THE UNIVERSITY OF SYDNEY

Doctoral Thesis

Learning with Noisy Labels Incorporating Fairness and Privacy Concerns

Author:
Songhua Wu

Supervisor:
Sen. Lec. Tongliang Liu

Co-Supervisor:
Hon. Lec. Ali Anaissi

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

TML Machine Learning Group
School of Computer Science, Faculty of Engineering

November 3, 2023

Abstract of thesis entitled

Learning with Noisy Labels Incorporating Fairness and Privacy Concerns

Submitted by

Songhua Wu

for the degree of Doctor of Philosophy

at The University of Sydney

in November, 2023

In the era of big data, the volume of data is growing at a tremendous rate as data has been generated from multifarious sources, but without reasonable supervision. Consequently, the generated data are primarily imperfect, such as inaccurate, biased, and even invading privacy. However, modern machine learning systems heavily rely on the quality of data. Noisy labels (inaccurate data) can be easily memorized by deep neural networks and thereby leads to overfitting and poor generalization problems; biased data guide models to give unfair predictions toward certain groups; privacy-invasion data are inherently harmful to the machine learning community.

Recent works address such issues in different subdomains. To deal with noisy labels, two principal methodologies have been developed: (1) learning the noise generating mechanism, i.e., a *transition matrix* $T(X)$ defining the mapping between clean label Y and noisy label $\tilde{Y}$ such that $P(\tilde{Y} \mid X) = T^{\top}(X)P(Y \mid X)$, where $P(\cdot \mid X)$ denotes the posterior vector, and then using it to build statistically consistent classifiers; (2) detecting confident samples $(X, \tilde{Y})$ with correct labels, i.e., $\tilde{Y} = Y$, and using them to train a clean classifier. Within this scope, we propose two solutions to this problem from diverse perspectives. The first one from a noise reduction perspective transforms data points with noisy *class labels* to data pairs with noisy *similarity labels*, which reduces the noise rate with a theoretical guarantee and thus makes the noise easier to handle. The second one from

a curriculum learning perspective designs a curriculum selecting training data based on their dynamics over the course of training to learn the clean classifier and the transition matrix simultaneously.

To mitigate the unfairness in machine learning algorithms, plenty of fairness notions and methods have been proposed. The methods focusing on statistical metrics discover the discrepancy of statistical metrics and give an equal probability of statistical metrics between individuals or subpopulations. The methods focusing on the causality-based fairness notions additionally employ causal graphs to take knowledge about the structure of real-world data into consideration and make causally fair predictions. To protect personal privacy, the indirect questioning method is commonly used to collect data when surveying sensitive topics such as sexual misconduct. Then the data with indirect supervision can be processed by similarity learning methods. However, noisy labels are ubiquitous, which makes some fairness-aware algorithms even more prejudiced than fairness-unaware ones, and similarity learning methods fail to learn optimal models.

To tackle these problems, we provide general frameworks for learning fair classifiers with noisy labels. For statistical fairness notions, we rewrite the classification risk and the fairness metric in terms of noisy data and thereby build robust classifiers. For the causality-based fairness notion, we exploit the internal causal structure of data to model the label noise and counterfactual fairness simultaneously; we propose a denoised and unbiased estimator for the classification risk with respect to the accurately labeled data by employing the noisy data with indirect supervision and then learn the optimal model under the empirical risk minimization framework.

Learning with Noisy Labels Incorporating Fairness and Privacy Concerns

by

Songhua Wu
B.E. *University of Science and Technology of China*

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy

at

University of Sydney

November, 2023

Declaration

I, Songhua Wu, declare that this thesis titled, "Learning with Noisy Labels Incorporating Fairness and Privacy Concerns", which is submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy, represents my own work and has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree, diploma, or other qualifications except where due acknowledgment has been made.

Chapter 3 of this thesis is published as [121]. I co-designed the study with the co-authors, proved the theorems, conducted the experiments, and wrote the drafts of the MS.

Chapter 5 of this thesis is published as [119]. I co-designed the study with the co-authors, proved the theorems, conducted the experiments, and wrote the drafts of the MS.

Chapter 6 of this thesis is published as [120]. I co-designed the study with the co-authors, proved the theorems, conducted the experiments, and wrote the drafts of the MS.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

For the joy of exploring the unknown and uncertainty.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Dr. Tongliang Liu for his invaluable advice, continuous support, and patience during my PhD study. His enormous knowledge and great experience have encouraged me in all the time of my academic research and daily life.

I would also like to sincerely thank Dr. Ali Anaissi and all members of the Trustworthy Machine Learning lab. It is their kind help and constructive discussion that made my PhD journey a wonderful time.

Lastly, thanks should also go to my family and my friends. Their belief in me has kept my spirits and motivation high during this journey.

Songhua Wu
University of Sydney
November 3, 2023

Contents

Chapter 1

Introduction

The volume of data is growing at a tremendous rate as data has been generated from multifarious sources but without proper supervision. As a result, these data are primarily imperfect, such as inaccurate, biased, and privacy-invasion. It is usually challenging to control the labeling quality of large-scale datasets because the labels were generated by complicated mechanisms such as non-expert workers [36]. An average of 3.3% noisy labels is identified in the test/validation sets of 10 of the most commonly-used datasets in computer vision, natural language, and audio analysis [84]. Besides, data, especially large-scale data, is often heterogeneous, generated by subgroups with their own characteristics and behaviors, and the heterogeneities can bias the data [74]. Moreover, many machine learning systems require private data. Hundreds of ethical issues regarding private data collection have been raised [101].

Notably, the quality of data is crucial to the success of modern machine learning systems. The training of neural networks easily fails in the presence of even the simple instance-independent noisy labels since they quickly lead to model overfitting of the noises [138], and thereby degenerate the generalization ability of the model. Besides, machine learning algorithms are very sensitive to biases that render their decisions unfair, i.e., having prejudice or favoritism toward an individual or a group based on their inherent or acquired characteristics [96], which is unacceptable as they are entrusted with important tasks, i.e., making high-stakes decisions in loan applications [80], dating and hiring [14, 21], and even parole [26]. Moreover, collecting and utilizing private data could be unethical and illegal [101].

Previous works tackle these problems in separate domains. However, since the label noise is ubiquitous in real-world datasets, this problem is entangled with fairness and privacy problems. Specifically, plenty of fairness metrics and methods have been proposed to mitigate the bias in machine learning algorithms, but label noise makes fairness-aware algorithms exhibit even more prejudice than fairness-unaware ones; the indirect questioning method is commonly employed in gathering data on sensitive topics, but label noise makes it struggling to learn optimal models from data collected with indirect supervision.

Therefore, in this thesis, we not only investigate the pure label noise problem focusing on generalization issues but also incorporate fairness and privacy concerns, which bring new insights and challenges as well.

To address the label noise problem, two principal methodologies have been developed: (1) learning the noise generating mechanism, i.e., a *transition matrix* T defining the mapping between clean label Y and noisy label $\tilde{Y}$, which is employed to construct statistically consistent classifiers [64, 88, 129]. (2) detecting samples $(X, \tilde{Y})$ with correct labels $\tilde{Y} = Y$ and using them to train a clean classifier [37, 133]; Despite the promising results achieved by both methodologies in some simplified scenarios, they encounter significant challenges when applied to practical and more complex settings, e.g., high noise rate, and instance-dependent label noise.

To address the fairness problem, numerous fairness notions and methods have been proposed. Statistical metric-focused methods aim to identify and rectify disparities in statistical metrics, ensuring equal probabilities across individuals or sub-populations [28, 38, 20]. On the other hand, fairness notions based on causality take into account the underlying causal relationships in real-world data by utilizing causal graphs, enabling the generation of causally fair predictions [50, 56, 141]. To preserve personal privacy, the indirect questioning method is commonly employed in gathering data on sensitive topics such as sexual misconduct that reduces the social desirability bias and increases data reliability [114, 29]. Then data collected with indirect supervision can be processed using similarity learning methods [5]. However, the presence of noisy labels severely damages the effectiveness of previous methods and brings new challenges as discussed above.

Targeting the aforementioned challenges, in this thesis we propose novel methods for the pure label noise problem in Chapters 3 and 4, the fairness-incorporated one in Chapter 5, and the privacy-incorporated one in Chapter 6, which are organized as follows:

Chapter 2. Preliminaries. In this chapter, we formulate the problem of learning with noisy labels (LNL) incorporating fairness and privacy concerns.

Chapter 3. LNL from a Noise Reduction Perspective. In this chapter, we propose a method from a noise reduction perspective on dealing with class label noise by transforming training data with noisy class labels into data pairs with noisy similarity labels. This approach effectively reduces the noise rate with a theoretical guarantee, thereby making the noise more manageable.

Chapter 4. LNL from a Curriculum Learning Perspective. In this chapter, we propose a novel time-consistency metric, i.e., *TCP* for the instance-dependent label noise problem. Based on TCP, we can detect examples with clean labels or correct pseudo labels better than the existing measures, and allocate reliable triplets for learning the transition matrix. Then we design an assumption-free curriculum that learns the clean classifier, as well as the transition matrix simultaneously.

Chapter 5. LNL incorporating Fairness Concerns. In this chapter, we provide general frameworks for learning fair classifiers with noisy labels. For statistical fairness notions, we rewrite the classification risk and the fairness metric in terms of noisy data and thereby build robust classifiers. For the causality-based fairness notion, We exploit the internal causal structure of data to effectively model both the label noise and counterfactual fairness.

Chapter 6. LNL incorporating Privacy Concerns. In this chapter, we propose a novel weakly supervised learning setting, which considers the case where similar data pairs collected from the indirect questioning survey method are corrupted with the mutually contaminated distributions model, and a robust risk-consistent estimator to solve this problem.

Chapter 7. Conclusion. In this chapter, we conclude this thesis.

Chapter 2

Preliminaries

In this chapter, we introduce the problem of learning with noisy labels incorporating fairness and privacy concerns.

General Settings. Let $(X, Y) \in \mathcal{X} \times \{1, \ldots, c\}$ be the random variables for instances and clean labels, where X denotes the variable of instances, Y the variable of labels, $\mathcal{X}$ the instance space and c the number of classes.

