Uncovering the Latent Structures of Crowd Labeling

Tian Tian and Jun Zhu

State Key Lab of Intelligent Technology & Systems; Tsinghua National TNLIST Lab,
Department of Computer Science & Technology, Tsinghua University,
Beijing 100084, China
{tiant13@mails,dcszj@mail}.tsinghua.edu.cn

Abstract. Crowdsourcing provides a new way to distribute enormous tasks to a crowd of annotators. The divergent knowledge background and personal preferences of crowd annotators lead to noisy (or even inconsistent) answers to a same question. However, diverse labels provide us information about the underlying structures of tasks and annotators. This paper proposes latent-class assumptions for learning-from-crowds models, that is, items can be separated into several latent classes and workers' annotating behaviors may differ among different classes. We propose a nonparametric model to uncover the latent classes, and also extend the state-of-the-art minimax entropy estimator to learn latent structures. Experimental results on both synthetic data and real data collected from Amazon Mechanical Turk demonstrate our methods can disclose interesting and meaningful latent structures, and incorporating latent class structures can also bring significant improvements on ground truth label recovery for difficult tasks.

1 Introduction

Researches and applications in the field of artificial intelligence are relying more and more on large-scale datasets as the age of Big-data comes. Conventionally, labels of tasks are collected from domain experts, which is expensive and time-consuming. Recently, online distributed working platforms, such as Amazon Mechanical Turk (MTurk) , provide a new way to distribute enormous tasks to a crowd of workers [1]. Each worker only needs to finish a small part of the entire task in this crowd labeling mode, so that the tasks can be done faster and cheaper. However, the labels given by the crowd annotators are less accurate than those given by experts. In order to well recover the true labels, multiple annotators are usually needed to evaluate every micro task. Furthermore, different annotators may have different backgrounds and personal preferences, and they may give inconsistent answers to a same question. This phenomenon brings us more difficulties to recover ground truth labels from noisy answers and raises a research topic in the crowdsourcing area.

On the other hand, the diverse labels can provide us with a lot of additional information for both data characteristics and people's behaviors [2]. For example, they may reflect some latent structures of the complicated data, such as the

grouping structure of tasks according to their difficulty levels and/or the grouping structure of annotators according to their similar education background or preferences. In the perspective of psychology, users' labels actually show their understanding of the given tasks. For example, in a problem of classifying flowers in pictures, users' choices may be influenced by many different features, such as petal color, petal shape, background, size in the picture, etc; and personal choices of different users are influenced by users' tastes. These features are usually unknown. Some features are significantly related to the flower species and some features are not. So we think the observed user labels are generated from tasks' latent structures and annotators' abilities, but not directly from the truth category. By exploring these latent structures, we can have a better understanding of the data, and may also accomplish tasks like category recovery better.

Dawid and Skene's work [3] is a milestone in learning from crowds. They proposed an annotator-specific confusion matrix model, which is able to estimate the ground truth category well. Raykar et al. [4] extended Dawid and Skene's model by ways, such as taking item features into account or modifying the output model to fit regression or ranking tasks. Zhou et al. [5,6] proposed a minimax entropy estimator, which outperforms most previous models in category estimating accuracy, and later on they extended their model to handle ordinal labels. However, none of these models have taken latent structures into account. We extend some of them to learn latent structures from dataset. Welinder et al. [7] proposed a multidimensional annotation model, which was the earliest to consider latent structure in this field. But this model often suffers from overfitting and so performs averagely on many tasks [8]. Tian and Zhu [9] also proposed an idea on the latent structure for crowdsourcing but aimed at a different problem; our work draws some inspiration from their nonparametric ideas.

We propose two latent-class assumptions for learning from crowds: **(I)** each item belongs to one latent class, and annotators have a consistent view on items of the same class but maybe inconsistent views on items of different classes; and **(II)** several different latent classes consist in one label category. To recover the latent-class structures, we propose a latent class estimator using a nonparametric prior. We also extend the minimax entropy estimator to fine tune such latent class structures. Under the latent class assumptions, the estimators remain compact through parameter sharing. The experimental results on both synthetic and real MTurk datasets demonstrate our methods can disclose interesting and meaningful latent structures, and incorporating latent class structures can bring significant improvements on ground truth label recovery for difficult tasks. We summarize our contributions as: **(1)** We propose the latent-class assumptions for crowdsourcing tasks. **(2)** We develop appropriate nonparametric algorithms for learning latent-class structures, and extend previous minimax entropy principle. **(3)** We present an algorithm to recover category labels from latent classes, and empirically demonstrate its efficiency.

