

End-to-End Learning from Noisy Crowd to Supervised Machine Learning Models

Taraneh Younesian*, Chi Hong*, Amirmasoud Ghiassi*, Robert Birke[†] and Lydia Y. Chen*

*TU Delft, Delft, the Netherlands. Email: {T.Younesian, C.Hong, S.Ghiassi, Y.Chen-10}@tudelft.nl

[†]ABB Research, Baden-Dättwil, Switzerland. Email: robert.birke@ch.abb.com

Abstract—Labeling real-world datasets is time consuming but indispensable for supervised machine learning models. A common solution is to distribute the labeling task across a large number of non-expert workers via crowd-sourcing. Due to the varying background and experience of crowd workers, the obtained labels are highly prone to errors and even detrimental to the learning models. In this paper, we advocate using hybrid intelligence, i.e., combining deep models and human experts, to design an end-to-end learning framework from noisy crowd-sourced data, especially in an on-line scenario. We first summarize the state-of-the-art solutions that address the challenges of noisy labels from non-expert crowd and learn from multiple annotators. We show how label aggregation can benefit from estimating the annotators’ confusion matrices to improve the learning process. Moreover, with the help of an expert labeler as well as classifiers, we cleanse aggregated labels of highly informative samples to enhance the final classification accuracy. We demonstrate the effectiveness of our strategies on several image datasets, i.e. UCI and CIFAR-10, using SVM and deep neural networks. Our evaluation shows that our on-line label aggregation with confusion matrix estimation reduces the error rate of labels by over 30%. Furthermore, relabeling only 10% of the data using the expert’s results in over 90% classification accuracy with SVM.

Index Terms—crowd-sourcing, label aggregation, active learning, confusion matrix estimation

I. INTRODUCTION

Many artificial intelligence applications rely on supervised learning and labeled datasets, such as image classification [1], activity recognition [2], and sentiment analysis [3]. The dataset size and quality directly affect the performance of learned models [4] making labeling a daunting task. Crowd-sourcing [5] aims to curtail the labeling effort by submitting the data to a large crowd for labeling. Different from traditional labeling campaigns which assume the presence of (few) expensive experts providing the labels, crowd-sourcing relies on several cheap annotators with highly varying knowledge and level of interest [6]. While the labels can be easily gathered from the crowd, the quality of crowd-sourced labels is still an outstanding issue.

Label aggregation is an efficient method to distill the noise of crowd data by finding the consensus among all workers. The main algorithms in this area can be categorized in three directions: *Majority Voting*, probabilistic models via EM algorithms, and *Annotators’ Expertise Estimation* [7]. In Majority Voting, the label with the highest consensus among the workers, is selected as the aggregated label for the data [8]. Although some studies rely more on accurate workers [9], they require a (small) set of golden standard data with known

ground truth labels. Most studies treat the problem as an unsupervised learning task. EM based studies maximize the data likelihood to infer the unknown true labels [8], [10]. Some works also estimate the expertise of workers either via their confusion matrix [11], [12], [13] or reliability parameter [14], as well as the difficulty of items [15]. The common objective among them is to infer the true labels, independently from the subsequent supervised learning.

While these methods try to estimate the true label in an unsupervised manner, they exclude the information in the data samples themselves, e.g., features and informativeness of data. Active learning techniques [16] are designed to query extra information from an oracle for the data whose (true) labels are not readily available. Such an oracle is assumed to know the ground truth, but at high costs, e.g. a human expert. Hence, only the most informative/uncertain data is queried within a given query budget. The majority of active learning approaches focuses on off-line scenarios with constant budgets, except [17], [18], [19] that explore active learning on one by one drifting streaming data, however, their focus is on single label scenarios.

The efficiency of active learning relies on identifying the most informative instances to be labeled. Several measures have been proposed in the active learning literature e.g., based on class probability [20], entropy value [21] or posterior predictive densities [22]. Moreover, some methods try to identify the samples that cause the highest expected gain in the learning performance once they are labeled [23].

While crowd-sourcing studies have leveraged informative sample selection [14], [24], [25], the labeling quality of crowd workers remains a challenge. Usually none of the crowd workers is an expert in the problem field. Hence, it can be beneficial to leverage active learning with an expert labeler to assist the learning process [26]. Moreover, the connection between label aggregation and training classification models seems to be neglected in many crowd-sourcing studies, as they only focus on label information and exclude information lying in the data features, where active learning can play an important role.

The prior art in both crowd sourcing and label aggregation focus on off-line scenarios where all the data is available at once. However, in some applications the data is collected over time in a streaming setting [27]. The challenges in such on-line settings are small training data in each time step and concept drift [28]. The small sample set in on-line scenarios

prevents the learning process from convergence especially in deep learning [29]. Moreover, concept drift, i.e. the change in the statistical properties of the data, require on-line models to be adaptive to the change [28]. There are on incremental learning algorithms to train models progressively from new data [30], [29]. Only few consider noisy stream data [31], [32]. However these studies consider ensembles of several classifiers to detect noisy labels which is not scalable to large datasets used in deep learning.

In this paper we bring our end-to-end vision to marry crowd-sourcing with active learning for increased efficiency, e.g. higher accuracy at lower number of queries. [33] has been a pioneer for off-line scenario. We go beyond by discussing the challenges arising in on-line scenario, where data collection happens continuously, and proposing a solution to address label aggregation in an on-line manner. We show the gain of human experts in further improving the quality of learning systems and elaborate the benefit of employing a small clean set of data to estimate the annotators' confusion matrix. Finally, we perform a comparative evaluation against the off-line version of the proposed label aggregation method.

II. STATE OF THE ART

In this section we discuss the state of the art in the area of noisy crowds, dividing them into three categories. First we give an overview of the related works on annotators' confusion matrix estimation, then we review the existing research in offline label aggregation, and in the end, we discuss the works tackling noisy annotators with active learning.

A. Off-line Confusion Matrix Estimation

The label confusion matrix is a good indicator to determine the noise pattern and ratio. The diagonal elements represent the probability of the correct label while the off-diagonal elements indicate the probabilities to flip the correct label with a wrong one. Estimating the confusion matrix can help to correct noisy labels. Some estimation methods rely on a (small) set of clean samples with known ground truth. For instance, GLC [34] estimates confusion matrix assuming a small proportion of trusted data is available. This clean fraction of the dataset improves the estimation accuracy significantly [35], [36]. Furthermore, the study in [37] approximate the matrix of noisy labels stochastically by using correct labels. In addition, they improve the robustness of DNNs using forward loss correction. A few methods approximate the confusion matrix by using Generative Adversarial Networks (GAN) [38]. These works try to produce noise similar to the noise pattern in the dataset. The generated data is then used to identify the pattern and estimate the confusion matrix.

Some works estimate the confusion matrix by leveraging the prior knowledge in the field. For instance, Forward [39] assumes a known noise transition matrix/estimates and tries to minimize the distance between classification outputs and transition matrix. Masking [40] uses human cognition to estimate noise and build a noise transition matrix. Goldberger et al. [12] on the contrary, estimate confusion matrix using an

additional softmax layer in the DNNs. SELFIE [13] proposes a correction method regarding making high precision for unclear samples, then improves the estimated confusion matrix. The work in [11] estimates the annotators confusion matrix and the true labels simultaneously. As the network predicts the true label distribution, one can achieve the estimated noisy labels. Here, we investigate estimating the confusion matrix for multiple annotators, which is not studied well in the prior art [11]. In the case of multi annotators, which there is no access to correct them [12], [13], we can estimate each labeler's confusion matrix. After estimation, and combining each matrix knowledge, we can assess the quality of annotators and correct the noisy labels during training by using them in the loss function optimization.

