computation and analysis, here we introduce an approximation for problem (1), and propose a new semi-supervised problem for crowdsourcing.

During each iteration of the EM algorithm for solving problem (1), we establish a lower bound on the marginal likelihood by introducing the posterior distribution of the unknown truths as $q(\boldsymbol{y})$. We iteratively update $q(\boldsymbol{y})$ and $\{\boldsymbol{\Phi}, \boldsymbol{\pi}\}$. After converging, the solution we find is $\{\hat{\boldsymbol{\Phi}}^{Unsup}, \hat{\boldsymbol{\pi}}^{Unsup}\}$, and $p(\boldsymbol{y}|\boldsymbol{X}, \hat{\boldsymbol{\Phi}}^{Unsup}, \hat{\boldsymbol{\pi}}^{Unsup})$ is the corresponding posterior distribution. Here we have changed the notation for the unsupervised estimate to distinguish it from the semi-supervised estimate. Moreover, we denote $\hat{\boldsymbol{W}}^{Unsup}$ as the vector composed by $\{\hat{\boldsymbol{\Phi}}^{Unsup}, \hat{\boldsymbol{\pi}}^{Unsup}\}$ in the way we introduced in above subsection. Then we have

$$\hat{\boldsymbol{W}}^{Unsup} = \underset{\boldsymbol{W}}{\operatorname{argmax}} \mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{X}, \hat{\boldsymbol{W}}^{Unsup})} [\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{W})].$$

When the EM algorithm for the unsupervised estimator is well initialized and has enough crowdsourced labels, its estimate $\hat{\boldsymbol{W}}^{Unsup}$ should be similar to the true underlying parameter $\boldsymbol{W}^*$, so we approximately assume that $\hat{\boldsymbol{W}}^{Unsup}$ is identical to $\boldsymbol{W}^*$ during analysis for the simplicity of analysis and learning. The effect of this approximation will be analyzed empirically in the experiment section. Then we can use this cross entropy objective, instead of Eq. (1), to denote the belief from the unsupervised model in the joint learning problem. In the new learning problem, we also ignore the positive constraints for $\boldsymbol{\Phi}$, so we can optimize $\boldsymbol{W}$ over an unconstrained parameter space, which is more flexible.

Based on this idea, we define

$$U(\boldsymbol{W}) = -N \cdot \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}, y\sim p(y|\boldsymbol{x}, \hat{\boldsymbol{W}}^{Unsup})}[\log p(y|\boldsymbol{x}, \boldsymbol{W})],$$

$$L(\boldsymbol{W}, \mathcal{L}) = -B \cdot \mathbb{E}_{\boldsymbol{x}\sim\mathcal{L}, y\sim p(y|\boldsymbol{x}, \hat{\boldsymbol{W}}^{*})}[\log p(y|\boldsymbol{x}, \boldsymbol{W})],$$

where $\boldsymbol{W}^*$ is the unknown underlying optimal parameter to generate the observations[2] and $B = |\mathcal{L}|$. The scalar constants are introduced to balance the influence of each sample. Our semi-supervised learning problem is defined as

$$\hat{\boldsymbol{W}}^{\mathcal{L}} = \underset{\boldsymbol{W}}{\operatorname{argmin}} G(\boldsymbol{W}, \mathcal{L}) = U(\boldsymbol{W}) + \lambda \cdot L(\boldsymbol{W}, \mathcal{L}), \quad (3)$$

where λ is a hyperparameter to balance the supervised and unsupervised losses, it can be solved by gradient descent methods.

Since this semi-supervised estimator considers the verification information, its estimate should be more precise compared with the estimates of the pure unsupervised estimators. In the next part, we formulate the expected loss error of problem (3) given its estimate $\hat{\boldsymbol{W}}$ to explicitly show the model's superiority. More importantly, the expected loss error gives us a criterion to select the best verification subset. We will discuss it in the following parts.

