Learning from Crowds under Experts' Supervision

Qingyang Hu[1], Qinming He[1], Hao Huang[2], Kevin Chiew[3], and Zhenguang Liu[1]

[1] College of Computer Science and Technology,
Zhejiang University, Hangzhou, China
{huqingyang,hqm,zhenguangliu}@zju.edu.cn
[2] School of Computing, National University of Singapore, Singapore
huanghao@comp.nus.edu.sg
[3] Provident Technology Pte. Ltd., Singapore
kev.chiew@gmail.com

Abstract. Crowdsourcing services have been proven efficient in collecting large amount of labeled data for supervised learning, but low cost of crowd workers leads to unreliable labels. Various methods have been proposed to infer the ground truth or learn from crowd data directly though, there is no guarantee that these methods work well for highly biased or noisy crowd labels. Motivated by this limitation of crowd data, we propose to improve the performance of crowdsourcing learning tasks with some additional expert labels by treating each labeler as a personal classifier and combining all labelers' opinions from a model combination perspective. Experiments show that our method can significantly improve the learning quality as compared with those methods solely using crowd labels.

Keywords: Crowdsourcing, multiple annotators, model combination, classification.

1 Introduction

Crowdsourcing services such as Amazon Mechanical Turk have made it possible to collect large amount of labels at relatively low cost. Nonetheless, since the reward is small and the ability of workers is not certified, the labeling quality of crowd labelers is often much lower than that of an expert. In the worst case, some workers just submit random answers to get the fee deviously. One approach to dealing with low quality labels is repeated-labeling. Sheng *et al.* [16] empirically showed that under certain assumptions, repeated-labeling can improve the label quality. Thus in crowdsourcing, people may collect multiple labels $y_i^1, y_i^2, \ldots, y_i^L$ from L different labelers for one instance x_i, while in traditional supervised learning, one instance x_i corresponds to one label y_i.

The problem remains as how to learn a reliable predictive model with the unreliable crowd labels. Various methods have been proposed to infer the ground truth [4, 10] or learn from crowd labels directly [8, 15]. The basic idea is employing generative models for the labeling processes of crowd labelers. While these models are useful under certain conditions, their assumptions on labelers are not easy to verify for a certain task.

This situation motivates us to investigate making full use of opinions collected from crowds by incorporating some expert labels, which seems more sensible than trying to verify the behavior of each labeler. Intuitively, combining expert labels with crowd

V.S. Tseng et al. (Eds.): PAKDD 2014, Part I, LNAI 8443, pp. 200–211, 2014.

labels is expected to achieve higher learning quality than solely using crowd labels though, little work has been done under this configuration since most of the existing work has focused on crowd labels.

This paper proposes to improve the performance of crowdsourcing learning tasks with a minimum number of expert labels by maximizing the utilization of both the crowd and expert labels[1]. Our major contribution is a formalized framework for utilizing expert labels in crowdsourcing. Following a series of existing work [8, 15, 19], our work focuses on supervised classification problems.

Some existing models [8, 13, 15] are capable of combining expert labels by straightforward extensions. The major difference between our method and these models is that we use prior beliefs on experts much more explicitly.

1.1 An Illustrative Example

In what follows, we illustrate the limitation of crowd data with an example and explain the idea which forms the basis of our framework. Fig. 1(a) shows a synthetic dataset for binary classification. For each class, we sample 100 points respectively from two different Gaussian distributions, and get four underlying clusters. We simulate two labelers whose opinions differ in one cluster as shown in Figs. 1(b) & 1(c). Here no model that uses the crowd labels without extra information can weight one labeler over the other since there is simply not enough evidence. Nonetheless, these two labelers provide very informative labels. Labeler 1 actually gave all correct labels. If we can identify this fact by a few expert labels, we achieve an efficient method.

However, the problem is not trivial even for this toy data set. Supposing that we choose a controversial point and let an expert label it, we will find that Labeler 1 gave the correct answer. This is far from enough to conclude that Labeler 1 gave true labels for all controversial points given that in practice we only have crowd labels and are not aware of the underlying data distribution. Adding more expert labels may increase our confidence on Labeler 1, still a formalized mechanism is needed to combine the ground truth with crowd data.

We address the problem by a model combination process. We train a logistic regression classifier for each labeler separately with the labels provided by that labeler, thus get 2 classifiers. A data instance $\boldsymbol{x}_i$ will then get 2 predictions $\{f_1(\boldsymbol{x}_i), f_2(\boldsymbol{x}_i)\}$ from the 2 classifiers, where $f_\ell(\boldsymbol{x}_i)$ ($\ell \in \{1, 2\}$) is the posterior probability of the class colored in blue. We treat the values of $f_\ell(\boldsymbol{x}_i)$ as features in a new space, shown in Figure 1(d). This is referred to as the *intermediate feature space* [11]. The final prediction is made by another classifier in this intermediate feature space.

