Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

A Convex Formulation for Learning from Crowds*

Hiroshi Kajino
Department of Mathematical Informatics
The University of Tokyo

Yuta Tsuboi
IBM Research - Tokyo

Hisashi Kashima
Department of Mathematical Informatics
The University of Tokyo

Abstract

Recently crowdsourcing services are often used to collect a large amount of labeled data for machine learning, since they provide us an easy way to get labels at very low cost and in a short period. The use of crowdsourcing has introduced a new challenge in machine learning, that is, coping with the variable quality of crowd-generated data. Although there have been many recent attempts to address the quality problem of multiple workers, only a few of the existing methods consider the problem of learning classifiers directly from such noisy data. All these methods modeled the true labels as latent variables, which resulted in non-convex optimization problems. In this paper, we propose a convex optimization formulation for learning from crowds without estimating the true labels by introducing personal models of the individual crowd workers. We also devise an efficient iterative method for solving the convex optimization problems by exploiting conditional independence structures in multiple classifiers. We evaluate the proposed method against three competing methods on synthetic data sets and a real crowdsourced data set and demonstrate that the proposed method outperforms the other three methods.

Introduction

Machine learning approaches have been the majority in various areas such as natural language processing, computer vision, and speech recognition. To reduce the time and financial costs to collect a large amount of labeled data for applying machine learning methods, it is becoming increasingly popular to use crowdsourcing services such as the *Amazon Mechanical Turk*[1] (AMT), which makes it easy to ask the general public to work on relatively simple tasks at very low cost through the Internet. For example, Snow et al. (2008) validated the use of crowdsourcing services for natural language processing tasks by using the AMT to collect annotations by non-experts.

However, a new problem has been introduced in machine learning by using crowdsourcing, the quality control problem for crowd workers. The quality of the data obtained from crowd workers is often much lower than that of data collected in the ordinary controlled environments. In crowdsourcing services, some workers are highly skilled and provide high quality data, while some are unskilled and often give almost random responses. In the worst case, a *spammer* intentionally produces random data to earn easy money. Welinder and Perona (2010) pointed out the existence of sloppy workers in their data set whose labels did not contain any information, and Snow et al. (2008) also reported that the labels given by some workers were almost random.

One promising solution to deal with such noisy workers is *repeated labeling* (Sheng, Provost, and Ipeirotis 2008). This involves obtaining multiple labels to each instance from multiple workers and estimating the true labels from the noisy labels. This solution is widely used in the context of learning from crowds. The existing approaches followed this approach and modeled various parameters such as the ability of workers and the difficulty of instances (Whitehill et al. 2009; Welinder et al. 2010; Yan et al. 2011). Since the goal of supervised learning is not only to estimate true labels for training data but to obtain predictive models for future data, Raykar et al. (2010) proposed a method which *jointly* estimates both the true labels and a classifier. Yan et al. (2011) also used such a policy and introduced active learning framework. The existing work about learning from crowds is summarized in Table 1.

These methods addressed the label uncertainty problem in various ways, but there is a serious problem of non-convexity in the prior approaches. Since most of the existing methods model the true labels as latent variables, the resulting optimization problems are not convex. Therefore most of them use EM-style inference methods, and there is no guarantee of obtaining optimal solutions. In fact, these methods depend on their initial states, which sometimes causes poor performance with high variance.

In this paper, we propose a new approach to address this problem. Instead of introducing latent variables to estimate the true labels, which is the source of the non-convexity, we introduce a personal classifier for each of the workers, and estimate the base classifier by relating it to the personal models. This model naturally takes account into the ability of each worker and the instance difficulty for each worker, and this idea leads to a convex optimization problem, Further-

*The preliminary version of this paper appeared in Japanese (Kajino and Kashima 2012).

[1] https://www.mturk.com/mturk/welcome

Table 1: Comparison between existing methods and our method. The proposed method is the first to formulate learning from crowds as a convex optimization problem. All methods modeled the variable ability of workers (worker ability), and some methods also modeled the variable difficulty of instances (instance difficulty).

	inference	worker ability	instance difficulty	convex formulation	comment
Dawid and Skene (1979)	label	✓			
Sheng, Provost, and Ipeirotis (2008)	label	✓			active learning
Whitehill et al. (2009)	label	✓	✓		
Donmez, Carbonell, and Schneider (2009)	label	✓	✓		active learning
Welinder et al. (2010)	label	✓	✓		
Welinder and Perona (2010)	label	✓			online learning
Raykar et al. (2010)	label & model	✓			
Yan et al. (2010)	label & model	✓	✓		semi-supervised
Yan et al. (2011)	label & model	✓	✓		active learning
Proposed	model	✓	✓	✓	

more, we exploited the problem structure to devise an efficient iterative optimization algorithm. Finally, we perform experiments using several data sets including a real data set and demonstrate the advantage of the proposed model over the model of Raykar et al. (2010), which is one of the state-of-the-art models.