Loss Error Analysis

Given the estimate based on the crowdsourced labels for N tasks and the ground truths for B tasks, we now compute the expected transductive learning loss error of our model. Similar error bounds have been established for linear regression and logistic regression (Chaudhuri et al. 2015; Gu, Zhang, and Han 2014; Gu et al. 2012). Here we extend their results to the semi-supervised learning problem for crowdsourcing in Eq. (3).

[2]There could exist several equivalent global optima. $\boldsymbol{W}^*$ denotes the one that closest to $\hat{\boldsymbol{W}}^{Unsup}$.

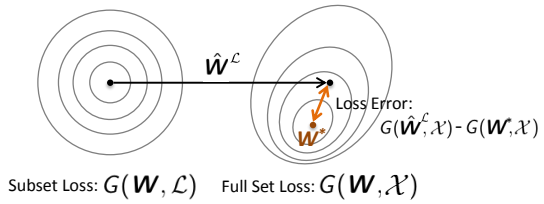

Figure 1: Illustration for the loss error.

Since different verification subset $\mathcal{L}$ induces different loss $G(\boldsymbol{W}, \mathcal{L})$, the estimate $\hat{\boldsymbol{W}}^{\mathcal{L}}$ found based on it will have different quality. Intuitively, the best loss among them is the one that the verification information can represent the overall shape of the full dataset, which is given by $G(\boldsymbol{W}, \mathcal{X})$, this loss function will give us the optimal estimate $\boldsymbol{W}^*$. However, it cannot be optimized since most items in $\mathcal{X}$ are not verified. So we define the loss error caused by the verification subset $\mathcal{L}$ as $G(\hat{\boldsymbol{W}}^{\mathcal{L}}, \mathcal{X}) - G(\boldsymbol{W}^*, \mathcal{X})$, and we define the best verification subset as the one that minimizes this loss error.

Now we compute the expectation of this loss error $\mathbb{E}[G(\hat{\boldsymbol{W}}^{\mathcal{L}}, \mathcal{X}) - G(\boldsymbol{W}^*, \mathcal{X})]$. This value relies on the Hessian of the negative likelihood function $-\partial^2 \log p(y|\boldsymbol{x}, \boldsymbol{W})/\partial \boldsymbol{W}^2$. Since our estimator is a generalized linear model (GLM), it's easy to show that its Hessian is only a function of $\boldsymbol{x}$ and $\boldsymbol{W}$, but not related to y. We use $I(\boldsymbol{x}, \boldsymbol{W})$ to denote this value. When the second-moment of the covariance matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ exists and is positive definite, the expectation of $I(\boldsymbol{x}, \boldsymbol{W})$ is strictly convex with respect to $\boldsymbol{W}$ (Reverdy and Leonard 2016). With Lemma 1 of Chaudhuri et al. (Chaudhuri et al. 2015), we give the following proposition about the expected loss error.

Proposition 1. *Define* $I_{\mathcal{L}}(\boldsymbol{W}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{L}}[I(\boldsymbol{x}, \boldsymbol{W})]$ *and* $I_{\mathcal{X}}(\boldsymbol{W}) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}[I(\boldsymbol{x}, \boldsymbol{W})]$*, the expected loss error of the semi-supervised estimate* $\mathbb{E}[G(\hat{\boldsymbol{W}}^{\mathcal{L}}, \mathcal{X}) - G(\boldsymbol{W}^*, \mathcal{X})]$ *with respect to* B *ground truths sampled from* $\mathcal{L}$ *is upper bounded by* $\mathcal{O}\Big(\big(1 + \frac{\lambda B}{N}\big) \operatorname{tr}\Big(\big(I_{\mathcal{X}}(\boldsymbol{W}^*) + \frac{\lambda B}{N} I_{\mathcal{L}}(\boldsymbol{W}^*)\big)^{-1} I_{\mathcal{X}}(\boldsymbol{W}^*)\Big)\Big)$.

