Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Clustering 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

We present a clustered personal classifier method (CPC method) that jointly estimates a classifier and clusters of workers in order to address the learning from crowds problem. Crowdsourcing allows us to create a large but low-quality data set at very low cost. The learning from crowds problem is to learn a classifier from such a low-quality data set. From some observations, we notice that workers form clusters according to their abilities. Although such a fact was pointed out several times, no method has applied it to the learning from crowds problem. We propose a CPC method that utilizes the clusters of the workers to improve the performance of the obtained classifier, where both the classifier and the clusters of the workers are estimated. The proposed method has two advantages. One is that it realizes robust estimation of the classifier because it utilizes prior knowledge about the workers that they tend to form clusters. The other is that we can obtain the clusters of the workers, which help us analyze the properties of the workers. Experimental results on synthetic and real data sets indicate that the proposed method can estimate the classifier robustly. In addition, clustering workers is shown to work well. Especially in the real data set, an outlier worker was found by applying the proposed method.

Introduction

Crowdsourcing enables us to outsource tasks to a large number of anonymous workers. The strength of crowdsourcing is that it allows us to process large quantities of tasks that need human intelligence to complete. In machine learning, computer vision, and natural language processing, it is used to collect labels for instances, which are used as a training set in supervised learning. However, it is often found that the quality of the data obtained in crowdsourcing (which we call *crowd-generated data*) is unknown due to the anonymity of the workers and is sometimes quite low. In the worst case, some workers called *spammers* give completely non-informative labels to earn easy money. Therefore, to make full use of crowdsourcing, we have to take into account of the quality of the data. In this paper, we deal with a problem setting called a *learning from crowds problem*, where we seek to learn a *target classifier* that predicts the true labels from the crowd-generated data.

There are two types of approaches to the learning from crowds problem. One is an indirect approach where first the true labels are estimated from the crowd-generated data and then the target classifier is obtained by using a standard supervised learning method. The other is a direct approach where the target classifier is learnt directly from the data. In the indirect approach, estimation of the true labels consists of two steps. The first step is to collect multiple labels for each instance, which is called a *repeated labeling* (Sheng, Provost, and Ipeirotis 2008). The second step is to aggregate the labels to estimate the true labels using such as a *latent class method* (LC method) (Dawid and Skene 1979). The LC method employs a *latent class model* (LC model), where a labeling process is modeled with ability parameters of workers and it estimates the true labels through inference of the model. Other than the LC model, there have been proposed many models for a labeling process, which take in complex models of workers (Welinder et al. 2010) and incorporate other factors such as task-difficulty (Whitehill et al. 2009). On the other hand, the direct approach addresses the learning from crowds problem in one step. By introducing the target classifier into a model of a labeling process, we can estimate the classifier just by inferring the model. Raykar et al. (2010) first proposed a direct approach by introducing the target classifier into the LC method. Kajino, Tsuboi, and Kashima (2012) proposed a *personal classifier method* (PC method) where each worker is modeled as a classifier (called a *personal classifier*) and the target classifier is obtained by aggregating these classifiers.

In both types of approaches, devising the model of workers is important to address the problem. Modeling various properties of workers can improve the performance of estimation and allows us to give insight into anonymous workers and help us understand the crowd-workers. In this paper, we focus on a "clusters of workers" property and propose a direct approach method called *clustered personal classifier method* (CPC method) based on the PC method. To the best of our knowledge, the proposed method is the first method to take into account of the clusters of workers, which distinguishes the proposed method from the existing methods. First, we discuss the cluster property of workers and then briefly introduce the CPC method.

We first discuss that there are workers with similar and dissimilar expertise, which can be represented as clusters of workers. The simplest example is a spammer, who gives the same labels to all the instances or gives totally random labels. It is possible to cluster workers into spammers and non-spammers. Another example is given in the paper by Welinder et al. (2010) who sought to improve the quality of the estimated labels and provide insight into the properties of workers, similar to our work. They introduced a latent feature space and each worker was modeled as a latent classifier. In the analysis on a real data set, they observed that workers formed three visible clusters according to their expertise, as shown in Fig. 6 in their paper. These examples clearly show that there are similar and dissimilar workers. Exploiting the information of similarity and dissimilarity can improve the quality of the target classifier and can be one method to analyze anonymous workers.

Based on this idea, we propose a CPC method that jointly learns a classifier and clusters of workers from crowd-generated data. Our method is the first one that clusters workers in crowdsourcing, which is the main feature of this work. This can be used in two ways: estimation of the target classifier and analysis of the workers. A key technique is a clustering regularization that induces similar workers into one clustered worker. Since the clustering regularization term is a convex function, the parameter estimation is formulated as a convex optimization problem, and therefore a global optimum can be obtained.

We conducted experiments using both synthetic and real data sets. In experiments using synthetic data sets, we studied the conditions under which the CPC method can estimate reasonable clusters of workers. Then we compared the CPC method with other competing methods for the learning from crowds problem to show that the CPC method can robustly estimate the target classifier. In the experiment using a real data set, we examined the performance of the CPC method and analyzed the workers using the CPC method to find an outlier worker. These experimental results suggest that the CPC method works well in both usages.