The rest paper of the is structured as follows. Sec. 2 describes related crowdsourcing models. Sec. 3 introduces latent-class assumptions and provides details

of our latent class models. Sec. 4 presents category recovery methods. Sec. 5 shows empirical results for latent class and category recovery. Sec. 6 concludes.

2 Preliminaries

We introduce three major methods for label aggregation in learning from crowds. We focus on classification tasks in this paper. In a dataset consisting of M items (e.g., pictures or paragraphs), each item m has a specific label Y_m to denote its affiliated category. $\boldsymbol{Y}$ is the collection of these ground truth category labels, and all the possible label values form a set $\boldsymbol{D}$. To obtain the unknown ground truth, we have N workers examine the dataset. W_{nm} is the label of item m given by worker n. $\boldsymbol{W}$ is the collection of these workers' labels. $\boldsymbol{I}$ is the collection of all worker-item index pairs corresponding to $\boldsymbol{W}$. The goal of learning from crowds is to infer the values of $\boldsymbol{Y}$ from the observations of $\boldsymbol{W}$.

2.1 Majority Voting (MV)

The simplest label aggregation model is the majority voting. This method assumes that: For every worker, the ground truth label is always the most common to be given, and the labels for each item are given independently. From this point of view, we just need to find the most frequently appeared label for each item. We use $q_{md} = P(Y_m = d)$ to denote the probability that the mth task has true label d, then

$$q_{md} = \frac{\sum_{(n,m)\in \boldsymbol{I}} \delta_{W_{nm},d}}{\sum_{d,(n,m)\in \boldsymbol{I}} \delta_{W_{nm},d}}, \forall m, \tag{1}$$

where $\delta_{\cdot,\cdot}$ is an indicator function: $\delta_{a,b}$ equals to 1 whenever $a = d$ is true, otherwise it equals to 0. The estimated label is represented by $Y_m = \max_d q_{md}, \forall m$.

2.2 Dawid-Skene Estimator (DS)

Dawid and Skene [3] proposed a probabilistic model, which is widely used in this area. They made an assumption that: The performance of a worker is consistent across different items, and his or her behavior can be measured by a confusion matrix. Diagonal entries of the confusion matrix indicate the probability that this worker gives correct labels; while off-diagonal entries indicate that this worker makes specific mistakes to label items in one category as another. Extensive analysis of this model's error bound has been presented [10,11].

More formally, we use $\boldsymbol{p}_n$ to denote the confusion matrix of worker n, with each element p_{ndl} being the probability that worker n gives label l to an item when the ground truth of this item is d. We use q_d to denote the probability that an item has the ground truth label d. Under these notations, parameters of workers can be estimated via a maximum likelihood estimator, $\{\hat{\boldsymbol{q}}, \hat{\boldsymbol{p}}\} = \operatorname{argmax} P(\boldsymbol{W}|\boldsymbol{q}, \boldsymbol{p})$, where the margined likelihood is

$$P(\boldsymbol{W}|\boldsymbol{q}, \boldsymbol{p}) = \prod_m \Big(\sum_d q_d \prod_{n,l} {p_{ndl}}^{\delta_{W_{nm},l}} \Big), \tag{2}$$

by marginalizing out the hidden variables $\boldsymbol{Y}$. This problem is commonly solved using an EM algorithm.

2.3 Minimax Entropy Estimator (ME)

Minimax entropy estimator [5,6] is another well-performing method which combines the idea of majority voting and confusion matrix. This model assumes that: Labels are generated by a probability distribution over workers, items, and labels; and the form of the probability distributions can be estimated under the maximum entropy principle. For example, $\boldsymbol{p}_{nm}$ is a probability distribution on the label of item m given by worker n. To incorporate the idea of majority voting that ground truth labels are always the most common labels to be given, the count of empirical observations that workers give an item a certain label should match the sum of workers' probability corresponding to these observations within the model. So they come up with the first type of constraints:

$$\sum_{n} p_{nmd} = \sum_{n} \delta_{W_{nm},d}, \forall m, d. \tag{3}$$

To combine the confusion matrix idea that a worker is consistent across different items in the same category, the count of empirical observations that workers give items in the same category a certain label should match the sum of workers' probability corresponding to these observations within the model. So there is another type of constraints:

$$\sum_{\substack{m \\ s.t. Y_m = d}} p_{nmd} = \sum_{\substack{m \\ s.t. Y_m = d}} \delta_{W_{nm},d}, \forall n, d. \tag{4}$$