Here $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. For clarity, we put the proof in the appendix. When B is fixed, the above proposition states that the expected loss error is actually controlled by $\operatorname{tr}\Big(\big(I_{\mathcal{X}}(\boldsymbol{W}^*) + \frac{\lambda B}{N} I_{\mathcal{L}}(\boldsymbol{W}^*)\big)^{-1} I_{\mathcal{X}}(\boldsymbol{W}^*)\Big)$. So we can find the $\mathcal{L}$ that minimizes this criterion to make the semi-supervised learning estimate more precise. This gives us a selective verification strategy for crowdsourcing. The algorithm to achieve this goal will be discussed in the next section.

Remark 1. *If* $B = 0$*, the semi-supervised setting reduces to the unsupervised setting without verification. For fairness, we reserve the term with* B *in the one-sample loss of the unsupervised setting, so the expected loss error for this*

setting is $\mathcal{O}(1+(\lambda B/N))$. Proposition 1 implies that introducing verification can reduce the expected loss error. To see this, we denote $\boldsymbol{A}=I_{\mathcal{X}}(\boldsymbol{W}^*)+\frac{\lambda B}{N}I_{\mathcal{L}}(\boldsymbol{W}^*)$. We have $\mathrm{tr}(\boldsymbol{A}^{-1}I_{\mathcal{X}}(\boldsymbol{W}^*))=\mathrm{tr}(I)-\frac{\lambda B}{N}\mathrm{tr}(\boldsymbol{A}^{-1/2}I_{\mathcal{L}}(\boldsymbol{W}^*)\boldsymbol{A}^{-1/2})$. Since both $\boldsymbol{A}$ and $I_{\mathcal{L}}(\boldsymbol{W}^*)$ are positive definite matrices, $\mathrm{tr}(\boldsymbol{A}^{-1/2}I_{\mathcal{L}}(\boldsymbol{W}^*)\boldsymbol{A}^{-1/2})>0$. Then we have $(1+(\lambda B/N))\mathrm{tr}(\boldsymbol{A}^{-1}I_{\mathcal{X}}(\boldsymbol{W}^*))<1+(\lambda B/N)$.

The Algorithm

We have presented our transductive semi-supervised learning model for crowdsourcing and the corresponding expected loss error, which can be used as a subset selecting criterion. In this section, we discuss the algorithm for minimizing this criterion as well as the overall learning pipeline.

Error Bound Minimization (EBM) Strategy

Now we introduce our error bound minimization (EBM) strategy. To select a best subset of tasks for querying the truths, we optimize $\mathcal{L}$ to minimize the value $\mathrm{tr}((NI_{\mathcal{X}}(\boldsymbol{W}^*)+\lambda BI_{\mathcal{L}}(\boldsymbol{W}^*))^{-1}I_{\mathcal{X}}(\boldsymbol{W}^*))$, which controls the expected loss error. To make this minimization problem tractable to solve, we first introduce two implementation details. Since the true parameter $\boldsymbol{W}^*$ is unknown, according to the assumption we made when deriving the strategy, we use $\hat{\boldsymbol{W}}^{Unsup}$ instead. During analysis, we assume that the covariance matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ is positive definite. However, in some situation, the features composed by the sparse crowdsourced labels may not be positive definite, so the error does not exist. To deal with this, we first use the principal component analysis (PCA) to find the eigenvalues of the original covariance matrix and project the features into a new space, then we remove the dimensions that correspond to zero eigenvalues. So the reduced covariance matrix of the remained features is positive definite. We compute the necessary statistical values using these new features.