Problem Setting and Notation

Let us introduce some notation and define the learning from crowds problem. We assume that there are I instances and each instance is indexed by $i \in \mathcal{I}(= \{1, \dots, I\})$ where $\mathcal{I}$ is the set of all of the instances. Each instance i is associated with its feature vector $\mathbf{x}_i \in \mathbb{R}^D$ and is assumed to have the ground truth label $y_i \in \{0, 1\}$, which is unobservable. We denote the set of feature vectors as $\mathcal{X} = \{\mathbf{x}_i \mid i \in \mathcal{I}\}$. We also assume that there are J workers and each worker $j \in \mathcal{J}(= \{1, \dots, J\})$ gave a noisy label y_{ij} to instance i where $\mathcal{J}$ is the set of all of the workers. Let $\mathcal{I}_j \subseteq \mathcal{I}$ be the set of the instances that worker j labeled, let $\mathcal{J}_i \subseteq \mathcal{J}$ be the set of the workers who gave labels to instance i, and let $\mathcal{Y}$ be the set of all the labels acquired by using crowdsourcing.

Our goal is to estimate a binary classifier that will output the true labels given $(\mathcal{X}, \mathcal{Y})$ as a training set. We call such a classifier the *target classifier*. For simplicity, we focus on the binary classification problem in this paper. However, the proposed approach can be applied to more general cases, including multi-class classification and regression problems.

Proposed Method

The proposed method is formulated based on the PC method (Kajino, Tsuboi, and Kashima 2012). We first review the PC method and introduce the proposed method and the algorithm. Our contribution is to introduce a *clustering regularization* that induces similar workers to be clustered.

Review of the Personal Classifier Method

The PC method is to estimate the target classifier by inferring the personal classifier model. We first review the personal classifier model. The target classifier is modeled as a logistic regression model

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

where the parameter $\mathbf{w}_0 \in \mathbb{R}^D$ is assigned a prior distribution as $p(\mathbf{w}_0 \mid \eta) = \mathcal{N}(\mathbf{0}_D, \eta^{-1}\mathbf{I}_D)$, where $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is a normal distribution with the mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ and the covariance matrix $\Sigma \in \mathbb{R}^{D \times D}$, $\mathbf{0}_D$ is the D-dimensional zero vector, $\mathbf{I}_D$ is the D-dimensional identity matrix, and $\eta > 0$ is a hyperparameter. Each worker j has a *classifier parameter* $\mathbf{w}_j \in \mathbb{R}^D$ and is modeled as a logistic regression model $p(y_{ij} = 1 \mid \mathbf{x}_i, \mathbf{w}_j) = \sigma(\mathbf{w}_j^\top \mathbf{x}_i)$. The classifiers are assumed to be related to the target classifier as

$$\mathbf{w}_j = \mathbf{w}_0 + \mathbf{c}_j, \ \mathbf{c}_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}_D, \lambda^{-1}\mathbf{I}_D),$$

where $\lambda > 0$ is a hyperparameter, and $\mathbf{c}_j$ models differences in the characteristics of the workers.

The model parameters $\mathbf{w}_0$ and $\mathbf{W} = \{\mathbf{w}_j \mid j \in \mathcal{J}\}$ are estimated by using the maximum-a-posteriori (MAP) estimators. The MAP estimators of $\mathbf{w}_0$ and $\mathbf{W}$ are obtained by solving the convex optimization problem

$$\min_{\mathbf{w}_0, \mathbf{W}} L(\mathbf{W}) + \frac{\lambda}{2} \sum_{j=1}^{J} \|\mathbf{w}_j - \mathbf{w}_0\|^2 + \frac{\eta}{2}\|\mathbf{w}_0\|^2, \quad (2)$$

where let $L(\mathbf{W}) = -\sum_{j=1}^{J} \sum_{i \in \mathcal{I}_j} [y_{ij} \log \sigma(\mathbf{w}_j^\top \mathbf{x}_i) + (1 - y_{ij}) \log(1 - \sigma(\mathbf{w}_j^\top \mathbf{x}_i))]$, and let $\| \cdot \|$ denotes the ℓ_2 norm.

Clustered Personal Classifier (CPC) Method

We propose a CPC method based on the PC method. To find clusters among the workers, we use a convex clustering penalty (Pelckmans et al. 2005; Hocking et al. 2011) as a regularizer. The convex clustering penalty is written as

$$\Omega(\mathbf{w}_0, \mathbf{W}) = \sum_{(j,k) \in \mathcal{K}} m_{jk} \|\mathbf{w}_j - \mathbf{w}_k\|, \quad (3)$$

where let $\mathcal{K} = \{(j, k) \mid 0 \leq j < k \leq J\}$, and let $m_{jk} > 0$ be hyperparameters[1]. By substituting the second term in Eq. (2) by Ω, we obtain

$$\min_{\mathbf{w}_0, \mathbf{W}} L(\mathbf{W}) + \mu\Omega(\mathbf{w}_0, \mathbf{W}) + \frac{\eta}{2}\|\mathbf{w}_0\|^2, \quad (4)$$

[1] We set $m_{jk} = 1/J$ for all $(j, k) \in \mathcal{K}$.

Algorithm 1 Optimization algorithm for Problem (4)

input: μ, $\{m_{jk} \mid (j,k) \in \mathcal{K}\}$, η, ρ, and $(\mathcal{X}, \mathcal{Y})$.
output: $\mathbf{w}_0$ and $\mathbf{W}$.