Under these constraints and the minimax entropy principle, we choose $\boldsymbol{Y}$ to minimize the entropy but choose $\boldsymbol{p}$ to maximize the entropy. This rationale leads to the learning problem:

$$\min_{\boldsymbol{Y}} \max_{\boldsymbol{p}} - \sum_{n,m,d} p_{nmd} \log p_{nmd}, \tag{5}$$

subject to constraints (3) and (4). In practice, hard constraints can be strict. Therefore, soft constraints with slack variables are introduced to the problem.

3 Extend to Latent Classes

Both DS and ME use specific probabilities to represent workers' behaviors. However, we can dig deeper into the structure of the items. For example, in a flower recognition task, we ask workers to decide whether the flower in a given picture is peach flower or not. When the standard DS estimator is used, the confusion matrix should contain 4 probabilities, that is, the probability that worker labels the picture correctly when it is peach flower; the probability that worker labels the picture incorrectly when it is peach flower; the probability that worker labels the picture correctly when it is not peach flower; and the probability that worker

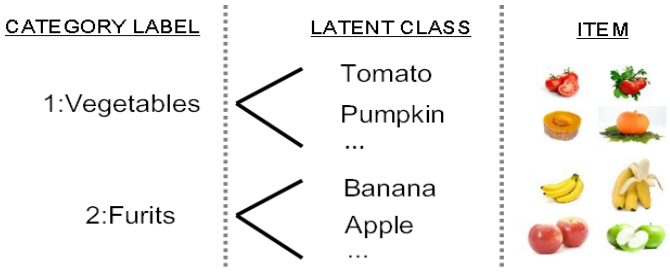


Fig. 1. Illustration for categories and latent classes of a vegetable vs. fruit classification.

labels the picture incorrectly when it is not peach flower. If there are 2 breeds of peach flowers in the testing set, say mountain peach flowers and flowering peach flowers, then the probabilities that a worker recognizes them as peach flowers correctly might be different. For example, some workers who are very familiar with mountain peach may point out mountain peach flowers as peach flowers with an extraordinary high accuracy, but their accuracy of recognizing flowering peach might be close to random guess. Our experiments show that this phenomenon does exist. So we come to one conclusion that the latent structure of items can affect the workers' labeling results significantly, and we can take this influence into account in our label aggregation algorithm. *Latent class structure* is one of the simplest latent structures of items. The latent class here refers to a finer level structure of items than the category. In the flower example, the latent classes may correspond to the flower species such as flowering peach and mountain peach, while the categories can only recognize both these species as peach flower with no inner structure. If we restrict the number of latent classes to be the same as the number of categories, different classes will naturally correspond to the classification categories. Yet as a general rule, the number of latent classes should be larger than the category number.

A category of items might be divided into several latent classes, but a latent class belongs to one specific category only. Thus, we make two basic assumptions in the crowd labeling situations:

- **Assumption I.** Each item belongs to one specific latent class only.
- **Assumption II.** Items in a same latent class belong to a same category.

From another point of view, we believe that no label is spam. When the standards of solving our problems match the workers' own criterion, based on which they make their choices, the DS estimator works well. But if they do not, much information will be left unutilized by this estimator. In order to improve the aggregation performance and uncover more information hiding behind the noisy labels, we build up new models which take latent structures into account.

3.1 Nonparametric Latent Class Estimator (NDS)

For the DS estimator, a confusion matrix is used to measure workers' behavior, with each entry p_{ndl} representing the probability that worker n gives label l to an item when the ground truth of this item is d. Now we realise that the latent classes affect the output labels directly. We can replace the category dimension of

the confusion matrix representation with the latent class dimension. Therefore, we have a latent class version confusion matrix $\boldsymbol{p}_n$ for each worker. An entry p_{nkl} denotes the probability that worker n gives label l to an item which belongs to latent class k. Similarly we use Z_m to represent the latent class that item m belongs to, and use $\boldsymbol{q}$ to denote the probability that each latent class appears, so that q_k denotes the probability that an item belongs to latent class k.