By summarizing the opinions of labelers using personal classifiers, the separation between classes becomes clearer and the controversial area is projected to the bottom right in the new space and becomes more compact. Incorporating expert label evidence in this space is much easier compared with the crowd labels in the original space. A few ground truth labels in the controversial area will enables most classifiers built in

[1] We assume that an expert always gives true labels and use the two terms 'expert labels' and 'ground truth' interchangeably. As experts can also make mistakes, this assumption is a simplification and may be relaxed in future work.

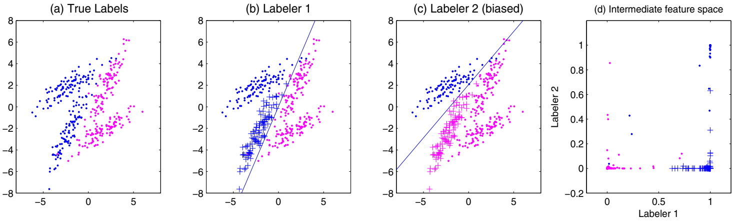

Fig. 1. An illustrative example. Instances labeled with cross(+) in (b)(c)(d) are controversial between the two labelers. These controversial data instances are gathered at the bottom right in the intermediate feature space as shown in (d).

this space to favor Labeler 1 over Labeler 2 naturally. We leave the the crucial step of combining expert evidence to the experiment section after we formalize our framework.

2 Related Work

With the arising of crowdsourcing services, crowd workers have shown their power in applications such as sentiment tracking [3], machine translation [1] and name entity annotating [5]. A key problem in crowdsourcing research is modeling data from multiple unreliable sources for inferring the ground truth. The problem has its origin in the early work [4] for combining multiple diagnostic test results. Recent work addressed problems with the same formulation by methods such as message transferring [10] and graphical models [13].

Our framework adopts the idea of learning a classifier from crowd data directly. Raykar *et al.* [15] and Yan *et al.* [19] treat true labels as hidden variables which are inferred by the EM algorithm. Kajino *et al.* [8] infer only the true classifier by personal classifiers without considering true labels explicitly. The nature of our method is similar to that of Kajino *et al.* [8], focusing on the final learning tasks and not being tangled with the correctness of a certain label.

To the best of our knowledge, very little work considered the case of learning from crowd and expert data simultaneously. Kajino *et al.* [9] addressed this problem by extending some existing models straightforwardly. Wauthier and Jordan [18] also used some expert labels. In their model crowd labels only make effects through the shared latent factors which express labelers. Our method differs from these work in both motivation and formulation.

We treat combining opinions of labelers as model combination. Getting the optimal combination of a group of pattern classifiers has been studied thoroughly for a long time and various methods have been proposed to employ the intermediate feature space. Merz [14] proposed to do feature extraction using singular decomposition in this space and Kuncheva *et al.* [12] proposed to combine classifiers giving soft labels using decision templates. In traditional model combination framework, multiple classifiers are obtained by different models trained on the *same* data set. Here the scenario is different, i.e., we have multiple unreliable label sets to train multiple classifiers, and we propose to use some reliable labels to combine them. Under the crowdsourcing setting the idea of absorbing the evidence of true labels in the intermediate feature space is also original.

3 Learning from Crowds and Experts

In this paper we focus on binary classification problems with crowdsourcing training data. The extension to multi-class cases is conceptually straightforward.

3.1 Problem Formulation

Formally, a crowdsourcing training set is denoted as $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, where instance $\boldsymbol{x}_i \in \mathbb{R}^D$ is a D-dimensional feature vector. We have L distinct labelers each of which gives labels to all N data instances.[2] The label given by the ℓth labeler for instance $\boldsymbol{x}_i$ is denoted as y_i^ℓ where $y_i^\ell \in \{-1, 1\}$. All labels corresponding to $\boldsymbol{x}_i$ are collected in the L-dimensional vector $\boldsymbol{y}_i$.

