Human Computation
AAAI Technical Report WS-12-08

Learning from Crowds and Experts

Hiroshi Kajino
Department of Mathematical Informatics
The University of Tokyo

Yuta Tsuboi
IBM Research - Tokyo

Issei Sato
Information Technology Center
The University of Tokyo

Hisashi Kashima
Department of Mathematical Informatics
The University of Tokyo

Abstract

Crowdsourcing services are often used to collect a large amount of labeled data for machine learning. Although they provide us an easy way to get labels at very low cost in a short period, they have serious limitations. One of them is the variable quality of the crowd-generated data. There have been many attempts to increase the reliability of crowd-generated data and the quality of classifiers obtained from such data. However, in these problem settings, relatively few researchers have tried using expert-generated data to achieve further improvements. In this paper, we extend three models that deal with the problem of learning from crowds to utilize ground truths: a latent class model, a personal classifier model, and a data-dependent error model. We evaluate the proposed methods against two baseline methods on a real data set to demonstrate the effectiveness of combining crowd-generated data and expert-generated data.

Introduction

Machine learning approaches have become the majority in various areas, however, it is cumbersome to collect a large amount of labeled data for the training data sets. To solve this problem, increasing attention has been given to ways to collect labeled data via crowdsourcing services, such as the *Amazon Mechanical Turk*[1] (AMT). Indeed, many approaches have tried to utilize crowdsourcing to gather labels in areas such as natural language processing (Snow et al. 2008; Finin et al. 2010), computer vision (Whitehill et al. 2009; Welinder and Perona 2010; Welinder et al. 2010), and machine learning. Using crowdsourcing to collect labeled data has both advantages and disadvantages. It is advantageous that we can reduce the time and financial costs because crowdsourcing services access a large amount of manpower at low cost, which allows us to construct large data sets. One of the disadvantages, which is often pointed out, is the quality control problem for crowd workers. The quality of the data obtained from crowd workers varies from person to person. To make full use of crowdsourcing, we have to solve the problem of learning from noisy workers.

In the field of machine learning, there are two primary goals. One is to estimate the ground truth labels and the other is to learn a classifier directly from noisy data. The problem of estimating ground truth from noisy labels is a relatively well-studied problem. The most successful approach is repeated labeling (Sheng, Provost, and Ipeirotis 2008). This approach improves the quality of the labels by querying labels from multiple workers for each instance and aggregating the labels to estimate the ground truths. To aggregate multiple labels, majority voting and a wide variety of EM-style estimation strategies have been used. The EM-style estimation strategies set the true labels as latent variables in a model and estimate them using the EM algorithms, an idea that dates back to the 1970s (Dawid and Skene 1979).

Recently, the problem of learning a classifier directly from crowd-generated data has appeared (Dekel and Shamir 2009; Raykar et al. 2010; Yan et al. 2010; Wauthier and Jordan 2011; Kajino, Tsuboi, and Kashima 2012). The approach of Raykar et al. (2010) and Yan et al. (2010) constructs a classifier from estimated ground truth labels, while the method of Dekel and Shamir (2009), Wauthier and Jordan (2011), and Kajino, Tsuboi, and Kashima (2012) estimates classifiers directly from the labels obtained from noisy workers. It is important to study this problem because in machine learning what we need is a high-quality classifier rather than a perfect training data set.

These approaches can address the problems of learning from crowds, but it is still problematic that little existing work combines crowd-generated data and expert-generated data, thus exploiting reliable labels obtained from experts who are known to be highly skilled. Combining crowd-generated data and expert-generated data is expected to improve the quality of a classifier more than using only crowd-generated data. In fact, there are many existing data sets that have ground truths and it is a natural idea to combine them

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

with crowd-generated data to improve the quality of the data sets and classifiers. To the best of our knowledge, only the work by Tang and Lease (2011) and the work by Wauthier and Jordan (2011) considered this situation.