Label Noise. In many real-world applications, the observed labels are not always correct but contain some noise. Let $\tilde{Y}$ be the random variable for the noisy label. The label noise structure is usually formulated by a $c \times c$ transition matrix, where c is the number of classes. The ij-th element of a transition matrix is $T_{ij}(x) = P(\tilde{Y} = j \mid Y = i, X = x)$, which represents the probability that the instance x with the clean label $Y = i$ actually has a noisy label $\tilde{Y} = j$. It can establish the connection between noisy posterior and clean posterior, i.e., $P(\tilde{Y} \mid X) = T^{\top} P(Y \mid X)$. Utilizing a transition matrix, consistent algorithms can be built [81, 97, 64, 88, 61]. Currently, there are three typical models for handling label noise, which are the random classification noise (RCN) model [12, 81, 73], the class-dependent label noise (CDN) model [88, 125, 143], and the instance-dependent label noise (IDN) model [10, 18]. Specifically, RCN assumes that clean labels flip randomly with a constant rate; CDN assumes that the flip rate only depends on the true class; IDN considers the most general case of label noise, where the flip rate depends on its instance.

Fairness. Generally, fairness is the absence of any prejudice or favoritism towards an individual or a group based on their intrinsic or acquired traits in the context of decision-making [96]. Before computer science delved into exploring fairness, philosophy and psychology had already attempted to define this concept. However, the lack of a universal definition of fairness highlights the complexity of solving this problem [74]. Different cultural backgrounds and perspectives lead to varying definitions of fairness. Principally, these definitions involve a protected attribute, e.g., race and gender, and a metric that is supposed to be equal with respect to the protected attribute. For example, The definition of equalized odds [38], states that "A predictor Y satisfies equalized odds with respect to protected attribute A and outcome Y, if $\hat{Y}$ and A are independent conditional on Y. $P(\hat{Y} = 1|A = 0, Y = y) = P(\hat{Y} = 1|A = 1, Y = y),\ y \in 0, 1$".

Privacy. Data reliability is a common concern especially when asking about sensitive topics such as sexual misconduct, or drug and alcohol abuse. Sensitive topics might cause refusals in surveys due to privacy concerns of the subjects [86]. This nonresponse reduces sample size and study power and increases bias. Various indirect questioning methods have been developed to reduce social desirability bias and increase data reliability. Questions in the form of 'With whom do you share the same opinion on issue I?' is one type of randomized response technique, which is a commonly used indirect questioning survey method [114, 29]. In this manner, similar data pairs (x, x') are collected [5]. A pair of instances are said to be similar if they are from the same class.

Given the presence of noisy labels, the fairness metric becomes imprecise. Due to the sensitivity of the questions, respondents might answer them in a manner that will be viewed favorably by others instead of answering truthfully [86], which makes the sampled examples contain dissimilar data pairs. Our aim is to learn robust classifiers that could assign clean labels to test data by exploiting the sample with noisy labels in compliance with fairness and privacy requirements.

Chapter 3

LNL from a Noise Reduction Perspective

Learning with noisy labels has attracted a lot of attention in recent years, where the mainstream approaches are in *pointwise* manners. Meanwhile, *pairwise* manners have shown great potential in supervised metric learning and unsupervised contrastive learning. Thus, a natural question is raised: does learning in a pairwise manner *mitigate* label noise? To give an affirmative answer, in this chapter, we propose a framework called *Class2Simi*: it transforms data points with noisy *class labels* to data pairs with noisy *similarity labels*, where a similarity label denotes whether a pair shares the class label or not. Through this transformation, the *reduction of the noise rate* is theoretically guaranteed, and hence it is in principle easier to handle noisy similarity labels. Amazingly, DNNs that predict the *clean* class labels can be trained from noisy data pairs if they are first pretrained from noisy data points. Class2Simi is *computationally efficient* because not only this transformation is on-the-fly in mini-batches, but also it just changes loss computation on top of model prediction into a pairwise manner. Its effectiveness is verified by extensive experiments.

3.1 Motivations and Contributions

Intuitively, the pairwise manners require less pointwise supervision information, i.e., class labels, and might be robust to label noise, which motivates us to introduce a pairwise manner to deal with noisy labels. Specifically, we propose *Class2Simi*, i.e., transforming training data with noisy class labels

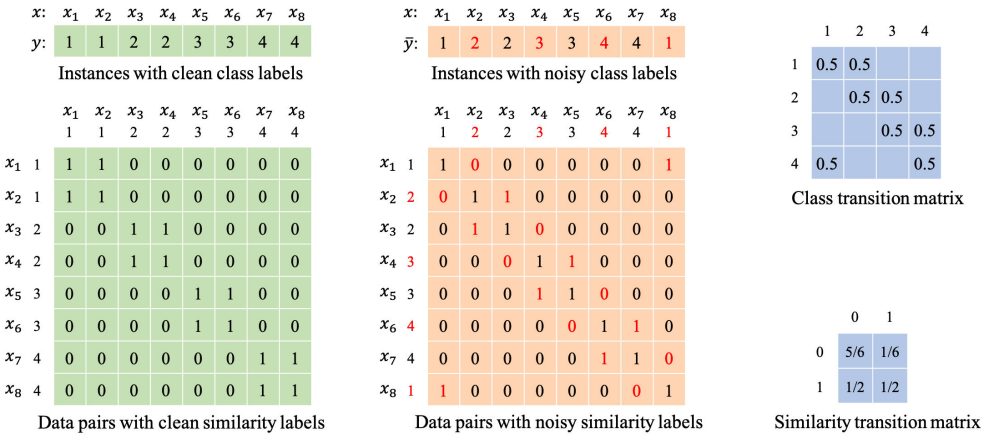


Figure 3.1: An illustration of the transformation from class labels to similarity labels. Note that $\tilde{y}$ stands for the noisy class label and y for the latent clean class label. The labels marked in red are incorrect. If we assume the class label noise is generated according to the transition matrix presented in the upper part of the right column, it can be calculated that the noise rate for the noisy class labels is 0.5 while the noise rate for the noisy similarity labels is 0.25. Note that the transition matrix for similarity labels can be calculated by exploiting the class transition matrix as in Theorem 3.3.1.

into data pairs with noisy similarity labels. A class label shows the class that an instance belongs to, while a similarity label indicates whether or not two instances belong to the same class. We theoretically prove that through this transformation, the noise rate becomes lower (see Theorem 3.3.2). This is because, given a data pair, of which if one point has an incorrect class label or even if both points have incorrect class labels, the similarity label could be correct. Moreover, this transformation also reduces a multi-class classification problem into a binary classification problem. In label noise learning, the binary problem is easier to handle and a lower noise rate usually results in higher classification performance [88].

We illustrate the transformation and the robustness of similarity labels in Figure 3.1. In the middle column, we can see the noisy similarity labels of example-pairs (x_2, x_5) and (x_2, x_4) are correct, although there is one mislabeled point in (x_2, x_5), and two mislabeled points in (x_2, x_4). Moreover, if we assume that the noisy class labels in Figure 3.1 are generated according to the latent clean class labels and the class transition matrix (the ij-th

entry of this matrix denotes the probability that the clean class label i flips into the noisy class label j), the noise rate of class labels is 0.5. Meanwhile, the corresponding similarity transition matrix can be derived from the class transition matrix with the class-priors (see Theorem 3.3.1). The noise rate of similarity labels is 0.25, which is the proportion of the number of incorrect similarity labels to the number of total similarity labels.

To handle the transformed data pairs with noisy similarity labels, the connection between noisy similarity posterior and clean class posterior should be established. Intuitively, noisy similarity posterior can be linked to clean similarity posterior, and then clean class posterior can be inferred from clean similarity posterior. For the first part, we can draw on the philosophy of dealing with noisy class labels, e.g., selecting reliable data pairs for training, and correcting the similarity loss to learn a robust similarity classifier. For the second part, plenty of similarity metrics can be adopted. As an example, we could adapt the *Forward* [88] to learn clean similarity posterior from data with noisy similarity labels. Then, by using the inner product of the clean class posterior [44] to approximate clean similarity posterior, the clean class posterior (and thus the robust classifier) can thereby be learned. It is obvious that Class2Simi suffers information loss because we can not recover the class labels from similarity labels, which implies that learning only from similarity labels can only cluster data points but can not identify the semantic classes of clusters. In [44], a pointwise cluster can be learned from similarity labels.

However, in our case, the pairs with similarity labels are constructed from points with class labels, and we could acquire the semantic class information of clusters by pretraining the model from points with class labels without any additional information. Note that when class labels of points are corrupted, leading to noisy similarity labels, the proposed pretraining still works because the noisy class is assumed to be dominated by its clean class in label noise learning. Thus we do not suffer the major information loss in noisy similarity learning.

It is worthwhile to mention Class2Simi increases the computation cost very slightly, compared with the standard pointwise training. It will be shown in Figure 3.2 that most computation is still pointwise. Only the

computation of the pairwise enumeration layer [43] and the loss are pairwise, while both the forward and backward propagation are pointwise. The pairwise enumeration layer was verified to only introduce a negligible overhead to the training time [44]. Moreover, the transformation is on-the-fly in mini-batches, which means the pairs are quadratic on the batch size other than the whole sample size.

3.2 Related Work

Existing methods for learning with noisy labels can be divided into two categories: algorithms that result in statistically inconsistent or consistent classifiers. Methods in the first category usually employ heuristics to reduce the side-effect of noisy labels, e.g., selecting reliable samples [37, 133, 115, 118, 122], reweighting samples [95, 48, 68, 53, 94], correcting labels [106, 144], designing robust loss functions [143, 128, 65, 67], employing side information [109, 62], and (implicitly) adding regularization [61, 62, 111, 109, 35, 140, 33, 45, 142, 34]. Those methods empirically work well in many settings. Methods in the second category aim to learn robust classifiers that could converge to the optimal ones defined by using clean data. They utilize the transition matrix, which denotes the probabilities that the clean labels flip into noisy labels, to build consistent algorithms [81, 97, 64, 88, 83, 135, 53, 42, 65, 132, 123]. The idea is that given the noisy class posterior probability and the transition matrix, the clean class posterior probability can be inferred.

Note that the noisy class posterior and the transition matrix can be estimated by exploiting the noisy data, where the transition matrix additionally needs anchor points [64, 88]. Some methods assume anchor points have already been given [135]. There are also methods showing how to identify anchor points from the noisy training data [64].

3.3 Methodology

3.3.1 Transformation on Labels and Transition Matrix

As in Figure 3.1, we combine every 2 instances in pairs, and if the two instances have the same class label, we assign this pair a similarity label 1, otherwise 0. If the class labels are corrupted, the generated similarity labels also contain noise.

