Multiclass Learning With Partially Corrupted Labels

Ruxin Wang, Tongliang Liu, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Traditional classification systems rely heavily on sufficient training data with accurate labels. However, the quality of the collected data depends on the labelers, among which inexperienced labelers may exist and produce unexpected labels that may degrade the performance of a learning system. In this paper, we investigate the multiclass classification problem where a certain amount of training examples are randomly labeled. Specifically, we show that this issue can be formulated as a label noise problem. To perform multiclass classification, we employ the widely used importance reweighting strategy to enable the learning on noisy data to more closely reflect the results on noise-free data. We illustrate the applicability of this strategy to any surrogate loss functions and to different classification settings. The proportion of randomly labeled examples is proved to be upper bounded and can be estimated under a mild condition. The convergence analysis ensures the consistency of the learned classifier to the optimal classifier with respect to clean data. Two instantiations of the proposed strategy are also introduced. Experiments on synthetic and real data verify that our approach yields improvements over the traditional classifiers as well as the robust classifiers. Moreover, we empirically demonstrate that the proposed strategy is effective even on asymmetrically noisy data.**

***Index Terms*— Importance reweighting, label noise, multiclass classification, random labels.**

I. Introduction

BIG data form a new frontier for innovation and productivity. The recent explosion in the availability of big data through social networks brings challenges but also opportunities for artificial intelligence and machine learning, including visualization, classification, and prediction. The ability to comprehend and extract actionable information and knowledge from data is thus essential for building reliable intelligent systems [1]–[3].

The performance of a system largely depends on the quality of the training data, which need to be manually collected. Undoubtedly, it is highly expected that one will correct the inaccurate information during collection, such that the quality of the obtained data can be ensured. However, such a training data collection mission is expensive and time-consuming, which often involves unaffordable expert effort. Considering that data are shared and processed by many groups, more and more labeling about the data offered by nonexperts becomes available. For example, the crowdsourcing mechanism [4], [5] (e.g., Amazon Mechanical Turk) offers an open platform for requesters to collect labels at low cost. It facilitates requesters to post tasks and enables online users (mostly nonexperts) to accomplish the labeling tasks for payment. Through such a platform, it is possible to either ask all labelers to do the task on the same data set, or ask different labelers to work on different subsets of the whole data without overlap. Here, we consider the latter case. Among the nonexperts, there may exist a certain number of labelers who just want to maximize personal profit without conscientiousness, and the rest are labelers who can provide accurate labels. In such a case, it is likely that parts of (not all) data examples are randomly labeled by the first kind of labelers. Since there is no oracle who can verify the obtained labels during the labeling process, the correctness of the produced labels cannot be guaranteed. This raises an inevitable issue that an inaccurate labeling process introduces noisy labels. More specifically, such noise can be seen as label flip noise, where an instance is erroneously given the label of another class within the data set. Considering that the distribution of the examples in each class can be well characterized, the flipped labels may confuse the learning of classifiers, thus rendering most low-capacity (e.g., linear) classification approaches suboptimal, or even poor in some situations [6]. This motivates the necessity of label noise-tolerant classification algorithms.

Explicitly, the aforementioned noise refers to observed labels that are incorrect. During label collection, no supervisor can provide an indication of correctness or a measure of uncertainty on the observed labels. We should clarify that only the observed labels of the examples are affected, while the clean labels are unavailable, and then, this induces the label noise problem.

Until now, a wide range of attention has been focused on the label noise problem for binary classification [7]–[13], whereas limited efforts have been conducted on multiclass classification with noisy labels. For example, a robust multiclass Gaussian process classifier was proposed [14], which suppresses the label noise effects by ignoring the distances of the incorrectly labeled examples to the decision boundaries of the classifier. This is achieved by introducing, in the Gaussian process model, a binary indictor of whether the label is correct or not. Bootkrajang and Kabán [15] developed a model-specified approach that extended the multiclass quadratic normal discriminant analysis to a robust variant, and empirically demonstrated its efficacy. A large-scale multiclass classifi-

Manuscript received May 6, 2016; revised November 3, 2016 and February 10, 2017; accepted April 18, 2017. Date of publication May 16, 2017; date of current version May 15, 2018. This work was supported by the Australian Research Council under Projects FT-130101457, DP-140102164, and LP-150100671. *(Corresponding author: Tongliang Liu.)*

R. Wang is with the Yunshangyun Artificial Intelligence Institute, Kunming 650500, China, and also with the Yunnan Union Visual Innovation Technology, Kunming 650500, China (e-mail: rosinwang@gmail.com).

T. Liu and D. Tao are with the UBTech Sydney Artificial Intelligence Institute and School of Information Technologies, the Faculty of Engineering and Information Technologies, The University of Sydney, Darlington, NSW 2008, Australia (e-mail: tongliang.liu@sydney.edu.au; dacheng.tao@sydney.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2699783

cation problem on the ImageNet benchmark has also been investigated in the case of label noise [16]. A noisy layer, which is instantiated by a label probability transition matrix, is built upon Convnet. The label noise-tolerant property is obtained via a thorough learning process. Sáez *et al.* [17] suggested to decompose the multiclass classification into one-vs-one subproblems to alleviate the influence of noisy labels. Sun *et al.* [18] designed a robust multiclass AdaBoost (Rob_MAd) algorithm, which explicitly detected the noisy labels and trained the classifiers based on a noise-aware loss function. In the loss function, an indicator function of the example being noisy or not was used to penalize the misclassified nonnoisy examples and the correctly classified noisy examples.

The above-mentioned methods have proven promising in handling different types of label noise in multiclass settings. However, a critical issue restricts them from benefitting from the previous research efforts on multiclass classification, which is, they are designed for specific surrogate loss functions or classifiers, and cannot be directly applied to other effective learning machines and classifiers. Hence, a general label noise-tolerant strategy is needed, which is capable of cooperating with any traditional multiclass classification methods, or specifically, with any surrogate loss functions and any multiclass classification settings [19].

In this paper, we consider the setting that when collecting the labels for multiclass classification, a proportion of data is randomly labeled. We show that this issue can be formulated as a label noise problem under the assumption of symmetric noise, which means that the labels have some constant probability of being flipped [7]. To tackle the problem of multiclass classification with noisy labels, we investigate the importance reweighting method [10]. Note that this investigation for multiclass classification is not trivial, since two major problems exist: one is how to handle the noisy data by estimating the influence of each example on the learning of classifiers, while another is how to estimate the noise rate. In the multiclass case, the label noise disrupts the distribution of the original clean data, thus changing the structure of the data space. Inspired by the idea of importance reweighting, the example that is incorrectly labeled is expected to have a weak influence on the learning of the classifiers, where the influence is revealed by a weight estimated from the distribution of the training data. We illustrate that in the multiclass case, this strategy is applicable to different surrogate loss functions and different classification settings. The proportion of randomly labeled examples is proved to be upper bounded, and can be estimated according to the distribution of the noisy data. We provide theoretical results to show that the classifier learned on noisy data by using our method is consistent with the optimal classifier learned on noise-free data, under certain conditions. Note that the classifier is regarded to be optimal only if it is the minimizer of the expected risk defined by a surrogate loss function. To verify the above-claimed advantages, we then propose an importance reweighted multinomial logistic regression (IWMLR) model and an importance reweighted multiclass support vector machine (IWSVM). Experimental results on synthetic and real data sets demonstrate the effectiveness of the proposed strategy. The trials on the data generated under an asymmetric noise model also verify that the proposed strategy can improve the traditional classification methods even when the symmetric assumption is not fulfilled.

The remainder of this paper is organized as follows. Section II reviews related work on label noise. Section III elaborates the formulation of multiclass classification when parts of training data are randomly labeled. The problem is modeled as a label noise issue. In Section IV, we introduce how the proposed importance reweighting strategy reduces the uncertainty caused by noisy labels, and discuss how to estimate the weights and the proportion of randomly labeled data; the convergence analysis is provided; two instantiations of the proposed strategy are detailed; and the differences between the proposed method and the previous ones are also discussed in this section. Empirical analyses on both synthetic data and real data are presented in Section V, with the conclusion drawn in Section VI.

II. Related Work in the Literature

Extensive efforts have been conducted on the label noise problem for the case of binary classification, and the literature reflects the progress. Label noise affects learning requirements or the complexity of designed models. In the classical setting, the random classification noise (RCN) model where each label is flipped independently with some probability is widely investigated. It has been shown [7] that the presence of RCN in the probably approximately correct (PAC) framework [20] increases the number of necessary examples for PAC identification. Several works [8], [21] analyze the PAC-learnability of the RCN model by deriving the upper bounds for the necessary examples.

To address RCN, there exist several trends in designing reliable algorithms. Along one trend, robust surrogate loss functions [22]–[24] or robust classifiers [25]–[27] are developed. However, label noise is not explicitly considered in these algorithms. The second trend concerns filtering [28]–[31]. In such a case, the mislabeled examples are typically identified, and then are either relabeled or removed. The preprocessed data are fed into the classification methods. An insufficiency is that it is likely to remove a substantial quantity of data, reducing the reliability of the learned classifiers. The third trend pays attention to algorithms that model label noise directly by using limited examples, such as the statistical query model [32], Bayesian model [33], SVM learning [34], loss factorization [35], [36], stochastic programming [37], and the perceptron algorithms [38]–[40]. The advantage is to explicitly model the effects of the label noise model, allowing exploiting the nature of label noise. There is one more type of approach, which is called label distribution learning [41]. Geng and Xia [42] proposed the so-called multivariate label distribution learning to correct the inaccurate labels in the application of head pose estimation. A review [43] could be referred to for missing references. In these works, there is a restriction that the proposed algorithm can only be used as shown in their paper and is not general. It is difficult or even impossible to broadcast

the algorithm to different surrogate loss functions and classifiers designed for the traditional noise-free classification problem.

Even though Aslam and Decatur [8] proved that the RCN model for the "0-1" loss is PAC-learnable under the condition of finite Vapnik-Chervonenkis-dimension, and van Rooyen *et al.* [13] summarized that the classification-calibrated loss functions are asymptotically robust to RCN, Manwani and Sastry [23] indicated that linear classifiers obtained under many other surrogate loss functions (e.g., hinge loss, exponential loss, and logistic loss) are not tolerant of label noise. Targeting a general learning framework in the presence of label noise, Natarajan *et al.* [9] investigated the problem of risk minimization under asymmetric RCN, where the noise rate is class-conditional. Two methods workable for most surrogate loss functions were proposed. Liu and Tao [10] addressed the class-conditional label noise problem by proposing a more general framework, which employs the importance reweighting strategy. Their theoretical results suggested that the classifiers learned by using this strategy would converge to the optimal classifiers for noise-free examples, and that the noise rate in data can be estimated with a proven upper bound. This method closely relates to the proposed method, while the differences are discussed in Section IV-E.

