ECS Sensors Plus

The Electrochemical Society
Advancing solid state & electrochemical science & technology

REVIEW PAPER • OPEN ACCESS

Review–A Survey of Learning from Noisy Labels

To cite this article: Xuefeng Liang *et al* 2022 *ECS Sens. Plus* **1** 021401

View the article online for updates and enhancements.

ECS Sensors Plus, 2022 **1** 021401

Review–A Survey of Learning from Noisy Labels

Xuefeng Liang,[z] Xingyu Liu,[z] and Longshan Yao[z]

School of Artificial Intelligence, Xidian University, People's Republic of China

Deep Learning has achieved remarkable successes in many industry applications and scientific research fields. One essential reason is that deep models can learn rich information from large-scale training datasets through supervised learning. It has been well accepted that the robust deep models heavily rely on the quality of data labels. However, current large-scale datasets mostly involve noisy labels, which are caused by sensor errors, human mistakes, or inaccuracy of search engines, and may severely degrade the performance of deep models. In this survey, we summaries existing works on noisy label learning into two main categories, Loss Correction and Sample Selection, and present their methodologies, commonly used experimental setups, datasets, and the state-of-the-art results. Finally, we discuss a promising research direction that might be valuable for the future study.

[DOI: 10.1149/2754-2726/ac75f5]

Manuscript submitted April 23, 2022; revised manuscript received May 24, 2022. Published June 29, 2022.

In recent years, deep learning has made great progress on many challenging tasks, such as face recognition, image segmentation, pedestrian detection, speech recognition, machine translation, to name a few. Without exception, all fields, which have successfully applied deep learning, rely on large-scale datasets with manual annotation, e.g. ImageNet,[1] MS-COCO,[2] Librispeech,[3] etc. Although emerging research inclines toward large-scale unsupervised learning methods, current robust deep models still heavily rely on datasets with high quality labels. Unfortunately, there are only a few such datasets available because of following two main reasons: (1) The construction of a large-scale dataset is time consuming and requires huge labor. (2) It is extremely difficult to label samples in some professional fields due to the data ambiguity, for which even experts struggle to reach consensus, e.g. medical images, speech emotion, lips, face aesthetics, etc. Above reasons hinder the further development and application of deep learning. To address the first problem, some recent low-cost alternatives have been proposed to build large-scale datasets. A common approach uses the crawler technology to collect a huge amount of data through search engines from social media, and automatically labels samples based on relevant description texts and tags. Another method is Amazon's Mechanical Turk. These methods inevitably introduce noisy labels to worsen the quality of data. According to some statistics, the proportion of data with noisy labels in web-crawled datasets is around 8.0%–40%.

It has been shown that deep neural networks may overfit noisy labeled data (Corrupted label),[4,5] which significantly degrades the generalization ability and robustness of deep models. Unfortunately, existing regularization methods, e.g. data augmentation, weight decay, dropout, and batch normalization, do not alleviate deep models overfitting noisy labels very much. Therefore, improving the robustness of deep models on noisy labeled data becomes a crucial and promising research topic.

With the increasing studies on noisy label learning in academic community, many valuable methods have emerged in this field. However, the problem of noisy labels spans a wide range of fields, such as computer vision, natural language processing, speech recognition, and so on. Hence, the perspectives and technical lines of this problem become very diverse. It significantly increases the reading difficulty and the learning costs. In order to assist researchers to understand the problem quickly, this survey focuses on "how to reduce the impact of noisy labels for deep models" in the field of Computer Vision. We group and link the representative works published in recent artificial intelligent conferences, including CVPR, ICCV, ICML, ICLR, NIPS, etc., and discuss their methods, finally, suggest possible solutions.

The rest of the paper is organized as follows: Section 2 gives the preliminary knowledge of the noisy label problem. Section 3 lists the existing works in details. Section 4 introduces the public datasets commonly used in noisy label learning. Section 5 discusses the problems of the existing methods and possible solutions. Finally, Section 6 summaries this survey.

Preliminary Knowledge

In this survey, we define noisy label samples as samples whose given labels are not consistent with their true labels. Therefore, the noise only exists in labels rather than data themselves when using the technical term "noisy samples" or "noisy labeled data" in this survey.

Noisy labels in deep learning.—The goal of deep learning tasks under supervised learning is to learn a mapping f_θ: $x_i \rightarrow y_i$ from the dataset $D = \{(x_i,\ y_i)|i = 1,\ 2,\ 3,\ \cdots,n\}$, where θ denotes a set of parameters of this mapping. In many cases, it is also called as the parameters of the deep model. Usually, an objective function (also called loss function), $L(f_\theta(x_i),\ y_i)$, is designed for regularizing the learning process. Thus, the optimal mapping can be reached by finding the best θ^* through the loss function,

$$argmin_\theta \sum_{i=0}^{n} L(f_\theta(x_i),\ y_i). \qquad [1]$$

If the dataset D contains certain noisy labels $(x_i,\ \tilde{y}_i)$, $\tilde{y}_i \neq y_i$, then they will mislead the mapping f_θ to an incorrect loss function $L(f_\theta(x_i),\ \tilde{y}_i)$. Therefore, the learned parameters $\tilde{\theta}^*$ will differ from θ^*. To alleviate the harm caused by noisy labels, the essential idea is to enable deep models to find θ^* through a noise-tolerant training strategy.

Sources and types of noisy label.—To better understand the nature of noisy labels, we firstly discuss the sources of noisy labels, then dig into their characteristics, finally group them into four categories.