In this paper, we extend the personal classifier model proposed by Kajino, Tsuboi, and Kashima (2012), the latent class model proposed by Raykar et al. (2010), and the data dependent error model proposed by Yan et al. (2010) to utilize expert-generated data. The personal classifier model (Kajino, Tsuboi, and Kashima 2012) assigns a personal classifier for each worker and these classifiers are assumed to be generated by perturbing the base classifier (here, the base classifier has no training data). We add training data to the base classifier and implement an efficient algorithm based on the algorithm of the original personal classifier model. The latent class model (Raykar et al. 2010) is a model that sets the true labels as latent variables and jointly estimates the true labels and a classifier using an EM-style algorithm. We add an expert worker to the model as a special worker to utilize expert-generated data. The data dependent error model (Yan et al. 2010) is one variant of the latent class model. The difference between these models is that the data dependent error model assumes that the ability of workers changes depending on instances. In the same way as the latent class model, we add an expert worker as a special worker to the model. We compare three proposed methods and two baseline methods on a real crowdsourced data set. The experiments show that using expert-generated data can improve the quality of classifiers.

In summary, our contributions are twofold: (i) we extend the personal classifier model, the latent class model, and the data dependent error model to combine expert-generated data and crowd-generated data, and (ii) we demonstrate the effectiveness of using expert-generated data on a real crowdsourced data set and gain insight to the proposed methods.

Problem Settings

We first define the problem considered in this paper. Let us assume that there are 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 each instance i has the ground truth label y_i, which is generally not observed in the problem setting of learning from crowds. We also assume that there are J workers who can give noisy labels to the instances via crowdsourcing. For the j-th worker ($j \in \{1, \dots, J\}$ is a worker ID), let $\mathcal{I}_j \subseteq \{1, \dots, N\}$ be an index set of instances that the j-th worker has labeled, let $y_{i,j} \in \{0, 1\}$ be a noisy label that the j-th worker gave to the i-th instance $\mathbf{x}_i$, let $\mathcal{Y}_j = \{y_{i,j} \mid i \in \mathcal{I}_j\}$ be a 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 of the labels acquired by using crowdsourcing. Similarly, let $\mathcal{J}_i \subseteq \{1, \dots, J\}$ be an index set of workers who gave labels to the i-th instance, and let $\mathcal{Y}_i = \{y_{i,j} \mid j \in \mathcal{J}_i\}$ be a set of labels assigned to the i-th instance.

In this paper, we additionally assume that some instances have much more reliable labels than the crowd-generated labels, called *expert-generated labels*. The expert-generated labels are given by an expert worker who are known to be of high ability. In this paper, we assume that an expert always gives ground truth labels. Assuming that the expert has a special worker ID $j = 0$, let $\mathcal{I}_0 \subseteq \{1, \dots, N\}$ be an index set of instances that the expert labeled, let $y_{i,0}$ $(= y_i)$ be a label that the expert gave to the i-th instance, and let $\mathcal{Y}_0 = \{y_{i,0} \mid i \in \mathcal{I}_0\}$ be the set of expert-generated labels.

Our goal is to estimate a binary classifier $f : \mathbb{R}^D \to \{0, 1\}$ given $(\mathcal{X}, \mathcal{Y}, \mathcal{Y}_0)$ as a training set. For simplicity, we focus on the binary classification problem in this paper. However, the proposed approaches can be directly applied to more general cases, including multi-class classification and regression problems.

Proposed Methods

We first extend the personal classifier model (Kajino, Tsuboi, and Kashima 2012). We then extend the latent class model (Raykar et al. 2009), and the data dependent error model (Yan et al. 2010). For each model, we describe a model of the labeling process and a parameter estimation algorithm. The significance of the proposed methods is that we take into account two types of workers: workers whose ability is not known (crowd workers) and workers whose ability is given beforehand (experts). Most of the existing methods consider only the workers whose ability is not known.

Personal Classifier Model (PC Model)

We propose a method to combine crowd-generated data and expert-generated data based on the personal classifier model (Kajino, Tsuboi, and Kashima 2012).

Labeling Process. We follow the idea of the personal classifier model (Kajino, Tsuboi, and Kashima 2012) and introduce personal classifiers for workers. Let us represent the base classifier as a logistic regression model parameterized by $\mathbf{w}_0$,

$$p(y_i = 1 \mid \mathbf{x}_i, \mathbf{w}_0) = \sigma(\mathbf{w}_0^\top \mathbf{x}_i),$$

where $\sigma(t) = (1 + e^{-t})^{-1}$ denotes the sigmoid function. We assume that the ground truth labels $\mathcal{Y}_0$ are generated from $p(y_i \mid \mathbf{x}_i, \mathbf{w}_0)$. This assumption distinguishes the proposed model from the model of Kajino, Tsuboi, and Kashima (2012). We also model the labeling process of each worker $j \in \{1, \dots, J\}$ as a logistic regression model parameterized by $\mathbf{w}_j$,

$$p(y_{i,j} = 1 \mid \mathbf{x}_i, \mathbf{w}_j) = \sigma(\mathbf{w}_j^\top \mathbf{x}_i).$$