B. Off-line Label Aggregating

Different works address label aggregation to distill the true label from redundant noisy labels posed as an unsupervised learning task. They differ in the estimation techniques as well as the latent variables on which the model relies.

One of the earliest works was proposed by Dawid and Skene [41]. They use the concept of confusion matrix to model the expertise of labelers estimated via an EM algorithm maximizing the data likelihood. BCC [42] is a probabilistic graphical model version of Dawid&Skene's EM. To learn the model parameters, the authors design a Gibbs sampler. During the learning process, the conditional distributions of the model parameters must be computed. This requires traversing all noisy labels in each iteration. Zhou et al. [43], [44] propose a minimax entropy estimator and its extensions to label aggregation. The authors set a separate probabilistic distribution for each worker-sample pair. Zhou and He [45] design a label aggregation approach based on tensor augmentation and completion. In these works [43], [44], [45], noisy labels are reorganized as a three-way label tensor. The aforementioned models can be regarded as off-line label aggregation. They target the data at hand and can not readily be adapted to learn incrementally to on-line scenarios. Here the data are collected periodically batch by batch. Off-line label aggregation needs to aggregate all batches together to achieve good performance on all labels. This is time consuming and not scalable, because the off-line model must wait to have all data at once to start or retrain on the whole accumulated data at each new batch arrival. Researchers [46], [47] have demonstrated that people's attention, fatigue and behaviors change over time. Therefore, we need on-line label aggregation algorithms which can continuously update the aggregation model according to the new observed labels to accurately infer the true labels.

C. Active Learning from Multiple and Noisy Annotators

Active learning aims to identify informative and representative unlabeled data samples and label them by an expert to increase the efficiency of the training procedure [16]. Traditional active learning methods consider an oracle knowing the ground truth for all the data readily available during the learning process [48]. However, this assumption does not apply

to real-world applications. A common solution is employing several labelers, weak or strong, in the form of crowd-sourcing. Therefore, leveraging active learning in crowd-sourcing has become an interesting topic.

While [49] considers imperfect labelers that may abstain from labeling, [9] assumes having multiple labelers with different costs and qualities. It actively selects both samples and labelers considering sample usefulness and labeler's accuracy and cost, assuming that all the labelers are prone to make mistakes. [50] focuses on the selection of informative samples in the presence of several non-expert labelers via majority voting of the labelers. Considering the same framework, an extension to unbalanced labels is studied in [51]. However, they fail to leverage the labelers based on their expertise in labeling. In contrast, [11] considers several noisy annotators with unknown expertise and jointly estimates the confusion matrix of the annotators and the true label distribution by minimizing the cross entropy function between predicted noisy labels and given noisy labels. To estimate the workers' expertise, [14] uses a low rank representation for the workers' skills and estimates this representation using EM algorithm. A bayesian neural network is used in this study to model the uncertainty in the data to choose the most uncertain samples for labeling by the expert crowd.. Asking the workers to provide their confidence level while labeling has been studied in [24], although relying on the user provided information seems challenging. Similarly, [25] asks workers to chose the option *unsure* if applicable, which is similar to the abstention of labelers in [49]. Moreover, a few studies consider annotators with various costs and adjust their active learning algorithm to select annotators with balanced cost/accuracy [52], [53], [54].

III. CHALLENGES

In this section, we discuss the challenges arising from label aggregation for online data streams with multiple annotators.

A. On-line Label Aggregating

On-line learning requires label aggregation algorithms to continuously aggregate the labels of a data stream. Because of storage limits, regulation constraints or other factors, data batches are available for a limited duration. Therefore on-line aggregating is different from traditional label aggregation tasks which process all observed noisy labels at once. Therefore, the design of a new learning framework for on-line scenarios is essential.

The first challenge is how to make use of the knowledge of the old received batches when aggregating a new observed batch. The knowledge of the old batches is valuable for the label aggregation algorithm to precisely evaluate the behaviors of the non-professional crowd workers. However, the old batches are missing when we get the new batch. So our model must be capable to continuously update its parameters according to the knowledge learned from every observed batch.

The second challenge is to design a suitable learning method and optimization goal for the aforementioned appli-

cation scenario. EM algorithms [41], Gibbs samplers [42] and tensor completion methods [45] have been proposed for label aggregation. However, they usually need to travel and count all noisy labels in each iteration. Therefore, these methods are not applicable for the on-line data arrival setting. The challenge then is to find an optimization method which can update the label aggregation model in the presence of a data stream. Besides, it is vital to design a reasonable optimization goal for the model to accurately aggregate the observed noisy labels.

B. Multiple Annotators

Having multiple annotators with different expertise rises the question of which annotator to choose for labeling each data sample. Although methods like label aggregation try to overcome this issue by combining the opinion of all annotators, there is still no guarantee that the aggregated labels are accurate. The challenges are: annotators having different level of knowledge for the task, some annotators could be malicious or simply not willing to put effort for the task, or there could be a relation between the data category and the annotator's expertise level [55]. Also, there might be some prior knowledge about each annotator, their skill and cost [25], [24], [56]. As mentioned earlier, majority voting is one of the simplest ways to combine the annotators' knowledge. However, in difficult cases where most of the annotators could make mistakes, majority voting or other label aggregation methods can fail [57]. Another category of methods tries to estimate the expertise of the annotators to select the most skilled ones [55], [14]. Taking steps further, selective majority voting [58] applies majority voting to the D most reliable voters based on their estimated expertise. These views, fail to consider the difficulty of the samples as well as the change in the expertise over time.

One could use an expert opinion to verify the aggregated labels. However, since the expert opinion is expensive [56], another challenge is to reduce this cost by efficiently choosing important samples to be relabeled by the expert [56]. Furthermore, as mentioned above, in the cases where there is a relation between the annotators' expertise and the data sample, it is vital to identify those samples for further investigation. In this case, expert knowledge could be used to label these informative samples [59].

IV. DEEP OFF-LINE LABEL AGGREGATION

In this section we introduce a method to estimate the confusion matrix of the workers, to select high quality labeler in an off-line manner, benefiting from a small clean set with known true labels.

A. Off-line Multi-Annotator Confusion Matrix Estimation: MCE

In many real-world applications, a small set of correctly labeled data is available. In these cases, an effective estimation method is to extract the annotators' confusion matrix probabilities by using a small proportion of trusted data.

Consider that each image in dataset $\mathbf{x}_i \in \mathbb{R}^{n \times m}$ has a set of labels from different annotator $\mathcal{Y}_i = \{y_i^1, y_i^2, \dots, y_i^K\}$ where $y_i^{(k)} \in \{1, 2, \dots, C\}$ denotes the i^{th} annotated label from k^{th} annotator. Also, n , m and K are number of images, features and annotators, respectively. The confusion matrix of each annotator $C^{(k)}$ is estimated by training a DNN on the dataset $\mathcal{D} = \{(\mathbf{x}_1, \mathcal{Y}_1), \dots, (\mathbf{x}_n, \mathcal{Y}_n)\}$ which is labeled by annotators.