Sources of noisy label.—

(1) Some data are mislabelled due to their own ambiguity and the cognitive bias of the annotators. When constructing a dataset, we usually use crowdsourcing. The annotators may give inconsistent labels to a sample due to their cognitive bias. Then, the "majority voting" is widely applied to obtain the final labels. However, the number of annotators is relatively few to

[z] E-mail: xliang@xidian.edu.cn; yaolongshan@stu.xidian.edu.cn; 975634372@qq.com

reduce the cost, e.g. 3 ∼ 5 persons. But the amount of data in large-scale databases is rather huge. Therefore, such cognitive bias might be amplified to introduce more noisy labels. In addition, there are many ambiguous data in medical images, emotional speech, lip recognition, face aesthetics, etc. These data are very difficult to be distinguished into just one category, even for experts.

(2) To construct a large-scale dataset, it is commonly to use web search engines to collect data from social networking sites. The labels usually come from the surrounding texts. Although this method is efficient and simple, it will naturally introduce a large amount of noisy labels due to the diversity and randomness of the surrounding texts. Some noisy labels are somewhat correlated with the data itself because of the sementic ambiguity. Such correlation will introduce out-of-vocabulary noisy into the dataset. For example, we expect the samples of ladybug category in the dataset to be the insect, whereas the search engine may give a result of animated characters.

Types of noisy label.—To facilitate the study of noisy label problem, many artificial noise types have been designed, such as pair noise (pair), symmetric noise (symmetric), asymmetric noise (asymmetric), real-world noise, etc. Please note that these artificial label noises are low-cost alternatives. As huge amounts of labor power would be consumed to verify real-world data labels, researchers designed these artificial noises on existing datasets to simulate the realistic noise in real world. Although unlikely to be perfectly, they are still valuable for evaluating the robustness of proposed methods.

Many AI research fields, such as computer vision, natural language processing, speech processing, etc., also encounter the similar noise types and noise patterns. For the visual classification problems, the input is a matrix and the label is a number representing the true class. For language and speech recognition problems, the input is a sequence of signals and the label is also a number. As the causes and patterns of noisy label in these fields are very similar, we survey the label noise problem by focusing on computer vision hereinafter.

Pair Noise

Pair noise type is to flip the labels between adjacent two categories according to a fixed ratio.[6] This noise type was designed in the early age of research on noisy label, which simulated the scenario where labels were mistakenly flipped between similar class-pairs. For example, a lot of labels are flipped between knitwear and sweater in clothing1M dataset. As not taking the similarity between classes into account, pair noise has been gradually replaced by asymmetric noise in recent research. Figure 1a shows an example with a noise rate of 40%. 60% data in the first category keep their original labels, labels of the other 40% data are transferred to the second category. Similarly, 60% data in the fifth category keep their original labels, labels of the other 40% data are transferred to the first category. The row represents the correct class labels of data and the column represents the given labels.

Symmetric Noise

Symmetric noise is also known as uniform noise. It keeps a certain percentage of original labels, and the rest are uniformly flipped to other categories. This noise type is designed to simulate the random noise in real world, which is often caused by random errors of web crawling or manual annotation. It does not take into account the similarity between classes. Figure 1b shows an example of symmetric noise with a noise rate of 50%. The first class (the first row) keeps 50% of the correct labels, and the other 50% of labels are equally distribute into the other four categories with the same proportion (12.5%). It is analogous to the other categories.

Asymmetric Noise

Asymmetric noise flips labels according to the given similar class-pairs. It is designed to better simulate the real-world noisy label.[7] For example, class-pairs in CIFAR-10 dataset: TRUCK → AUTOMOBILE, BIRD → PLANE, DEER → HORSE; class-pairs in MNIST dataset: 2 → 7, 3 → 8, 7 → 1, and 5 → 6. Analogous to pair noise, this noise type was designed to simulate the noise label which is caused by sample ambiguity in real world. When annotators cannot well distinguish two classes during the label annotation, they might mislabel samples, e.g., one can easily mislabel a picture of dolphin as a whale. Figure 1c shows the examples. 60% of original labels in the first class are kept, and the other 40% of the labels are transferred to the third class. It is analogous to the other categories.

Real-world Noise

The real-world noise has 2 subcategories: in-distribution noise and out-of-distribution noise.

For in-distribution real-world noise, all data and their labels are in the scope of this dataset. For example, a sample comes from MNIST or CIFAR. If it is mislabeled, the true label of this sample still belongs to one of classes in MNIST or CIFAR.

For out-of-distribution real-world noise, the true label of the mislabled sample is not included in the scope of original dataset. For example, a clean sample in MINIST is replaced by an image in CIFAR with its true label, which is a class in CIFAR.

Controlled Web Label Noise

The controlled web label noise is proposed to mimic the real-world noise. It firstly collects data with incorrect web labels by Google image search from two sources: text-to-image and image-to-image. Then, after deduplicating the images which are similar to the ones in the testing dataset, it replaces p% of the original training samples with the collected noisy data, where $p \in [0, 100]$. Similar to symmetric noise, p is uniform across classes. This noise type has a higher similarity to the true positive images and can freely control the noise rate.