Although the resulting convex optimization problem is similar to that for the multi-task learning model proposed by Evgeniou and Pontil (2004), our model differs from the model of Evgeniou and Pontil (2004) in two ways. One is that our model gives a new interpretation to the centroid of the parameters of multiple tasks, which doesn't have an obvious meaning in the model of Evgeniou and Pontil. Another is that the main goals of our method and the method of Evgeniou and Pontil are different. In a multi-task context, the main goal is to estimate a parameter for each task, while the main goal in a crowdsourcing context is to get the centroid, which is interpreted as the parameter of the base classifier.

In summary, this work makes two main contributions: (i) We formulated the learning-from-crowds problem as a convex optimization problem by introducing personal models, and (ii) we devised an efficient iterative method for solving the convex optimization problem by exploiting the conditional independence relationships among the models.

Learning from Crowds

We first define the problem of learning from crowds. For simplicity, we focus on a binary classification problem in this paper. However, the proposed approach can be directly applied to more general cases, including multi-class classification and regression problems.

The problem of learning from crowds is defined as a generalized case of supervised learning in that we have multiple noisy labels for each instance. Let us assume that we have N problem instances $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^D$ is a D-dimensional real-valued feature vector, and that there are J workers who can give labels to the instances via crowdsourcing. Let $I_j \subseteq \{1,\dots,N\}$ be the index set of instances that the j-th worker gives labels. Let $y_{ij} \in \{0,1\}$ be a noisy label that the j-th worker gives to the i-th instance $\mathbf{x}_i$, let $\mathcal{Y}_j = \{y_{ij} \mid i \in I_j\}$ be the set of labels given by the j-th worker, and let $\mathcal{Y} = \bigcup_{j=1}^J \mathcal{Y}_j$ be the set of all labels acquired by using crowdsourcing.

Our goal is to estimate a binary classifier $f : \mathbb{R}^D \to \{0,1\}$ given $(\mathcal{X}, \mathcal{Y})$ as a training data set.

A Convex Formulation for Learning from Crowds

We first model the labeling processes of the multiple workers by introducing personal classifiers, which leads to a convex optimization problem. We also propose an efficient iterative optimization algorithm by exploiting the conditional independence structure in multiple classifiers.

Labeling Process Using Personal Classifiers

The basic idea to avoid a non-convex optimization problem is to bring in a personal classifier for each worker instead of using latent variables to represent the true labels.

Let us represent the base model as the logistic regression model parameterized by $\mathbf{w}_0$,

$$\Pr[y = 1 \mid \mathbf{x}, \mathbf{w}_0] = \sigma(\mathbf{w}_0^\top \mathbf{x}) = (1 + \exp(-\mathbf{w}_0^\top \mathbf{x}))^{-1},$$

where σ denotes the sigmoid function. We also model the labeling process of each worker j as a logistic regression model parameterized by $\mathbf{w}_j$ as

$$\Pr[y_j = 1 \mid \mathbf{x}, \mathbf{w}_j] = \sigma(\mathbf{w}_j^\top \mathbf{x}) \ \ (j \in \{1,\dots,J\}).$$

Our assumption is that the personal models are related with the base model. Therefore, we associate the base model and the personal models with the relation

$$\mathbf{w}_j = \mathbf{w}_0 + \mathbf{v}_j \quad (j \in \{1,\dots,J\}), \tag{1}$$

where $\mathbf{v}_j$ models the differences in ability and characteristics of the individuals.

One may notice that this model looks very similar to the multi-task learning model proposed by Evgeniou and Pontil (2004) if we consider each worker as a task. However, it is the objective of each method that distinguishes our model from their model. The goal of multi-task learning is to obtain

the model parameters $\{\mathbf{w}_j\}_{j=1}^J$ for different tasks (which are the personal models in our context), while our goal is to obtain the base model with $\mathbf{w}_0$. This point clearly distinguishes the semantics of the models.

Convex Objective Function

To solve the estimation problem of the model parameters $\{\mathbf{w}_j\}_{j=0}^J$ as statistical inference, we consider a generative process.