Different from most of the existing methods, we use some additional expert-labeled instances to improve the model quality. If there are N_0 expert labels, then the expert training set is $\mathcal{D}_0 = \{(\boldsymbol{x}_j, y_j^0)\}_{j=1}^{N_0}$ where $\boldsymbol{x}_j$ is again a D-dimensional feature vector and y_j^0 is the true label provided by the expert. Note that an expert-labeled instance $\boldsymbol{x}_j$ in $\mathcal{D}_0$ is not necessarily in $\mathcal{D}$. The task is to learn a reliable predictive function $f : \mathbb{R}^D \to [0, 1]$ for unseen data by taking both training sets $\mathcal{D}$ and $\mathcal{D}_0$ as inputs where $f(\boldsymbol{x}) = p(y = 1|\boldsymbol{x})$ is the posterior probability of the positive class. We denote the predictive function in this way for the convenience of the following steps.

3.2 Building Intermediate Feature Space

We extract the crowd opinions by treating labelers as personal classifiers. For the ℓth labeler, we use the personal training set $\mathcal{D}_\ell = \{(\boldsymbol{x}_i, y_i^\ell)\}_{i=1}^{N}$ to learn a classifier. Any classification model that expresses predictions as posterior probabilities of classes is compatible with our approach. Here we follow the work [8] and use a logistic regression model for each labeler, which is given by

$$\Pr[y = 1|\boldsymbol{x}, \boldsymbol{w}] = \boldsymbol{\sigma}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{w}$ is the model parameter and the logistic sigmoid function is defined as $\boldsymbol{\sigma}(a) = 1/(1 + e^{-a})$. We express all prediction functions of classifiers as an ensemble $\mathcal{F} = \{f_1, f_2, \ldots, f_L\}$ where $f_\ell(\boldsymbol{x})$ is the prediction of the classifier obtained from labeler ℓ on instance $\boldsymbol{x}$. The outputs of all L classifiers for a particular instance $\boldsymbol{x}_i$ is organized in an L-dimensional vector $[f_1(\boldsymbol{x}_i), f_2(\boldsymbol{x}_i), \ldots, f_L(\boldsymbol{x}_i)]^{\mathrm{T}}$, which is referred to as a *decision profile* [11]. In what follows, we denote this vector as $\boldsymbol{dp}_i$ with the ℓth element $dp_i^\ell = f_\ell(x_i)$. We treat values of dp_i^ℓ as features in a new feature space, namely the intermediate feature space, and use another classifier taking these values as inputs for making the final prediction.

[2] We assume at this point that all labelers give full labels to keep the notations simple. We will discuss the case of missing labels in Section 3.5.

3.3 Combination of Evidence from Crowds and Experts

The next step is to train a classifier in the intermediate feature space by utilizing expert labels. As expert labels are much more reliable than crowd labels, we should put more weights on them. However, if we discard crowd labels and use expert labels solely, building a stable model can be costly even in the more compact and representative intermediate feature space. Thus a balance has to be made between the crowd opinions and expert evidence.

We address the problem by imposing a Bayesian treatment on the model parameters of the classifier in the intermediate feature space. We use some straightforward combination of personal classifiers as the prior distribution of model parameters, and absorb expert label evidence by updating the posterior distribution sequentially. We believe that a fully Bayesian method is essential here for utilizing the prior distribution on parameters, which is informative in our framework as we will show later.

Specifically, we use the Bayesian logistic regression model [7] as our classifier in the intermediate feature space. The model achieved a tractable approximation of the posterior distribution over parameter $\boldsymbol{w}$ in Equation (1) by using accurate variational techniques. In our problem, the decision profile $\boldsymbol{dp}_i$ in the new space corresponding to the instance $\boldsymbol{x}_i$ in the original space is an $(L+1)$-dimensional vector consisting of all values of $dp_i^\ell, \ell = 1, \ldots, L$ and an additional constant 1 corresponding to the bias in parameter $\boldsymbol{w}$. The corresponding true label is y_i^0. The model assumes that the prior distribution over $\boldsymbol{w}$ is Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Absorbing the evidence of expert-labeled instance $\boldsymbol{dp}$ and the true label y amounts to updating the mean and covariance matrix by

$$\boldsymbol{\Sigma}_{post}^{-1} = \boldsymbol{\Sigma}^{-1} + 2|\lambda(\xi)|\boldsymbol{dp} \cdot \boldsymbol{dp}^{\mathrm{T}} \tag{2}$$

$$\boldsymbol{\mu}_{post} = \boldsymbol{\Sigma}_{post}[\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + (y/2)\boldsymbol{dp}] \tag{3}$$

where $\lambda(\xi) = [1/2 - \boldsymbol{\sigma}(\xi)]/2\xi$ and $\xi = [\boldsymbol{dp}^{\mathrm{T}}\boldsymbol{\Sigma}_{post}\boldsymbol{dp} + (\boldsymbol{dp}^{\mathrm{T}}\boldsymbol{\mu}_{post})^2]^{0.5}$. The update process is iterative and converges very fast (about two iterations) [7].