The definition of the similarity transition matrix is similar to the class transition matrix. The elements in a similarity transition matrix denote probabilities that clean similarity labels H flip into noisy similarity labels $\tilde{H}$, i.e., $T_{s,mn} \coloneqq P(\tilde{H} = n|H = m)$. The dimension of the similarity transition matrix is always 2×2. Since the similarity labels are generated from class labels, the similarity noise is determined and, thus can be calculated, by the class transition matrix.

Theorem 3.3.1. *Assume that the dataset is balanced (each class has the same amount of instances, and c classes in total), and the noise is class-dependent. Given a class transition matrix T_c, such that $T_{c,ij} = P(\tilde{Y} = j|Y = i)$. The elements of the corresponding similarity transition matrix T_s can be calculated as*

$$
\begin{aligned}
T_{s,00} &= \frac{c^2 - c - \left(\sum_j (\sum_i T_{c,ij})^2 - ||T_c||_{\text{Fro}}^2\right)}{c^2 - c}, \\
T_{s,01} &= \frac{\sum_j (\sum_i T_{c,ij})^2 - ||T_c||_{\text{Fro}}^2}{c^2 - c}, \\
T_{s,10} &= \frac{c - ||T_c||_{\text{Fro}}^2}{c}, \qquad T_{s,11} = \frac{||T_c||_{\text{Fro}}^2}{c}.
\end{aligned}
$$

A detailed proof is provided in Appendix A.1.

Remark 3.3.1. *Theorem 3.3.1 can easily extend to the setting where the dataset is unbalanced in classes by multiplying each $T_{c,ij}$ by a coefficient n_i. n_i is the number of instances from the i-th class.*

Note that the similarity labels are only dependent on class labels. If the class noise is class-dependent, the similarity noise is also 'class-dependent' (class means similar and dissimilar). Under class-dependent label noise,

a binary classification is learnable as long as $T_{00} + T_{11} > 1$ [76], where T is the corresponding binary transition matrix; a multi-class classification is learnable if the corresponding transition matrix T_c is invertible. For Class2Simi, in the most general sense, i.e., T_c is invertible, $T_{s,00} + T_{s,11} > 1$ holds. Namely, the learnability of the pointwise classification implies the learnability of the reduced pairwise classification. A proof is provided in Appendix A.2. However, the latter cannot imply the former: As shown in Figure 3.1, the class transition matrix is not invertible, and thus the pointwise classification is not learnable while the reduced pairwise classification is learnable. Note that this 'learnable' is only for the binary pairwise classification in this case. Technically, two conditions must be met to learn a pointwise classifier from pairwise data: (1) The reduced pairwise classification is learnable; (2) The semantic class information is learnable. Generally, the second condition is equivalent to the learnability of the pointwise classification. Thus the learnability for a pointwise classifier of the two learning manners is consistent.

Theorem 3.3.2. *Assume that the dataset is balanced (each class has the same amount of samples), and the noise is class-dependent. When the number of classes $c \geq 8$, the noise rate of noisy similarity labels is lower than that of the noisy class labels.*

A detailed proof is provided in Appendix A.3.

In multi-class classification problems, the number of classes is usually larger than 8. As c becomes larger, the range of 'dissimilarity' of data pairs becomes larger, which is conducive to the reduction of the noise rate. Through Class2Simi, the number of *d-pairs* (with similarity label 0) is $(c-1)$ times as much as that of *s-pairs* (with similarity label 1). Meanwhile, compared with the original noise rate of noisy class labels, the noise rate of noisy similarity labels of s-pairs is higher and that of d-pairs is lower, while the overall noise rate of data pairs is lower, which partially reflects that the impact of label noise is less bad. Notably, the flip from 'dissimilar' to 'similar' should be more adversarial and thus more important. In practice, it is common that one class has more than one clusters, while it is rare that two or more classes are in the same cluster. If there is a flip from 'similar' to 'dissimilar' and based on it we split a (latent) cluster into two (latent)

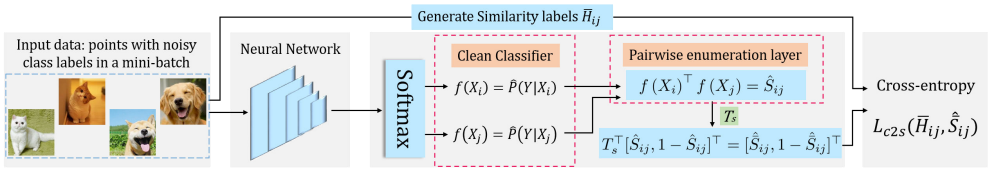


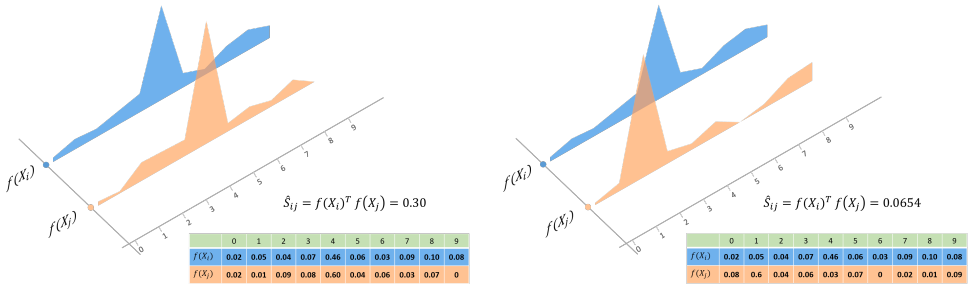
Figure 3.2: An overview of the proposed method. We add a pairwise enumeration layer and similarity transition matrix to calculate and correct the predicted similarity posterior. By minimizing the proposed loss L_{c2s}, a classifier f can be learned for assigning clean labels. The detailed structures of the Neural Network are provided in Section 4.

clusters, we still have a high chance to label these two clusters correctly later. If there is a flip from 'dissimilar' to 'similar' and based on it we join two clusters belonging to two classes into a single cluster, we nearly have zero chance to label this cluster correctly later. As a consequence, the flip from 'dissimilar' to 'similar' is more adversarial and important, thus deserving a larger weight when calculating the noise rate. Here we assign all data pairs the same weight, otherwise, there would be a more reduction in the noise rate. On balance, considering the reduction of the overall noise rate is meaningful.

When dealing with label noise, a low noise rate has many benefits. The most important one is that the noise-robust algorithms will consistently achieve higher performance when the noise rate is lower [37, 125, 88]. Another benefit is that, when the noise rate is low, the complex instance-dependent label noise can be well approximated by class-dependent label noise [19], which is easier to handle.

3.3.2 Learning with Noisy Similarity Labels

In order to learn a multi-class classifier from similarity labeled data, we should establish relationships between class posterior probability and similarity posterior probability. Here we employ the relationship established in [44], which is derived from a likelihood model. As in Figure 3.2, we denote the predicted clean similarity posterior by the inner product between two categorical distributions: $\hat{S}_{ij} = f(X_i)^\top f(X_j)$. Intuitively, $f(X)$ outputs the predicted categorical distribution of input data X and $f(X_i)^\top f(X_j)$ can measure how similar the two distributions are. For clarity, we visualize



(a) Similar example **(b)** Dissimilar example

Figure 3.3: Examples of predicted noisy similarity. Assume the class number is 10; $f(X_i)$ and $f(X_j)$ are categorical distributions of X_i and X_j respectively, which are shown above in the form of area charts. $\hat{S}_{ij}$ is the predicted similarity posterior between two instances, calculated by the inner product between two categorical distributions.

the predicted similarity posterior in Figure 3.3. If X_i and X_j are predicted belonging to the same class, i.e., $\arg\max_{m \in c} f_m(X_i) = \arg\max_{n \in c} f_n(X_j)$, the predicted similarity posterior should be relatively high ($\hat{S}_{ij} = 0.30$ in Figure 3.3(a)). By contrast, if X_i and X_j are predicted belonging to different classes, the predicted similarity posterior should be relatively low ($\hat{S}_{ij} = 0.0654$ in Figure 3.3(b)). Note that the noisy similarity posterior $P(\tilde{H}_{ij}|X_i, X_j)$ and clean similarity posterior $P(H_{ij}|X_i, X_j)$ satisfy

$$P(\tilde{H}_{ij}|X_i, X_j) = T_s^\top P(H_{ij}|X_i, X_j). \tag{3.1}$$

Therefore, we can infer the predicted noisy similarity posterior $\hat{\tilde{S}}_{ij}$ from the predicted clean similarity posterior $\hat{S}_{ij}$ with the similarity transition matrix. To measure the error between the predicted noisy similarity posterior $\hat{\tilde{S}}_{ij}$ and noisy similarity label $\tilde{H}_{ij}$, we employ a binary cross-entropy loss function. The final optimization function is

$$L_{c2s}(\tilde{H}_{ij}, \hat{\tilde{S}}_{ij}) = -\sum_{i,j} \tilde{H}_{ij} \log \hat{\tilde{S}}_{ij} + (1 - \tilde{H}_{ij}) \log(1 - \hat{\tilde{S}}_{ij}).$$

The pipeline of the proposed Class2Simi is summarized in Figure 3.2. The softmax function outputs an estimation for the clean class posterior,

Algorithm 1 Class2Simi

Input: training data with noisy class labels; validation data with noisy class labels.

Stage 1: Learn $\hat{T}_s$

1: Learn $g(X) = \hat{P}(\tilde{Y}|X)$ by training data with noisy class labels, and save the model for Stage 2;

2: Estimate $\hat{T}_c$ following the optimization method in [88];

3: Transform $\hat{T}_c$ to $\hat{T}_s$.

Stage 2: Learn the classifier $f(X) = \hat{P}(Y|X)$

4: Load the model saved in Stage 1, and train the whole pipeline shown in Figure 3.2.

Output: classifier f.

i.e., $f(X) = \hat{P}(Y|X)$, where $\hat{P}(Y|X)$ denotes the estimated class posterior. Then a pairwise enumeration layer is added to calculate the predicted clean similarity posterior $\hat{S}_{ij}$ of every two instances. According to Eq. (3.1), by pre-multiplying the transpose of the noise similarity transition matrix, we can obtain the predicted noisy similarity posterior $\hat{\tilde{S}}_{ij}$. Therefore, by minimizing L_{c2s}, we can learn a classifier for predicting noisy similarity labels. Meanwhile, before the transition matrix layer, the pairwise enumeration layer will output a prediction for the clean similarity posterior, which guides $f(X)$ to predict clean class labels.