To represent $\mathcal{L}$, we use a distribution characterized by parameters $\{a_i : i\in[N]\}$, where Ba_i denotes the probability of that task i is in $\mathcal{L}$. These parameters are defined on the $(N-1)$-simplex, which is a convex set $\Delta^{N-1}=\{\sum_{i\in[N]}a_i=1,\ 0\le a_i\le 1 : i\in[N]\}$. To learn the parameters $\boldsymbol{a}$, we define the following objective function L:

$$L=\left(NI_{\mathcal{X}}(\boldsymbol{W}^*)+\lambda B\sum_{i\in[N]}[a_iI(\boldsymbol{x}_i,\boldsymbol{W}^*)]\right)^{-1}I_{\mathcal{X}}(\boldsymbol{W}^*),$$

and we solve the following optimization problem:

$$\min_{\boldsymbol{a}\in\Delta^{N-1}}\mathrm{tr}\,(L). \tag{4}$$

This problem can be solved by the gradient descent method on the probability simplex (Byrne and Girolami 2014). Specifically, the gradient with respect to a_i is $\lambda\mathrm{tr}\,(I(\boldsymbol{x}_i,\boldsymbol{W}^*)\boldsymbol{M})$, where $\boldsymbol{M}=\boldsymbol{V}I_{\mathcal{X}}(\boldsymbol{W}^*)\boldsymbol{V}$ and $\boldsymbol{V}=(NI_{\mathcal{X}}(\boldsymbol{W}^*)+\lambda B\sum_{i\in[N]}[a_iI(\boldsymbol{x}_i,\boldsymbol{W}^*)])^{-1}$. For each update, we first project this gradient into the subspace that contains the probability simplex. Then if the next state exceeds the simplex boundaries, we reflect it back to the feasible region. After converging, the optimal distribution is given by $\hat{\boldsymbol{a}}$. Then we select the B tasks that correspond to the largest $\hat{a}_i$ values to give the verification subset $\mathcal{L}$.

Remark 2. *The goal of our paper is to propose a selective verification strategy for crowdsourcing, which can work robustly on most noise types. Since complex models rely on complex assumptions and thus will fail if the assumptions are invalid, we derived our strategy from the DS model which makes minimum assumptions and thus more robust. Moreover, almost all state-of-the-art label aggregation methods are partly based on the DS model, so it's possible to derive exact model-specific strategies based on similar techniques.*

Although the EBM strategy is derived from the DS model, it can be independently used on any aggregation model. For example, we can use any unsupervised model to estimate an initial $\boldsymbol{y}$, then select a verification subset with the EBM strategy, and finally, update the results by any (semi-)supervised aggregation model. Our experiments with the EM-based semi-supervised DS model will empirically demonstrate the effectiveness of the EBM strategy in different settings.

Learning Pipeline

With the above techniques, we are ready to present the complete pipeline of learning from crowds with selective verification. At the beginning, we query the crowds about the tasks we care and collect the responses $\boldsymbol{X}$. Then we run EM algorithm for the unsupervised DS estimator in problem (1) to learn the parameter estimate $\hat{\boldsymbol{W}}^{Unsup}$. After that, if the covariance matrix of the features is not positive definite, we reduce the feature dimension by PCA, which removes the dimensions that correspond to zero eigenvalues. Then to learn the best verification subset $\mathcal{L}$, the expected loss error given the semi-supervised estimate is characterized by Eq. (4). We minimize this error with the gradient descent method on the probability simplex to find the optimal parameter $\hat{\boldsymbol{a}}$, and select the subset $\mathcal{L}$ based on them. Finally, after querying the oracle, we learn the crowdsourcing parameter $\hat{\boldsymbol{W}}$ by solving problem (3), which is a softmax regression. The estimated truths for the unlabeled tasks are given by Eq. (2).

Usually, we suggest selecting all the verification tasks at the same epoch, so that these tasks can be distributed to a group of experts through crowdsourcing. However, the EBM strategy can also be extended to multi-epochs mode. Specifically, after the t-th epoch of verification finished, we can use the current estimate $\hat{\boldsymbol{W}}_t$ to select the verification tasks for the next epoch. This mode could be more precise than the single-epoch mode with the same total number of verification tasks since the estimates become closer to the $\boldsymbol{W}^*$ after several epochs. But since the tasks are not paralleled, it could be less efficient.