Existing Methods of Noisy Label Learning

Thanks to the development of deep learning in recent years, many valuable studies for noisy label learning problem has been emerged.[8,9] These methods have different ideas, perspectives and strategies. Each of them has its own advantages. In this survey, we categorize the major ideas in this area into the following groups: *loss*

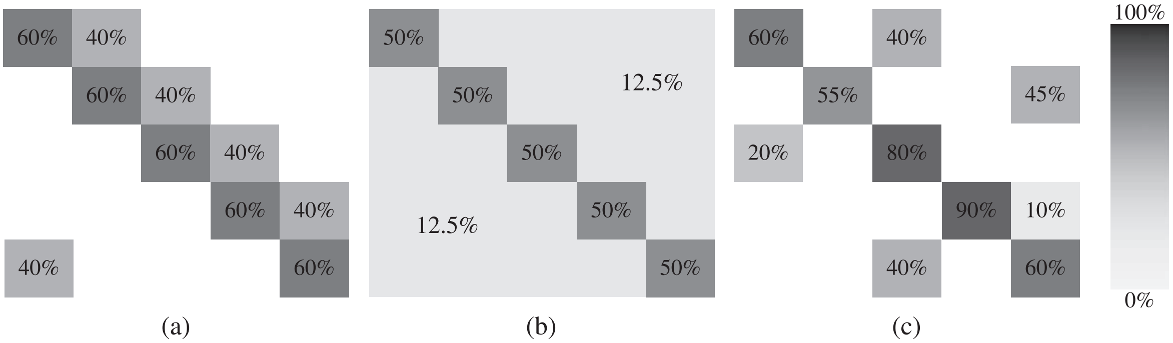


Figure 1. Transition matrix of different noise type: (a) Pair, (b) Symmetric, (c) Asymmetric.

correction methods and *sample selection* methods. Each of them will be described in detail below. Furthermore, we summarize the pros and cons of those approaches in Table I.

Loss correction.—Loss correction methods are used to reduce the effect of noisy labels during network training stage by directly modifying (or adjusting) the losses through various methods. One of the advantages of these approaches is that they can be used in any model. Most of the methods treat the noisy samples and clean samples in the same way. They usually add a regularization item in loss function to penalize the low confident prediction, which may be related to noisy samples, or correct the network prediction by multiplying the estimated label transition matrix. This category includes: *estimating the noise transition matrix, designing robust loss function for noisy label, designing robust network structure for noisy label, modifying the noisy labels (pseudo-labels)*, and *adjusting the weights of the samples*.

Estimating the noise transition matrix.—This kind of methods usually constructs a noise transfer matrix to determine the probability of noise transfer between different classes, which is applied when calculating the cross-entropy loss. It will adjust the loss by multiplying the estimated noise transition matrix with the softmax output during forward propagation. Several methods have been proposed to estimate the transition matrix. Patrini G, Rozza A, Krishna Menon A, Nock R and Qu L[7] estimate this matrix using a pre-trained model. It firstly does a pre-train without loss correction, and estimates the noise transition matrix using the softmax output of the network. Then, it re-trains the model and carries out the loss correction according to the estimated noise transition matrix. Hendrycks D, Mazeika M, Wilson D and Gimpel K[10] use a clean validation set to calculate the transition matrix, while Sukhbaatar S and Fergus R[11] propose the use of the difference between the transition matrices calculated from clean and noisy data. Reed S E, Lee H, Anguelov D, Szegedy C, Erhan D and Rabinovich A[12] uses a transition matrix combined with a regularized loss which uses both of the noisy labels and labels predicted by the model. Goldberger J and Ben-Reuven E[13] use the expectation-maximization (EM) algorithm to find the optimal parameters of both network and the noise. The Dual-T method,[14] on the other hand, estimates the noise transfer matrix in two steps to simplifies the problem. Firstly, estimate the confusion transfer matrix from the clean labels to the intermediate category labels (network prediction labels), secondly estimate the confusion transfer matrix from the intermediate category labels to the noisy labels.

Robust loss function.—The purpose of designing noise-robust loss functions is to modify or lower the loss of samples whose labels may be incorrect while calculating the loss value. These methods focus on designing new loss functions that aim to mathematically lower the impact of noisy labels during backpropagation. Manwani N and Sastry P S[15] shows that 0-1 losses are more noise-tolerant than commonly used convex losses, such as Mean Absolute Error (MAE), Improved MAE (which is a weighted MAE). From perspective of math, it is a sufficient condition for a loss function in binary classification problem. Generalized Cross-Entropy (GCE)[16] loss applies a Box-Cox transformation to probabilities (A power function of probability). Motivated by KL,[17] the authors of Symmetric Cross-Entropy[18] found that the traditional cross-entropy loss has the following problems: (1) The value of Cross-entropy only depends on the probability of true class. (2) Simple samples are more likely to be learned (and overfitted under high noise). To address this problem, the authors adds a reverse cross-entropy term to the conventional cross-entropy loss. APL[19] divides the existing loss functions into "Active" and "Passive", which are separately used at different stages of training. In addition to the above distance-based loss function, the loss function based on information entropy also achieved good results. For example, the function L_DMI[20] is not only monotonous in information, but also relatively invariant.

Robust network structure.—Noise-robust network structures deal with noisy labels by designing specific layers or branches, which can facilitate the network for identifying or rectifying noisy labels. Lee K H, He X, Zhang L and Yang L propose the CleanNet[21] that uses a predefined reference subset. The visual features of the reference subset are extracted using auto-encoder and each new training sample is compared with the features from the reference set. Based on the distance, a weight is set for each training sample and the weighted cross-entropy is calculated. This method uses an auto-encoder to learn a prototype from each class. For each input, the distance between input and class prototype is calculated to determine its category. Meanwhile, the method can verify whether the input is noisy or not and gives a low weight for noisy one. MetaCleaner[22] divides the traditional training process into two steps: (1) Estimate the confidence of each sample through the network. (2) Generate a set of clean training samples by aggregating the confidence scores. ClothingNet[23] sets multiple tasks for the network. It requires the network is able to predict both the category and noise type of the input. Then, it will calculate the posterior probability of each sample based on the noise type prediction. Self-learning[24] incorporates an additional clustering module to the network, which can estimate the class prototypes and relabel the training data depend on the similarity between the data features and class prototypes. SIGUA[25] cannot only estimate the noise transfer matrix, but also adjust the gradient of the clean label data in each batch to gradually reduce the learning rate of the noisy labeled data.

