Neural Processing Letters
https://doi.org/10.1007/s11063-018-9963-9

Robust Class-Specific Autoencoder for Data Cleaning and Classification in the Presence of Label Noise

Weining Zhang[1,2] · **Dong Wang**[1,2] · **Xiaoyang Tan**[1,2]

Abstract
We present a simple but effective method for data cleaning and classification in the presence of label noise. The fundamental idea is to treat the data points with label noise as outliers of the class indicated by the corresponding noisy label. This essentially allows us to deal with the traditional supervised problem of classification with label noise as an unsupervised one, i.e., identifying outliers from each class. However, finding such dubious observations (outliers) from each class is challenging in general. We therefore propose to reduce their potential influence using class-specific feature learning by autoencoder. Particularly, we learn for each class a feature space using all the samples labeled as that class, including those with noisy (but unknown to us) labels. Furthermore, in order to solve the situation when the noise is relatively high, we propose a weighted class-specific autoencoder by considering the effect of each data point on the postulated model. To fully exploit the advantage of the learned class-specific feature space, we use a minimum reconstruction error based method for finding out the outliers (label noise) and solving the classification task. Experiments on several datasets show that the proposed method achieves state of the art performance on the task of data cleaning and classification with noisy labels.

Keywords Class-specific autoencoder · Label noise · Classification · Data cleaning · Outliers

1 Introduction

Classification is one of the most basic and core tasks in pattern recognition and machine learning. A classifier is first trained based on a labeled training set and is then used to predict the label of the test set. Since collecting reliably labeled data is often expensive and time-

This is an extended version of the paper published in ISNN 2018, [45].

✉ Xiaoyang Tan
x.tan@nuaa.edu.cn

1 College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, Jiangsu, China

2 Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China

Published online: 14 December 2018

consuming, nowadays people tend to use alternative simple and convenient methods, such as crowdsourcing [18], Amazon Mechanical Turk [14] and some other non-expert methods. However, the data collected in this way may result in a certain degree of label noise [21], i.e., some samples are incorrectly labeled. Figure 1 shows some images with label noise when harvesting specific category of data from the Internet web. Similar problems also exist in scenes where it is difficult to obtain completely correct labels, e.g. in some medical diagnosis problems [27].

The consequences of label noise are important and diverse. When the data set is polluted by label noise, it directly affects the performance of the classifier [26,46], and more samples are needed for effective learning. Moreover, inaccurate label information can seriously deteriorate the data quality, making the learning algorithm unnecessarily complex as it needs to handle additional uncertainty [1,5]. Some related tasks are also adversely affected by label noise, e.g., feature learning and feature selection [44]. Due to the above reasons, the problem of data cleaning and classification in the presence of label noise have recently attracted a lot of attention from researchers [9].

In this paper, we present a novel, simple but effective method for data cleaning and classification in the presence of label noise. Our key idea is based on the observations that the data points with label noise are highly likely to be the outliers of the class indicated by the corresponding noisy label. This conceptual connection between label noise and outliers essentially allows us to deal with the traditional supervised problem of classification with label noise as an unsupervised one, i.e., outliers analysis per class. To this end, we proposed a robust class-specific autoencoder based method to deal with label noise problem.

Fig. 1 Illustration of searching images using keyword 'horse face' from the Internet web. Images with noisy labels are marked with a red square. (Color figure online)

Particularly, we learn a separate feature space by using autoencoder for each class—note that although each class contains some portion of points that actually do not belong to it due to label noise, the influence of these points is supposed to be reduced by the class-specific autoencoder, hence yielding reliable feature space. For cases with higher label noise, we also propose a weighted class-specific autoencoder by robust optimization method which considers the impact of each data point on the postulated model. The verification experiments indicate that the proposed weighted optimization method can get a more robust feature representation. Based on the well learned class-specific feature space, the data cleaning and classification tasks can be solved in a unified and effective way. To our knowledge, it is the first attempt to use class-specific feature learning to deal with the label noise problem. To fully exploit the advantage of the learned class-specific feature space, we use a minimum reconstruction error based method to find out the outliers (label noise) in the training set and classify the data in the test set. Experiments on several datasets show that the proposed method achieves the state-of-the-art data cleaning results and gets an excellent performance on the task of classification with noisy labels.

The remaining parts of this paper are organized as follows. Section 2 briefly reviews the related work about autoencoder and label noise problem. Details of the proposed method are described in Sect. 3. The extensive experimental results are presented in Sect. 4. We conclude in Sect. 5 with a summary and directions for future work.

2 Related Work

In Sect. 2.1, we briefly review autoencoder and its extensions. Recent advances in solving the label noise problem are also reviewed in Sect. 2.2.