III. Formulation of Multiclass Classification With Noisy Labels

The goal of multiclass classification [44]–[48] is to leverage a set of n training examples to design a classifier that is capable of distinguishing m classes based on an input vector of dimension d. We follow the common technique of representing the class labels as a "1-of-m" indicator vector $\boldsymbol{Y} = [\boldsymbol{Y}^1, \boldsymbol{Y}^2, \ldots, \boldsymbol{Y}^m]^\top$, such that $\sum_{k=1}^m \boldsymbol{Y}^k = 1$, $\boldsymbol{Y}^k = 1$ if the corresponding example $\boldsymbol{X} \in \mathbb{R}^d$ belongs to the kth class and $\boldsymbol{Y}^k = 0$ otherwise. The n training examples are represented as $\{(\boldsymbol{X}_1, \boldsymbol{Y}_1), \ldots, (\boldsymbol{X}_n, \boldsymbol{Y}_n)\}$, which are independent and identically distributed examples drawn from an underlying (noise-free) distribution D. Here, we consider to transform the multiclass classification problem into binary classification subproblems in two different ways, including *one-vs-rest* and *one-vs-one*. The *one-vs-rest* setting states that we will learn m classifiers $\{f_k\}_{k=1}^m$ where the classifier f_k separates the kth class against all the other classes. Denoting $f = \{f_k\}_{k=1}^m$ and predefining a surrogate loss function ℓ, we need to optimize the following expected risk:

$$R_{\ell,D}(f) = \sum_{k=1}^{m} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim D}[\ell(f_k(\boldsymbol{X}), \boldsymbol{Y})] \tag{1}$$

where $\mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim D}$ denotes the expectation operator where the variables $(\boldsymbol{X}, \boldsymbol{Y})$ are drawn from the distribution D. Considering that D is unknown, generally, the empirical risk is utilized to approximate (1), that is

$$\hat{R}_{\ell,D}(f) = \sum_{k=1}^{m} \left(\frac{1}{n} \sum_{i=1}^{n} \ell(f_k(\boldsymbol{X}_i), \boldsymbol{Y}_i) \right). \tag{2}$$

The *one-vs-one* setting involves multiple two-class classifiers with each classifying one category against another. The corresponding expected risk can be written as

$$R_{\ell,D}(\{f_{tk}\}) = \sum_{t=1}^{m-1} \sum_{k=t+1}^{m} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim D^{tk}}[\ell(f_{tk}(\boldsymbol{X}), \boldsymbol{Y})] \tag{3}$$

where f_{tk} is the classifier separating the tth and kth classes, and D^{tk} is the distribution generating the examples of the tth and kth classes.

Considering the setting that the labels are partially corrupted, we denote the obtained labels as $\hat{\boldsymbol{Y}}$ and the true labels as $\boldsymbol{Y}$. Given that a proportion $\gamma < 1$ of the examples are randomly labeled, we assume that the examples in each class have the same probability of γ to be randomly processed. Without loss of generality, we denote the probability of the examples belonging to the kth class as $P(\boldsymbol{Y}^k = 1) = \rho_k$. Then, the correctly labeled examples in the kth class have a probability of

$$P(\hat{\boldsymbol{Y}}^k = 1, \boldsymbol{Y}^k = 1) = \rho_k(1 - \gamma) + \rho_k \frac{\gamma}{m}.$$

Complementarily, the examples that do not belong to the kth class but are labeled as this class have the ratio of

$$P(\hat{\boldsymbol{Y}}^k = 1, \boldsymbol{Y}^k \neq 1) = (1 - \rho_k)\frac{\gamma}{m}.$$

Hence, we can obtain the following conditional probabilities via simple derivations:

$$P(\hat{\boldsymbol{Y}}^k = 1 | \boldsymbol{Y}^k = 1) = 1 - \gamma + \frac{\gamma}{m} \tag{4}$$

$$P(\hat{\boldsymbol{Y}}^k \neq 1 | \boldsymbol{Y}^k = 1) = \gamma - \frac{\gamma}{m} \tag{5}$$

$$P(\hat{\boldsymbol{Y}}^k = 1 | \boldsymbol{Y}^k \neq 1) = \frac{\gamma}{m} \tag{6}$$

for $k \in \{1, \ldots, m\}$.

From (5) and (6), we know that for the kth class, the labels are randomly flipped to other classes with the probability of $\gamma - (\gamma/m)$ (the *out-flip probability*); the labels of other classes are flipped to the kth with the probability of (γ/m) (the *in-flip probability*). This can be regarded as a label noise problem in the multiclass case; more specifically, it closely relates to the RCN model in the classic binary classification setting, where each label is flipped independently with some probability $\rho \in [0, 0.5)$. More discussions about the relationship between the proposed setting and the binary setting can be found in Section IV-E. We assume that either the out-flip probability or the in-flip probability is the same for all k, indicating that we impose a symmetric RCN model on the multiclass setting. How to estimate γ is a critical issue that will be addressed in this paper. However, an asymmetric RCN model may attract more attention in practice. In this regard, we investigate in experiments the tolerance of the proposed method to the violation of the assumption, i.e., the noise is generated according to an asymmetric RCN model. On the other hand, in the asymmetric setting, the noise rate in each class is possible to be estimated, which could be a further extension of this paper; see the binary case [10].

Let D_γ be the distribution of the collected noisy examples $\{(\boldsymbol{X}_1, \hat{\boldsymbol{Y}}_1), \ldots, (\boldsymbol{X}_n, \hat{\boldsymbol{Y}}_n)\}$. In the above-mentioned setting,

the classifiers need to be learned by exploiting the knowledge from D_γ. Many traditional classification methods crucially rely on correct labels. Inaccurate label information may increase the uncertainty of classifiers during learning, inevitably degrading the classification performances. Therefore, the challenging problems are: 1) how to estimate the proportion of the randomly labeled data (or the noise rate), which, if known, benefits the statistical learning of the classifiers; 2) how to guarantee that the learned classifiers on noisy data can converge to the optimal classifiers with respect to clean data; and 3) how to design a strategy that is applicable to any surrogate loss functions and algorithms. With this regard, we reexplore the importance reweighting strategy for multiclass classification in Section IV.

IV. Multiclass Classification With Importance Reweighting

Domain adaptation [49]–[51], where importance reweighting is widely used, is to exploit the knowledge of a source domain to improve the model performance in a target domain. From this perspective, the true distribution D (regarded as the target domain) is transformed to the noisy distribution D_γ (regarded as the source domain) due to the noisy labels. To make a clear presentation, we use $P_D(\boldsymbol{X}, \boldsymbol{Y})$ [or $P_{D_\gamma}(\boldsymbol{X}, \boldsymbol{Y})$] to denote the joint probability of the variables $(\boldsymbol{X}, \boldsymbol{Y})$ under the distribution D (or D_γ), similar to $P_D(\boldsymbol{Y}|\boldsymbol{X})$ and $P_{D_\gamma}(\boldsymbol{Y}|\boldsymbol{X})$. Without the subscript D or D_γ, we use $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$ to indicate the clean and noisy labels, respectively, as in (4)–(6).

Given the condition that the input vectors $\boldsymbol{X}$ are noise-free [i.e., $P_D(\boldsymbol{X}) = P_{D_\gamma}(\boldsymbol{X})$], we have[1]

$$P_D(\boldsymbol{Y}|\boldsymbol{X}) = \frac{P_D(\boldsymbol{X}, \boldsymbol{Y})}{P_{D_\gamma}(\boldsymbol{X}, \boldsymbol{Y})} P_{D_\gamma}(\boldsymbol{Y}|\boldsymbol{X}). \tag{7}$$

Replacing the conditional probabilities in the above-mentioned equation with the surrogate loss function, and considering (1), we can deduce the expected risk under D_γ as

$$\begin{aligned} R_{\ell,D}(f) &= \sum_{k=1}^{m} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim D}[\ell(f_k(\boldsymbol{X}), \boldsymbol{Y})] \\ &= \sum_{k=1}^{m} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim D_\gamma}\left[\frac{P_D(\boldsymbol{X}, \boldsymbol{Y})}{P_{D_\gamma}(\boldsymbol{X}, \boldsymbol{Y})}\ell(f_k(\boldsymbol{X}), \boldsymbol{Y})\right]. \end{aligned}$$

We assume that the ratio $(P_D(\boldsymbol{X}, \boldsymbol{Y})/P_{D_\gamma}(\boldsymbol{X}, \boldsymbol{Y}))$ determines the contribution of the loss of each noisy example to $R_{\ell,D}(f)$. In this sense, we define $\beta(\boldsymbol{X}, \boldsymbol{Y}) = (P_D(\boldsymbol{X}, \boldsymbol{Y})/P_{D_\gamma}(\boldsymbol{X}, \boldsymbol{Y}))$, which can be used to suppress the influence of the incorrect labels on $R_{\ell,D}(f)$. Accordingly, we obtain a reweighted expected risk

$$\begin{aligned} R_{\ell,D}(f) &= \sum_{k=1}^{m} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim D_\gamma}\left[\beta(\boldsymbol{X}, \boldsymbol{Y})\ell(f_k(\boldsymbol{X}), \boldsymbol{Y})\right] \\ &= R_{\beta\ell,D_\gamma}(f). \end{aligned} \tag{8}$$

[1]According to the definitions in Section III, D is a distribution generating the examples $\{(\boldsymbol{X}_i, \boldsymbol{Y}_i)\}$, rather than the examples $\{(\boldsymbol{X}_i, \boldsymbol{Y}_i)\}$ form the distribution D. This is important for understanding that $P_D(\boldsymbol{Y}|\boldsymbol{X}) \neq 1$. The same condition is applied in $P_{D_\gamma}(\boldsymbol{Y}|\boldsymbol{X})$.

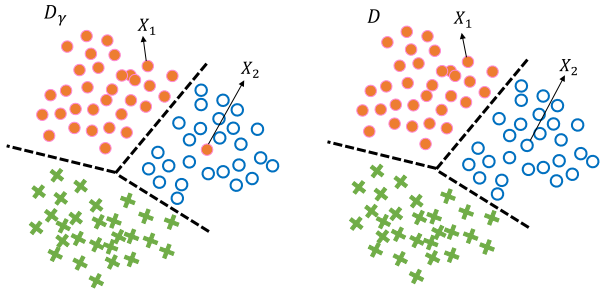


Fig. 1. Illustration of noisy labels. Left: two examples including a correctly labeled example $\boldsymbol{X}_1$ and an incorrectly labeled example $\boldsymbol{X}_2$. Right: underlying noiseless examples. In both D_γ and D, $\boldsymbol{X}_1$ is labeled as dot, saying that $P_{D_\gamma}(\boldsymbol{Y}_1 = \text{dot}|\boldsymbol{X}_1)$ and $P_D(\boldsymbol{Y}_1 = \text{dot}|\boldsymbol{X}_1)$ reach to high values (ideally one). On the other hand, in D_γ, $\boldsymbol{X}_2$ is labeled as dot, whereas it is labeled as *circle* in D. This indicates that $P_{D_\gamma}(\boldsymbol{Y}_2 = \text{dot}|\boldsymbol{X}_2)$ and $P_D(\boldsymbol{Y}_2 = \textit{circle}|\boldsymbol{X}_2)$ reach to high values, but importantly, $P_D(\boldsymbol{Y}_2 = \text{dot}|\boldsymbol{X}_2)$ has a low value (even zero). According to the definition of β, we thus have that $\beta(\boldsymbol{X}_1, \boldsymbol{Y}_1 = \text{dot})$ is large, while $\beta(\boldsymbol{X}_2, \boldsymbol{Y}_2 = \text{dot})$ is small.