Correction of noisy labels.—This kind of methods aims to estimate the true label of each sample, and then replace the mislabeled labels by the estimated ones. Inspired by the cognitive continuity of neural networks, SELF[24] considers that the noisy labeled data have the same feature distribution with the clean data. Thus, the noisy labels could be gradually corrected by the accumulated predictions of the model based on the training data in each epoch. In each iteration, an updated model is trained using the corrected labels. It uses the corrected labels for training and reduces the effect of noisy labels. P-correction (PENCIL)[26] tries to obtain a more accurate pseudo label by treating the pseudo label (the corrected label) as an independent parameter, which is updated during training (just like the network parameters). However, this method targets at all training data during the updating. This makes the network often mistakenly "correct" true labels to incorrect pseudo labels. Joint-optimization[27] uses a single network to simultaneously train and correct noisy labels. To lower the possibility of mistaken correction, it adds a regularization on the loss function. ELR[28] reveals that the network does not overfit noisy labels in the early training stage, then adds a regularization to prevent the network from memorizing the noisy labels.

Sample selection.—Sample selection methods aim to directly modify (or adjust) the loss and lower the harm of noisy labels to the network. A distinctive characteristic of sample selection is that it will explicitly divide the training data into a clean subset and a noisy subset before training. Afterwards, the network is trained on two subsets separately. The commonly used criteria of dividing data are: Small-loss criterion,[6] Gaussian mixture model GMM,[29,30] Bayesian mixture model BMM,[29] etc. Based on the training strategy of the filtered subsets, sample selection methods can be further divided into 2 categories: Non-Combined and Combined methods.

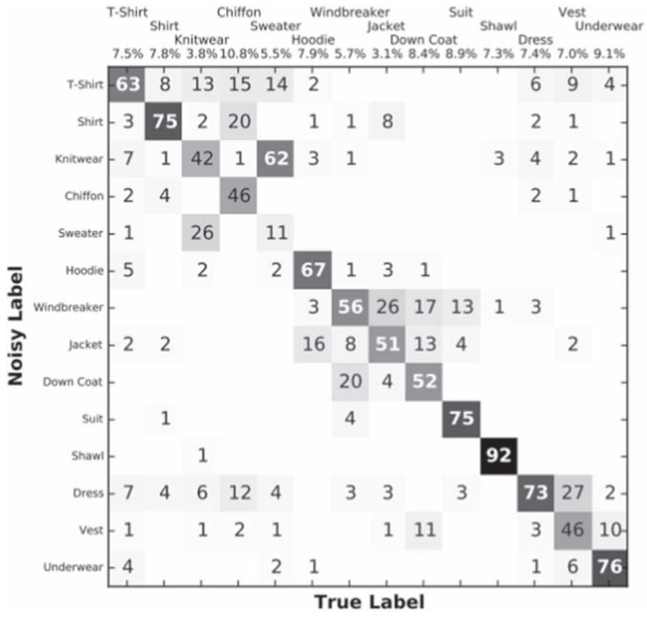
Non-Combined.—This kind of methods focuses on utilizing the clean data for training. Decouple[31] proposes to decouple the problem of "how to update" to "when to update" during training. Based on the inconsistent information of the network, the proposed update strategy first randomly initializes two networks and do updating only when the two networks disagree during the subsequent training process. Co-teaching[6] is a representative method based on the idea of Co-training.[32] This method will maintain two networks. During training, each network calculates the losses and selects a certain

Table I. The main approaches to deal with noisy labels.

Approaches	Methods	Advantages	Disadvantage
Loss Correction	Estimate the noise transition matrix	Easy to implement.	Difficult and complex to estimate the transition matrix in practice.
	Robust loss function	Easy to be added to most training models, and theoretically guaranteed.	Results are often not optimal. Other auxiliary methods are required.
	Robust network structure	Targeted structures can be designed for different noise types, with many selectable heuristic components.	Not applicable to all datasets.
	Correction of noisy labels	Able to fully use all training data.	The effectiveness of the label correction is not guaranteed. Incorrect relabeling may further affect models.
Sample Selection	Non-Combined	Able to select clean samples.	Not competitive with state-of-the-arts due to the noisy samples are not used.
	Combined	Able to select clean samples. The state-of-the-art performance.	The partition criteria are mostly empirical, and lack of theoretical guarantee.

number of small-loss samples. Then, one network feeds the selected samples to another network for further training. This method considers the data with smaller losses as clean data. It only uses the clean ones for training to avoid the negative effect of noisy labels. On the other hand, Mandal D, Bharadwaj S and Biswas S[33] add the self-supervised idea to Co-teaching to improve the classification accuracy of clean data. This further improves the quality of the clean subset and results in a better performance of the model. Inspired by Co-teaching and Decouple, Jo-CoR[34] introduces the ”agreement” idea for two networks. It assumes that different models trained on the same dataset will agree on most of the clean samples but likely disagree on noisy labeled samples.[35,36] This idea improves the quality of data filtering. To regularize the ”agreement”, a contrast loss (JS divergence) is applied between the two networks. Finally, it filters out the clean data based on the small-loss criterion. MentorNet[37] applies the idea of course learning (Motivated by human learning models) to two networks (one teacher and one student) to achieve a progressive learning from easy data to difficult data that may have incorrect labels.