Remark 3.3.2. *For a better understanding, we formulate Class2Simi in the form combined with Forward as an illustration. However, Class2Simi is a meta method that can be applied on top of sample selection, loss correction, label correction, and many other label noise learning methods. We provide another implementation with Reweight in Appendix A.4.*

3.3.3 Implementation

The proposed algorithm is summarized in Algorithm 1. Since learning only from similarity labels will lose the semantic class information, we load the model trained on the data with noisy class labels to provide the semantic class information for similarity learning in Stage 2.

3.3.4 Generalization Error Bound

We formulate the above problem in the traditional risk minimization framework [79]. The expected and empirical risks of employing estimator f can be defined as

$$R(f) = E_{(X_i,X_j,\tilde{Y}_i,\tilde{Y}_j,\tilde{H}_{ij},T_s)\sim\mathcal{D}_\rho}[\ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij})],$$

and

$$R_n(f) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij}),$$

where n is the training sample size of the noisy data. Assume that the neural network has d layers with parameter matrices $W_1, \ldots, W_d$, and the activation functions $\sigma_1, \ldots, \sigma_{d-1}$ are Lipschitz continuous, satisfying $\sigma_j(0) = 0$. We denote by $H : X \mapsto W_d\sigma_{d-1}(W_{d-1}\sigma_{d-2}(\ldots\sigma_1(W_1X))) \in \mathbb{R}$ the standard form of the neural network. $h = \arg\max_{i\in\{1,\ldots,c\}} h_i$. Then the output of the softmax function is defined as $f_i(X) = \exp(h_i(X))/\sum_{j=1}^{c}\exp(h_j(X)), i = 1, \ldots, c$. We can then obtain the following generalization error bound.

Theorem 3.3.3. *Assume the parameter matrices $W_1, \ldots, W_d$ have Frobenius norm at most $M_1, \ldots, M_d$, and the activation functions are 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Assume the transition matrix is given, and the instances X are upper bounded by B, i.e., $\|X\| \leq B$ for all X, and the loss function ℓ is upper bounded by M. Then, for any $\delta > 0$, with probability at least $1 - \delta$,*

$$R(\hat{f}) - R_n(\hat{f}) \leq M\sqrt{\frac{\log 1/\delta}{2n}} + \frac{(T_{s,11} - T_{s,01})2Bc(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d}M_i}{T_{s,11}\sqrt{n}}. \tag{3.2}$$

A detailed proof is provided in Appendix A.5.

Theorem 3.3.3 implies that if the training error is small and the training sample size is large, the expected risk $R(\hat{f})$ of the representations for noisy similarity posterior will be small. If the transition matrix is well estimated, the clean similarity posterior as well as the classifier for the clean

class will also have a small risk according to Eq. (3.1) and the Class2Simi relations. This theoretically justifies why the proposed method works well. In the experiment section, we will show that the transition matrices are well estimated and that the proposed method significantly outperforms the baselines.

In Class2Simi, a multi-class classification is reduced to a pairwise binary classification. For data pairs, if a surrogate loss is classification-calibrated, minimizing it leads to minimizing the zero-one loss on the pointwise random variables in the limit case. Otherwise, we cannot guarantee the worst-case learnability of learning pointwise labels from pairwise labels, but it cannot imply the average-case non-learnability either. Theoretically, [6] proved that when the pairwise labels are all correct, for the special case $c = 2$, a good model for predicting s-/d-pairs must also be a good model for predicting the original classes, under mild assumptions. In practice, it seems fine to use non-classification-calibrated losses. According to [107], the multi-class margin loss (i.e., one-vs-rest loss) and the pairwise comparison loss (i.e., one-vs-one loss) are proved to be non-calibrated, but they are still the main multi-class losses in [79, 99].

3.4 Experiments

Experiment setup. We employ three widely used image datasets, i.e., *MNIST* [57], *CIFAR-10*, and *CIFAR-100* [54], one text dataset *News20*, and one real-world noisy dataset *Clothing1M* [127]. *News20* is a collection of approximately 20,000 newsgroup documents, partitioned nearly evenly across 20 different newsgroups. *Clothing1M* has 1M images with real-world noisy labels and additional 50k, 14k, 10k images with clean labels for training, validation and test, and we only use noisy training set in the training phase. Note that the similarity learning method of Class2Simi is based on *clustering* because there is no class information. Intuitively, for a noisy class, if most instances in it belong to another specific class, we can hardly identify it. For example, assume that a class with noisy labels $\tilde{i}$ contains n_i instances with ground-truth labels i and n_j instances with ground-truth

labels j. If n_j is bigger than n_i, the model will cluster class i into j. Unfortunately, in *Clothing1M*, most instances with label '5' belong to class '3' actually. Therefore, we merge the two classes and denote the modified dataset by *Clothing1M** which contains 13 classes. For all the datasets, we leave out 10% of the training data as a validation set, which is for model selection.

For *MNIST*, *CIFAR-10*, and *CIFAR-100*, we use LeNet [58], ResNet-26 with shake-shake regularization [30], and ResNet-56 with pre-activation [41], respectively. For *News20*, we first use GloVe [91] to obtain vector representations for the raw text data, and employ a 3-layer MLP with the Softsign active function. For *Clothing1M**, we use pre-trained ResNet-50 [40]. Further details for the experiments are provided in Appendix A.6.1.

Noisy labels generation. For clean datasets, we artificially corrupt the class labels of training and validation sets according to the class transition matrix. Specifically, for each instance with clean label i, we replace its label by j with a probability of $T_{c,ij}$. In this chapter, we consider both symmetric and asymmetric noise settings which are defined in Appendix A.6.2. *Sym-0.2* means symmetric noise type with a 0.2 noise rate and *Asym-0.2* means asymmetric noise type with a 0.2 noise rate.

Baselines. In this chapter, we compare our method with the following baselines: *Reweight* [64], *Forward* [88], and *Revision* [125], which utilize a class-dependent transition matrix to model the noise, and learn a robust classifier. Besides, we externally conduct experiments on *Co-teaching* [37], which is a representative algorithm of selecting reliable samples for training; *JoCoR* [115], which employs a joint loss function to select small-loss samples; *PHuber-CE* [77], which introduces gradient clipping to mitigate the effects of noise; *APL* [67], which applies simple normalization on loss functions and makes them robust to noisy labels; *S2E* [130], which properly controls the sample selection process so that deep networks can benefit from the memorization effect. Besides, we conduct experiments on another implementation of the proposed method, which employs *Reweight* (More details are provided in Appendix A.4). To distinguish these two methods, we call them 'F-Class2Simi' and 'R-Class2Simi'.

Table 3.1: Means and Standard Deviations of Classification Accuracy over 5 trials on image datasets.

MNIST	Sym-0.2	Sym-0.4	Sym-0.6	Asym-0.2	Asym-0.4	Asym-0.6
Co-teaching	97.34±0.26	94.68±0.52	93.36±0.47	97.37±0.20	96.63±0.41	91.33±0.38
JoCor	97.48±0.12	96.31±0.20	93.18±0.27	97.31±0.09	95.73±0.29	91.43±0.28
PHuber-CE	98.65±0.18	98.17±0.15	97.63±0.36	98.73±0.09	98.36±0.25	97.37±0.41
APL	98.77±0.21	97.06±0.37	97.67±0.35	98.72±0.10	98.45±0.29	97.58±0.25
S2E	98.96±0.27	93.27±2.18	89.37±0.70	99.19±0.05	94.47±1.08	92.36±2.40
Revision	98.92±0.09	98.42±0.50	98.10±0.37	98.97±0.06	98.58±0.19	98.21±0.19
Reweight	98.78±0.16	98.26±0.22	97.02±0.58	98.62±0.19	98.12±0.31	96.98±0.29
Forward	98.76±0.03	98.37±0.25	96.89±0.49	98.61±0.22	98.08±0.33	97.43±0.25
R-Class2Simi	99.04±0.06	98.87±0.06	98.40±0.17	99.06±0.05	98.75±0.08	98.23±0.20
F-Class2Simi	**99.26±0.07**	**99.18±0.06**	**98.91±0.09**	**99.26±0.05**	**99.08±0.07**	**98.91±0.07**
CIFAR10	Sym-0.2	Sym-0.4	Sym-0.6	Asym-0.2	Asym-0.4	Asym-0.6
Co-teaching	88.92±0.45	85.97±1.02	75.97±1.33	89.14±0.36	84.77±1.08	76.07±1.27
JoCor	88.46±0.25	85.19±0.75	77.03±0.92	88.96±0.70	85.19±0.58	75.76±1.31
PHuber-CE	90.37±0.26	86.05±0.37	74.06±0.92	90.73±0.22	86.06±0.53	73.25±1.04
APL	89.07±0.92	85.77±0.84	70.06±1.06	89.97±0.19	85.60±0.91	72.33±1.68
S2E	90.04±1.22	82.05±1.95	57.96±4.70	90.12±0.97	83.16±1.58	64.77±3.06
Revision	90.02±0.48	85.47±0.71	73.92±2.02	89.77±0.28	85.32±1.36	75.24±1.87
Reweight	89.05±0.32	84.60±0.45	74.87±1.18	89.28±0.26	84.61±0.62	72.77±1.91
Forward	89.63±0.20	87.08±0.31	73.24±1.33	90.03±0.41	86.64±0.71	77.41±0.43
R-Class2Simi	90.91±0.26	87.80±0.23	79.19±1.65	91.07±0.21	87.78±0.33	78.56±0.63
F-Class2Simi	**91.38±0.19**	**88.22±0.19**	**79.45±0.53**	**91.24±0.27**	**87.79±0.36**	**79.05±0.56**
CIFAR100	Sym-0.2	Sym-0.4	Sym-0.6	Asym-0.2	Asym-0.4	Asym-0.6
Co-teaching	57.14±0.49	52.62±1.03	37.32±1.67	57.82±0.37	51.32±0.83	35.32±1.68
JoCoR	58.32±0.71	51.76±1.07	37.02±1.33	58.61±0.30	49.18±1.05	37.09±1.82
PHuber-CE	57.90±0.31	52.36±0.77	37.93±0.86	57.33±0.71	51.29±0.96	36.03±1.34
APL	54.03±0.92	49.06±0.93	36.06±2.02	55.62±0.92	48.37±0.94	35.02±1.72
S2E	59.37±1.09	43.29±1.94	30.08±3.91	58.92±1.21	42.88±2.16	29.93±4.05
Revision	59.62±0.97	53.26±0.84	35.82±2.06	58.77±0.93	52.72±1.38	37.72±1.75
Reweight	49.59±0.74	39.72±0.57	22.79±1.35	48.87±0.96	36.65±0.90	17.24±1.97
Forward	48.68±0.57	39.78±1.23	27.01±0.89	47.90±0.23	37.89±0.57	21.71±1.53
R-Class2Simi	55.45±0.55	50.38±0.49	35.57±0.75	54.95±0.65	47.56±0.72	34.82±0.58
F-Class2Simi	**60.26±0.18**	**54.85±0.60**	**40.38±0.58**	**59.10±0.13**	**52.99±0.78**	**38.69±2.84**

Results on noisy image datasets. The results in Table 3.1 and Figure 3.4 demonstrate that Class2Simi achieves distinguished classification accuracy and is robust against the estimation errors on the transition matrix.