Empirical Results

We conduct experiments on both synthetic and real-world datasets to evaluate the benefits of oracle verification, as well as to show the efficacy of our selecting strategy.

Synthetic Datasets

We generate synthetic datasets with $N=1,000$ tasks, which are averagely assigned with $D=5$ kinds of ground truths.

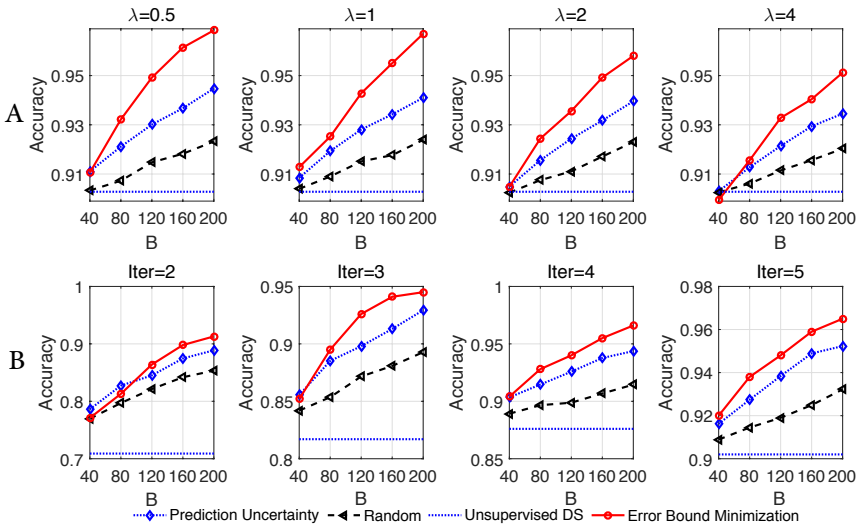

Figure 2: Prediction accuracies of different strategies on synthetic datasets. The horizontal axis denotes the number of tasks for verification, while the vertical axis denotes the prediction accuracy. Row A compares results of strategies with different λ. Row B compares the results of strategies induced by $\hat{\boldsymbol{W}}^{Unsup}$ with different qualities. The accuracy axes are clipped for clarity.

$M = 25$ workers are simulated with random confusion matrices. When sampling each worker's confusion matrix, for each simulated worker, we first pick a random matrix with elements sampled from $[0, 1]$. Then we add it with a $0.1 * \boldsymbol{I}$ diagonal matrix to simulate the superiority of truths. Finally, we normalize it as the confusion matrix. This trick simulates the common assumption that workers always have a larger chance to give correct answers than wrong answers. Then crowdsourced labels are generated according to the ground truths and the confusion matrices.

For all experiments, we use the semi-supervised learning method proposed in Eq. (3), which is general and simple. We compare the **Error Bound Minimization (EBM)** selective verification strategy proposed in above sections with two simple and intuitive methods. **Random (RAN)**: This strategy randomly selects B tasks with equal chances for verification. **Prediction Uncertainty (PU)**: The uncertainty based methods are popular for active learning (Hung et al. 2015; Khattak and Salleb-Aouissi 2016), here we compute the uncertainties based on the predicted distributions on $\boldsymbol{y}$ given by the unsupervised DS model. The information entropy of each distribution is used to represent the importance of a task. We also compare the semi-supervised methods with the basic **unsupervised Dawid-Skene (uDS)** method.