We assume that the parameter of the base classifier is generated from a prior distribution $\Pr[\mathbf{w}_0]$, and the parameter of the j-th worker is generated from $\Pr[\mathbf{w}_j \mid \mathbf{w}_0]$. Specifically, we define them as Gaussian distributions,

$$\Pr[\mathbf{w}_0 \mid \eta] = \mathcal{N}(\mathbf{0}, \eta^{-1}\mathbf{I}),$$
$$\Pr[\mathbf{w}_j \mid \mathbf{w}_0, \lambda] = \mathcal{N}(\mathbf{w}_0, \lambda^{-1}\mathbf{I}),$$

where η and λ are positive constants. After $\{\mathbf{w}_j\}_{j=1}^J$ are determined, each label y_{ij} is generated from $\Pr[y_{ij} \mid \mathbf{x}_i, \mathbf{w}_j]$.

By denoting $\mathbf{W} = \{\mathbf{w}_j \mid j \in \{1,\ldots,J\}\}$, the posterior distribution of $\mathbf{w}_0$ and $\mathbf{W}$ given the training data $(\mathcal{X}, \mathcal{Y})$ can be written as

$$\Pr[\mathbf{W}, \mathbf{w}_0 \mid \mathcal{X}, \mathcal{Y}, \eta, \lambda]$$
$$\propto \Pr[\mathcal{Y} \mid \mathbf{W}, \mathcal{X}] \Pr[\mathbf{W} \mid \mathbf{w}_0, \lambda] \Pr[\mathbf{w}_0 \mid \eta].$$

Let $F(\mathbf{w}_0, \mathbf{W})$ be the negative log-posterior distribution of $\mathbf{w}_0$ and $\mathbf{W}$ omitting the constants, which is written as

$$F(\mathbf{w}_0, \mathbf{W}) = -\sum_{j=1}^{J} \sum_{i \in I_j} l(y_{ij}, \sigma(\mathbf{w}_j^\top \mathbf{x}_i)) + \frac{\lambda}{2} \sum_{j=1}^{J} \|\mathbf{w}_j - \mathbf{w}_0\|^2 + \frac{1}{2}\eta \|\mathbf{w}_0\|^2$$

where $l(s,t) = s \log t + (1-s)\log(1-t)$. Note that the objective function $F(\mathbf{w}_0, \mathbf{W})$ is convex. Therefore, the maximum-a-posteriori (MAP) estimators of $\mathbf{W}$ and $\mathbf{w}_0$ are obtained by solving an optimization problem:

$$\text{minimize } F(\mathbf{w}_0, \mathbf{W}) \text{ w.r.t. } \mathbf{w}_0 \text{ and } \mathbf{W}.$$

Algorithm

Noticing the conditional independence relationships among the model parameters $\{\mathbf{w}_j\}_{j=0}^J$, we can devise the following alternating optimization algorithm, where we repeat the two optimization steps, one with respect to $\mathbf{w}_0$ and the other with respect to $\{\mathbf{w}_j\}_{j=1}^J$, until convergence.

Step 1. Optimization w.r.t. $\mathbf{w}_0$

Given $\{\mathbf{w}_j\}_{j=1}^J$ fixed, the optimal $\mathbf{w}_0$ is easily obtained as a closed form solution:

$$\mathbf{w}_0^\star = \frac{\lambda \sum_{j=1}^J \mathbf{w}_j}{\eta + J\lambda}.$$

Step 2. Optimization w.r.t. $\mathbf{W}$

Given $\mathbf{w}_0$ fixed, the parameters $\{\mathbf{w}_j\}_{j=1}^J$ are independent of each other. Therefore, we can work on the independent optimization problem for a particular $j \in \{1,\ldots,J\}$. This implies that a large problem can be decomposed into relatively small problems, which leads to an efficient algorithm. To solve each optimization problem, we can use any numerical optimization method. In our implementation, we employ the Newton-Raphson update,

$$\mathbf{w}_j^{\text{new}} = \mathbf{w}_j^{\text{old}} - \alpha \cdot \mathbf{H}^{-1}(\mathbf{w}_j^{\text{old}}) \mathbf{g}(\mathbf{w}_j^{\text{old}}, \mathbf{w}_0),$$

where $\alpha > 0$ is the step length, and the gradient $\mathbf{g}(\mathbf{w}_j, \mathbf{w}_0)$ and the Hessian $\mathbf{H}(\mathbf{w}_j)$ are given as

$$\mathbf{g}(\mathbf{w}_j, \mathbf{w}_0) = -\left(\sum_{i \in I_j} (y_{ij} - \sigma(\mathbf{w}_j^\top \mathbf{x}_i))\mathbf{x}_i\right) + \lambda(\mathbf{w}_j - \mathbf{w}_0),$$

$$\mathbf{H}(\mathbf{w}_j) = \left[\sum_{i \in I_j} (1 - \sigma(\mathbf{w}_j^\top \mathbf{x}_i))\sigma(\mathbf{w}_j^\top \mathbf{x}_i) x_{ik} x_{il}\right]_{k,l} + \lambda \mathbf{I}_d,$$

where x_{ik} represents the k-th elements of $\mathbf{x}_i$, and $[a_{k,l}]_{k,l}$ is a $D \times D$ matrix with the (k,l)-element equal to $a_{k,l}$.