Table 3.2: Means and Standard Deviations of Classification Accuracy over 5 trials on text datasets.

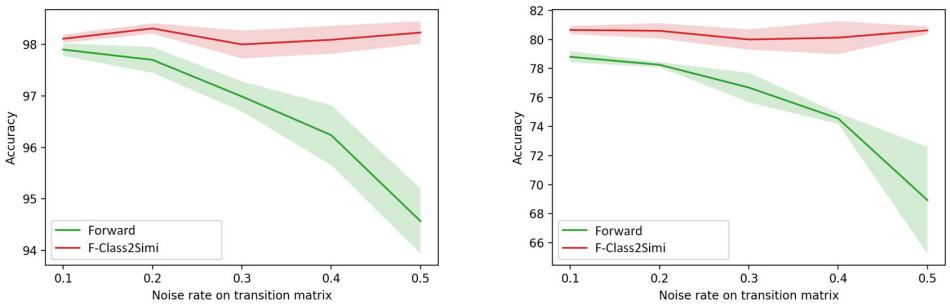
NEWS20	Sym-0.2	Sym-0.4	Sym-0.6	Asym-0.2	Asym-0.4	Asym-0.6
Co-teaching	55.32±0.28	51.09±1.06	47.07±0.83	55.29±0.41	53.08±0.26	45.63±0.75
JoCor	52.21±0.70	49.84±0.92	48.83±0.43	55.58±0.27	49.35±0.62	46.21±0.73
PHuber-CE	55.73±0.38	54.33±0.92	45.05±0.49	56.76±0.26	51.15±0.65	41.59±1.05
APL	56.91±0.21	53.12±1.21	43.60±1.28	56.11±0.23	50.93±1.05	43.60±1.28
S2E	57.93±0.37	47.16±1.32	28.53±5.04	54.89±1.92	50.42±1.71	30.67±3.12
Revision	58.06±0.19	52.30±1.73	46.84±1.09	56.41±0.77	53.44±0.83	43.77±1.08
Reweight	53.34±1.08	50.15±1.33	44.73±0.79	53.37±0.66	49.82±0.44	39.46±1.27
Forward	57.30±0.32	53.94±0.42	46.91±1.48	53.58±0.54	49.90±1.44	42.55±3.81
R-Class2Simi	**58.67±0.38**	56.59±0.74	50.48±0.97	58.44±0.66	**55.03±1.55**	**47.75±2.17**
F-Class2Simi	58.27±0.47	**56.70±1.13**	**50.18±0.89**	**58.46±0.68**	54.92±1.66	46.07±3.54

Table 3.3: Classification Accuracy on *Clothing1M**.

Co-teaching	74.70	JoCoR	74.98
PHuber-CE	73.16	APL	58.93
S2E	72.30	Revision	74.65
Forward	73.88	F-Class2Simi	75.41
Reweight	74.44	R-Class2Simi	**75.76**

From Table 3.1, overall, we can see that after the transformation, better performance are achieved due to a lower noise rate and the similarity transition matrix being robust to noise. Even for challenging noise rates of 0.6, Class2Simi achieves good accuracy, uplifting about 5 and 10 points on *CIFAR-10* and *CIFAR-100* respectively, compared with the corresponding pointwise methods.

In Figure 3.4, we show that the similarity transition matrix is robust against estimation errors. To verify this, we add some random noise to the ground-truth T_c through multiplying every element in class T_c by a random variable α_{ij}. We control the noise rate on the T_c by sampling α_{ij} in different intervals, i.e., 0.1 noise means that α_{ij} is uniformly sampled from $\pm[1.1, 1.2]$. Then we normalize T_c to make its row sums equal to 1. From Figure 3.4, we can see that the accuracy of Forward drops dramatically with the increase of the noise on T_c. By contrast, there is only a slight fluctuation of F-Class2Simi, indicating Class2Simi is robust against the



(a) *MNIST* (b) *CIFAR10*

Figure 3.4: Means and Standard Deviations of Classification Accuracy over 5 trials on *MNIST* and *CIFAR10* with perturbational ground-truth $\hat{T}_c$.

Table 3.4: Classification Accuracy on clean datasets. CE uses class labels and the cross-entropy loss function. C2S refers to Class2Simi.

Dataset	***MNIST***	***CIFAR10***	***CIFAR100***	***News20***
CE	**99.30±0.02**	94.03±0.14	58.74±0.51	**59.86±0.39**
C2S	99.24±0.05	**94.05±0.27**	**60.36±0.89**	59.74±0.20

estimation errors on the transition matrix.

Results on the noisy text dataset. Results in Table 3.2 show that the proposed method works well on the text dataset under both symmetric and asymmetric noise settings.

Results on the real-world noisy dataset. Results in Table 3.3 show that the proposed method also performs well against agnostic noise.

Ablation study. To investigate how the similarity loss function influences the classification accuracy, we conduct experiments with the cross-entropy loss function and the similarity loss function on clean datasets over 3 trials, where the T_c is set to an identity matrix. All other settings are kept the same. As shown in Table 3.4, on *MNIST*, *CIFAR10*, and *News20*, the similarity loss function does not improve the classification accuracy on clean data, and on *CIFAR100*, the improvement is marginal. However, in Table 3.1 and 3.2, the improvements are significant, which reflects the improvements are mainly benefited from the lower noise rate and the reduced noisy binary paradigm.

3.5 Summary

This chapter proposes a noise reduction perspective on dealing with class label noise by transforming training data with noisy class labels into data pairs with noisy similarity labels. We establish the connection between noisy similarity posterior and clean class posterior and propose a deep learning framework to learn classifiers from the transformed noisy similarity labels. The core idea is to transform pointwise information into pairwise information, which makes the noise rate lower. We also prove that not only the similarity labels but the similarity transition matrix is robust to noise. Experiments are conducted on benchmark datasets, demonstrating the effectiveness of our method.

Chapter 4

LNL from a Curriculum Learning Perspective

Many machine learning algorithms are known to be fragile on simple instance-independent noisy labels. However, noisy labels in real-world data are more devastating since they are produced by more complicated mechanisms in an instance-dependent manner. In this chapter, we target this practical challenge of *Instance-Dependent Noisy Labels* by jointly training (1) a model reversely engineering the noise generating mechanism, which produces an *instance-dependent mapping* between the clean label posterior and the observed noisy label; and (2) a robust classifier that produces clean label posteriors. Compared to previous methods, the former model is novel and enables end-to-end learning of the latter directly from noisy labels. An extensive empirical study indicates that the time-consistency of data is critical to the success of training both models and motivates us to develop a curriculum selecting training data based on their dynamics on the two models' outputs over the course of training. We show that the curriculum-selected data provide both clean labels and high-quality input-output pairs for training the two models. Therefore, it leads to promising and robust classification performance even in notably challenging settings of instance-dependent noisy labels where many SoTA methods could easily fail. Extensive experimental comparisons and ablation studies further demonstrate the advantages and significance of the time-consistency curriculum in learning from instance-dependent noisy labels on multiple benchmark datasets.

4.1 Motivations and Contributions

Two principal methodologies have been developed to address the label noises: (1) detecting samples $(X, \tilde{Y})$ with correct labels $\tilde{Y} = Y$ (empirically, they are the ones with the smallest loss values) and using them to train a clean classifier [37, 133]; (2) learning the noise generating mechanism, i.e., a *transition matrix* T defining the mapping between clean label Y and noisy label $\tilde{Y}$ such that $P(\tilde{Y} \mid X) = T^\top P(Y \mid X)$, where $P(\cdot \mid X)$ denotes the posterior vector, and then using it to build statistically consistent classifiers [64, 88, 129]. Although both methodologies have achieved promising results in the simplified instance-independent (class-dependent) setting, they have non-trivial drawbacks when applied to the more practical but complicated instance-dependent noises: (1) the "small loss" trick is no longer effective in detecting correct labels [18] because the loss threshold drastically varies across instances and is determined by each transition matrix $T(X)$; (2) the instance-dependent transition matrix $T(X)$ is not identifiable given only the noisy sample and it heavily relies on the estimation of clean label Y in the triple $(X, Y, \tilde{Y})$ [129], which is an unsolved challenge in (1).

Therefore, the two learning problems are entangled, i.e., the training of a clean label predictor and the transition matrix estimator depends on each other's accuracy, which substantially relies on the quality of training data $(X, Y, \tilde{Y})$. Specifically, the "small loss" trick cannot provide a high-quality estimation of Y due to the instance-specific threshold of loss. Moreover, the estimation of Y can change rapidly due to the non-stationary loss, which can fluctuate during training and provide inconsistent training signals over time for both models if selected for training. Furthermore, the data subset selection inevitably introduces biases toward easy-to-fit samples and degrades the data diversity [129, 18, 10, 19], which in fact is critical to the training and the accuracy of both models, especially the transition matrix estimator, because easy-to-fit samples usually have extremely sparse transition matrices.

To tackle the above issues, we propose a novel metric "Time-Consistency of Prediction (TCP)" to select high-quality data to train both models. TCP measures the consistency of model prediction for an instance over the course

of training, which reflects whether its given label results in gradients consistent with the majority of other instances, and this criterion turns out to be a more reliable identifier of clean labels. When applied to the training of clean label predictor, TCP is more accurate in clean label detection than "small loss" (or high confidence) criterion, because it avoids the comparison of confidence for samples with instance-dependent loss/confidence thresholds. Moreover, when applied to the training of the transition matrix estimator, TCP measures the time-consistency of predicted noisy labels $\tilde{Y}$. Surprisingly, it also faithfully reflects the correctness of the predicted clean label Y. Since the objective to estimate the transition matrix is defined by both Y and $\tilde{Y}$, selecting samples with high TCP considerably improves the training of the transition matrix estimator. In addition, to exploit the data diversity in training the two models, we apply a curriculum that starts from selecting only a few high TCP data for early-stage training but progressively includes more training data once the two models become maturer and more consistent.