Probabilistic Model Since it is hard to decide the number of latent classes K in advance, we put a nonparametric prior on the latent class assignment variable $\boldsymbol{Z}$, which can represent arbitrary number of classes. The *Chinese restaurant process* (CRP) is used here, it is a construction of Dirichlet process [12], and can be described using the metaphor of a restaurant with customers entering and sitting next to tables with different probabilities depending on the tables'jj relative sizes. α_c is the discount parameter of this process. We also put a Dirichlet prior Dirichlet(α_d) on every $\boldsymbol{p_{nk}}$, where α_d is the concentration parameter. So the probabilistic model is represented as follow,

$$\boldsymbol{Z}|\alpha_c \sim \text{CRP}(\alpha_c), \qquad \boldsymbol{p_{nk}}|\alpha_d \sim \text{Dirichlet}(\alpha_d),\ \forall n, k, \tag{6}$$

$$W_{nm}|\boldsymbol{Z}, \boldsymbol{p}_{n\cdot} \sim \text{Multinomial}(\boldsymbol{A}_{nm}),\ \forall n, m, \tag{7}$$

where $\boldsymbol{A}_{nm} = \{A_{nm1}, \cdots, A_{nmD}\}$, and $A_{nmd} = \prod_{k=1}^{K} p_{nkd}{}^{\delta_{Z_m,k}}$. Here $\boldsymbol{W}$ is the given labels, $\boldsymbol{p}$ is the parameters to learn, and $\boldsymbol{Z}$ is the hidden variable. If annotator n do not give item m a label, the probabilities of all W_{nm} values are set to be one.

Conditional Distribution To infer their values, we build a Gibbs sampler to get samples from the joint posterior distribution. The conditional distribution for the confusion matrix parameter is

$$\begin{aligned} P(\boldsymbol{p}_{nk}|\boldsymbol{Z}, \boldsymbol{W}) &\propto P(\boldsymbol{p}_{nk}) \prod_{m=1}^{M} P(W_{nm}|\boldsymbol{Z}, \boldsymbol{p}_{nk}) \\ &\propto \Big(\prod_{d=1}^{D} p_{nkd}{}^{\alpha_d/D-1} \Big) \Big(\prod_{m=1}^{M} \prod_{d=1}^{D} p_{nkd}{}^{\delta_{W_{nm},d}\delta_{Z_m,k}} \Big). \end{aligned} \tag{8}$$

So the conditional distribution $\boldsymbol{p}_{nk}|\boldsymbol{Z}, \boldsymbol{W} \sim \text{Dirichlet}(\boldsymbol{p}_{nk}|\boldsymbol{B}_{nk}), \forall n, k,$ where $\boldsymbol{B}_{nk} = \{B_{nk1}, \cdots, B_{nkD}\}$, and $B_{nkd} = \sum_{m=1}^{M} \delta_{W_{nm},d}\delta_{Z_m,k} + \alpha_d/D$. As for the hidden variables, when $k \leq K$,

$$\begin{aligned} P(Z_m = k|\boldsymbol{Z}_{-m}, \boldsymbol{p}, \boldsymbol{W}) &\propto P(Z_m = k) \prod_{n=1}^{N} P(W_{nm}|Z_m = k, \boldsymbol{p}_{nk}) \\ &\propto n_k \prod_{n=1}^{N} \prod_{d=1}^{D} p_{nkd}{}^{\delta_{W_{nm},d}}, \end{aligned} \tag{9}$$

where n_k is the number of tasks that have latent class label k. When generating a new class,

$$P(\ Z_m = k_{new}|\boldsymbol{Z}_{-m}, \boldsymbol{p}, \boldsymbol{W}) \propto P(Z_m = k_{new}) \prod_{n=1}^{N} P(W_{nm}|Z_m = k_{new}) \tag{10}$$

$$\propto\ P(Z_m = k_{new}) \prod_{n=1}^{N} \int P(W_{nm}|Z_m = k_{new}, \boldsymbol{p}_{nk_{new}}) P(\boldsymbol{p}_{nk_{new}}) d\boldsymbol{p}_{nk_{new}}$$

$$\propto\ \alpha_c \prod_{n=1}^{N} \frac{\prod_{d=1}^{D} \Gamma(\delta_{W_{nm},d} + \alpha_d/D)}{\Gamma(1 + \alpha_d)}.$$

Then we can get samples from the posterior distribution of our model by iteratively updating hidden variables and parameters.

