Harnessing Side Information for Classification Under Label Noise

Yang Wei, Chen Gong, *Member, IEEE*, Shuo Chen, Tongliang Liu, *Member, IEEE*, Jian Yang, *Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Practical data sets often contain the label noise caused by various human factors or measurement errors, which means that a fraction of training examples might be mistakenly labeled. Such noisy labels will mislead the classifier training and severely decrease the classification performance. Existing approaches to handle this problem are usually developed through various surrogate loss functions under the framework of empirical risk minimization. However, they are only suitable for binary classification and also require strong prior knowledge. Therefore, this article treats the example features as side information and formulates the noisy label removal problem as a matrix recovery problem. We denote our proposed method as "label noise handling via side information" (LNSI). Specifically, the observed label matrix is decomposed as the sum of two parts, in which the first part reveals the true labels and can be obtained by conducting a low-rank mapping on the side information; and the second part captures the incorrect labels and is modeled by a row-sparse matrix. The merits of such formulation lie in three aspects: 1) the strong recovery ability of this strategy has been sufficiently demonstrated by intensive theoretical works on side information; 2) multi-class situations can be directly handled with the aid of learned projection matrix; and 3) only very weak assumptions are required for model design, making LNSI applicable to a wide range of practical problems. Moreover, we theoretically derive the generalization bound of LNSI and show that the expected classification error of LNSI is upper bounded. The experimental results on a variety of data sets including UCI benchmark data sets and practical data sets confirm the superiority of LNSI to state-of-the-art approaches on label noise handling.**

***Index Terms*— Classification, generalization bound, label noise, matrix recovery, side information.**

I. Introduction

TRADITIONALLY, a reliable supervised classifier, such as support vector machines (SVMs) or convolutional neural networks (CNNs), is usually trained based on the sufficient correctly labeled data. Unfortunately, the real-world data sets often contain the noise in label space, which means that a fraction of training examples are erroneously labeled [1]. For instance, as the numerous examples in many applications (e.g., image classification and document categorization) are manually annotated, the labeling errors are inevitably introduced due to the human fatigue. Disease diagnosis, in which the decision is strongly dependent on the experience and expertise of the doctors, is also very likely to include labeling errors. These noisy labels will significantly mislead the classifier training and then severely decrease the classification performance [2]. Hence, designing algorithms that account for the data with noisy labels is of great significance and has become a critical issue in the machine learning community.

Several approaches have been proposed to deal with the learning problem with label noise to prevent the performance decrease, and most of them are based on the minimization of empirical risk via a conditional probability model [3]–[6]. For example, Gao *et al.* [3] and Patrini *et al.* [6] analyze the empirical risk minimization in the presence of label noise by decomposing the loss function into a label-independent part and a label-dependent part. Manwani and Sastry [5] study the noise tolerance properties of risk minimization under different loss functions and provide insightful theoretical results. However, they are only applicable to binary classification and the extension to multi-class is nontrivial [7]. Moreover, these methods require the estimation of class prior, which is actually quite difficult in the presence of corrupted observed data.

On the other hand, side information is often utilized as additional knowledge to boost the performance of the certain

Manuscript received August 13, 2018; revised January 17, 2019 and April 11, 2019; accepted August 26, 2019. This work was supported by NSF of China under Grant 61602246, Grant 61973162, and Grant U1713208, in part by NSF of Jiangsu Province under Grant BK20171430, in part by the Fundamental Research Funds for the Central Universities under Grant 30918011319, in part by the Open Project of the State Key Laboratory of Integrated Services Networks through Xidian University under Grant ISN19-03, in part by the "Summit of the Six Top Talents" Program under Grant DZXX-027, in part by the "Young Elite Scientists Sponsorship Program" by Jiangsu Province, in part by the "Young Elite Scientists Sponsorship Program" by CAST under Grant 2018QNRC001, in part by the Program for Changjiang Scholars, in part by the "111" Program AH92005, in part by the ARC FL-170100117, in part by DP180103424, and in part by DE190101473. *(Corresponding author: Chen Gong.)*

Y. Wei and C. Gong are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China (e-mail: csywei@njust.edu.cn; chen.gong@njust.edu.cn).

S. Chen and J. Yang are with the PCA Laboratory, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the Jiangsu Key Laboratory of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: shuochen@njust.edu.cn; csjyang@njust.edu.cn).

T. Liu and D. Tao are with the UBTECH Sydney Artificial Intelligence Centre, School of Computer Science, Faculty of Engineering, 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 article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2938782

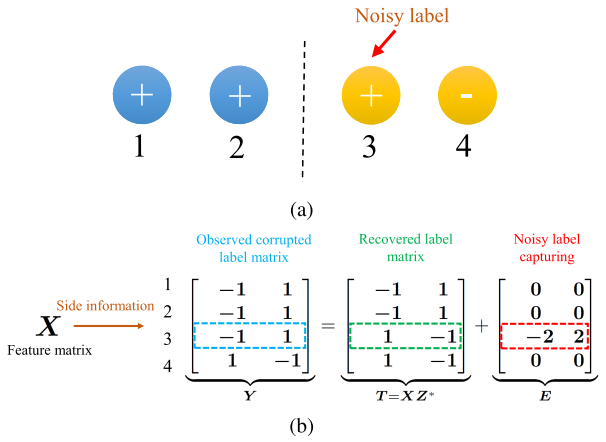


Fig. 1. Motivation illustration. (a) Four examples from two classes, among which the label of the 3rd example is incorrect. (b) $\boldsymbol{Y}$ is the corrupted label matrix, of which the rows represent the label vectors of four examples displayed in (a). By taking the example feature matrix $\boldsymbol{X}$ as side information, the observed label matrix $\boldsymbol{Y}$ can be ideally decomposed as the sum of a low-rank recovered label matrix $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{Z}^*$ and a row-sparse matrix $\boldsymbol{E}$. Note that the nonzero row in $\boldsymbol{E}$ exactly corresponds to the 3rd example with noisy label.

task, which has been widely used in many machine learning fields such as clustering [8] and multi-label learning [9]. For example, Zhao *et al.* [8] propose the matrix completion-based approach for multi-view clustering and first introduce the side information to aid the clustering process. Xu *et al.* [9] explore the side information to reduce the requirement on the number of observed entries for matrix completion and apply the method to transductive incomplete multi-label learning. From the review, we know that the side information has not been utilized for removing noisy labels. Therefore, this article provides a new paradigm for dealing with the label noise problem from the viewpoint of side information. Specifically, we formulate label correction as a label matrix recovery problem and treat the example features as side information to aid the recovery process [9]–[11]. Therefore, our proposed method is named as "label noise handling via side information" (LNSI). The paradigm of this article is shown in Fig. 1, which intuitively explains how to transform the noisy label removing problem to the label matrix recovery task by exploiting the side information. Fig. 1(a) shows the case where four examples belong to two classes, in which the 3rd example has been mistakenly labeled as positive and the noisy label is constituted. As illustrated in Fig. 1(b), given the observed label matrix $\boldsymbol{Y}$ and corresponding feature matrix $\boldsymbol{X}$, the true labels $\boldsymbol{T}$ can be obtained by conducting a low-rank mapping $\boldsymbol{Z}^*$ on the example features (i.e., $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{Z}^*$), and the incorrect labels are captured by a row-sparse matrix $\boldsymbol{E}$. The merits of our paradigm lie in three aspects: 1) LNSI is inherently suitable for multi-class classification, which does not need the one-versus-one or one-versus-the-rest operations; 2) sufficient theoretical results have demonstrated that the real label matrix can be exactly recovered under mild conditions [12], so LNSI is guaranteed to obtain satisfactory performance; and 3) LNSI seamlessly integrates the label noise removal and classifier parameter optimization into a unified framework. Due to the above merits, a reliable classifier can be learned to accurately classify the unseen test examples with different levels of training label noise. Furthermore, the experimental results on both benchmark data sets and practical data sets verify the superiority of the learned classifier.

II. Related Work

This section briefly reviews the representative prior works on label noise handling and side information utilization, as they are related to the proposed LNSI.

A. Label Noise Handling

Practically, the labels of training examples are often not reliable due to various limitations in data acquisition and data processing, and the noisy labels often occur that hinders the machine learning model to achieve sound performance.

One straightforward idea to address this problem is to improve the quality of training data. Since the training data are associated with noisy labels, the early-stage approaches first detect and eliminate label noise and then conduct the standard supervised classification algorithm. To implement noise detection, some works [13] explore the neighborhood relationship while some approaches rely on ensemble filters [14]. Nevertheless, the performances of these approaches are very sensitive to the quality of noise detection, which makes them unreliable for practical use.

To avoid the explicit noise detection step, plenty of efforts have been made recently to develop the algorithms that are inherently effective and robust to the noisy labels. Patrini *et al.* [6] decompose the conventional loss function into a label-independent part and a label-dependent part, in which only the latter is affected by label noise. Consequently, various surrogate loss functions can be designed. Similarly, Gao *et al.* [3] tackle the second part by deploying the labeled instance centroid to reduce the influence caused by label noise. Natarajan *et al.* [4] provide an unbiased estimator of loss function to deal with the symmetric noise, while Van Rooyen *et al.* [15] modify the traditional hinge loss and prove its robustness to label noise. Although these algorithms can reduce the adverse impact of label noise to some degree, they can only handle canonical binary classification and lack the theoretical guarantee of exact recovery on accurate labels.

Recently, some methods try to extend deep learning models to the case of noisy labels. For example, Khetan *et al.* [16] propose a new supervised learning algorithm which can jointly model labels and worker quality from noisy data. Patrini *et al.* [17] present a loss correction approach to train deep models that are robust to label noise. Han *et al.* [18] train two deep neural networks simultaneously and let them teach each other given every minibatch for combating with noisy labels. Meanwhile, Han *et al.* [19] also estimate the noise transition matrix with the assistance of human cognition and then derive a structure-aware probabilistic model for label noise handling. However, these deep learning-based methods are only suitable for specific tasks related to image analysis or natural language processing [17]. Moreover, the performance of these methods generally lacks theoretical guarantees.

Other representative works targeting label noise include [20] and [21].

B. Side Information

Side information serves as the additional information for accomplishing a certain task, which has been shown to be useful in solving many related problems. Practically, side information may have diverse formations.

Some works on the recommender system treat the user features and product features as side information so that the quality of completion of user-product preference matrix can be improved. For instance, in order to recover the unknown user scores of the preference matrix, Chiang *et al.* [10] incorporate the user and product side information to "describe" the row entities and column entities of a matrix, respectively. Similarly, Guo [11] develops a coembedding framework for matrix completion with side information, which learns a feature mapping as well as a label embedding. Zhao and Guo [22] propose a joint discriminative prediction model for personalized top-N recommendations with side information.

Other works take the available relationship between examples as side information. For example, Aggarwal *et al.* [23] perform text clustering by utilizing the links in the document, user-access behavior from web logs, or other nontextual attributes. Zhao *et al.* [8] conduct clustering with the aid of must-links and cannot-links between examples in addition to the regular input features. Zhang *et al.* [24] tackle the semisupervised classification by harnessing the example pairwise constraints as side information. Ahn *et al.* [25] study the binary rating estimation problem to quantify the value of graph side information, such as social graphs. In addition, Xue *et al.* [26] assimilate side information of the same format as observation in the robust principal component analysis.

From the above-mentioned analyses, we see that side information has been widely used in matrix completion, clustering, and semisupervised learning. However, it has not been well deployed in label noise handling that is the main target of this article.

III. Preliminaries

The utilization of side information for matrix completion has been studied by Chiang *et al.* [10], in which the model and theoretical analyses are studied when the side information contains noise. Suppose $\hat{\boldsymbol{R}} \in \mathbb{R}^{n_1 \times n_2}$ is a matrix with rank r that should be recovered, and $\hat{\boldsymbol{X}} \in \mathbb{R}^{n_1 \times d_1}$ and $\hat{\boldsymbol{Y}} \in \mathbb{R}^{n_2 \times d_2}$ (d_1 and d_2 denote feature dimensionality), respectively, record the row feature and column feature of $\hat{\boldsymbol{R}}$. The matrices $\hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{Y}}$ serve as the side information for recovering $\hat{\boldsymbol{R}}$. The model proposed by Chiang *et al.* [10] to recover $\hat{\boldsymbol{R}}$ is formulated as

$$\min_{\boldsymbol{M},\boldsymbol{N}} \sum_{(i,j)\in\Omega} \ell((\hat{\boldsymbol{X}}\boldsymbol{M}\hat{\boldsymbol{Y}}^\top + \boldsymbol{N})_{ij}, \hat{\boldsymbol{R}}_{ij}) + \lambda_M \|\boldsymbol{M}\|_* + \lambda_N \|\boldsymbol{N}\|_* \tag{1}$$

where "$\|\cdot\|_*$" is the nuclear norm, and "Ω" is the index set containing all observed entries of $\hat{\boldsymbol{R}}$. The observed $\hat{\boldsymbol{R}}$ can be decomposed into two parts: one is the low-rank matrix estimated from feature space $\hat{\boldsymbol{X}}\boldsymbol{M}\hat{\boldsymbol{Y}}^\top$, in which $\boldsymbol{M} \in \mathbb{R}^{d_1 \times d_2}$ is a coembedding matrix from features $\hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{Y}}$, and the other part is $\boldsymbol{N} \in \mathbb{R}^{n_1 \times n_2}$, which is used to capture the information that noisy features ($\hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{Y}}$) fail to describe. The loss function "$\ell(\cdot)$" requires that the recovered matrix is consistent with $\hat{\boldsymbol{R}}$ on the observed entries. Naturally, both $\hat{\boldsymbol{X}}\boldsymbol{M}\hat{\boldsymbol{Y}}^\top$ and $\boldsymbol{N}$ are preferred to be low-rank since they are aggregated to estimate a low-rank matrix $\hat{\boldsymbol{R}}$, which further leads $\boldsymbol{M}$ to be a low-rank matrix. λ_N and λ_M are the tradeoff parameters. The underlying matrix $\hat{\boldsymbol{R}}$ can be estimated by $\hat{\boldsymbol{X}}\boldsymbol{M}^*\hat{\boldsymbol{Y}}^\top + \boldsymbol{N}^*$, where $\boldsymbol{M}^*$ and $\boldsymbol{N}^*$ are the optimizers of (1). The effectiveness of this model for recovering the imperfect matrix has been theoretically and empirically verified.