Combined.—Methods of this kind firstly divide the dataset into clean subset and noisy subset. Then, they are going to use both of them with different training strategies instead of dropping the noisy subset. Most of the training strategies for noisy subset are based on semi-supervised learning. It treats noisy labeled data as unlabeled data. As there have been many well developed semi-supervised learning methods, combined methods usually focus on more effective data-filtering algorithms to achieve better results. Also inspired by the Co-training,[32] DivideMix[30] uses Gaussian mixture model to separate clean samples and noise-labeled samples. It treats noise-labeled samples as unlabeled data and training them with MixMatch[38] which is an excellent algorithm in semi-supervised learning for training. It firstly divides the training data into a clean subset and a noisy subset. Then, it applies semi-supervised learning for the noisy subset without using the given noisy labels. DSOS[39] uses the entropy of the interpolation of prediction and given label to distinguish clean, in-distribution (ID) noise and out-of-distribution (OOD) noise. Then, it corrects the labels for ID samples and proposes a dynamic softening strategy for OOD samples to lower the harm of noisy labels.



	T-Shirt	Shirt	Knitwear	Chiffon	Sweater	Hoodie	Windbreaker	Jacket	Down Coat	Suit	Shawl	Dress	Vest	Underwear
	7.5%	7.8%	3.8%	10.8%	5.5%	7.9%	5.7%	3.1%	8.4%	8.9%	7.3%	7.4%	7.0%	9.1%
T-Shirt	63	8	13	15	14	2						6	9	4
Shirt	3	75	2	20		1	1	8				2	1	
Knitwear	7	1	42	1	62	3	1				3	4	2	1
Chiffon	2	4		46								2	1	
Sweater	1		26		11									1
Hoodie	5		2		2	67	1	3	1					
Windbreaker						3	56	26	17	13	1	3		
Jacket	2	2				16	8	51	13	4			2	
Down Coat							20	4	52					
Suit		1					4			75				
Shawl			1								92			
Dress	7	4	6	12	4		3	3		3		73	27	2
Vest	1		1	2	1			1	11			3	46	10
Underwear	4				2	1						1	6	76

Figure 2. Confusion matrix between clean and noisy labels of Clothing1M dataset.

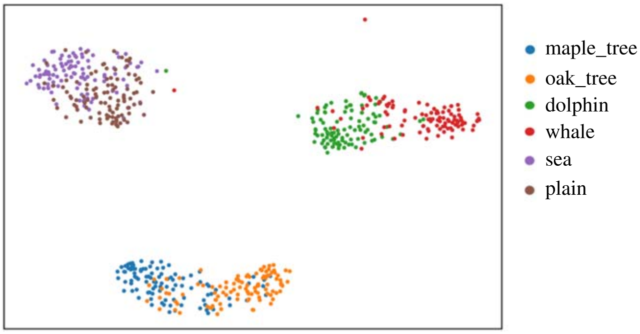


Figure 3. The visualization of the feature distributions of three similar class-pairs in CIFAR-100 using t-SNE.

Experimental Setting

Both synthetic noise datasets and real-world noise datasets are commonly used for noisy label learning.

Synthetic datasets.—Synthetic datasets are artificially designed to simulate the label noise, which include several different noisy label types as listed in Section "Types of noisy label". As can be well controlled, they are good to evaluate the generalization ability of testing models. Often, some small-scale clean datasets are used as the base datasets, such as MNIST,[40] CIFAR-10,[41] CIFAR-100,[41] and then, the correct labels are flipped into other classes to simulate the noisy label data. The advantage of synthetic dataset is that the noise rate and noise type can be precisely controlled. It helps us to effectively evaluate noisy label learning methods from varied perspectives. However, the distribution of synthetic noisy label data differs from fit the real-world ones. Specific datasets are as follows.

CIFAR-10 includes 10 classes, each class has 5,000 training images and 1,000 test images. The size of image is 32×32. CIFAR-100 includes 10 super-class, one super-class has 10 sub-class. Each sub-class has 500 training images and 100 test images. The size of image is same with CIFAR-10. Most of the methods validate symmetric noise and asymmetric noise on them, the noise rate often varies from 20% to 80%. The Red Mini-ImageNet is a recent benchmark of controlled web-crawled datasets for coarse-grained image classification provided by Jiang L, Huang D, Liu M and Yang W.[42] It provides various ratios of web-crawled noises which contains both In-vocabulary and Out-vocabulary noises. This dataset has images of size 84×84 with 100 classes from ImageNet.

Real-world datasets.—The real-world datasets are often builded by low-cost data collection methods, such as crawlers and search engines. The most representative ones are Food-101N,[21] Clothing1M[23] and WebVision1.0,[43] to name a few. The real-world datasets have a wider variety and larger number of training data with more complex noise types and uncontrolled noise rates. Besides, they often contain both in-distribution noise and out-of-distribution noise.

Clothing1M contains 14 classes including 1 million training images acquired from online shopping websites with labels generated by surrounding texts provided by sellers. The image size of this dataset is not uniform. Commonly, the images are resized to 256 × 256 for training. This dataset is the only one that has relatively complete information of noise transfer matrix. Figure 2 shows the complicated noise distribution of Clothing1M. It contains noise type like pair noise, such as Windbreaker, Chiffon and Shirt, etc. It also includes some noise type similar to symmetric noise, such as Hoodie, T-Shirt, Dress and Vest. Besides, it contains some radome noise distribution which does not belong to pair, symmetric or asymmetric type. For example, almost each class has 1% of samples that flip their labels to unrelated classes.