Performance Comparison. We conducted a series of experiments with different verification subset sizes B and balance hyperparameters λ. Specifically, we vary B in the range $\{40,\ 80,\ 120,\ 160,\ 200\}$, which is enough to show the changing trends of the performance. Each λ is selected in $\{0.5,\ 1,\ 2,\ 4\}$. For each setting, the average results on 5 randomly generated datasets are shown in the first row of Fig. 2. Since the crowdsourced labels are generated following the DS model, the unsupervised method can achieve a good average accuracy of 0.9012. Meanwhile, after introducing the verification, the performance further increases. For all selecting strategies, a larger verification subset size B leads to a higher prediction accuracy. The figures show that the EBM strategy can achieve higher accuracies than the uncertainty method and the random method in most cases. We conduct pairwise t-test on all the performance pairs between EBM and the other two strategies, results show that the difference is significant with $p < 5 \times 10^{-8}$. We think an important reason to explain this superiority is that EBM considers the semi-supervised property of the learning framework.

Different sub-figures in the first row of Fig. 2 plot the performance with different λ. They show that in the tested range, the prediction accuracies slightly decrease when λ grows. Apparently, when λ reduce to 0, the semi-supervised methods degenerated to the unsupervised version, whose prediction accuracy is relatively low. So we can find an optimal λ for each dataset by validating. Since the changes are not remarkable, we just fix $\lambda = 1$ for following experiments.

We notice that the prediction accuracy of EBM is still lower than the *perfect* strategy, which only selects verification tasks that are predicted incorrectly by the unsupervised model. But since it's almost impossible to achieve this perfect performance, the consistent superiority over the baselines is sufficient to show the significance of this method. Moreover, in many tasks that are sensitive to the data quality, such as medical tasks, even one percent of improvement is important for the applications.

Approximation Analysis. During the loss error analysis, we use the unsupervised estimate $\hat{\boldsymbol{W}}^{Unsup}$ to approximate the optimal parameter $\boldsymbol{W}^*$, which may influence the selecting performance of the strategy. Here we empirically show the effect. To test the approximation for $\boldsymbol{W}^*$ with different qualities, we fix a synthetic dataset, and run an unsupervised DS model with training iterations (Iter) varies from 2 to 5, then each time we can get a different estimate $\hat{\boldsymbol{W}}^{Unsup}$ with different qualities, and the corresponding unsupervised pre-

Table 1: Prediction accuracy (%) on three real-world datasets.

	BLUEBIRDS				AGES				WEBSEARCH			
UDS	88.9				66.7				81.5			
ME	91.7				68.9				88.9			
CROWDSVM	89.6				67.0				92.0			
SEMI-SUPERVISED METHODS	B	RAN	PU	EBM	B	RAN	PU	EBM	B	RAN	PU	EBM
	5	89.0	89.8	**90.7**	50	68.4	**71.6**	71.2	100	87.3	**89.0**	87.4
	10	88.5	90.7	90.7	100	70.1	73.2	**74.8**	200	89.0	**91.3**	91.1
	15	89.4	93.4	**94.4**	150	71.6	76.1	**77.0**	300	90.0	93.1	**94.0**
	20	89.7	92.6	**93.5**	200	73.7	77.8	**80.4**	400	91.2	94.4	**95.1**
	25	91.0	94.4	**97.2**	250	75.6	78.2	**82.6**	500	92.0	94.4	**95.4**

Table 2: Prediction accuracies (%) with EM-based semi-supervised learning methods (Tang and Lease 2011).

BLUEBIRDS				AGES				WEBSEARCH			
B	RAN-EM	PU-EM	EBM-EM	B	RAN-EM	PU-EM	EBM-EM	B	RAN-EM	PU-EM	EBM-EM
5	88.7	88.0	**88.9**	50	68.4	68.8	**69.0**	100	82.4	83.6	**84.2**
10	89.4	89.8	89.8	100	70.3	71.2	**72.0**	200	83.4	84.8	**85.7**
15	90.2	90.7	**91.7**	150	71.6	72.4	**74.8**	300	84.6	86.2	**87.1**
20	91.1	91.7	**92.6**	200	73.7	73.8	**77.1**	400	85.2	87.6	**88.4**
25	91.6	93.5	93.5	250	75.0	74.4	**79.1**	500	86.4	88.9	**89.3**