This is similar to the idea from the field of importance sampling [52] [except for the condition that $P_D(\boldsymbol{X}) = P_{D_\gamma}(\boldsymbol{X})$]. The same operation can also be applied to (3) to reach the corresponding reweighted risk for the *one-vs-one* setting, that is

$$R_{\ell,D}(\{f_{tk}\}) = R_{\beta\ell,D_\gamma}(\{f_{tk}\}). \tag{9}$$

Given the true distribution D (although it is unknown in practice) and the noisy distribution D_γ, $\beta(\boldsymbol{X}, \boldsymbol{Y})$ varies according to the example position in the data space and its label. It is expected that a correct-labeled example has a large $\beta(\boldsymbol{X}, \boldsymbol{Y})$ and contributes more to the risk, while an incorrect-labeled example has a small $\beta(\boldsymbol{X}, \boldsymbol{Y})$ and contributes less. Fig. 1 shows this scenario clearly.

In conclusion, the above-mentioned reweighting strategy reveals the following three remarks.

1) $\beta(\boldsymbol{X}, \boldsymbol{Y})$ will likely assign a low ratio to the examples that are incorrectly labeled, since $P_D(\boldsymbol{X}, \boldsymbol{Y})$ induces small values for those examples, while $P_{D_\gamma}(\boldsymbol{X}, \boldsymbol{Y})$ has normal values.
2) $\beta(\boldsymbol{X}, \boldsymbol{Y})$ is independent of the surrogate loss function ℓ and the multiclass classification setting (i.e., *one-vs-rest* or *one-vs-one*).
3) According to (7)

$$\beta(\boldsymbol{X}, \boldsymbol{Y}) = \frac{P_D(\boldsymbol{X}, \boldsymbol{Y})}{P_{D_\gamma}(\boldsymbol{X}, \boldsymbol{Y})} = \frac{P_D(\boldsymbol{Y}|\boldsymbol{X})}{P_{D_\gamma}(\boldsymbol{Y}|\boldsymbol{X})}$$

which is useful in the estimation of β and will be discussed in the following.

In (8), the pair $(\boldsymbol{X}, \boldsymbol{Y})$ follows the distribution D_γ, meaning that the weight is estimated based on noisy example $(\boldsymbol{X}, \hat{\boldsymbol{Y}})$. Thus, we can equivalently write the weight as $\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}})$.

A. Estimation of $\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}})$

Since the true distribution D is unknown, the estimation of $\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}})$ becomes difficult and can only rely on the collected noisy data. We note that for each class label $k \in \{1, \ldots, m\}$, $\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^k)$ can be estimated independently. Regarding this, we have the following lemma.

Lemma 1: For multiclass classification, when a proportion γ of data are randomly labeled, we can estimate the weights through the noisy data

$$\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^k) = \frac{P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X}) - \hat{\boldsymbol{Y}}^k \cdot \frac{\gamma}{m} - (1 - \hat{\boldsymbol{Y}}^k) \cdot \gamma \left(1 - \frac{1}{m}\right)}{(1-\gamma) P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X})}. \quad (10)$$

$\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^k)$ is nonnegative. If $P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X}) = 0$, we set $\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^k) = 0$.

Proof: Starting with the $\hat{\boldsymbol{Y}}^k = 1$ case, we have

$$\begin{aligned} P(\hat{\boldsymbol{Y}}^k = 1|\boldsymbol{X}) &= \sum_{l=1}^{m} P(\hat{\boldsymbol{Y}}^k = 1, \boldsymbol{Y}^l = 1|\boldsymbol{X}) \\ &= \sum_{l=1}^{m} P(\hat{\boldsymbol{Y}}^k = 1|\boldsymbol{Y}^l = 1, \boldsymbol{X}) P(\boldsymbol{Y}^l = 1|\boldsymbol{X}) \\ &= \sum_{l=1}^{m} P(\hat{\boldsymbol{Y}}^k = 1|\boldsymbol{Y}^l = 1) P(\boldsymbol{Y}^l = 1|\boldsymbol{X}). \end{aligned}$$

According to (4) and (6) and $\sum_{k=1}^{m} \boldsymbol{Y}^k = 1$

$$\begin{aligned} P(\hat{\boldsymbol{Y}}^k = 1|\boldsymbol{X}) &= \left((1-\gamma) + \frac{\gamma}{m}\right) P(\boldsymbol{Y}^k = 1|\boldsymbol{X}) \\ &\quad + \sum_{l \neq k} \frac{\gamma}{m} P(\boldsymbol{Y}^l = 1|\boldsymbol{X}) \\ &= \left((1-\gamma) + \frac{\gamma}{m}\right) P(\boldsymbol{Y}^k = 1|\boldsymbol{X}) \\ &\quad + \frac{\gamma}{m}(1 - P(\boldsymbol{Y}^k = 1|\boldsymbol{X})) \\ &= (1-\gamma) P(\boldsymbol{Y}^k = 1|\boldsymbol{X}) + \frac{\gamma}{m}. \end{aligned} \quad (11)$$

In the $\hat{\boldsymbol{Y}}^k = 0$ case, by using

$$\sum_{\boldsymbol{Y}^k = \{0,1\}} P(\boldsymbol{Y}^k|\boldsymbol{X}) = \sum_{\hat{\boldsymbol{Y}}^k = \{0,1\}} P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X}) = 1 \quad (12)$$

we have

$$P(\hat{\boldsymbol{Y}}^k = 0|\boldsymbol{X}) = (1-\gamma) P(\boldsymbol{Y}^k = 0|\boldsymbol{X}) + \gamma \left(1 - \frac{1}{m}\right). \quad (13)$$

Combining (11) and (13), we obtain

$$\begin{aligned} P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X}) &= (1-\gamma) P(\boldsymbol{Y}^k|\boldsymbol{X}) + \hat{\boldsymbol{Y}}^k \cdot \frac{\gamma}{m} \\ &\quad + (1 - \hat{\boldsymbol{Y}}^k) \cdot \gamma \left(1 - \frac{1}{m}\right) \end{aligned} \quad (14)$$

where $\boldsymbol{Y}^k \in \{0, 1\}$. Thus

$$\begin{aligned} \beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^k) &= \frac{P(\boldsymbol{Y}^k|\boldsymbol{X})}{P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X})} \\ &= \frac{P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X}) - \hat{\boldsymbol{Y}}^k \cdot \frac{\gamma}{m} - (1 - \hat{\boldsymbol{Y}}^k) \cdot \gamma \left(1 - \frac{1}{m}\right)}{(1-\gamma) P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X})}. \end{aligned} \quad (15)$$

$\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^k)$ is nonnegative, because $P(\boldsymbol{Y}^k|\boldsymbol{X}) \geq 0$ and $P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X}) \geq 0$. Thus, the lemma is proved. □

From Lemma 1, we see that $\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^k)$ has different values for different k values, which means that each instance $(\boldsymbol{X}, \hat{\boldsymbol{Y}})$ is assigned multiple weights $\{\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^1), \ldots, \beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^m)\}$. The expected risk of the *one-vs-rest* case is accordingly modified as

$$R_{\beta\ell, D_\gamma}(f) = \sum_{k=1}^{m} \mathbb{E}_{D_\gamma}[\beta(\boldsymbol{X}, \boldsymbol{Y}^k) \ell(f_k(\boldsymbol{X}), \boldsymbol{Y}^k)] \quad (16)$$

and that of the *one-vs-one* case is similar

$$R_{\beta\ell, D_\gamma}(\{f_{tk}\}) = \sum_{t=1}^{m-1} \sum_{k=t+1}^{m} \mathbb{E}_{D_\gamma^{tk}}[\beta(\boldsymbol{X}, \boldsymbol{Y}^{tk}) \ell(f_{tk}(\boldsymbol{X}), \boldsymbol{Y}^{tk})] \quad (17)$$

where the reweighting strategy is reduced to a binary case. $\boldsymbol{Y}^{tk} = 1$ if $\boldsymbol{X}$ belongs to the tth class, while $\boldsymbol{Y}^{tk} = 0$ if $\boldsymbol{X}$ belongs to the kth class. $\beta(\boldsymbol{X}, \boldsymbol{Y}^{tk})$ denotes the estimation of β for the tth and kth classes under the distribution D_γ^{tk}.

From (15), we know that the computation of $\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^k)$ relies on two quantities, which are $P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X})$ and γ. These two quantities need to be estimated from noisy examples.

To estimate $P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X})$, there exist three types of approaches, including the probabilistic classification approach, the kernel density estimation approach, and the density ratio estimation approach. The probabilistic classification method highly depends on the selected classification model. When the model is misspecified, this method may introduce a large approximation error for learning the conditional distribution. The kernel density estimation method has a slow convergence rate when estimating $P(\boldsymbol{X}|\hat{\boldsymbol{Y}}^k)$ and $P(\boldsymbol{X})$ separately. Also, it needs a large amount of data examples for estimation. Considering these shortages, we choose the density ratio estimation approach [53], which is an effective way to alleviate the curse of dimensionality for the estimation of high-dimensional variables.

The density ratio estimation can be realized by three types of approaches [54]: probabilistic classification, moment matching, and ratio matching. As already proved [55], the moment matching approaches and the ratio matching approaches can produce lower approximation error and higher efficiency than the probabilistic classification methods. Therefore, we employ a ratio matching method, namely, KLIEP [56], to estimate $P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X})$. The reliability and the consistency assurance of this method have already been proved in [10].

B. Estimation of γ

The noise rate is a critical element and needs to be known in most algorithms designed for the RCN problem [57]–[59], as well as in our work. According to (5) and (6), the noise rate depends on the proportion γ, which is thus required. How to estimate γ is still a challenging problem, and there exist very limited approaches to accomplish this task. Next, we provide a theoretical result on the upper bound of γ, and illustrate that under mild assumptions, γ can be efficiently estimated through the proven upper bound.

Theorem 1: Let $A_{\boldsymbol{X}} = m \cdot \min_k P_{D_\gamma}(\boldsymbol{Y}^k = 1|\boldsymbol{X})$ and $B_{\boldsymbol{X}} = (m/m - 1) \cdot \min_k P_{D_\gamma}(\boldsymbol{Y}^k = 0|\boldsymbol{X})$, we have

$$\gamma \leq \min(A_{\boldsymbol{X}}, B_{\boldsymbol{X}}). \quad (18)$$

Furthermore, γ can be estimated via

$$\gamma = \min\left(\min_{\boldsymbol{X}\in\mathcal{X}} A_{\boldsymbol{X}}, \min_{\boldsymbol{X}\in\mathcal{X}} B_{\boldsymbol{X}}\right) \tag{19}$$

where $\mathcal{X}$ is the support of $\boldsymbol{X}$.

