Machine Learning (2023) 112:1053–1075
https://doi.org/10.1007/s10994-022-06211-x

Beyond confusion matrix: learning from multiple annotators with awareness of instance features

Jingzheng Li[1,3] · Hailong Sun[2,3] · Jiyi Li[4]

Received: 18 October 2021 / Revised: 31 May 2022 / Accepted: 8 June 2022 /
Published online: 7 July 2022

Abstract
Learning from multiple annotators aims to induce a high-quality classifier from training instances, where each of them is associated with a set of observed labels provided by multiple annotators under the impact of their varying abilities and own biases. When modeling the probability transition process from latent true labels to observed labels, most existing methods adopt class-level confusion matrices of annotators which assume that observed labels do not depend on the instance features and are just determined by the true labels. However, in practice the labeling process of annotators is impacted not only by the correlation between classes but also by the content of instances. Thus using only class-level confusion matrices to characterize the probability transition process may limit the performance that the classifier can achieve. In this work, we propose the noise transition matrix, that incorporates the impact of instance features on annotators' performance based on confusion matrices. Furthermore, we propose a simple and effective learning framework, which consists of a classifier module and a noise transition matrix module in a unified neural network architecture. Experimental results on synthetic and real datasets demonstrate the noise transition matrix is better than the confusion matrix for modeling multiple annotators and the superiority of our method in comparison with state-of-the-art methods.

Editors: Bo Han, Tongliang Liu, Quanming Yao, Mingming Gong, Gang Niu, Ivor W. Tsang, Masashi Sugiyama.

✉ Hailong Sun
sunhl@buaa.edu.cn

Jingzheng Li
lijz19@act.buaa.edu.cn

Jiyi Li
jyli@yamanashi.ac.jp

[1] SKLSDE Lab, School of Computer Science and Engineering, Beihang University, XueYuan Road No.37, Beijing 100191, China

[2] SKLSDE Lab, School of Software, Beihang University, XueYuan Road No.37, Beijing 100191, China

[3] Beijing Advanced Innovation Center for Big Data and Brain Computing, XueYuan Road No.37, Beijing 100191, China

[4] Department of Computer Science and Engineering, University of Yamanashi, Takeda 4-3-11, Kofu 400-8511, Yamanashi, Japan

Keywords Answer aggregation · Learning from crowds · Crowdsourcing · Deep learning

1 Introduction

The success of supervised learning applications often relies on large-scale well-labeled datasets. Unfortunately, obtaining high-quality annotations from experts can be costly in terms of time and money budget. Alternatively, crowdsourcing (Han et al., 2019) provides an inexpensive approach to data labeling by hiring world-wide annotators on public platforms like Amazon Mechanical Turk (AMT). However, crowdsourced labels are usually noisy due to the existence of inexperienced or malicious annotators. Using these noisy labels in supervised learning may result in an inaccurate classifier. A straightforward way to solve this problem is redundant labeling, *i.e.,* obtaining multiple labels for each instance from multiple annotators. Hence this raises one fundamental problem termed as Learning from Crowds (LFC) (Rodrigues & Pereira, 2018): *"How can we learn a good classifier from a set of possibly noisy labeled data provided by multiple annotators?"*

To address the above issue, a two-stage approach is commonly adopted. First, in *answer aggregation* stage (Zheng et al., 2017; Sheng & Zhang, 2019; Jin et al., 2020a), the latent true labels are estimated. Then, a classifier is trained based on the estimated true labels. Alternatively, the one-stage approach (Raykar et al., 2009; Tanno et al., 2019) has been shown to be a promising direction that presents a maximum-likelihood estimator that jointly learns the classifier, abilities of multiple annotators (Yan et al., 2014), and the latent true labels. Among various research efforts on LFC, the probability transition process from latent true labels to observed crowdsourced labels is usually modeled with *confusion matrices* of annotators, which represents class-level probability transition. This means that the annotator's performance is consistent across different instances within the same class, *i.e.,* the transition from class j to class l is independent of instance features.

However, in the real world, the difficulty of labeling can vary among instances within the same class and the instance features themselves will affect annotators' performance (Misra et al., 2016). Consider an example from LabelMe dataset (Rodrigues et al., 2017) in Fig. 1(Top), which illustrates various cases of incorrect annotations given the true label "highway". The first indicates an inexperienced/malicious annotator who gives a random label "coast"; the second indicates an annotator have biased understanding on different classes, preferring to label "highway" as "street", because there is a strong correlation between those two classes. In both cases, the class-level confusion matrix of annotator can be used to characterize their varying abilities and own biases. Nevertheless, the third depicts one instance in class "highway" contains related visual features of other classes, misleading the annotators label it as "forest", although these two classes are weakly relevant. Therefore, the class-level confusion matrices cannot cover the diverse noisy cases and thus cannot completely characterize the performance of multiple annotators across different instances within the same class. This would limit the ability of a LFC model to estimate latent true labels, resulting in sub-optimal performance of the classifier. It is necessary to consider the impact of instance features in the process of characterizing performance of multiple annotators for LFC.

To address the aforementioned deficiency, this work aims at proposing a novel LFC framework, LFC-**x**, which can learn a classifier directly from the instances and the associated crowdsourced labels provided by multiple annotators. In particular, beyond confusion matrices, LFC-**x** models the probability transition process with *noise*

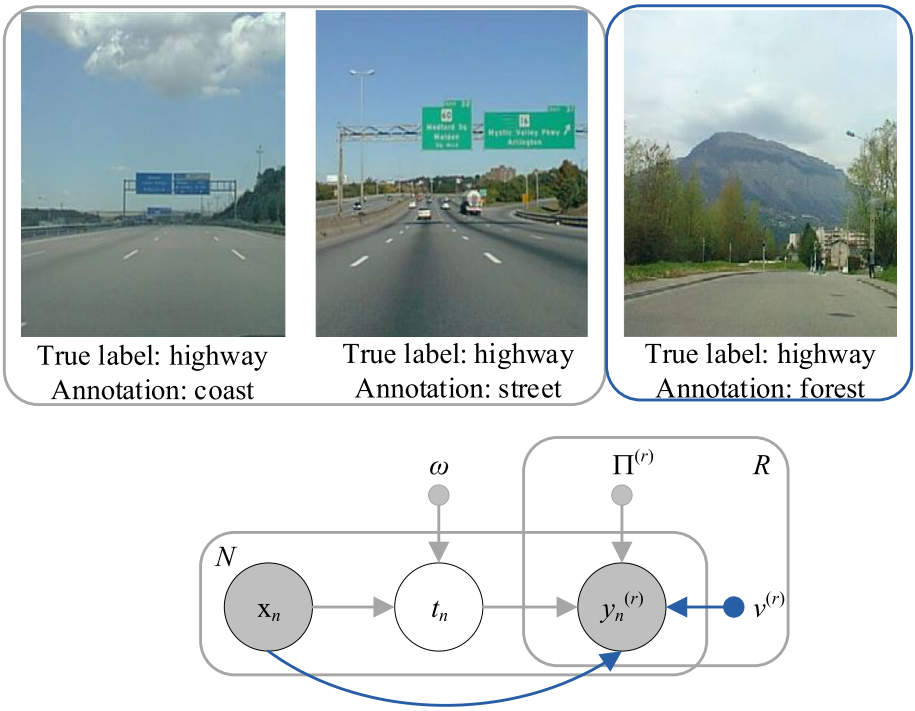

Fig. 1 Top: An example describes various incorrect annotations. The first randomly flips the true label to one of other classes; the second is that the true label is corrupted to the relevant class according to a fixed probability; the third is that the true label is corrupted to the irrelevant class due to the impact of instance features. Bottom: The graphical model of LFC-**x** represents the correlation of the instance $\mathbf{x}_n$, the true label t_n, and crowdsourced labels y_n. The annotation depends not only on the true label but also on instance features

transition matrices by incorporating instance features into confusion matrices. To this end, we need to deal with two practical challenges. One is how to quantify the impact of instance features on the performance of annotators in order to construct the noise transition matrix, the other is how to incorporate the noise transition matrix into LFC method. To cope with these challenges, we model the correlation among instance features, latent true labels and crowdsourced labels in the probabilistic graphical model to construct the noise transition matrix. Furthermore, the LFC-**x** consists of two modules: the noise transition matrix module and the classifier module. These two modules are integrated into an end-to-end neural network system through a principled combination for maximizing a likelihood function. The graphical model of the LFC-**x** is presented in Fig. 1(Bottom). In a nutshell, the main contributions and results of this work are summarized as follows:

- We propose a method to construct the noise transition matrix by incorporating the impact of instance features into the confusion matrix for modeling annotators' performance across instance features.
- We propose a novel LFC framework, which consists of a classifier module and a noise transition matrix module in an end-to-end neural network architecture. We

show that the proposed noise transition matrix is easy to implement and can be directly optimized by the standard SGD.
- We conduct extensive experiments on crowdsourced datasets which show that our method outperforms the compared methods in terms of test accuracy and robustness. In addition, we also verify that the noise transition matrix is superior to the confusion matrix for modeling noisy labels in a singly-labeled scenario.