1) *Confusion Matrix Estimation*: To achieve a robust DNN training with images labeled by multiple annotators, we leverage additional information from a small set of clean samples to estimate the confusion matrix which is introduced by [34]. The noise confusion matrix guides DNNs to recover the true label of each image. They can either derive the true labels directly, then train DNNs with new cleansed data or correct the loss function implicitly. The proposed method by [34] starts with training an image classifier on the noisy label data. We train $f_{(k)}(\cdot, \Theta)$ on the dataset $\mathcal{D}_{(k)} \subset \mathcal{D}$ which denotes the dataset annotated by annotator k . In other word, each annotator generates $\mathcal{D}_{(k)} = \{(\mathbf{x}_1, y_1^{(k)}), \dots, (\mathbf{x}_n, y_n^{(k)})\}$ for training corresponding DNNs $f_{(r)}(\cdot, \Theta)$. After training each network, the elements of confusion matrix $C_{i,j}^{(k)}$ are approximated via a small fraction of trusted data \mathcal{D}' including true label y' . Given $A_i \subset \mathcal{D}'$ the subset of trusted data, each elements of $C_{i,j}^{(k)}$ can be estimated by:

$$\hat{C}_{i,j}^{(k)} = P(y^{(k)} = j | y' = i) \approx \frac{1}{|A_i|} \sum_{\mathbf{x} \in A_i} f_{(k)}(y^{(k)} = j | \mathbf{x}, \Theta) \quad (1)$$

where $f(y^{(k)} = j | \mathbf{x}, \Theta)$ denotes the probability of predicted label of \mathbf{x} having class j . Hence, estimated confusion matrix $\hat{C}_{i,j}^{(k)}$ is the mean predicted probability of class j for true label of class i for the trusted data samples. The estimation depends on the annotator's skills and the number of clean labels in trusted data for each class. Annotators skills include the number of incorrect labels assigned to each instance, also in many cases, the pattern of wrong labels follows a specific transition function.

As mentioned in Eq. 1, a trained DNN is used to extract the elements of the confusion matrix. In this method, not only the quality of input dataset is essential, but also the architecture of DNN plays a crucial role in approximating a useful noise confusion matrix.

2) *Multiple Annotators Multiple Confusion Matrices (MCE)*: After estimating the noise confusion matrix, we need to find the best annotator among them. $\hat{C}^{(k)}$ is our metric to identify the most accurate one for each annotator k . The diagonal elements of the matrix $\hat{C}_{i,i}^{(k)}$ indicate the probability that a label is correctly annotated. Generally, we can find the least noisy datasets based on the confusion matrix by calculating the average of $\mathit{trace}(\cdot)$ for each matrix $\hat{C}^{(k)}$. We first define a set \mathcal{T} consisting of the average value of $\mathit{trace}(\cdot)$ for each k . The set can be written as:

$$\mathcal{T} = \left\{ \frac{1}{C} \mathit{trace}(\hat{C}^{(1)}), \frac{1}{C} \mathit{trace}(\hat{C}^{(2)}), \dots, \frac{1}{C} \mathit{trace}(\hat{C}^{(K)}) \right\} \quad (2)$$

where $\mathit{trace}(\hat{C}^{(k)}) = \sum_{i=1}^C \hat{C}_{i,i}^{(k)}$. To choose the most clean dataset, we consider \mathcal{T} as a reference to illustrate average noise ratio for each annotator. The selected dataset can be written as the following:

$$\mathcal{D}_S = \{(\mathbf{x}_i, y_i^{(k)}) | k = \mathit{index}(\min_{j \in \mathcal{T}} j)\} \quad (3)$$

where $\mathit{index}(\cdot)$ describes the index of an element in a set. Next, we can train a DNN with the selected dataset \mathcal{D}_S , which contains less corruption than other labeled datasets. In other word, we choose the most accurate annotator, and as a result the obtained labels will be more reliable than the rest. The aforementioned method works based on the diagonal elements in each annotator's confusion matrix.

V. PROPOSED ON-LINE LABEL AGGREGATION

With increasing practise of on-line data curation, the label set is continuously updated or may not be possible to store for all applications. Moreover, training a neural network can be expensive in some cases and requires the dataset to be available ahead of time. Intelligent selection of the clean labeled data could be more efficient for the learning process. In this section, we introduce a novel end-to-end framework to aggregate labels of the data annotated by crowd workers in an on-line streaming data setting. To further improve the quality of the aggregated labels, benefiting from a classifier, we leverage active learning to cleanse/relabel informative samples by an expert and train the model on high quality data.

A. Problem Definition

We focus on an on-line data arrival setting that consists of two steps: *i*) label aggregation, and *ii*) active learning. Consider data which periodically streams into the classifier in small batches D for training. The instances of the training data are labeled by the crowd, as each instance takes the form $(\mathbf{x}_j, \tilde{y}_{j,1}, \dots, \tilde{y}_{j,K})$, where $\tilde{y}_{j,k}$ means the potentially noisy label provided by worker k for sample j . \mathbf{x}_j represents the feature inputs. Therefore, we have the feature inputs together with multiple potentially noisy labels in an instance. Our task is to train a classifier with this data stream.

The data stream will be processed by label aggregation first. The label aggregation algorithm can give an aggregated label (the predicted true label) \tilde{y}_j for each instance j according to the corresponding noisy labels $\mathbf{Y}_j = \{\tilde{y}_{j,1}, \dots, \tilde{y}_{j,K}\}$. The label aggregation aims to lower the noise rate in the labels. However, since the label aggregation algorithm is not perfect, the aggregated labels can be wrong. Therefore, active learning is essential to clean the aggregated labels. In contrast to label aggregation, in active learning, the information of the machine learning classifier is also used to detect informative/useful samples to relabel. Note that it is expensive to verify every sample because of the limited budget. Therefore, the label aggregating process before active learning is useful to increase the quality of the labels. The goal of the active learning step is to identify samples potentially mislabeled by the label aggregation step and relabel them by an oracle to reach the ultimate goal of high classification accuracy. Finally, the

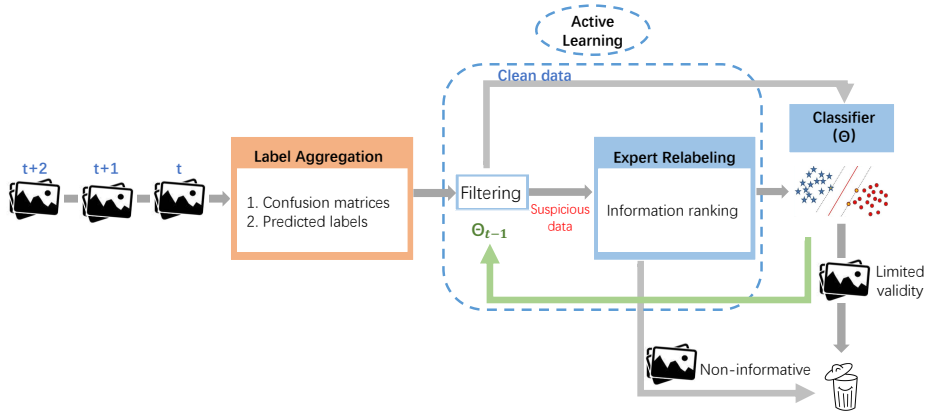


Fig. 1. The Workflow of the End-to-End On-line *NN-MC* with Active Learning

classifier will be trained by high quality data in a supervised manner. Figure 1 shows the full framework of our algorithm. The details of each step will be discussed in the following sections.