Proof: According to (11), if there exists $\boldsymbol{X}^{\backslash k} \in \mathcal{X}$, such that $P_D(\boldsymbol{Y}^k = 1|\boldsymbol{X}^{\backslash k}) = 0$, where $\boldsymbol{X}^{\backslash k}$ denotes the example that does not belong to the kth class under the distribution D, we have

$$P_{D_\gamma}(\boldsymbol{Y}^k = 1|\boldsymbol{X}^{\backslash k}) = \frac{\gamma}{m}. \tag{20}$$

Similarly, if there exists $\boldsymbol{X}^k \in \mathcal{X}$, such that $P_D(\boldsymbol{Y}^k = 0|\boldsymbol{X}^k) = 0$, where $\boldsymbol{X}^k$ is the example of the kth class under D, (13) means that

$$P_{D_\gamma}(\boldsymbol{Y}^k = 0|\boldsymbol{X}^k) = \gamma\left(1 - \frac{1}{m}\right). \tag{21}$$

Considering $P_D(\boldsymbol{Y}^k = 1|\boldsymbol{X}) \geq 0$ and $P_D(\boldsymbol{Y}^k = 0|\boldsymbol{X}) \geq 0$ for all $k \in \{1, \ldots, m\}$, we have

$$P_{D_\gamma}(\boldsymbol{Y}^k = 1|\boldsymbol{X}) \geq \frac{\gamma}{m} \tag{22}$$

and

$$P_{D_\gamma}(\boldsymbol{Y}^k = 0|\boldsymbol{X}) \geq \gamma\left(1 - \frac{1}{m}\right). \tag{23}$$

By combining (22) and (23), both $\gamma \leq A_{\boldsymbol{X}}$ and $\gamma \leq B_{\boldsymbol{X}}$ hold. Thus, the inequality in (18) is proved.

For multiclass classification, it is possible to find an example $\boldsymbol{X} \in \mathcal{X}$, which is far from the classification hyperplane, such that $P_D(\boldsymbol{Y}^k = 1|\boldsymbol{X})$ [or $P_D(\boldsymbol{Y}^k = 0|\boldsymbol{X})$] tends to zero. In this case, the above-mentioned assumption is satisfied, and the equality in (18) holds. Hence, the estimation of γ can be directly deduced as in (19).

The theorem is proved. □

Theorem 1 provides a consistent estimator of γ under the condition that there exists $\boldsymbol{X}^{\backslash k}$ or $\boldsymbol{X}^k \in \mathcal{X}$, such that $P_D(\boldsymbol{Y}^k = 1|\boldsymbol{X}^{\backslash k}) = 0$ or $P_D(\boldsymbol{Y}^k = 0|\boldsymbol{X}^k) = 0$. According to (19), the convergence rate of estimating γ is the same as that of estimating the conditional distribution $P_{D_\gamma}(\boldsymbol{Y}^k|\boldsymbol{X})$. The estimation is thus efficient, and its effectiveness is verified in experiments.

C. Convergence Analysis

Here, we provide theoretical analyses on the convergence of the proposed importance reweighting strategy in multiclass classification. We show that with a sufficiently large amount of noisy data, the classifier learned using the proposed strategy can converge to the classifier, which is optimal for the clean data.

Theorem 2: Assume that there exists $\boldsymbol{X}^{\backslash k}$ or $\boldsymbol{X}^k \in \mathcal{X}$, such that $P_D(\boldsymbol{Y}^k = 1|\boldsymbol{X}^{\backslash k}) = 0$ or $P_D(\boldsymbol{Y}^k = 0|\boldsymbol{X}^k) = 0$. When the loss function ℓ is convex and the number of training examples $n \to \infty$, we have

$$f_n = \arg\min_{f\in F} \hat{R}_{\hat{\beta}\ell,D_\gamma} \to f^* = \arg\min_{f\in F} R_{\ell,D} \tag{24}$$

where f_n is the minimizer of the empirical risk $\hat{R}_{\hat{\beta}\ell,D_\gamma}$ with n examples, $\hat{\beta}$ is the weight by empirically learning the conditional density function $P_{D_\gamma}(\boldsymbol{Y}^k|\boldsymbol{X})$ and the proportion γ, $\hat{R}$ denotes the empirical risk, and F is a predefined hypothesis class.

Proof: When the sample size goes to infinity, the conditional distribution $P_{D_\gamma}(\boldsymbol{Y}^k|\boldsymbol{X})$ can be unbiasedly estimated by empolying some nonparametric methods, such as the Parzen window approach. This means that under the assumption in Theorem 2, the weight $\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}}^k)$ can also be unbiasedly estimated. We have that

$$\begin{aligned}
&R_{\hat{\beta}\ell,D_\gamma}(f_n) - R_{\hat{\beta}\ell,D_\gamma}(f^*) \\
&= R_{\hat{\beta}\ell,D_\gamma}(f_n) - \hat{R}_{\hat{\beta}\ell,D_\gamma}(f_n) + \hat{R}_{\hat{\beta}\ell,D_\gamma}(f_n) - \hat{R}_{\hat{\beta}\ell,D_\gamma}(f^*) \\
&\quad + \hat{R}_{\hat{\beta}\ell,D_\gamma}(f^*) - R_{\hat{\beta}\ell,D_\gamma}(f^*) \\
&\leq R_{\hat{\beta}\ell,D_\gamma}(f_n) - \hat{R}_{\hat{\beta}\ell,D_\gamma}(f_n) + \hat{R}_{\hat{\beta}\ell,D_\gamma}(f^*) - R_{\hat{\beta}\ell,D_\gamma}(f^*) \\
&\leq 2\sup_{f\in F} |R_{\hat{\beta}\ell,D_\gamma}(f) - \hat{R}_{\hat{\beta}\ell,D_\gamma}(f)|.
\end{aligned} \tag{25}$$

The first inequality holds, because f_n is the minimizer of $\hat{R}_{\hat{\beta}\ell,D_\gamma}$ and thus $\hat{R}_{\hat{\beta}\ell,D_\gamma}(f_n) - \hat{R}_{\hat{\beta}\ell,D_\gamma}(f^*) \leq 0$. Using the law of large numbers, when there are a sufficiently large number of examples, we have that $R_{\hat{\beta}\ell,D_\gamma}(f) - \hat{R}_{\hat{\beta}\ell,D_\gamma}(f)$ goes to zero and that $\hat{\beta}$ goes to β. Hence, $R_{\hat{\beta}\ell,D_\gamma}(f_n)$ goes to $R_{\hat{\beta}\ell,D_\gamma}(f^*)$ and $R_{\hat{\beta}\ell,D_\gamma}(f)$ goes to $R_{\beta\ell,D_\gamma}(f) = R_{\ell,D}(f)$ because of (8), which means that f_n approaches f^*. □

D. Two Instantiations

According to (8) and (9), we can state that any surrogate loss functions designed for either the *one-vs-rest* or the *one-vs-one* setting of multiclass classification can be applied in the presence of randomly labeled data by employing the proposed importance reweighting strategy. Given that the conditional probability $P_{D_\gamma}(\boldsymbol{Y}^k|\boldsymbol{X})$ and the proportion γ are accurately estimated, as the number of training examples increases, the classifiers learned using the noisy data are toward the optimal classifiers for the clean data according to Theorem 2. For empirical verification, in this section, we give two instantiations of the importance reweighting strategy, including MLR, which belongs to the *one-vs-rest* setting, and SVM, which belongs to the *one-vs-one* setting.

1) Importance Reweighted Multinomial Logistic Regression: MLR [60] predicts the probability of each example belonging to each class. Specifically, the probability that $\boldsymbol{X}$ belongs to the kth class is

$$P(\boldsymbol{Y}^k = 1|\boldsymbol{X}, \boldsymbol{W}) = \frac{\exp\left(\boldsymbol{W}_k^\top \boldsymbol{X}\right)}{\sum_{l=1}^m \exp\left(\boldsymbol{W}_l^\top \boldsymbol{X}\right)} \tag{26}$$

where $\boldsymbol{W}$ is the parameter matrix for multiclass classification, $\boldsymbol{W}_k$ encodes the parameters for the kth class, and T denotes the transpose of a matrix. The associated empirical risk for optimization is written as the reweighted negative log-likelihood

$$\hat{R}(\boldsymbol{W}) = -\frac{1}{n}\sum_{i=1}^n \log P(\boldsymbol{Y}_i|\boldsymbol{X}_i, \boldsymbol{W}). \tag{27}$$

Considering that noisy data are observed, the empirical risk is then modified by applying the importance reweighting strategy

according to (8), and we obtain

$$\begin{aligned}\hat{R}_\beta(\boldsymbol{W}) &= -\frac{1}{n}\sum_{i=1}^{n}\beta(\boldsymbol{X}_i,\hat{\boldsymbol{Y}}_i)\log P(\hat{\boldsymbol{Y}}_i|\boldsymbol{X}_i,\boldsymbol{W}) \\ &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{m}\beta\big(\boldsymbol{X}_i,\hat{\boldsymbol{Y}}_i^k\big)\log P\big(\hat{\boldsymbol{Y}}_i^k|\boldsymbol{X}_i,\boldsymbol{W}_k\big) \\ &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{m}\beta\big(\boldsymbol{X}_i,\hat{\boldsymbol{Y}}_i^k\big) \\ &\quad\times\left[\hat{\boldsymbol{Y}}_i^k\boldsymbol{W}_k^\top\boldsymbol{X}_i-\log\sum_{l=1}^{m}\exp\big(\boldsymbol{W}_l^\top\boldsymbol{X}_i\big)\right]. \quad (28)\end{aligned}$$

To avoid overfitting, a Frobenius regularization on $\boldsymbol{W}$ can be imposed. The objective function becomes

$$\begin{aligned}\hat{R}_\beta(\boldsymbol{W}) &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{m}\beta\big(\boldsymbol{X}_i,\hat{\boldsymbol{Y}}_i^k\big) \\ &\quad\times\left[\hat{\boldsymbol{Y}}_i^k\boldsymbol{W}_k^\top\boldsymbol{X}_i-\log\sum_{l=1}^{m}\exp\big(\boldsymbol{W}_l^\top\boldsymbol{X}_i\big)\right]+\lambda\|\boldsymbol{W}\|_F^2 \quad (29)\end{aligned}$$

where λ is the regularization parameter. The above-mentioned function can be optimized by utilizing the Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm algorithm, as in the original MLR.

2) Importance Reweighted Support Vector Machine: Regarding SVM, we consider the *one-vs-one* setting [61]. There are $(m(m-1)/2)$ classifiers involved in this setting, with each one learned independently. Assume that the classifier between the tth and kth classes is

$$f(\boldsymbol{X};\boldsymbol{W}_{tk})=\boldsymbol{W}_{tk}^\top\boldsymbol{X}+b_{tk} \quad (30)$$

where $(\boldsymbol{W}_{tk},b_{tk})$ denotes the parameters to be optimized. Then, the associated objective function is

$$\begin{aligned}&\min_{\boldsymbol{W}_{tk},b_{tk},\xi_{tk}}\ \frac{1}{2}\|\boldsymbol{W}_{tk}\|^2+C\sum_{i=1}^{n}\xi_i^{tk} \\ &\text{s.t. }\boldsymbol{W}_{tk}^\top\boldsymbol{X}_i+b_{tk}\geq 1-\xi_i^{tk},\quad \text{if }\hat{\boldsymbol{Y}}_i^{tk}=1 \\ &\quad\boldsymbol{W}_{tk}^\top\boldsymbol{X}_i+b_{tk}\leq -1+\xi_i^{tk},\quad \text{if }\hat{\boldsymbol{Y}}_i^{tk}=0 \\ &\quad\xi_i^{tk}\geq 0,\quad i=1,\ldots,n \quad (31)\end{aligned}$$

where ξ_i^{tk} is the slack variable. In (31), ξ_i^{tk} controls the margin of the example $\boldsymbol{X}_i$ to the classification hyperplane. In other words, ξ_i^{tk} reflects the influence of $\boldsymbol{X}_i$ on the learning of the classifier. Hence, by reweighting ξ_i^{tk}, we can accomplish the goal of (8). The modified objective function is as follows:

$$\begin{aligned}&\min_{\boldsymbol{W}_{tk},b_{tk},\xi_{tk}}\ \frac{1}{2}\|\boldsymbol{W}_{tk}\|^2+C\sum_{i=1}^{n}\beta(\boldsymbol{X}_i,\hat{\boldsymbol{Y}}_i^{tk})\xi_i^{tk} \\ &\text{s.t. }\boldsymbol{W}_{tk}^\top\boldsymbol{X}_i+b_{tk}\geq 1-\xi_i^{tk},\quad \text{if }\hat{\boldsymbol{Y}}_i^{tk}=1 \\ &\quad\boldsymbol{W}_{tk}^\top\boldsymbol{X}_i+b_{tk}\leq -1+\xi_i^{tk},\quad \text{if }\hat{\boldsymbol{Y}}_i^{tk}=0 \\ &\quad\xi_i^{tk}\geq 0,\quad i=1,\ldots,n. \quad (32)\end{aligned}$$

The optimization of the above-mentioned problem is similar to that of the original SVM [see (31)], by using the conventional quadratic programming algorithm[2] or more effective algorithms [62], [63].