2 Related work

There are mainly two lines of efforts on learning a classifier from crowdsourced labels provided by multiple annotators.

Answer aggregation: Two-stage approaches first infer true labels with *answer aggregation* (Zheng et al., 2017; Sheng & Zhang, 2019), then learn a classifier. One of the pioneer works is the DS model (Dawid & Skene, 1979), which applies the EM algorithm to estimate latent true labels and confusion matrices of annotators. On this basis, Whitehill et al. (2009) consider the generalized DS model, which involves the difficulty of each instance (Khetan & Oh, 2016; Han et al., 2016). Subsequently, Liu et al. (2012) aggregate the crowdsourced labels by applying approximate variational methods in graphical models. By analogy to ensemble learning, Kim and Ghahramani (2012) and Li et al. (2018) formalize the answer aggregation problem as Bayesian classifier combination that is capable of capturing correlations between different annotators. Except for the aforementioned probabilistic frameworks, weighted majority voting (Aydin et al., 2014) adopt weighted aggregation schemes for estimating true labels. Yin et al. (2017) integrate a classifier and a reconstructor into a unified model to estimate labels in an unsupervised manner. Similarly, even training deep neural networks directly to aggregate crowdsourced labels also achieve a good result (Gaunt et al., 2016). More details about answer aggregation can refer to Zheng et al. (2017); Sheng and Zhang (2019); Jin et al. (2020a). Nevertheless, two-stage approaches do not realize the full potential of combining answer aggregation and classifier (Khetan et al., 2018).

One-stage approaches: Raykar et al. (2009) come up with the one-stage approach, which implements an EM algorithm to jointly model abilities of annotators and learn a logistic regression classifier. This line of work is further extended to other types of models such as convolutional neural networks (Albarqouni et al., 2016) and supervised latent Dirichlet allocation (Rodrigues et al., 2017). Further, Khetan et al. (2018) allocate labeling budget to maximize the performance of a classifier via jointly modeling labels and confusion matrix from noisy crowdsourced labels. The above-mentioned methods reduce the LFC to a maximum likelihood estimation (MLE) problem and then use EM algorithm to solve it. Of particular interest, Kajino et al. (2013) notice that annotators form clusters according to their abilities, and apply clusters of annotators to resolve the LFC problem. Closer to our work, Rodrigues and Pereira (2018) propose the Crowd Layer to train deep neural networks end-to-end directly from the noisy crowdsourced labels, using only backpropagation. On this basis, Chen et al. (2020b) present a structured probabilistic model which incorporates the constraints of probability axioms into parameters of the Crowd Layer. More recently, Cao et al. (2019) and Li et al. (2020) simultaneously aggregate the crowdsourced labels and learn an accurate classifier via a multi-view learning. Chu et al. (2020) decompose the confusion matrix into two parts: one is commonly shared confusion matrix, and the other one is individual confusion matrix. Unlike our method, those

methods are based on a common assumption: the crowdsourced labels and the instance features are independent conditioning on the true labels.

To our knowledge, some methods also focus on the impact of instance features in LFC. Yan et al. (2010) employ logistic regression directly on the original instance content to characterize the confusion matrix in EM iterations, which not only ignores prior class-level probability transition information but also is not suitable for large-scale data. In contrast, our LFC-**x** incorporates instance features into the confusion matrix to construct the noise transition matrix in a unified neural network architecture. Zhang et al. (2019) infer true labels by propagating multiple noisy label distribution of each instance to its nearest neighbors, since it assumes that the multiple noisy label space share similar topological structure with the instance feature space. Unlike their work, we aim to characterize the impact of instance features on the performance of multiple annotators, rather than utilizing KNN graph to reconstruct the instances. Zhong et al. (2017) propose quality sensitive LFC method by using robust loss function, which estimates the reliability of crowdsourced labels by using the disagreement between crowdsourced labels and the model predictions on instance features, and then applies this term to loss function in SVM implementation. Unlike their work, we use noise transition matrices to correct incorrect annotations by considering the impact of instance features in the process of characterizing performance of multiple annotators.

3 Learning from crowds

In this section, we first present basic notations and the goal of interest. Then, we introduce a typical EM algorithm for LFC to learn a classifier from crowdsourced labels and reveal the deficiency of LFC method using the confusion matrix.

3.1 Notation and problem formulation

We assume that there are N i.i.d instances $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$, and each instance has an unknown true label. Let $y_n^{(r)}$ represents the annotation/label for $\mathbf{x}_n$ provided by annotator r in a set of R annotators. The labels from individual annotators may not be correct. Formally, we set the matrix $\mathbf{X} = [\mathbf{x}_1^T;...;\mathbf{x}_N^T] \in \mathbb{R}^{N\times D}$ and $\mathbf{Y} = [y_1^{(1)}, ..., y_1^{(R)};...;y_N^{(1)}, ..., y_N^{(R)}] \in \mathbb{R}^{N\times R}$ in which $(\cdot)^\top$ represents matrix transposition. We denote the unknown true labels for $\mathbf{X}$ by $\mathbf{T} = [t_1;...;t_N]$. Given the observed training data $\mathbf{X}$ and $\mathbf{Y}$, the goal of interest is to jointly estimate abilities of multiple annotators and latent true labels and train an accurate classifier.

In existing general methods of LFC, there are two common assumptions: 1) Given input instances, multiple annotators independently provide crowdsourced labels; 2) crowdsourced labels do not depend on the instance features, and just determined by the true labels. Conditioning on the true labels, the probability of crowdsourced labels on instance features can be factored as

$$p(\mathbf{Y} \mid \mathbf{X}, \Theta) = \prod_{n=1}^{N} \sum_{t_n} p(t_n \mid \mathbf{x}_n, \boldsymbol{w}) \prod_{r=1}^{R} p(y_n^{(r)} \mid t_n, \boldsymbol{\Pi}^{(r)}), \tag{1}$$

where $p(t_n \mid \mathbf{x}_n, \boldsymbol{w})$ represents true label distribution parameterized by $\boldsymbol{w}$, and $p(y_n^{(r)} \mid t_n, \boldsymbol{\Pi}^{(r)})$ parameterized by matrix $\boldsymbol{\Pi}$ depicts the class-level probability transition that the annotator, r, will annotate class $y_n^{(r)}$ given true label t_n. Specifically, the matrix $\boldsymbol{\Pi}^{(r)} = (\pi_{ij}^{(r)})_{C\times C} \in [0, 1]^{(C\times C)}$ is called the *confusion matrix* for representing the r-th

annotator's ability whose $(i,j)^{\text{th}}$ element is parameterized by $\pi_{ij}^{(r)}$ where $i,j \in \{1,\ldots,C\}$ and C is the number of classes. To achieve the goal, following (Raykar et al., 2009) and extending it from binary classification to multi-class classification task, the EM algorithm can be used to compute the maximum-likelihood solution, formalized as

$$\hat{\Theta}_{\text{ML}} = \arg\max_{\Theta} p(\mathbf{Y} \mid \mathbf{X}, \Theta). \tag{2}$$

E-step: Given the observation $\mathbf{X}$ and $\mathbf{Y}$ and the current estimate of the model parameters, the expected-value of complete-data log-likelihood (a lower bound on the true likelihood) can be computed as

$$\mathbb{E}\{\log p(\mathbf{Y}, \mathbf{T} \mid \mathbf{X}, \Theta)\} = \sum_{n=1}^{N} \sum_{t_n} q(t_n) \log(p(t_n \mid \mathbf{x}_n, \boldsymbol{w}) \prod_{r=1}^{R} p(y_n^{(r)} \mid t_n, \boldsymbol{\Pi}^{(r)})), \tag{3}$$

where the expectation is w.r.t. $p(t_n \mid y_n^{(1)}, \ldots, y_n^{(R)}, \mathbf{x}_n, \Theta)$ and we use $q(t_n)$ when referring to it for brevity. Given the estimate of the model parameters Θ_{old} in the current M-step, we can compute $q(t_n)$ by using Bayes' rule as follows.

$$q(t_n) \propto p(t_n \mid \mathbf{x}_n, \boldsymbol{w}_{\text{old}}) \prod_{r=1}^{R} p(y_n^{(r)} \mid t_n, \boldsymbol{\Pi}_{\text{old}}^{(r)}). \tag{4}$$

M-step: Based on the observation $\mathbf{X}$ and $\mathbf{Y}$ and the estimation of posterior probabilities of ground truth in current E-step, the confusion matrix can be updated by maximizing the expected-value of complete data log-likelihood. By equating the derivative of Eq. (1) to zero, we obtain the following estimate for updating the confusion matrix.

$$\pi_{jl}^{(r)} = \frac{\sum_{n=1}^{N} q(t_n = j) \cdot \mathbb{I}[y_n^{(r)} = l]}{\sum_{n=1}^{N} q(t_n = j)}, \tag{5}$$

where $\mathbb{I}[\cdot]$ is an indicator function. The update of parameter-set $\boldsymbol{w}$ in classifier depends on which type of classifier is used. If the classifier used is neural network, then we can use the posterior probability of true labels to back-propagate the error by using gradient descent optimization algorithm.