Then we relate the parameters $\{\mathbf{w}_j\}_{j=1}^J$ and $\mathbf{w}_0$. We assume that the parameters $\{\mathbf{w}_j\}_{j=1}^J$ are generated from $p(\mathbf{w}_j \mid \mathbf{w}_0, \lambda)$ and the parameter $\mathbf{w}_0$ is generated from $p(\mathbf{w}_0 \mid \eta)$ (λ and η are hyperparameters). Specifically, we define them as Gaussian distributions,

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

where η and λ are positive constants.

Parameter Estimation. We estimate the model parameters by maximizing their posterior distribution. 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 $(\mathcal{X}, \mathcal{Y}, \mathcal{Y}_0)$ is written as

$$\begin{aligned} &p(\mathbf{W}, \mathbf{w}_0 \mid \mathcal{X}, \mathcal{Y}, \mathcal{Y}_0, \eta, \lambda) \\ &\propto p(\mathcal{Y} \mid \mathbf{W}, \mathcal{X}) p(\mathcal{Y}_0 \mid \mathbf{w}_0, \mathcal{X}) p(\mathbf{W} \mid \mathbf{w}_0, \lambda) p(\mathbf{w}_0 \mid \eta). \end{aligned}$$

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

$$\begin{aligned} F(\mathbf{w}_0, \mathbf{W}) = &- \sum_{j=0}^{J} \sum_{i \in \mathcal{I}_j} l(y_{i,j}, \sigma(\mathbf{w}_j^\top \mathbf{x}_i)) \\ &+ \frac{\lambda}{2} \sum_{j=1}^{J} \|\mathbf{w}_j - \mathbf{w}_0\|^2 + \frac{1}{2} \eta \|\mathbf{w}_0\|^2, \end{aligned}$$

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

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

This is a convex optimization problem.

Algorithm. Noticing the conditional independence relationships among the model parameters $\{\mathbf{w}_j\}_{j=0}^J$, we can divide the optimization problem into subproblems and devised the following alternating optimization algorithm, in which 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 $\mathbf{w}_0$ can be obtained by applying any numerical optimization method. In our implementation, we use the Newton-Raphson update,

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

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

$$\begin{aligned} \mathbf{g}_0(\mathbf{w}_0, \mathbf{W}) = &- \left(\sum_{i \in \mathcal{I}_0} (y_{i,0} - \sigma(\mathbf{w}_0^\top \mathbf{x}_i)) \mathbf{x}_i \right) \\ &+ \lambda \sum_{j=1}^{J} (\mathbf{w}_0 - \mathbf{w}_j) + \eta \mathbf{w}_0, \text{ and} \\ \mathbf{H}_0(\mathbf{w}_0) = &\left[\sum_{i \in \mathcal{I}_0} (1 - \sigma(\mathbf{w}_0^\top \mathbf{x}_i)) \sigma(\mathbf{w}_0^\top \mathbf{x}_i) x_{ik} x_{il} \right]_{k,l} \\ &+ (\eta + J\lambda) \mathbf{I}_d, \end{aligned}$$

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

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

Given $\mathbf{w}_0$ fixed, the $\{\mathbf{w}_j\}_{j=1}^J$ are independent of each other. Therefore, we can work on an independent optimization problem for each $j \in \{1, \ldots, J\}$. Again, 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 a step length, and the gradient $\mathbf{g}(\mathbf{w}_j, \mathbf{w}_0)$ and the Hessian $\mathbf{H}(\mathbf{w}_j)$ are given as

$$\begin{aligned} &\mathbf{g}(\mathbf{w}_j, \mathbf{w}_0) \\ &= - \left(\sum_{i \in \mathcal{I}_j} (y_{i,j} - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)) \mathbf{x}_i \right) + \lambda(\mathbf{w}_j - \mathbf{w}_0), \text{ and} \\ &\mathbf{H}(\mathbf{w}_j) \\ &= \left[\sum_{i \in \mathcal{I}_j} (1 - \sigma(\mathbf{w}_j^\top \mathbf{x}_i)) \sigma(\mathbf{w}_j^\top \mathbf{x}_i) x_{ik} x_{il} \right]_{k,l} + \lambda \mathbf{I}_d, \end{aligned}$$

respectively.