B. On-line Label Aggregating

1) *Basic Setting*: In our online learning setting, the main task of label aggregating is to choose one label \tilde{y}_j for each sample j according to its noisy labels $\mathbf{Y}_j = \{\tilde{y}_{j,1}, \dots, \tilde{y}_{j,K}\}$. In this online case, for each batch of samples, our label aggregation model will give the corresponding aggregated labels in real time.

2) *Algorithm Framework*: We use p to denote our label aggregation model. In order to update the model in our online learning setting, we use stochastic optimization methods like SGD and RMSProp [60]. These methods are easy to apply for mini-batch learning which fits our on-line learning setting well. Then, we need to choose an optimization goal for the optimization method. Our goal is to maximize the data likelihood of the noisy labels. The optimization function of this algorithm is designed according to variational inference. We use an implicit distribution q as the approximate distribution. Then we set minimizing the Kullback-Leibler divergence between $p(\tilde{\mathbf{y}}|\mathbf{Y})$ and $q(\tilde{\mathbf{y}}|\mathbf{Y})$ as the optimization function, where minimizing the Kullback-Leibler divergence is equivalent to maximizing the evidence lower bound of the log data likelihood $\log p(\mathbf{Y})$.

3) *Neural Network based Multi Class Aggregation (NN-MC)*: According to the algorithm framework, we can define our model by specifying the forms of p and q . In order to apply stochastic optimization methods in the label aggregation model, the loss function must be differentiable with respect to the model parameters of p and q . The definition of our *NN-MC* model is discussed below. p is defined using the concept of confusion matrix [41]. $C^{(k)}$ represents the confusion matrix of worker k , where its element $C_{i,j}^{(k)}$ is the probability that worker k gives a label j when the true label of the item is i . That is to say, $C_{i,j}^{(k)} = p(\tilde{y}_{j,k} = c | y_j = t)$. In *NN-MC*, q is a neural network which represents a distribution $q(\tilde{y}_j|\mathbf{Y}_j)$.

Then, according to the definition, we can calculate the loss function and apply mini-batch stochastic learning for *NN-MC*.

In on-line learning scenarios, at the beginning, *NN-MC* uses the noisy labels of some samples to initialize the model parameters. After the initialization, the data batches will be input into *NN-MC* one by one. For each batch of noisy labels, *NN-MC* uses them to update the model parameters (e.g., confusion matrices, neural network parameters) and then estimates the most confident labels (aggregated labels) for the corresponding samples. After learning the values of the confusion matrices, it is easy to compute the aggregated labels by maximizing the data likelihood of the observed noisy labels. *NN-MC* can also be applied to off-line cases by updating the model parameters with all noisy labels and aggregating all the noisy labels using the learned parameters together.

It should be note that as the introduced off-line confusion matrix estimation *MCE* aims to estimate the confusion matrices, we can modify the confusion matrix estimation step of the *NN-MC* based on the proposed *MCE*, and leverage the rest of *NN-MC* approach by using maximum likelihood to extract the aggregated labels. In other words, in the applications where a small clean data is available, *MCE* can assist *NN-MC* to estimate the expertise of the workers, however, only in an off-line setting.

C. Active Label Cleansing

After getting the aggregated labels for each batch of data, we aim to use the expert knowledge to further cleanse the potential wrongly aggregated labels, i.e. noisy labels. This step uses the relationship between a classifier and the features of the data samples. Our framework is based on streaming data where the data arrives in small batches, is used in the training process and then discarded. Each data instance $(\mathbf{x}_j, \tilde{y}_j)$ in the upcoming batch D contains feature inputs $\mathbf{x}_j \in \mathcal{X} \subset \mathbb{R}^m$ and a potentially noisy aggregated label $\tilde{y} \in \mathcal{Y} := \{1, \dots, N\}$. The goal is to relabel informative wrongly annotated samples by their true label y . Our algorithm consists of three steps: *i) filtering*, *ii) informative sample selection*, and *iii) relabeling*.

1) *Filtering*: The first step is to identify the samples that have been annotated with a wrong label during the label aggregation process. One way is to leverage the classifier’s prediction. By comparing the classifier’s prediction \hat{y}_j with the aggregated label \tilde{y}_j , we consider a sample to be clean if the predicted label and the aggregated label are the same, i.e. $\hat{y}_j = \tilde{y}_j$, and add them to the clean set $C = \{(\mathbf{x}_j^c, y_j)\}$. The rest of the samples are considered suspicious $U = \{(\mathbf{x}_j^u, \tilde{y}_j)\}$.

2) *Informative Sample Selection*: The next step is to identify the informative samples among the suspicious set U to query their true label from the expert. The purpose of this step is to avoid the cost of relabeling the whole suspicious set, due to the expensiveness of the expert. We use two methods to measure informativeness and rank the samples: *Least Confident(LC)* and *Best-versus-second-best (BvSB)* [61]. Both of these methods consider the samples highly informative, if the classifier’s uncertainty in their classification is high. Consider the classifier’s prediction probability vector for the data sample \mathbf{x}_j as $\mathbf{p}(\mathbf{x}_j)$, therefore, p_{best} and $p_{second-best}$ represent the most likely and the second most likely class to assign for that data sample. LC compares samples based on how least confident the model is to classify them, i.e. $I(\mathbf{x}_j) = p_{best}(\mathbf{x}_j)$. Whereas, BvSB compares samples based on how much the model is confused between the two most probable classes, i.e. $I(\mathbf{x}_j) = p_{best}(\mathbf{x}_j) - p_{second-best}(\mathbf{x}_j)$. The value $I(\mathbf{x}_j)$ shows the informativeness of the data sample \mathbf{x}_j . The lower the $I(\mathbf{x}_j)$ is, the more difficult and confusing the sample is, therefore the sample is highly informative and useful to be relabeled.

To select highly informative samples, we rank them based on their $I(\mathbf{x}_j)$ value in an increasing order.

3) *Relabeling*: After ranking the samples based on their informativeness, we select the top r samples to relabel by the expert labeler, i.e. the oracle. We add the relabeled clean samples to the samples filtered as clean in the filtering step and re-train the classifier with the clean dataset for the current batch. This process is repeated at each batch arrival.

VI. PRELIMINARY EVALUATION

In this section we evaluate our off-line confusion matrix estimation *MCE*, as well as the proposed end-to-end on-line *NN-MC* with active learning, on two classifiers. First, we compare our proposed off-line confusion matrix estimation and label aggregation methods using convolutional neural networks. Second, we present our experimental results for end-to-end on-line label aggregation with active learning using SVM.

A. Experimental Setup

1) *Datasets*: We evaluate the introduced frameworks on two types of datasets. The first type represents less complicated smaller sized data, with fewer and handcrafted features that are suitable to train standard ML approaches. The second type instead uses directly the pixels values and represents the deep learning approach which integrates feature selection

TABLE I
SUMMARY OF THE MAIN PROPERTIES OF EVALUATED DATASETS.