3.2 Limitations

The noisy crowdsourced labels, in the most popular noise model hitherto, are corrupted from ground truth by an unknown *noise transition matrix* (Han et al., 2018a; Yao et al., 2020) to depict such probability transition process. We can notice that the conventional LFC method makes a simplistic assumption that crowdsourced labels only depend on the ground truth but not the input instance features. That is, the noise transition matrix is characterized only by confusion matrices of annotators. However, in the real world, the content of instances (*i.e.,* instance features) including foreground and background varies among instances within the same class so that the instance features themselves will affect the annotator's judgment on labels. Modeling the probability transition process considering only confusion matrices of multiple annotators would limit the ability to infer the latent true labels and lead to sub-optimal performance of the classifier. In summary, the noise transition matrix cannot be completely constructed by class-level confusion matrices of

annotators, and it is also necessary to consider each annotator's performance depending on instance features.

For the EM-based LFC described above, a potential issue of combining a classifier and EM algorithm direction is scalability (Goldberger & Ben-Reuven, 2017). The model requires training a classifier in each iteration of EM algorithm, so many EM iterations are likely to be needed for convergence. In addition, the other primary criticism of EM-based LFC approaches is that in practice, each instance would not be labeled too many times considering the labeling cost. With relatively little redundancy, the standard applications of EM are of limited use (Khetan et al., 2018). On the other hand, it is intractable to construct the dedicated noise transition matrix as a part of the EM algorithm because it cannot depict confusion matrix and the awareness of instance features separately, let alone incorporates the instance features into LFC framework. Inspired by the Crowd Layer (Rodrigues & Pereira, 2018) that trains deep neural networks end-to-end directly from the crowdsourced labels using only back-propagation, we present the principled solution which can incorporate instance features in modeling the probability transition process for designing LFC method.

4 Methodology

4.1 Proposed LFC-x

In our setting, we here make a key assumption that the crowdsourced labels depend on not only true labels but also instance features. We propose the *noise transition matrix* to model probability transition process on multiple annotators given input instances through incorporating input instance features into confusion matrix. On this basis, now we rewrite the probability of given crowdsourced labels, then we take its log and the part of the probability transition in Eq. (1) related to confusion matrix is replaced with the proposed noise transition matrix, yielding

$$\begin{aligned}\log p(\mathbf{Y} \mid \mathbf{X}, \Theta) &= \sum_{n=1}^{N}\sum_{r=1}^{R} \log p(\mathbf{Y}^{(r)} \mid \mathbf{X}, \Theta^{(r)}) \\ &= \sum_{n=1}^{N}\sum_{r=1}^{R} \log \sum_{t_n} \underbrace{p(t_n \mid \mathbf{x}_n, \boldsymbol{w})}_{\text{classifier}} \underbrace{p(y_n^{(r)} \mid t_n, \mathbf{x}_n, \boldsymbol{\Pi}^{(r)}, \boldsymbol{v}^{(r)})}_{\text{noise transition matrix}},\end{aligned} \tag{6}$$

where the Θ is a collection of $\{\boldsymbol{w}, \{\boldsymbol{\Pi}^{(r)}\}_{r=1}^{R}, \{\boldsymbol{v}^{(r)}\}_{r=1}^{R}\}$, and the parameter set $\{\boldsymbol{v}^{(r)}\}_{r=1}^{R}$ represents the impact of instance features themselves on annotators' performance that we will discuss later.

To instantiate our probabilistic graphical model as shown in Fig. 1(Bottom), we propose LFC-**x** that minimizes the negative log-likelihood function with respect to the parameter Θ within the framework of neural network that consists of two modules: a *classifier module* and a *noise transition matrix module*. Specifically, the log-likelihood is the output of LFC-**x** through a principled combination of the classifier module and the noise transition matrix module. Figure 2 presents the overall design of LFC-**x**. Next, we describe how to jointly optimize parameters of the classifier module and the noise transition matrix module.

Classifier Module: Without loss of generality, suppose that a softmax neural classifier is given, parameterized by $\boldsymbol{w}$, for inferring the true label distribution. We denote

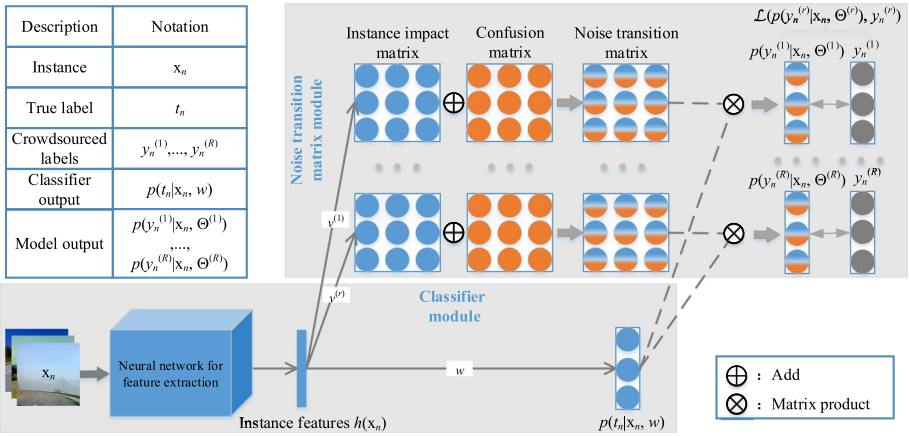

Fig. 2 General schematic of the LFC-**x** for classification with 3 classes and R annotators. It integrates a classifier module and a noise transition matrix module into a unified neural network architecture

the non-linear function applied to an instance $\mathbf{x}_n$ by $h(\mathbf{x}_n)$ for extracting the instance features. Given the instance features, the softmax layer is adopted to predict the true label t_n by

$$p(t_n = j \mid \mathbf{x}_n, \boldsymbol{w}) = \frac{\exp(\mathbf{u}_j^\top h(\mathbf{x}_n) + b_j)}{\sum_{i=1}^{C} \exp(\mathbf{u}_i^\top h(\mathbf{x}_n) + b_i)}, \tag{7}$$

in which the vector $\mathbf{u}$ and scalar b indicate weight and bias, respectively, and $i, j \in \{1, \ldots, C\}$. However, we cannot have access to the true labels and only have access to the observed crowdsourced labels. The classifier module alone is not enough. Therefore, a noise transition matrix module that can model the labeling process of multiple annotators, *i.e.,* the probability transition from true labels and instance features to crowdsourced labels, is required to provide weakly supervised information for the classifier. In doing so, the classifier can be trained by multiplying its output with the estimated noise transition matrices which is shown in Eq. (6).

Noise Transition Matrix Module: Different from annotator-specific confusion matrix that characterizes class-level probability transition, we propose annotator-specific noise transition matrix that characterizes instance-level probability transition through incorporating instance features into confusion matrix. Now, a major challenge is how to construct the noise transition matrix. Naturally, it contains two problems. The first one is how to quantify the impact of instance features on annotators' performance. The second one is how to incorporate such impact of instance features into confusion matrix for constructing noise transition matrix.

The first problem comes from how to specify the mapping function from instance features to the annotators' performance. To this end, we construct *instance impact matrix* parameterized by $\boldsymbol{v}^{(r)}$ to characterize the instance features' impact on annotators' performance. More concretely, we explore a solution by adding a linear layer with C^2 units on top of instance features in the classifier module for each annotator. Recall that the instance features are extracted by the non-linear function $h(\mathbf{x}_n)$, which is the output of penultimate layer of classifier. This linear layer, called *instance impact matrix layer*, is parallel to the

softmax output layer of the classifier module (shared by the classifier module and the noisy transition matrix module) with only weights and no bias parameters, formalized as

$$f(\mathbf{x}_n)^{(r)} = \boldsymbol{\nu}^{(r)} h(\mathbf{x}_n), \tag{8}$$

where the vector $\boldsymbol{\nu}^{(r)}$ is the parameter of instance impact matrix layer.