2.1 Autoencoder and Its Extensions

A vanilla autoencoder [3] is a neural network that tries to reconstruct its input, i.e., the expected output of the autoencoder is the input of the model. Assuming that we have a set of training samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and $\mathbf{x}_i \in R^d$, an autoencoder first encodes an input $\mathbf{x}$ to a hidden representation and then decodes it back to a reconstruction $\hat{\mathbf{x}}$. The $\hat{\mathbf{x}}$ is formulated as Eq. (1):

$$\hat{\mathbf{x}} = g(\mathbf{W}_2 f(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \tag{1}$$

where $\mathbf{W}_1$ and $\mathbf{b}_1$ are weight matrix and bias for encoder while $\mathbf{W}_2$ and $\mathbf{b}_2$ are for decoder. The activation function for $f(\cdot)$ and $g(\cdot)$ is sigmoid function. To learn the autoencoder which can well reconstruct the training samples, the loss function is written as Eq. (2):

$$L(\theta; \mathbf{X}) = \frac{1}{n} \sum_{i=1}^{n} ||\hat{\mathbf{x}}_i - \mathbf{x}_i||^2 \tag{2}$$

where $\theta = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2\}$. The loss function shown in Eq. (2) can be solved easily by mini-batch gradient descent method since the sigmoid function is smooth and continuously differentiable. After learning the encoding weights, we get compressed feature representation $\mathbf{z} = f(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)$ from the hidden layer which can be seen as a feature extractor [16], since the number of neurons in the hidden layer is inferior to the input ones in general.

Over time some variations and extensions are proposed based on the vanilla autoencoder. To discover more robust features and prevent it from simply learning the identity, a denoising

autoencoder [38] reconstructs the input from a corrupted version by randomly setting part of the neurons to zero. By manually zeroing all but few strongest hidden unit activations, Makhzani and Frey [23] imposes sparsity on the hidden units to get the most useful features of the input data. Rifai et al. [31] adds explicit regularizer in the objective function for avoiding the problem of overfitting. Also, Kingma and Welling [17] extends the vanilla autoencoder to a generative model which can generate various images by regulating hidden variables. Moreover, autoencoder can be stacked to form a deep autoencoder [39] which captures more complicated and abstract features. This kind of deep network has achieved great success in various applications, such as feature learning [24], object recognition [10] and etc.

2.2 Label Noise Problem in Survey

In the literature of solving the label noise problem, there exist two main approaches, which are label noise cleaning algorithms and label noise robust algorithms respectively.

(1) Label noise cleaning algorithms Typically, to reduce the influence of data points with noisy labels, one can clean the data first, which is the so-called data cleaning techniques, or filter approaches. Particularly, they turn the noisy training set into a cleaned training set by removing or relabeling suspicious instances and conduct the classification task in the cleaned training set. Jeatrakul et al. [15] learns a complementary neural network by using the training set with label noise and removes all the instances that are misclassified by both the Truth NN classifier and Falsity NN classifier. A similar attempt is proposed by Pruengkarn et al. [28] for SVM classifier. Hoz et al. [12] introduces a multi-objective optimization method for feature selection based on Growing Hierarchical Self-Organising Maps (GHSOMs) which can be used as a cleaning or relabeling method in outlier detection problem. However, the label noise cleaning algorithms based on classification results may remove some correctly labeled instances, and only minority mislabeled instances are removed because of the misclassification of the classifier. In fact, these methods may suffer from a *chicken-and-egg* dilemma, since learning from a polluted training set may get poor classifiers and noise cleaning technique highly relies on good classifiers.

To solve the above problem, based on the fact that support vectors of an SVM contain almost all of the mislabeled data, Fefilatyev et al. [8] introduces a human expert to check and verify the suspicious instances from the support vectors and relabel the instances which are likely mislabeled. Also, an improved method is proposed by Ekambaram et al. [7] for reducing the number of examples to be reviewed by the human expert. Although the human expert based methods can get the state-of-the-art cleaning results in real datasets, human experts are hard to get in most of the applications, and we can't guarantee that the new label is correct.

(2) Label noise robust algorithms Alternatively, one may make those data points with noisy labels less harmful using various robust methods. Some early approaches try to solve the label noise problem by avoiding overfitting, such as ensemble learning [19,33] and regularization techniques [35]. These methods try to make classifier not too sensitive for training data which can relieve label noise problem to a certain degree. However, experiments show that this kind of methods are still affected by label noise, which can only suit for simple cases where label noise is highly correlated with overfitting problem.

On the other hands, some algorithms are particularly designed for label noise problem using various robust statistics techniques, e.g., robust loss functions [4,22,25] and robust optimization methods [40,43]. However, these are mainly supervised methods in which the

effect of each data point on the postulated model is carefully controlled by design but at the risk of reduced learning efficiency.

Particularly, Rolnick et al. [32] studies the behavior of standard neural network training procedures in settings with massive label noise. Plenty of experiments prove that neural networks are robust to label noise which can learn from data that has been diluted by amount of label noise. Inspired by their conclusion, we use and modify a specific neural network (i.e., autoencoder) to find the difference between data with correct label and wrong label.

3 The Proposed Method

In this section, we first explain how to connect the label noise and outliers in conceptual. Then we verify that our proposed class-specific autoencoder is robust to outliers. Finally, a minimum reconstruction error based method is proposed to find out the outliers (label noise) in training set and classify the data in test set.

3.1 Label Noise as Outliers

Assume that we have N labeled training data, $\{\mathbf{x}_i, y_i\}, i = 1, \ldots, N$ in K classes, and some portions ϵ of the training data contain label noise, i.e., their labels are incorrectly annotated. Our goal is to learn a classifier f that assigns a new data point $\mathbf{x}$ to one of the K categories, in spite of the data contamination by label noise.

To reduce the influence of samples with noisy label, a natural idea is to separate them from the whole dataset. Since the label generation process is complicated, it is hard to identify mislabeled samples from a global perspective. However, if we think it locally (i.e. we consider the subset of the whole dataset where each subset has the same label), finding label noise samples would be much easier—they are just like outliers in its corresponding subset. Hereby "outliers", we mean it by Hawkins's definition as follows, '*An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism*' [11].[1] This conceptual connection between label noise and outliers inspires us to solve label noise problem by using outlier detection methods. There are numerous methods in the literature on this topic [2,6], which can therefore be immediately borrowed to handle the label noise problem.