Experiments

We conducted three types of experiments to assess the proposed method using a synthetic data set without spammers, a synthetic data set with spammers, a benchmark data set, and a real data set called *microblogging message data set*, and we demonstrate the advantage of the proposed method over the existing method of Raykar et al. (2010) and two other baseline methods.

Competing Methods

Baseline Methods. First we introduce two baseline methods. One is called the Majority Voting method that uses a majority voting strategy to estimate the true labels. The other is called the All-in-One-Classifier method that abandons all of the worker IDs and merges all of the acquired labels into one classifier.

- Majority Voting Method (MV method)
  This is a typical heuristic in the context of learning from crowds. Given labels $\{y_{ij}\}_{j=1}^J$ for an instance $\mathbf{x}_i$, the true label y_i for the instance $\mathbf{x}_i$ is estimated using majority voting as

$$y_i = \begin{cases} 1 & \text{if } \sum_{j=1}^J y_{ij} > J/2, \\ 0 & \text{if } \sum_{j=1}^J y_{ij} < J/2, \\ \text{random} & \text{otherwise.} \end{cases}$$

- All-in-One-Classifier Method (AOC method)
  This method is also a popular heuristic that considers $(\mathcal{X}, \mathcal{Y})$ as training data for one classifier, i.e., we forget the worker IDs and use all labels to learn one classifier.

Latent Class Model (LC model). We review the method proposed by Raykar et al. (2010) as one of the state-of-the-art methods, calling it the latent class model in this paper. Similar to our model, they also assume a logistic regression model for the classification model as

$$\Pr[y_i = 1 \mid \mathbf{x}_i, \mathbf{w}_0] = \sigma(\mathbf{w}_0^\top \mathbf{x}_i).$$

To model the labeling process of each worker, they introduce the two-coin model

$$\begin{cases} \alpha_j &= \Pr[y_{ij} = 1 \mid y_i = 1], \\ \beta_j &= \Pr[y_{ij} = 0 \mid y_i = 0]. \end{cases} \quad (2)$$

If the true label is 1, the j-th worker gives the true label 1 with probability α_j and 0 with probability $1-\alpha_j$. If the true label is 0, the j-th worker gives the true label 0 with probability β_j and 1 with probability $1-\beta_j$.

By using the EM algorithm[2], we obtain an approximation of the maximum-likelihood estimators of model parameters.

Data Sets

We set up four types of data sets, synthetic data sets with and without spammers, a benchmark data set, and a microblogging message data set. The former three data sets are simulated data sets and the last data set is a real data set. For the simulated data sets, we evaluated the predictive performance with the average and the standard deviation of the AUCs over 100 runs and the number of instances each worker gives was fixed as $|I_j| = N$, i.e., each worker gave labels to all instances. For the real data set, we evaluated the predictive performance with the precision, the recall, and the F-measure.

Synthetic Data Sets without Spammers. We tested two sets of synthetic data; one was generated with the proposed model, and the other with the LC model.

(i) Data generation with the proposed model (Our data)
The parameter of the j-th worker $\mathbf{w}_j$ was generated from $\Pr[\mathbf{w}_j \mid \mathbf{w}_0, \lambda]$. We assigned the label $y_{ij} = 1$ with probability $\Pr[y_{ij} = 1 \mid \mathbf{x}_i, \mathbf{w}_j]$ and $y_{ij} = 0$ with probability $\Pr[y_{ij} = 0 \mid \mathbf{x}_i, \mathbf{w}_j]$.

(ii) Data generation with the LC model (Raykar's data)
For each instance $\mathbf{x}_i \in \mathcal{X}$, if $\Pr[y_i = 1 \mid \mathbf{x}_i, \mathbf{w}_0] \geq \gamma$, we assigned the true label $y_i = 1$ to $\mathbf{x}_i$, else $y_i = 0$ to $\mathbf{x}_i$, where γ was a positive constant. Then we flipped the true label by using the rule of Eq. (2) to simulate the j-th worker's label.