Inspired by this work, we treat the noisy label removal problem as a label matrix recovery problem, and the example features are regarded as the descriptions of the row entities. Therefore, our LNSI inherits the good property of the model (1), and thus, the good performance is guaranteed.

IV. Our Proposed LNSI

In this section, we first describe our proposed model in Section IV-A and then provide the optimization algorithm in Section IV-B.

A. Model

We now introduce LNSI on label noise handling from the new viewpoint of side information. Let $\boldsymbol{Y} \in \mathbb{R}^{n \times c}$ be the observed label matrix with corrupted entries, where n is the number of examples and c is the number of classes. $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ (d denotes the feature dimensionality) is the feature matrix, where the $\boldsymbol{X}_i$ representing the ith row of $\boldsymbol{X}$ records the feature of the ith example. The element $\boldsymbol{Y}_{ij} = 1$ if the ith example has the label j ($j = 1, 2, \ldots, c$), and $\boldsymbol{Y}_{ij} = -1$ otherwise. As explained in Section I, the features in $\boldsymbol{X}$ are taken as the side information for recovering the correct label matrix in this article.

Specifically, we propose to decompose the observed $\boldsymbol{Y}$ into two parts, where the first one is the low-rank groundtruth label matrix $\boldsymbol{XZ}$ estimated by the projection $\boldsymbol{Z}$ on the feature matrix $\boldsymbol{X}$, and the second part is $\boldsymbol{E} \in \mathbb{R}^{n \times c}$ describing the difference between the observed labels and accurate labels. Here $\boldsymbol{Z} \in \mathbb{R}^{d \times c}$ is the projection matrix, and $\boldsymbol{E}$ can be used to capture the noisy labels. Since the "clean" label matrix $\boldsymbol{XZ}$ is low-rank, the projection matrix $\boldsymbol{Z}$ should be low-rank as well. This is because only a small fraction of columns in $\boldsymbol{X}$ are sufficient to construct the low-rank space $\boldsymbol{XZ}$. As $\boldsymbol{XZ}$ is the low-rank groundtruth label matrix, each entry in it should be -1 or 1. The label errors can be captured by a row-sparse matrix $\boldsymbol{E}$ since the label noise in training examples is usually sparse, and thus, the number of corresponding nonzero rows should be small. In addition, the Frobenius norm is imposed to $\boldsymbol{Z}$ to avoid overfitting. By putting all the above together, we consider solving the following problem:

$$\begin{aligned} &\min_{\boldsymbol{Z},\boldsymbol{E}} \ \|\boldsymbol{Z}\|_* + \lambda_1 \|\boldsymbol{Z}\|_F^2 + \lambda_3 \|\boldsymbol{E}\|_{2,1} \\ &\text{s.t. } \boldsymbol{Y} = \boldsymbol{XZ} + \boldsymbol{E}, \quad \boldsymbol{XZ} \in \{-1, 1\}^{n \times c} \end{aligned} \tag{2}$$

in which $\|\boldsymbol{Z}\|_*$ with nuclear norm is employed to achieve low-rank effect, $\|\boldsymbol{E}\|_{2,1}$ with $\ell_{2,1}$ norm is utilized to realize row sparsity, and λ_1 and λ_3 are the nonnegative tradeoff parameters. By solving (2), we can obtain the optimal $\boldsymbol{Z}^*$, which can be used to compute the label $\boldsymbol{y} \in \mathbb{R}^c$ of a test example $\boldsymbol{x} \in \mathbb{R}^d$ as $\boldsymbol{y} = \boldsymbol{Z}^{*\top}\boldsymbol{x}$. Then $\boldsymbol{x}$ is classified into the jth class if $j = \arg\max_{j'=1,2,\ldots,c} \boldsymbol{y}_{j'}$ with $\boldsymbol{y}_{j'}$ being the j'th element in the label vector $\boldsymbol{y}$.

It is worth pointing out that although the formulation of (2) is similar to the low-rank representation (LRR) proposed in [27], their usages and implications are quite different. First, LRR is developed for subspace clustering while our method is for label noise removal. Second, LRR aims to select sparse atoms from a predefined dictionary to reconstruct a clean space, while our method tries to learn a proper mapping $\boldsymbol{Z}$ from feature space to label space in presence of the label noise encoded by $\boldsymbol{E}$. Therefore, these two models are different although they look similar at first glance.

To sufficiently exploit the side information, we further use the Laplacian regularizer based on graph embedding. Graph Laplacian has been widely utilized in several machine learning tasks such as semisupervised learning [28], [29], spectral clustering [30], multi-task learning [31], and metric learning [32]. However, to the best of our knowledge, it has not been used to deal with side information. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be an undirected weighted graph with vertex set $\mathcal{V}$ consisting of all n examples and $\mathcal{E}$ is the edge set encoding the similarity between these examples. The symmetric adjacency matrix $\hat{\boldsymbol{W}} \in \mathbb{R}^{n\times n}$ is utilized to quantify the graph $\mathcal{G}$, where $\hat{\boldsymbol{W}}_{ij} = \exp(-(\|\boldsymbol{X}_i - \boldsymbol{X}_j\|^2/2\sigma_k^2))$ [33] (σ_k is the kernel width) measures the similarity between examples $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ ($i, j = 1, 2, \ldots, n$). The diagonal matrix $\boldsymbol{D}$ and the Laplacian matrix $\boldsymbol{L}$ of the graph $\mathcal{G}$ are, respectively, defined as

$$\boldsymbol{D}_{ii} = \sum_j \hat{\boldsymbol{W}}_{ij} \quad \boldsymbol{L} = \boldsymbol{D} - \hat{\boldsymbol{W}}. \tag{3}$$

Ideally, we hope that the similar examples revealed by $\mathcal{G}$ obtain similar clean labels, and the labels of dissimilar examples can be quite different. Therefore, we have the Laplacian regularizer that is derived as

$$\sum_i \sum_j \hat{\boldsymbol{W}}_{ij} \|\boldsymbol{X}_i\boldsymbol{Z} - \boldsymbol{X}_j\boldsymbol{Z}\|_2^2 = \mathrm{tr}((\boldsymbol{X}\boldsymbol{Z})^\top \boldsymbol{L}(\boldsymbol{X}\boldsymbol{Z})) \tag{4}$$

in which "tr$(\cdot)$" computes the trace of the corresponding matrix.

By combining (2) and (4), the proposed LNSI model is finally formulated as

$$\begin{aligned} &\min_{\boldsymbol{Z},\boldsymbol{E}} \ \|\boldsymbol{Z}\|_* + \lambda_1\|\boldsymbol{Z}\|_F^2 + \lambda_2 \mathrm{tr}((\boldsymbol{X}\boldsymbol{Z})^\top \boldsymbol{L}(\boldsymbol{X}\boldsymbol{Z})) + \lambda_3\|\boldsymbol{E}\|_{2,1} \\ &\text{s.t. } \boldsymbol{Y} = \boldsymbol{X}\boldsymbol{Z} + \boldsymbol{E}, \quad \boldsymbol{X}\boldsymbol{Z} \in \{-1, 1\}^{n\times c} \end{aligned} \tag{5}$$

in which λ_1, λ_2, and λ_3 are the nonnegative tradeoff parameters.

We note that (5) falls into an integer programming problem, which is generally NP-hard. To make problem (5) tractable, we relax the discrete constraint $\boldsymbol{X}\boldsymbol{Z} \in \{-1, 1\}^{n\times c}$ to a continuous convex set $\boldsymbol{X}\boldsymbol{Z} \in [-1, 1]^{n\times c}$. It is a linear programming relaxation, which has been used in several prior works [34], [35]. By doing so, we pursue to solve a simpler problem

$$\begin{aligned} &\min_{\boldsymbol{Z},\boldsymbol{E}} \ \|\boldsymbol{Z}\|_* + \lambda_1\|\boldsymbol{Z}\|_F^2 + \lambda_2 \mathrm{tr}((\boldsymbol{X}\boldsymbol{Z})^\top \boldsymbol{L}(\boldsymbol{X}\boldsymbol{Z})) + \lambda_3\|\boldsymbol{E}\|_{2,1} \\ &\text{s.t. } \boldsymbol{Y} = \boldsymbol{X}\boldsymbol{Z} + \boldsymbol{E}, \quad \boldsymbol{X}\boldsymbol{Z} \in [-1, 1]^{n\times c}. \end{aligned} \tag{6}$$

B. Optimization

Directly solving the problem (6) is difficult due to the existence of coupled variables, which will make its optimization not have a closed-form solution. Consequently, we introduce two auxiliary variables $\boldsymbol{J}$ and $\boldsymbol{B}$, and then, the problem (6) is converted to the following equivalent version:

$$\begin{aligned} &\min_{\boldsymbol{Z},\boldsymbol{E},\boldsymbol{J},\boldsymbol{B}} \ \|\boldsymbol{Z}\|_* + \lambda_1\|\boldsymbol{Z}\|_F^2 + \lambda_2 \mathrm{tr}((\boldsymbol{X}\boldsymbol{J})^\top \boldsymbol{L}(\boldsymbol{X}\boldsymbol{J})) + \lambda_3\|\boldsymbol{E}\|_{2,1} \\ &\text{s.t. } \boldsymbol{Y} = \boldsymbol{B} + \boldsymbol{E}, \boldsymbol{B} = \boldsymbol{X}\boldsymbol{J}, \quad \boldsymbol{Z} = \boldsymbol{J}, \ \boldsymbol{B} \in [-1, 1]^{n\times c}. \end{aligned} \tag{7}$$

The optimization problem (7) is convex and many off-the-shelf methods can be adopted to solve it. For efficiency, here we use the alternating direction method of multipliers (ADMMs), which alternatively optimizes the related variables in an iterative manner. The augmented Lagrangian function of (7) with the continuous convex constraint can be written as

$$\begin{aligned} &\mathcal{L}(\boldsymbol{Z}, \boldsymbol{E}, \boldsymbol{B}, \boldsymbol{J}, \boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3) \\ &= \|\boldsymbol{Z}\|_* + \lambda_1\|\boldsymbol{Z}\|_F^2 + \lambda_2 \mathrm{tr}((\boldsymbol{X}\boldsymbol{J})^\top \boldsymbol{L}(\boldsymbol{X}\boldsymbol{J})) + \lambda_3\|\boldsymbol{E}\|_{2,1} \\ &\quad + \mathrm{tr}\big(\boldsymbol{M}_1^\top(\boldsymbol{Y} - \boldsymbol{B} - \boldsymbol{E})\big) + \mathrm{tr}\big(\boldsymbol{M}_2^\top(\boldsymbol{B} - \boldsymbol{X}\boldsymbol{J})\big) \\ &\quad + \mathrm{tr}\big(\boldsymbol{M}_3^\top(\boldsymbol{Z} - \boldsymbol{J})\big) + \frac{\mu}{2}(\|\boldsymbol{Y} - \boldsymbol{B} - \boldsymbol{E}\|_F^2 + \|\boldsymbol{B} - \boldsymbol{X}\boldsymbol{J}\|_F^2 \\ &\qquad + \|\boldsymbol{Z} - \boldsymbol{J}\|_F^2) \end{aligned} \tag{8}$$

where $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, and $\boldsymbol{M}_3$ are the Lagrangian multipliers, and $\mu > 0$ is the penalty coefficient. We can sequentially minimize each of the variables $\boldsymbol{Z}$, $\boldsymbol{E}$, $\boldsymbol{B}$, and $\boldsymbol{J}$ by fixing the others in every iteration.

Update Z: The subproblem related to $\boldsymbol{Z}$ is

$$\begin{aligned} &\min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_* + \lambda_1\|\boldsymbol{Z}\|_F^2 + \mathrm{tr}\big(\boldsymbol{M}_3^\top(\boldsymbol{Z} - \boldsymbol{J})\big) + \frac{\mu}{2}\|\boldsymbol{Z} - \boldsymbol{J}\|_F^2 \\ &\Rightarrow \min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_* + \lambda_1\|\boldsymbol{Z}\|_F^2 + \mathrm{tr}\big(\boldsymbol{M}_3^\top\boldsymbol{Z} - \boldsymbol{M}_3^\top\boldsymbol{J}\big) \\ &\qquad + \frac{\mu}{2}\mathrm{tr}((\boldsymbol{Z}^\top - \boldsymbol{J}^\top)(\boldsymbol{Z} - \boldsymbol{J})) \\ &\Rightarrow \min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_* + \lambda_1\mathrm{tr}(\boldsymbol{Z}^\top\boldsymbol{Z}) + \mathrm{tr}\big(\boldsymbol{M}_3^\top\boldsymbol{Z}\big) \\ &\qquad + \frac{\mu}{2}\mathrm{tr}(\boldsymbol{Z}^\top\boldsymbol{Z} - 2\boldsymbol{J}^\top\boldsymbol{Z}) \\ &\Rightarrow \min_{\boldsymbol{Z}} \|\boldsymbol{Z}\|_* + \frac{2\lambda_1 + \mu}{2} \\ &\qquad \times \mathrm{tr}\left(\boldsymbol{Z}^\top\boldsymbol{Z} - \frac{2}{2\lambda_1 + \mu}(\mu\boldsymbol{J} - \boldsymbol{M}_3)^\top\boldsymbol{Z}\right) \\ &\Rightarrow \min_{\boldsymbol{Z}} \frac{1}{2\lambda_1 + \mu}\|\boldsymbol{Z}\|_* + \frac{1}{2}\left\|\boldsymbol{Z} - \frac{1}{\mu + 2\lambda_1}(\mu\boldsymbol{J} - \boldsymbol{M}_3)\right\|_F^2 \\ &\Rightarrow \min_{\boldsymbol{Z}} \tau\|\boldsymbol{Z}\|_* + \frac{1}{2}\|\boldsymbol{Z} - \hat{\boldsymbol{T}}\|_F^2 \end{aligned} \tag{9}$$

where $\hat{\boldsymbol{T}} = (1/(\mu + 2\lambda_1))(\mu\boldsymbol{J} - \boldsymbol{M}_3)$ and $\tau = (1/(2\lambda_1 + \mu))$.