Moreover, we would like to justify the proposed "class-specific" idea for label noise from the aspect of methodology. Indeed, it can be understood with the well-known "divide and conquer" principle—the proposed method essentially partitions the data points with label noise by class and handle them separately, hence avoiding the error accumulating effect if a global loss function is adopted. A similar idea also exists in the previous literatures [34].

3.2 Class-Specific Autoencoder

Despite the simplicity of the concept of outliers, in practice it is not always obvious how to quantify 'deviates so much' as this has a deep connection with the underlying model and its training procedure (which in turn could be biased by the outliers themselves). Fortunately, we can bypass the issue by robust feature learning method, i.e., class-specific autoencoder.

[1] In general, the nature of outliers can be hard to grasp and actually there is no unanimous definition of outlier in literature [13,37].

Fig. 2 The reconstruction results for correct labeled images and mislabeled images respectively. Each pair of the images is the digit image and its reconstruction. Images in the first row are the correct labeled images, while the others are the mislabeled images. **a** Labeled "1", **b** labeled "3" and **c** labeled "8"

For N labeled training data, $\{\mathbf{x}_i, y_i\}, i = 1, \ldots, N$ in K classes, we first separate them based on their corresponding label (note that some of the samples actually do not belong to it due to label noise), and then learn separately autoencoder by the loss function shown in Eq. (3):

$$L(\theta; \mathbf{X}) = \frac{1}{n} \sum_{i=1}^{n} ||\hat{\mathbf{x}}_i - \mathbf{x}_i||^2 + \lambda ||\theta||^2 \tag{3}$$

In Eq. (3), the first term is the average reconstruction error and the second term is an L2 regularization loss since avoiding overfitting can relieve label noise to a certain degree. Hereby, λ is used to balance the model complexity and reconstruction ability. This kind of loss function can be easily solved by mini-batch gradient descent shown in Eq. (4):

$$\theta_{j+1} = \theta_j - \alpha \frac{1}{m} \sum_{k=1}^{m} \nabla_\theta f(\mathbf{x}_k) \tag{4}$$

where θ is the model parameter, α is the learning rate, m is the batch size, and $f(\cdot)$ is the loss function for the model.

Hereby, we do a toy example to prove that the proposed class-specific autoencoder is effective and robust to label noise problem. 1000 images with 100 per class are randomly selected in MNIST dataset which is a well-known digit dataset [20]. We inject 10% label noise into these images following the widely used criteria [9]: (1) randomly select some portion of instances per class and (2) flip their labels into one of the other remaining labels. After introducing label noise, digits which are labeled "1", "3" and "8" are chosen to learn three class-specific autoencoders. The reconstruction results for correct labeled images and mislabeled images are visualized in Fig. 2. It can be shown that the correct labeled images are well reconstructed, although it exists 10% label noise when we learn the class-specific autoencoder. However, the reconstruction of mislabeled images is blurry and inaccurate which also reflects that the robustness of the proposed method to a certain degree. Furthermore, we also analyze the distribution of reconstruction error by using boxplot from a statistical perspective and the result is shown in Fig. 3. We can see that the reconstruction error for the correctly labeled images is much lower than the error for mislabeled images, which means our proposed method can effectively distinguish outliers from normal data. On the other hand, the perspective of reconstruction error may also give a suitable answer to how to quantify 'deviates so much' for outliers.

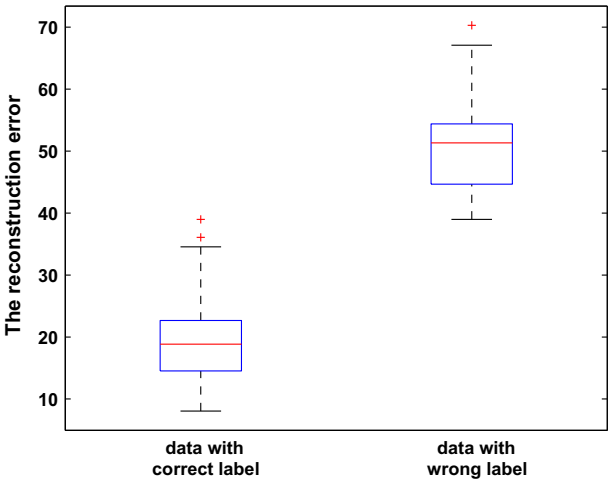

Fig. 3 The distribution of reconstruction error for data with 10% label noise

However, when the noise level is relatively high, simply learning a feature space by class-specific autoencoder is not enough. One can further improve the quality of the learned feature space using some label noise-robust methods [22,40]. Considering the effect of each data point on the postulated model, we propose a weighted class-specific autoencoder. Specifically, when updating the parameters of the class-specific autoencoder, a weighted mini-batch gradient descent is used as Eq. (5):

$$\theta_{j+1} = \theta_j - \alpha \sum_{k=1}^{m} w_k \nabla_\theta f(\mathbf{x}_k) \tag{5}$$

where θ is the parameter of an autoencoder, α is the learning rate, m is the batch size, $f(\cdot)$ is the loss function and w_k is the importance weight for $\mathbf{x}_k$, which is calculated by Eq. (6):

$$w_k = \frac{e^{-||\hat{\mathbf{x}}_k - \mathbf{x}_k||^2}}{\sum_{i=1}^{m} e^{-||\hat{\mathbf{x}}_i - \mathbf{x}_i||^2}}. \tag{6}$$

Different from the traditional mini-batch gradient descent where each data point plays the same role as shown in Eq. (4), we assign a weight w to consider their corresponding importance to the model based on the reconstruction error. The greater the reconstruction error of $\mathbf{x_k}$, the more it likely to be a data with noisy label which should get a lower importance in updating the gradient. Note that the proposed method is similar to the cost-sensitive learning, since they both consider different weights for different samples. However, our method is emphasized as a robust weighted optimization method where the importance weights for each samples are changeable during the training process.