Dataset	letters	pendigits	usps	optdigits	CIFAR-10
# classes k	26	10	10	10	10
# features d	16	16	256	64	32x32x3
# train	15000	7494	7291	3823	50000
# test	5000	3498	2007	1797	10000

into the training process. For the first type we use four multi-class datasets with different sizes and features from the *UCI machine learning repository* [62]: *letters*, *pendigits*, *usps* and *optdigits*. The *letters* dataset tries to identify the 26 capital letters of the English alphabet with 20 different fonts. The remaining three target the recognition of handwritten digits via different handcrafted features and from different number of people. These datasets are used to evaluate *NN-MC* and active relabeling. For the second type we use the well-known CIFAR-10 dataset [63]. This dataset consists of colored 32×32 -pixel images divided into ten categories. This dataset is used for the comparison between *MCE* and *NN-MC* for confusion matrix estimation. This dataset is selected since *MCE* uses deep neural networks that are successful in classifying more complex datasets like CIFAR-10. Table I summarises the characteristics of both groups of datasets. Since these datasets contain only one label per data, we need to synthesize the noisy crowd, where each worker assigns a noisy label to the data using the procedure in following section.

2) *Annotation Noise*: To model imperfect annotators (workers), we use four noise pattern with noise rate ε . Worker k with noise rate ε assigns the ground truth label for each data point with probability of $1 - \varepsilon$ and makes a mistake (assigns another class) with probability ε . The wrong class can be selected in various ways that are associated with different noise patterns as follows:

- *truncnorm*: uses a truncated normal distribution $\mathcal{N}^T(\mu, \sigma, a, b)$ motivated by [64]. We scale $\mathcal{N}^T(\mu, \sigma, a, b)$ by the number of classes C and center it around a target class \tilde{c} by setting $\mu = \tilde{c}$ and use σ to control how spread out the noise is. a and b simply define the class label boundaries, i.e. $a = 0$ and $b = C - 1$. We set $\mu = 3$ and $\sigma = 1$ in our experiments.
- *bimodal*: is an extension of *truncnorm*. This pattern combines two truncated normal distributions. It has two peaks in μ_1 and μ_2 with two different shapes controlled by σ_1 and σ_2 . The peaks are centered on two different target classes $\mu_1 = \tilde{c}_1$ and $\mu_2 = \tilde{c}_2$. We use $\mu_1 = 3$, $\sigma_1 = 1$, $\mu_2 = 7$, and $\sigma_2 = 0.5$.
- *flip*: considers partial targeted noise where only a subset of classes, $\{2, 3, 4, 5, 9\}$ in our example, are affected by targeted noise, i.e. swapped with a specific other class [65].
- *uniform*: uniformly selects one of the wrong labels.

Figure 2 illustrates the confusion matrix of an annotator with the introduced noise patterns using $\varepsilon = 0.6$.

3) *Training Parameters for UCI Datasets*: As *NN-MC* needs an initial training phase, we use an initial set of 50

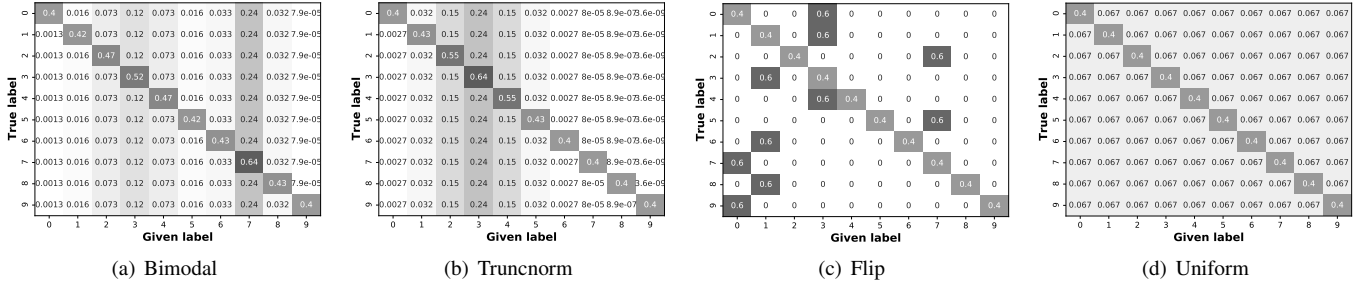


Fig. 2. Generated confusion matrices, where the number of classes is 10 and $\varepsilon = 0.6$

samples for all datasets except *letters*. For *letters*, since the number of classes is higher and the dataset is more complex, we let the initial set be 150 samples. The instances are chosen randomly from the training set. To speed up training, we limit the datasets to the total training size of $N = 1050$ (and $N = 8000$ for *letters*), including the initial set. After the initial clean data batch, noisy data arrives in batches of 50 instances. For the classifier with the UCI datasets we use SVM.

To synthesize the crowd labels, we evaluated *NN-MC* using the following parameters: number of workers $K \in \{6, 8, 10\}$, empty proportion $e \in \{0.1, 0.2, 0.3\}$, and noise rate $\varepsilon \in \{0.4, 0.6, 0.8\}$ with all four noise patterns mentioned above. Note that the empty proportion indicates the proportion of missing labels for each worker. Moreover, the mentioned noise rates ε are the average of the noise rates of all the workers. The noise rates of each worker are randomly selected in the range of 10% to 90% with respect to the average of ε .

Our algorithm queries the true label of the $r = 5$ most informative noisy samples per batch via the oracle. We repeat each experiment 50 times and report the average final accuracy computed on the test set.

4) *Networks Architecture and Training*: For *NN-MC*, we use a Multi-layer Perceptron (MLP) with two hidden layers with 64 and 32 neurons respectively, and *tanh* activation function, where the input layer size corresponds to the number of workers. For *MCE*, for CIFAR-10 we consider a CNN architecture which consists of 6 convolutional layers followed by 2 fully connected layers [66]. The activation function is ReLU. To estimate each annotator confusion matrix, our DNN is trained for 130 epochs using SGD optimizer with momentum 0.9, weight decay 10^{-4} , learning rate 0.01, and mini-batch size of 128 instances.

B. Confusion Matrix Comparison

Here we compare the performance of off-line *NN-MC* and *MCE*. In Table II reports the error rate of aggregated labels for *NN-MC* and *MCE* with different number of annotators (workers). We vary the noise rate $\varepsilon \in \{0.4, 0.6, 0.8\}$, which this value is the mean of noise over the annotators for the *uniform* noise pattern. Furthermore, we test the effect of mixed noise pattern, where the annotators noise patterns are different from each other while the average noise rate is $\varepsilon = 0.6$ (termed mixed pattern). We set the patterns with 10 workers based on

TABLE II
ERROR RATES (%) FOR DIFFERENT CONFUSION MATRIX ESTIMATION METHODS WITH EMPTY PROPORTION OF 0.1, AND DIFFERENT NOISE RATES AND PATTERNS.