According to [36], the closed-form solution to (9) can be expressed as

$$\boldsymbol{Z} = \boldsymbol{U}\text{diag}(\max\{\boldsymbol{\Sigma}_{ii} - \tau, 0\})\boldsymbol{V}^\top \quad \forall i = 1, 2, \ldots, \min(d, c) \tag{10}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are obtained by conducting the singular value decomposition (SVD) on $\hat{\boldsymbol{T}}$ (i.e., $\hat{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$), and $\boldsymbol{\Sigma}_{ii}$ is the ith diagonal element of the singular value matrix $\boldsymbol{\Sigma}$.

Update E: By dropping the unrelated terms to $\boldsymbol{E}$ in (8), the subproblem of $\boldsymbol{E}$ is

$$\begin{aligned}
&\min_{\boldsymbol{E}} \lambda_3\|\boldsymbol{E}\|_{2,1} + \text{tr}\big(\boldsymbol{M}_1^\top(\boldsymbol{Y} - \boldsymbol{B} - \boldsymbol{E})\big) + \frac{\mu}{2}\|\boldsymbol{Y} - \boldsymbol{B} - \boldsymbol{E}\|_F^2 \\
&\Rightarrow \min_{\boldsymbol{E}} \lambda_3\|\boldsymbol{E}\|_{2,1} - \text{tr}\big(\boldsymbol{M}_1^\top\boldsymbol{E}\big) \\
&\qquad + \frac{\mu}{2}\text{tr}(\boldsymbol{E}^\top\boldsymbol{E} - 2(\boldsymbol{Y} - \boldsymbol{B})^\top\boldsymbol{E}) \\
&\Rightarrow \min_{\boldsymbol{E}} \lambda_3\|\boldsymbol{E}\|_{2,1} \\
&\qquad + \frac{\mu}{2}\text{tr}\left(\boldsymbol{E}^\top\boldsymbol{E} - 2\left(\boldsymbol{Y} - \boldsymbol{B} + \frac{1}{\mu}\boldsymbol{M}_1\right)^\top\boldsymbol{E}\right) \\
&\Rightarrow \min_{\boldsymbol{E}} \frac{\lambda_3}{\mu}\|\boldsymbol{E}\|_{2,1} + \frac{1}{2}\left\|\boldsymbol{E} - \left(\boldsymbol{Y} - \boldsymbol{B} + \frac{\boldsymbol{M}_1}{\mu}\right)\right\|_F^2 \\
&\Rightarrow \min_{\boldsymbol{E}} \eta\|\boldsymbol{E}\|_{2,1} + \frac{1}{2}\|\boldsymbol{E} - \widetilde{\boldsymbol{M}}\|_F^2
\end{aligned} \tag{11}$$

where $\widetilde{\boldsymbol{M}} = \boldsymbol{Y} - \boldsymbol{B} + (\boldsymbol{M}_1/\mu)$ and $\eta = (\lambda_3/\mu)$.

Herein, the closed-form solution to the general optimization problem related to $\ell_{2,1}$ norm is provided in the following lemma.

Lemma 1 [27], [37]: Let $\tilde{\boldsymbol{Q}}$ be a given matrix and $\tilde{W}$ is the variable to be optimized. If the optimal solution to

$$\min_{\tilde{\boldsymbol{W}}} \tilde{\alpha}\|\tilde{\boldsymbol{W}}\|_{2,1} + \frac{1}{2}\|\tilde{\boldsymbol{W}} - \tilde{\boldsymbol{Q}}\|_F^2 \tag{12}$$

is $\tilde{\boldsymbol{W}}^*$, then the ith row of $\tilde{\boldsymbol{W}}^*$ is

$$\tilde{\boldsymbol{W}}_i^* = \begin{cases} \dfrac{\|\tilde{\boldsymbol{Q}}_i\|_2 - \tilde{\alpha}}{\|\tilde{\boldsymbol{Q}}_i\|_2}\tilde{\boldsymbol{Q}}_i, & \text{if } \|\tilde{\boldsymbol{Q}}_i\|_2 > \tilde{\alpha} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Obviously, the subproblem related to $\boldsymbol{E}$ has the same formulation with (13). Therefore, the closed-form solution to (11) is expressed as

$$\boldsymbol{E}_i = \begin{cases} \dfrac{\|\widetilde{\boldsymbol{M}}_i\|_2 - \eta}{\|\widetilde{\boldsymbol{M}}_i\|_2}\widetilde{\boldsymbol{M}}_i & \text{if } \|\widetilde{\boldsymbol{M}}_i\|_2 > \eta \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where $\boldsymbol{E}_i$ and $\widetilde{\boldsymbol{M}}_i$ represent the ith row of the related matrices, respectively.

Update J: The subproblem regarding $\boldsymbol{J}$ is

$$\begin{aligned}
&\min_{\boldsymbol{J}} \lambda_2\text{tr}((\boldsymbol{X}\boldsymbol{J})^\top\boldsymbol{L}(\boldsymbol{X}\boldsymbol{J})) + \text{tr}\big(\boldsymbol{M}_2^\top(\boldsymbol{B} - \boldsymbol{X}\boldsymbol{J})\big) \\
&\qquad + \text{tr}\big(\boldsymbol{M}_3^\top(\boldsymbol{Z} - \boldsymbol{J})\big) + \frac{\mu}{2}\big(\|\boldsymbol{B} - \boldsymbol{X}\boldsymbol{J}\|_F^2 + \|\boldsymbol{Z} - \boldsymbol{J}\|_F^2\big) \\
&\Rightarrow \min_{\boldsymbol{J}} \text{tr}\left(\lambda_2\boldsymbol{J}^\top(\boldsymbol{X}^\top\boldsymbol{L}\boldsymbol{X})\boldsymbol{J} + \frac{\mu}{2}(\boldsymbol{J}^\top(\boldsymbol{X}^\top\boldsymbol{X})\boldsymbol{J} + \boldsymbol{J}^\top\boldsymbol{J})\right) \\
&\qquad - \text{tr}\big(\boldsymbol{M}_2^\top\boldsymbol{X}\boldsymbol{J} + \boldsymbol{M}_3^\top\boldsymbol{J} + \mu(\boldsymbol{B}^\top\boldsymbol{X}\boldsymbol{J} + \boldsymbol{Z}^\top\boldsymbol{J})\big).
\end{aligned} \tag{15}$$

By computing the derivative of (15) with respect to $\boldsymbol{J}$ and then setting it to zero, $\boldsymbol{J}$ can be updated as

$$\begin{aligned}
\boldsymbol{J} = &(2\lambda_2\boldsymbol{X}^\top\boldsymbol{L}\boldsymbol{X} + \mu\boldsymbol{X}^\top\boldsymbol{X} + \mu\boldsymbol{I})^{-1} \\
&(\boldsymbol{X}^\top\boldsymbol{M}_2 + \boldsymbol{M}_3 + \mu\boldsymbol{X}^\top\boldsymbol{B} + \mu\boldsymbol{Z})
\end{aligned} \tag{16}$$

where $\boldsymbol{I}$ denotes the identity matrix with proper size throughout this article.

Update B: The subproblem on $\boldsymbol{B}$ with the continuous convex set $\boldsymbol{B} \in [-1, 1]^{n\times c}$ is

$$\begin{aligned}
&\min_{\boldsymbol{B}} \text{tr}\big(\boldsymbol{M}_1^\top(\boldsymbol{Y} - \boldsymbol{B} - \boldsymbol{E})\big) + \text{tr}\big(\boldsymbol{M}_2^\top(\boldsymbol{B} - \boldsymbol{X}\boldsymbol{J})\big) \\
&\qquad + \frac{\mu}{2}(\|\boldsymbol{Y} - \boldsymbol{B} - \boldsymbol{E}\|_F^2 + \|\boldsymbol{B} - \boldsymbol{X}\boldsymbol{J}\|_F^2) \\
&\text{s.t. } \boldsymbol{B} \in [-1, 1]^{n\times c}.
\end{aligned} \tag{17}$$

By first computing the derivative of (17) with respect to $\boldsymbol{B}$ and setting it to zero, the optimal $\boldsymbol{B}$ (i.e., $\hat{\boldsymbol{B}}$) can be represented as

$$\hat{\boldsymbol{B}} = \frac{\mu(\boldsymbol{Y} - \boldsymbol{E} + \boldsymbol{X}\boldsymbol{J}) + \boldsymbol{M}_1 - \boldsymbol{M}_2}{2\mu}. \tag{18}$$

To restrict $\hat{\boldsymbol{B}}$ to the feasible region, we further project all its elements to $[-1, 1]$ as

$$\boldsymbol{B}_{ij} = \Pi(\hat{\boldsymbol{B}}_{ij}) \tag{19}$$

where the projection $\Pi(x)$ is defined as

$$\Pi(x) = \begin{cases} 1, & \text{if } x > 1 \\ x, & \text{if } x \in [-1, 1] \\ -1, & \text{if } x < -1. \end{cases} \tag{20}$$

The entire optimization process for LNSI is summarized in Algorithm 1.

V. Theoretical Analyses

This section provides the theoretical analyses on LNSI. We first prove that the optimization process explained in Section IV-B will converge to a stationary point and then analyze the computational complexity of Algorithm 1. Finally, we theoretically prove that the generalization risk of LNSI is upper bounded.

A. Proof of Convergence

In this section, we discuss the convergence property of the ADMM method in Algorithm 1. As discussed in [38], the convergence of ADMM has been proved when there are only two blocks of variables. However, (7) contains four variables $\boldsymbol{Z}$, $\boldsymbol{E}$, $\boldsymbol{J}$, and $\boldsymbol{B}$, and thus, such a convergence property of ADMM is not theoretically guaranteed. By demonstrating that (7) is equivalent to the standard optimization problem with two variables, we show that the iterative solution provided in Algorithm 1 also enjoys the good property of convergence in Theorem 2.

Theorem 2: Given the optimization problem (7), the iterative process of ADMM will converge to a stationary point.

First, we provide the convergence conditions for general ADMM solver and then prove that the proposed algorithm satisfies the required conditions. Therefore, the iterative solution in Algorithm 1 is guaranteed to converge to a stationary point. The detailed proof can be found in the Appendix.

Algorithm 1 Algorithm for Solving LNSI

Input: feature matrix $\boldsymbol{X}$, observed label matrix $\boldsymbol{Y}$;
trade-off parameters: λ_1, λ_2, and λ_3;
$\boldsymbol{Z} = \boldsymbol{O}$, $\boldsymbol{J} = \boldsymbol{Z}$, $\boldsymbol{E} = \boldsymbol{O}$, $\boldsymbol{B} = \boldsymbol{O}$, $\boldsymbol{M}_1 = \boldsymbol{O}$, $\boldsymbol{M}_2 = \boldsymbol{O}$, $\boldsymbol{M}_3 = \boldsymbol{O}$; $\mu = 10^{-3}$, $\mu_{max} = 10^6$, $\rho = 1.2$, $\epsilon = 10^{-6}$, $iter\_max = 1000$; $iter = 0$;
1: Construct graph $\mathcal{G}$ and calculate the Laplacian matrix $\boldsymbol{L}$ via (3);
2: **while** not converge **do**
3: Update $\boldsymbol{Z}$ via (10),
4: Update $\boldsymbol{E}$ via (14),
5: Update $\boldsymbol{J}$ via (16),
6: Update $\boldsymbol{B}$ via (19),
7: Update the multipliers
$\boldsymbol{M}_1 := \boldsymbol{M}_1 + \mu(\boldsymbol{Y} - \boldsymbol{B} - \boldsymbol{E})$,
$\boldsymbol{M}_2 := \boldsymbol{M}_2 + \mu(\boldsymbol{B} - \boldsymbol{X}\boldsymbol{J})$,
$\boldsymbol{M}_3 := \boldsymbol{M}_3 + \mu(\boldsymbol{Z} - \boldsymbol{J})$,
8: Update the parameter μ by $\mu := min(\rho\mu, \mu_{max})$,
9: $iter := iter + 1$,
10: Check the convergence conditions:
$\|\boldsymbol{Y} - \boldsymbol{B} - \boldsymbol{E}\|_F \leq \epsilon$ and $\|\boldsymbol{B} - \boldsymbol{X}\boldsymbol{J}\|_F \leq \epsilon$ and $\|\boldsymbol{Z} - \boldsymbol{J}\|_F \leq \epsilon$; or $iter > iter\_max$.
11: **end while**
Output: optimized $\boldsymbol{Z}^*$ and $\boldsymbol{E}^*$.

B. Computational Complexity

This section studies the computational complexity of Algorithm 1. The graph construction in Line 1 of Algorithm 1 takes $\mathcal{O}(n^2)$ complexity. Line 3 is accomplished by using the SVD, of which the complexity is $\mathcal{O}(\min(d^2c, dc^2))$. In Line 4, one should compute the ℓ_2-norm of each row of a $n \times c$ matrix $\boldsymbol{E}$, so the complexity is $\mathcal{O}(nc)$. Note that a $d \times d$ matrix should be inverted in Line 5, so the complexity of this step is $\mathcal{O}(d^3)$. Therefore, the total complexity of our proposed algorithm is $\mathcal{O}(n^2 + (\min(d^2c, dc^2) + nc + d^3)k)$ by assuming that Lines 2–9 are iterated k times. Note that the complexity of Algorithm 1 is squared to the number of training examples n, so its complexity is acceptable.

C. Generalization Bound

In this section, we derive the generalization bound of LNSI.