Additional verification experiment is conducted to show the effectiveness of the proposed optimization method. The experiment settings are the same as the above toy example except that the label noise expands to 30%. Specifically, we analyze the distribution of reconstruction error when considering the importance weight and compare it with the original one. Note that the weighted optimization method is used to fine-tuning the model of the original one. The results are shown in Fig. 4. For Fig. 4a, it is clear to see that discrimination of reconstruction error for mislabeled data and correctly labeled data is reduced compared with the case of 10%

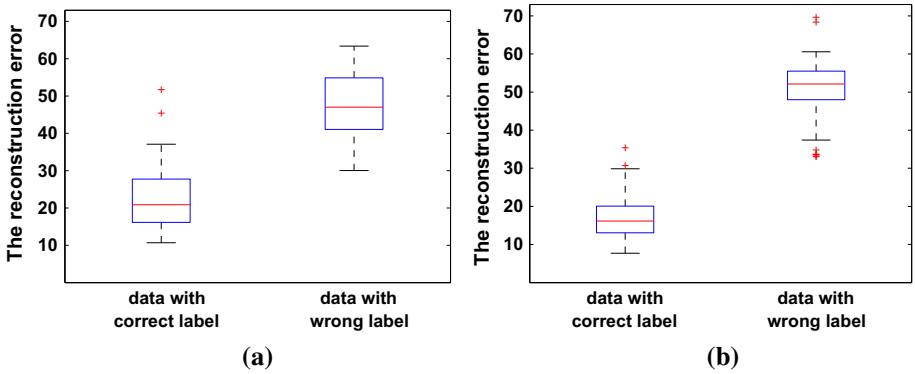

Fig. 4 The distribution of reconstruction error for data with 30% label noise. **a** Results for original optimization method, **b** results for weighted optimization method

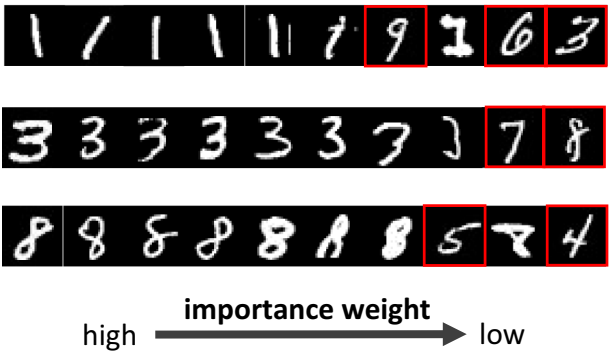

Fig. 5 The importance weight for some samples after finishing the training process based on the weighted optimization method. Each row has the same label which are '1', '3' and '8' respectively. The mislabeled data are marked with a red square. (Color figure online)

label noise shown in Fig. 3, since a higher label noise can further affect the feature learning. However, when using the weighted optimization method, the discrimination of reconstruction error shown in Fig. 4b is improved which has less overlap between the error for mislabeled data and correctly labeled data compared with the original one in Fig. 4a.

On the other hand, after finishing the training process based on the weighted optimization method, we visualized the images according to the order of importance weight from largest to smallest and the results are shown in Fig. 5. Figure 5 reflects that the mislabeled samples all get a low importance weight, although few correctly labeled samples also get a low weight since higher ambiguity. The above experiment results show that our proposed optimization method can actually get a more reliable feature space where the impact of label noise is further reduced.

3.3 Data Cleaning and Classification with Minimum Reconstruction Error

After feature learning by class-specific autoencoder, we obtain two sets of mappings. Each of the first set of mappings $f_j(\mathbf{x})$, $j = 1, \ldots, K$ projects a new point $\mathbf{x}$ into the jth feature spaces, while each of the second set of mappings $g_j(\mathbf{z})$, $j = 1, \ldots, K$ reconstructs a high-

dimensional sample from the point $\mathbf{z}$ in the latent space, which can be regarded as compact representation of some prototypes of the jth class. As the proposed class-specific autoencoder has effectively reduced the influence of the label noise, meanwhile, the reconstruction error can distinguish outliers from normal data, the data cleaning and classification tasks can be solved in a unified way based on the minimum reconstruction error criterion.

Particularly, for the classification task, we predict the label y_t of a test data $\mathbf{x}_t$ by simply calculating the reconstruction error on each autoencoder and then assign it to the class with minimum reconstruction error, as follows:

$$y_t = argmin_{j=1,2,3...,K} \|\|g_j(f_j(\mathbf{x}_t)) - \mathbf{x}_t\|\|^2. \tag{7}$$

For data cleaning task, whether a data contains label noise can be judged by an indicator function,

$$I(\mathbf{x}) = \begin{cases} 1 & y \neq \hat{y} \\ 0 & otherwise \end{cases} \tag{8}$$

where y is the label of the data and $\hat{y}$ is the predicted label by (6). A data will be regarded as an outlier (i.e. a mislabeled data) if y is not equal to $\hat{y}$.