While one common criticism of the Bayesian approach is that the prior distribution is often selected on the basis of mathematical convenience rather than as a reflection of any prior beliefs [2], the prior distribution here is informative with a specific mean and an isotropic covariance matrix given by

$$\boldsymbol{\mu} = [-\frac{1}{2}, \frac{1}{L}, \ldots, \frac{1}{L}]^{\mathrm{T}} \tag{4}$$

$$\boldsymbol{\Sigma} = \alpha^{-1}\boldsymbol{I} \tag{5}$$

The mean is chosen such that all personal classifiers are combined by weighting them equally, and the bias is -0.5 to fit the shape of the logistic sigmoid function which is equal to 0.5 for $a = 0$.

There is a single precision parameter α governing the covariance matrix. We can interpret α as our confidence on the crowds. A large α will cause the prior distribution over $\boldsymbol{w}$ to peak steeply on the mean, thus the affect of absorbing one expert label will

Algorithm 1. Learning from crowd labelers and experts

```
Input: Crowd and expert training sets $\mathcal{D}$ and $\mathcal{D}_0$;
Train the ensemble $\mathcal{F}$ of logistic regression classifiers defined by Equation (1) using $\mathcal{D}_\ell$ where $\ell = 1, \dots, L$;
Use $\mathcal{F}$ to get predictions of data instances in $\mathcal{D}_0$, collect results in $\mathcal{DP}$;
Initialize $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by Equations (4) & (5);
for $j$=1 to $N_0$ do
    Calculate $\boldsymbol{\mu}_{post}$ and $\boldsymbol{\Sigma}_{post}$ by Equations (2) & (3) using the evidence from $\boldsymbol{dp}_j$ and $y_j^0$;
    Set $\boldsymbol{\mu}$ = $\boldsymbol{\mu}_{post}$, $\boldsymbol{\Sigma}$ = $\boldsymbol{\Sigma}_{post}$;
end for
Output: Personal classifier ensemble $\mathcal{F}$, mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$;
```

be relatively small, leading to a final classifier depending heavily on the mean of prior, which is the simple combination of personal classifiers. On the other hand, a small α means that the prior is close to uniform, causing the final classifier to make predictions mainly based on expert labels.

Intuitively, we should use a large α when personal classifiers are generally good, and use a small one when crowd labels are inaccurate. In a crowdsourcing scenario however, we usually do not have such knowledge. One alternative is to let α be related to the number of expert labels N_0 given by $\alpha = 1/N_0$. As this number increases, we decrease the confidence on crowds to let the final model put more weight on expert labels. Experiments show that with such selection of α, our model achieves relatively stable performance under various values of N_0.

Once the prior over $\boldsymbol{w}$ is chosen, we update its posterior distribution sequentially with Equations (2) & (3) by adding one expert label each time. If an instance $\boldsymbol{x}_j$ labeled by the expert is not in $\mathcal{D}$, we should first calculate its predictions $\boldsymbol{dp}_j$ by personal classifiers and use these values to update the model. We collect all $\boldsymbol{dp}_j$ in a set $\mathcal{DP} = \{\boldsymbol{dp}_j\}_{j=1}^{N_0}$. The complete steps of learning our model are summarized in Algorithm 1.

3.4 Classification

To classify a new coming instance $\boldsymbol{x}_k$ using the above results, we firstly get the predictions $\boldsymbol{dp}_k$ of personal classifiers on $\boldsymbol{x}_k$, and calculate the predictive distribution of the true label y_k^0 in the intermediate feature space by marginalizing w.r.t. the final distribution $\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The predictive likelihood is given by

$$\begin{aligned}\log P(y_k^0|\boldsymbol{x}_k, \mathcal{D}, \mathcal{D}_0) = \log \boldsymbol{\sigma}(\xi_k) - \frac{1}{2}\xi_k - \lambda(\xi_k)\xi_k^2 \\ -\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k + \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma}|}\end{aligned} \tag{6}$$

where subscript k assigned to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ refers to the posterior distribution over $\boldsymbol{w}$ after absorbing the evidence of $\boldsymbol{dp}_k$ and y_k^0.