1) Preliminaries: Recall that our goal is to find a suitable project matrix $\boldsymbol{Z}$ by recovering the clean label matrix $\boldsymbol{X}\boldsymbol{Z}$, given the observed noisy label matrix $\boldsymbol{Y}$ and example features $\boldsymbol{X}$. Similar to [10], (6) can be reformulated to the following expression with hard constraints, namely:

$$\begin{aligned} \min_{\boldsymbol{Z},\boldsymbol{E}} \quad & \sum_{(i,j)\in\{1,\ldots,n\}\times\{1,\ldots,c\}} \ell((\boldsymbol{X}\boldsymbol{Z} + \boldsymbol{E})_{ij}, \boldsymbol{Y}_{ij}) \\ \text{s.t. } & \|\boldsymbol{Z}\|_* \leq \mathcal{Z}_*, \|\boldsymbol{Z}\|_F^2 \leq \mathcal{Z}_F, \quad \boldsymbol{X}\boldsymbol{Z} \in [-1, 1]^{n\times c} \\ & \operatorname{tr}((\boldsymbol{X}\boldsymbol{Z})^\top \boldsymbol{L}(\boldsymbol{X}\boldsymbol{Z})) \leq \mathcal{Z}_{\text{tr}}, \quad \|\boldsymbol{E}\|_{2,1} \leq \mathcal{E}_{2,1}. \end{aligned} \tag{21}$$

Let $\theta = (\boldsymbol{Z}, \boldsymbol{E})$ be any feasible solution, and $\Theta = \{(\boldsymbol{Z}, \boldsymbol{E}) \mid \|\boldsymbol{Z}\|_* \leq \mathcal{Z}_*, \|\boldsymbol{Z}\|_F \leq \sqrt{\mathcal{Z}_F}, \boldsymbol{X}\boldsymbol{Z} \in [-1, 1]^{n\times c}, \operatorname{tr}((\boldsymbol{X}\boldsymbol{Z})^\top \boldsymbol{L}(\boldsymbol{X}\boldsymbol{Z})) \leq \mathcal{Z}_{\text{tr}}, \|\boldsymbol{E}\|_{2,1} \leq \mathcal{E}_{2,1}\}$ be the feasible solution set. Also, let $f_\theta(i, j) = \boldsymbol{X}_i \boldsymbol{Z} \boldsymbol{I}^j + \boldsymbol{E}_{ij}$ be the estimation function for $\boldsymbol{Y}_{ij}$ parameterized by $\theta = (\boldsymbol{Z}, \boldsymbol{E})$, and $\mathcal{F}_\Theta = \{f_\theta \mid \theta \in \Theta\}$ be the set of feasible functions. $\boldsymbol{I}^j$ is the jth column of identity matrix $\boldsymbol{I} \in \mathbb{R}^{c\times c}$. We are interested in the following two "ℓ-risk" quantities:

1) expected ℓ-risk: $R_\ell(f) = \mathbb{E}_{i,j}[\ell(f(i, j), \boldsymbol{Y}_{ij})]$;
2) empirical ℓ-risk: $\hat{R}_\ell(f) = (1/n_r) \sum_{(i,j)} \ell(f(i, j)\boldsymbol{Y}_{ij})$,

where n_r is the number of observed entries. Thus, LNSI is to find a proper $\theta^* = (\boldsymbol{Z}^*, \boldsymbol{E}^*)$ that parameterizes $f^* = \arg\min_{f\in\mathcal{F}_\Theta} \hat{R}_\ell(f)$.

2) Generalization Bound of LNSI: To bound the generalization error of LNSI, we first link the quality of training labels to Rademacher complexity, which theoretically measures the complexity of a function class. We will show that high-quality labels of training examples will result in a lower model complexity and thus a smaller error bound.

To begin with, we apply the following lemma to bound the expected ℓ-risk.

Lemma 3 (Bound on Expected ℓ-Risk [39]): Let ℓ be the loss function bounded by $\mathcal{B}$ with Lipschitz constant L_ℓ, and δ be a constant where $0 < \delta < 1$. With probability at least $1 - \delta$, we have

$$\max_{f\in\mathcal{F}} |R_\ell(f) - \hat{R}_\ell(f)| \leq 2L_\ell \mathcal{R}_n(\mathcal{F}) + \mathcal{B}\sqrt{\frac{\ln(1/\delta)}{2n_r}}$$

where

$$\mathcal{R}_n(\mathcal{F}) := \mathbb{E}[\mathcal{R}(\mathcal{F})]$$

is the Rademacher complexity of the function class $\mathcal{F}$ and

$$\mathcal{R}(\mathcal{F}) := \mathbb{E}_\sigma \left[\sup_{f\in\mathcal{F}} \frac{1}{n_r} \sum_{\alpha=1}^{n_r} \sigma_\alpha f(\alpha) \right]$$

is the empirical Rademacher complexity on the training examples. Note that σ_α $(\alpha = 1, 2, \ldots, n_r)$ are independent identically distributed (i.i.d.) Rademacher random variables.

Given Lemma 3, we see that the key to derive the upper bound of a function $f \in \mathcal{F}$ is to bound the complexity $\mathcal{R}_n(\mathcal{F}_\Theta)$. More formally, the Rademacher complexity can be bounded in terms of the constraints in (21). Before diving into the details, we first provide several useful theorems and lemmas.

Lemma 4 (Complexity Bound [40]): Let S be a closed convex set and let $F : S \to \mathbb{R}$ be β-strongly convex with respect to $\|\cdot\|$. In addition, we assume that $F^\star(\boldsymbol{O}) = 0$ with $F^\star$ being the Fenchel conjugate of function F. Further, let $\mathcal{A} = \{\boldsymbol{A} : \|\boldsymbol{A}\|_\star \leq A\}$ and define $\mathcal{W} = \{\boldsymbol{W} \in S : F(\boldsymbol{W}) \leq F_{\max}\}$. Considering the class of linear functions $\mathcal{F} = \{\boldsymbol{A} \to \langle \boldsymbol{W}, \boldsymbol{A}\rangle : \boldsymbol{W} \in \mathcal{W}\}$, we have

$$\mathcal{R}(\mathcal{F}) \leq A\sqrt{\frac{2F_{\max}}{\beta n_r}} \tag{22}$$

where $\langle \boldsymbol{W}, \boldsymbol{A}\rangle = \operatorname{tr}(\boldsymbol{W}^\top \boldsymbol{A})$.

Lemma 5 [41]: The function $F : \mathbb{R}^{n\times c} \to \mathbb{R}$ defined as $F(\boldsymbol{W}) = (1/2)\|\boldsymbol{W}\|_{2,q}^2$ for $q = (\ln(c)/(\ln(c) - 1))$ is $(1/(3\ln(c)))$-strongly convex with respect to $\|\cdot\|_{2,1}$ over $\mathbb{R}^{n\times c}$.

Lemma 6 [40]: The function $F : \mathbb{R}^{d\times c} \to \mathbb{R}$ defined as $F(\boldsymbol{W}) = (1/2)\|\boldsymbol{W}\|_{2,2}^2$ is $(1/2)$-strongly convex with respect to $\|\cdot\|_{2,2}$ over $\mathbb{R}^{d\times c}$, where $\|\cdot\|_{2,2} := \|\cdot\|_F$.

By combining Lemmas 5 and 6 with the bound given in Lemma 4, we obtain the following two corollaries.

Corollary 7: Let $\mathcal{W} = \{\boldsymbol{W} : \|\boldsymbol{W}\|_{2,1} \le \mathcal{W}_{2,1}\}$ and $\mathcal{A} = \{\boldsymbol{A} \in \mathbb{R}^{n\times c} : \|\boldsymbol{A}\|_{2,\infty} \le \mathcal{A}_{2,\infty}\}$, and then the empirical Rademacher complexity of the function class with $F(\boldsymbol{W}) = (1/2)\|\boldsymbol{W}\|_{2,q}^2$ for $q = (\ln(c)/(\ln(c)-1))$ is bounded as

$$\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\frac{1}{n_r}\sum_{\alpha=1}^{n_r}\sigma_\alpha \mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{A}^{(\alpha)})\right] \le \mathcal{W}_{2,1}\mathcal{A}_{2,\infty}\sqrt{\frac{3\ln(c)}{n_r}} \tag{23}$$

with the fact that the dual norm of $\ell_{2,1}$ is $\ell_{2,\infty}$.

Corollary 8: Let $\mathcal{W} = \{\boldsymbol{W} : \|\boldsymbol{W}\|_F \le \mathcal{W}_F\}$ and $\mathcal{A} = \{\boldsymbol{A} \in \mathbb{R}^{d\times c} : \|\boldsymbol{A}\|_F \le \mathcal{A}_F\}$, and then the empirical Rademacher complexity of the function class with $F(\boldsymbol{W}) = (1/2)\|\boldsymbol{W}\|_{2,2}^2$ is bounded as

$$\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\frac{1}{n_r}\sum_{\alpha=1}^{n_r}\sigma_\alpha \mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{A}^{(\alpha)})\right] \le \mathcal{W}_F\mathcal{A}_F\sqrt{\frac{2}{n_r}} \tag{24}$$

with the fact that the dual norm of the Frobenius norm is the Frobenius norm.

Lemma 9 [10]: Let $\mathcal{W} = \{\boldsymbol{W} : \|\boldsymbol{W}\|_* \le \mathcal{W}_*\}$ and $\mathcal{A} = \{\boldsymbol{A} \in \mathbb{R}^{d\times c} : \|\boldsymbol{A}\|_2 \le \mathcal{A}_2\}$, and then the empirical Rademacher complexity of the function class with $F(\boldsymbol{W}) = (1/2)\|\boldsymbol{W}\|_*^2$ is bounded as

$$\mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\frac{1}{n_r}\sum_{\alpha=1}^{n_r}\sigma_\alpha \mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{A}^{(\alpha)})\right] \le \mathcal{W}_*\mathcal{A}_2\sqrt{\frac{\ln(2d_c)}{n_r}} \tag{25}$$

with the fact that the dual norm of the nuclear norm is the spectral norm and $d_c = \max(d, c)$.

Lemma 10: Let $\boldsymbol{A}_{m_1} \in \mathbb{R}^{m_1\times n_1}$ and $\boldsymbol{A}_{m_2} \in \mathbb{R}^{n_1\times m_2}$ be two matrices, and then the following inequality holds:

$$\lambda_{\min}\big(\boldsymbol{A}_{m_1}^\top \boldsymbol{A}_{m_1}\big)\|\boldsymbol{A}_{m_2}\|_F^2 \le \|\boldsymbol{A}_{m_1}\boldsymbol{A}_{m_2}\|_F^2 \tag{26}$$

where $\lambda_{\min}(\cdot)$ represents the smallest eigenvalue of the matrix inside the bracket.

Proof: For a given complex Hermitian matrix $\boldsymbol{M}_r$ and nonzero vector $\boldsymbol{x}_r$, the Rayleigh quotient $R(\boldsymbol{M}_r, \boldsymbol{x}_r)$ [42], [43] is defined as

$$R(\boldsymbol{M}_r, \boldsymbol{x}_r) = \frac{\boldsymbol{x}_r^*\boldsymbol{M}_r\boldsymbol{x}_r}{\boldsymbol{x}_r^*\boldsymbol{x}_r} \tag{27}$$

and the following inequality holds:

$$\lambda_{\min}(\boldsymbol{M}_r) \le R(\boldsymbol{M}_r, \boldsymbol{x}_r) \le \lambda_{\max}(\boldsymbol{M}_r). \tag{28}$$

Let $\boldsymbol{a} = \mathrm{Vec}(\boldsymbol{A}_{m_2})$, where $\mathrm{Vec}(\cdot)$ is an operator converting a matrix into a vector, and then, $\|\boldsymbol{A}_{m_1}\boldsymbol{A}_{m_2}\|_F^2$ can be rewritten in the form of ℓ_2-norm as

$$\begin{aligned}\|\boldsymbol{A}_{m_1}\boldsymbol{A}_{m_2}\|_F^2 &= \|(\boldsymbol{I}\otimes\boldsymbol{A}_{m_1})\boldsymbol{a}\|_2^2\\ &= \boldsymbol{a}^\top(\boldsymbol{I}\otimes\boldsymbol{A}_{m_1})^\top(\boldsymbol{I}\otimes\boldsymbol{A}_{m_1})\boldsymbol{a}\\ &= \boldsymbol{a}^\top\big(\boldsymbol{I}\otimes\boldsymbol{A}_{m_1}^\top\big)(\boldsymbol{I}\otimes\boldsymbol{A}_{m_1})\boldsymbol{a}\\ &= \boldsymbol{a}^\top\big(\boldsymbol{I}\otimes\boldsymbol{A}_{m_1}^\top\boldsymbol{A}_{m_1}\big)\boldsymbol{a}\end{aligned} \tag{29}$$

where $\otimes$ represents the Kronecker product [44] and $\boldsymbol{I}$ is an identity matrix with proper size.

Combining (28) with (29), the following inequality holds, namely:

$$\begin{aligned}\lambda_{\min}\big(\boldsymbol{A}_{m_1}^\top\boldsymbol{A}_{m_1}\big)\|\boldsymbol{A}_{m_2}\|_F^2 &= \lambda_{\min}\big(\boldsymbol{I}\otimes\boldsymbol{A}_{m_1}^\top\boldsymbol{A}_{m_1}\big)\|\boldsymbol{A}_{m_2}\|_F^2\\ &= \lambda_{\min}\big(\boldsymbol{I}\otimes\boldsymbol{A}_{m_1}^\top\boldsymbol{A}_{m_1}\big)(\boldsymbol{a}^\top\boldsymbol{a})\\ &\le \boldsymbol{a}^\top\big(\boldsymbol{I}\otimes\boldsymbol{A}_{m_1}^\top\boldsymbol{A}_{m_1}\big)\boldsymbol{a}\\ &= \|\boldsymbol{A}_{m_1}\boldsymbol{A}_{m_2}\|_F^2\end{aligned} \tag{30}$$

which completes the proof. □

Provided Lemma 10, $\mathrm{tr}((\boldsymbol{XZ})^\top\boldsymbol{L}(\boldsymbol{XZ})) \le \mathcal{Z}_{\mathrm{tr}}$ in (21) can be derived into a constraint in the form of Frobenius norm on $\boldsymbol{Z}$. Specifically, note that the Laplacian matrix is positive semidefine and its SVD is $\boldsymbol{L} = \boldsymbol{U}_L\boldsymbol{\Sigma}_L\boldsymbol{U}_L^\top = \boldsymbol{U}_L\boldsymbol{\Sigma}_L^{(1/2)}\boldsymbol{\Sigma}_L^{(1/2)}\boldsymbol{U}_L^\top$, so we have

$$\begin{aligned}\mathrm{tr}((\boldsymbol{XZ})^\top\boldsymbol{L}(\boldsymbol{XZ})) &= \mathrm{tr}\big((\boldsymbol{XZ})^\top\boldsymbol{U}_L\boldsymbol{\Sigma}_L^{\frac{1}{2}}\boldsymbol{\Sigma}_L^{\frac{1}{2}}\boldsymbol{U}_L^\top(\boldsymbol{XZ})\big)\\ &= \mathrm{tr}\Big(\Big(\boldsymbol{\Sigma}_L^{\frac{1}{2}}\boldsymbol{U}_L^\top\boldsymbol{XZ}\Big)^\top\Big(\boldsymbol{\Sigma}_L^{\frac{1}{2}}\boldsymbol{U}_L^\top\boldsymbol{XZ}\Big)\Big)\\ &= \Big\|\boldsymbol{\Sigma}_L^{\frac{1}{2}}\boldsymbol{U}_L^\top\boldsymbol{XZ}\Big\|_F^2\\ &\ge \lambda_{\min}(\boldsymbol{X}^\top\boldsymbol{LX})\|\boldsymbol{Z}\|_F^2.\end{aligned} \tag{31}$$

If $\lambda_{\min_{\mathrm{tr}}} = \lambda_{\min}(\boldsymbol{X}^\top\boldsymbol{LX}) > 0$, we have $\|\boldsymbol{Z}\|_F \le \mathcal{Z}_t$, where $\mathcal{Z}_t = (\mathcal{Z}_{\mathrm{tr}}/\lambda_{\min_{\mathrm{tr}}})^{1/2}$.