While (28) and (32) consider the linear case, using a nonlinear kernel is preferred to improve the classification performance on nonlinear data. Specifically, the polynomial kernel is used in this paper. However, due to the heavy computational costs of calculating the kernel matrix, we use linear classifiers in experiments, if unspecified.

E. Discussion

Both our method and [10] employ the importance reweighting strategy to handle the label noise problem, but in different classification settings. The achieved results have noticeable differences. The weight estimation is generalized from binary classes to multiple classes. Specifically, assuming the noise rate for class $\hat{\boldsymbol{Y}}\in\{+1,-1\}$ is $\rho_{\hat{\boldsymbol{Y}}}$ (i.e., ρ_{+1} for the positive class and ρ_{-1} for the negative class), the weight is estimated via

$$\beta(\boldsymbol{X},\hat{\boldsymbol{Y}})=\frac{P(\hat{\boldsymbol{Y}}|\boldsymbol{X})-\rho_{-\hat{\boldsymbol{Y}}}}{(1-\rho_{+1}-\rho_{-1})P(\hat{\boldsymbol{Y}}|\boldsymbol{X})} \quad (33)$$

where $-\hat{\boldsymbol{Y}}$ is the opposite label of $\hat{\boldsymbol{Y}}$. This assumes asymmetric RCN, i.e., ρ_{+1}, differs from ρ_{-1}. In this paper, we analyze the noise rate for each class according to (5) and (6), and obtain

$$\rho_0^k = P(\hat{\boldsymbol{Y}}^k\neq 1|\boldsymbol{Y}^k=1) \quad (34)$$

$$\rho_1^k = P(\hat{\boldsymbol{Y}}^k=1|\boldsymbol{Y}^k\neq 1). \quad (35)$$

According to Lemma 1, (15) can be rewritten as

$$\beta(\boldsymbol{X},\hat{\boldsymbol{Y}}^k)=\frac{P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X})-\hat{\boldsymbol{Y}}^k\cdot\rho_1^k-(1-\hat{\boldsymbol{Y}}^k)\cdot\rho_0^k}{\big(1-\rho_1^k-\rho_0^k\big)P(\hat{\boldsymbol{Y}}^k|\boldsymbol{X})}. \quad (36)$$

We notice that ρ_0^k differs from ρ_1^k in general, which reveals an asymmetric attribute for class k. Considering all classes, either ρ_0^k or ρ_1^k is different with respect to each class. Hence, all noise rates are required to be estimated, which is substantially different from binary classification. Considering the difficulty of such estimation, we simplify the classification setting by enforcing the in-flip probability ρ_1^k as well as the out-flip probability ρ_0^k to be the same for all classes. Then, the noise rates for different classes are determined by a quantity γ, which can be estimation according to Theorem 1. Theoretical analyses are provided to guarantee the convergence of the learned classifier by using our strategy toward the optimal classifier on noise-free data.

We also investigate the relationship between our method and the deep learning-based method [16], [64], which models label flip noise by employing a labeling transition matrix and estimates it using plenty of data. Specifically, in [16], the $m\times m$ labeling transition matrix is

$$Q=\begin{bmatrix}P(\hat{\boldsymbol{Y}}^1=1|\boldsymbol{Y}^1=1) & \ldots & P(\hat{\boldsymbol{Y}}^1=1|\boldsymbol{Y}^m=1)\\ \vdots & \ddots & \vdots\\ P(\hat{\boldsymbol{Y}}^m=1|\boldsymbol{Y}^1=1) & \ldots & P(\hat{\boldsymbol{Y}}^m=1|\boldsymbol{Y}^m=1)\end{bmatrix}. \quad (37)$$

[2]LIBSVM has published an application program interface for the weighted SVM: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#weights_for_data_instances.

Q is realized as a linear fully connected layer built upon the base deep model. As expected, the output of the base model predicts the true label of an example. By using the Q layer, the true label is transformed to a noisy label. In this way, the learning of the model can be supervised by the collected noisy data. To restrict the parameter space of Q, a regularizer $\mathrm{tr}(Q)$ is added to the objective, where $\mathrm{tr}(\cdot)$ is the trace of a matrix. In their method, the authors impose an asymmetric noise assumption on Q, i.e., the elements in Q are different with respect to each other. This introduces an extremely difficult problem, which is alleviated through a thorough learning on extensive data. However, the accuracy of the learned Q cannot be guaranteed, even though a good performance improvement is achieved in experiments. Differently, in our method, we assume a symmetric noise model, meaning that the diagonal elements of Q are the same, and the off-diagonal elements are identical. All parameters in Q are determined by the proportion γ, thus alleviating the difficulty of the problem. Applying the proposed importance reweighting strategy to handle asymmetric RCN in the multiclass case is possible, but beyond the scope of this paper. A possible solution is to derive an upper bound for each element of Q by considering the statistics of the examples in each class.

V. Experiments

In this section, we conduct experiments on both synthetic and real data. To verify the effectiveness of the proposed method, the traditional classifiers, including MLR and SVM, are selected as baselines, and the robust models, including robust Gaussian process (RGP) [14] and Rob_MAd [18], are employed as competitors for comparison. Following [18], we use the decision tree as the base learners of Rob_MAd. We denote the two instantiations in Section IV-D as the IWMLR and the IWSVM, respectively. In each experiment, we conduct 5×2-fold cross validation to compare the above-mentioned methods, that means, we repeat twofold cross validation for five times. The accuracy of the classification result is defined as the ratio of the correctly classified examples with respect to the true distribution D. Average accuracy in the 5×2-fold cross validation is calculated in each experiment setting. Note that during training, there are hyperparameters that should be set for the classifiers, for example, λ in MLR and C in SVM. For this, a held-out validation set with 20% training data is randomly collected from the training set, and is used for selecting the best hyperparameter from a finite set of possible values.

We synthesize noisy data from clean data by stochastically changing a set of the labels: γ percent of data examples are randomly selected from the training set by ignoring the true labels, and then, these examples are assigned to the labels that are generated from a uniform distribution on $\{1, \ldots, m\}$. This operation is also applied to the validation data, but not to the test data. In experiments, the considered values of γ include $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$.

A. Synthetic Data

We synthesized the 2-D linearly separable data set, as shown in Fig. 2(a), which has five classes: *cyan diamond* (top-left corner), *blue plus* (top-right), *purple circle* (right), *red cross* (bottom-right), and *azure triangle* (bottom-left). The classification hyperplanes pass through the coordinate origin. The training and test sets contain 7000 and 5000 examples, respectively. The percentages of the examples in those classes are 31%, 16%, 11%, 29%, and 13%, respectively. The noisy training data are synthesized by setting $\gamma = 0.3$, as in Fig. 2(b). In this experiment, MLR is utilized as the classifier. We visualize the classification results of the methods in Fig. 2(d)–(f), where we denote by $\mathrm{IWMLR}_{\gamma*}$ the IWMLR method that uses the true value of γ rather than the estimated value. As shown in Fig. 2(d)–(f), the examples that are the most difficult for classification are around the coordinate origin, due to the influence of noisy examples. In the case of MLR [Fig. 2(d)], we see that the classification hyperplanes of *azure versus cyan*, *azure versus red*, *purple versus red*, and *blue versus cyan* are significantly biased. We note that the predicted labels tend to the *cyan* and *red* classes, which may be due to the imbalanced distribution of the classes and the influence of noisy examples. By contrast, the classification hyperplanes move toward the ground truth [Fig. 2(c)] by using the proposed weighting strategy, as shown in Fig. 2(e). If we

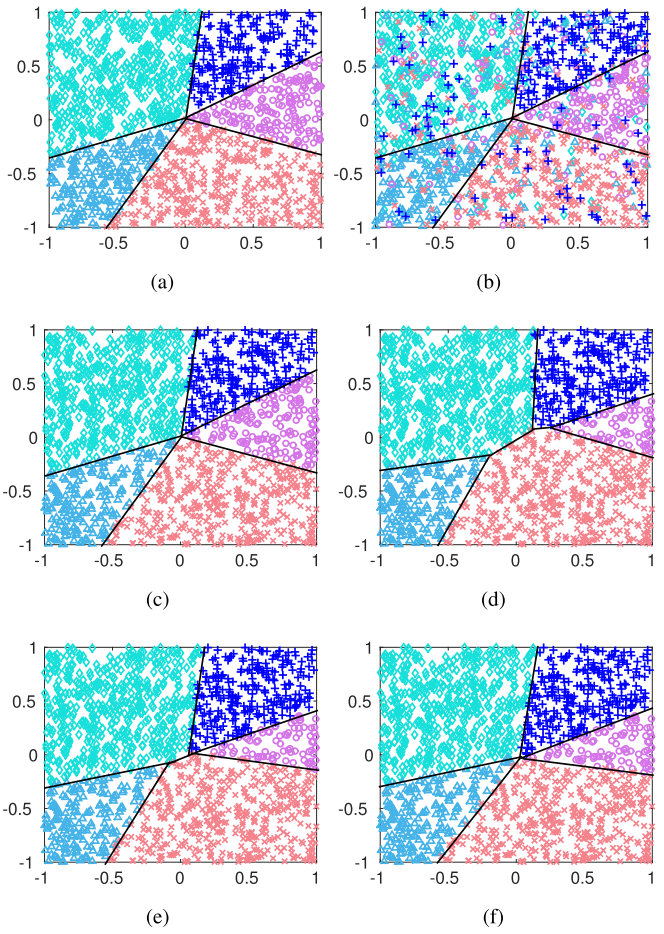


Fig. 2. Classification of the synthetic data. The classification planes are plotted as lines among the classes. (a) Noise-free data. (b) Noisy training data when $\gamma = 0.3$. (c) Test data. (d) Classification results of MLR (89.80%). (e) Classification results of IWMLR (91.96%). (f) Classification results of $\mathrm{IWMLR}_{\gamma*}$ (94.40%).