# of Workers = 6				
Method	Uniform 0.4	Uniform 0.6	Uniform 0.8	Mixed Patterns 0.6
<i>NN-MC</i>	7.47	29.61	67.77	18.75
<i>MCE</i>	6.10	28.33	55.70	8.28
# of Workers = 8				
<i>NN-MC</i>	3.77	21.76	63.72	12.34
<i>MCE</i>	3.66	21.73	54.34	4.90
# of Workers = 10				
<i>NN-MC</i>	1.77	16.77	60.42	8.08
<i>MCE</i>	2.88	19.37	53.99	4.51

the following sequence: [*bimodal*, *truncnorm*, *flip*, *uniform*, *bimodal*, *truncnorm*, *flip*, *uniform*, *bimodal*, *truncnorm*]. For the case of 6 and 8 workers, we use the first 6 and 8 patterns respectively. Across the experiment in Table II, *MCE* obtains a better error rate than *NN-MC*. It shows that a small proportion of trusted data with clean samples can improve the estimation of the confusion matrix. The difference between these two models becomes larger by increasing the noise rate. In other words, when the number of annotators is 6, for $\varepsilon = 0.4$ and $\varepsilon = 0.8$, the difference is 1.37 and 12.07, respectively. In addition, increasing the number of annotators reduces the error rate as it increase the chance of extracting the true label. Moreover, an interesting observation is on the effect of the mixed noise pattern. As the table shows, having a mixed pattern results in lower error rates compared to the case of all *uniform* noises.

C. NN-MC Performance on UCI Dataset

We extensively analyze the performance of on-line *NN-MC* and off-line *NN-MC* under different settings of number of workers, empty proportions, noise rates and noise patterns in Figure 3 and 4. We investigate the influence of every single parameter by changing one parameter value while fixing the other parameter values. The fixed parameters are $k = 6$, $e = 0.1$, $\varepsilon = 0.6$, and the noise pattern *bimodal*. As a baseline, we use Majority Voting and compare its performance with *NN-MC*.

We can see that label aggregation methods can achieve higher accuracy when we have more workers to provide poten-

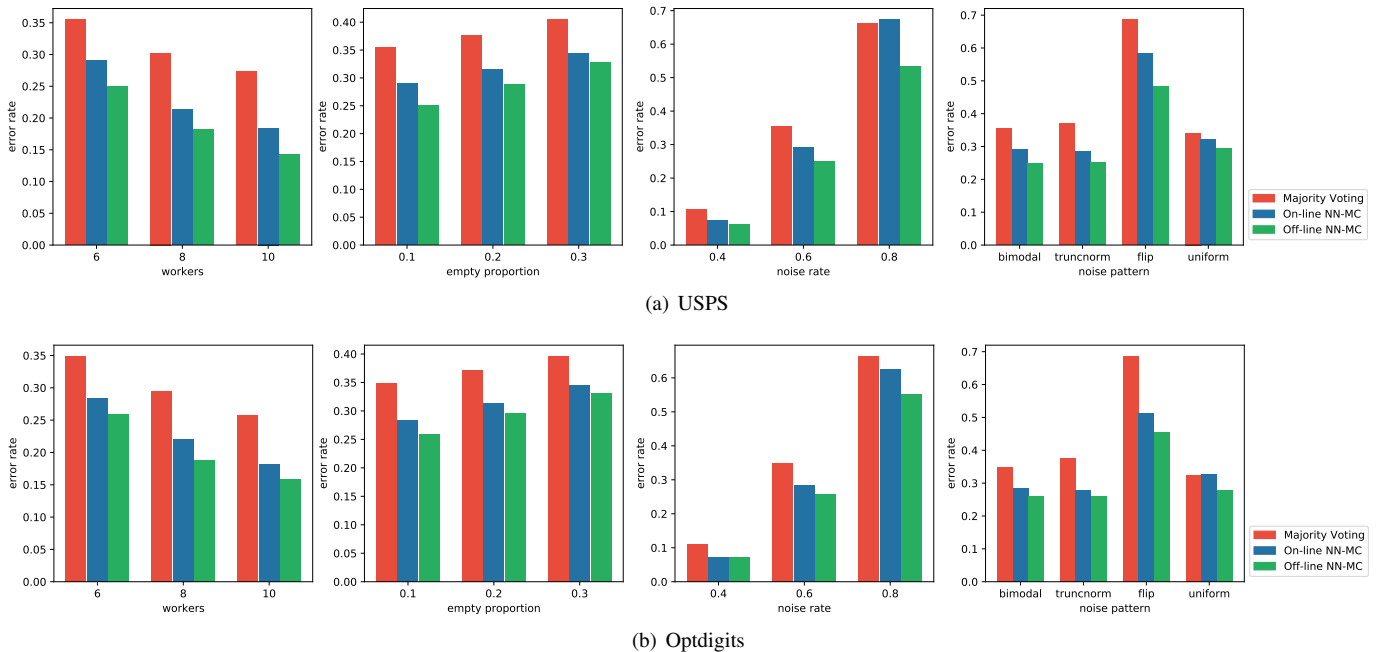


Fig. 3. Error rates for on-line and off-line *NN-MC*, and Majority Voting for *usps* and *optdigits*. At each row, one parameter is changing while the other parameter are fixed. The values for fixed parameters are: number of workers $K = 6$, empty proportion $e = 0.1$, noise rate $\varepsilon = 0.6$, noise pattern bimodal.

TABLE III

LABEL ERROR-RATES (%) FOR ON-LINE LABEL AGGREGATION, AND FINAL ACCURACY WITH ACTIVE LEARNING FOR $r = 5$, $K = 6$, $e = 0.1$, $\varepsilon = 0.6$ AND *bimodal* NOISE PATTERN.

Method	USPS	Optdigits	Pendigits	Letters
Error Rate On-line <i>NN-MC</i>	29.04	28.34	28.85	23.51
Accuracy On-line <i>NN-MC</i> +AL (<i>LC</i>)	89.64	88.89	89.94	86.20
Accuracy On-line <i>NN-MC</i> +AL (<i>BvSB</i>)	96.12	92.36	91.62	88.24

tially noisy labels for each instance (see plots in first column). The reason is that more eligible workers for each instance corresponds to more information to correctly calculate the data likelihood and make a more precise prediction. According to the plots in the second column, we can see that a lower empty proportion will increase the accuracy of all methods, because the lower empty proportion represents more labels for each instance on average. The plots in the third column show that lower noise rates of the potentially noisy labels can help the label aggregation algorithms make a better prediction. In the last column, the *flip* noisy pattern has the worst accuracy for all datasets. Comparing to other patterns, for each true label class (each row in the confusion matrix), *flip*'s probability mass concentrate on one single wrong label. This phenomenon will significantly disturb the label aggregation algorithm to correctly learn the confusion matrices of the workers according to the potentially noisy labels.

D. Results on Relabeling with Active Learning

We study the effect of active learning to identify and relabel wrong aggregated labels after applying on-line *NN-MC*. For each batch of data that arrives, first we find the aggregated labels and then further cleanse them by applying

active learning with *LC* and *BvSB* to relabel the informative data. Table III shows the error rate for on-line *NN-MC*, and the effect on accuracy of incorporating active relabeling after *NN-MC* with the fixed parameters used in the previous section. As the results show, active relabeling helps in achieving a high accuracy by only relabeling 10% of data instances per batch. Among the datasets, *letters* seems to be more difficult to classify since it has more classes, although label aggregation succeeds in estimating a more clean label set. Among the active learning methods, *BvSB* performs better in selecting the informative data, since it focuses on the two top classes, whereas *LC* considers only the highest probable class.