In addition, we manage to rewrite the constraint $\boldsymbol{XZ} \in [-1, 1]^{n\times c}$ as the form of Frobenius norm on $\boldsymbol{Z}$. Since $\boldsymbol{XZ} \in [-1, 1]^{n\times c}$ and $\|\boldsymbol{XZ}\|_F^2 \le nc$, then similar to (31), we may obtain that $\|\boldsymbol{Z}\|_F \le \mathcal{Z}_b$ if $\lambda_{\min_b} = \lambda_{\min}(\boldsymbol{X}^\top\boldsymbol{X}) > 0$, where $\mathcal{Z}_b = (nc/\lambda_{\min_b})^{1/2}$.

Taking the three different Frobenius norm-based constraints (i.e., $\|\boldsymbol{Z}\|_F \le \sqrt{\mathcal{Z}_F}$, $\|\boldsymbol{Z}\|_F \le \mathcal{Z}_t$, $\|\boldsymbol{Z}\|_F \le \mathcal{Z}_b$) on the matrix $\boldsymbol{Z}$ into account, and let $\mathcal{Z}_{\min} = \min\{\sqrt{\mathcal{Z}_F}, \mathcal{Z}_b, \mathcal{Z}_t\}$, we have

$$\|\boldsymbol{Z}\|_F \le \begin{cases}\sqrt{\mathcal{Z}_F}, & \text{if } \lambda_{\min_{\mathrm{tr}}} \le 0 \text{ or } \lambda_{\min_b} \le 0\\ \mathcal{Z}_{\min}, & \text{otherwise.}\end{cases} \tag{32}$$

For convenience, we denote the upper bound of $\|\boldsymbol{Z}\|_F$ as $\mathcal{Z}$ that can be $\sqrt{\mathcal{Z}_F}$ or $\mathcal{Z}_{\min}$ according to different conditions in (32).

Herein, we begin to formally derive the generalization error of LNSI.

Theorem 11: Let $\mathcal{X}_2 = \max_i\|\boldsymbol{X}_i\|_2$ and $\mathcal{X}_F = \max_i\|\boldsymbol{X}_i\|_F$, and then the model complexity of function class $\mathcal{F}_\Theta$ is upper bounded by

$$\mathcal{R}_n(\mathcal{F}_\Theta) \le \min\left\{\mathcal{Z}_*\mathcal{X}_2\sqrt{\frac{\ln(2d_c)}{nc}}, \mathcal{Z}\mathcal{X}_F\sqrt{\frac{2}{nc}}\right\} + \mathcal{E}_{2,1}\sqrt{\frac{3\ln(c)}{nc}}. \tag{33}$$

Proof: First, the Rademacher complexity of linear function class $\mathcal{F}_\Theta$ in our case can be written as

$$\mathcal{R}(\mathcal{F}_\Theta) := \mathbb{E}_\sigma\left[\sup_{\theta\in\Theta}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha\big(\boldsymbol{X}_{i_\alpha}\boldsymbol{Z}\boldsymbol{I}^{j_\alpha} + \boldsymbol{E}_{i_\alpha,j_\alpha}\big)\right]. \tag{34}$$

Since $\boldsymbol{Z}$ and $\boldsymbol{E}$ are independent variables, the Rademacher complexity can be written as

$$\begin{aligned}\mathcal{R}(\mathcal{F}_\Theta) &= \mathbb{E}_\sigma\left[\sup_{\boldsymbol{Z}\in\Theta_Z}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha \boldsymbol{X}_{i_\alpha}\boldsymbol{Z}\boldsymbol{I}^{j_\alpha}\right] \\ &\quad+\mathbb{E}_\sigma\left[\sup_{\boldsymbol{E}\in\Theta_E}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha \boldsymbol{E}_{i_\alpha,j_\alpha}\right] \\ &= \mathbb{E}_\sigma\left[\sup_{\boldsymbol{Z}\in\Theta_Z}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha \mathrm{tr}\big(\boldsymbol{Z}^\top \boldsymbol{X}_{i_\alpha}^\top \boldsymbol{I}^{j_\alpha\top}\big)\right] \\ &\quad+\mathbb{E}_\sigma\left[\sup_{\boldsymbol{E}\in\Theta_E}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha \mathrm{tr}(\boldsymbol{E}^\top \boldsymbol{I}^{i_\alpha}\boldsymbol{I}^{j_\alpha\top})\right] \qquad (35)\end{aligned}$$

where $\boldsymbol{I}^i$ is the ith column of identity matrix $\boldsymbol{I}\in\mathbb{R}^{n\times n}$, $\Theta_Z=\{\boldsymbol{Z}\mid \|\boldsymbol{Z}\|_*\le \mathcal{Z}_*, \|\boldsymbol{Z}\|_F\le\sqrt{\mathcal{Z}_F}, \boldsymbol{X}\boldsymbol{Z}\in[-1,1]^{n\times c}, \mathrm{tr}((\boldsymbol{X}\boldsymbol{Z})^\top\boldsymbol{L}(\boldsymbol{X}\boldsymbol{Z}))\le\mathcal{Z}_{\mathrm{tr}}\}$, and $\Theta_E=\{\boldsymbol{E}\mid \|\boldsymbol{E}\|_{2,1}\le\mathcal{E}_{2,1}\}$.

Since the last three constraints in Θ_Z can be rewritten as a much simpler formulation (32) according to Lemma 10, therefore, we have $\Theta_{Z_1}=\{\boldsymbol{Z}\mid \|\boldsymbol{Z}\|_*\le\mathcal{Z}_*, \|\boldsymbol{Z}\|_F\le\mathcal{Z}\}$.

From the definition of Rademacher complexity, we know that it measures the richness of a class of real-valued functions with respect to a probability distribution. Hence, the tighter the constraints of functions are, the smaller the Rademacher complexity will be. By using the Rademacher contraction principle [45], (35) can be transformed as

$$\begin{aligned}\mathcal{R}(\mathcal{F}_\Theta) &\le \mathbb{E}_\sigma\left[\sup_{\boldsymbol{Z}\in\Theta_{Z_1}}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha \mathrm{tr}\big(\boldsymbol{Z}^\top \boldsymbol{X}_{i_\alpha}^\top \boldsymbol{I}^{j_\alpha\top}\big)\right] \\ &\quad+\mathbb{E}_\sigma\left[\sup_{\boldsymbol{E}\in\Theta_E}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha \mathrm{tr}(\boldsymbol{E}^\top \boldsymbol{I}^{i_\alpha}\boldsymbol{I}^{j_\alpha\top})\right] \\ &\le \min\left\{\mathbb{E}_\sigma\left[\sup_{\boldsymbol{Z}\in\Theta_{Z_2}}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha \mathrm{tr}\big(\boldsymbol{Z}^\top \boldsymbol{X}_{i_\alpha}^\top \boldsymbol{I}^{j_\alpha\top}\big)\right]\right. \\ &\qquad\left.\mathbb{E}_\sigma\left[\sup_{\boldsymbol{Z}\in\Theta_{Z_3}}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha \mathrm{tr}\big(\boldsymbol{Z}^\top \boldsymbol{X}_{i_\alpha}^\top \boldsymbol{I}^{j_\alpha\top}\big)\right]\right\} \\ &\quad+\mathbb{E}_\sigma\left[\sup_{\boldsymbol{E}\in\Theta_E}\frac{1}{nc}\sum_{\alpha=1}^{nc}\sigma_\alpha \mathrm{tr}(\boldsymbol{E}^\top \boldsymbol{I}^{i_\alpha}\boldsymbol{I}^{j_\alpha\top})\right] \qquad (36)\end{aligned}$$

where $\Theta_{Z_2}=\{\boldsymbol{Z}\mid\|\boldsymbol{Z}\|_*\le\mathcal{Z}_*\}$ and $\Theta_{Z_3}=\{\boldsymbol{Z}\mid\|\boldsymbol{Z}\|_F\le\mathcal{Z}\}$.

Taking Corollaries 7 and 8 and Lemma 9 into consideration, (36) leads to

$$\begin{aligned}\mathcal{R}(\mathcal{F}_\Theta) &\le \mathcal{E}_{2,1}\|\boldsymbol{I}^i\boldsymbol{I}^{j\top}\|_{2,\infty}\sqrt{\frac{3\ln(c)}{nc}} \\ &\quad+\min\left\{\mathcal{Z}_*\max_{i,j}\left\|\boldsymbol{X}_i^\top\boldsymbol{I}^{j\top}\right\|_2\sqrt{\frac{\ln(2d_c)}{nc}}\right. \\ &\qquad\left.\mathcal{Z}\max_{i,j}\left\|\boldsymbol{X}_i^\top\boldsymbol{I}^{j\top}\right\|_F\sqrt{\frac{2}{nc}}\right\}. \qquad (37)\end{aligned}$$

Since $\max_{i,j}\|\boldsymbol{X}_i^\top\boldsymbol{I}^{j\top}\|_2=\max_i\|\boldsymbol{X}_i\|_2\max_j\|\boldsymbol{I}^j\|_2=\max_i\|\boldsymbol{X}_i\|_2$, $\max_{i,j}\|\boldsymbol{X}_i^\top\boldsymbol{I}^{j\top}\|_F=\max_i\|\boldsymbol{X}_i\|_F$ and $\max_{i,j}\|\boldsymbol{I}^i\boldsymbol{I}^{j\top}\|_{2,\infty}=1$, we arrive at the upper bound of $\mathcal{R}_n(\mathcal{F}_\Theta)$, which is

$$\begin{aligned}\mathcal{R}_n(\mathcal{F}_\Theta) &\le \min\left\{\mathcal{Z}_*\mathcal{X}_2\sqrt{\frac{\ln(2d_c)}{nc}}, \mathcal{Z}\mathcal{X}_F\sqrt{\frac{2}{nc}}\right\} \\ &\quad+\mathcal{E}_{2,1}\sqrt{\frac{3\ln(c)}{nc}}. \qquad (38)\end{aligned}$$

□

Based on Theorem 11 and Lemma 3, the following inequality holds:

$$\begin{aligned}&\max_{f\in\mathcal{F}}\ |R_\ell(f)-\hat{R}_\ell(f)| \\ &\quad\le 2L_\ell\min\left\{\mathcal{Z}_*\mathcal{X}_2\sqrt{\frac{\ln(2d_c)}{nc}}, \mathcal{Z}\mathcal{X}_F\sqrt{\frac{2}{nc}}\right\} \\ &\qquad+2L_\ell\mathcal{E}_{2,1}\sqrt{\frac{3\ln(c)}{nc}}+\mathcal{B}\sqrt{\frac{ln(1/\delta)}{2nc}} \qquad (39)\end{aligned}$$

and thus the expected loss is upper bounded. As mentioned before, the matrix $\boldsymbol{E}$ is utilized to capture the label noise and the $\ell_{2,1}$ norm on it is upper bounded by $\mathcal{E}_{2,1}$. Specifically, we observe that $\mathcal{E}_{2,1}$ is governed by the label noise rate. A small noise rate will lead to a small $\mathcal{E}_{2,1}$, which further reduces the upper bound of the expected ℓ-risk in the right-hand side of (39).

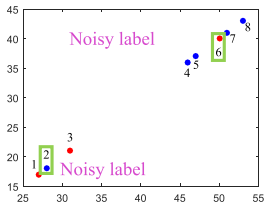
VI. Experiments

In this section, we first use a toy data set to validate the motivation of LNSI (Section VI-A) and then compare LNSI with several representative baselines on various UCI benchmark data sets (Section VI-B). Next, we compare LNSI with these baseline algorithms on four practical data sets including *ISOLET*, *COIL20*, *MNIST*, and *CIFAR-10* (Sections VI-C–VI-F). Also, the convergence process of the ADMM adopted in the Algorithm 1 is illustrated (Section VI-G). Afterward, we conduct the ablation study to verify the effectiveness of the critical regularizers of LNSI (Section VI-H) and also study the parametric sensitivity of LNSI (Section VI-I). Finally, we summarize the experimental results and give some insightful analyses (Section VI-J).

Our LNSI is compared with five representative methods for noisy label handling, including labeled instance centroid smoothing (LICS) [3], unbiased logistic estimator (ULE) [4], μ stochastic gradient descent (μSGD) [6], learning with symmetric label noise (LSLN) [15], and coteaching [18]. It is worth noting that the first four baseline methods are only suitable for binary classification, so we use the one-versus-the-rest strategy to make them applicable to multi-class situations. In addition, the prior knowledge such as the noise rate for all the compared methods is provided accurately. Note that coteaching can only handle images, so it is not compared on the nonimage data sets including five UCI data sets and *ISOLET* data set.

A. Algorithm Validation

First, we demonstrate the effectiveness of LNSI on a toy data set. As shown in Fig. 2(a), we manually generate 8 examples in a 2-D space, which include 3 negative examples



(a)

$$\boldsymbol{E} = \begin{bmatrix} 0 & 0 \\ -1.9680 & 1.9680 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1.9677 & -1.9677 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{matrix}$$

(b)

Fig. 2. Algorithm validation on the synthetic data set. (a) Initial state with labeled negative examples (in red) and labeled positive examples (in blue). Note that two examples are labeled incorrectly and they form the label noise. (b) Error matrix $\boldsymbol{E}$, in which the nonzero rows capture the mislabeled examples successfully.