Now we obtain the instance impact matrix, the other crucial point is how to construct the noise transition matrix. To solve this issue, we make an intuitive assumption that the noise transition matrix is the result of the sum of the instance impact matrix $f(\mathbf{x}_n)_{ij}^{(r)}$ and the confusion matrix $\boldsymbol{\Pi}^{(r)}$. Also, we discuss other variants of constructing noise transition matrices in Sect. 5.5 For the sake of computational feasibility, we convert the output of instance impact matrix layer from vector form to matrix form with the same shape as the confusion matrix and then add them up, followed by a softmax operation, yielding

$$p(y_n^{(r)} = j \mid t_n = i, \mathbf{x}_n, \boldsymbol{\Pi}^{(r)}, \boldsymbol{\nu}^{(r)}) = \frac{\exp(f(\mathbf{x}_n)_{ij}^{(r)} + \pi_{ij}^{(r)})}{\sum_{k=1}^{C} \exp(f(\mathbf{x}_n)_{ik}^{(r)} + \pi_{ik}^{(r)})}. \tag{9}$$

Note that in our method we use the π to denote the parameters of confusion matrix, which is actually logits without softmax operation in the neural network.

We propose the LFC-**x** that integrates a classifier module and a noise transition matrix module into a unified neural network in an end-to-end manner for jointly estimating true labels, abilities of annotators, and learning a classifier.

We take a closer look at the optimization objective of LFC-**x** as Eq. (6). The penultimate layer (extracted instance features) of the classifier module is shared among every annotator and becomes an "information hub" that connects the noise transition matrix of each annotator. Given a loss function $\mathcal{L}(p(y_n^{(r)} \mid \mathbf{x}_n, \Theta^{(r)}), y_n^{(r)})$ such as commonly-used Cross Entropy loss between the model outputs and the crowdsourced labels, minimizing the negative log-likelihood encourages the outputs of LFC-**x** to be as close as possible to the observed crowdsourced labels. In doing so, we can perform back-propagation end-to-end for updating parameters in classifier module and noise transition matrix module. Besides, the problem of missing labels from some of annotators can be addressed by setting their gradients to 0.

4.2 Training procedure

There are degrees of freedom in the outputs of a classifier. In other words, the outputs of classifier may not semantically correspond to the true labels even if the negative log-likelihood function is minimized (Sukhbaatar et al., 2014). Therefore, a reasonable initialization of noise transition matrix is crucial for successful convergence of the LFC-**x** for training a high-quality classifier. As for the confusion matrix, the diagonal element corresponds to the probability of correctly labeling a certain class. In this paper, we assume that there is no malicious annotator. We initialize the confusion matrix so that it has relatively large diagonal elements (*i.e.,* $\pi_{ii} > \pi_{ij}$ for $\forall i, j \neq i$), and small symmetric noise in off-diagonal elements, *i.e.,*

$$\pi_{ij}^{(r)} = \log(\epsilon^{\mathbb{I}[i=j]} \times (\frac{1-\epsilon}{C-1})^{(1-\mathbb{I}[i=j])}), \tag{10}$$

in which ϵ is set to 0.46 for all datasets and we set the value of ϵ via a grid search within the range [0.4, 0.7] in line with ability of real annotators. The parameters of instance impact matrix layer are initially set to 0.

Let us finally describe the training procedure which consists of two stages, illustrated in Fig. 3. The first stage is to update confusion matrices. In detail, we freeze the instance impact matrix layer so that the noise transition matrix degenerates into the confusion matrix since the instance impact matrix is fixed to 0. Then, we train the LFC-**x** for updating confusion matrices. The second stage is to update noise transition matrices. Concretely, we unfreeze the instance impact matrix layer and retrain the LFC-**x** except for the learned confusion matrices. Once the LFC-**x** is trained, the classifier module can be used separately to make predictions for unseen instances.

4.3 Minimax error analysis

Motivated by previous theoretical works (Imamura et al., 2018; Gao et al., 2016), here we analyze the minimax error of our LFC-**x**. The error rate can be measured by $\mathcal{L}(\hat{\mathbf{T}}, \mathbf{T}) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}[\hat{t}_n \neq t_n]$, in which $\hat{\mathbf{T}}$ is the collection of estimated true labels of all instances. We use $\zeta(n)^r$ to represent instance impact matrix $f(x_n)^r$ of an annotator acting on an instance in short. Denote by $\rho^n = \{\rho_i^n\}_{i=1}^C$ the instance-specific class probabilistic distribution of model output. Given the instances, the crowdsourced labels, and Θ representing a collection of $\{\{\mathbf{\Pi}^{(r)}\}_{r=1}^R, \{\boldsymbol{\nu}^{(r)}\}_{r=1}^R\}$, we bound the minimax error rate with respect to LFC-**x** as follows.

Theorem 1 *The minimax error rate of our method is lower bounded by*

$$\inf_{\hat{\mathbf{T}}} \sup_{\mathbf{T}\in[C]^N} \mathbb{E}\left[\mathcal{L}(\hat{\mathbf{T}}, \mathbf{T})\right] \geq \frac{1}{N^2 \log C} \sum_{n=1}^{N} F(\rho^n, \Theta) - \frac{\log 2}{N^2 \log C} \tag{11}$$

where

$$F(\rho^n, \Theta) = H(\rho^n) - \sum_{r=1}^{R}\sum_{i=1}^{C}\sum_{j=1}^{C} \rho_i^n \rho_j^n KL\Big((\zeta(n)_{i*}^r + \pi_{i*}^r) || (\zeta(n)_{j*}^r + \pi_{j*}^r)\Big) \tag{12}$$

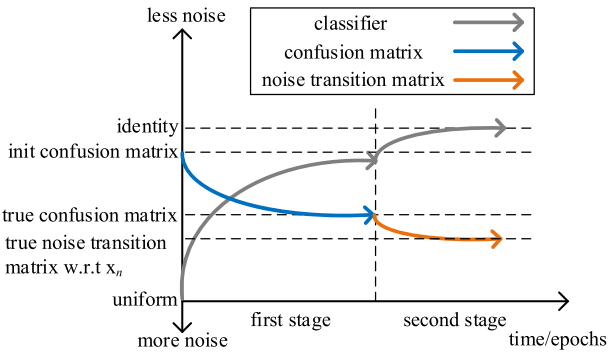

Fig. 3 Training procedure (see text for details)

where $H(\rho^n) = -\sum_{c=1}^{C} \rho_c^n \log \rho_c^n$ *indicates the entropy of class probabilistic distribution,* $\zeta(n)_{i*}^r$ *and* π_{i*}^r *denote the i-th row in the matrices respectively.*

The proof of minimax error rate can be found in previous theoretical analysis (Imamura et al., 2018). The noise transition matrix is decomposed into the sum of the class-level confusion matrix and the instance impact matrix in our method.

Remark 1 Since $\inf_{\hat{\mathbf{T}}} \sup_{\mathbf{T} \in [C]^N} \mathbb{E}[\mathcal{L}(\hat{\mathbf{T}}, \mathbf{T})]$ contains the infimum over estimate $\hat{\mathbf{T}}$, we provide the lower bound of the minimax error rate to analyse the behavior of the model itself which does not depend on the classifier module that estimates true labels. Theorem 1 sheds light on how the LFC-**x** reduces the error rate through the interaction of the confusion matrix and instance impact matrix included in the noise transition matrix. Given an example to illustrate the advantage of our method, one annotator prefers to label class "highway" as "street" due to the existence of class confusion (Jin et al., 2020b) and his/her own bias. There are some instances of class "highway" misleading the annotator to label them as "forest". If the instance impact matrix is not considered in Theorem 1 as utilized by Imamura et al. (2018), the KL divergence between classes "street" and "forest" in confusion matrix becomes smaller across each instance because their entries on "highway" are close. Theorem 1 suggests that incorrect labels provided by annotators influenced by instance features can be learned by the instance impact matrix, ensuring that KL divergence between classes "street" and "forest" in the confusion matrix across other instances belonging to the same class would not became smaller. We can observe that considering the instance impact matrix has the potential to reduce the error rate over all instances. Furthermore, the two-stage training produce introduced in Sect. 4.2 encourages the confusion matrix and instance impact matrix to be learned separately without confusing each other.

4.4 Relations among the noise transition matrix and the quality factors of crowdsourced labels and singly-labeled data

Existing methods usually model the factors that influence the quality of crowdsourced labels, including ability of annotators and difficulty of instances. For example, the crowd layer (Rodrigues & Pereira, 2018) considers only class-level confusion matrix for characterizing the ability of annotator. Whitehill et al. (2009) propose the generalized DS model involving the difficulty of instance, in which the difficulty of instance is implicitly modeled only through access to crowdsourced labels. In this paper, we propose to construct the noise transition matrix by incorporating the impact of instance features into confusion matrix for modeling annotators' performance across instance features, which can be regarded as a fusion of instance difficulty and annotator ability. Specifically, the difficulty of instance is modeled by explicitly utilizing instances features.