1: initialize $\mathbf{w}_j = \mathbf{0}$ for all $j \in \mathcal{J} \cup \{0\}$ and $\mathbf{u}_{jk} = \boldsymbol{\phi}_{jk} = \mathbf{0}$ for all $(j,k) \in \mathcal{K}$.
2: **repeat**
3: $(\mathbf{w}_0, \mathbf{W}) \leftarrow \arg\min_{\mathbf{w}_0, \mathbf{W}} \mathcal{L}_\rho(\mathbf{w}_0, \mathbf{W}, \mathbf{U}; \boldsymbol{\Phi})$.
4: $\mathbf{v}_{jk} \leftarrow \rho(\mathbf{w}_j - \mathbf{w}_k) - \boldsymbol{\phi}_{jk}$ for all $(j,k) \in \mathcal{K}$.
5: $\mathbf{u}_{jk} \leftarrow \max\left(0, \frac{\|\mathbf{v}_{jk}\| - \mu m_{jk}}{\rho \|\mathbf{v}_{jk}\|}\right) \mathbf{v}_{jk}$ for all $(j,k) \in \mathcal{K}$.
6: $\boldsymbol{\phi}_{jk} \leftarrow \boldsymbol{\phi}_{jk} + \rho(\mathbf{u}_{jk} - (\mathbf{w}_j - \mathbf{w}_k))$ for all $(j,k) \in \mathcal{K}$.
7: **until** the objective function converges.

where $\mu > 0$ is a hyperparameter that controls the strength of clustering. If μ is large, then the personal classifiers tend to be fused. Solving the convex optimization problem (4) allows us to jointly learn the target classifier and the clusters of workers. The target classifier can be obtained because the regularizer Ω defines the relation among the target classifier and the personal classifiers. The clusters can be obtained because if μ is appropriately large, some of the terms in Eq. (3) become 0, that is $\mathbf{w}_j = \mathbf{w}_k$ for some $(j,k) \in \mathcal{K}$, in the same way as in the group lasso (Yuan and Lin 2006). In this paper, we define worker j and worker k as being in the same cluster if and only if $\mathbf{w}_j = \mathbf{w}_k$ holds.

Two Usages of the CPC Method

The proposed method has two usages: estimation of the target classifier and clustering workers. Estimation of the target classifier is performed by solving Problem (4), which can be done by applying Algorithm 1 described below. The clusters of workers can be found in a hierarchical clustering manner by iteratively solving Problem (4) varying the parameter μ from 0 to arbitrarily large when all of the personal classifiers fuse into one. The result of clustering is obtained as a *pseudo-dendrogram* as shown in Fig. 4. The reason for the prefix "pseudo" is that the obtained graph is not a tree.

Algorithm

We devise an efficient algorithm for solving Problem (4) based on the Alternating Direction Method of Multipliers (ADMM) (Gabay and Mercier 1976; Boyd et al. 2010). The algorithm is shown in Algorithm 1. Intuitively, the algorithm updates the parameters gradually tightening the constraint equations. We provide the detailed derivation of the algorithm below.

We first introduce a new variable $\mathbf{u}_{jk} = \mathbf{w}_j - \mathbf{w}_k$ for all $(j,k) \in \mathcal{K}$ to make the non-differentiable terms separable and let $\mathbf{U} = \{\mathbf{u}_{jk}\}_{(j,k)\in\mathcal{K}}$. Problem (4) is equivalently written as

$$\min_{\mathbf{w}_0, \mathbf{W}, \mathbf{U}} L(\mathbf{W}) + \mu \sum_{(j,k)\in\mathcal{K}} m_{jk} \|\mathbf{u}_{jk}\| + \frac{\eta}{2}\|\mathbf{w}_0\|^2 \quad (5)$$
$$\text{s.t. } \mathbf{u}_{jk} = \mathbf{w}_j - \mathbf{w}_k,\ \forall (j,k) \in \mathcal{K}.$$

The augmented Lagrangian $\mathcal{L}_\rho$ for the constrained optimization problem (5) is written as

$$\begin{aligned}\mathcal{L}_\rho(\mathbf{w}_0, \mathbf{W}, \mathbf{U}; \boldsymbol{\Phi}) =& L(\mathbf{W}) + \mu \sum_{(j,k)\in\mathcal{K}} m_{jk}\|\mathbf{u}_{jk}\| + \frac{\eta}{2}\|\mathbf{w}_0\|^2 \\ &+ \sum_{(j,k)\in\mathcal{K}} \boldsymbol{\phi}_{jk}^\top (\mathbf{u}_{jk} - (\mathbf{w}_j - \mathbf{w}_k)) \\ &+ \frac{\rho}{2} \sum_{(j,k)\in\mathcal{K}} \|\mathbf{u}_{jk} - (\mathbf{w}_j - \mathbf{w}_k)\|^2,\end{aligned}$$

where let $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_{jk} \in \mathbb{R}^D\}_{(j,k)\in\mathcal{K}}$ be the set of the Lagrangian multipliers, and let ρ be a positive constant. We repeat three update, each with respect to $(\mathbf{w}_0, \mathbf{W})$, $\mathbf{U}$, and $\boldsymbol{\Phi}$ cyclically until convergence. The detailed update rules are described below. Let $t\ (\geq 0)$ denote the number of iterations.

(i) Update $(\mathbf{w}_0^{(t+1)}, \mathbf{W}^{(t+1)})$ given $\mathbf{U}^{(t)}$ and $\boldsymbol{\Phi}^{(t)}$.
Given $\mathbf{U}^{(t)}$ and $\boldsymbol{\Phi}^{(t)}$, $(\mathbf{w}_0^{(t+1)}, \mathbf{W}^{(t+1)})$ are updated as

$$(\mathbf{w}_0^{(t+1)}, \mathbf{W}^{(t+1)}) = \arg\min_{\mathbf{w}_0, \mathbf{W}} \mathcal{L}_\rho(\mathbf{w}_0, \mathbf{W}, \mathbf{U}^{(t)}; \boldsymbol{\Phi}^{(t)}). \quad (6)$$

Because $\mathcal{L}_\rho$ is differentiable and convex with respect to $\mathbf{w}_0$ and $\mathbf{W}$, the optimum can be obtained by using the L-BFGS algorithm (Byrd et al. 1995).