diction accuracy varies from 0.709 to 0.902. We test the selecting strategies with different $\hat{\boldsymbol{W}}^{Unsup}$, and the prediction accuracies from different settings are plotted in the second row of Fig. 2. It shows that better $\hat{\boldsymbol{W}}^{Unsup}$ leads to a better prediction accuracy. But even with a relatively bad $\hat{\boldsymbol{W}}^{Unsup}$ that the theoretical results do not hold, EBM selecting strategy is still better than the uncertainty and random strategies in most cases. This result demonstrates the efficacy of the proposed algorithm.

Real-World Datasets

We also conduct experiments on three widely used real-world datasets: **Bluebirds** (Welinder et al. 2010): There are 2 breeds among 108 bluebird pictures, and each image is labeled by all 39 workers. 4,214 labels are collected in total. **Ages** (Han and Jain 2014): 165 workers are asked to estimate the ages for 1,002 face images. The final estimates are discretized into 7 bins, and the dataset consists of 10,020 labels in total. **Web Search** (Zhou et al. 2012): 15,567 responses are collected on the relevance rating for 2,665 query-URL pairs. 177 workers were involved and each response scales from 1 to 5. We include another two state-of-the-art unsupervised models during comparing, the **Minimax Entropy (ME)** (Zhou et al. 2012) and the **CrwodSVM** (Tian and Zhu 2015a).

Results presented in Tab. 1 show that for all datasets and strategies, more oracle verification tasks lead to better overall accuracy, which is as we expected. The semi-supervised methods are better than the state-of-the-art unsupervised methods by given a reasonably small amount of verification labels. In most cases, the EBM selecting strategy is better than baselines, and when selecting more verification tasks, the benefits become more significant. We think this is because, with a small subset, the value of oracle verification mainly focuses on individual tasks. However, a large verification subset can construct a summary for all tasks, so all unlabeled tasks share the benefits. Thus EBM can show its advantages.

As we stated, the selective verification strategy can be independently used on other semi-supervised models. So we compare the proposed strategies on the EM-based semi-supervised model (Tang and Lease 2011). Results are shown in Tab. 2. The performance of EM-based model is usually lower than the softmax-based method. It's possibly due to the local optima issue of the EM algorithm. The accuracies achieved by EBM is better than those of other strategies in most cases. It again demonstrates the benefits of our proposal.

Conclusions

We propose an active semi-supervised learning framework for crowdsourcing with verification. We establish the expected loss error and propose the EBM selective verification strategy. The empirical results demonstrate that the oracle verification can help to improve the label aggregation performance, and the EBM strategy is better than baseline strategies on several datasets. In the future, this method can be extended to crowdsourcing for more complex data types. Improvements on unsupervised learning from crowds methods can also be combined into our proposal.

Acknowledgments

This work is supported by the National NSF of China (Nos. 61620106010, 61621136008, 61332007), the MIIT Grant of

Int. Man. Comp. Stan (No. 2016ZXFB00001) and a Grant from Siemens.

References

Byrne, S., and Girolami, M. 2014. Geodesic monte carlo on embedded manifolds. *Scandinavian Journal of Statistics Theory & Applications* 40(4):825.

Chaudhuri, K.; Kakade, S. M.; Netrapalli, P.; and Sanghavi, S. 2015. Convergence rates of active learning for maximum likelihood estimation. In *NIPS*.

Dawid, A. P., and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*.

Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Li, K.; and Li, F. F. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Gu, Q.; Zhang, T.; Han, J.; and Ding, C. H. 2012. Selective labeling via error bound minimization. In *NIPS*.

Gu, Q.; Zhang, T.; and Han, J. 2014. Batch-mode active learning via error bound minimization. In *UAI*.