It is an important issue to consider the impact of instance features in learning with noisy singly-labeled data that has received increasing attention recently (Yao et al., 2021; Chen et al., 2020a; Zhang et al., 2021; Zhu et al., 2021; Liu, 2021). However, the impact of instance features on the labeling process of multiple annotators in learning from crowds has not been effectively addressed. The crowdsourced labels differ from the singly-labeled scenario in that (1) each instance may correspond to multiple annotations provided by multiple annotators and the information of annotators is required; (2) learning from crowds needs to consider the aggregation process of multiple annotators when modeling noisy annotations. There are some works (Jiang et al., 2021; Berthon et al., 2021) in singly-labeled scenario

that rely on confusion matrix to model the label noise statistically. Although the confusion matrix-based methods possess theoretical guarantee, it is difficult to estimate the confusion matrix for each instance under the instance dependent noise. To ease the estimation, some unrealistic assumptions have to be posed on the confusion matrix, including class-level confusion matrix (Liu & Guo, 2020; Li et al., 2021), symmetric confusion matrix (Menon et al., 2018), upper bounded noise rate (Cheng et al., 2020), and part-dependent label noise (Xia et al., 2020). In addition, some works (Jiang et al., 2021; Berthon et al., 2021) consider the impact of the instance on the confusion matrix by re-weighting/correcting the loss term of the instance according to the confidence score of the noisy labels. Unlike their works, our method assumes that the instance-level noise transition matrix can be obtained by a learnable instance impact matrix acting on the class-level confusion matrix. To the best of our knowledge, two works (Goldberger & Ben-Reuven, 2017; Yan et al., 2010) most relevant to our method consider that the class-level confusion matrix is affected by the instance features. They directly use a nonlinear mapping from instance features to the instance-level confusion matrix which causes the class-level and instance-level probability transitions to become indistinguishable. Different from their works, our method adopts a two-stage training procedure which considers both class-level and instance-level transition probability information without confusing each other, thus yielding more stable and superior result.

5 Experiments

We begin by investigating and discussing the behavior of the neural-based LFC methods during training in the presence of noisy crowdsourced labels. Afterward, we evaluate our proposed LFC-**x** by comparing it with representative LFC baselines on both synthetic and real datasets. Moreover, we test our method combined with a robust loss function against noisy crowdsourced labels. Finally, our method has good flexibility to be applied to noisy singly-labeled scenario. All methods are implemented using the Keras framework.

5.1 Datasets

Synthetic datasets: Most existing public crowdsourcing datasets do not contain instance features information and are not suitable for LFC scenario. Following previous works (Yan et al., 2010; Rodrigues & Pereira, 2018), we simulate several annotators to provide crowdsourced labels based on CIFAR-10 and four UCI datasets, where UCI datasets are from UC Irvine machine learning repository (Dua & Graff, 2017). Table 1 provides detailed information of datasets used, including binary and multiclass classification tasks, which represents a wide range of domains and data characteristics.

To create reliable and rational synthetic datasets, we generate two types of synthetic crowdsourced labels, *i.e., uniform labels* and *clustering-based labels*, which indicate the overall ability of annotator and the personal bias on some similar instances features, respectively. To generate uniform labels, we randomly flip a correct label to one of the other incorrect labels uniformly and refer to the portion of incorrect labels as the ability of annotator. We simulate five annotators with different abilities varying from 0.3 to 0.7, *i.e.,* $p \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$. To generate clustering-based labels, we employ data clustering method for each dataset, which follows the previous works (Yan et al., 2010; Zhong et al., 2017) that consider instance features in LFC proceeded as follows. Firstly, we simulate five

Table 1 Characteristics of Datasets

Datasets	#Instances	#Features	#Classes	Real labels
CIFAR-10	50000	32×32	10	NO
Parkinson speech	1208	26	2	NO
QSAR(biodegradation)	1055	41	2	NO
Statlog(image)	2310	18	7	NO
Waveform(version 1)	5000	21	3	NO
LabelMe	1000	256×256	8	YES
Sentiment Polarity	10428	1200	2	YES

annotators and perform k-means clustering on training data to group them into five clusters. Then, for each annotator, we assume that the r-th annotator in the five annotators is good at labeling instances belonging to r-th cluster, where their annotations coincide with their true labels; meanwhile the r-th annotator correctly labels the rest instances belonging to other clusters with a probability $p^r \sim U(0.2, 0.3)$ (U indicates uniform distribution). In doing so, we obtain the crowdsourced labels. In practice, since each annotator only labels a small subset of all instances, we introduce a probability $\eta \sim U(0.5, 0.7)$ for each annotator to decide whether to label an instance in label generation process.

Real datasets: LabelMe dataset (Rodrigues et al., 2017) is an image classification dataset involving eight classes including "highway", "inside city", "tall building", "street", "forest", "coast", "mountain" and "open country", and it contains 2,688 images in total. Among them, 1,000 images are annotated by AMT annotators and each is annotated by 2.547 annotators on average. The remaining 1,688 images are used for testing.

Sentiment Polarity dataset (Pang & Lee, 2005) is a textual sentiment analysis dataset containing 10,428 sentences about movie review snippets from Rotten Tomatoes. Rodrigues et al. (2013) provided crowdsourced labels for this dataset on AMT platform. 4,999 sentences of this dataset are annotated by AMT workers with the sentiment polarity "positive" or "negative", and each is labeled by an average of 136.68 annotators. The remaining 5,429 sentences are used for testing. Further, Rodrigues et al. (2013) also provided feature vectors version of the text dataset by applying latent semantic analysis to bag-of-words feature vectors.

5.2 Understanding the training process of neural-based LFC

Since few works analyze the learning process of the neural-based LFC in the presence of noisy crowdsourced labels, here we present and analyze the behavior of proposed LFC-**x** and existing neural-based LFC methods including one-stage and two-stage respectively on synthetic CIFAR-10 dataset.

Competing strategies: Two-stage approaches with neural networks include: (1) NN-MV: the neural network classifier baseline of training with the labels inferred with majority voting; (2) NN-DS: the neural network classifier baseline of training with the labels inferred with DS model (Dawid & Skene, 1979). One-stage LFC approach Crowd Layer (Rodrigues & Pereira, 2018): train deep neural networks end-to-end directly from the crowdsourced labels using only back-propagation. Note that the Crowd Layer can be seen as a degenerated case of LFC-**x** if we freeze the instance impact matrix layer and fix its

parameters to 0 in LFC-**x**. The comparison of our LFC-**x** with Crowd Layer can be seen as an ablation study to test the efficacy of instance impact matrix.

Experimental setup: The experiments are conducted on the CIFAR-10 with two types of noisy crowdsourced labels. We use a 5-layer CNN architecture combined with ReLU activation and max-pooling, followed by two fully connected layers to build a classifier, which is standard test bed (Laine & Aila, 2016; Han et al., 2018b) for CIFAR-10. Unless otherwise specified, for a fair comparison, all comparative methods use the same network architecture with Cross Entropy loss. The model is trained using SGD with momentum of 0.9, weight decay of 10^{-4}, and an initial learning rate of 0.01. The learning rate is divided by 10 after epochs 40 and 80. The batch size is set to 1024.

In Fig. 4, we report test accuracy vs. number of epochs on uniform labels and clustering-based labels respectively, from which the observations are as follows.

One-stage approaches Crowd Layer and LFC-**x** exhibit relatively robust performance when confronted with both uniform labels and clustering labels as the number of epochs increases. It indicates that the potential advantage to jointly learn the classifier and estimate abilities of annotators in combating label noise. More importantly, LFC-**x** performs the best and consistently provides a more modest improvement than Crowd Layer, demonstrating the advantage of noise transition matrix over confusion matrix in designing the LFC method.

Unlike the one-stage approaches learning directly from crowdsourced labels, the two-stage approaches uses fixed labels inferred by the answer aggregation and then learns a neural network classifier. We can observe that two-stage approaches NN-MV and NN-DS show different behaviors under different types of crowdsourced labels. On uniform labels, neural networks in two-stage approaches can automatically learn generalizable "pattern" in the early training epochs before overfitting occurs (*i.e.,* performance decline). Nonetheless, two-stage approaches are relatively robust to clustering-based labels, probably because clustering-based labels that are corrupted non-uniformly are more complicated than uniform labels, which makes it difficult for neural networks to automatically capture meaningful "patterns".

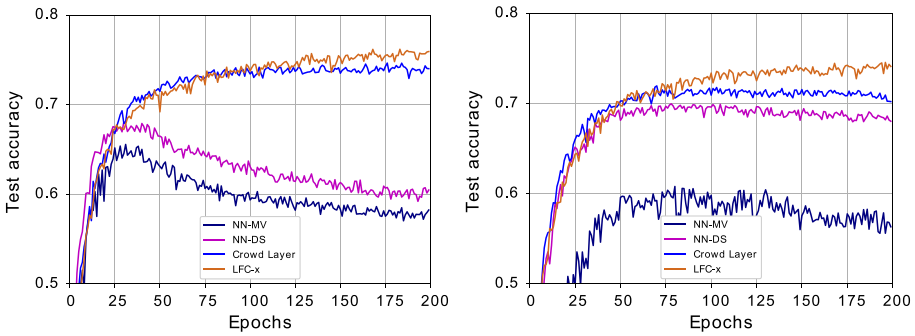

Fig. 4 The test accuracy vs. number of epochs on synthetic CIFAR-10. Left: uniform labels. Right: clustering-based labels

5.3 Evaluation on synthetic and real datasets

Next, to further verify the effectiveness of proposed LFC-**x**, we conduct comprehensive experiments on synthetic UCI datasets with two types of noisy crowdsourced labels and two real crowdsourcing datasets. Moreover, the comparative LFC baselines are not limited to neural network-based LFC approaches, and also include EM-based LFC approaches.