Instances $\{\mathbf{x}_i \in \mathbb{R}^2\}_{i=1}^N$ were sampled from the uniform distribution $\mathcal{U}([-20,20] \times [-20,20])$. We set the model parameters $\alpha_j = \beta_j = 0.55$ ($j \in \{1,\ldots,J\}$), $\mathbf{w}_0 = [1,0]^\top, \sigma =$ 1and $\lambda = 10^{-3}$ to generate the data sets. The initial values were generated from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. We varied the number of instances N from 10 to 50 by 10s, and the number of workers J from 10 to 100 by 10s. We calculated the AUC by generating 100,000 test instances.

Synthetic Data Sets with Spammers. In real-world situations, some workers act as *spammers*, who simply seek to make money with little effort. It is important to exclude the effect of such noisy labels because learning from noisy data can contaminate the results. We simulated a data set that included labels by spammers, especially *random workers* who gave random labels. Instances $\{\mathbf{x}_i \in \mathbb{R}^2\}_{i=1}^N$ were sampled from the uniform distribution $\mathcal{U}([-20,20] \times [-20,20])$ and labels were generated from the LC model. We assumed two types of workers: skilled workers and spammers. We set the ability of the skilled workers to be $\alpha_j = \beta_j = 0.85$ and that of the spammers to be $\alpha_j = \beta_j = 0.5$. We varied the ratio of the number of skilled workers to the total number of workers from 0 to 1 by steps of 0.1 (where the ratio is $r =$ (#skilled workers)/(#workers)).

Benchmark Data Set. We used a benchmark data set, "Wine Quality" (Cortez et al. 2009), from the UCI Machine Learning Repository (Frank and Asuncion 2010). We simulated multiple noisy workers by following the latent class model, because the UCI Machine Learning Repository had no data set with multiple workers. We used the data of red wine in the "Wine Quality" data set, which had 4,898 instances with 11-dimensional feature vectors. The multi-class labels of this data set were binarized to be used as training labels. The data were randomly divided into a training set (70%) and a test set (30%) in each run. We fixed the number of instances N and varied the number of workers J and the ability of the workers $\{\alpha_j\}_{j=1}^J, \{\beta_j\}_{j=1}^J$.

Microblogging Message Data Set. We used a data set for a *Named Entity Recognition* (NER) task, which deals with the identification of the names of persons, organizations, locations, and similar entities in sentences. Finin et al. (2010) created a Twitter data set[3] where each token in tweets (texts) was labeled by workers of the AMT, and we used this data as a training and test set. Unlike standard data sets for NER, the segment boundary of each entity was not given in the data set. Therefore we simply considered the task as a binary classification problem to identify whether each token was in a named entity ($y_i = 1$) or not ($y_i = 0$). The number of the labeled tokens was 212,720, and each token was labeled by two workers. The data set had 269 workers in total. We constructed a training set, which contained 120,968 instances and 135 workers. There were also gold standard labels which contained 8,107 instances, and we used them as a test set. We omitted the named entity labels for the @usernames[4] in the same way as the paper of Ritter, Clark, and Etzioni (2011), because it was too easy to identify them. The feature representation for each token was the same as that for the named entity segmentation of tweets in the previous work (Ritter, Clark, and Etzioni 2011). To reduce the number of the model parameters, we selected the features that appeared more than once in the training set, and we obtained 161,903-dimensional feature vectors. We evaluated the performance of classifiers by calculating the precision, recall, and F-measure on the test set.

Results

The averages of the AUCs are summarized in Figs. 1–4 and the averages and the standard deviations of the AUCs in Table 2 for all of the data sets. We experimentally show that

[2]We added a regularization term for $\mathbf{w}_0$ to avoid over-fitting in our experiments.

[3]The data set is available at http://sites.google.com/site/amtworkshop2010/data-1

[4]The @ symbol followed by their unique username is used to refer to other users.

the proposed method and the AOC method outperformed the other two methods with homogeneous workers in **Synthetic Data Sets without Spammers**, the proposed method outperformed the other three methods with heterogeneous workers in **Synthetic Data Sets with Spammers** and **Microblogging Message Data Set**, and all the methods performed almost equally in **Benchmark Data Set**.