Han, H., O. C. L. X., and Jain, A. 2014. Demographic estimation from face images: Human vs. machine performance. *IEEE Trans. on PAMI*.

Hu, Q.; He, Q.; Huang, H.; Chiew, K.; and Liu, Z. 2014. Learning from crowds under experts supervision. In *PAKDD*.

Hung, N. Q. V.; Tam, N. T.; Lam, N. T.; and Aberer, K. 2013. An evaluation of aggregation techniques in crowdsourcing. In *WISE (2)*, 1–15.

Hung, N. Q. V.; Thang, D. C.; Weidlich, M.; and Aberer, K. 2015. Minimizing efforts in validating crowd answers. In *SIGMOD*.

Jagabathula, S.; Subramanian, L.; and Venkataraman, A. 2014. Reputation-based worker filtering in crowdsourcing. In *NIPS*.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *ICML*.

Kajino, H.; Tsuboi, Y.; Sato, I.; and Kashima, H. 2012. Learning from crowds and experts. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Khattak, F. K., and Salleb-Aouissi, A. 2016. Toward a robust crowd-labeling framework using expert evaluation and pairwise comparison. *Journal of Artificial Intelligence Research*.

Kingma, D. P.; Mohamed, S.; Rezende, D. J.; and Welling, M. 2014. Semi-supervised learning with deep generative models. In *NIPS*.

Li, H., and Yu, B. 2014. Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086*.

Liu, Q.; Peng, J.; and Ihler, A. 2013. Variational inference for crowdsourcing. In *NIPS*.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning from crowds. *JMLR*.

Reverdy, P., and Leonard, N. E. 2016. Parameter estimation in softmax decision-making models with linear objective functions. *IEEE Transactions on Automation Science and Engineering* 13(1):54–67.

Settles, B. 2010. Active learning literature survey. *University of Wisconsin, Madison*.

Shah, N. B., and Zhou, D. 2015. Double or nothing: Multiplicative incentive mechanisms for crowdsourcing. In *NIPS*.

Sheshadri, A., and Lease, M. 2013. Square: A benchmark for research on computing crowd consensus. In *HCOMP*.

Snow, R.; O'Connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*.

Tang, W., and Lease, M. 2011. Semi-supervised consensus labeling for crowdsourcing. In *Special Interest Group on Information Retrieval 2011 Workshop on Crowdsourcing for Information Retrieval*.

Tian, T., and Zhu, J. 2015a. Max-margin majority voting for learning from crowds. In *NIPS*.

Tian, T., and Zhu, J. 2015b. Uncovering the latent structures of crowd labeling. In *PAKDD*.

Venanzi, M.; Guiver, J.; Kazai, G.; Kohli, P.; and Shokouhi, M. 2014. Community-based bayesian aggregation models for crowdsourcing. In *WWW*.

Wang, L., and Zhou, Z.-H. 2016. Cost-saving effect of crowdsourcing learning. In *IJCAI*.

Wang, Y.; Yu, A. W.; and Singh, A. 2016. On computationally tractable selection of experiments in regression models. In *arXiv preprint*.

Wauthier, F., and Jordan, M. 2011. Bayesian bias mitigation for crowdsourcing. *NIPS*.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The multidimensional wisdom of crowds. In *NIPS*.

Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. R. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*.

Yang, Y., and Loog, M. 2016. A benchmark and comparison of active learning for logistic regression. *arXiv preprint arXiv:1611.08618*.

Zhang, Y.; Chen, X.; Zhou, D.; and Jordan, M. I. 2016. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *JMLR*.

Zheng, Y.; Li, G.; Li, Y.; Shan, C.; and Cheng, R. 2017. Truth inference in crowdsourcing: Is the problem solved? In *VLDB*.

Zhou, D.; Basu, S.; Mao, Y.; and Platt, J. C. 2012. Learning from the wisdom of crowds by minimax entropy. In *NIPS*.

Zhu, X., and Goldberg, A. B. 2009. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*.