Latent Class Model (LC Model)

We extend the latent class model proposed by Raykar et al. (2010) based on the idea of Tang and Lease (2011). The difference between our model and the model proposed by Tang and Lease is the existence of feature vectors.

Labeling Process. Similar to the PC model, Raykar et al. also assume a logistic regression model for the classification model as

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

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

$$\begin{cases} \alpha_j &= p(y_{i,j} = 1 \mid y_i = 1), \\ \beta_j &= p(y_{i,j} = 0 \mid y_i = 0). \end{cases} \tag{1}$$

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 $1 - \beta_j$. In our extension, we assume that the true labels $\mathcal{Y}_0$ are given by an expert ($j = 0$) who never makes mistakes, i.e., $\alpha_0 = \beta_0 = 1$.

Parameter Estimation. We estimate the parameters by maximizing the likelihood function. The likelihood function is written as

$$p(\mathcal{Y}, \mathcal{Y}_0 \mid \mathcal{X}, \theta) = \prod_{i=1}^{N} [a_i p_i + b_i (1 - p_i)],$$

where $\theta = \{\mathbf{w}_0, \{\alpha_j\}_{j=1}^J, \{\beta_j\}_{j=1}^J\}$ are model parameters, and let $p_i = \sigma(\mathbf{w}_0^\top \mathbf{x}_i), a_i = \prod_{j \in \mathcal{J}_i} \alpha_j^{y_{i,j}} (1 - \alpha_j)^{1 - y_{i,j}}, b_i = \prod_{j \in \mathcal{J}_i} \beta_j^{1 - y_{i,j}} (1 - \beta_j)^{y_{i,j}}$.

Algorithm. Approximations of the maximum-likelihood estimators of model parameters θ are obtained by using the EM algorithm, where the following E-step and M-step are repeated until convergence:

E-step. Update $\mu_i = p(y_i = 1 \mid \mathcal{Y}_i, \mathbf{x}_i, \theta)$ using

$$\mu_i = \frac{a_i p_i}{a_i p_i + b_i (1 - p_i)}.$$

Note that $\mu_i = y_{i,0}$ holds for all $i \in \mathcal{I}_0$ because we set $\alpha_0 = \beta_0 = 1$.

M-step. Update $\{\alpha_j\}_{j=1}^J$ and $\{\beta_j\}_{j=1}^J$ using

$$\alpha_j = \frac{\sum_{i \in \mathcal{I}_j} \mu_i y_{i,j}}{\sum_{i \in \mathcal{I}_j} \mu_i}, \beta_j = \frac{\sum_{i \in \mathcal{I}_j} (1-\mu_i)(1-y_{i,j})}{\sum_{i \in \mathcal{I}_j} (1-\mu_i)},$$

and update $\mathbf{w}_0$ by maximizing the lower bound of the log-likelihood function using the Newton-Raphson update. Note that we don't update α_0 and β_0 in the M-step.

For initialization, we use the majority voting estimation $\mu_i = \frac{1}{|\mathcal{J}_i|} \sum_{j \in \mathcal{J}_i} y_{i,j}$ following the method of Raykar et al. (2010).

Data Dependent Error Model (DDE Model)

We extend the model proposed by Yan et al. (2010) in a similar way to the extension of the latent class model.

Labeling Process. Similar to the previous models, Yan et al. also assume a logistic regression model for the classification model as

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

To model the labeling process of each worker, they introduce a two-coin model that additionally considers the difficulty of the instances for each worker $j \in \{1, \ldots, J\}$:

$$\begin{aligned} &p(y_{i,j} \mid \mathbf{x}_i, y_i, \mathbf{u}_j) \\ &= \left(1 - \sigma(\mathbf{u}_j^\top \mathbf{x}_i)\right)^{|y_{i,j} - y_i|} \cdot \sigma(\mathbf{u}_j^\top \mathbf{x}_i)^{1-|y_{i,j}-y_i|}. \end{aligned}$$