VII. CONCLUSION

In this paper we address the challenges and solutions of how to design an end-to-end learning framework from noisy crowd-sourced data, with special focus on on-line scenarios. We illustrate the challenges arising with on-line label aggregation of multiple workers. We propose a visionary framework which incrementally combines noisy data, expert relabelling, and supervised models for better learning results. We introduce a method to estimate the expertise of multiple annotators by estimating their confusion matrix while leveraging a small clean dataset. To increase the quality of the labels and benefit from an expert labeler, we relabel suspiciously noisy aggregated labels in an efficient manner. Our results show that the proposed label aggregation can successfully lower the labeling error rate by more than 30%, while relabeling only 10% of the most informative samples, which results in a highly accurate classification model.

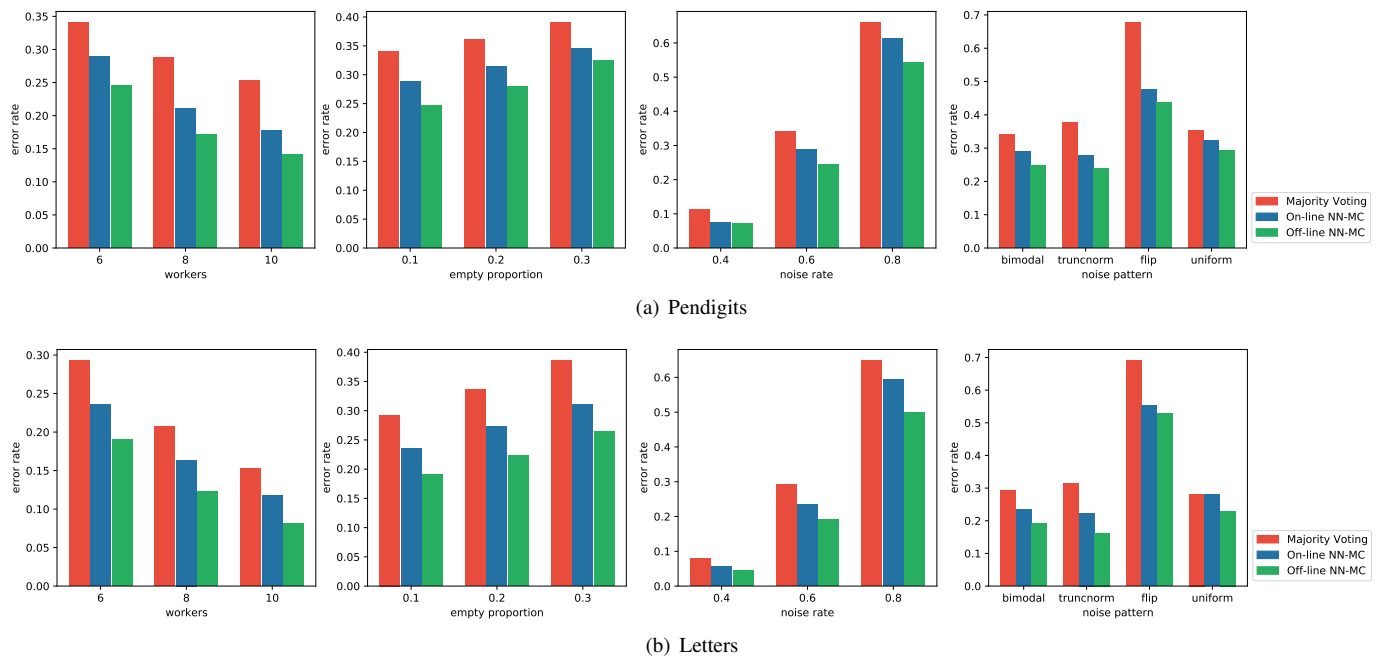


Fig. 4. Error rates for on-line and off-line *NN-MC*, and Majority Voting for *pendigits* and *optdigits*. At each row, one parameter is changing while the other parameter are fixed. The values for fixed parameters are: number of workers $K = 6$, empty proportion $e = 0.1$, noise rate $\epsilon = 0.6$, noise pattern bimodal.

VIII. ACKNOWLEDGMENT

This work has been partly funded by the Swiss National Science Foundation NRP75 project 407540_167266.

REFERENCES

- [1] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 27, no. 12, pp. 2591–2600, 2017.
- [2] B. Krawczyk, "Active and adaptive ensemble learning for online activity recognition from data streams," *Knowl. Based Syst.*, vol. 138, pp. 69–78, 2017.
- [3] J. Smailovic, M. Grcar, N. Lavrac, and M. Znidarsic, "Stream-based active learning for sentiment analysis in the financial domain," *Inf. Sci.*, vol. 285, pp. 181–203, 2014.
- [4] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *ICLR*, OpenReview.net, 2017.
- [5] M. Yuen, I. King, and K. Leung, "A survey of crowdsourcing systems," in *PASSAT*, pp. 766–773, IEEE Computer Society, 2011.
- [6] J. Howe, "Crowdsourcing: Why the power of the crowd is driving the future of business," 2008.
- [7] M. Georgescu, D. D. Pham, C. S. Firan, W. Nejdl, and J. Gaugaz, "Map to humans and reduce error: crowdsourcing for deduplication applied to digital libraries," in *CIKM* (X. Chen, G. Lebanon, H. Wang, and M. J. Zaki, eds.), pp. 1970–1974, ACM, 2012.
- [8] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, pp. 1297–1322, 2010.
- [9] S. Huang, J. Chen, X. Mu, and Z. Zhou, "Cost-effective active learning from diverse labelers," in *IJCAI* (C. Sierra, ed.), pp. 1879–1885, ijcai.org, 2017.
- [10] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *WWW* (A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, eds.), pp. 469–478, ACM, 2012.
- [11] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman, "Learning from noisy labels by regularized estimation of annotator confusion," in *CVPR*, pp. 11244–11253, 2019.
- [12] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," in *ICLR*, 2017.
- [13] H. Song, M. Kim, and J.-G. Lee, "Selfie: Refurbishing unclean samples for robust deep learning," in *ICML*, pp. 5907–5915, 2019.
- [14] J. Yang, T. Drake, A. C. Damianou, and Y. Maarek, "Leveraging crowdsourcing data for deep active learning - an application: Learning intents in alexa," *CoRR*, vol. abs/1803.04223, 2018.
- [15] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *NeurIPS* (Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds.), pp. 2035–2043, Curran Associates, Inc., 2009.
- [16] B. Settles, "Active learning literature survey," tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.
- [17] A. Ghiassi, T. Younesian, Z. Zhao, R. Birke, V. Schiavoni, and L. Y. Chen, "Robust (deep) learning framework against dirty labels and beyond," in *TPS-ISA*, pp. 236–244, IEEE, 2019.
- [18] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 27–39, 2013.
- [19] T. Younesian, Z. Zhao, A. Ghiassi, R. Birke, and L. Y. Chen, "Qac-tor: On-line active learning for noisy labeled stream data," *CoRR*, vol. abs/2001.10399, 2020.
- [20] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *ICML*, pp. 839–846, 2000.
- [21] A. Holub, P. Perona, and M. C. Burl, "Entropy-based active learning for object recognition," in *CVPR Workshops*, pp. 1–8, IEEE, 2008.
- [22] M. Haufmann, F. A. Hamprecht, and M. Kandemir, "Deep active learning with adaptive acquisition," in *IJCAI*, pp. 2470–2476, 2019.
- [23] W. Fu, M. Wang, S. Hao, and X. Wu, "Scalable active learning by approximated error reduction," in *SIGKDD* (Y. Guo and F. Farooq, eds.), pp. 1396–1405, ACM, 2018.
- [24] J. Song, H. Wang, Y. Gao, and B. An, "Active learning with confidence-based answers for crowdsourcing labeling tasks," *Knowl. Based Syst.*, vol. 159, pp. 244–258, 2018.
- [25] J. Zhong, K. Tang, and Z. Zhou, "Active learning from crowds with unsure option," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015* (Q. Yang and M. J. Wooldridge, eds.), pp. 1061–1068, AAAI Press, 2015.