(ii) Update $\mathbf{U}^{(t+1)}$ given $(\mathbf{w}_0^{(t+1)}, \mathbf{W}^{(t+1)})$ and $\boldsymbol{\Phi}^{(t)}$.
Given $(\mathbf{w}_0^{(t+1)}, \mathbf{W}^{(t+1)})$ and $\boldsymbol{\Phi}^{(t)}$, $\mathbf{U}^{(t+1)}$ is updated as

$$\mathbf{U}^{(t+1)} = \arg\min_{\mathbf{U}} \mathcal{L}_\rho(\mathbf{w}_0^{(t+1)}, \mathbf{W}^{(t+1)}, \mathbf{U}; \boldsymbol{\Phi}^{(t)}).$$

$\mathbf{u}_{jk}^{(t+1)}$ can be calculated independently for each $(j,k) \in \mathcal{K}$ because the objective function is separable with respect to $\mathbf{U}$. The subdifferential of $\mathcal{L}_\rho$ with respect to $\mathbf{u}_{jk}$ is written as

$$\begin{aligned}&\partial_{\mathbf{u}_{jk}} \mathcal{L}_\rho(\mathbf{w}_0^{(t+1)}, \mathbf{W}^{(t+1)}, \mathbf{U}; \boldsymbol{\Phi}^{(t)}) \\ =& \mu m_{jk} \partial_{\mathbf{u}_{jk}} \|\mathbf{u}_{jk}\| + \boldsymbol{\phi}_{jk}^{(t)} + \rho(\mathbf{u}_{jk} - \mathbf{w}_j^{(t+1)} + \mathbf{w}_k^{(t+1)}),\end{aligned}$$

where $\partial_{\mathbf{x}} f(\mathbf{x})$ denotes the subdifferential of a function f with respect to $\mathbf{x}$. Considering the optimal condition is

$$\mathbf{0} \in \partial_{\mathbf{u}_{jk}} \mathcal{L}_\rho(\mathbf{w}_0^{(t+1)}, \mathbf{W}^{(t+1)}, \mathbf{U}; \boldsymbol{\Phi}^{(t)}),$$

we can obtain $\mathbf{u}_{jk}^{(t+1)}$ in a closed form,

$$\mathbf{u}_{jk}^{(t+1)} = \max\left(0, \frac{\|\mathbf{v}_{jk}\| - \mu m_{jk}}{\rho \|\mathbf{v}_{jk}\|}\right) \mathbf{v}_{jk}, \quad (7)$$

where we denote $\mathbf{v}_{jk} = \rho(\mathbf{w}_j^{(t+1)} - \mathbf{w}_k^{(t+1)}) - \boldsymbol{\phi}_{jk}^{(t)}$.

(iii) Update $\boldsymbol{\Phi}^{(t+1)}$ given $(\mathbf{w}_0^{(t+1)}, \mathbf{W}^{(t+1)})$ and $\mathbf{U}^{(t+1)}$.
Following the rule of the ADMM, $\boldsymbol{\Phi}^{(t+1)}$ is updated as

$$\boldsymbol{\phi}_{jk}^{(t+1)} = \boldsymbol{\phi}_{jk}^{(t)} + \rho(\mathbf{u}_{jk}^{(t+1)} - (\mathbf{w}_j^{(t+1)} - \mathbf{w}_k^{(t+1)})). \quad (8)$$

By cyclically repeating the update rules (6), (7), and (8) as shown in Algorithm 1 until convergence, we obtain the optimum for Problem (4). The convergence proof can be given in the same way as the proof for the ADMM (Gabay and Mercier 1976).

Discussions

Computational Complexity. Space complexity of Algorithm 1 is $O(J(dJ + I))$ and time complexity of each step of Algorithm 1 is $O(dJ^2)$ except for the update (6). These complexities are larger than those of the PC method, where space complexity is $O(J(d + I))$ and time complexity is smaller than that of the update (6).

Selecting Hyperparameters. Generally, it is difficult to select hyperparameters of *any* method for the learning from crowds problem because the true labels are not used in a learning phase, and we cannot apply the standard cross-validation technique. In this paper, we employ a heuristic criterion where the hyperparameters are selected that perform the best in a small-sized test set created by experts. Although this heuristics can cause the over-fitting problem, it helps us to tune the parameters more or less. We leave a theoretically guaranteed criterion as future work.

Experiments

We evaluated the CPC method using synthetic and real data sets. First, we used synthetic data sets to examine the properties of the clustering regularization and the conditions under which the CPC method worked well. Then we used a real data set to evaluate the performance.

Competing Methods

We used the following four methods as competing methods.

(i) Majority Voting (MV) Method. Given crowd-generated labels $\mathcal{Y}$, y_i is estimated using majority voting as

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

The target classifier is modeled as a logistic regression model and is estimated by maximizing the likelihood function given $(\mathcal{X}, \{y_i\}_{i \in \mathcal{I}})$ as a training set.

(ii) All-in-One-Classifier (AOC) Method. This considers $(\mathcal{X}, \mathcal{Y})$ as a training set for one logistic regression model. In other words, we ignore the worker IDs and use all of the labels to learn the target classifier. Note that the CPC method is the same as the AOC method when μ is large enough to fuse all the parameters.