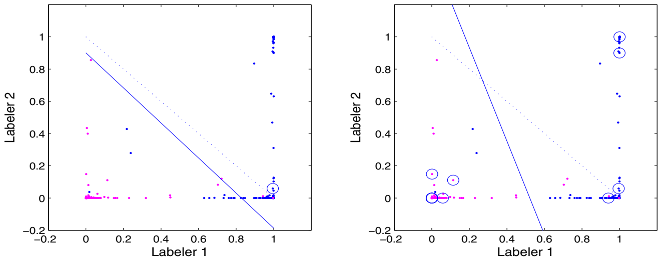

Fig. 2. Decision boundaries before and after absorbing expert label evidence. Dotted lines are means of prior distributions over $\boldsymbol{w}$, and solid lines are means of posterior distributions respectively after absorbing the label information of the circled instance(s) .

3.5 Missing Labels

In real crowdsourcing tasks, workers may label part of the instances instead of the whole set. Our model handles this problem naturally by training multiple personal classifiers independently. A worker only labels a few instances may lead to a pool personal classifier. But this is not fatal as he uses only a tiny proportion of the whole budget. Also in practice, we can avoid such cases simply by designing HITs with a moderate size.

4 Experiments

We use synthetic data to illustrate the process of absorbing expert evidence, and evaluate the performance of our method on both UCI benchmark data and real crowdsourcing data.

4.1 Synthetic Data

We complete our example in Figure 1 by illustrating the process of absorbing expert labels, shown in Figure 2. For clarity, we only show the decision boundaries given by means of the distributions over model parameter $\boldsymbol{w}$. Dotted lines are priors before adding expert labels. This prior is given by weighting each labeler equally following our framework.

In the left sub-figure, we add one expert label and get the posterior. Since the true label is blue, the decision boundary moves downward to suggest that data points near this labeled instance is more likely to be blue. In the right sub-figure, we add four expert labels for each class. The final decision boundary separates the actual class very well using merely eight expert labels. In this experiment we adjusted the model parameter α to get the best illustrative effect.

Table 1. Results on Waveform 1

Classifier	A_1	A_2	A_3
GT	0.853± 0.010		
MV	0.408± 0.123	0.638± 0.074	0.831± 0.007
AOC	0.490± 0.153	0.547± 0.101	0.831± 0.006
ML	0.437± 0.190	0.743± 0.063	0.842± 0.009
EL-10	0.718± 0.046	0.740± 0.045	0.740± 0.059
PCE-10	0.737± 0.051	0.742± 0.043	0.732± 0.051
CCE-10	0.725± 0.075	0.740± 0.068	0.816± 0.032
EL-20	0.756± 0.037	0.759± 0.034	0.783± 0.034
PCE-20	0.755± 0.033	0.768± 0.037	0.773± 0.048
CCE-20	0.801± 0.056	0.812± 0.057	0.822± 0.023
EL-50	0.792± 0.028	0.788± 0.033	0.798± 0.014
PCE-50	0.799± 0.025	0.796± 0.037	0.805± 0.016
CCE-50	0.816± 0.027	0.780± 0.039	0.833± 0.007
EL-100	0.797± 0.023	0.782± 0.023	0.767± 0.046
PCE-100	0.803± 0.008	0.811± 0.010	0.807± 0.015
CCE-100	0.831± 0.017	0.830± 0.024	0.829± 0.014

Table 2. Results on Spambase

Classifier	A_1	A_2	A_3
GT	0.924± 0.008		
MV	0.477± 0.327	0.641± 0.228	0.885± 0.013
AOC	0.535± 0.302	0.578± 0.208	0.879± 0.013
ML	0.510± 0.357	0.711± 0.302	0.925± 0.007
EL-10	0.672± 0.057	0.606± 0.113	0.665± 0.069
PCE-10	0.592± 0.095	0.641± 0.083	0.770± 0.049
CCE-10	0.857± 0.035	0.755± 0.165	0.890± 0.022
EL-20	0.860± 0.025	0.758± 0.033	0.755± 0.047
PCE-20	0.764± 0.080	0.708± 0.062	0.799± 0.046
CCE-20	0.891± 0.025	0.802± 0.087	0.894± 0.016
EL-50	0.830± 0.041	0.826± 0.032	0.831± 0.051
PCE-50	0.820± 0.053	0.803± 0.028	0.850± 0.017
CCE-50	0.900± 0.017	0.860± 0.053	0.895± 0.013
EL-100	0.860± 0.025	0.859± 0.025	0.858± 0.034
PCE-100	0.856± 0.025	0.861± 0.017	0.883± 0.010
CCE-100	0.891± 0.025	0.879± 0.031	0.903± 0.010

4.2 UCI Data

We test our method on three data sets from UCI Machine Learning Repository [6], Waveform 1(5000 points, 21 dimensions), Wine Quality(6497 points, 12 dimensions) and Spambase(4601 points, 57 dimensions). These data sets have moderate sizes which enable us to perform experiments when number of crowd labels varies.