Table II. Accuracy comparisons on CIFAR-10 and CIFAR-100 datasets with symmetric noise.

Dataset *symmetric*	CIFAR-10			CIFAR-100		
Method	20%	50%	80%	20%	50%	80%
Standard CE	86.8	79.4	62.9	62.0	46.7	19.9
Bootstrap(2015)[12]	86.8	79.8	63.3	62.1	46.6	19.9
F-correction(2017)[7]	86.8	79.8	63.3	61.5	46.6	19.9
Co-teaching+(2019)[47]	89.5	85.7	67.4	65.6	51.8	27.9
P-correction(2019)[26]	92.4	89.1	77.5	69.4	57.5	31.1
Meta-Learning(2019)[48]	92.9	89.3	77.4	68.5	59.2	42.4
M-correction(2019)[29]	94.0	92.0	86.8	73.9	66.1	48.2
DivideMix(2020)[30]	**96.1**	94.6	93.2	77.3	**74.6**	60.2
ELR+(2020)[28]	95.8	**94.8**	**93.3**	**77.6**	73.6	**60.8**
Co-learning(2021)[49]	92.5	84.8	63.5	66.7	55.0	36.2
DSOS(2022)[39]	92.7	87.4	54.3	75.1	66.2	32.4

Table III. Accuracy comparison on Food-101N dataset.

Food-101N	
Methods	Acc.
Standard CE	84.03
CleanNet[21]	83.95
Decoupling[31]	85.53
Co-teaching[6]	61.91
Co-teaching+[47]	81.61
JoCoR[34]	77.94
Jo-SRC[50]	86.66
Co-learning[49]	87.57
DSOS[39]	**87.70**

Food101-N[21] is a large image dataset containing about 310 ,009 training images and 25, 000 testing images of food recipes classified into 101 classes. Similar as Clothing1M dataset, the images are resized to 256×256 for training. It is based on the Food101 dataset,[44] but has more images and a higher noise rate, 20%.

WebVision is a large-scale dataset with 2.4 million images collected from the web using the 1,000 concepts in ImageNet ILSVRC12.[1] The images also have varied sizes, then are commonly resized to 256×256 for training. Its estimated noise rate is about 20%. For ease of comparison with other methods, the test subset only contains the first 50 classes.

Results of State-of-the-Art Methods

In Section "Existing Methods of Noisy Label Learning", we categorized the mainstream methods into six types and briefly summarized the pros and cons of each type. In this section, they are compared in terms of effectiveness on common datasets. As these methods used different backbones and datasets in their own papers, we try our best to carry out fair comparisons for the validation,.

Table II shows the SOTA results of methods, which use PreActResNet18 (PRN18) as backbone, on CIFAR-10 and CIFAR-100. Table III and Table IV shows the SOTA results of methods, which use ResNet50 as backbone, on Food-101N and Clothing1M. Table V shows the SOTA results of methods, which use Inception-Resne as backbone, on Webvision and ILSVRC12. For a fair comparison, methods[45] and[46] are excluded, because they need auxiliary clean validation sets. The source code used to reproduce the experimental results can be found in the original paper.

Table IV. Accuracy comparison on Clothing1M dataset.

Clothing1M	
Methods	Acc.
Standard CE	69.21
F-correction[7]	69.84
Joint[27]	72.16
Meta-Learning[48]	73.47
P-correction[26]	73.49
DivideMix[30]	74.76
ELR+[28]	**74.81**
JNPL[51]	74.15
DSOS[39]	73.63

Results listed in Table II show that the performances of noise label learning methods on symmetric label noise in CIFAR datasets has been gradually improved in recent years, not only on low noise rates but also high noise rates. Although the loss correction methods can theoretically guarantee the convergence of model training. The instability of model training significantly degrades their performances. Instead, the sample selection methods demonstrate a considerable effectiveness. More than half of the best methods belong to this kind of methods. It indicates that filtering out the noisy labeled data and improving the quality of clean subset before training are more effective strategies. In addition, another helpful strategy is to utilize the noisy subset using data augmentation, semi-supervised methods and unsupervised methods, and so on. Because these methods can also learn certain useful information from noisy labeled data to alleviate the overfitting to such data.

Tables III and IV show the results on the real-world noisy labeled datasets. One can observe that the difference in performance among these methods is not as great as the difference on the simulated noise types. As shown in the Fig. 2, the distributions of noise labels in real-world datasets are more complex than the symmetric noise. The simulated noise types usually create noisy labels regularly into other classes with well controlled rules. By contrast, the real-world noise is more random. In addition, the real-world noisy labels tend to appear in ambiguous data. There is a higher possibility of data between similar classes are mislabeled. Figure 3 shows the distributions of three similar class-pairs in CIFAR-100, we should consider that labels are more likely to be mistakenly flipped between similar class (e.g. maple_tree and oak_tree). As few existing methods can effectively handle such complicated cases, they often perform worse on real-world datasets than the simulated noise datasets, meanwhile, the difference in performance among them is not significant.

In conclusion, Bootstrap,[12] F-correction,[7] P-correction,[26] M-correction,[29] ELR+[28] do not distinguish training data into clean and noisy subsets. They may mistakenly rectify the losses of clean data and introduce new noisy labels into training data. Among them, Bootstrap uses a transition matrix approach. It works well on simulated noisy labels in small-scale datasets, but performs worse on real-world datasets. Co-teaching+[47] uses the sample selection method. However, it only uses the selected clean data with smaller loss for training without utilizing the data with larger loss. Meanwhile, finding a feasible threshold, T, to define the small loss is very challenging. DivideMix divides clean labels and noisy labels by fitting a mixed Gaussian distribution, and uses both clean data and noisy labeled data for training. However, GMM may mistake the hard noisy labeled samples and the hard clean samples because of the little difference in their losses. We observed that the hard samples might be the bottleneck of existing sample selection methods to further improve their performances because the training losses of hard samples are neither small nor significantly different. Up to now, very few studies address this issue, no efficient and effective solution can handle it either.