(iii) Latent Class (LC) Method (Raykar et al. 2010). The target classifier is assumed to be a logistic regression model and a labeling process of each worker is modeled as a two-coin model:

$$\alpha_j = p(y_{ij} = 1 \mid y_i = 1),\ \beta_j = p(y_{ij} = 0 \mid y_i = 0).$$

The target classifier can be estimated by the EM algorithm.

(iv) Personal Classifier (PC) Method. The detailed description appeared in the previous section.

Table 1: Parameters of two synthetic data sets. Each column shows the parameter of workers in a corresponding cluster.

	p_j		$\mathbf{w}_j$	
	$\mathcal{J}_g$	$\mathcal{J}_b$	$\mathcal{J}_g$	$\mathcal{J}_b$
Spam	$p_{\mathcal{J}_g}\ (> 0.5)$	0.5	$\mathbf{w}_0$	
Corrupt	$p_{\mathcal{J}}\ (> 0.5)$		$\mathbf{w}_0$	$\mathbf{w}_{\mathcal{J}_b}$

Experiment 1: Properties

We used synthetic data sets to see the properties of the CPC method. In the experiments *Clustering*, we examined the conditions under which the CPC method could estimate clusters of workers. In the experiments *Comparison*, we compared the CPC method with the competing methods to see the advantages and disadvantages of the CPC method.

Synthetic Data Sets. We used two synthetic data sets: a *Spam Data Set* and a *Corrupt Data Set*. Both simulate a situation in which workers form one *good* cluster $\mathcal{J}_g\ (\subseteq \mathcal{J})$ and one *bad* cluster $\mathcal{J}_b\ (= \mathcal{J} \backslash \mathcal{J}_g)$. A Spam Data Set simulates spammers who give random labels and experts who give reliable labels. A Corrupt Data Set simulates two clusters of the workers that have different abilities. The difference from the Spam Data Set is that even the bad workers can give slightly informative labels.

We generated both data sets based on the following model that was a combination of the LC model and the PC model. Each worker $j \in \mathcal{J}$ is assumed to have an *ability parameter* $p_j \in [0, 1]$ and a *classifier parameter* $\mathbf{w}_j \in \mathbb{R}^D$ and gives labels as

$$y_{ij} = \begin{cases} \mathbf{1}_{\mathbf{w}_j^\top \mathbf{x}_i > 0} & \text{with probability } p_j, \\ 1 - \mathbf{1}_{\mathbf{w}_j^\top \mathbf{x}_i > 0} & \text{with probability } 1 - p_j, \end{cases}$$

where $\mathbf{1}_{\text{condition}} = 1$ if the condition holds, and $\mathbf{1}_{\text{condition}} = 0$ otherwise. We simulate clusters of workers by assuming that workers in the same cluster have the same parameters. Based on this model, we define both data sets as shown in Table 1[2]. In the Corrupt Data Set, the bad workers have a corrupted classifier parameter $\mathbf{w}_{\mathcal{J}_b}$ where some elements of $\mathbf{w}_{\mathcal{J}_b}$ are the same as those of $\mathbf{w}_0$ while the others are 0. For example, if $\mathbf{w}_{\mathcal{J}_g} = [1\ 1]^\top$, then the corrupted classifier parameter may be $\mathbf{w}_{\mathcal{J}_b} = [1\ 0]^\top$. This means that the bad workers dismiss the differences in the second dimension while the good workers can distinguish objects using all dimensions.

For a test set, we generated $10,000$ instances, and the true labels were generated as

$$y_i = \mathbf{1}_{\mathbf{w}_0^\top \mathbf{x}_i > 0}.$$

$\{\mathbf{x}_i\}_{i \in \mathcal{I}}$ were sampled from the uniform distribution $\mathcal{U}([-10, 10)^D)$, and we assumed $|\mathcal{I}_j| = I$ for all $j \in \mathcal{J}$.

Settings and Results. We conducted two experiments to investigate the properties of the clustering regularization. The detailed experimental settings are shown in Table 2 where let $\mathbf{1}_D \in \mathbb{R}^D$ be the vector whose elements are 1.

[2]We abuse notation to write $p_{\mathcal{J}_b} = p_j\ (j \in \mathcal{J}_b)$ etc.

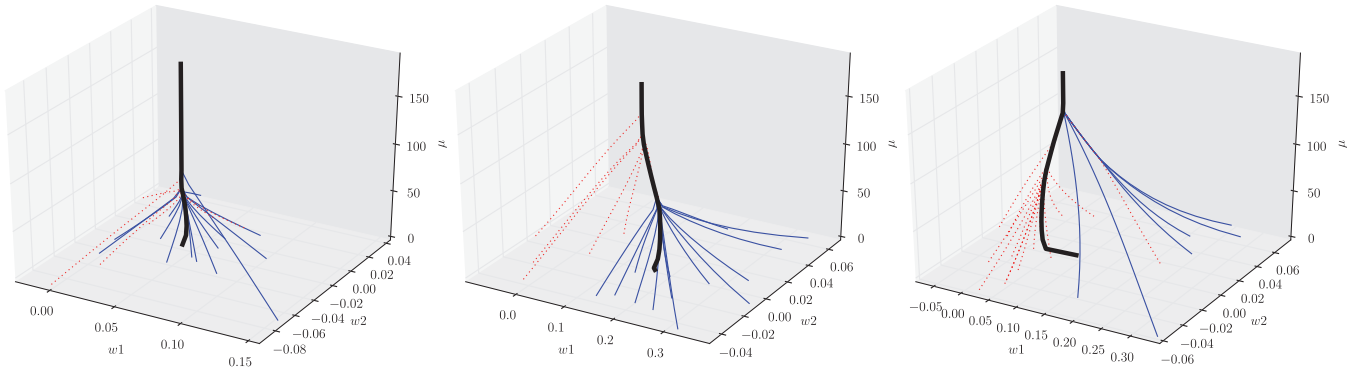

Figure 1: The estimated $\mathbf{w}_0$ (a bold black line) and $\mathbf{w}_j$ (a blue solid line for an expert and a red dotted line for a spammer) of the CPC method on the Spam Data Sets with different parameters, $(p_{\mathcal{J}_g}, r) = (0.6, 0.3), (0.8, 0.3), (0.8, 0.7)$ from the left to the right figure. The x and y-axis correspond to the first and the second dimension of $\mathbf{w}_j$ and the z-axis to μ.