- [26] Y. Yan, R. Rosales, G. Fung, and J. G. Dy, "Active learning from crowds," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011* (L. Getoor and T. Scheffer, eds.), pp. 1161–1168, Omnipress, 2011.
- [27] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal, and J. Gama, "Machine learning for streaming data: state of the art, challenges, and opportunities," *SIGKDD Explor.*, vol. 21, no. 2, pp. 6–22, 2019.
- [28] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *CoRR*, vol. abs/2004.05785, 2020.
- [29] D. Sahoo, Q. Pham, J. Lu, and S. C. H. Hoi, "Online deep learning: Learning deep neural networks on the fly," in *IJCAI*, pp. 2660–2666, 2018.
- [30] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.
- [31] X. Zhu, X. Wu, and Y. Yang, "Effective classification of noisy data streams with attribute-oriented dynamic classifier selection," *Knowledge and Information Systems*, vol. 9, no. 3, pp. 339–363, 2006.
- [32] F. Chu, Y. Wang, and C. Zaniolo, "An adaptive learning approach for noisy data streams," in *ICDM*, pp. 351–354, IEEE, 2004.
- [33] B. Mozafari, P. Sarkar, M. J. Franklin, M. I. Jordan, and S. Madden, "Scaling up crowd-sourcing to very large datasets: A case for active learning," *Proc. VLDB Endow.*, vol. 8, no. 2, pp. 125–136, 2014.
- [34] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in *NIPS*, pp. 10456–10465, 2018.
- [35] A. Ghiassi, T. Younesian, R. Birke, and L. Y. Chen, "Trustnet: Learning from trusted data against (a)symmetric label noise," *CoRR*, vol. abs/2007.06324, 2020.
- [36] A. Ghiassi, R. Birke, R. Han, and L. Y. Chen, "Expertnet: Adversarial learning and recovery against noisy labels," 2020.
- [37] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," *ICLR Workshops*, 2015.
- [38] K. K. Thekumparampil, A. Khetan, Z. Lin, and S. Oh, "Robustness of conditional gans to noisy labels," in *NIPS*, pp. 10271–10282, 2018.
- [39] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *CVPR*, pp. 1944–1952, 2017.
- [40] B. Han, J. Yao, G. Niu, M. Zhou, I. W. Tsang, Y. Zhang, and M. Sugiyama, "Masking: A new perspective of noisy supervision," in *NIPS*, pp. 5841–5851, 2018.
- [41] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied statistics*, pp. 20–28, 1979.
- [42] H.-C. Kim and Z. Ghahramani, "Bayesian classifier combination," in *Artificial Intelligence and Statistics*, pp. 619–627, 2012.
- [43] D. Zhou, S. Basu, Y. Mao, and J. C. Platt, "Learning from the wisdom of crowds by minimax entropy," in *Advances in Neural Information Processing Systems*, pp. 2195–2203, 2012.
- [44] D. Zhou, Q. Liu, J. Platt, and C. Meek, "Aggregating ordinal labels from crowds by minimax conditional entropy," in *International Conference on Machine Learning*, pp. 262–270, 2014.
- [45] Y. Zhou and J. He, "Crowdsourcing via tensor augmentation and completion," in *IJCAI*, pp. 2435–2441, 2016.
- [46] P. J. Jongen, K. Wesnes, B. van Geel, P. Pop, E. Sanders, H. Schrijver, L. H. Visser, H. J. Gilhuis, L. G. Sinnige, A. M. Brands, et al., "Relationship between working hours and power of attention, memory, fatigue, depression and self-efficacy one year after diagnosis of clinically isolated syndrome and relapsing remitting multiple sclerosis," *PLoS one*, vol. 9, no. 5, p. e96444, 2014.
- [47] J. Pencavel, "The productivity of working hours," *The Economic Journal*, vol. 125, no. 589, pp. 2052–2076, 2015.
- [48] X. Li and Y. Guo, "Adaptive active learning for image classification," in *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pp. 859–866, IEEE Computer Society, 2013.
- [49] S. Yan, K. Chaudhuri, and T. Javidi, "Active learning from imperfect labelers," in *NeurIPS*, pp. 2128–2136, 2016.
- [50] V. S. Sheng, F. J. Provost, and P. G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *ACM SIGKDD*, pp. 614–622, 2008.
- [51] J. Zhang, X. Wu, and V. S. Sheng, "Active learning with imbalanced multiple noisy labeling," *IEEE Trans. Cybernetics*, vol. 45, no. 5, pp. 1081–1093, 2015.
- [52] Y. Zheng, S. D. Scott, and K. Deng, "Active learning from multiple noisy labelers with varied costs," in *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010* (G. I. Webb, B. Liu, C. Zhang, D. Gunopulos, and X. Wu, eds.), pp. 639–648, IEEE Computer Society, 2010.
- [53] P. Donmez and J. G. Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *CIKM* (J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K. Choi, and A. Chowdhury, eds.), pp. 619–628, ACM, 2008.
- [54] T. Younesian, D. Epema, and L. Y. Chen, "Active learning for noisy data streams using weak and strong labelers," 2020.
- [55] Y. Yan, R. Rosales, G. Fung, M. Schmidt, G. H. Valadez, L. Bogoni, L. Moy, and J. G. Dy, "Modeling annotator expertise: Learning when everybody knows a bit of something," in *AISTATS* (Y. W. Teh and D. M. Titterton, eds.), vol. 9 of *JMLR Proceedings*, pp. 932–939, 2010.
- [56] C. Zhang and K. Chaudhuri, "Active learning from weak and strong labelers," in *NeurIPS*, pp. 703–711, 2015.
- [57] L. Zhao, G. Sukthankar, and R. Sukthankar, "Incremental relabeling for active learning with noisy crowdsourced annotations," in *PASSAT*, pp. 728–733, IEEE Computer Society, 2011.
- [58]
- [59] Y. Liu and M. Liu, "An online learning approach to improving the quality of crowd-sourcing," in *Proceedings of the 2015 ACM SIGMETRICS* (B. Lin, J. J. Xu, S. Sengupta, and D. Shah, eds.), pp. 217–230, ACM, 2015.
- [60] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop, coursera: Neural networks for machine learning," *University of Toronto, Tech. Rep.*, 2012.
- [61] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *IEEE CVPR*, pp. 2372–2379, 2009.
- [62] D. Dua and C. Graff, "UCI machine learning repository," 2017.
- [63] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research),"
- [64] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.
- [65] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *IEEE ICCV*, pp. 322–330, 2019.
- [66] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S.-T. Xia, "Iterative learning with open-set noisy labels," in *IEEE CVPR*, pp. 8688–8696, 2018.