3.2 Latent Class Minimax Entropy Estimator (LC-ME)

Many existing estimators can be extended to learn latent class structures. The nonparametric latent class estimator can be regarded as an extension of DS estimator, we can also incorporate latent class structures into the minimax entropy estimator. Some constraints need to change for this extension, as detailed below.

We still assume that the ground truth label will always get more probability to be given by workers, so the first type constraints remain unchanged. As for the other constraints, now we apply the idea of latent class version DS estimator: When worker n deals with items in latent class k, he may label it as category d with a constant probability. So the constraints can be written as

$$\sum_{\substack{m \\ s.t. Z_m = k}} p_{nmd} = \sum_{\substack{m \\ s.t. Z_m = k}} \delta_{W_{nm},d}, \forall n, k. \tag{11}$$

To relax constraints, we introduce slack variables $\boldsymbol{\tau}$ and $\boldsymbol{\sigma}$ and their regularization terms. Under these new constraints, the optimization problem is slightly changed comparing with the previous version:

$$\min_{\boldsymbol{Z}} \max_{\boldsymbol{p},\boldsymbol{\tau},\boldsymbol{\sigma}} - \sum_{n,m,d} p_{nmd} \log p_{nmd} - \sum_{m,d} \frac{\alpha_m \tau_{md}^2}{2} - \sum_{n,m,d} \frac{\beta_n \sigma_{ndk}^2}{2}$$

$$s.t. \sum_{n} \left(p_{nmd} - \delta_{W_{nm},d}\right) = \tau_{md}, \forall m, d, \tag{12}$$

$$\sum_{m} \left(p_{nmd} - \delta_{W_{nm},d}\right) \delta_{Z_m,k} = \sigma_{ndk}, \forall n, k, \quad \sum_{d} p_{nmd} = 1, \forall n, m.$$

To solve this optimization problem, we update $\{\tau_{md}, \sigma_{ndk}\}$ and q_{mk} respectively. Since the inference procedure is similar to the original minimax entropy estimator in [5], we only express the final iterative formula here.

Step-1: we need to solve a simple sub-problem:

$$\{\tau_{md}^t, \sigma_{ndk}^t\} = \underset{\tau,\sigma}{\operatorname{argmin}} \sum_{n,k,d} q_{mk}^{t-1} \Big[\log \sum_d \exp(\tau_{md} + \sigma_{ndk}) - \sum_d (\tau_{md} + \sigma_{ndk}) \delta_{W_{nm},d} \Big] + \sum_{m,d} \frac{1}{2} \alpha_m \tau_{md}^2 + \sum_{n,m,d} \frac{1}{2} \beta_n \sigma_{ndk}^2, \forall n, m, d, k, \tag{13}$$

where $q_{mk}^t \propto P^t(Z_m = k)$ represents the probability that the item m is in latent class k. This optimization task can be solved by gradient descent and any other optimization methods.

Step-2: the probability distribution of each item's label is

$$q_{mk}^t \propto q_{mk}^{t-1} \prod_n \frac{\exp\left(\sum_d (\tau_{md}^t + \sigma_{ndk}^t) \delta_{W_{nm},d}\right)}{\sum_d \exp(\tau_{md}^t + \sigma_{ndk}^t)}, \forall m, k. \tag{14}$$

Iteratively updating $\{\tau_{md}, \sigma_{ndk}\}$ and q_{mk}, it will converge to a stationary point. Then we can get the latent class numbers $\boldsymbol{Z}$ by the peak positions of $\boldsymbol{q}$. Since the algorithm is sensitive to the initial point, we use the result of NDS as the latent class number K and the initial point $\boldsymbol{Z}$ of the LC-ME.

4 Category Recovery

In order to obtain the ground truth labels, we need to recover the category information from latent classes. According to our second basic assumption that each latent class belongs to one specific category, we can recover the ground truth labels by associating latent classes to categories.

A re-estimating method can be used here to recover the categories. After we get the latent class information for items, we can regard items in a same class as one imaginary item, here we call it a *hyper-item*. Then there are totally K hyper-items, every hyper-item may have several different labels by each worker. This setting has been considered in the original Dawid-Skene estimator.