In this chapter, we develop a three-stage training strategy with the TCP curriculum embedded. In every training step, we first update the clean label predictor using selected data with high TCP on this model (Sec.4.3.2). Then, we train the transition matrix estimator given the predicted clean label posterior and the noisy labels on selected data with high TCP on the estimator (Sec.4.3.2). Finally, we fine-tune the clean label predictor directly using the noisy label and the estimated transition matrix (Sec.4.3.3). It is worth noting that the TCP metrics for the two models are updated using the model outputs collected from this dynamic training process without causing additional cost. As demonstrated by extensive empirical studies and experimental comparisons, our method leads to efficient joint training of the two models that mutually benefits from each other and produces an accurate estimation of both the clean label and instance-dependent transition matrix. On multiple benchmark datasets with either synthetic or real-world noises, our method achieves state-of-the-art performance with significant improvements.

4.2 Related Work

To estimate the transition matrix, a cross-validation method can be applied for the binary classification task [81]. For CDN, the transition matrix could be learned by exploiting anchor points [88, 134]. For IDN, the transition matrix for an instance could be approximated by a combination of the transition matrices for the parts of the instance [124] or a Bayes label transition matrix [129]. [131] exploited the causal graph to estimate the transition relations between clean and noisy labels.

Curriculum learning was first proposed by [9], which describes a learning paradigm in which a model is learned by gradually introducing samples of increasing hardness to training. Its effectiveness has been empirically verified in a wide range of applications, e.g., computer vision [17], natural language processing [108], and multitask learning [32]. Curriculum for label-noise learning has been also investigated. MentorNet [48] pretrains an extra network producing a data-driven curriculum selecting data instances to guide the training. When the clean validation data is not available, MentorNet has to use a predefined curriculum. RoCL [145] develops a curriculum learning strategy that smoothly transitions between (1) detection and supervised training on clean data; and (2) relabeling and self-supervision on noisy data. Nevertheless, RoCL has no convergence guarantee and needs extra data augmentations to collect spatial-consistent pseudo labels.

4.3 Methodology

4.3.1 Examples Selection Criterion: Time-Consistency of Prediction

According to the observation that the loss on instances with clean labels is usually smaller than instances with noisy labels, the loss computed at an instantaneous step has been widely adopted as a selection criterion for confident examples [37, 133, 113]. It is because instances with clean labels are mutually consistent with each other in producing gradient updates,

allowing the model to fit them better and thereby make the loss smaller than instances with noisy labels.

Unfortunately, the instantaneous loss was found only work well on the instance-independent label noise [18]. For a deep neural network, because of the non-smooth nature of the loss and the randomness of stochastic gradient descent, the instantaneous loss of each instance can change dramatically between consecutive epochs, leading to a huge gap between training sets selected over consecutive epochs. Therefore, it is necessary to take the training history of each instance into consideration. [146] proposed a robust version of the instantaneous loss as the exponential moving average of it over the course of training. Nevertheless, in the IDN case, each instance with its noisy label is a unique pattern, which is more complex and thereby requires a more robust selection criterion. Apparently, at the instance level, the one-hot prediction of an instance is a more robust metric than the loss because the former has a tolerance to the change of predicted class posterior while the latter has not, i.e., the one-hot prediction remains unchanged if the position of the max element in the predicted class posterior vector is maintained but the cross-entropy loss changes once the predicted class posterior changes.

Inspired by the above insights, we propose a time-consistency of prediction (TCP) metric as follows:

$$\mathrm{TCP}_{t+1}(x) = \frac{t}{t+1}\mathrm{TCP}_{t}(x) + \frac{1}{t+1}\mathrm{InP}_{t+1}(x), \tag{4.1}$$

where $\mathrm{InP}_{t+1}(x) = \mathbb{1}[\hat{y}_{t+1} = \hat{y}_{t}]$ and $\hat{y}_{t}$ is the predicted label at epoch t. This metric considers the prediction consistency over the course of training, which can better describe the IDN data and select confident examples than the previous ones.

To see this, we first manually add IDN at 0.4 noise rate (see Section 4.4 for the noise generation method) onto a benchmark dataset *CIFAR10* and train a ResNet34 [40] for 100 epochs with a constant learning rate. Since no curriculum strategy is applied here, we select confident examples at every epoch t with a fixed number 5,000 according to four types of selection criterion, i.e., instantaneous prediction $\mathrm{InP}_{t}(x)$, instantaneous loss

$\ell(x)$, time-consistency of prediction $\mathrm{TCP}_{\mathrm{t}}(x)$, and time-consistency of loss (defined in the same way as TCP).

Then we count the number of instances with clean labels from the selected confident examples and calculate the clean ratios (ratio of clean instances to selected instances). As shown in Figure 4.1, we can find that the two instantaneous metrics have clean ratios lower than 0.6, which are worse than random selection. As for time-consistency of loss, the clean ratio is slightly higher than the random selection. Those three metrics are basically not discriminative to the noisy data. By contrast, the proposed TCP metric has a distinguishable performance, uplifting more than 20 percent of the clean ratio of the selected confident examples. More empirical studies are provided in Appendix B.1 to support our claims.

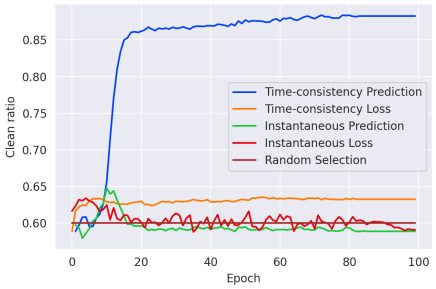
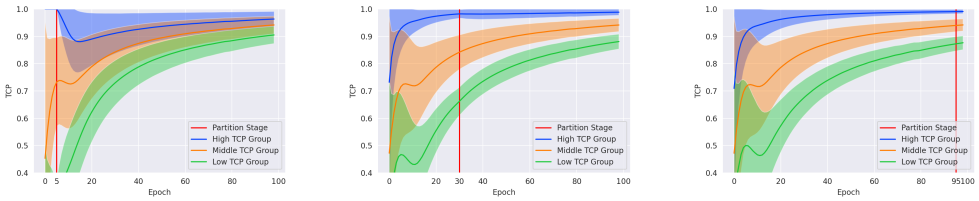


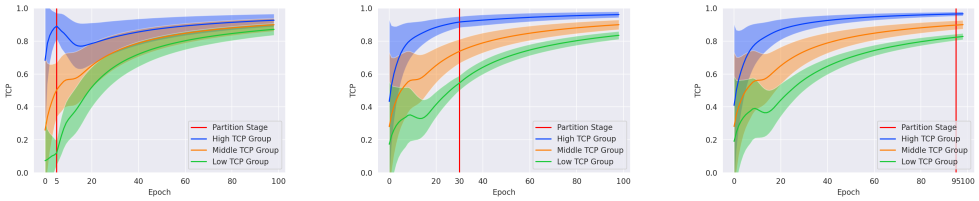
Figure 4.1: Clean ratios of the selected top 5000 instances ranked by four kinds of instance hardness measures, respectively, during a standard training for 100 epochs. The clean ratio of randomly selected instances is 0.6 since the noise rate is 0.4.

Moreover, we partition the whole data into three groups (high TCP (10%), middle TCP (80%), and low TCP (10%)) by the TCP calculated at the start stage (epoch 5), early stage (epoch 30), and end stage (epoch 95). We visualize the mean and variance of the groups through the whole training epochs. As shown in Figure 4.2 and B.3, the start-stage partition fails at the end stage as three groups are entangled together while the early-stage partition shares the almost same pattern as the end-stage partition. Thus, we can conclude that the early-stage TCP is reflective of the property of each instance in the future, which means the time-consistent examples selected in the early stage will not mislead the classifier because their TCP are still high and thus they will still be selected as time-consistent examples in the rest training epochs. Besides, a warmup for the TCP is proved to be necessary.



(a) Partitioned at a start stage. **(b)** Partitioned at an early stage. **(c)** Partitioned at an end stage.

Figure 4.2: TCP (mean and std.) of three groups (high TCP (10%), middle TCP (80%), and low TCP (10%)) partitioned by the TCP calculated at the start stage (epoch 5), early stage (epoch 30), and end stage (epoch 95) during training a ResNet34 on CIFAR10 with IDN-0.4 for 100 epochs.



(a) Partitioned at a start stage. **(b)** Partitioned at an early stage. **(c)** Partitioned at an end stage.

Figure 4.3: TCP (mean and std.) of three groups (high TCP (10%), middle TCP (80%), and low TCP (10%)) partitioned by the TCP calculated at the start stage (epoch 5), early stage (epoch 30), and end stage (epoch 95) during training a ResNet50 on CIFAR100 with IDN-0.4 for 100 epochs.

4.3.2 TCP Guided Curriculum Learning for Instance-Dependent Noisy Labels

Our curriculum contains two sub-curriculum corresponding to two main challenges for solving the IDN problem: (1) detecting examples with clean labels; (2) learning the transition matrix. In this section, we elaborate on how to use TCP to design curriculum to achieve these goals. Below we use clean- and noisy-TCP to refer to time consistency of clean label predictions ($\arg\max \hat{P}(y \mid x)$) and noisy label predictions ($\arg\max \hat{P}(\tilde{y} \mid x)$) over historical training steps, respectively.

Learning a Primary Clean Classifier with High Clean-TCP Instances

In Section 4.3.1, we demonstrate that clean-TCP can be used to select examples with clean labels because they are mutually consistent with each other in producing gradient updates and easy to be learned to have high clean-TCP. However, as shown in Figure 4.1, not all of the selected high clean-TCP instances have clean labels. The reason why those instances have high clean-TCP but not clean labels is because they are far away from the classification boundary and inherently hard to be misclassified even though their given labels are wrong. Namely, their labels get corrected by the classifier, and high clean-TCP indicates the pseudo labels assigned by the classifier are correct.