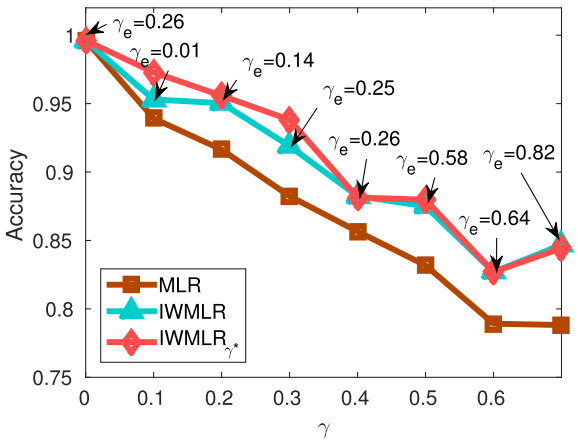


Fig. 3. Accuracy comparison of MLR, IWMLR, and IWMLR$_{\gamma*}$ on synthetic data. γ_e denotes the estimated value of γ in IWMLR.

TABLE I
STATISTICS OF THE DATA SETS

Dataset	#Classes	#Features	#Examples
glass	6	9	214
vehicle	4	18	846
dna	3	180	3186
letter	26	16	20000
protein	3	357	24387
satimage	6	36	6435
sector	105	55197	9619

use the true value γ^*, the predicted hyperplanes are more accurate [see Fig. 2(f)]. Fig. 2 indicates that the proposed method can achieve a considerable improvement (above 2%) over the original MLR.

Next, we investigate the influence of the estimation of γ with respect to the classification accuracy. We compare the performances of MLR, IWMLR, and IWMLR$_{\gamma*}$ for the considered γ values. Fig. 3 plots the results, from which we see that there exist perceived biases between the estimated γ values and the true values. Despite this, the biases do not have significant impacts on the classification accuracy, and IWMLR works almost as well as IWMLR$_{\gamma*}$, and is better than MLR in all cases. In the noise-free case, i.e., $\gamma = 0$, all the classifiers achieve the same accuracy. All these inform us that the method of estimating γ proposed in Theorem 1 is effective, and moreover, the proposed importance reweighting strategy is tolerant to the biases introduced by the γ estimation.

B. Real Data

We carried out experiments on seven data sets from the LIBSVM repository[3] to evaluate the classification performance of RGP, Rob_MAd, SVM, IWSVM, MLR, and IWMLR when different proportions of noisy labels were present in the data. The information about the data sets is summarized in Table I. The competitor RGP was shown to have the complexity of $\mathcal{O}(n^3)$ [14], failing to be applied on data sets of large size. Thus, the results of RGP on *glass* and *vehicle* are only reported. We also note that RGP and Rob_MAd are nonlinear classifiers, which perform better than the linear classifiers on nonlinear data. To make a fair comparison on small-sized data sets, including *glass* and *vehicle*, we realized the other methods, including SVM, IWSVM, MLR, and IWMLR, by using the polynomial kernel to boost their performances. On the large-sized data sets, including *dna*, *letter*, *protein*, *satimage*, and *sector*, linear SVM and MLR classifiers were utilized, while the decision tree was used in Rob_MAd.

Table II details the results of different methods on each data set. We have applied a significance test (i.e., the F-test) on the 5×2-fold cross-validation results obtained by SVM and IWSVM, and also on the results obtained by MLR and IWMLR. Specifically, in each data set and γ case, the 5×2-fold cross-validation results in five averaged accuracies for each classifier. The F-test is conducted on, for example, the five accuracies of SVM and the five accuracies of IWSVM. Such a significance test is to statistically evaluate the significance of the performance difference between IWSVM and SVM, and the significance of the difference between IWMLR and MLR. The comparison indicates that the proposed importance reweighting strategy significantly improves the performance of the classifiers in most settings. For example, in all cases, IWMLR and IWSVM perform better than MLR and SVM, respectively. RGP and Rob_MAd perform poorly compared with the proposed methods. Notably, the strategy used in either RGP or Rob_MAd is nontrivial to be applied to other surrogate loss functions, whereas our method can be easily used to assist a well-performing classifier when handling noisy data. The standard deviations of the results on small data sets are larger than that on large data sets due to insufficient training samples. Overall, we observe that the proposed method can enhance the noise tolerance of the classifiers to high amounts of label noise in the data.

Note that the efficiency of the proposed strategy is independent of any specified surrogate loss function and any multiclass classification setting. One main factor which may affect the applicability of our method to large data sets is the estimation of the conditional probability $P_{D_\gamma}(\boldsymbol{Y}|\boldsymbol{X})$, where we resort to the method of KLIEP. Since KLIEP requires numerical optimization to obtain the solution [56], this process would be slow when the number of training examples becomes too large. However, empirical experience tells us that KLIEP can be applied on a moderately sized data set.

C. Effects of Asymmetric RCN Model

The previous experiments are based on a symmetric noise assumption, saying that the proportion γ is identical for all classes. In this section, we investigate the performance of the proposed method when the assumption is violated. To simulate the data generation process under an asymmetric RCN model, we operate as follows. Given a data set containing m classes, we randomly choose $40\% \times m$ classes, such that the examples in those classes are not corrupted by noise. For the rest $60\% \times m$ classes, γ proportion of those examples are reassigned random labels as before. In this way, the resultant label noise exhibits an asymmetric property.

[3]https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/multiclass.html

TABLE II

ACCURACY (%) COMPARISON OF DIFFERENT MULTICLASS CLASSIFICATION ALGORITHMS ON REAL DATA. • INDICATES THAT IWSVM (IWMLR) IS SIGNIFICANTLY BETTER THAN SVM (MLR) AT LEVEL $p < 0.05$ IN F-TEST. THE BEST AVERAGE RESULT OF ALL γ CASES IN EACH DATA SET IS MARKED BY BOLD

Datasets	γ	RGP	Rob_MAd	SVM	IWSVM	MLR	IWMLR
glass	0.1	59.62±3.23	68.30±6.49	66.23±4.74	68.17±4.96 •	63.58±3.81	65.77±2.63 •
	0.2	52.45±4.14	62.45±6.24	62.08±3.49	62.74±3.36	62.08±1.81	63.11±1.29 •
	0.3	53.96±5.48	60.57±3.16	56.23±5.68	57.96±6.46 •	57.74±5.72	61.14±3.24 •
	0.4	52.45±6.03	57.92±5.40	53.77±11.22	55.04±10.91	56.42±7.85	57.25±7.61
	0.5	45.28±8.33	51.51±8.86	52.64±2.94	54.45±3.56 •	52.64±6.91	54.98±5.99 •
	0.6	54.72±4.06	47.74±5.53	52.08±9.77	53.43±9.96 •	51.51±5.48	53.38±5.03 •
	0.7	38.68±12.12	39.43±6.85	48.94±2.04	51.39±2.88 •	42.26±6.85	43.91±7.08
	avg.	51.02	55.42	55.99	**57.60**	55.18	57.08
vehicle	0.1	72.89±2.00	70.81±2.26	74.74±1.70	76.93±1.98 •	76.35±2.92	78.50±0.99 •
	0.2	72.70±1.87	71.85±2.28	75.07±1.92	75.79±2.24	75.73±2.40	77.25±2.55 •
	0.3	70.47±2.69	64.31±3.63	71.66±2.96	72.25±3.03 •	71.71±3.10	73.49±3.26 •
	0.4	68.25±1.61	62.89±2.35	69.72±2.55	71.62±2.18 •	70.47±3.62	73.03±1.20 •
	0.5	60.00±1.99	60.52±1.94	63.65±8.51	64.58±8.55 •	64.79±3.44	66.61±2.84 •
	0.6	49.15±7.55	52.99±0.96	63.08±2.31	64.22±2.05 •	64.88±3.34	66.49±1.81 •
	0.7	44.74±5.81	47.49±4.39	54.83±1.82	56.00±1.77 •	54.98±5.58	56.64±6.13 •
	avg.	62.60	61.55	67.54	68.77	68.42	**70.29**
dna	0.1	-	92.44±1.12	92.35±0.32	92.90±0.50	93.03±0.88	93.49±0.83
	0.2	-	91.21±0.46	89.92±0.78	91.11±0.60 •	91.86±0.99	92.56±0.72 •
	0.3	-	88.61±0.83	87.58±0.77	88.55±0.65 •	90.64±1.02	91.27±0.67 •
	0.4	-	85.59±1.51	84.13±1.19	84.49±1.28	88.77±1.50	89.50±1.51 •
	0.5	-	78.22±4.19	78.34±2.22	79.71±1.87 •	86.17±0.57	86.88±0.73
	0.6	-	72.39±1.54	72.76±4.23	72.80±3.15	81.06±2.17	82.41±2.62 •
	0.7	-	63.98±3.96	65.18±1.84	67.63±2.13 •	77.79±4.42	78.84±3.76 •
	avg.	-	81.77	81.47	83.17	87.04	**87.91**
letter	0.1	-	59.02±0.67	64.44±0.39	64.75±0.57 •	75.24±0.43	75.69±0.89
	0.2	-	57.37±0.27	62.61±0.27	63.02±0.45	73.42±0.56	73.73±0.59 •
	0.3	-	55.92±0.34	60.88±0.35	61.65±0.51 •	71.51±0.56	72.13±0.65 •
	0.4	-	54.42±0.40	59.01±0.75	59.70±0.65 •	69.51±0.39	70.15±0.47 •
	0.5	-	52.52±0.83	58.21±0.54	59.00±0.21 •	67.24±0.38	67.78±0.87 •
	0.6	-	49.96±0.53	55.04±0.79	55.81±1.04	64.21±0.65	64.92±1.07
	0.7	-	47.35±1.20	51.15±1.02	52.20±0.55 •	60.36±0.70	60.98±0.59 •
	avg.	-	53.79	58.76	59.45	68.78	**69.34**
protein	0.1	-	59.85±0.26	67.44±0.40	67.69±0.33 •	67.64±0.44	68.05±0.51 •
	0.2	-	59.18±0.67	67.07±0.37	67.34±0.60	67.45±0.35	67.73±0.23 •
	0.3	-	57.50±0.52	66.26±0.38	66.75±0.35 •	66.76±0.11	67.16±0.27 •
	0.4	-	52.32±5.26	65.50±0.48	65.86±0.35 •	66.37±0.66	66.72±0.78 •
	0.5	-	51.89±1.08	63.94±0.38	64.49±0.34 •	64.95±0.82	65.41±0.90 •
	0.6	-	48.93±0.28	61.58±0.94	62.40±0.78 •	63.37±0.95	63.81±0.70 •
	0.7	-	45.68±0.59	58.66±0.69	59.11±0.62 •	61.12±1.00	61.88±0.66 •
	avg.	-	53.62	64.35	64.81	65.38	**65.82**
satimage	0.1	-	80.60±0.53	79.31±0.71	79.69±0.55 •	83.47±0.23	83.80±0.40 •
	0.2	-	78.78±0.46	77.41±1.08	77.85±0.87 •	82.25±0.40	82.75±0.42 •
	0.3	-	77.77±0.94	76.95±0.67	77.35±0.72 •	81.42±0.73	81.82±0.81 •
	0.4	-	77.39±1.42	76.06±0.55	76.17±0.56	79.75±0.86	80.09±0.97 •
	0.5	-	77.06±1.21	75.73±0.75	76.00±0.91 •	78.44±1.32	79.21±0.87 •
	0.6	-	76.51±1.27	75.25±0.83	75.51±0.91	77.30±1.69	78.12±1.34 •
	0.7	-	75.24±2.73	74.29±0.95	74.66±0.98	74.60±1.67	75.12±1.18 •
	avg.	-	77.62	76.43	76.75	79.61	**80.13**
sector	0.1	-	90.63±1.42	90.40±0.24	91.23±0.48 •	88.85±0.48	89.07±0.61 •
	0.2	-	86.45±2.56	87.42±0.59	87.85±0.19 •	86.52±0.67	86.79±0.78
	0.3	-	82.65±2.10	82.56±0.56	83.03±0.41 •	82.24±0.62	82.78±0.54 •
	0.4	-	77.00±1.32	77.87±1.01	78.41±0.56 •	76.96±0.65	77.53±0.66 •
	0.5	-	68.76±3.59	71.93±1.07	72.04±1.03	69.48±0.65	69.87±0.46
	0.6	-	55.25±2.15	62.50±1.27	62.73±1.10	58.90±1.31	59.76±0.74 •
	0.7	-	50.11±1.88	50.75±0.79	51.69±0.33 •	46.70±0.77	47.38±0.46 •
	avg.	-	72.98	74.78	**75.28**	72.81	73.31

The compared methods include RGP, Rob_MAd, SVM, IWSVM, MLR, and IWMLR. The specifications for each method and the data sets selected in this experiment are the same as those in Section V-B. Note that γ is estimated over the whole data set, rather than over the corrupted classes. The weight $\beta(\boldsymbol{X}, \hat{\boldsymbol{Y}})$ is estimated for all examples in a data set.