We use a generalized Dawid-Skene estimator with hyper-items to estimate the category assignments, which solves a maximum likelihood estimation problem. The margined likelihood of given labels is

$$P(\boldsymbol{W}|\boldsymbol{q}, \boldsymbol{p}) = \prod_k \Big(\sum_d q_d \prod_{n,l} {p_{ndl}}^{n_{nkd}} \Big), \tag{15}$$

where $n_{nkd} = \sum_m \delta_{W_{nm},d} \delta_{Z_m,k}$ is the count of labels that worker n gives to hyper-item k. The EM algorithm for solving this problem also needs some modification. Specifically, we use C_k to represent the category of latent class k. Then in the **E-Step**, the probability distribution is

$$P(C_k = d|\boldsymbol{W}, \boldsymbol{q}, \boldsymbol{p}) \propto P(C_k = d) P(\boldsymbol{W}|C_k = d) \propto q_d \prod_{n,l} {p_{ndl}}^{n_{nkd}}, \tag{16}$$

and the estimated category of each latent class is $C_k = \max_d P(C_k = d|\boldsymbol{W}), \forall k$. In the **M-Step**, we have the update equations:

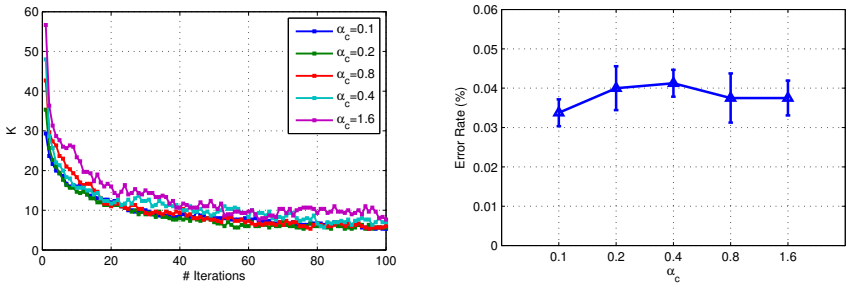
$$q_d = \frac{1}{K}\sum_k \delta_{C_k,d}, \qquad p_{ndl} = \frac{\sum_k n_{nkl}\delta_{C_k,d}}{\sum_{k,l} n_{nkl}\delta_{C_k,d}}. \tag{17}$$

5 Experiment Results

We now present experimental results to evaluate the performance of the proposed models on both one synthetic dataset and real dataset collected from MTurk. We present both quantitative results on ground truth label recovery and quantitative results on latent structure discovery, with comparison to various competitors.

5.1 Synthetic Dataset

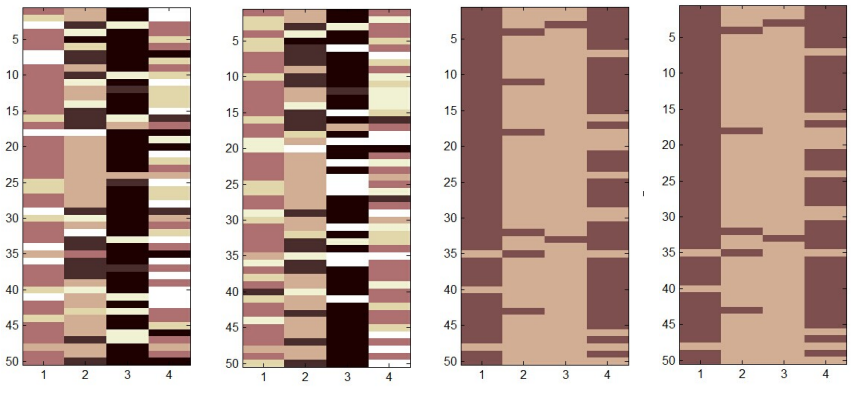
We designed a synthetic dataset to show the latent class recovery ability of each model. This dataset consists of 4 latent classes and 2 types of workers. We generated 40 items' parameters for each latent class and simulated 20 workers of each type. We set the confusion matrix for all simulating worker types and randomly sample labels. The probabilistic distribution values of different classes in the confusion matrices are dispersive, e.g. $[0.8, 0.2], [0.5, 0.5], [0.2, 0.8]$. So the effect of latent structure is more significant. The results on learning latent classes and category recovery are shown below.



(a) (b)

Fig. 2. Performance on synthetic dataset. (a) shows the numbers of latent classes found by NDS with different color. (b) shows the average category recovery error rates.

Sensitivity: We use the NDS model to recover the latent structure of this dataset. Fig. 2(a) shows the learnt latent class number K by models with different parameters. We set $\alpha_d = 2$ for all trials, and vary α_c from 0.1 to 1.60. We can see when parameter changes, the steady state value only changes a little, and all the values are close to the true latent class number. This result shows that our model is insensitive to the discount parameter. So when we use this model to learn latent structures for some purposes, we only need to find a rough range of the parameter with a validate dataset.