Synthetic Data Sets without Spammers. We plot the averages of the AUCs on our, Raykar's, and the two baseline methods in Figs. 1 and 2 and the averages and the standard deviations for specific parameters are extracted in Table 2. For almost all values of J and N, the AUCs of the proposed method and the AOC method were higher than those of the MV method and the LC model. Interestingly, even on synthetic data generated by the LC model (Raykar's Data), our method and the AOC method were better than the LC model. The advantage of our method and the AOC method were also seen in Table 2, where the standard deviations of these methods were smaller than those of the other methods in the experiments on Raykar's data. These facts suggest that the convexity of the objective function contributed to the high average of the AUCs, because there is no dependency on initial conditions and the probability that a poor classifier is obtained is eliminated. However, there was almost no difference between the results of the proposed method and the AOC method in this setting. This is because all of the workers are homogeneous in their ability. In the next experiment with spammers, these two methods behaved differently.

Synthetic Data Sets with Spammers. We plot the averages of the AUCs on our, Raykar's, and the two baseline methods in Fig. 3. In this setting, the proposed method outperformed the other three methods in both the averages and the standard deviations. In particular, when the number of experts was small ($r < 0.4$), the difference between the proposed method and the AOC method became large. In addition, looking at the results of the LC model in Table 2, its AUC increases and the standard deviation decreases rapidly as r increases. This suggests an important property of the LC model in that it works well as long as there are a certain number of skilled workers.

Benchmark Data Set. We also summarize the result of the experiments on the benchmark data set in Fig. 4. For almost all values of α_j, β_j, and J, the performance of the four methods was almost the same. This suggests that if the number of instances is large, the classifier can be estimated well with any method.

Microblogging Message Data Set. The results of experiments on this data set are summarized in Table 3. For each model, we chose the result of the highest F-measure for each method. This table clearly shows that our model is superior to the other methods in the F-measure and the recall.

Table 3: Precision, Recall, and F-measure comparisons on the Microblogging Message Data Set

	Precision	Recall	F-measure
Our Model	0.651	**0.766**	**0.704**
Raykar's Model	**0.761**	0.553	0.640
AOC Model	0.571	0.700	0.629
MV Model	0.575	0.701	0.632

Related Work

Research on learning from multiple noisy workers without true labels started in the context of aggregating diagnoses by multiple doctors to make more accurate decisions. In the seminal work by Dawid and Skene (1979), they modeled the ability and bias of the workers, and used the EM algorithm to estimate the true labels considered as latent variables. Most of the existing research in this field followed their model and extended it in various ways. For example, Smyth et al. (1995) dealt with the problem of inferring true labels from subjective labels of Venus images.

Around 2005, the appearance of crowdsourcing services based on the Web such as the AMT kindled research interests in this field. Sheng, Provost, and Ipeirotis (2008) showed that accuracy of classifiers could be improved by using the repeated labeling technique and the repeated labeling required lower costs compared to the single labeling even if the cost of acquiring labels was expensive.

However they made a strong assumption that all of the workers were of the same ability. To address the problem, some extended the method proposed by Dawid and Skene (1979) to model the ability of workers and the difficulty of instances. Raykar et al. (2010) modeled the ability of each worker, and Whitehill et al. (2009) and Welinder et al. (2010) also modeled the difficulty of each instance. Moreover, recent efforts introduced many other aspects into machine learning such as cost-sensitive learning. Sheng, Provost, and Ipeirotis (2008), Donmez, Carbonell, and Schneider (2009), and Yan et al. (2011) used active learning approaches to this problem, and Welinder and Perona (2010) used online learning approach. These approaches are summarized in Table 1.

An inference target is one of the important aspects which categorizes the existing research into two groups: the one aiming to infer the true labels and the other aiming to infer mainly predictive models. Most of the existing methods are categorized into the former group, while the methods of Raykar et al. (2010), Yan et al. (2011) and our proposed method are categorized into the latter group. Raykar et al. (2010) and Yan et al. (2011) modeled a classifier as a model parameter and the unobserved true labels as latent variables and inferred them using the EM algorithm, while we infer *only* the models without estimating the true labels, which enables us to formulate the problem as a convex optimization problem. Also the method proposed by Dekel and Shamir (2009) aimed to infer only predictive models. In the method, low-quality workers and labels provided by them were excluded to improve the quality of classifiers.

The purpose of using crowdsourcing is not limited to the construction of labeled data sets. Recently, crowdsourcing has been used to realize *human computation*, which is a new paradigm to unify machine-power and man-power to solve

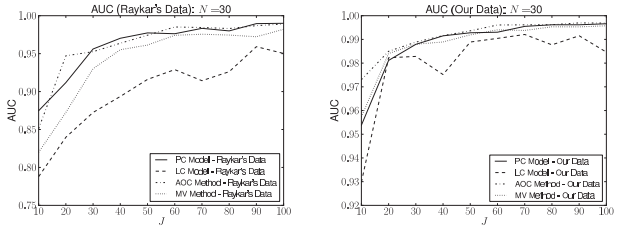

Figure 1: AUC comparisons on the synthetic data without spammers. Given constant $N = 30$, we varied J from 10 to 100 by steps of 10. The horizontal axis corresponds to J, and the vertical axis to the AUC. **(Left)** The results for the data set generated with the LC model. **(Right)** The results for the data set generated with our model.