This means that the j-th worker gives the true label y_i to an instance $\mathbf{x}_i$ with probability $\sigma(\mathbf{u}_j^\top \mathbf{x}_i)$ and gives a flipped label $1 - y_i$ with probability $1 - \sigma(\mathbf{u}_j^\top \mathbf{x}_i)$. In our extension, we assume that the worker 0 is an expert who labels as

$$p(y_{i,0} \mid \mathbf{x}_i, y_i, \mathbf{u}_0) = 1 - |y_{i,0} - y_i|.$$

Parameter Estimation. We estimate the model parameters by maximizing the likelihood function. The likelihood function is written as

$$\begin{aligned} &p(\mathcal{Y}, \mathcal{Y}_0 \mid \mathcal{X}, \theta) \\ &= \prod_{j=0}^{J} \prod_{i \in \mathcal{I}_j} \sum_{y_i} p(y_{i,j} \mid \mathbf{x}_i, y_i, \mathbf{u}_j) p(y_i \mid \mathbf{x}_i, \mathbf{w}_0). \end{aligned}$$

Algorithm. Approximations of the maximum-likelihood estimators of model parameters $\theta = \{\mathbf{w}_0, \{\mathbf{u}_j\}_{j=1}^J\}$ are obtained by using the EM algorithm, where the following E-step and M-step are repeated until convergence:

E-step. Update $\mu_i = p(y_i = 1 \mid \mathcal{Y}_i, \mathbf{x}_i, \theta)$ for $i \notin \mathcal{I}_0$ using

$$\begin{aligned} \mu_i \propto& \sigma(\mathbf{w}_0^\top \mathbf{x}_i) \prod_{j \in \mathcal{J}_i} (1 - \sigma(\mathbf{u}_j^\top \mathbf{x}_i))^{1-y_{i,j}} \sigma(\mathbf{u}_j^\top \mathbf{x}_i)^{y_{i,j}}, \\ 1 - \mu_i \propto& (1 - \sigma(\mathbf{w}_0^\top \mathbf{x}_i)) \\ & \prod_{j \in \mathcal{J}_i} (1 - \sigma(\mathbf{u}_j^\top \mathbf{x}_i))^{y_{i,j}} \sigma(\mathbf{u}_j^\top \mathbf{x}_i)^{1-y_{i,j}}, \end{aligned}$$

and update μ_i using $\mu_i = y_{i,0}$ for $i \in \mathcal{I}_0$.

M-step. Update θ by maximizing the objective function $L(\theta)$ using the L-BFGS quasi-Newton method based on the algorithm proposed by Yan et al. (2010). The objective function, which is a conditional expectation, is written as

$$\begin{aligned} L(\theta) =& \sum_{j=1}^{J} \sum_{i \in \mathcal{I}_j} \mathbb{E}_{p(y_i \mid \mathbf{x}_i, \mathcal{Y}_i)} [\log p(y_{i,j}, y_i \mid \mathbf{x}_i)] \\ =& \sum_{i=1}^{N} (\log(1 - \sigma(\mathbf{w}_0^\top \mathbf{x}_i)) + \mu_i \mathbf{w}_0^\top \mathbf{x}_i) \\ &+ \sum_{j=1}^{J} \sum_{i \in \mathcal{I}_j} \big[y_{i,j} (\log(1 - \sigma(\mathbf{u}_j^\top \mathbf{x}_i)) + \mu_i \mathbf{u}_j^\top \mathbf{x}_i) \\ &\qquad + (1 - y_{i,j})(\log \sigma(\mathbf{u}_j^\top \mathbf{x}_i) - \mu_i \mathbf{u}_j^\top \mathbf{x}_i) \big], \end{aligned}$$

and its gradients with respect to $\mathbf{w}_0$ and $\mathbf{u}_j$ are given as

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}_0} =& \sum_{i=1}^{N} (\mu_i - \sigma(\mathbf{w}_0^\top \mathbf{x}_i)) \mathbf{x}_i, \\ \frac{\partial L}{\partial \mathbf{u}_j} =& \sum_{i \in \mathcal{I}_j} \big[y_{i,j} (\mu_i - \sigma(\mathbf{u}_j^\top \mathbf{x}_i)) \\ &+ (1 - y_{i,j})(1 - \mu_i - \sigma(\mathbf{u}_j^\top \mathbf{x}_i)) \big] \mathbf{x}_i. \end{aligned}$$

For initialization, we set $\mathbf{w}_0 = \mathbf{0}$ and initialize $\{\mathbf{u}_j\}_{j=1}^J$ randomly from the method of Yan et al. (2010).

Experiments

Our experiments tested the effectiveness of combining the crowd-generated data and the expert-generated data on a real data set.