Table V. Top-1 (Top-5) accuracy comparisons on WebVision1.0 and ILSVRC12 datasets.

Dataset Method	WebVision		ILSVRC12	
	Top-1	Top-5	Top-1	Top-5
F-correction(2017)[7]	61.12	82.68	57.36	82.36
Decoupling(2017)[31]	62.54	84.74	58.26	82.26
D2L(2018)[52]	62.68	84.00	57.80	81.36
MentorNet(2018)[37]	63.00	81.40	57.80	79.92
Co-teaching(2018)[6]	63.58	85.20	61.48	84.70
Iterative-CV(2019)[53]	65.24	85.34	61.60	84.98
DivideMix(2020)[30]	77.32	91.64	**75.20**	**90.84**
ELR+(2020)[28]	**77.78**	91.68	70.29	89.76
DSOS[39]	77.76	**92.04**	74.36	90.80

We think in the future study on noisy label learning, it might be wise to focus more on these hard samples. We define ”hard data” as data that distribute close to the decision boundary. As close to the decision boundary between no less than two categories, hard data have some shared features of these categories. Thus, they have relatively large losses whether their labels are clean or noisy. This inspires us to further divided training dataset into three subsets: clean subset, hard subset and noisy subset. There should exist an order of losses of three subsets: losses of noisy subset $>$ losses of hard subset $>$ losses of clean subset. To obtain three subsets, a straightforward idea is to find two thresholds T_1 and T_2 for partition as below, where $T_1 < T_2$.

Losses of easy subset $< T_1$,
$T_1 <$ Losses of hard subset $< T_2$,
Losses of noisy subset $> T_2$.

Nevertheless, selecting the feasible T_1 and T_2 is a non-trivial task. We think either finding feasible T_1 and T_2 or precisely filtering out hard samples will be a valuable and promising research direction.

Conclusions

Noise label learning aims to investigate how to use datasets with noisy labels for deep model training. Specifically, it focuses on how to minimize the negative impact of noise labels to deep models and help them to learn correct information from training data effectively. Its great value is that it could significantly reduces the cost of building large-scale datasets and lower the heavy reliance of deep models on high quality training data. Furthermore, a robust noise label learning also impacts other machine learning fields, such as semi-supervised and unsupervised learning. It can extend deep learning to a wider variety of applications.

In this survey, we summarize the existing methods and ideas for noise label learning problem. We categorize them into two major groups, loss correction and sample selection, and analyze their main ideas, advantages and disadvantages. Although these ideas focus on image classification task, they are quite general and can be transferred to other fields. Meanwhile, we have seen increasing achievements and interests in this problem, but there is still much room for improvement. For example, how to more accurately separate clean data and noisy labeled data, particularly in real-world noise datasets. For further research on this issue, we give a consideration on how to distinguish noisy hard samples from noisy labeled simples and use them in different ways. In addition, the out-of-distribution noisy labels should receive more attention in the future work.