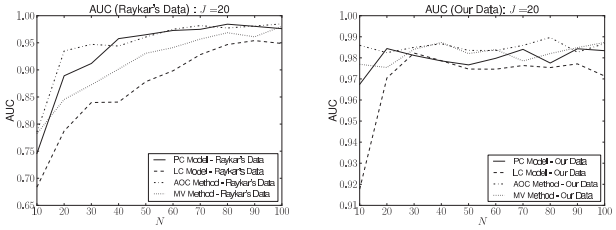

Figure 2: AUC comparisons on the synthetic data without spammers. Given constant $J = 20$, we varied N from 10 to 50 by steps of 10. The horizontal axis corresponds to N, and the vertical axis to the AUC. **(Left)** The results for the data set generated with the LC model. **(Right)** The results for the data set generated by our model.

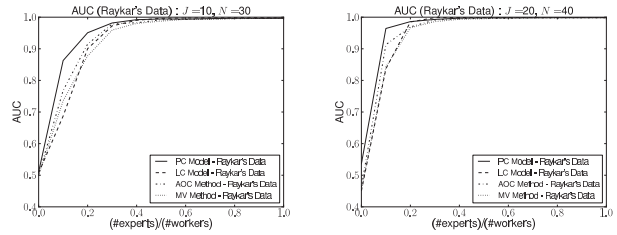

Figure 3: AUC comparisons on synthetic data with spammers. We varied r from 0 to 1 by steps of 0.1. The horizontal axis corresponds to r, and the vertical axis to the AUC. **(Left)** The results given constants $J = 10$ and $N = 30$. **(Right)** The results given constants $J = 20$ and $N = 40$.

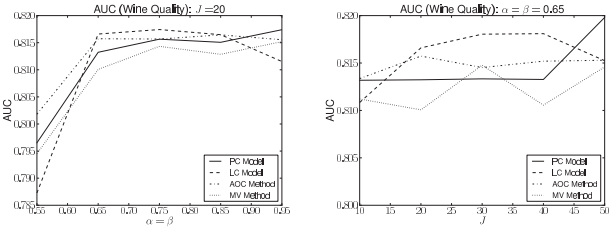

Figure 4: AUC comparisons on "Wine Quality". The horizontal axis corresponds to $\alpha_j = \beta_j$, and the vertical axis to the AUC. **(Left)** The results with constant $J = 20$, and we varied $\{\alpha_j\}_{j=1}^{J}$ and $\{\beta_j\}_{j=1}^{J}$ from 0.55 to 0.95 by steps of 0.1. **(Right)** The results with constant $\alpha_j = \beta_j = 0.65$, and we varied J from 10 to 50 by steps of 10.

difficult computational problems. For examples, Tamuz et al. (2011) used crowdsourcing to construct kernel functions, and Gomes et al. (2011) performed clustering via crowdsourcing. Both approaches used crowdsourcing to identify the similarity between two objects, which is a relative easy task for human beings but difficult for machines.

Multi-task learning is a learning task for simultaneously estimating multiple predictive models from multiple related tasks. The ideas of multi-task learning date back to the middle '90s. One of the most representative studies in the early stages of multi-task learning is the study by Caruana (1997), which proposed to share a hidden layer of artificial neural networks among multiple tasks. There is much existing research on multi-task learning, so we refrain from listing them, but the most relevant work to ours is that by Evgeniou and Pontil (2004). Their formulation is very similar to our formulation (Eq. (1)). However, the proposed model is distinguished clearly from their model in that they have totally different objectives. Learning from crowds aims to estimate the parameter for the base model, while multi-task learning aims to estimate the parameters for different T tasks.

Conclusions

In this paper, we proposed a new approach to deal with multiple noisy workers. The proposed method is formulated as a convex optimization problem by introducing personal classifiers. Experiments on the synthetic data set with and without spammers, the benchmark data set, and the real data set demonstrated that our approach showed the same or better performance than that of the existing method of Raykar et al. (2010) and the two baseline methods.

Acknowledgment

H. Kajino and H. Kashima were supported by the FIRST program.

References

Caruana, R. 1997. Multitask Learning. *Machine Learning*.

Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; and Reis, J. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems* 47(4):547–553.

Dawid, A. P., and Skene, A. M. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Al-

Table 2: AUC Comparisons along with Standard Deviations.

Data Set	Parameters	Our Model	LC Model	AOC Method	MV Method
Our Data	$J = 20, N = 30$	0.981 ± 0.035	0.982 ± 0.032	$\mathbf{0.985 \pm 0.020}$	0.984 ± 0.025
Our Data	$J = 20, N = 40$	0.979 ± 0.042	0.979 ± 0.043	$0.987 \pm \mathbf{0.016}$	$\mathbf{0.988} \pm 0.023$
Raykar's Data without spammers	$J = 20, N = 30$	0.911 ± 0.146	0.840 ± 0.235	$\mathbf{0.947 \pm 0.093}$	0.873 ± 0.196
Raykar's Data without spammers	$J = 20, N = 40$	$\mathbf{0.958 \pm 0.073}$	0.841 ± 0.235	0.944 ± 0.116	0.901 ± 0.153
Raykar's Data with spammers	$J = 10, N = 30$ $r = 0.3$	$\mathbf{0.982 \pm 0.028}$	0.972 ± 0.138	0.978 ± 0.048	0.959 ± 0.079
Raykar's Data with spammers	$J = 20, N = 40$ $r = 0.1$	$\mathbf{0.964 \pm 0.048}$	0.837 ± 0.319	0.913 ± 0.152	0.843 ± 0.209
Wine Data	$J = 20$, $\alpha_j = \beta_j = 0.55$	$0.797 \pm \mathbf{0.017}$	0.787 ± 0.023	$\mathbf{0.802} \pm 0.020$	0.795 ± 0.022

gorithm. *Journal of the Royal Statistical Society. Series C (Applied Statics)* 28(1):20–28.

Dekel, O., and Shamir, O. 2009. Vox Populi: Collecting High-Quality Labels from a Crowd. In *Proceedings of the 22nd Annual Conference on Learning Theory*.

Donmez, P.; Carbonell, J. G.; and Schneider, J. 2009. Efficiently Learning the Accuracy of Labeling Sources for Selective Sampling. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 259–268.

Evgeniou, T., and Pontil, M. 2004. Regularized multi–task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 109–117.

Finin, T.; Murnane, W.; Karandikar, A.; Keller, N.; Martineau, J.; and Dredze, M. 2010. Annotating Named Entities in Twitter Data with Crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 80–88.

Frank, A., and Asuncion, A. 2010. UCI Machine Learning Repository.

Gomes, R.; Welinder, P.; Krause, A.; and Perona, P. 2011. Crowdclustering. In *Advances in Neural Information Processing 24*, 558–566.

Kajino, H., and Kashima, H. 2012. Convex Formulations of Learning from Crowds. *Journal of Japanese Society of Artificial Intelligence* 27(3):133–142.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning From Crowds. *Journal of Machine Learning Research* 11:1297–1322.

Ritter, A.; Clark, S.; and Etzioni, O. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1524–1534.

Sheng, V. S.; Provost, F.; and Ipeirotis, P. G. 2008. Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 614–622.

Smyth, P.; Fayyad, U.; Burl, M.; Perona, P.; and Baldi, P. 1995. Inferring Ground Truth from Subjective Labelling of Venus Images. In *Advances in Neural Information Processing Systems 7*, 1085–1092.

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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 254–263.

Tamuz, O.; Liu, C.; Belongie, S.; Shamir, O.; and Kalai, A. T. 2011. Adaptively Learning the Crowd Kernel. In *Proceedings of the 28th International Conference on Machine Learning*, 673–680.

Welinder, P., and Perona, P. 2010. Online crowdsourcing: Rating annotators and obtaining cost-effective labels. In *Workshop on Advancing Computer Vision with Humans in the Loop, IEEE Conference on Computer Vision and Pattern Recognition*, 25–32.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The Multidimensional Wisdom of Crowds. In *Advances in Neural Information Processing Systems 23*, 2424–2432.

Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. 2009. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Advances in Neural Information Processing Systems 22*, 2035–2043.

Yan, Y.; Rosales, R.; Fung, G.; and Dy, J. 2010. Modeling Multiple Annotator Expertise in the Semi-Supervised Learning Scenario. In *Proceedings of Conference on Uncertainty in Artificial Intelligence 2010*, 674–682.

Yan, Y.; Rosales, R.; Fung, G.; and Dy, J. G. 2011. Active Learning from Crowds. In *Proceedings of the 28th International Conference on Machine Learning*, 1161–1168.