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 where each token of tweets (texts) was labeled by workers of the AMT[2], and we used this data as a training and test set. Unlike standard data sets for an NER task, 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 ($y_i = 0$). Here, we omitted the named entity labels for the @usernames[3] in the same way as the paper by Ritter, Clark, and Etzioni (2011), because it was too easy to identify them.

The number of instances was $212,720$, and $8,107$ of them had expert-generated labels. The instances with expert-generated labels were also labeled by more than ten crowd-workers, and the instances without expert-generated labels were labeled by two workers. The data set had 269 workers

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

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

in total. 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 model parameters, we selected the features that appeared more than once in the training set, and we obtained sparse feature vectors with $161,901$ dimensions.

Setting

We varied the number of expert-generated labels in a training set to see the change in the performance of the classifiers. We constructed a basic training set from the instances without expert-generated labels, which consisted of $17,747$ instances and 42 workers. Then we constructed an additional training and test set from the instances with ground truth labels. We chose $2,790$ instances as an additional training set from the instances that have labels given by matching workers with the basic training set. The remaining $5,317$ instances were used as a test set. All of the noisy labels in both the basic and the additional training set were used as a training set, and $|\mathcal{I}_0|$ expert-generated labels of the additional training set were randomly chosen to be used as a training set. We varied $|\mathcal{I}_0|$ from 0 to $2,750$ in steps of 250. Each instance in the training set was labeled by two workers and the number of labels each worker gave is summarized in Table .

Classifiers were trained using the training set constructed above, and evaluated by calculating the precision, recall, and F-measure against the test set. We repeated the process 10 times and calculated the mean and the standard deviation of the precisions, recalls, and F-measures.

Competing Methods

We used 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 noisy labels $\{y_{i,j}\}_{j=1}^{J}$ for an instance $\mathbf{x}_i$, the true label y_i for the instance $\mathbf{x}_i$ is estimated using majority voting as

$$y_i = \begin{cases} 1 & \text{if } \sum_{j\in\mathcal{J}_i} y_{i,j} > |\mathcal{J}_i|/2, \\ 0 & \text{if } \sum_{j\in\mathcal{J}_i} y_{i,j} < |\mathcal{J}_i|/2, \\ \text{random} & \text{otherwise.} \end{cases}$$

For an instance $\mathbf{x}_i$ with an expert-generated label $y_{i,0}$, we used the $y_{i,0}$ as a true label without doing majority voting.

All-in-One-Classifier Method (AOC method). This 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. In this method, the ground truth data were considered as one worker and added to the anonymized training data.

Results

The averages and standard deviations of the precisions, recalls, and F-measures are summarized in Figures 1, 2, and 3,

		$\|\mathcal{I}_0\| = 1000$		$\|\mathcal{I}_0\| = 0$	
j	$\|\mathcal{I}_j\|$	α_j	β_j	α_j	β_j
0	1000	1	1	$*$	$*$
1	16684	0.542	1	0.538	1
2	7212	0.909	0.998	0.912	1
3	3960	0.518	0.997	0.531	0.998
4	2937	0.66	0.999	0.655	1
5	2407	0.909	0.997	0.963	1
6	1266	0.684	0.998	0.672	1
7	1210	0.767	0.99	0.763	0.992
8	809	0.789	0.993	0.742	0.999
9	708	0.75	0.987	0.765	0.991
10	634	0.723	0.996	0.728	1
11	518	0.609	0.993	0.631	0.998
12	468	0.747	1	0.661	1
13	462	0.802	0.995	0.833	1
14	273	0.624	1	0.62	1
15	237	0.975	0.956	0.974	0.963
16	218	0.915	0.994	0.919	1
17	214	0.312	1	0.312	1
18	189	0	1	0	1
19	180	0.416	0.971	0.439	0.970
20	165	0.797	1	0.796	1
21	164	0.877	1	0.872	1
22	146	0	1	0	1
23	144	0.828	0.907	0.693	0.902
24	133	0.997	0.992	0.884	1
25	123	0.363	0.785	0.369	0.785
26	122	1	0.949	1	0.949
27	112	1	0.898	1	0.910
28	109	0	1	0	1
29	104	1	0.187	1	0.187
30	101	1	1	1	1
31	98	0.87	0.976	0.898	1
32	95	0	1	0	1
33	94	1	0.965	1	0.976
34	92	1	0.945	1	0.945
35	90	1	0.971	1	0.971
36	88	0.63	0.488	0.626	0.488
37	86	0.674	0.79	0.767	0.811
38	81	1	0.984	1	0.971
39	78	0	1	0	1
40	78	0.686	0.804	0.692	0.804
41	76	1	1	1	1
42	74	0	1	0	1

Table 1: The model parameters estimated on a training set with $1,000$ expert-generated labels (middle columns), and a training set without expert-generated labels (right columns).

respectively, and the estimated parameters of the LC model on a crowd-generated data set ($|\mathcal{I}_0| = 0$) and a combined data set ($|\mathcal{I}_0| = 1000$) are summarized in Table .