(a) NDS (b) LC-ME (c) NDS+CR (d) LC-ME+CR

Fig. 3. Latent class and category visualization. Each subfigure shows a 50×4 matrix, with each entry corresponding to a flower image and each column corresponding to a unique flower species, which is flowering peach, sakura, apricot and mountain peach from left to right. For (a) and (b), each color denotes one latent class. For (c) and (d), each color denotes a classification category. (a) and (c) are learned by NDS, (b) and (d) are learned by LC-ME. (best viewed in color).

Category Recovery: To evaluate the ground truth category recovery accuracy, we compare the error rates of NDS with different α_c. We can see from Fig. 2(b) that the final accuracy is insensitive to the parameter α_c, and it is about 3.75% for all parameter settings. We also compare the NDS with other methods. Majority voting achieves error rate 9.38%, original Dawid-Skene estimator achieves error rate 12.50%, both of them are worse than NDS.

5.2 Flowers Dataset

To show the semantic meaning of the latent structure learned by our models, we designed a flower recognition task and collected crowd labeling data from MTurk annotators. Four flower species, mountain peach flower, flowering peach flower, apricot flower and sakura, make up the dataset of 200 images. Each species have 50 different pictures. Only mountain peach flower and flowering peach flower are peach flower while apricot flower and sakura are not. Workers were asked to choose whether the flower in picture is prunus persica (peach flower).

We collected labels on the Amazon Mechanical Turk (MTurk) platform. 36 of all the different participants completed more than 10 *Human Intelligence Tasks* (HIT) on each. And they provided 2366 HIT in total. During the annotating procedure, two hints are shown to make sure that workers can distinguish prunus persica and sakura or distinguish prunus persica and apricot. Each picture was labeled by 11.8 workers and each worker provided 65.7 labels on average.

To visualize the structures learned by our models, we draw colormaps to show the partitions of different latent classes and different categories in Fig. 3(b)-3(d). Each subfigure contains a 50×4 color matrix, with each entry representing a

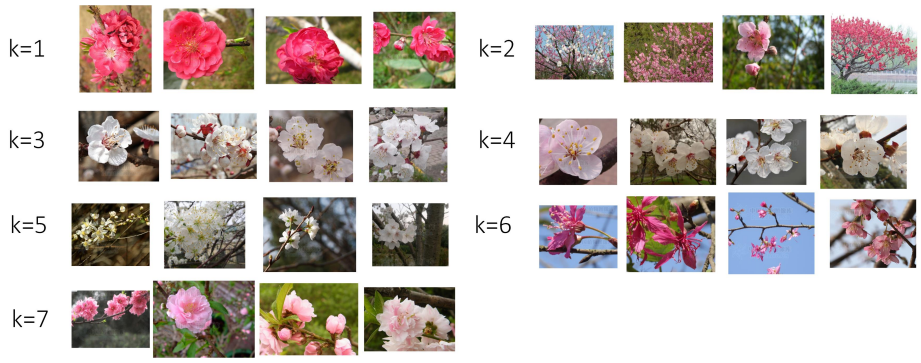


Fig. 4. Representative pictures for different latent classes.(best viewed in color).

flower image in the dataset, and each column corresponding to a unique flower species. Specifically, the first column is flowering peach flower, second is sakura, third is apricot flower and forth is mountain peach flower.

In Fig. 3(a) and Fig. 3(b), each color denotes one latent class learned by the estimator. We can see that the first three columns almost have pure color boxes, which means these three latent classes are strongly related to the flower species. The fourth column is kind of miscellaneous, which means that lots of mountain peach flowers are misclassified into other species. This is because mountain peach flowers have no distinct features comparing with other flower species.

In Fig. 3(c) and Fig. 3(d), each color denotes a classification category, either peach flower or not. This result comes from putting blue and azure boxes into peach flower category and other two colors' boxes into another. Fig. 4 shows some representative flower pictures for different latent classes we learned. These results suggest that the structures we learned have explicit semantic meaning, and these latent class patterns could be used in many further applications.