Competing strategies: Apart from neural-based LFC baselines, we also compare the LFC-**x** to the EM-based LFC methods including: (1) Raykar Raykar et al. (2009): a maximum-likelihood estimator that jointly learns the classifier, the confuse matrices, and the underlying true labels based on EM algorithm; (2) AggNet (Albarqouni et al., 2016): a generalized version of Raykar method, in which the classifier is a neural network. In addition, we also compare a method named Max-MIG (Cao et al., 2019): it jointly estimates a neural classifier and a label aggregation network using an information-theoretical loss function.

Experimental setup: For each UCI dataset, we retain 90% of the data as training set, the remaining 10% of the data as test set. Since the dimension of datasets is relatively small, we employ a two-layer neural network with 32 units for each layer followed by a softmax output layer to build a classifier. For LabelMe dataset, we use a pre-trained VGG-16 as the backbone network and replace the last fully connected layer with the task-specific fully connected layer. For Sentiment Polarity dataset, we use a two-layer neural network comprised 500 and 128 units and one softmax output layer on top to build a classifier. We choose ReLU as the activation function and use dropout with parameter 0.5 for all datasets. The batch size is set to 256 and we run 400 epochs. Jiang et al., (2020) point out that early stopping is not always effective on label noise, although some previous studies report the best results. Therefore, we report not only the optimal test score during training, but also the test score of last epoch, to see the robustness of methods. Each experiment is repeated ten times and we report the mean test accuracy and standard deviation.

Table 2 summarizes the comparisons of LFC-**x** to other LFC approaches. We have the following observations. Firstly, for the uniform labels as shown in Table 2a, the LFC-**x** exhibits competitive results and it surpasses competing methods in most cases. Secondly, Table 2b illustrates our LFC-**x** consistently outperforms its competitors by a clear margin across all datasets under the clustering-based labels. Finally, Table 2c presents the test accuracy on the real crowdsourcing datasets. The LFC-**x** is superior to all the compared LFC methods, which empirically demonstrates that we provide a realistic and applicable noise transition matrix for designing an LFC framework. Moreover, the LFC-**x** is very robust, and there is no big drop between the best score and the last score over almost all datasets.

5.4 Evaluation on combination with robust loss functions

We model the labeling process of multiple annotators for learning from noisy crowdsourced labels, which is also a case of weakly supervised learning (Karimi et al., 2020; Song et al., 2020). Our method is orthogonal to many label-denoising techniques such as instances re-weighting (Jiang et al., 2018) and robust loss functions (Zhang & Sabuncu, 2018; Wang et al., 2019; Patrini et al., 2017), which can be used to enhance our method. We evaluate a combination of LFC-**x** and some robust loss functions include Symmetric Learning (SL) (Wang et al., 2019) and Generalized Cross Entropy (Zhang & Sabuncu,

Table 2 Comparison of the LFC-**x** to other LFC baselines on test accuracy (%)

Datasets		NN-MV	NN-DS	Raykar	Aggnet	Crowd Layer	Max-MIG	LFC-**x**
(a) Uniform labels								
Parkinson speech	best	69.55±0.93	72.37±0.62	71.90±0.00	72.56±1.16	70.66±2.87	71.90±1.53	**72.73±1.79**
	last	68.24±0.48	69.92±1.62	70.25±0.00	71.38±1.06	67.77±2.73	**71.90±1.51**	68.60±2.10
QSAR	best	81.19±1.40	84.71±1.49	82.08±0.00	82.83±1.52	84.43±2.37	82.64±1.68	**85.85±1.07**
	last	79.64±3.01	81.89±1.34	82.08±0.00	81.89±1.52	83.96±1.63	82.08±1.35	**85.38±0.98**
Statlog	best	92.05±0.52	93.14±0.36	93.94±0.00	94.02±1.02	94.81±2.02	93.94±1.63	**95.45±1.31**
	last	92.01±0.57	92.79±0.38	93.94±0.00	93.85±0.96	94.59±1.83	93.77±1.64	**95.02±1.29**
Waveform	best	86.08±0.92	**89.20±0.13**	89.04±0.08	88.92±0.54	89.10±1.86	88.51±0.93	89.10±0.23
	last	83.04±1.57	86.52±1.06	**89.00±0.00**	87.92±0.53	87.90±2.05	88.51±0.84	88.50±0.52
(b) Clustering-based labels								
Parkinson speech	best	65.95±1.32	68.91±3.35	66.94±0.00	63.64±0.96	69.01±5.24	68.39±0.83	**71.90±1.20**
	last	62.64±1.08	67.98±3.74	66.94±0.00	63.31±1.11	62.40±3.69	**68.39±0.83**	68.18±1.92
QSAR	best	70.94±2.34	77.74±1.13	68.87±0.00	68.11±0.92	81.60±2.61	83.25±0.96	**85.85±0.72**
	last	66.98±2.53	69.24±1.28	66.04±0.00	68.11±0.92	79.25±1.93	83.01±0.87	**85.38±0.81**
Statlog	best	52.81±1.40	90.21±1.93	90.48±0.00	76.28±7.31	93.07±2.11	90.87±0.72	**94.81±0.84**
	last	48.57±2.15	87.19±3.58	90.39±0.17	71.69±8.77	92.20±1.53	90.87±0.72	**94.37±0.86**
Waveform	best	70.04±3.06	85.12±0.89	84.92±0.10	83.12±5.83	87.80±1.92	87.80±1.54	**88.30±1.25**
	last	61.88±3.43	78.68±0.35	84.32±0.16	79.52±9.17	86.49±1.65	86.63±1.37	**87.00±1.56**
(c) Real crowdsourced labels								
LabelMe	best	80.67±0.90	82.69±0.78	75.58±0.01	83.76±0.64	84.18±0.36	85.23±0.21	**85.86±0.13**
	last	76.48±1.30	77.98±0.66	75.08±0.01	83.20±0.53	81.44±1.43	84.26±0.18	**84.60±0.15**
Sentiment Polarity	best	72.15±0.12	72.62±0.13	72.00±0.00	71.43±0.05	72.90±0.20	72.13±0.04	**73.40±0.06**
	last	70.66±0.18	71.02±0.47	72.00±0.00	70.95±0.25	72.44±0.27	72.13±0.04	**72.70±0.09**

The test accuracy of the best during training and the last epoch are listed on four synthetic UCI datasets with two types of crowdsourced labels and two real crowdsourcing datasets. The best results are in boldface

2018), called "LFC-**x** (SL)" and "LFC-**x** (GCE)" respectively. It is readily implemented by replacing the commonly-used Cross Entropy loss with the robust loss function. We also report experimental results for the combination of the Crowd Layer with robust loss functions. Table 3 reports the test accuracy of comparative methods on synthetic datasets with uniform labels and two real crowdsourcing datasets. We can observe that robust loss functions maintain the effectiveness of our method, and further boost the performance of LFC-**x** and Crowd Layer in many cases. For example, the best test accuracy of LFC-**x** (GCE) can reach 86.75% on LabelMe dataset. Furthermore, the LFC-**x** combined with robust loss functions performs better than the Crowd Layer combined with robust loss functions, which demonstrates the advantage of the noise transition matrix compared to the confusion matrix.

5.5 Ablation study and analysis

As previously mentioned, the comparison of our LFC-**x** with Crowd Layer can be seen as an ablation study to test the efficacy of instance impact matrix in learning from crowds. The experimental comparisons in the Fig. 4 and Table 2 demonstrate the advantage of noise transition matrix over confusion matrix in designing the LFC method. Moreover, we add an experimental comparison between our algorithm and Crowd Layer for confusion matrix visualization on the LabelMe dataset, which reflects the difference between the noise pattern evaluated for annotators and the real noise pattern. Since the instance-level noise transition matrix cannot be visualized on the overall training data, we compare only the confusion matrices learned by LFC-**x** and Crowd Layer with the true confusion matrix on LabelMe dataset. The Fig. 5 shows the comparison of the confusion matrices for four annotators, where the higher color intensity indicates a larger value, demonstrating that the proposed LFC-**x** can model the labeling process of multiple annotators.