Since multiple labelers for these UCI datasets are unavailable, we simulate L labelers for each dataset. We firstly cluster the data into L clusters using k-means and assign some labeling accuracy to each cluster for every labeler. Thus each labeler can have different labeling qualities for different clusters. We use an $L \times L$ matrix $\boldsymbol{A} = [a_{ij}]_{L\times L}$ to express the simulation process, in which a_{ij} is the probability that labeler i gives the true label for an instance in the jth cluster, thus a row corresponds to a labeler and a column to a cluster. We set $L = 5$ and use three different accuracy matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, and $\boldsymbol{A}_3$ to simulate different situations of labelers as follows.

$$\boldsymbol{A}_1 = \begin{bmatrix} 0\,1\,0\,1\,0 \\ 1\,1\,0\,0\,1 \\ 0\,0\,1\,1\,0 \\ 1\,0\,0\,1\,1 \\ 1\,0\,1\,1\,0 \end{bmatrix}, \boldsymbol{A}_2 = \begin{bmatrix} 0.3\;0.1\;0.8\;0.8\;0.8 \\ 0.3\;0.8\;0.1\;0.8\;0.8 \\ 0.3\;0.8\;0.8\;0.1\;0.8 \\ 0.3\;0.8\;0.8\;0.8\;0.1 \\ 0.8\;0.1\;0.1\;0.1\;0.1 \end{bmatrix}, \boldsymbol{A}_3 = \begin{bmatrix} 0.55\;0.55\;0.55\;0.55\;0.55 \\ 0.65\;0.65\;0.65\;0.65\;0.65 \\ 0.75\;0.75\;0.75\;0.75\;0.75 \\ 0.68\;0.68\;0.68\;0.68\;0.68 \\ 0.95\;0.95\;0.95\;0.95\;0.95 \end{bmatrix}.$$

$\boldsymbol{A}_1$ simulates severely biased labelers. $\boldsymbol{A}_2$ simulates labelers whose labels are both noisy and biased. $\boldsymbol{A}_3$ simulates simply noisy labels. Note that $\boldsymbol{A}_3$ satisfies the model assumption in the work by Raykar *et al.* [15].

We choose three baseline methods that learn with crowd data solely for comparison. To verify the ability of our method to utilize the crowd labels, we compare the results trained on expert labels solely. For comparison with existing methods we use the model proposed by Kajino *et al.* [9], which is a state-of-art model that addresses the same problem. We use the results trained on the original datasets which have all ground truth labels as the approximate upper bounds of the classification performance. Methods used in experiments are summarized as follows.

Table 3. Results on Wine Quality

Classifier	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$
GT	0.743± 0.010		
MV	0.424± 0.119	0.571± 0.110	0.739± 0.007
AOC	0.582± 0.118	0.500± 0.109	0.740± 0.004
ML	0.417± 0.133	0.701± 0.020	0.739± 0.004
EL-10	0.550± 0.042	0.583± 0.047	0.582± 0.083
PCE-10	0.591± 0.035	0.578± 0.063	0.613± 0.033
CCE-10	0.634± 0.078	0.651± 0.092	0.715± 0.022
EL-20	0.623± 0.064	0.575± 0.075	0.623± 0.063
PCE-20	0.629± 0.019	0.604± 0.047	0.642± 0.041
CCE-20	0.679± 0.042	0.688± 0.047	0.720± 0.022
EL-50	0.666± 0.036	0.675± 0.040	0.682± 0.024
PCE-50	0.648± 0.019	0.644± 0.011	0.662± 0.022
CCE-50	0.687± 0.038	0.707± 0.025	0.722± 0.019
EL-100	0.707± 0.017	0.706± 0.017	0.711± 0.016
PCE-100	0.666± 0.011	0.665± 0.009	0.685± 0.020
CCE-100	0.718± 0.017	0.720± 0.010	0.733± 0.006

Table 4. Results under the variation of crowd label numbers on Spambase

Num.	50	100	200	500	1000
GT	0.826± 0.037	0.855± 0.023	0.878± 0.022	0.900± 0.009	0.913± 0.003
EL-50	0.835± 0.032				
MV	0.757± 0.057	0.734± 0.039	0.770± 0.021	0.849± 0.012	0.884± 0.012
AOC	0.587± 0.045	0.727± 0.025	0.798± 0.025	0.852± 0.010	0.880± 0.010
ML	0.807± 0.065	0.822± 0.025	0.876± 0.012	0.905± 0.005	0.921± 0.005
PCE-50	0.838± 0.025	0.839± 0.024	0.842± 0.016	0.837± 0.018	0.861± 0.026
CCE-50	0.792± 0.045	0.807± 0.045	0.820± 0.018	0.873± 0.008	0.903± 0.006