TABLE I
STATISTICS OF THE BENCHMARK DATA SETS

Dataset	# Classes	# Features	# Examples
CNAE9	9	856	1080
Wine	3	13	178
Breast Tissue	6	9	106
Pendigits	10	16	10992
Connect-4	3	126	67557

(Nos. 1–3) and 5 positive examples (Nos. 4–8). The negative examples are denoted by red dots, while the positive examples are represented by blue dots. Note that the 2nd example is mistakenly labeled as positive and the 6th example is erroneously labeled as negative, so they form the label noise in this data set. As explained in Section I, the observed label matrix $\boldsymbol{Y}$ is decomposed as two parts, one is the groundtruth label matrix $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{Z}$ and the other is the error capturing matrix $\boldsymbol{E}$. After applying LNSI to this synthetic data set, we investigate whether the matrix $\boldsymbol{E}$ can accurately detect the mislabeled examples. In Fig. 2(b), we see that all rows in $\boldsymbol{E}$ are zeros except the 2nd and 6th rows that exactly correspond to the examples with label noise. This implies that the matrix $\boldsymbol{E}$ is able to capture the label errors successfully and also indicates that a suitable projection matrix $\boldsymbol{Z}$ has been learned.

B. Experiments on Benchmark Data Set

In this section, we compare LNSI with LICS, ULE, μSGD, and LSLN on five UCI benchmark data sets[1] including *CNAE9*, *Wine*, *Breast Tissue*, *Pendigits*, and *Connect-4*. The information about the data sets is summarized in Table I. For each data set, all the algorithms are tested at different levels (0%, 20%, 40%, and 60%) of label noise on training sets. Given the noise rate ζ, we manually inject the label noise by randomly picking up $\zeta \times n$ training examples and then switching the correct label of each of them to a random wrong label, in which n is the number of training examples. All the reported accuracies are the mean values of the outputs of five independent runs.

Generally, the number of the nearest neighbors in the graph is suggested to set to a small value, as it has been widely observed that a sparse graph can usually lead to good performance. The parameter σ_k controls the connective strength of pairwise examples, and it is chosen below 1 as all features have been normalized. The number of the nearest neighbors in the graph was selected by searching the grid {5, 10, 15, 20}. Similarly, the kernel width σ_k was also turned by searching the grid {0.01, 0.1, 0.5, 1}.

We established the 5-NN graph for LNSI on *Wine* and *Breast Tissue* and the 10-NN graph on *CNAE9*, *Pendigits* and *Connect-4*. The kernel width σ_k on *Connect-4* was 1 and on the other four data sets was chosen as 0.1. The tradeoff parameters λ_1, λ_2, and λ_3 were selected by searching the grid $\{10^{-4}, 10^{-3}, \ldots, 10^{5}\}$ on *Wine* and *Breast Tissue*, and the grid $\{10^{-2}, 10^{-1}, \ldots, 10^{3}\}$ on *CNAE9*, *Pendigits*, and *Connect-4*. The classification accuracies of all the compared methods are shown in Fig. 3. Note that LICS and LSLN are not compared on the *Connect-4* data set as they are not scalable to this data set.

From Fig. 3, we observe that LNSI yields better performance than other baselines in most cases. An exceptional case is that ULE and LSLN obtain the best results on *Breast Tissue* and *Pendigits* under the noise level 0%, respectively. Besides, we note that ULE performs satisfactorily on *Wine* under the noise level 60%. However, LNSI is generally the most robust method compared with all the other baselines. Therefore, formulating the classification task under label noise as a matrix recovery problem does help to improve the robustness of the trained classifier.

C. Experiments on ISOLET Data Set

To demonstrate the superiority of LNSI in dealing with different kinds of practical problems, we first use the *ISOLET*[2] data set to test the ability of all compared methods on the speech recognition task. A subset of *ISOLET* is established for our experiments, which contains 30 speakers speaking the name of each letter of the alphabet (i.e., "A"–"Z") twice. As a result, we have totally $30 \times 26 \times 2 = 1560$ examples, and each example is encoded as a 617-dimensional feature vector. Our task is to identify which of the 26 letters each example belongs to. Among these examples, we randomly pick up 20% of examples to establish the test set, and the remaining 80% of examples form the training set. To incorporate different levels of label noise, we randomly select 0%, 20%, 40%, and 60% of examples in the training set and then switch their accurate labels to the wrong values. Such example selection and label corruption are conducted five times, so every compared algorithm should independently run five times on the contaminated data set and the average accuracy on the test set over these five different runs is particularly investigated. The standard deviation over the five runs is also reported to display the stability of each compared algorithm.

In LNSI, a 10-NN graph with kernel width $\sigma_k = 0.1$ was established. The tradeoff parameters in (6) such as λ_1, λ_2 and λ_3 were tuned by searching the grid $\{10^{-2}, 10^{-1}, \ldots, 10^{3}\}$. From Table II, we see that the performances of LNSI are significantly better than baseline methods in almost all cases.

[1] https://archive.ics.uci.edu/ml/data_sets.html

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

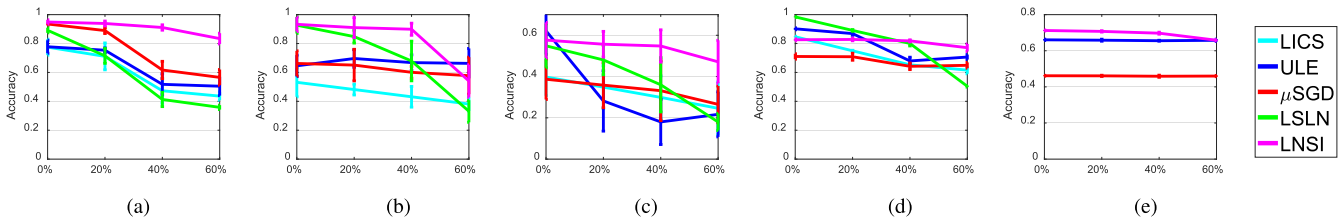


Fig. 3. Experimental results of the compared methods on five UCI benchmark data sets. (a)–(e) *CNAE9*, *Wine*, *Breast Tissue*, *Pendigits*, and *Connect-4* data sets, respectively.

TABLE II

COMPARISON OF VARIOUS METHODS ON *ISOLET* DATA SET. THE CLASSIFICATION ACCURACIES (MEAN ± STD) UNDER DIFFERENT LEVELS OF LABEL NOISE ARE PRESENTED. ●/○ INDICATES THAT LNSI IS SIGNIFICANTLY BETTER/WORSE THAN THE CORRESPONDING METHOD (PAIRED t-TEST AT 95% CONFIDENCE LEVEL). THE BEST ACCURACY UNDER EACH LABEL NOISE LEVEL IS MARKED IN BOLD

Method	0%	20%	40%	60%
LICS [3]	0.8827 ± 0.0218●	0.7577 ± 0.0483●	0.6147 ± 0.0601●	0.4442 ± 0.0521●
ULE [4]	0.8846 ± 0.0220	0.7942 ± 0.0368●	0.6010 ± 0.0436●	0.4296 ± 0.0478●
μSGD [6]	0.8128 ± 0.0528●	0.6660 ± 0.0407●	0.5581 ± 0.0337●	0.3879 ± 0.0462●
LSLN [15]	0.8622 ± 0.0109●	0.5974 ± 0.0397●	0.4703 ± 0.0301●	0.3327 ± 0.0042●
LNSI	**0.9058 ± 0.0153**	**0.8827 ± 0.0129**	**0.8519 ± 0.0239**	**0.7763 ± 0.0158**

TABLE III

COMPARISON OF VARIOUS METHODS ON *COIL20* DATA SET. THE CLASSIFICATION ACCURACIES (MEAN ± STD.) UNDER DIFFERENT LEVELS OF LABEL NOISE ARE PRESENTED. ●/○ INDICATES THAT LNSI IS SIGNIFICANTLY BETTER/WORSE THAN THE CORRESPONDING METHOD (PAIRED t-TEST AT 95% CONFIDENCE LEVEL). THE BEST ACCURACY UNDER EACH LABEL NOISE LEVEL IS MARKED IN BOLD

Method	0%	20%	40%	60%
LICS [3]	0.8263 ± 0.0428●	0.7513 ± 0.0457●	0.4083 ± 0.0661●	0.2792 ± 0.0359●
ULE [4]	0.9868 ± 0.0079●	0.9347 ± 0.0382●	0.4472 ± 0.0600●	0.4181 ± 0.0344●
μSGD [6]	0.8965 ± 0.0212●	0.8542 ± 0.0409●	0.4264 ± 0.0627●	0.4042 ± 0.0765●
LSLN [15]	0.9993 ± 0.0016	0.9104 ± 0.0206●	0.3743 ± 0.0564●	0.2167 ± 0.0090●
Co-teaching [18]	0.9969 ± 0.0017●	0.9617 ± 0.0033●	0.9148 ± 0.0222●	0.8266 ± 0.0211●
LNSI	**1.0000 ± 0.0000**	**0.9951 ± 0.0019**	**0.9875 ± 0.0087**	**0.9736 ± 0.0138**

In addition, LNSI is much more robust than other compared methods, as their performances decrease sharply with the increase of levels of label noise.

D. Experiments on COIL20 Data Set

This section tests the performance of LNSI on an image data set, i.e., the COIL20[3]. *COIL20* is a popular public data set for object classification, which includes 1440 object images belonging to 20 classes, and each class has 72 images shot from different angles. The resolution of each gray-level image is 32×32 [46]. We use the output of the first fully connected layer of VGGNet-16 as the CNN features, and therefore, each image in *COIL20* can be represented by a feature vector with 4096 dimensions.

Similar to the experimental settings in Section VI-C, we randomly pick up 20% of examples for test, and the remaining 80% of examples are served as training data. Also, we randomly select 0%, 20%, 40%, and 60% of training examples and switch the accurate label of each of them to a random wrong label. In this data set, we establish a 10-NN graph with $\sigma_k = 0.1$. In addition, the tradeoff parameters in (6) such as λ_1, λ_2, and λ_3 were tuned by searching the grid $\{10^{-2}, 10^{-1}, \ldots, 10^{3}\}$ in order to obtain the satisfactory results.

The classification accuracies of all compared methods under different label noise levels are presented in Table III. It can be observed that the performances of all methods decrease with the increase in noise level. However, LNSI achieves the best results in most cases when compared with other baseline methods. Another notable fact is that LNSI performs robustly under different levels of label noise, while the performances of baselines dramatically decrease when the noise rate ranges from 0% to 60%.

E. Experiments on MNIST Data Set

We use the *MNIST*[4] data set to evaluate the capabilities of various methods on handwritten digit recognition. Here we use a subset of *MNIST* for our experiments, which contains 2000 training examples and 2000 test examples across ten different classes. The number of examples belonging to each

[3]http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

[4]http://yann.lecun.com/exdb/mnist/

TABLE IV

COMPARISON OF VARIOUS METHODS ON *MNIST* DATA SET. THE CLASSIFICATION ACCURACIES (MEAN ± STD) UNDER DIFFERENT LEVELS OF LABEL NOISE ARE PRESENTED. ●/○ INDICATES THAT LNSI IS SIGNIFICANTLY BETTER/WORSE THAN THE CORRESPONDING METHOD (PAIRED t-TEST AT 95% CONFIDENCE LEVEL). THE BEST ACCURACY UNDER EACH LABEL NOISE LEVEL IS MARKED IN BOLD

Method	0%	20%	40%	60%
LICS [3]	0.7352 ± 0.0137●	0.7104 ± 0.0137●	0.6502 ± 0.0374●	0.6404 ± 0.0198●
ULE [4]	0.8815 ± 0.0065●	0.8637 ± 0.0052●	0.6465 ± 0.0047●	0.6385 ± 0.0282●
μSGD [6]	0.7717 ± 0.0208●	0.7620 ± 0.0201●	0.6285 ± 0.0404●	0.6278 ± 0.0360●
LSLN [15]	0.9575 ± 0.0088	0.9320 ± 0.2000	0.4895 ± 0.0162●	0.4108 ± 0.0053●
Co-teaching [18]	**0.9823 ± 0.0044**○	**0.9505 ± 0.0071**	0.9240 ± 0.0098	0.8328 ± 0.0356●
LNSI	0.9558 ± 0.0130	0.9463 ± 0.0069	**0.9252 ± 0.0113**	**0.8885 ± 0.0171**

TABLE V

COMPARISON OF VARIOUS METHODS ON *CIFAR-10* DATA SET. THE CLASSIFICATION ACCURACIES (MEAN ± STD) UNDER DIFFERENT LEVELS OF LABEL NOISE ARE PRESENTED. ●/○ INDICATES THAT LNSI IS SIGNIFICANTLY BETTER/WORSE THAN THE CORRESPONDING METHOD (PAIRED t-TEST AT 95% CONFIDENCE LEVEL). THE BEST ACCURACY UNDER EACH LABEL NOISE LEVEL IS MARKED IN BOLD

Method	0%	20%	40%	60%
ULE [4]	0.8146 ± 0.0051●	0.8096 ± 0.0034●	0.7518 ± 0.0052●	0.7597 ± 0.0061●
μSGD [6]	0.7427 ± 0.0049●	0.7406 ± 0.0087●	0.7236 ± 0.0098●	0.7162 ± 0.0081●
Co-teaching [18]	0.8303 ± 0.0096●	0.7604 ± 0.0041●	0.7066 ± 0.0067●	0.6078 ± 0.0085●
LNSI	**0.8530 ± 0.0026**	**0.8485 ± 0.0036**	**0.8373 ± 0.0043**	**0.7756 ± 0.0028**

class ranges from 359 to 454 and 80% of these examples are randomly picked up to establish the training set. The resolution of each gray-level handwritten digit image is 28×28. We use the output of the first fully connected layer of VGGNet-16 to extract the CNN features for each image, and therefore, the dimensionality of one feature vector is 4096.

All experimental settings are the same as those in Sections VI-C and VI-D. In this data set, a 10-NN graph with $\sigma_k = 1$ was established and the tradeoff parameters in (6) such as λ_1, λ_2, and λ_3 were tuned by searching the grid $\{10^0, 10^1, \ldots, 10^5\}$. Similarly, the mean accuracies and standard deviations of five independent runs of all comparators are reported, which can be found in Table IV. We see that when the noise rate is 0%, 20%, and 40%, LNSI is comparable with other baseline methods. However, LNSI achieves very robust and satisfactory performances on *MNIST* when the label noise rate is 60%, while all other baseline methods perform unsatisfactorily. Consequently, the existing methods are inferior to LNSI in terms of the classification accuracy under serious label noise.