Note that the proposed pipeline for solving label noise problem can be extended to an iterative one by (1) learning a class-specific autoencoder based on a training set. (2) cleaning (relabeling) the training set based on the well learnt autoencoder. By iterating the two steps repeatedly, we can find the mislabeled data as much as possible and get a more robust feature representation for each class which suits for the classification task on the test set. Particularly, we only iterate once for data cleaning task and iterate twice (i.e. retrain the autoencoder after the noise data are removed or relabeled) for classification task. More details are shown in the experiments section.

4 Experiments and Analysis

In this section, we evaluate the effectiveness of our proposed methods on two tasks: (1) finding the mislabeled data in the training set, i.e., data cleaning, and (2) classifying the test set based on the training set with label noise, i.e., classification in the presence of label noise.

4.1 Parameter Setting

In the implementation of the proposed methods, we use class-specific autoencoder and weighted class-specific autoencoder for feature learning, which are respectively denoted as CS-AE and CS-WAE. Parameter settings are mainly related to the model of autoencoder and its training process. The hidden layer for autoencoder comprised 200 neurons. We set 0.1 for regularization weight λ and 0.01 for learning rate α. Batch size is equal to the number of training samples in each class and epochs for training process is 100. Particularly, CS-WAE is used to fine-tuning the model of CS-AE in our follow-up experiments.

4.2 Data Cleaning

In this section, we verify the performance of our method in the data cleaning task, i.e., identifying those points that most likely to be outliers and removing them. Particularly, we

(a) (b)

Fig. 6 Illustration of artificial label noise for two pairwise confusing classes where each raw belongs to the same class and images with noisy labels are marked in a red square. **a** "7" versus "9", **b** "yo-yo" versus "roulette-wheel". (Color figure online)

Table 1 Noise removal performance (%) on pairwise confusing classes with different label noise

Pairwise classes	Noise level (%)	ICCN-SMO [30]	TC-SVM [8]	ALNR [7]	CS-AE	CS-WAE
7 versus 9	10	71.38	**98.83**	95.84	97.32	98.65
	20	78.60	97.21	96.01	95.25	**97.30**
	30	82.49	95.03	95.69	94.57	**96.24**
yo-yo versus roulette-wheel	10	54.94	**83.31**	79.67	82.03	82.71
	20	57.81	80.69	80.29	78.05	**81.23**
	30	59.33	73.45	70.10	69.26	**76.91**

The best results are in bold

follow the experimental protocol outlined in [8] and carry out the experiments on two datasets which are MNIST digit dataset and Caltech-10 image dataset. For MNIST digit dataset, 1000 instances per class are chosen from digits "7" and "9" respectively, which are the two visually confusing classes. For Caltech-10 image dataset, the confusing classes between "yo-yo" and "roulette-wheel" are selected where 60 instances per class are randomly chosen. Label-noise is randomly injected at the amount of 10%, 20%, and 30%, respectively. Samples with label noise in two datasets can be seen in Fig. 6. In order to exclude the possibility of a biased result caused by the instances of those two classes, we repeated the experiment 30 times and randomly selected the instances each time.

The proposed methods are compared with three state of the art label noise removing methods, i.e., ICCN-SMO [30], TC-SVM [8], and ALNR [7]. The settings of the hyper-parameters are based on its corresponding paper. Note that these latter three methods learn from the annotation from human expert to determine which samples are outliers, while in this respect, our method is completely 'unsupervised' in the sense that we do not assume the existence of such supervised information.

Table 1 gives the average results, where only the outlier detection accuracy is given. It can be seen that although no human expert correction, our proposed method achieves comparable results with the state-of-the-art methods. This shows that the proposed method captures well the characteristics of instances in spite of the deliberate disturbance imposed on. Moreover, the proposed CS-WAE method gets a higher accuracy compared with the CS-AE method which shows that the optimization method proposed in CS-WAE can reduce the effect of label noise on the feature space.

Note that our proposed method may suffer a problem that some of the correctly labeled instances are erroneously predicted as mislabeled samples. To further understand the behavior

Table 2 ER_1 (%) for CS-AE and CS-WAE (in bold) on MNIST dataset

Data size	Noise level		
	10%	20%	30%
N = 500	2.67 **2.40**	3.25 **3.00**	4.00 **3.71**
N = 1000	0.89 **0.78**	2.38 **2.13**	2.86 **2.43**
N = 2000	0.39 **0.39**	1.44 **1.31**	1.86 **1.64**
N = 4000	0.28 **0.28**	0.97 **0.84**	1.39 **1.29**

CS-WAE results are in bold for simple to compare

Table 3 ER_2 (%) for CS-AE and CS-WAE (in bold) on MNIST dataset

Data size	Noise level		
	10%	20%	30%
N = 500	14.00 **14.00**	17.00 **16.00**	18.67 **18.00**
N = 1000	11.00 **11.00**	13.00 **12.00**	14.00 **12.67**
N = 2000	4.00 **3.50**	8.25 **7.50**	9.33 **8.33**
N = 4000	3.50 **3.25**	4.63 **4.13**	6.75 **5.58**