- **Majority Voting (MV)** method learns from the single-labeled training set estimated by majority voting.
- **All-in-One-Classifier (AOC)** treats all labels as in one training set.
- **Multiple Labelers (ML)** method [15] learns from crowd labels directly.
- Kajino *et al.* [9] extended their personal classifier model [8] to incorporate expert labels, which we refer to as **Personal Classifiers with Experts (PCE)**. **PCE-**N_0 is the results trained with N_0 expert labels.
- We refer to our method as **Classifier Combination with Experts (CCE)**. **CCE-**N_0 is the results after absorbing the evidence of N_0 expert labels.
- Training with expert labels solely is referred as **Expert Labels (EL)** classifiers. **EL-**N_0 is the results trained with N_0 expert labels.
- **Ground Truth (GT)** classifier uses the original datasets for training.

For MV, AOC, GT, and EL, we use a logistic regression model respectively to train the classifiers. For PCE, CCE and EL, the set of expert labels are randomly chosen from the original datasets given the number of expert labels N_0 which is restricted to a small proportion of N. We divide each dataset into a 70% training set and a 30% test set and each result is averaged on 10 runs.

Tables 1–3 show the results for different datasets respectively. Results are in the form of classification accuracy and averaged on 10 trials. The GT classifier is independent of crowd labels thus it has only one result on each dataset. Our CCE outperforms EL, and also outperforms MV, ACL and ML in most cases. This validates the ability of CCE for combining crowd and expert labels. The only exception appears in ML under $\boldsymbol{A}_3$ where labelers are not biased. CCE outperforms PCE with clear advantages. There are a number of cases that PCE performs worse than EL, which suggests that in the PCE model expert evidences are easily disturbed by inaccurate crowd labels.

Table 4 shows the results under the variation of numbers of crowd labels. We show the results on Spambase data under $\boldsymbol{A}_3$ since under this situation all methods seem to work well. The top number of each column represents the number of labels provided by each labeler. This is also the number of expert labels used for GT. We use 50 expert labels for EL, PCE and CCE. EL has only one result as it is independent of crowd labels.

There is no surprise that ML performs very well in this experiment as the configuration here meets ML's model assumption. Yet we should not forget that ML fails in many cases as shown in Tables 1–3. We do not choose those cases because showing groups of failed results does not make any sense. Generally PCE and CCE outperform MV and AOC by using extra expert labels. CCE performs slightly worse than PCE when the number of crowd labels is small, while the performance raise of PCE is quite limited when using more crowd labels.

In summary, our method CCE achieved reasonable performance on different data sets with various labeler properties. The accuracy and stability of our CCE increase as we use more expert labels. On the other hand, learning solely from crowd labels is risky, especially when crowd labels are biased. PCE's performance is limited compared with CCE when we have enough crowd labels.

4.3 Affective Text Analysis Data

In this section we show results on the data for affective text analysis collected by Snow *et al.* [17]. The data is collected from Amazon Mechanical Turk. Annotators are asked to rate the emotions of a list of short headlines. The emotions include anger, disgust, fear, joy, sadness, surprise and the overall positive or negative valence. The former six are expressed with an interval $[0, 100]$ respectively while valence is in $[-100, 100]$. There is a total number of 100 headlines labeled by 38 workers. For each headline 10 workers rated for each of the seven emotions. Most workers labeled 20 or 40 instances thus more than one half labels are missing. All 100 instances are also labeled by the experts and have an average rating for each emotion, which we treat as ground truth.

We design the classification task which predicts the surprising level of a headline using other emotion ratings as features. We define that a headline of which the surprise rating is above 20 is a surprise, while others not, and use ratings of other six emotions provided by the experts to express a headline. Thus we get a binary classification task in a 6-dimensional feature space.

Figure 3 shows classification accuracy when continually adding expert labels. Results of MV, AOC and ML are not shown in this figure, which are three horizontal lines below GT and stay close to each other. PCE only performs similarly with EL, which collapses to GT when using all expert labels.