F. Experiments on CIFAR-10 Data Set

This section tests the performance of LNSI and baselines on the natural image data set, i.e., the CIFAR-10[5]. Here we use a subset of *CIFAR-10* for our experiments which contains 30 000 image examples randomly sampled from original data set across different classes. The resolution of each color image is $32 \times 32 \times 3$. We extract the CNN features for each image, which are calculated as the output of the first fully connected layer of VGGNet-16, and therefore, the dimensionality of one feature vector is 4096.

[5]https://www.cs.toronto.edu/ kriz/cifar.html

All experimental settings are the same as those in Sections VI-C–VI-E. In this data set, 80% of the examples are randomly selected as the training set. A 15-NN graph with $\sigma_k = 0.5$ was established, and the tradeoff parameters in (6) such as λ_1, λ_2, and λ_3 were tuned by searching the grid $\{10^{-1}, 10^0, \ldots, 10^3\}$.

The classification accuracies of the compared algorithms under different label noise levels are presented in Table V, from which we can see that LNSI outperforms other baselines and is much more robust to noisy labels with the increase in the level of label noise. Note that LICS and LSLN are not compared on the *CIFAR-10* data set as they are not scalable to this data set.

G. Illustration of Convergence

In Section V-A, we have theoretically proved that the optimization process in Algorithm 1 will converge to a stationary point. In this section, we present the convergence curves of LNSI on the four practical data sets appeared in Sections VI-C–VI-F. From the curves shown in Fig. 4, we see that the differences between the two sides of equality constraints in (7) decrease rapidly on all data sets. This observation justifies our previous theoretical results and demonstrates that ADMM is effective and efficient for solving (7).

H. Ablation Study

From the above-mentioned experimental results presented in Sections VI-B–VI-F, we see that LNSI performs favorably to other existing methods. Therefore, this section investigates the effects of key components of LNSI that leads to the good performance. Specifically, we conduct the ablation study on LNSI to explore the individual contributions of the low-rank regularizer $\|\boldsymbol{Z}\|_*$ and the Laplacian regularizer

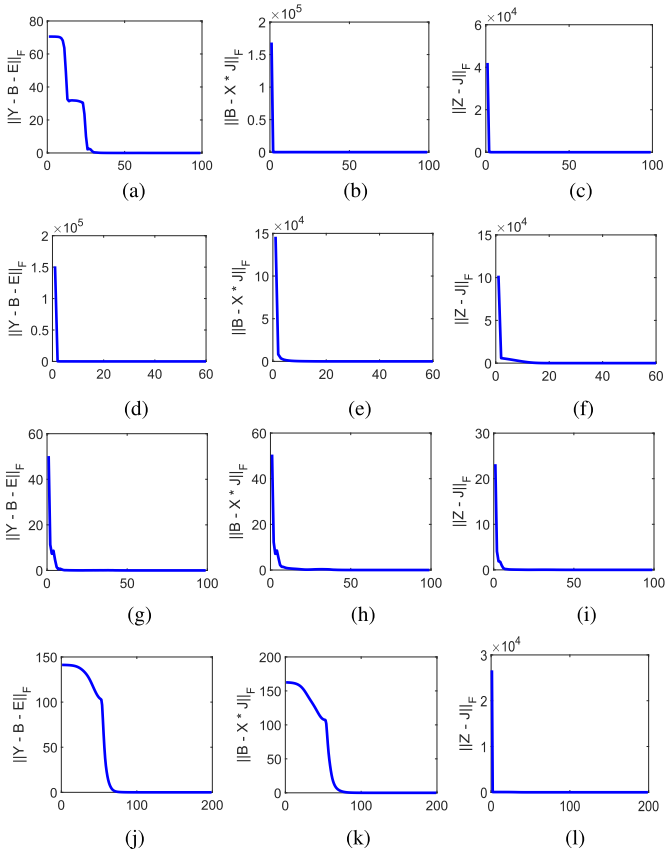


Fig. 4. Illustration of convergence process of the ADMM method adopted by LNSI on the four practical data sets. For each data set, we present the convergence curves under different convergence criteria in the Algorithm 1. (a)–(c) *ISOLET* data set. (d)–(f) *COIL20* data set. (g)–(i) *MNIST* data set. (j)–(l) *CIFAR-10* data set.

$\mathrm{tr}((\boldsymbol{XZ})^\top \boldsymbol{L}(\boldsymbol{XZ}))$. To this end, we study the performances of three different settings on the above four practical data sets (such as *ISOLET*, *COIL20*, *MNIST*, and *CIFAR-10*).

First, both low-rank regularizer and Laplacian regularizer are reserved to constitute the original model (abbreviated as "LNSI"); second, the Laplacian regularizer is removed from the original model to see how this term influences the model performance (abbreviated as "No Laplacian"); third, we remove the low-rank regularizer while keeping the Laplacian regularizer to observe the effect of low-rank regularizer (abbreviated as "No Low-Rank").

The experimental results of these three models are illustrated in Fig. 5, in which 40% and 60% of training examples have incorrect labels. The results reveal that LNSI achieves the best performance on all four data sets, especially when the label noise rate is relatively high. By contrast, the accuracy of LNSI will drop without any of the two terms such as low-rank regularizer and Laplacian regularizer, and therefore, these two regularization terms are essential to boost the performance of LNSI.

I. Parametric Sensitivity

Note that the objective function (8) in LNSI contains three tradeoff parameters λ_1, λ_2, and λ_3 that should be manually

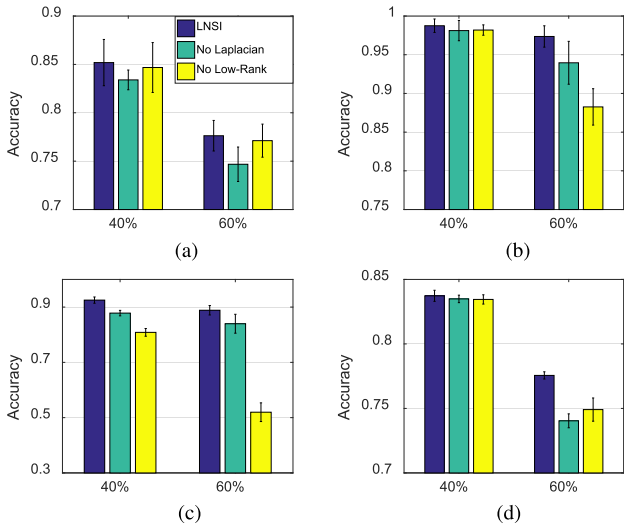


Fig. 5. Results of ablation study on four practical data sets. For convenience, the original model is denoted as "LNSI," the setting without the Laplacian regularization term is dubbed as "No Laplacian," and the setting without the low-rank term is named as "No Low-Rank." (a) ISOLET data set. (b) COIL20 Data set. (c) MNIST data set. (d) CIFAR-10 data set.

tuned. Therefore, in this section, we discuss whether the choices of them will significantly influence the performance of LNSI. To this end, we examine the classification accuracy at two different levels (20% and 60%) of label noise via changing one of λ_1, λ_2, and λ_3, and meanwhile fixing the others to the optimal constant values under different data sets and different noise rates. The above-mentioned four practical data sets are adopted here, and the results are shown in Fig. 6. Fig. 6(a)–(c) shows the experiments on the *ISOLET* data set, (d)–(f) shows the experiments on the *COIL20* data set, (g)–(i) shows the experiments on the *MNIST* data set, and (j)–(l) shows the experiments on the *CIFAR-10* data set. The results reveal that LNSI is robust to the variations of λ_1 and λ_2 in a wide range, so they can be easily tuned for practical use. Meanwhile, the performance of LNSI varies when λ_3 changes in a wide range, as λ_3 controls the capability of our method for capturing the label noise via the error matrix $\boldsymbol{E}$. Specifically, if λ_3 is large, the label noise will be greatly ignored, leading to the performance degradation of LNSI on *COIL20* and *MNIST* when the noise rate is 60%.

J. Summary of Experiments

Based on the above-mentioned experimental results of Sections VI-B–VI-I, we observe that: 1) LNSI performs better than other baseline algorithms in most cases, both on benchmark data sets and practical data sets; 2) the proposed algorithm in Algorithm 1 converges quickly to a stationary point; 3) LNSI is robust to the variation of the two tradeoff parameters including λ_1 and λ_2; 4) when the label noise is serious, the performance of LNSI might drop when λ_3 is set to a large value; 5) the introduced low-rank regularizer and Laplacian regularizer are both beneficial to improve the classification performance; and 6) the proposed LNSI outperforms other compared baselines when 40% and 60% labels

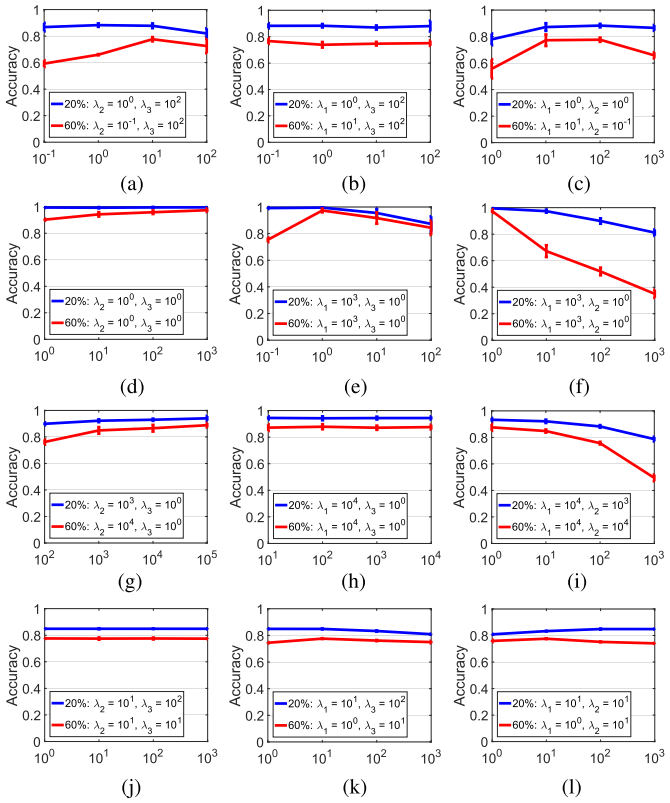


Fig. 6. Parametric sensitivity of LNSI. (a)–(c), (d)–(f), (g)–(i), and (j)–(l) *ISOLET*, *COIL20*, *MNIST*, and *CIFAR-10* data sets, respectively. (a), (d), (g), and (j) Variation in accuracy with respect to the tradeoff parameter λ_1 when λ_2 and λ_3 are fixed to the values indicated in the legend. (b), (e), (h), and (k) Influence of λ_2 to the classification performance with other parameters fixed. (c), (f), (i), and (l) Effect of λ_3 while other two tradeoff parameters are fixed.

of training examples are incorrect because the error matrix $\boldsymbol{E}$ can capture the label errors successfully and the Laplacian regularizer is essential to boost the performance of LNSI.

VII. Conclusion

To solve the label inaccuracy that often occurs in plenty of real-world data sets for classification, this article provides a novel paradigm that formulates the noisy label removal problem as a matrix recovery problem and treats the example features as the side information to aid the recovery process. Our proposed LNSI seamlessly forms the label noise removal and classifier parameter optimization into a unified framework. The convergence property, computational complexity, and generalization bound are also theoretically analyzed. We tested LNSI on various benchmark and practical data sets under different levels of label noise and found that LNSI outperforms other compared baseline methods and achieves robust results to different levels of label noise.

Appendix
Proof of the Convergence

To begin with, we provide the following two lemmas for the general ADMM solver.

Lemma 12 [47]: Given the optimization problem with linear constraints as

$$\begin{aligned} &\min_{\boldsymbol{P},\boldsymbol{Q}} \ f(\boldsymbol{P})+g(\boldsymbol{Q}) \\ &\text{s.t. } \boldsymbol{A}_P\boldsymbol{P}+\boldsymbol{B}_Q\boldsymbol{Q}=\boldsymbol{C} \end{aligned} \tag{40}$$

where $\boldsymbol{A}_P$ and $\boldsymbol{B}_Q$ are the coefficient matrices, and $f(\boldsymbol{P})$, $g(\boldsymbol{Q})$ are two functions with respect to the variables $\boldsymbol{P}$, $\boldsymbol{Q}$, respectively. $\boldsymbol{C}$ is a constant. Then, the Lagrangian function of (40) is

$$\begin{aligned} \mathcal{L}_\mu = f(\boldsymbol{P})+g(\boldsymbol{Q})+\operatorname{tr}(\boldsymbol{M}'^{\top}(\boldsymbol{A}_P\boldsymbol{P}+\boldsymbol{B}_Q\boldsymbol{Q}-\boldsymbol{C})) \\ +\frac{\mu}{2}\|\boldsymbol{A}_P\boldsymbol{P}+\boldsymbol{B}_Q\boldsymbol{Q}-\boldsymbol{C}\|_F^2 \end{aligned} \tag{41}$$

where $\boldsymbol{M}'$ is the Lagrangian multiplier and $\mu > 0$ is the penalty coefficient. ADMM consists of the iterations

$$\boldsymbol{P}^{k+1} := \arg\min_{\boldsymbol{P}} \mathcal{L}_\mu(\boldsymbol{P}, \boldsymbol{Q}^k, \boldsymbol{M}'^k) \tag{42}$$

$$\boldsymbol{Q}^{k+1} := \arg\min_{\boldsymbol{Q}} \mathcal{L}_\mu(\boldsymbol{P}^{k+1}, \boldsymbol{Q}, \boldsymbol{M}'^k) \tag{43}$$

$$\boldsymbol{M}'^{k+1} := \boldsymbol{M}'^k+\mu(\boldsymbol{A}_P\boldsymbol{P}^{k+1}+\boldsymbol{B}_Q\boldsymbol{Q}^{k+1}-\boldsymbol{C}). \tag{44}$$

If $f(\boldsymbol{P})$ and $g(\boldsymbol{Q})$ are convex, proper, and closed functions, and the unaugmented Lagrangian $\mathcal{L}_0 = f(\boldsymbol{P})+g(\boldsymbol{Q})+\operatorname{tr}(\boldsymbol{M}'^{\top}(\boldsymbol{A}_P\boldsymbol{P}+\boldsymbol{B}_Q\boldsymbol{Q}-\boldsymbol{C}))$ has a saddle point, the sequences $\{\boldsymbol{P}^k, \boldsymbol{Q}^k, \boldsymbol{M}'^k\}$ generated by the ADMM algorithm are guaranteed to converge.