CS-WAE results are in bold for simple to compare

of the proposed method, we conduct a more challenge experiment on the MNIST digits dataset where all of the class are considered. The size of training data is set within a range of {500, 1000, 2000, 4000}. We also use three performance metrics for evaluation, i.e.,

$$ER_1 = \frac{\#\text{ of correctly labeled instances which are removed}}{\#\text{ of correctly labeled instances}} \tag{9}$$

$$ER_2 = \frac{\#\text{ of mislabeled instances which are not removed}}{\#\text{ of mislabelled instances}} \tag{10}$$

$$NEP = \frac{\#\text{ of mislabeled instances which are removed}}{\#\text{ of removed instances}}. \tag{11}$$

ER_1 (Type 1 errors) reflects the percentage of correctly labeled instances which are wrongly removed while ER_2 (Type 2 errors) reflects the percentage of mislabeled instances which are not found out. The lower values of these two metrics get, the better model we obtains. As for NEP (noise elimination precision), it reflects the percentage of removed instances which are actually mislabeled, and a higher value of NEP means a more effective model.These three performance metrics are commonly used in the label noise cleaning problem [9].

Tables 2, 3 and 4 give the results. One can see that regardless of the noise level and training data size, the ER_1 performance of our method is less than 4%, which shows our class-specific feature representation rarely changes the correct label. More importantly, it can be observed that the ER_1 and ER_2 get lower while the NEP gets higher with the increase of training data size, indicating that our method is more effective when the size of training samples is larger. Compared with CS-AE, the CS-WAE gets better performance, especially in high label noise situation. This observation fully shows the benefit of our proposed optimization method in CS-WAE.

Table 4 NEP (%) for CS-AE and CS-WAE (in bold) on MNIST dataset

Data size	Noise level		
	10%	20%	30%
N = 500	78.18 **79.63**	86.46 **87.50**	89.71 **90.44**
N = 1000	91.75 **92.71**	90.16 **91.19**	92.81 **93.91**
N = 2000	96.48 **96.50**	94.10 **94.63**	95.44 **95.99**
N = 4000	97.47 **97.48**	96.10 **96.60**	96.63 **96.92**

CS-WAE results are in bold for simple to compare

4.3 Classification in the Presence of Label Noise

We conduct our classification experiments on the MNIST digit dataset [20] and the Caltech-10 image dataset [29], which are two popular classification benchmarks with ten classes. Totally 600 images with 60 per class are randomly selected in both datasets, and they are partitioned into training set and test set with equal number. We inject label noise at three levels, i.e., 10%, 20% and 30%. Note that in our setting only training data contains label noise, while the test set is kept clean.

In the implementation of the proposed methods, we try three methods based on CS-WAE which are: (1) directly classification by the minimum reconstruction (2) removing the label noise data before classification and (3) relabeling the label noise data before classification. These three methods are denoted as WAE, WAE-Remove, WAE-Relabel respectively. Note that WAE-Remove and WAE-Relabel methods need twice feature learning where the CS-WAE will be retrained after removing or relabelling the noise data, while WAE method only needs once.

We also learn a global PCA subspace using the whole training set (without partition for each class) and classify the test set in it using K-NN. This naive method is named PCA-KNN and is used as the baseline method. Besides these, we also compare our method with the two types of methods, both of which are state of the art methods that are designed to handle the label noise problem:

1. Label noise cleaning algorithms Including the classical PCA-based outlier detection method (PCA-Outlier) [36] and Complementary Fuzzy SVM based noise cleaning method (FSVM-Clean) [28].

2. Label noise robust algorithms Including L1-norm metric learning (L1-norm) [41], Bayesian metric learning (BML) [42], and Robust Neighbourhood Component Analysis (RNCA) [40].

Since the training samples are randomly selected in each experiment, we repeat thirty times for excluding the possibility of a biased result. And in each experiment, the related hyper-parameters of the comparison methods are selected by tenfold cross-validation. As for the proposed method, the hyper-parameters are set based on Sect. 4.1. The mean accuracy and standard deviation are reported in Fig. 7.

Figure 7 shows that the WAE-Relabel method achieves the best mean accuracy among the compared methods consistently at both low-level noise and high-level noise, indicating the effectiveness of relabeling the label noise data. When the training set is clean, our WAE method gets the best classification accuracy which shows that the minimum reconstruction error based classifier is a suitable choice. On the other hand, the baseline PCA-KNN method tolerates label noise to some degree when the noise level is relatively low, but its performance decreases significantly if the noise level is beyond some threshold (e.g., 20%). The PCA-