ORCID

Xuefeng Liang https://orcid.org/0000-0002-1448-0477

References

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Commun. ACM*, **60**, 84 (2012).
2. T. Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context.” *ECCV* (2014).
3. V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, *ICASSP*, 5206 (2015).
4. C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, *Commun. ACM*, **64**, 107 (2021).
5. D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, and Y. Bengio, “A closer look at memorization in deep networks.” *ICML*(PMLR), 233 (2017).
6. B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama, “Co-teaching: Robust training of deep neural networks with extremely noisy labels.” *NeurIPS* (2018).
7. G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, “Making deep neural networks robust to label noise: A loss correction approach.” *CVPR*, Piscataway, NJ (IEEE), 1944 (2017).
8. G. Algan and I. Ulusoy, *Knowl.-Based Syst.*, **215**, 106771 (2021).
9. H. Song, M. Kim, D. Park, Y. Shin, and J. G. Lee, *IEEE Transactions on Neural Networks and Learning Systems*, **0**, 1 (2022).
10. D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, “Using trusted data to train deep networks on labels corrupted by severe noise.” *NeurIPS* (2018).
11. S. Sukhbaatar and R. Fergus, “Learning from noisy labels with deep neural networks.” *ICLR* (2015).
12. S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, “Training deep neural networks on noisy labels with bootstrapping.” *ICLR* (2015).
13. J. Goldberger and E. Ben-Reuven, “Training deep neural-networks using a noise adaptation layer.” *ICLR* (2017).
14. Y. Yao, T. Liu, B. Han, M. Gong, J. Deng, G. Niu, and M. Sugiyama, “Dual t: Reducing estimation error for transition matrix in label-noise learning.” *NeurIPS* (2020).
15. N. Manwani and P. S. Sastry, *IEEE Transactions on Cybernetics*, **43**, 1146 (2013).
16. Z. Zhang and M. R. Sabuncu, “Generalized cross entropy loss for training deep neural networks with noisy labels.” *NeurIPS* (2018).
17. S. Kullback and R. A. Leibler, *Annals of Mathematical Statistics*, **22**, 79 (1951).
18. Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, “Symmetric cross entropy for robust learning with noisy labels.” *ICCV*, 322 (2019).
19. X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey, “Normalized loss functions for deep learning with noisy labels.” *ICML*(PMLR), 6543 (2020).
20. Y. Xu, P. Cao, Y. Kong, and Y. Wang, “Ldmi: A novel information-theoretic loss function for training deep nets robust to label noise.” *NeurIPS*, 6222 (2019).
21. K. H. Lee, X. He, L. Zhang, and L. Yang, “Cleannet: Transfer learning for scalable image classifier training with label noise.” *CVPR*, Piscataway, NJ (IEEE), 5447 (2018).
22. W. Zhang, Y. Wang, and Y. Qiao, “Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition.” *CVPR*, Piscataway, NJ (IEEE), 7373 (2019).
23. T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, “Learning from massive noisy labeled data for image classification.” *CVPR*, Piscataway, NJ (IEEE), 2691 (2015).
24. J. Han, P. Luo, and X. Wang, “Deep self-learning from noisy labels.” *CVPR*, Piscataway, NJ (IEEE), 5138 (2019).
25. B. Han, G. Niu, X. Yu, Q. Yao, M. Xu, I. Tsang, and M. Sugiyama, “Sigua: Forgetting may make learning with noisy labels more robust.” *ICML*(PMLR), 4006 (2020).
26. K. Yi and J. Wu, “Probabilistic end-to-end noise correction for learning with noisy labels.” *CVPR*, Piscataway, NJ (IEEE), 7017 (2019).
27. D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, “Joint optimization framework for learning with noisy labels.” *CVPR*, Piscataway, NJ (IEEE), 5552 (2018).
28. S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, “Early-learning regularization prevents memorization of noisy labels.” *NeurIPS* (2020).
29. E. Arazo, D. Ortego, P. Albert, N. O’Connor, and K. McGuinness, “Unsupervised label noise modeling and loss correction.” *ICML*(PMLR), 312 (2019).
30. J. Li, R. Socher, and S. C. Hoi, “Dividemix: Learning with noisy labels as semi-supervised learning.” *ICLR* (2020).
31. E. Malach and S. Shalev-Shwartz, “Decoupling when to update from how to update.” *NIPS* (2017).
32. M. F. Balcan, A. Blum, and K. Yang, “Co-training and expansion: Towards bridging theory and practice.” *NIPS* (2004).
33. D. Mandal, S. Bharadwaj, and S. Biswas, “A novel self-supervised re-labeling approach for training with noisy labels.” *WACV*, 1381 (2020).
34. H. Wei, L. Feng, X. Chen, and B. An, “Combating noisy labels by agreement: A joint training method with co-regularization.” *CVPR*, Piscataway, NJ(IEEE), 13726 (2020).
35. V. Sindhwani, P. Niyogi, and M. Belkin, “A co-regularization approach to semi-supervised learning with multiple views.” *ICML workshop*, **2005**, 74 (2005).
36. A. Kumar, A. Saha, and H. Daume, “Co-regularization based semi-supervised domain adaptation.” *NIPS* (2010).
37. L. Jiang, Z. Zhou, T. Leung, L. J. Li, and L. Fei-Fei, “Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels.” *ICML* (PMLR), 2304 (2018).
38. D. Berthelot, N. Carlini, I. J. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, “Mixmatch: A holistic approach to semi-supervised learning.” *NeurIPS* (2019).
39. P. Albert, D. Ortego, E. Arazo, N. E. O’Connor, and K. McGuinness, “Addressing out-of-distribution label noise in webly-labelled data.” *WACV*, 392 (2022).

40. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Proc. IEEE*, **86**, 2278 (1998).
41. A. Krizhevsky and G. E. Hinton, Technical Report University of Toronto (2009).
42. L. Jiang, D. Huang, M. Liu, and W. Yang, “Beyond synthetic noise: Deep learning on controlled noisy labels.” *ICML*(PMLR) (2020).
43. W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool, Webvision database: Visual learning and understanding from web data arXiv:1708.02862 (2017).
44. T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, “Learning from massive noisy labeled data for image classification.” *CVPR*, Piscataway, NJ(IEEE) (2015).
45. Z. Zhang, H. Zhang, S. O. Arik, H. Lee, and T. Pfister, “Distilling effective supervision from severe label noise.” *CVPR*, Piscataway, NJ(IEEE) (2020).
46. D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. P. Nguyen, L. Beggel, and T. Brox, “Self: Learning to filter noisy labels with self-ensembling.” *ICLR* (2020).
47. X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, “How does disagreement help generalization against label corruption?” *ICML*(PMLR), 7164 (2019).
48. J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, “Learning to learn from noisy labeled data.” *CVPR*, Piscataway, NJ(IEEE), 5051 (2019).
49. C. Tan, J. Xia, L. Wu, and S. Z. Li, “Co-learning: Learning from noisy labels with self-supervision.” *ACM MM*, New York1405 (2021).
50. Y. Yao, Z. Sun, C. Zhang, F. Shen, Q. Wu, J. Zhang, and Z. Tang, “Jo-src: A contrastive approach for combating noisy labels.” *CVPR*, Piscataway, NJ(IEEE), 5192 (2021).
51. Y. Kim, J. Yun, H. Shon, and J. Kim, “Joint negative and positive learning for noisy labels.” *CVPR*, Piscataway, NJ(IEEE), 9442 (2021).
52. X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. Erfani, S. Xia, S. Wijewickrema, and J. Bailey, “Dimensionality-driven learning with noisy labels.” *ICML*(PMLR), 3355 (2018).
53. P. Chen, B. B. Liao, G. Chen, and S. Zhang, “Understanding and utilizing deep neural networks trained with noisy labels.” *ICML*(PMLR), 1062 (2019).