Table 2: Experimental settings. Each row explains the parameters of the data set used in each experiment.

Experiments	Data Set	D	J	I	r	$\mathbf{w}_{\mathcal{J}_g}$	$\mathbf{w}_{\mathcal{J}_b}$	$p_{\mathcal{J}_g}$	$p_{\mathcal{J}_b}$	Result
Clustering	Spam	2	20	100	**0.3**	$[1\ 0]^\top$		**0.6**	0.5	Fig. 1 (**left**)
	Spam	2	20	100	**0.3**	$[1\ 0]^\top$		**0.8**	0.5	Fig. 1 (**mid**)
	Spam	2	20	100	**0.7**	$[1\ 0]^\top$		**0.8**	0.5	Fig. 1 (**right**)
Comparison	Spam	**2**	10	10	0 to 1.0 by 0.1	$[1\ 0]^\top$		0.8	0.5	Fig. 2 (**left**)
	Corrupt	**2**	10	10	0 to 1.0 by 0.1	$[1\ 1]^\top$	$[1\ 0]^\top$	0.8		Fig. 2 (**right**)
	Spam	**10**	10	10	0 to 1.0 by 0.1	$[\mathbf{1}_5^\top\ \mathbf{0}_5^\top]^\top$		0.8	0.5	Fig. 3 (**left**)
	Corrupt	**10**	10	10	0 to 1.0 by 0.1	$[\mathbf{1}_5^\top\ \mathbf{1}_5^\top]^\top$	$[\mathbf{1}_5^\top\ \mathbf{0}_5^\top]^\top$	0.8		Fig. 3 (**right**)

(i) Clustering. We examined the conditions of the data sets under which we could get reasonable clusters. We generated Spam Data Sets with $(p_{\mathcal{J}_g}, r) = (0.6, 0.3), (0.8, 0.3), (0.8, 0.7)$ where let $r = |\mathcal{J}_b|/J$ and obtained a pseudo-dendrogram by repeatedly estimating the model parameters by increasing μ from 0 to 200 by 10.

The estimated $\{\mathbf{w}_j\}_{j=0}^J$ are shown in Fig. 1. First, the result of $(p_{\mathcal{J}_g}, r) = (0.8, 0.3)$ (Fig. 1, **(mid)**) shows that as we increase μ, the parameters of the experts and $\mathbf{w}_0$ fuse around $\mu = 50$, and those of the spammers gradually join the cluster around $\mu = 100$. This cannot be observed when $(p_{\mathcal{J}_g}, r) = (0.6, 0.3)$ (Fig. 1, **(left)**). This suggests that if $p_{\mathcal{J}_g}$ is small, it is difficult to distinguish between experts and spammers, and hence clustering fails. Second, by comparing the result of $(p_{\mathcal{J}_g}, r) = (0.8, 0.3)$ (Fig. 1, **(mid)**) with that of $(p_{\mathcal{J}_g}, r) = (0.8, 0.7)$ (Fig. 1, **(right)**), we notice that if r is small, $\mathbf{w}_0$ is fused with the parameters of the experts and if r is big, $\mathbf{w}_0$ is fused with those of the spammers. This suggests that if r is big, the CPC method will not work well. From these, we can conclude that if $p_{\mathcal{J}_g}$ is big and r is small, a reasonable clustering result will be obtained. Note that this condition will not be hard in reality.

(ii) Comparison. We show that the CPC method is more robust than the PC method. As we show later, the PC method does not work well in the case of a high-dimensional feature space because it is difficult to estimate personal classifiers. Therefore we examined whether the CPC method could work even in the high-dimensional case. First, we examined the case when $D = 2$. Then we increased D to 10 to make unfavorable data sets to the PC method. We generated training and test data sets with parameters as shown in Table 2, estimated classifiers using all the methods, and calculated the AUCs of the ROC curves. This procedure was repeated 50 times to obtain the means of the AUCs. We set $\eta = 1.0$ for all of the methods to make the comparison and analysis easier. However, this is not unfair because we assign the same prior knowledge to $\mathbf{w}_0$.

First, we study the results of $D = 2$ (Fig. 2). If $r < 0.5$, the AUCs of all of the methods are similar. If $r > 0.5$, the ranking by their AUCs is CPC=PC=AOC>LC=MV. Then we study the results of $D = 10$ (Fig. 3). The ranking by their AUCs is roughly LC=MV>CPC=PC>AOC. This is because in the high-dimensional case, the number of the model parameters of the PC and CPC method is much larger than that of the other methods, and it is difficult to estimate the personal classifiers. However, the CPC method was always better than the PC method, and in the experiment using a Corrupt Data Set (Fig. 3 **(right)**), if $r > 0.5$, the AUCs of the PC and CPC method are comparable to those of the MV method and higher than those of the LC method.