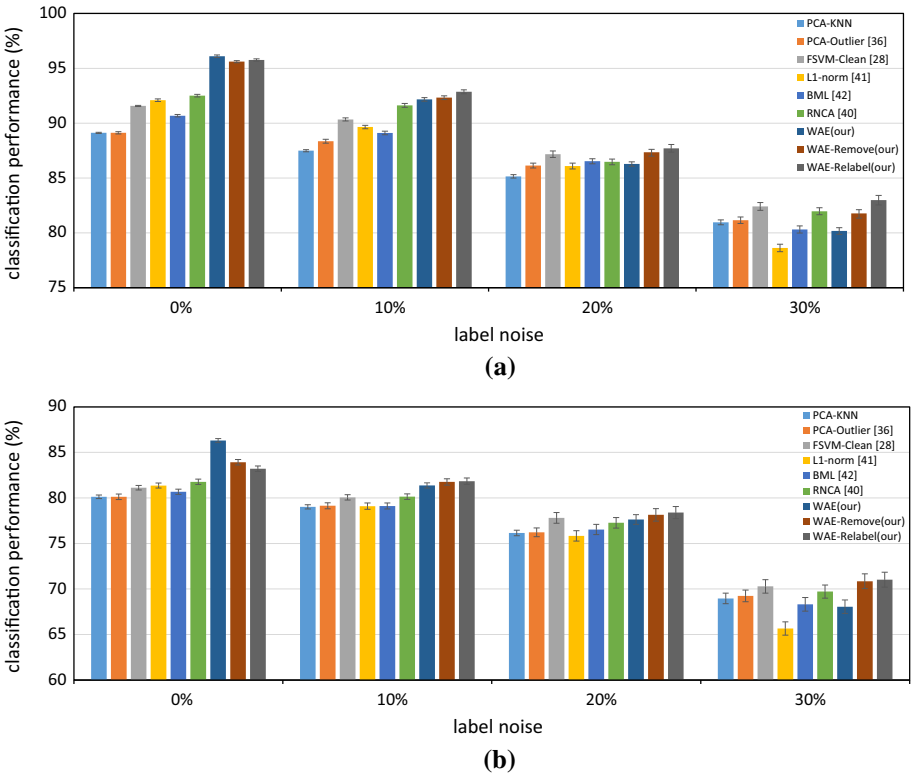

Fig. 7 Classification performance (%) on two dataset with varying degree of label noise. **a** Classification performance (%) on MNIST digit dataset, **b** classification performance (%) on Caltech-10 image dataset

Outlier [36] and FSVM-Clean method [28] achieve higher accuracy compared with the baseline method, as they have some built-in mechanism to deal with outliers and mislabeled instances. As for the second type of methods, i.e., those label noise robust algorithms [40–42], they achieve better results at 10% noise level compared with the state of the art first type methods (i.e., label noise cleaning methods, e.g., [28]). However, these methods perform worse at higher noise levels, especially in 30% label noise, highlighting the difficulty of obtaining reliable point estimation (e.g., L1-norm [41], RNCA [40]) under the high-level label noise.

Moreover, since the results provided by Fig. 7 are close to each other, the paired t test between the proposed WAE-Relabel method and other comparison methods is added for further analysis where the result of p values are shown in Table 5. Specifically, our method is compared with RNCA in the cases of 10% label noise while our method is compared with FSVM in the cases of 20% and 30% label noise (since RNCA and FSVM get the second best performance except the proposed method in its corresponding label noise level). It can be seen that the P value is smaller than 0.001 in the case of 10% label noise which means that our method has absolute improvement compared with RNCA method. For cases where label noise are higher (i.e. 20% and 30%), the P value are also smaller than 0.05 which show the differences between our method and FSVM-Clean method are statistically significant.

Table 5 The t test results for different models

Methods (noise level)	*P* value	
	MNIST	Caltech-10
WAE-relabel versus RNCA (10%)	< 0.001	< 0.001
WAE-relabel versus FSVM-clean (20%)	0.001	0.016
WAE-relabel versus FSVM-clean (30%)	0.003	0.027

5 Conclusion

In this paper, we proposed a simple but novel and effective method to deal with the label noise problem in data cleaning and classification tasks. The key idea of our method is to address the label noise problem from the perspective of feature learning by class-specific autoencoder. This is based on our observation (assumption) of the connection between two conceptually different problems: although one is the data contamination problem in the output space (label noise) while the other is in the input space (outliers), locally data points with noisy labels in some class are likely to be outliers of that class. We wish that this simple observation could help to inspire more methods to deal with the less-studied label noise problem. Extensive experiments on the MNIST and Caltech-10 datasets show that our proposed method outperforms several state of the art methods in data cleaning task and classification with label noise task.

Acknowledgements The authors thank the anonymous reviewers for their valuable comments and suggestions. This work is partially supported by National Science Foundation of China (61672280, 61373060, 61732006), AI+ Project of NUAA (56XZA18009), Jiangsu 333 Project (BRA2017377) and Qing Lan Project.

References

1. Abellán J, Masegosa AR (2010) Bagging decision trees on data sets with classification noise. In: International symposium on foundations of information and knowledge systems. Springer, pp 248–265
2. Aggarwal CC (ed) (2015) Outlier analysis. In: Data mining. Springer, Berlin, pp 237–263
3. Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks. In: Advances in neural information processing systems, pp 153–160
4. Biggio B, Nelson B, Laskov P (2011) Support vector machines under adversarial label noise. ACML 20:97–112
5. Brodley CE, Friedl MA (1999) Identifying mislabeled training data. J Artif Intell Res 11:131–167
6. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv (CSUR) 41(3):15
7. Ekambaram R, Fefilatyev S, Shreve M, Kramer K, Hall LO, Goldgof DB, Kasturi R (2016) Active cleaning of label noise. Pattern Recognit 51:463–480
8. Fefilatyev S, Shreve M, Kramer K, Hall L, Goldgof D, Kasturi R, Daly K, Remsen A, Bunke H (2012) Label-noise reduction with support vector machines. In: 2012 21st International Conference on Pattern Recognition (ICPR). IEEE, pp 3504–3508
9. Frénay B, Verleysen M (2014) Classification in the presence of label noise: a survey. IEEE Trans Neural Netw Learn Syst 25(5):845–869
10. Gupta K, Majumdar A (2017) Imposing class-wise feature similarity in stacked autoencoders by nuclear norm regularization. Neural Process Lett 48:1–15
11. Hawkins DM (1980) Identification of outliers, vol 11. Springer, Berlin
12. Hoz EDL, Hoz EDL, Ortiz A, Ortega J, Martnez-lvarez A (2014) Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps. Knowl Based Syst 71:322–338