Table III lists the performances of different methods on the generated data. F-test is again employed to statistically

TABLE III

ACCURACY (%) COMPARISON OF DIFFERENT MULTICLASS CLASSIFICATION ALGORITHMS ON ASYMMETRICALLY NOISY DATA. • INDICATES THAT IWSVM (IWMLR) IS SIGNIFICANTLY BETTER THAN SVM (MLR) AT LEVEL $p < 0.05$ IN F-TEST. THE BEST AVERAGE RESULT OF ALL γ CASES IN EACH DATA SET IS MARKED BY BOLD

Datasets	γ	RGP	Rob_MAd	SVM	IWSVM	MLR	IWMLR
glass	0.1	58.37±3.27	68.63±6.05	66.13±4.90	67.17±4.82	61.66±3.49	65.55±3.38 •
	0.2	51.90±4.18	62.67±5.38	61.32±5.26	61.97±3.80 •	61.24±1.12	62.40±1.11 •
	0.3	53.79±5.39	60.08±3.88	55.54±6.33	57.55±5.39 •	57.71±5.50	61.37±4.75 •
	0.4	50.95±6.50	57.83±5.03	52.74±6.60	55.34±6.22	55.78±7.23	56.89±8.04
	0.5	43.87±8.20	51.64±9.30	52.37±3.46	54.19±2.90 •	51.61±6.49	53.68±5.61
	0.6	55.09±4.26	46.64±5.11	52.36±9.76	52.51±9.56	52.53±5.32	53.02±5.32
	0.7	38.56±12.82	38.18±6.25	48.30±1.57	51.36±3.33 •	41.43±8.19	42.69±8.01
	avg.	50.36	55.10	55.54	**57.16**	54.56	56.51
vehicle	0.1	72.93±3.26	70.00±2.47	74.45±1.83	77.13±1.74 •	75.57±3.43	78.94±1.18 •
	0.2	71.34±2.28	71.03±3.17	74.48±1.80	75.37±1.75	75.81±3.56	76.64±3.31
	0.3	69.07±2.38	63.00±3.52	72.05±3.04	71.22±4.29	72.08±4.01	74.07±3.26 •
	0.4	68.67±2.67	62.69±0.83	68.96±3.09	69.97±1.86	70.89±2.19	72.74±2.12 •
	0.5	59.89±3.43	60.39±2.91	63.43±7.32	63.78±7.06	64.32±3.75	66.93±2.18 •
	0.6	48.65±8.25	51.49±1.62	62.70±2.07	63.94±1.29 •	64.08±3.47	64.48±2.70
	0.7	43.64±6.74	47.22±3.54	53.95±1.69	55.15±1.76 •	54.66±5.27	55.92±6.37
	avg.	62.03	60.83	67.15	68.08	68.20	**69.96**
dna	0.1	-	91.84±1.36	92.08±1.51	92.69±0.64 •	91.50±1.14	93.88±1.48 •
	0.2	-	90.90±1.50	89.45±1.52	90.07±1.42	91.43±2.11	91.66±0.51
	0.3	-	89.10±2.03	87.58±1.47	89.58±1.56 •	89.65±1.68	90.85±1.25
	0.4	-	83.96±1.06	82.91±2.26	84.17±1.05 •	88.91±1.17	90.59±1.69
	0.5	-	77.41±4.24	78.24±2.68	78.98±2.07	86.36±1.61	86.80±1.68 •
	0.6	-	71.48±2.52	72.92±4.11	74.85±3.92 •	81.14±2.99	82.16±2.86
	0.7	-	64.32±3.27	65.05±1.45	66.23±2.09	77.28±5.00	78.21±4.29
	avg.	-	81.29	81.18	82.37	86.61	**87.74**
letter	0.1	-	59.14±1.87	64.50±2.10	64.51±0.82	74.65±1.62	76.07±0.98 •
	0.2	-	57.26±2.34	63.12±1.03	63.30±0.29	74.22±1.95	74.71±0.71
	0.3	-	56.50±1.12	61.44±1.84	62.36±1.39 •	70.90±1.15	71.47±1.42
	0.4	-	54.02±1.66	58.94±1.91	59.72±1.33 •	69.39±1.01	71.29±1.59 •
	0.5	-	51.96±0.61	58.65±2.74	58.97±0.79	66.45±0.91	66.64±0.32
	0.6	-	48.70±1.41	54.59±1.63	55.63±1.43	63.55±1.86	64.51±1.53 •
	0.7	-	47.30±1.80	50.34±1.95	51.78±1.43 •	61.42±0.50	61.46±0.10
	avg.	-	53.55	58.80	59.42	68.65	**69.28**
protein	0.1	-	59.54±1.02	67.16±1.86	67.87±1.11	66.86±1.21	68.49±1.22 •
	0.2	-	59.61±1.13	66.58±2.12	67.64±0.86 •	67.43±0.99	67.76±0.89
	0.3	-	56.71±1.61	65.41±1.37	66.67±0.85 •	66.56±2.99	67.05±1.19
	0.4	-	51.93±5.32	64.02±1.69	65.54±1.23 •	65.40±2.77	66.74±1.55 •
	0.5	-	52.01±2.03	63.26±1.18	64.44±0.88 •	63.95±0.95	65.28±1.24 •
	0.6	-	47.18±1.11	62.08±1.90	62.58±1.53	63.51±1.41	64.41±0.35
	0.7	-	45.12±0.63	59.63±0.80	60.61±0.35 •	60.63±1.30	60.82±1.05
	avg.	-	53.16	64.03	65.05	64.91	**65.79**
satimage	0.1	-	81.08±1.35	78.61±1.01	79.56±0.90 •	83.44±2.38	83.55±0.12
	0.2	-	78.36±1.30	76.13±0.75	77.83±2.03 •	82.46±1.51	82.68±0.49
	0.3	-	77.53±1.89	76.84±0.97	77.29±0.62	79.88±1.86	81.34±1.43 •
	0.4	-	77.01±1.72	75.69±1.37	75.70±0.02	78.31±0.97	79.73±2.05
	0.5	-	75.29±2.26	75.59±0.95	76.18±0.70 •	78.45±1.77	79.36±1.34 •
	0.6	-	76.33±1.81	74.50±1.00	75.25±0.92 •	76.43±2.08	77.77±1.65 •
	0.7	-	74.89±2.67	74.46±2.47	74.61±0.95	74.58±2.61	74.88±0.82
	avg.	-	77.21	75.97	76.63	79.08	**79.90**
sector	0.1	-	89.82±2.77	90.55±1.19	90.72±0.26	86.86±1.71	88.12±1.60 •
	0.2	-	85.95±2.96	88.02±2.22	87.93±0.72	87.70±2.85	87.78±0.64
	0.3	-	80.58±1.76	82.76±2.48	83.20±0.98	80.87±1.78	82.96±1.53 •
	0.4	-	76.98±1.74	78.97±1.03	79.43±0.88 •	75.84±0.75	76.51±0.92 •
	0.5	-	67.98±4.83	72.93±1.61	73.34±0.58	69.45±1.76	69.95±0.44
	0.6	-	54.91±3.64	61.91±1.46	63.10±1.40 •	58.43±1.74	59.00±0.64
	0.7	-	49.91±2.45	50.04±1.13	51.64±1.86 •	46.09±0.74	46.92±1.72
	avg.	-	72.31	75.03	**75.62**	72.18	73.09

assess the significance of the performance difference between IWSVM and SVM and that between IWMLR and MLR. From those results, we can conclude that even though the symmetric noise assumption is violated, the performances of all methods are consistent with those in Table II. Specifically, the proposed importance reweighting strategy can improve the performances of both SVM and MLR according to the marked values by significant test. On average, the IW variants achieve

the best performances in all data sets. Notably, although the asymmetric assumption may bias the estimation of γ according to Section IV-B, the classifiers can still benefit from the proposed importance reweighting strategy.

The results in Tables II and III suggest that in the presence of either asymmetric or symmetric noise, the learning of the classifiers can be easily biased by equally treating the influence of each example. Instead, reweighting according to the importance of the examples is a preferred way to suppress the effects of noise, demonstrating the effectiveness of the proposed method.

VI. Conclusion

Collecting extensive training data for an intelligent learning system is an extremely time-consuming and labor-consuming task. Employing nonexperts may reduce the cost, but cannot guarantee the correctness of the obtained labels. Therefore, how to alleviate the difficulty of the learning system by utilizing inaccurate data remains an open issue. In this paper, we target the problem of multiclass classification of the data, a proportion of which are randomly labeled. We show that such an issue is indeed a label noise problem, or more specifically, a label flip noise problem in the multiclass classification. Considering that the in-flip and out-flip probabilities of any one class are the same as those of the other classes, we impose a symmetric RCN model on the multiclass setting. We reexplore the importance reweighting strategy to address this problem, which is applicable to any traditional surrogate loss functions and to different multiclass classification settings. Our theoretical results provide an effective method of weight estimation, and indicate that the proportion of randomly labeled examples can be estimated with a proven upper bound. Convergence analyses are also provided to assure that the learned classifier is consistent with the optimal classifier for noise-free data. We apply the proposed strategy to two conventional classifiers, including MLR and SVM. Experimental results on synthetic and real data demonstrate the effectiveness of the proposed strategy for handling the noisy labels, and also verify that our method is not significantly affected when the symmetric noise assumption is violated.

Further work will concentrate on asymmetric noise in the multiclass setting. Specifically, an accurate estimation of the weights will be researched. In addition, the estimation of noise rate for each class is a critical yet difficult issue that needs to be solved.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[2] C. Xu, D. Tao, and C. Xu, "Robust extreme multi-label learning," in *Proc. SIGKDD*, Aug. 2016, pp. 13–17.