From these, we conclude that the CPC method performs

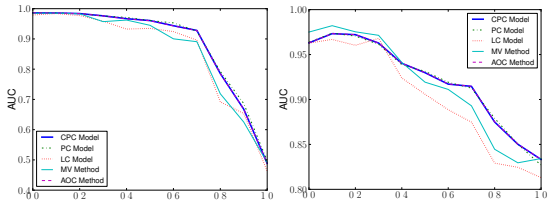

Figure 2: AUC comparison on 2-dimensional data sets varying r from 0 to 1 in steps of 0.1. The data sets are **(left)** a Spam Data Set and **(right)** a Corrupt Data Set.

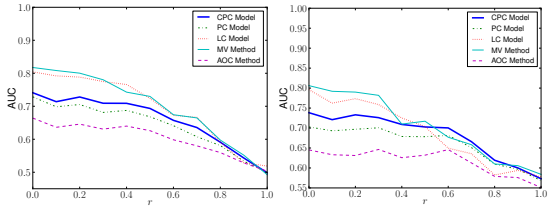

Figure 3: AUC comparison on 10-dimensional data sets varying r from 0 to 1 in steps of 0.1. The data sets are **(left)** a Spam Data Set and **(right)** a Corrupt Data Set.

better than the PC method even if it is difficult to estimate personal classifiers, and performs as good as the PC method and better than other methods if it is easy to estimate them.

Experiment 2: Performance

We used a real data set to see the performances.

Real Data Set. We used a data set for a *Named Entity Recognition* (NER) task, which aimed at identifying the names of persons, organizations, locations, and similar entities in sentences. Finin et al. (2010) created a Twitter data set where each token in a tweet (a message on Twitter) was labeled by workers of the AMT. Unlike the standard NER data sets, the segment boundaries of each entity were not given. Therefore we considered the task as a binary classification problem to identify whether each token was in a named entity or not. We omitted the named entity labels for the usernames[3] in the same way as Ritter, Clark, and Etzioni (2011), because it was too easy to identify them. The training set consists of $17,747$ tokens. Each token was labeled by two workers and 42 workers gave labels. The feature representation for each token was the same as Ritter, Clark, and Etzioni (2011). To reduce the dimensionality, we selected the features that appeared more than once in the training set, and thus obtained $161,901$-dimensional sparse feature vectors. The test set consists of $8,107$ tokens that have ground truth labels.

Settings and Results. Classifiers were trained using the training set and evaluated by calculating precision, recall, and the F-measure on the test set. For each method, hyperparameters were selected using the F-measure.

[3]The @ symbol followed by the unique username is used to refer to an other user in Twitter.

Table 3: Performance comparisons on the real data set.

	Precision	Recall	F-measure
CPC Method	0.647	0.716	**0.680**
PC Method	0.637	0.721	0.677
LC Method	0.625	**0.732**	0.675
AOC Method	0.680	0.670	0.675
MV Method	**0.686**	0.651	0.668

The results are shown in Table 3. First, the F-measure shows that the CPC method is better than the other methods. Second, when the CPC method is compared with the PC method and the AOC method, the ranking by their precision is AOC>CPC>PC, whereas that by their recall is PC>CPC>AOC. These reflect the fact that the CPC method has an intermediate regularization of the PC method and the AOC method. Therefore we can conclude that this formulation helps to improve the precision-recall tradeoff. The improvement seems to be small, but this result is acceptable considering the fact that the performances of the PC method and the CPC method are worse than those of other methods when the feature space is high-dimensional (see Experiment 1, Comparison). Although combining dimension reduction methods may improve the performance of the CPC method, such methods are out of our scope, and we leave them as future work.

We also plotted a pseudo-dendrogram in Fig. 4. This shows that clustering here is not an exact hierarchical clustering because the clusters sometimes split as we increase μ. We can see that $\mathbf{w}_0$ does not join the clusters if $\mu < 200$, which suggests that the abilities of the workers are diverse, and there are several clusters with much different parameters, which is partly because the task is difficult. In addition, we found an outlier worker (worker 30) who did not join the other clusters when $\mu < 250$. He had low precision and high recall (0.454 and 0.857 on $\mu = 10$) although the others had well-balanced precision and recall like the result shown in Table 3. This shows that the CPC method can also be used as a method to analyze workers.

Related Work

This paper is related to two research areas: crowdsourcing and a sparsity-inducing regularization.

Crowdsourcing

After several crowdsourcing systems were launched, many researchers have done research on it. Although diverse problems have been studied, the most frequent problem is a quality problem of results obtained in crowdsourcing. The most promising approach for this is a statistical one that cleans up a low-quality data set. The existing methods can be classified into three based on their objectives.

Estimating True Labels. Many existing approaches dealt with the *quality control problem* (Lease 2011) where a