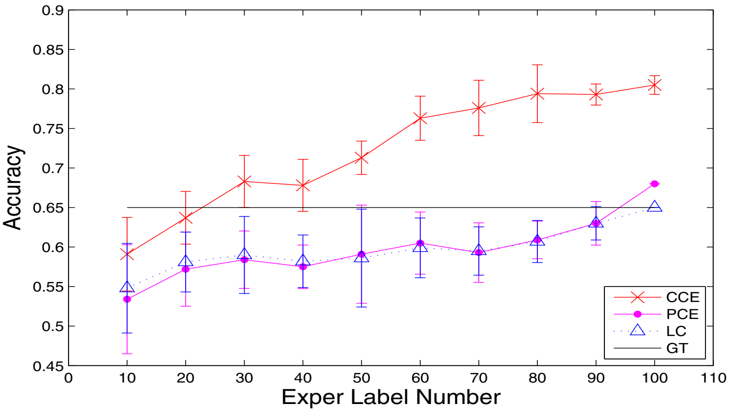

Fig. 3. Results on Affective Text Analysis data. The x-axis is the number of expert labels used while the y-axis is the classification accuracy.

The result of CCE is promising. The value of GT is 0.65, which suggests that according to the experts, there is no strong correlation between the surprising level and other emotions. However, CCE only uses about 20 expert labels to get a similar performance level with GT, and when adding more expert labels, CCE outperforms GT and achieves an accuracy up to 0.8. We attribute this fact to the power of our CCE model as a 'feature extractor'. Among the 38 workers, one or more of them did give ratings in manners that relate surprising levels to other emotions even if experts did not do so. Personal classifiers trained from these workers will then be able to predict the target and our model identifies these classifiers successfully using expert labels.

5 Conclusion and Future Work

In this paper, we have proposed a framework for improving the performance of crowdsourcing learning tasks by incorporating the evidence of expert labels with a Bayesian logistic regression classifier in the intermediate feature space. Experimental results have verified that by combining crowd and expert labels, our method has achieved better performance as compared with some existing methods, and has been stable under the variation of the number of expert labels and crowd labeler properties.

A promising direction of future work is to investigate actively querying for the expert labels, for which we can develop models by adopting basic ideas from active learning and considering the particular situation of crowdsourcing.

Acknowledgment. This work is supported by the National Key Technology R&D Program of the Chinese Ministry of Science and Technology under Grant No. 2012BAH94F03.

References

1. Ambati, V., Vogel, S., Carbonell, J.: Active learning and crowd-sourcing for machine translation. Language Resources and Evaluation (LREC) 7, 2169–2174 (2010)
2. Bishop, C.M., et al.: Pattern recognition and machine learning, vol. 4. Springer, New York (2006)
3. Brew, A., Greene, D., Cunningham, P.: Using crowdsourcing and active learning to track sentiment in online media. In: ECAI 2010, pp. 145–150 (2010)
4. Dawid, A.P., Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm. Applied Statistics, 20–28 (1979)
5. Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., Dredze, M.: 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, pp. 80–88. Association for Computational Linguistics (2010)
6. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
7. Jaakkola, T., Jordan, M.: A variational approach to Bayesian logistic regression models and their extensions. In: Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics (1997)
8. Kajino, H., Tsuboi, Y., Kashima, H.: A convex formulation for learning from crowds. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence (2012) (to appear)
9. Kajino, H., Tsuboi, Y., Sato, I., Kashima, H.: Learning from crowds and experts. In: Proceedings of the 4th Human Computation Workshop, HCOMP 2012 (2012)
10. Karger, D.R., Oh, S., Shah, D.: Iterative learning for reliable crowdsourcing systems. In: Advances in Neural Information Processing Systems (NIPS 2011), pp. 1953–1961 (2011)
11. Kuncheva, L.I.: Combining pattern classifiers: Methods and algorithms (kuncheva, li; 2004)[book review]. IEEE Transactions on Neural Networks 18(3), 964 (2007)
12. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition 34(2), 299–314 (2001)
13. Liu, Q., Peng, J., Ihler, A.: Variational inference for crowdsourcing. In: Advances in Neural Information Processing Systems (NIPS 2012), pp. 701–709 (2012)
14. Merz, C.J.: Using correspondence analysis to combine classifiers. Machine Learning 36(1), 33–58 (1999)
15. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. The Journal of Machine Learning Research 11, 1297–1322 (2010)
16. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get another label? Improving data quality and data mining using multiple, noisy labelers. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 614–622 (2008)
17. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: 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, pp. 254–263. Association for Computational Linguistics (2008)
18. Wauthier, F.L., Jordan, M.I.: Bayesian bias mitigation for crowdsourcing. In: Advances in Neural Information Processing Systems (NIPS 2011), pp. 1800–1808 (2011)
19. Yan, Y., et al.: Modeling annotator expertise: Learning when everybody knows a bit of something. In: Proceedings of 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010), vol. 9, pp. 932–939 (2010)