13. Huber PJ (2011) Robust statistics. Springer, Berlin
14. Ipeirotis PG, Provost F, Wang J (2010) Quality management on amazon mechanical turk. In: Proceedings of the ACM SIGKDD workshop on human computation. ACM, pp 64–67
15. Jeatrakul P, Wong KW, Fung CC (2010) Data cleaning for classification using misclassification analysis. J Adv Comput Intell Intell Inform 14(3):297–302
16. Kamimura R, Nakanishi S (1995) Feature detectors by autoencoders: decomposition of input patterns into atomic features by neural networks. Neural Process Lett 2(6):17–22
17. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114
18. Krishna RA, Hata K, Chen S, Kravitz J, Shamma DA, Fei-Fei L, Bernstein MS (2016) Embracing error to enable rapid crowdsourcing. In: Proceedings of the 2016 CHI conference on human factors in computing systems. ACM, pp 3167–3179
19. Lab R, Gunnar Rtsch PD (2001) Soft margins for adaboost. Mach Learn 42(3):287–320
20. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
21. Li W, Wang L, Li W, Agustsson E, Van Gool L (2017) Webvision database: visual learning and understanding from web data. arXiv preprint arXiv:1708.02862
22. Liu T, Tao D (2016) Classification with noisy labels by importance reweighting. IEEE Trans Pattern Anal Mach Intell 38(3):447–461
23. Makhzani A, Frey B (2013) K-sparse autoencoders. arXiv preprint arXiv:1312.5663
24. Maria J, Amaro J, Falcao G, Alexandre LA (2016) Stacked autoencoders using low-power accelerated architectures for object recognition in autonomous systems. Neural Process Lett 43(2):445–458
25. Natarajan N, Dhillon IS, Ravikumar PK, Tewari A (2013) Learning with noisy labels. In: Advances in neural information processing systems, pp 1196–1204
26. Nettleton DF, Orriols-Puig A, Fornells A (2010) A study of the effect of different types of noise on the precision of supervised learning techniques. Artif Intell Rev 33(4):275–306
27. Pechenizkiy M, Tsymbal A, Puuronen S, Pechenizkiy O (2006) Class noise and supervised learning in medical domains: the effect of feature extraction. In: 19th IEEE international symposium on computer-based medical systems. CBMS 2006. IEEE, pp 708–713
28. Pruengkarn R, Wong KW, Fung CC (2016) Data cleaning using complementary fuzzy support vector machine technique. In: International conference on neural information processing. Springer, pp 160–167
29. Qian Q, Hu J, Jin R, Pei J, Zhu S (2014) Distance metric learning using dropout: a structured regularization approach. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 323–332
30. Rebbapragada UD (2010) Strategic targeting of outliers for expert review. Ph.D. thesis, Tufts University
31. Rifai S, Vincent P, Muller X, Glorot X, Bengio Y (2011) Contractive auto-encoders: explicit invariance during feature extraction. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 833–840
32. Rolnick D, Veit A, Belongie S, Shavit N (2017) Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694
33. Rtsch G, Schlkopf B, Smola AJ, Mika S, Onoda T, Mller KR (2000) Robust ensemble learning for data mining. In: Pacific-Asia conference on knowledge discovery and data mining, Current Issues and New Applications, pp 341–344
34. Sáez JA, Galar M, Luengo J, Herrera F (2014) Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. Knowl Inf Syst 38(1):179–206
35. Teng CM (2005) Dealing with data corruption in remote sensing. In: International conference on advances in intelligent data analysis, pp 452–463
36. Vidal R, Ma Y, Sastry S (2005) Generalized principal component analysis (GPCA). IEEE Trans Pattern Anal Mach Intell 27(12):1945–1959
37. Vidal R, Ma Y, Sastry SS (2016) Robust principal component analysis. In: Antman SS (ed) Generalized Principal Component Analysis. Springer, Berlin pp 63–122
38. Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on machine learning. ACM, pp 1096–1103
39. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408
40. Wang D, Tan X (2014) Robust distance metric learning in the presence of label noise. In: AAAI, pp 1321–1327
41. Wang H, Nie F, Huang H (2014) Robust distance metric learning via simultaneous l1-norm minimization and maximization. In: International conference on machine learning, pp 1836–1844

42. Yang L, Jin R, Sukthankar R (2012) Bayesian active distance metric learning. arXiv preprint arXiv:1206.5283
43. Yang T, Mahdavi M, Jin R, Zhang L, Zhou Y (2012) Multiple kernel learning from noisy labels by stochastic programming. arXiv preprint arXiv:1206.4629
44. Zhang W, Rekaya R, Bertrand K (2005) A method for predicting disease subtypes in presence of misclassification among training samples using gene expression: application to human breast cancer. Bioinformatics 22(3):317–325
45. Zhang W, Wang D, Tan X (2018) Data cleaning and classification in the presence of label noise with class-specific autoencoder. In: International symposium on neural networks
46. Zhu X, Wu X (2004) Class noise vs. attribute noise: a quantitative study. Artif Intell Rev 22(3):177–210