Therefore, with clean-TCP, we design a curriculum which exploits both examples with clean labels and instances with correct pseudo labels to learn a clean classifier. We theoretically prove that introducing high clean-TCP instance with its pseudo label does not cause *catastrophic forgetting*[1] of the learned confident examples. Consider the situation we have a labeled set $\mathcal{L}$ (in practice it can be the selected confident examples set) and one unlabeled instance x'. By training on $\mathcal{L}$ for one step, we have $\theta_{t+1} = \theta_t - \eta \sum_{x \in \mathcal{L}} \nabla_\theta \ell\left(x; \theta_t\right)$; and by training on $\mathcal{L}$ and x' for one step, we have $\theta'_{t+1} = \theta_t - \eta \left(\sum_{x \in \mathcal{L}} \nabla_\theta \ell\left(x; \theta_t\right) + \nabla_\theta \ell\left(x'; \theta_t\right)\right)$, where θ_t denotes the network parameters at step t and η denotes the learning rate. Then we have

$$\frac{1}{\eta}\left|\sum_{x \in \mathcal{L}}\left[\ell\left(x; \theta_{t+1}\right) - \ell\left(x; \theta'_{t+1}\right)\right]\right| = \left|\frac{p_{t+1}^{\hat{y}'_t}(x')}{p_t^{\hat{y}'_t}(x')} - 1\right|,$$

where $p_t^{\hat{y}'_t}(x')$ is the probability of x' belonging to $\hat{y}'_t$ at step t, and $\hat{y}'_t$ is the prediction (pseudo label) of x' at step t. The detailed derivation is provided in Appendix B.2. If x' is selected with high clean-TCP, $p_t^{\hat{y}'_t}(x')$ is very close to $p_{t+1}^{\hat{y}'_t}(x')$ because it has been verified in Figure 4.2 that instances with high clean-TCP in the early stage maintain their high clean-TCP in the future, which means the loss change can be bounded with a very small value. As a result, changes of the gradient and thereby the parameter

[1] Catastrophic forgetting denotes the tendency of DNNs to forget previously learned information upon learning new information.

of the DNN are also small. If the new data is not properly selected, it will cause catastrophic forgetting as the DNN forgets previously learned information upon learning new information. Therefore, exploiting high clean-TCP instances with pseudo labels helps to correct corrupted labels and learn a clean classifier without causing catastrophic forgetting of the learned examples with correct labels.

In Figure 4.4, we show the clean ratios of the original noisy labels and pseudo labels of instances selected with our curriculum during the whole training process. The clean ratio for pseudo labels maintains an amazing high value, much better than the original clean ratio. Therefore, the clean classifier can be learned by minimizing $\sum_{n=1}^{N} \mathcal{L}\left(f(x_n), y_n^*\right)$, where y^* can be the original noisy label or pseudo label in different learning phases. Implementation details can be found in Section 4.3.3.

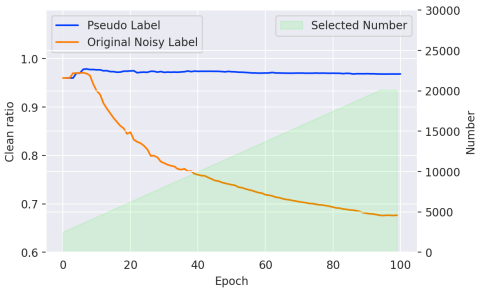


Figure 4.4: Clean ratios of selected high clean-TCP examples w.r.t. their original noisy labels and pseudo labels with linear growth of the selected number during our curriculum learning on CIFAR10 with IDN-0.4 for 100 epochs.

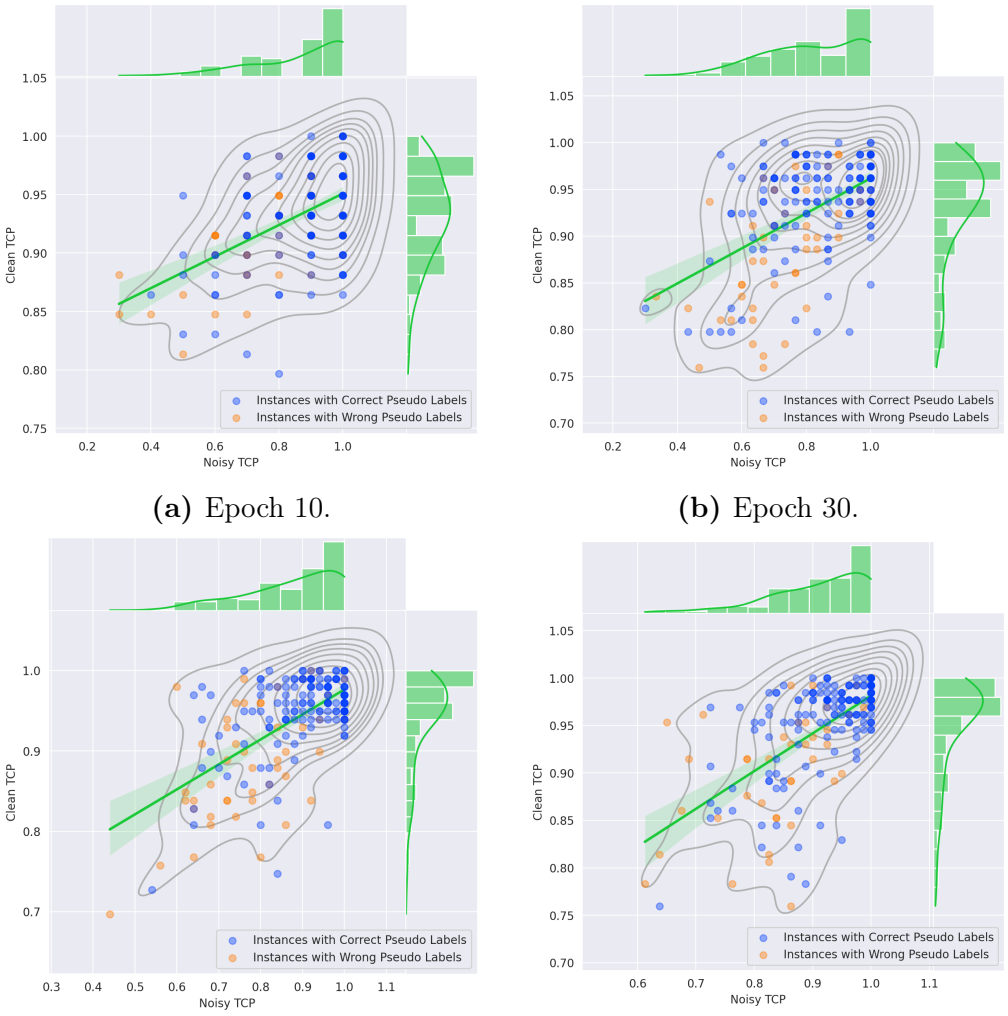
Learning a Transition Matrix with High Noisy-TCP Instances

The transition matrix is not identifiable by only exploiting noisy data without introducing additional assumptions, therefore we formulate the objective function for learning the transition matrix based on the equation $P(\tilde{Y} \mid X) = T^{\top}(X)P\left(Y \mid X\right)$ as follow:

$$\min_{T} \quad \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}\left(T^{\top}(x_n)f(x_n), \tilde{y}_n\right), \qquad \text{where } f(x_n) = \hat{P}\left(y \mid x_n\right). \quad (4.2)$$

To select the high-quality triplets $(X, Y, \tilde{Y})$ for the above objective, two conditions should be considered. First, it is necessary for $f(\cdot)$ to output a precise clean class posterior, otherwise, T cannot be optimized in the correct direction, in the case $\tilde{y}$ is given and $f(\cdot)$ has been learned in advance and is

fixed. As we discussed above, instances with high clean-TCP tend to have the correct pseudo label, and thereby a precise clean class posterior, which satisfies this necessary condition. Second, the noisy-TCP should be high. By treating $T^\top(x)f(x)$ as a whole predictor for $\tilde{y}$, the corresponding new objective is to predict $\tilde{y}$. Therefore, high noisy-TCP instances naturally indicate the instance is learned better and faster for predicting $\tilde{y}$, leading to stable and fast learning.



(a) Epoch 10. **(b)** Epoch 30.

(c) Epoch 50. **(d)** Epoch 80.

Figure 4.5: Data distribution in terms of noisy- and clean-TCP at epoch 10/30/50/80 during our curriculum learning on CIFAR10 with IDN-0.4 for 100 epochs.

We discover that the noisy-TCP inherently has a strong correlation

with clean-TCP so we can use it to select triplets fulfilling both conditions above. To see this, at each epoch, we calculate the Spearman rank-order correlation coefficient[2] between the noisy-TCP and clean-TCP of the whole dataset. Besides, we calculate the clean ratio of the selected high noisy-TCP instances w.r.t. their pseudo labels. In Figure 4.5, we show the data distribution in terms of clean- and noisy-TCP at epochs 10/30/50/80 through the training procedure. The green regression line partially implies the linear correlation between the clean- and noisy-TCP. Also, instances with correct pseudo labels are mainly distributed in the high TCP area, and vice versa. As shown in 4.6, the Spearman rank-order correlation coefficient is above 0.6 after 10 epochs with a consistent 0 p-value, roughly indicating that noisy-TCP is strongly Spearman rank-order correlated with clean-TCP for 100% sure. Meanwhile, the clean ratio is consistently above 0.8, which means those high noisy-TCP instances also have correct clean predictions and thereby probably precise clean class posterior. Note that the clean ratio decreases at the late stage because the curriculum selects almost all the data at the end. Overall, high noisy-TCP instances not only are naturally stable for the new objective to predict $\tilde{y}$ but also satisfy the necessary condition to have precise clean class posterior, which makes them perfect examples for learning the transition matrix.

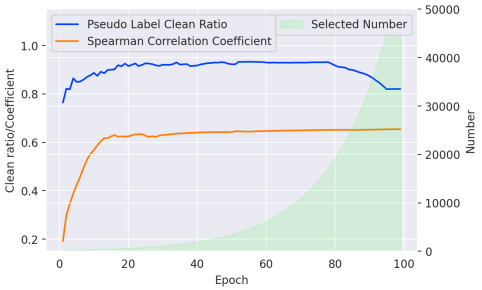


Figure 4.6: Clean ratios of selected high noisy-TCP examples w.r.t. their pseudo labels with exponential growth of the selected number and Spearman rank-order correlation coefficient between the noisy- and clean-TCP during our curriculum learning on CIFAR10 with IDN-0.4 for 100 epochs.

4.3.3 TCP Guided Curriculum Learning Algorithm

The main steps of our algorithm are summarized in Algorithm 3. First, we warm up the feature extractor ϕ, classification layer c by minimizing a standard cross-entropy (CE) loss on noisy data, and meanwhile compute

[2]The Spearman rank-order correlation coefficient is a nonparametric measure of the monotonicity of the relationship between two sets [137].