Finally, we evaluate the category recovery performance. The average worker error rate in this flower recognition task is 30.00%, and majority voting gets an error rate of 22.00%. The latent class minimax entropy estimator (LC-ME) wins on this dataset with error rate 11.00%, and the nonparametric latent class estimator (NDS,$\alpha_c = 1.6, \alpha_d = 2$) achieves 11.50%. The original Dawid-Skene estimator (DS) achieves 13.00%. The minimax entropy estimator (ME) [1] also achieves 13.00%. We also generated some sub-datasets with different numbers of workers in order to make more comparisons. Results are shown in Table 1, which consistently show the improvements by exploring our latent class assumptions.

6 Conclusions and Future Work

We have carefully examined the effectiveness of latent class structures in crowdsourcing. Our methods characterize that items in one dataset can be separated into several latent classes and workers' annotating behaviors may differ among different classes. By incorporating such fine-grained structures, we can describe the generation mechanism of noisy labels more clearly. Our methods can disclose

[1] Implementation from http://research.microsoft.com/en-us/projects/crowd.

Table 1. Performance of models on flowers dataset. Workers in use are randomly selected for each trial, and the average error rate of 10 trials, together with standard deviation, are presented. $\alpha_c = 0.09$ and $\alpha_d = 1$ are used for latent class recovery.

#	20	25	30	35
MV	0.1998 ± 0.0506	0.2383 ± 0.0216	0.2153 ± 0.0189	0.2170 ± 0.0096
DS	0.1590 ± 0.0538	0.1555 ± 0.0315	0.1310 ± 0.0213	0.1300 ± 0.0041
NDS	0.1595 ± 0.0737	0.1605 ± 0.0434	0.1330 ± 0.0371	0.1475 ± 0.0354
ME	0.1535 ± 0.0695	0.1470 ± 0.0339	0.1315 ± 0.0200	0.1335 ± 0.0078
LC-ME	$\mathbf{0.1415 \pm 0.0382}$	$\mathbf{0.1430 \pm 0.0286}$	$\mathbf{0.1215 \pm 0.0133}$	$\mathbf{0.1190 \pm 0.0168}$

meaningful latent classes, as demonstrated in real data experiments. After we get the latent class assignments, a category label recovery algorithm is developed, which is empirically demonstrated to achieve higher accuracies on category recovery tasks. Our latent structure models can preserve the structure information of data. For the future work, we plan to investigate the effectiveness of such hidden structure information further in handling other interesting tasks, such as online task selection and user behavior analysis.

Acknowledgement. The work was supported by the National Basic Research Program (973 Program) of China (No. 2013CB329403), National Natural Science Foundation of China (Nos. 61322308, 61332007), and the Tsinghua National Laboratory for Information Science and Technology Big Data Initiative.

References

1. Snow, R., O'Connor, B., Jurafsky, D., and Ng, A.Y.: Cheap and fast–but is it good?: evaluating non-expert annotations for natural language tasks. In EMNLP (2008)
2. Zhu,J., Chen,N., and Xing, E. P.: Bayesian inference with posterior regularization and applications to infinite latent svms. JMLR, 15 (2014), 1799-1847
3. Dawid, A.P., and Skene, A.M.: Maximum likelihood estimation of observer error-rates using the em algorithm. Applied statistics (1979), 20–28
4. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., and Moy, L.: Learning from crowds. JMLR, 11 (2010), 1297–1322
5. Zhou, D., Platt, J.C., Basu, S., anFd Mao, Y.: Learning from the wisdom of crowds by minimax entropy. In NIPS (2012)
6. D. Zhou, Q. Liu, J. C. Platt, and C. Meek.: Aggregating Ordinal Labels from Crowds by Minimax Conditional Entropy. In ICML (2014)
7. Welinder, P., Branson, S., Belongie, S., and Perona, P.: The multidimensional wisdom of crowds. In NIPS (2010)
8. Sheshadri, A., and Lease, M.: Square: A benchmark for research on computing crowd consensus. In First AAAI Conference on Human Computation and Crowdsourcing (2013)
9. Tian, Y., and Zhu, J.: Learning from crowds in the presence of schools of thought. In ICDM (2012)
10. Li, H., Yu, B., and Zhou, D.: Error Rate Analysis of Labeling by Crowdsourcing. ICML Workshop: Machine Learning Meets Crowdsourcing. Atalanta, Georgia, USA. 2013
11. Gao, C., and Zhou, D.: Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. arXiv preprint arXiv:1310.5764 (2013)
12. Neal, R. M.: Markov chain sampling methods for Dirichlet process mixture models. Journal of computational and graphical statistics, 9(2), (2000), 249-265