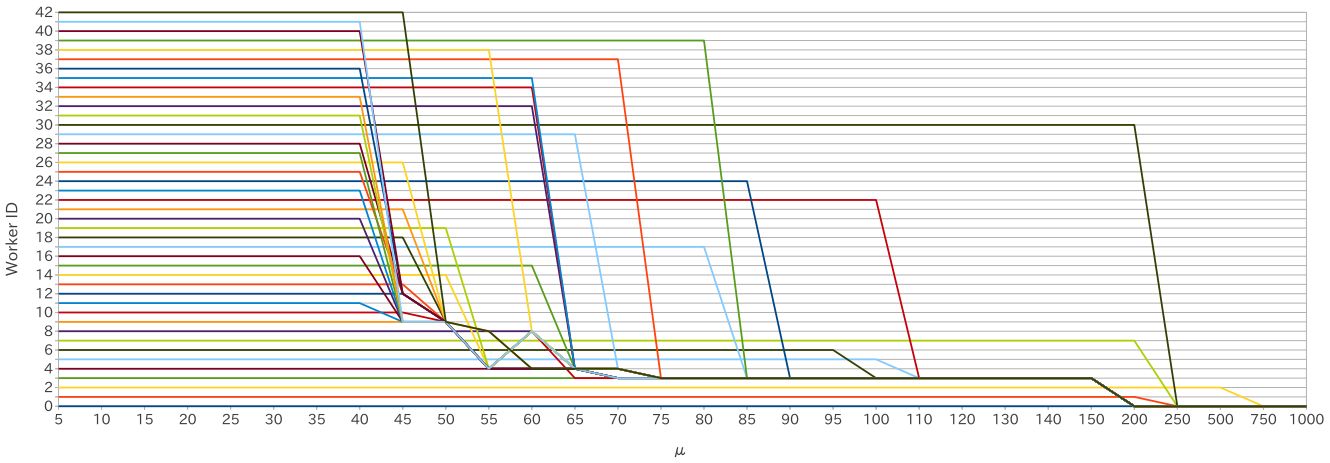

Figure 4: Result of clustering on the real data set. The x-axis corresponds to μ with which we estimated the model and the y-axis to the worker ID. The target classifier is represented as Worker ID 0. The line for worker j joins that for worker k when $\mathbf{w}_j = \mathbf{w}_k$ holds, which implies that worker j and worker k are in the same cluster.

goal is to estimate the true labels from crowd-generated data. Most of them were proposed based on two ideas, a repeated labeling approach (Sheng, Provost, and Ipeirotis 2008) and the LC model (Dawid and Skene 1979). A repeated labeling approach assigns multiple workers to each instance for the purpose of estimating the true labels. The LC model is a probabilistic model to estimate the true labels from multiple noisy labels considering the abilities of the workers, which was originally proposed by Dawid and Skene (1979) to aggregate diagnoses by multiple doctors. Whitehill et al. (2009) introduced task-difficulty parameters. Lin, Mausam, and Weld (2012) considered a problem where the output space of the workers is infinitely large. Welinder et al. (2010) introduced a latent feature space to find not only the true labels but richer information such as *schools of thought* among the workers. Their research is related to our research in that they found the clusters of workers in a real data set by visual observation, but different from ours in that they did not propose any algorithm to find the clusters. Tian and Zhu (2012) also dealt with a problem of estimating schools of thought when there were multiple true labels. Their research is also related to ours, but has different settings and objectives. They aimed at finding multiple true labels whereas we aim at finding the target classifier.

Estimating a Classifier. Some aimed at estimating the target classifier from noisy data. Raykar et al. (2010) extended the model by Dawid and Skene to estimate a classifier. Yan et al. (2011) introduced an active learning strategy. Dekel and Shamir (2009) proposed a data-cleaning approach and provided theoretical guarantees for their algorithm. Kajino, Tsuboi, and Kashima (2012) proposed the PC method that was different from the LC method and did not require the repeated labeling. Although they had the same purpose with us, none of them did not take clusters of workers into account, which distinguishes the CPC method from others.

Estimating Relationships between Data. This setting is to estimate the relationships between data. Tamuz et al. (2011) used crowdsourcing to construct kernel functions, and Gomes et al. (2011) performed clustering of data like pictures via crowdsourcing. Both approaches used crowdsourcing to identify the similarities between pairs of objects, which is easy for humans but difficult for machines.

Sparsity-inducing Regularization

Since the lasso (Tibshirani 1996) was proposed, intensive research has been conducted about sparsity-inducing regularizations. This allows us to find a comprehensive model by solving a convex optimization problem. The most related one is the group lasso (Yuan and Lin 2006) where predefined groups of variables are selected. This idea was employed to formulate a hierarchical clustering in a convex form (Pelckmans et al. 2005; Hocking et al. 2011).

Conclusion

We introduced a clustered personal classifier method that provided a joint estimation of the model parameters and the clusters of workers. The experiments using synthetic data sets showed both the advantages and disadvantages of the proposed method, and the experiments using the real data set showed that the proposed method worked well in a real setting. By applying the proposed method, we found an outlier worker in the real data set, which indicated that the analysis using the proposed method was effective.

Acknowledgments

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

References

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2010. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* 3(1):1–122.

Byrd, R. H.; Lu, P.; Nocedal, J.; and Zhu, C. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing* 16(5):1190–1208.

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 Statistics)* 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.

Gabay, D., and Mercier, B. 1976. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications* 2(1):17–40.

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

Hocking, T. D.; Joulin, A.; Bach, F.; and Vert, J.-P. 2011. Clusterpath: an algorithm for clustering using convex fusion penalties. In *Proceedings of the 28th International Conference on Machine Learning*, 745–752.

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*, 73–79.

Lease, M. 2011. On Quality Control and Machine Learning in Crowdsourcing. In *Proceedings of the third Human Computation Workshop*, 97–102.

Lin, C. H.; Mausam; and Weld, D. S. 2012. Crowdsourcing control: moving beyond multiple choice. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, 491–500.

Pelckmans, K.; De Brabanter, J.; Suykens, J.; and De Moor, B. 2005. Convex clustering shrinkage. In *Proceedings of Workshop on Statistics and Optimization of Clustering*.

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.

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.

Tian, Y., and Zhu, J. 2012. Learning from crowds in the presence of schools of thought. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 226–234.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58(1):267–288.

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. G. 2011. Active learning from crowds. In *Proceedings of the 28th International Conference on Machine Learning*, 1161–1168.

Yuan, M., and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(1):49–67.