Although LFC-**x** is proposed to learn a classifier from crowdsourced labels, it can also be applied to more general weakly supervised learning scenario, *i.e.,* singly-labeled scenario, where each instance is labeled by only one annotator while the information of annotator is not considered. Note that many LFC methods cannot be applied directly in singly-labeled scenario. To further verify the effectiveness of considering the awareness of instance features in our method on singly-labeled scenario. We implement an ablation study by comparing our LFC-**x** with LFC-0. The LFC-0 indicates that the instance impact matrix layer is frozen or removed in our LFC-**x**. In addition, we also compare with a traditional method Noisy Classifier which represents training a classifier directly on noisy singly-labeled dataset.

Table 3 Test accuracy of comparative methods combined with robust loss functions on synthetic and real datasets

Datasets	Crowd Layer (SL)	Crowd Layer (GCE)	LFC-**x** (SL)	LFC-**x** (GCE)
Parkinson speech	72.43/71.39	73.76/72.43	**75.20/72.72**	74.28/73.07
QSAR(biodegradation)	84.13/84.13	85.52/83.70	**86.79**/83.96	85.24/**84.70**
Statlog(image)	93.94/93.94	93.94/93.77	94.37/**94.37**	**94.81**/93.77
Waveform(version 1)	88.51/88.51	89.10/88.51	**90.60/89.60**	90.60/88.51
LabelMe	84.73/83.34	84.26/84.26	86.28/83.59	**86.75/84.66**
Sentiment Polarity	72.90/72.13	72.13/72.13	72.97/**72.36**	**73.40**/71.48

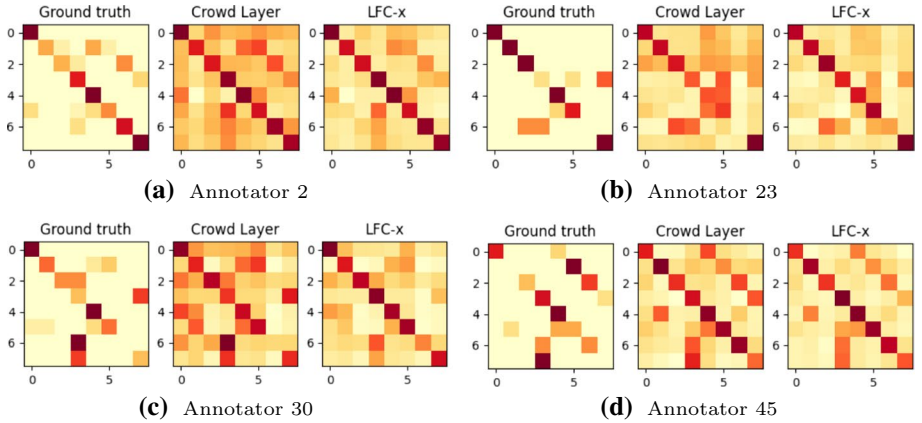

Fig. 5 Comparison between ground truth confusion matrices and learned ones by Crowd Layer and LFC-**x** on LabelMe dataset

Experimental setup: We conduct experiments on a synthetic singly-labeled dataset in a controlled noise setting, to evaluate the performance of methods under different noise levels and noise patterns. The synthetic dataset is generated by injecting noisy labels into the CIFAR-10 dataset. Unlike the two types of synthetic noise in the previous section, we generate *non-uniform labels* commonly used in singly-labeled scenario, where true labels are transformed to noisy labels with varying degrees according to a predefined noise pattern (Reed et al., 2014). The noise pattern is the transition between similar classes (visually or semantically). In CIFAR-10 dataset, we transform the class "aircraft" to class "bird" (class 0 to class 2), class "deer" to class "horse" (class 4 to class 7), by setting different degrees of probability such as $p \in \{0.10, 0.12, \ldots, 0.58, 0.60\}$.

As shown in Fig. 6, when the percentage of noise increases, the classification accuracy of LFC-**x** decreases more slowly than that of LFC-0 and Noisy Classifier, which illustrates that LFC-**x** can better deal with noisy labels. More importantly, LFC-**x** significantly outperforms LFC-0 since it considers the impact of instance features to construct the noise transition matrix.

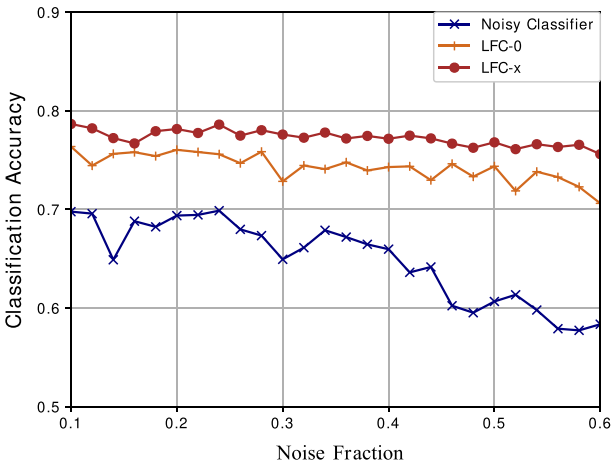

Fig. 6 Results with varying percentage of noisy labels

Table 4 Comparison of different variants regarding the construction of noise transition matrix on LabelMe dataset. The test accuracy of the best during training (left) and the last epoch (right) are listed

Methods	Test accuracy
LFC-**x**	85.86/84.60
LFC-**x** (addition w/ one stage)	83.95/81.25
LFC-**x** (dot product)	81.94/81.94

Variants of the LFC-x: We construct the noise transition matrix through addition operation between the instance impact matrix and the confusion matrix, and adopt a two-stage training procedure to learns the confusion matrix first and then the noise transition matrix. In addition, we consider two variants regarding the construction of the noise transition matrix: (1) LFC-**x** (addition w/ one stage): the noise transition matrix is constructed in the same way as LFC-**x**, but the training procedure is one-stage approach. Namely, this variant is equivalent to learning the noise transition matrix directly from instance features regardless of the class-level transition probability. It is worth mentioning that LFC-**x** (addition w/ one stage) can be regarded as an end-to-end neural network version of the method by Yan et al. (2010). (2) LFC-**x** (dot product): the noise transition matrix is obtained by dot product between the instance impact matrix and the confusion matrix, where the parameters of the instance impact matrix are generated with a normal distribution. Experiments on LabelMe dataset are run 10 times and an average of the test accuracy is reported in Table 4. The experimental results show that the two-stage addition operation between confusion matrix and instance impact matrix yields better results than other variants in the construction of the noise transition matrix.

6 Conclusion and future work

In this paper, we propose to learn a classifier from crowdsourced labels provided by multiple annotators. Specifically, we first construct the noise transition matrix by incorporating instance features into confusion matrix. Furthermore, we propose LFC-**x** that integrates a classifier module and a noise transition matrix module into a unified neural network in an end-to-end manner. Extensive experiments show the advantages of LFC-**x**, confirming the effectiveness of noise transition matrix compared to class-level confusion matrix. In addition, our approach can also integrate some other techniques to further improve the performance. For example, label-denoising techniques (Song et al., 2020) such as symmetric learning can be applied to LFC-**x** framework to further improve performance. We also verify the effectiveness of our method by considering the awareness of instance features in general noisy singly-labeled scenario. In future, we plan to extend LFC-**x** to other types of labels, *e.g.,* learning from crowdsourced sequence annotation (Lan et al., 2019) and multi-object bounding box annotation (Acuna et al., 2019).

Author contributions Jingzheng Li wrote and revised the manuscript, designed and implemented the research, and analyzed the results. Hailong Sun contributed to the revision of the manuscript and the analysis of the results. Jiyi Li contributed to the revision of the manuscript and the analysis of the results.

Funding This work was supported by National Natural Science Foundation under Grant Nos. (61932007, 61972013, 62141209).

Data availability The datasets are the benchmark datasets available online (Data Source available in manuscript).

Code availability The code is available at https://github.com/wumingv2/machine_learning_journal.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Acuna, D., Kar, A., & Fidler, S. (2019). Devil is in the edges: Learning semantic boundaries from noisy annotations. In: CVPR, pp 11,075–11,083.

Albarqouni, S., Baur, C., Achilles, F., et al. (2016). Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Transactions on Medical Imaging, 35*(5), 1313–1321.

Aydin, BI., Yilmaz, YS., Li, Y., et al. (2014). Crowdsourcing for multiple-choice question answering. In: AAAI, pp 2946–2953.

Berthon, A., Han, B., Niu, G., et al. (2021). Confidence scores make instance-dependent label-noise learning possible. In: International Conference on Machine Learning, PMLR, pp 825–836.

Cao, P., Xu, Y., Kong, Y., et al. (2019). Max-mig: An information theoretic approach for joint learning from crowds. In: Proceedings of the Seventh International Conference on Learning Representations.

Chen, P., Ye, J., Chen, G., et al. (2020a). Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. arXiv preprint arXiv:2012.05458.