Lemma 13 [47]: The generic-constrained convex optimization problem regarding the variable $\boldsymbol{P}$ is

$$\begin{aligned} &\min_{\boldsymbol{P}} \ f(\boldsymbol{P}) \\ &\text{s.t. } \boldsymbol{P}\in\mathcal{C} \end{aligned} \tag{45}$$

where $f(\cdot)$ and $\mathcal{C}$ are convex. Problem (45) can then be rewritten in the form of ADMM as

$$\begin{aligned} &\min \ f(\boldsymbol{P})+g(\boldsymbol{Z}_c) \\ &\text{s.t. } \boldsymbol{P}-\boldsymbol{Z}_c=\boldsymbol{O} \end{aligned} \tag{46}$$

where g is the indicator function of $\mathcal{C}$.

Now we begin to formally verify the convergence of Algorithm 1.

The proof of Theorem 2 is presented as follows.

Proof: To facilitate the proof, we rewrite the optimization problem (7) in the formation of (40) according to Lemma 12.

First, the optimization problem (7) with the convex set constraint can be transformed to the formation of (46) in Lemma 13. Therefore, problem (7) is equivalent to

$$\begin{aligned} \min_{\boldsymbol{Z},\boldsymbol{E}} \ &\|\boldsymbol{Z}\|_*+\lambda_1\|\boldsymbol{Z}\|_F^2+\lambda_2\operatorname{tr}((\boldsymbol{X}\boldsymbol{J})^{\top}\boldsymbol{L}(\boldsymbol{X}\boldsymbol{J})) \\ &+\lambda_3\|\boldsymbol{E}\|_{2,1}+\mathcal{I}_{\mathcal{C}}(\boldsymbol{K}) \\ \text{s.t. } &\boldsymbol{Y}=\boldsymbol{B}+\boldsymbol{E}, \quad \boldsymbol{B}=\boldsymbol{X}\boldsymbol{J}, \quad \boldsymbol{J}=\boldsymbol{Z}, \quad \boldsymbol{K}=\boldsymbol{B} \end{aligned} \tag{47}$$

where

$$\mathcal{I}_{\mathcal{C}}(x)=\begin{cases} x, & \text{if } x\in\mathcal{C} \\ \infty, & \text{othrewise.} \end{cases} \tag{48}$$

It can be easily verified that the above problem (47) can be represented in the form of (40) by setting

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{Z} \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{E} \\ \boldsymbol{J} \\ \boldsymbol{K} \end{bmatrix} \tag{49}$$

and

$$\boldsymbol{A}_P = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{B}_Q = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{X} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & -\boldsymbol{I} \\ \boldsymbol{O} & -\boldsymbol{I} & \boldsymbol{O} \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} \boldsymbol{Y} \\ \boldsymbol{O} \\ \boldsymbol{O} \\ \boldsymbol{O} \end{bmatrix} \tag{50}$$

where $\boldsymbol{I}$ and $\boldsymbol{O}$ are the identity matrices and zero matrices with proper sizes, respectively. The functions $f(\boldsymbol{P})$ and $g(\boldsymbol{Q})$ in (40) can be, respectively, expressed as

$$f(\boldsymbol{P}) = \|\boldsymbol{Z}\|_* + \lambda_1 \|\boldsymbol{Z}\|_F^2 \tag{51}$$

$$g(\boldsymbol{Q}) = \lambda_2 \mathrm{tr}((\boldsymbol{XJ})^\top \boldsymbol{L}(\boldsymbol{XJ})) + \lambda_3 \|\boldsymbol{E}\|_{2,1} + \mathcal{I}_{\mathcal{C}}(\boldsymbol{K}). \tag{52}$$

The unaugmented Lagrangian is formulated as

$$\begin{aligned} \mathcal{L}_0 = {} & \|\boldsymbol{Z}\|_* + \lambda_1 \|\boldsymbol{Z}\|_F^2 + \lambda_2 \mathrm{tr}((\boldsymbol{XJ})^\top \boldsymbol{L}(\boldsymbol{XJ})) + \lambda_3 \|\boldsymbol{E}\|_{2,1} \\ & + \mathrm{tr}(\boldsymbol{M}_1^\top (\boldsymbol{Y} - \boldsymbol{B} - \boldsymbol{E})) + \mathrm{tr}\big(\boldsymbol{M}_2^\top (\boldsymbol{B} - \boldsymbol{XJ})\big) \\ & + \mathrm{tr}\big(\boldsymbol{M}_3^\top (\boldsymbol{Z} - \boldsymbol{J})\big). \end{aligned} \tag{53}$$

Obviously, both $f(\boldsymbol{P})$ and $g(\boldsymbol{Q})$ are closed, proper, and convex, and the unaugmented Lagrangian $\mathcal{L}_0$ has a saddle point, which demonstrate that the optimization process for (7) is convergent. □

REFERENCES

[1] R. J. Hickey, "Noise modelling and evaluating learning from examples," *Artif. Intell.*, vol. 82, nos. 1–2, pp. 157–179, 1996.

[2] C. Gong, H. Zhang, J. Yang, and D. Tao, "Learning with inadequate and incorrect supervision," in *Proc. Int. Conf. Data Mining*, Nov. 2017, pp. 889–894.

[3] W. Gao, L. Wang, Y.-F. Li, and Z.-H. Zhou, "Risk minimization in the presence of label noise," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1575–1581.

[4] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1196–1204.

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

[6] G. Patrini, F. Nielsen, R. Nock, and M. Carioni, "Loss factorization, weakly supervised learning and label noise robustness," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 708–717.

[7] R. Wang, T. Liu, and D. Tao, "Multiclass learning with partially corrupted labels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2568–2580, Jun. 2018.

[8] P. Zhao, Y. Jiang, and Z.-H. Zhou, "Multi-view matrix completion for clustering with side information," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, 2017, pp. 403–415.

[9] M. Xu, R. Jin, and Z. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2301–2309.

[10] K.-Y. Chiang, C.-J. Hsieh, and I. S. Dhillon, "Matrix completion with noisy side information," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3447–3455.

[11] Y. Guo, "Convex co-embedding for matrix completion with predictive side information," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1955–1961.

[12] K.-Y. Chiang, C.-J. Hsieh, and I. Dhillon, "Robust principal component analysis with side information," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2291–2299.

[13] 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.

[14] X. Zhu, X. Wu, and Q. Chen, "Eliminating class noise in large datasets," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 920–927.

[15] B. van Rooyen, A. K. Menon, and R. C. Williamson, "Learning with symmetric label noise: The importance of being unhinged," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 10–18.

[16] A. Khetan, Z. C. Lipton, and A. Anandkumar, "Learning from noisy singly-labeled data," in *Proc. Int. Conf. Learn. Represent.*, 2018. [Online]. Available: https://openreview.net/forum?id=H1sUHgb0Z

[17] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2233–2241.

[18] B. Han *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8536–8546.

[19] B. Han *et al.*, "Masking: A new perspective of noisy supervision," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5836–5846.

[20] B. Han, I. W. Tsang, L. Chen, C. P. Yu, and S.-F. Fung, "Progressive stochastic learning for noisy labels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5136–5148, Oct. 2018.

[21] J. Zhang, V. S. Sheng, T. Li, and X. Wu, "Improving crowdsourced label quality using noise correction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1675–1688, May 2018.

[22] F. Zhao and Y. Guo, "Learning discriminative recommendation systems with side information," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 3469–3475.

[23] C. C. Aggarwal, Y. Zhao, and P. S. Yu, "On text clustering with side information," in *Proc. 28th Int. Conf. Data Eng.*, Apr. 2012, pp. 894–904.

[24] R. Zhang, F. Nie, and X. Li, "Semisupervised learning with parameter-free similarity of label and side information," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 405–414, Feb. 2019.

[25] K. Ahn, K. Lee, H. Cha, and C. Suh, "Binary rating estimation with graph side information," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4272–4283.

[26] N. Xue, Y. Panagakis, and S. Zafeiriou, "Side information in robust principal component analysis: Algorithms and applications," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4317–4325.

[27] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[28] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph Laplacian for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2261–2274, Oct. 2015.

[29] Y. Huang, D. Xu, and F. Nie, "Semi-supervised dimension reduction using trace ratio criterion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 519–526, Mar. 2012.

[30] Y. Wang, Y. Jiang, Y. Wu, and Z.-H. Zhou, "Spectral clustering on multiple manifolds," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1149–1161, Jul. 2011.

[31] J. Yu, C. Hong, Y. Rui, and D. Tao, "Multitask autoencoder model for recovering human poses," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 5060–5068, Jun. 2018.

[32] J. Yu, X. Yang, F. Gao, and D. Tao, "Deep multimodal distance metric learning using click constraints for image ranking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4014–4024, Dec. 2017.

[33] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, "Label propagation via teaching-to-learn and learning-to-teach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1452–1465, Jun. 2017.

[34] C.-J. Hsieh, N. Natarajan, and I. S. Dhillon, "PU learning for matrix completion," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 2445–2453.

[35] N. Komodakis and G. Tziritas, "Approximate labeling via graph cuts based on linear programming," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1436–1453, Aug. 2007.

[36] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2010, *arXiv:1009.5055*. [Online]. Available: https://arxiv.org/abs/1009.5055

[37] C. Gong, "Exploring commonality and individuality for multi-modal curriculum learning," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1926–1933.

[38] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Program.*, vol. 155, nos. 1–2, pp. 57–79, 2016.

[39] P. L. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, Mar. 2003.
[40] S. Kakade, S. Shalev-Shwartz, and A. Tewari. (2009). *On the Duality of Strong Convexity and Strong Smoothness: Learning Applications and Matrix Regularization*. [Online]. Available: http://ttic.uchicago.edu/shai/papers/KakadeShalevTewari09.pdf
[41] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari, "Regularization techniques for learning with matrices," *J. Mach. Learn. Res.*, vol. 13, pp. 1865–1890, Jun. 2012.
[42] P.-A. Absil and P. Van Dooren, "Two-sided Grassmann–Rayleigh quotient iteration," *Numerische Mathematik*, vol. 114, no. 4, pp. 549–571, 2010.
[43] R. Mahony and P.-A. Absil, "The continuous-time Rayleigh quotient flow on the sphere," *Linear Algebra Appl.*, vol. 368, pp. 343–357, Jul. 2003.
[44] H. Zhang and F. Ding, "On the kronecker products and their applications," *J. Appl. Math.*, vol. 2013, Jun. 2013, Art. no. 296185.
[45] R. Meir and T. Zhang, "Generalization error bounds for Bayesian mixture algorithms," *J. Mach. Learn. Res.*, vol. 4, pp. 839–860, Dec. 2003.
[46] J. Yu, D. Tao, J. Li, and J. Cheng, "Semantic preserving distance metric learning and applications," *Inf. Sci.*, vol. 281, pp. 674–686, Oct. 2014.
[47] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

Yang Wei received the B.S. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2015, where she is currently pursuing the Ph.D. degree.

Her current research interests include pattern recognition, incomplete data-based learning, and deep learning.

Chen Gong (M'17) received the dual Ph.D. degree from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the University of Technology Sydney (UTS), Ultimo, NSW, Australia, in 2016.

He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. He has authored or coauthored more than 50 technical articles at prominent journals and conferences such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), the IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), the IEEE TRANSACTIONS ON CYBERNETICS (T-CYB), CVPR, AAAI, IJCAI, ICDM, and so on. His current research interests include machine learning and data mining.

Dr. Gong was a recipient of the Excellent Doctoral Dissertation Award by SJTU and the Chinese Association for Artificial Intelligence (CAAI). He was also enrolled by the Summit of the Six Top Talents Program of Jiangsu Province, China.

Shuo Chen received the B.S. degree from the School of Computer Science and Engineering, Jinling Institute of Technology, Nanjing, China, in 2014. He is currently pursuing the Ph.D. degree with the Nanjing University of Science and Technology, Nanjing.

His current research interests include pattern recognition, metric learning, and deep learning.

Tongliang Liu (M'14) is currently a Lecturer with the School of Computer Science and the Faculty of Engineering, and a Core Member with the UBTECH Sydney AI Centre, The University of Sydney, Darlington, NSW, Australia. He has authored and coauthored more than 60 research articles, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (T-PAMI), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), the IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), ICML, AAAI, IJCAI, CVPR, ECCV, KDD, and ICME, with best paper awards. His current research interests include machine learning, computer vision, and data mining.

Mr. Liu was a recipient of the 2019 ICME Best Paper Award and the Discovery Early Career Researcher Award (DECRA) from Australian Research Council (ARC).

Jian Yang (M'08) received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002.

In 2003, he was a Post-Doctoral Researcher with the University of Zaragoza, Zaragoza, Spain. From 2004 to 2006, he was a Post-Doctoral Fellow with the Biometrics Centre, The Hong Kong Polytechnic University, Hong Kong. From 2006 to 2007, he was a Post-Doctoral Fellow with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Chang-Jiang Professor with the School of Computer Science and Technology, NUST. He has authored more than 200 scientific articles in pattern recognition and computer vision. His articles have been cited more than 5000 times in the Web of Science and 13 000 times in the Scholar Google. His current research interests include pattern recognition, computer vision, and machine learning.

Dr. Yang is a fellow of IAPR. He is/was currently an Associate Editor of *Pattern Recognition*, *Pattern Recognition Letters*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), and *Neurocomputing*.

Dacheng Tao (F'15) is a Professor of computer science and an ARC Laureate Fellow with the School of Computer Science and the Faculty of Engineering, and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney, Darlington, NSW, Australia. His research results in artificial intelligence have expounded in one monograph. He has authored or coauthored more than 200 publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (T-PAMI), the *International Journal of Computer Vision* (IJCV), the *Journal of Machine Learning Research* (JMLR), AAAI, IJCAI, NIPS, ICML, CVPR, ICCV, ECCV, ICDM, and KDD, with several best paper awards.

Prof. Tao is a fellow of the Australian Academy of Science. He was a recipient of the 2018 IEEE ICDM Research Contributions Award and the 2015 Australian Scopus-Eureka Prize.