We deduced four findings from these experimental results. First, the performance of the PC model and the AOC method measured by the F-measure was improved as the number of expert-generated labels increased. Second, the performance of the DDE model and the MV method measured by the F-measure seemed to be improved, but the variance was large. Third, the performance of the LC model was invariant with respect to the number of expert-generated labels. Forth, the estimated parameters of the LC model are almost the same regardless of the number of expert-generated labels in a training set.

These findings reflect the characteristics of these models. The first finding shows that the PC model can improve its ability by combining noisy labels and ground truth labels.

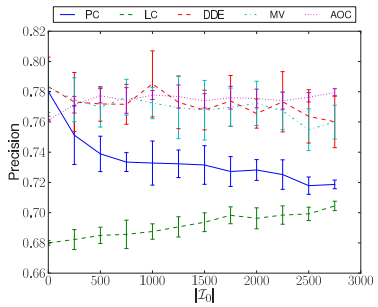

Figure 1: Precision versus the number of expert-generated labels.

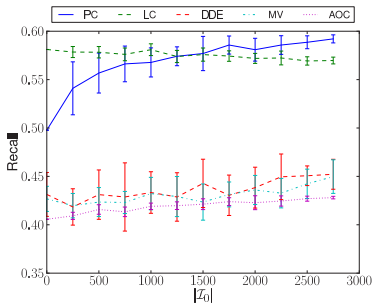

Figure 2: Recall versus the number of expert-generated labels.

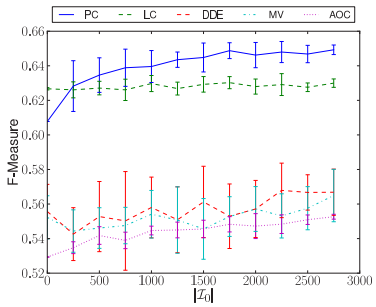

Figure 3: F-measure versus the number of expert-generated labels.

The second finding shows that the DDE model and the MV method are unstable. This is because each instance had only a small number of labels (in this experiment, each instance had only two labels), and the estimation algorithms of these models have some randomness. The MV method estimates ground truth labels randomly when the majority voting strategy doesn't work, and the DDE model initializes some variables randomly. The third and forth findings show that the LC model can estimate model parameters well even without expert-generated labels if we have a sufficient number of labels or the ability of the workers is not too low. In contrast, Kajino, Tsuboi, and Kashima (2012) reported that if there were too few labels or the ability of the workers was too low, then the LC model sometimes performed poorly.

Conclusion

In this paper, we extended three models to combine crowd-generated data and expert-generated data. A model proposed by Kajino, Tsuboi, and Kashima (2012) was extended by introducing a training set to the base classifier. A model proposed by Raykar et al. (2010) and a model proposed by Yan et al. (2010) were extended based on the idea proposed by Tang and Lease (2011). The experimental results on real data showed both improved and invariant performance and revealed several characteristics of the models.

Acknowledgment

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

References

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

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.

Kajino, H.; Tsuboi, Y.; and Kashima, H. 2012. A Convex Formulation for Learning from Crowds. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (to appear)*.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Jerebko, A.; Florin, C.; Valadez, G. H.; Bogoni, L.; and Moy, L. 2009. Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 889–896.

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.

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.

Tang, W., and Lease, M. 2011. Semi-Supervised Consensus Labeling for Crowdsourcing. In *ACM SIGIR Workshop on Crowdsourcing for Information Retrieval*.

Wauthier, F. L., and Jordan, M. I. 2011. Bayesian Bias Mitigation for Crowdsourcing. In *Advances in Neural Information Processing 24*, 1800–1808.

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.; Schmidt, M.; Hermosillo, G.; Bogoni, L.; Moy, L.; Dy, J.; and Malvern, P. 2010. Modeling annotator expertise: Learning when everybody knows a bit of something. In *International Conference on Artificial Intelligence and Statistics*, volume 9, 932–939.