Chen, Z., Wang, H., Sun, H., et al. (2020b). Structured probabilistic end-to-end learning from crowds. In: IJCAI, pp 1512–1518.

Cheng, J., Liu, T., Ramamohanarao, K., et al. (2020). Learning with bounded instance and label-dependent label noise. In: International Conference on Machine Learning, PMLR, pp 1789–1799.

Chu, Z., Ma, J., & Wang, H. (2020). Learning from crowds by modeling common confusions. AAAI.

Dawid, A. P., & 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.

Dua, D., & Graff, C. (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Gao, C., Lu, Y., & Zhou, D. (2016). Exact exponent in optimal rates for crowdsourcing. In: International Conference on Machine Learning, PMLR, pp 603–611.

Gaunt, A., Borsa, D., & Bachrach, Y. (2016). Training deep neural nets to aggregate crowdsourced responses. In: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, pp 242–251.

Goldberger, J., & Ben-Reuven, E. (2017). Training deep neural-networks using a noise adaptation layer. In: Proceedings of the Fifth International Conference on Learning Representations.

Han, B., Yao, J., Niu, G., et al. (2018). Masking: A new perspective of noisy supervision. *Advances in Neural Information Processing Systems, 31,* 5836–5846.

Han, B., Yao, Q., Yu, X., et al. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems, 31,* 8527–8537.

Han, B., Yao, Q., Pan, Y., et al. (2019). Millionaire: a hint-guided approach for crowdsourcing. *Machine Learning, 108*(5), 831–858.

Han, T., Sun, H., Song, Y., et al. (2016). Incorporating external knowledge into crowd intelligence for more specific knowledge acquisition. *IJCAI, 2016,* 1541–1547.

Imamura, H., Sato, I., & Sugiyama, M. (2018). Analysis of minimax error rate for crowdsourcing and its application to worker clustering model. arXiv preprint arXiv:1802.04551.

Jiang, L., Zhou, Z., Leung, T., et al. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *International Conference on Machine Learning, 2018,* 2304–2313.

Jiang, L., Huang, D., Liu, M., et al. (2020). Beyond synthetic noise: Deep learning on controlled noisy labels. In: International Conference on Machine Learning, PMLR, pp 4804–4815.

Jiang, Z., Zhou, K., Liu, Z., et al. (2021). An information fusion approach to learning with instance-dependent label noise. In: International Conference on Learning Representations.

Jin, Y., Carman, M., Zhu, Y., et al. (2020). A technical survey on statistical modelling and design methods for crowdsourcing quality control. *Artificial Intelligence., 287,* 103351.

Jin, Y., Wang, X., Long, M., et al. (2020). Minimum class confusion for versatile domain adaptation. *European Conference on Computer Vision* (pp. 464–480). Cham: Springer.

Kajino, H., Tsuboi, Y., & Kashima, H. (2013). Clustering crowds. *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence., 27,* 1.

Karimi, D., Dou, H., Warfield, S. K., et al. (2020). Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical Image Analysis, 65*(101), 759.

Khetan, A., & Oh, S. (2016). Achieving budget-optimality with adaptive schemes in crowdsourcing. *Advances in Neural Information Processing Systems, 29,* 4844–4852.

Khetan, A., Lipton, ZC., & Anandkumar, A. (2018). Learning from noisy singly-labeled data. In: Proceedings of the Sixth International Conference on Learning Representations.

Kim, HC., & Ghahramani, Z. (2012). Bayesian classifier combination. In: Artificial Intelligence and Statistics, pp 619–627.

Laine, S., & Aila, T. (2016). Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242.
Lan, O., Huang, X., Lin, BY., et al. (2019). Learning to contextually aggregate multi-source supervision for sequence labeling. In: arXiv preprint arXiv:1910.04289.
Li, S., Ge, S., Hua, Y., et al. (2020). Coupled-view deep classifier learning from multiple noisy annotators. In: Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, pp 4667–4674.
Li, X., Liu, T., Han, B., et al. (2021). Provably end-to-end label-noise learning without anchor points. In: International Conference on Machine Learning, PMLR, pp 6403–6413.
Li, Y., Rubinstein, BIP., & Cohn, T. (2018). Exploiting worker correlation for label aggregation in crowdsourcing. In: International Conference on Machine Learning, pp 3886–3895.
Liu, Q., Peng, J., & Ihler, AT. (2012). Variational inference for crowdsourcing. In: Advances in Neural Information Processing Systems, pp 692–700.
Liu, Y. (2021). Understanding instance-level label noise: Disparate impacts and treatments. In: International Conference on Machine Learning, PMLR, pp 6725–6735.
Liu, Y., & Guo, H. (2020). Peer loss functions: Learning from noisy labels without knowing noise rates. In: International Conference on Machine Learning, PMLR, pp 6226–6236.
Menon, A. K., Van Rooyen, B., & Natarajan, N. (2018). Learning from binary labels with instance-dependent noise. *Machine Learning, 107*(8), 1561–1595.
Misra, I., Lawrence Zitnick, C., Mitchell, M., et al. (2016). Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2930–2939.
Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: ACL, pp 115–124.
Patrini, G., Rozza, A., Krishna Menon, A., et al. (2017). Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1944–1952.
Raykar, VC., Yu, S., Zhao, LH., et al. (2009). Supervised learning from multiple experts: Whom to trust when everyone lies a bit. In: International Conference on Machine Learning, pp 889–896.
Reed, SE., Lee, H., Anguelov, D., et al. (2014). Training deep neural networks on noisy labels with bootstrapping. In: arXiv preprint arXiv:1412.6596.
Rodrigues, F., Pereira, FC. (2018). Deep learning from crowds. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, pp 1611–1618.
Rodrigues, F., Pereira, F., & Ribeiro, B. (2013). Learning from multiple annotators: distinguishing good from random labelers. *Pattern Recognition Letters, 34*(12), 1428–1436.
Rodrigues, F., Lourenc, M., Ribeiro, B., et al. (2017). Learning supervised topic models for classification and regression from crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(12), 2409–2422.
Sheng, VS., & Zhang, J. (2019). Machine learning with crowdsourcing: A brief summary of the past research and future directions. In: Proceedings of the Thirty-Forth AAAI Conference on Artificial Intelligence, pp 9837–9843.
Song, H., Kim, M., Park, D., et al. (2020). Learning from noisy labels with deep neural networks: A survey. arXiv preprint arXiv:2007.08199.
Sukhbaatar, S., Bruna, J., Paluri, M., et al. (2014). Training convolutional networks with noisy labels. arXiv preprint arXiv:1406.2080.
Tanno, R., Saeedi, A., Sankaranarayanan, S., et al. (2019). Learning from noisy labels by regularized estimation of annotator confusion. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 11,* 244–253.
Wang, Y., Ma, X., Chen, Z., et al. (2019). Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE International Conference on Computer Vision, pp 322–330.
Whitehill, J., Ruvolo, P., Wu, T., et al. (2009). Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems, 22,* 2035–2043.
Xia, X., Liu, T., Han, B., et al. (2020). Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems, 33*, 7597–7610.
Yan, Y., Rosales, R., Fung, G., et al. (2010). Modeling annotator expertise: Learning when everybody knows a bit of something. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp 932–939.
Yan, Y., Rosales, R., Fung, G., et al. (2014). Learning from multiple annotators with varying expertise. *Machine Learning, 95*(3), 291–327.

Yao, Y., Liu, T., Han, B., et al. (2020). Dual t: Reducing estimation error for transition matrix in label-noise learning. arXiv preprint arXiv:2006.07805.

Yao, Y., Liu, T., Gong, M., et al. (2021). Instance-dependent label-noise learning under a structural causal model. *Advances in Neural Information Processing Systems, 34,* 4409–4420.

Yin, L., Han, J., Zhang, W., et al. (2017). Aggregating crowd wisdoms with label-aware autoencoders. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017,* 1325–1331.

Zhang, H., Jiang, L., & Xu, W. (2019). Multiple noisy label distribution propagation for crowdsourcing. In: IJCAI, pp 1473–1479.

Zhang, Y., Zheng, S., Wu, P., et al. (2021). Learning with feature-dependent label noise: A progressive approach. arXiv preprint arXiv:2103.07756.

Zhang, Z., & Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems, 31,* 8778–8788.

Zheng, Y., Li, G., Li, Y., et al. (2017). Truth inference in crowdsourcing: Is the problem solved? In: Proceedings of the Forty-Second International Conference on Very Large Data Bases, pp 541–552.

Zhong, J., Yang, P., & Tang, K. (2017). A quality-sensitive method for learning from crowds. *IEEE Transactions on Knowledge and Data Engineering, 29*(12), 2643–2654.

Zhu, Z., Liu, T., & Liu, Y. (2021). A second-order approach to learning with instance-dependent label noise. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021,* 10113–10123.