[3] Y. Wang, C. Xu, C. Xu, and D. Tao, "Beyond RPCA: Flattening complex noise in the frequency domain," in *Proc. AAAI*, 2017, pp. 500–505.

[4] J. Howe, *Crowdsourcing: How the Power of the Crowd is Driving the Future of Business*. New York, NY, USA: Random House, 2008.

[5] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on Amazon mechanical turk," in *Proc. ACM SIGKDD Workshop Human Comput.*, 2010, pp. 64–67.

[6] P. M. Long and R. A. Servedio, "Random classification noise defeats all convex potential boosters," *Mach. Learn.*, vol. 78, no. 3, pp. 287–304, 2010.

[7] D. Angluin and P. Laird, "Learning from noisy examples," *Mach. Learn.*, vol. 2, no. 4, pp. 343–370, 1988.

[8] J. A. Aslam and S. E. Decatur, "On the sample complexity of noise-tolerant learning," *Inf. Process. Lett.*, vol. 57, no. 4, pp. 189–195, 1996.

[9] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Proc. NIPS*, 2013, pp. 1196–1204.

[10] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.

[11] H. R. Moore *et al.*, "Robust regression using maximum-likelihood weighting and assuming cauchy-distributed random error," DTIC, Fort Belvoir, VA, USA, Tech. Rep., Jun. 1977.

[12] K. Crammer and D. D. Lee, "Learning via Gaussian herding," in *Proc. NIPS*, 2010, pp. 451–459.

[13] B. van Rooyen, A. Menon, and R. C. Williamson, "Learning with symmetric label noise: The importance of being unhinged," in *Proc. NIPS*, 2015, pp. 10–18.

[14] D. Hernández-Lobato, J. M. Hernández-Lobato, and P. Dupont, "Robust multi-class Gaussian process classification," in *Proc. NIPS*, 2011, pp. 280–288.

[15] J. Bootkrajang and A. Kabán, "Multi-class classification in the presence of labelling errors," in *Proc. ESANN*, 2011, pp. 345–350.

[16] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolution networks with noisy labels," in *Proc. ICLR Workshop Track*, 2015, pp. 1–11.

[17] J. A. Sáez, M. Galar, J. Luengo, and F. Herrera, "Analyzing the presence of noise in multi-class problems: Alleviating its influence with the one-vs-one decomposition," *Knowl. Inf. Syst.*, vol. 38, no. 1, pp. 179–206, 2014.

[18] B. Sun, S. Chen, J. Wang, and H. Chen, "A robust multi-class adaboost algorithm for mislabeled noisy data," *Knowl.-Based Syst.*, vol. 102, pp. 87–102, Jun. 2016.

[19] S. Mehrkanoon, C. Alzate, R. Mall, R. Langone, and J. A. K. Suykens, "Multiclass semisupervised learning based upon kernel spectral clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 720–733, Apr. 2015.

[20] L. G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[21] C. Gentile and D. P. Helmbold, "Improved lower bounds for learning from noisy examples: An information-theoretic approach," in *Proc. Annu. Conf. Comput. Learn. Theory*, 1998, pp. 104–115.

[22] W. Liu, P. P. Pokharel, and J. C. Príncipe, "Correntropy: Properties and applications in non-Gaussian signal processing," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5286–5298, Nov. 2007.

[23] N. Manwani and P. Sastry, "Noise tolerance under risk minimization," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1146–1151, Jun. 2013.

[24] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S.-I. Amari, "Bayesian robust tensor factorization for incomplete multiway data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 736–748, Apr. 2016.

[25] Q. Miao, Y. Cao, G. Xia, M. Gong, J. Liu, and J. Song, "RBoost: Label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2216–2228, Nov. 2016.

[26] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.

[27] Y. Freund, "An adaptive version of the boost by majority algorithm," *Mach. Learn.*, vol. 43, no. 3, pp. 293–318, 2001.

[28] J.-W. Sun, F.-Y. Zhao, C.-J. Wang, and S.-F. Chen, "Identifying and correcting mislabeled training instances," in *Proc. Future Generat. Commun. Netw.*, vol. 1. Dec. 2007, pp. 244–250.

[29] F. Muhlenbach, S. Lallich, and D. A. Zighed, "Identifying and handling mislabelled instances," *J. Intell. Inf. Syst.*, vol. 22, no. 1, pp. 89–109, 2004.

[30] G. H. John, "Robust decision trees: Removing outliers from databases," in *Proc. KDD*, 1995, pp. 174–179.

[31] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.

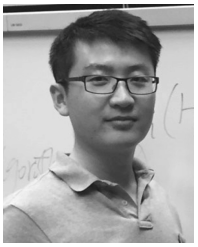
[32] M. Kearns, "Efficient noise-tolerant learning from statistical queries," *J. ACM*, vol. 45, no. 6, pp. 983–1006, 1998.

[33] N. D. Lawrence and B. Schölkopf, "Estimating a kernel fisher discriminant in the presence of label noise," in *Proc. ICML*, 2001, pp. 306–313.

[34] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," in *Proc. ACML*, 2011, pp. 97–112.
[35] G. Patrini, F. Nielsen, R. Nock, and M. Carioni. (2016). "Loss factorization, weakly supervised learning and label noise robustness." [Online]. Available: https://arxiv.org/abs/1602.02450
[36] W. Gao, L. Wang, Y.-F. Li, and Z.-H. Zhou, "Risk minimization in the presence of label noise," in *Proc. AAAI*, 2016, pp. 1575–1581.
[37] T. Yang, M. Mahdavi, R. Jin, L. Zhang, and Y. Zhou, "Multiple kernel learning from noisy labels by stochastic programming," in *Proc. ICML*, 2012, pp. 233–240.
[38] E. Cohen, "Learning noisy perceptrons by a perceptron in polynomial time," in *Proc. Annu. Symp. Found. Comput. Sci.*, Oct. 1997, pp. 514–523.
[39] G. Stempfel and L. Ralaivola, "Learning kernel perceptrons on noisy data using random projections," in *Algorithmic Learning Theory*. Sendai, Japan: Springer, 2007, pp. 328–342.
[40] R. Khardon and G. Wachman, "Noise tolerant variants of the perceptron algorithm," *J. Mach. Learn. Res.*, vol. 8, pp. 227–248, 2007.
[41] X. Geng, "Label distribution learning," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1734–1748, Jul. 2016.
[42] X. Geng and Y. Xia, "Head pose estimation based on multivariate label distribution," in *Proc. CVPR*, 2014, pp. 1837–1842.
[43] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.
[44] A. Rocha and S. Klein Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ECOC-based approaches," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 289–302, Feb. 2014.
[45] J. Ortigosa-Hernandez, I. Inza, and J. A. Lozano, "Semisupervised multiclass classification problems with scarcity of labeled data: A theoretical study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2602–2614, Dec. 2016.
[46] M. Lin, K. Tang, and X. Yao, "Dynamic sampling approach to training neural networks for multiclass imbalance classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 4, pp. 647–660, Apr. 2013.
[47] X.-Y. Liu, Q.-Q. Li, and Z.-H. Zhou, "Learning imbalanced multiclass data with optimal dichotomy weights," in *Proc. ICDM*, 2013, pp. 478–487.
[48] Z.-H. Zhou and X.-Y. Liu, "On multi-class cost-sensitive learning," *Comput. Intell.*, vol. 26, no. 3, pp. 232–257, 2010.
[49] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.
[50] K. Zhang, M. Gong, and B. Schölkopf, "Multi-source domain adaptation: A causal view," in *Proc. AAAI*, 2015, pp. 3150–3157.
[51] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *Proc. ICML*, 2016, pp. 2839–2848.
[52] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, "Covariate shift by kernel mean matching," *Dataset Shift Mach. Learn.*, vol. 3, no. 4, p. 5, 2009.
[53] V. Vapnik, I. Braga, and R. Izmailov. (2013). "Constructive setting of the density ratio estimation problem and its rigorous solution." [Online]. Available: https://arxiv.org/abs/1306.0407
[54] T. Kanamori, "Density ratio estimation: A comprehensive review," *RIMS Kokyuroku*, vol. 1703, pp. 10–31, Mar. 2010.
[55] T. Kanamori, T. Suzuki, and M. Sugiyama, "Theoretical analysis of density ratio estimation," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 93, no. 4, pp. 787–798, 2010.
[56] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proc. NIPS*, 2008, pp. 1433–1440.
[57] C. Scott, G. Blanchard, and G. Handy, "Classification with asymmetric label noise: Consistency and maximal denoising," in *Proc. COLT*, 2013, pp. 489–511.
[58] C. Scott, "A rate of convergence for mixture proportion estimation, with application to learning from noisy labels," in *Proc. AISTATS*, 2015, pp. 838–846.
[59] A. K. Menon, B. van Rooyen, C. S. Ong, and R. C. Williamson, "Learning from corrupted binary labels via class-probability estimation," in *Proc. ICML*, 2015, pp. 125–134.
[60] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 957–968, Jun. 2005.
[61] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
[62] S. Zhou, "Sparse LSSVM in primal using Cholesky factorization for large-scale problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 783–795, Apr. 2016.
[63] C. G. Sentelle, G. C. Anagnostopoulos, and M. Georgiopoulos, "A simple method for solving the svm regularization path for semidefinite kernels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 709–722, Apr. 2016.
[64] G. Patrini, A. Rozza, A. Menon, R. Nock, and L. Qu. (2016). "Making neural networks robust to label noise: A loss correction approach." [Online]. Available: https://arxiv.org/abs/1609.03683

Ruxin Wang received the B.Eng. degree from Xidian University, Xi'an, China, the M.Sc. degree from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from the University of Technology Sydney, Ultimo, NSW, Australia.

He is currently a Research Scientist with the Yunshangyun Artificial Intelligence Institute, Kunming, China, and also a Core Member with the Yunnan Union Visual Innovation Technology, Kunming. His current research interests include machine learning, deep learning, and computer vision.



Tongliang Liu is currently a Lecturer with the School of Information Technologies and the Faculty of Engineering and Information Technologies, and a core member in the UBTech Sydney AI Institute, at The University of Sydney. He received the BEng degree in electronic engineering and information science from the University of Science and Technology of China, and the PhD degree from the University of Technology Sydney. His research interests include statistical learning theory, computer vision, and optimization. He has authored and co-authored 20+ research papers including IEEE T-PAMI, T-NNLS, T-IP, ICML, and KDD.

Dacheng Tao (F'15) is Professor of Computer Science and ARC Future Fellow in the School of Information Technologies and the Faculty of Engineering and Information Technologies, and the Inaugural Director of the UBTech Sydney Artificial Intelligence Institute, at The University of Sydney. He mainly applies statistics and mathematics to Artificial Intelligence and Data Science. His research interests spread across computer vision, data science, image processing, machine learning, and video surveillance. His research results have expounded in one monograph and 500+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, CIKM, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM; and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM'07, the best student paper award in IEEE ICDM'13, and the 2014 ICDM 10-year highest-impact paper award. He received the 2015 Australian Scopus-Eureka Prize, the 2015 ACS Gold Disruptor Award and the 2015 UTS Vice-Chancellor's Medal for Exceptional Research. He is a Fellow of the IEEE, OSA, IAPR and SPIE.