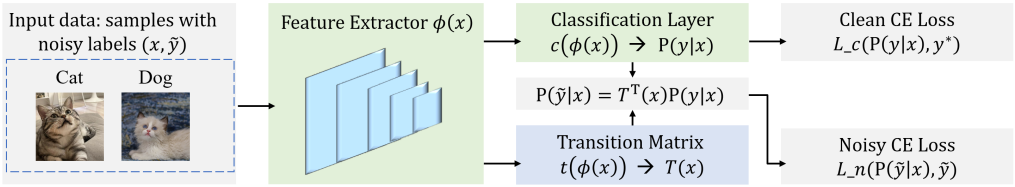


Figure 4.7: An overview of the proposed method. The second image of cat has a noisy label as "dog". The transition matrix $T(\cdot) = t(\phi(\cdot))$ and classifier $f(\cdot) = c(\phi(\cdot))$ share a common feature extractor.

the clean-TCP for every instance. Then, we warm up the transition matrix layer t with high clean-TCP instances and obtain the noisy-TCP for every instance. From now on, iteratively, high clean-TCP instances are fed to the clean classifier (green part in Figure 4.7) to train a primary clean classifier with the clean CE loss, and based on the primary clean classifier, instances with high noisy-TCP are fed to the transition matrix (blue part in Figure 4.7) to train a transition matrix with the noisy CE loss while the parameters of the primary clean classifier are frozen. Then the clean classifier gets improved by being fine-tuned on the whole data with the fixed transition matrix. The clean- and noisy-TCP of every instance are updated at the end of each epoch. Finally, a transition matrix with a small estimation error and a clean classifier with a performance improvement can be obtained.

4.4 Experiments

Dataset. We employ three widely used datasets, i.e., *F-MNIST* [126], *SVHN* [82], and *CIFAR10/100* [54], and four versions of the real-world noisy dataset *CIFAR10N* [117], *CIFAR100N* [117], and *Clothing1M* [127]. F-MNIST contains 60,000 training images and 10,000 test images with 10 classes. SVHN and CIFAR10 both have 10 classes of images, but the former contains 73,257 training images and 26,032 test images, and the latter contains 50,000 training images and 10,000 test images while CIFAR100 has 100 classes. CIFAR10N (CIFAR100N) provides CIFAR10 (CIFAR100) images with human-annotated noisy labels obtained from Amazon Mechanical Turk. Four versions of CIFAR10N label sets are employed here, three

Algorithm 2 TCP Guided Curriculum Learning Algorithm.

Input: Noisy training sample $\mathcal{D}$.
Modules and Hyperparameters: feature extractor ϕ, classification layer c, transition matrix layer t, $f(\cdot) = c(\phi(\cdot))$, $T(\cdot) = t(\phi(\cdot))$, number sequences N_c and N_t, training epoch e_1, e_2, and e_3.
Warmup clean-TCP:
for e in $\{1, \cdots, e_1\}$ **do**
Train ϕ and c on $\mathcal{D}$ by minimizing a standard CE loss $\sum_{n=1}^{N} \mathcal{L}\left(f(x_n), \tilde{y}_n\right)$.
Record the clean prediction and calculate the clean-TCP.
end for
Warmup noisy-TCP:
for e in $\{1, \cdots, e_2\}$ **do**
Select $N_c[e]$ high clean-TCP instances as $\mathcal{D}_{\mathrm{c}}$.
Train ϕ, c and t on $\mathcal{D}_{\mathrm{c}}$ by minimizing $\sum_{n=1}^{N} \mathcal{L}_n\left(f(x_n), \tilde{y}_n\right) + \sum_{n=1}^{N} \mathcal{L}\left(T^{\top}(x_n) f(x_n), \tilde{y}_n\right)$.
Record the clean and noisy prediction and calculate the clean- and noisy-TCP.
end for
Curriculum training:
for e in $\{1, \cdots, e_3\}$ **do**
Select $N_t[e_2 + e]$ high noisy-TCP instances as $\mathcal{D}_{\mathrm{t}}$.
Train t while fixing ϕ and c on $\mathcal{D}_{\mathrm{t}}$ by minimizing $\sum_{n=1}^{N} \mathcal{L}_n\left(T^{\top}(x_n) f(x_n), \tilde{y}_n\right)$.
Select $N_c[e_2 + e]$ high clean-TCP instances as $\mathcal{D}_{\mathrm{c}}$.
Train ϕ and c on $\mathcal{D}_{\mathrm{c}}$ by minimizing $\sum_{n=1}^{N} \mathcal{L}_c\left(f(x_n), y_n^*\right)$, where y^* is the pseudo label.
Fix t and fine-tune ϕ and c on $\mathcal{D}$ by minimizing $\sum_{n=1}^{N} \mathcal{L}_n\left(T^{\top}(x_n) f(x_n), \tilde{y}_n\right)$.
Record the clean and noisy prediction and calculate the clean- and noisy-TCP by Eq. (4.1).
end for
Output: Optimized feature extractor ϕ, classification layer c, transition matrix layer t.

of which are labeled by three independent workers (named *CIFAR10N-1/2/3*) and one of which is negatively aggregated from the above three sets (named *CIFAR10N-W*). Clothing1M has 1M images with real-world noisy labels and additional 50k, 14k, 10k images with clean labels for training, validation and test, and we only use noisy training set in the training phase.

For all the datasets, we leave out 10% of the training data as a validation set, which is for model selection. The final test model is selected with the highest validation accuracy.

Noisy labels generation. For clean datasets, we artificially corrupt the class labels of training and validation sets following the instance-dependent noisy labels generalization method in [124]. We generate noisy datasets of $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ five noise rates.

Baselines and measurements. On synthetic noisy datasets, without introducing data augmentation techniques and semi-supervised learning, we compare the proposed method TCP with the following baselines: (i). CE, which optimizes the standard cross-entropy loss on noisy datasets. (ii). Decoupling [72], which trains two networks on samples whose predictions are different. (iii). MentorNet [48], Co-teaching [37], and Co-teaching+ [133], that mainly handle noisy labels by training on instances with small loss. (iv). Joint [106], which jointly optimizes labels and network parameters. (v). DMI [128], which uses a novel information-theoretic loss function to learn a robust classifier. (vi). Forward [88], Reweight [64], and T-Revision [125], that utilize a class-dependent transition matrix T to correct the loss function. (vii). PTD [124] and Bayes [129], estimate instance-dependent transition matrix under some additional assumptions; CRUST [78] iteratively selects subsets of clean data points that provide an approximately low-rank Jacobian matrix; CausalINL [131] exploits the causal graph to estimate the transition relations between clean and noisy labels. On real-world noisy datasets, we apply the transition matrix learning and fine-tuning parts to the SoTA method Dividemix [60], i.e., at each epoch, in addition to the Dividemix training, we select high noisy-TCP data to learn the transition matrix and use it to fine-tune the whole data. Then we compare this combined method **TCP-D** with the following SoTA methods: (i). PES [4]. (ii). Dividemix [60]. (iii). CORES [18]. (iv). ELR+ [63]. (v). JoCoR [116]. (vi). CAL [147]. We use a ResNet18 network for F-MNIST, a ResNet34 network for SVHN and CIFAR10, a ResNet50 network for CIFAR100, a PreAct-ResNet18 for CIFAR10N and CIFAR100N, and a pre-trained ResNet50 network for Clothing1M. Classification accuracy is employed to evaluate the performance of each model on

the clean test set. Results over 5 trials on all datasets except Clothing1M, for which the result is over 1 trial, are reported.

Implementation details. We use a ResNet18 network for F-MNIST, a ResNet34 network for SVHN and CIFAR10, a ResNet50 network for CIFAR100, a PreAct-ResNet18 for CIFAR10N and CIFAR100N, and a pre-trained ResNet50 network for Clothing1M. The transition matrix is modeled by a linear layer which transforms the latent representation vector to a c^2 vector, and then reshaped to a $c \times c$ matrix. The batchsize is set to 32 for Clothing1M and 128 for others. The weight decay is set to $1e^{-4}$, $5e^{-4}$, 0, $5e^{-4}$, and 0.001 for F-MNIST, SVHN, CIFAR10/100, CIFAR10N/CIFAR100N, and Clothing1M, respectively. For synthetic noisy datasets, we use the Adam optimizer. At the warmup clean-TCP stage, the learning rate is initialized to 0.001 and decayed every 5 epochs with 50 epochs in total by a factor of 0.1, 1/3, and 1 for F-MNIST, SVHN, and CIFAR10/100, respectively. In the rest of 100-epoch training, the leaning rate of the feature extractor ϕ and classification layer c is $1e^{-4}$ and divided by 10 at epoch 30 and 80; the leaning rate of the transition matrix layer t is $3e^{-4}$ before epoch 30 and $1e^{-5}$ otherwise. The learning rate for fine-tuning is $1e^{-6}$. For real-world noisy dataset CIFAR10N/CIFAR100N and Clothing1M, we follow the optimization method as Dividemix. For CIFAR10N/CIFAR100N, in the warmup clean-TCP stage, the learning rate is initialized to 0.001 and decayed every 5 epochs with 50 epochs in total by a factor of 1/3. In the rest of 300-epoch training, the leaning rate of the transition matrix layer t is $6e^{-3}$ before epoch 80 and $2e^{-4}$ between epoch 80 and 150, and $2e^{-4}$ otherwise. The learning rate for fine-tuning is $2e^{-3}$ before epoch 80, and $6e^{-4}$ between epoch 80 and 150, and $2e^{-4}$ between epoch 150 and 250, and $2e^{-5}$ otherwise. For Clothing1M, in the warmup clean-TCP stage, the learning rate is initialized to 0.002 and decayed every 2 epochs with 5 epochs in total by a factor of 1/3. In the rest of 20-epoch training, the leaning rate of the transition matrix layer t is $6e^{-4}$ before epoch 8 and $2e^{-5}$ between epoch 8 and 12, and $5e^{-6}$ otherwise. The learning rate for fine-tuning is $2e^{-4}$ before epoch 10, and $6e^{-5}$ between epoch 10 and 14, and $2e^{-5}$ between epoch 14 and 17, and $2e^{-6}$ otherwise.

Comparison with the State-of-the-Arts. We compare TCP with

Table 4.1: Means and stds of classification accuracy on *CIFAR10* with different label noise rates.

	IDN-0.1	IDN-0.2	IDN-0.3	IDN-0.4	IDN-0.5
CE	74.49±0.29	68.21±0.72	60.48±0.62	49.84±1.27	38.86±2.71
Decoupling	74.09±0.78	70.01±0.66	63.05±0.65	44.27±1.91	38.63±2.32
MentorNet	74.45±0.66	70.56±0.34	65.42±0.79	46.22±0.98	39.89±2.62
Co-teaching	76.99±0.17	72.99±0.45	67.22±0.64	49.25±1.77	42.77±3.41
Co-teaching+	74.27±1.20	71.07±0.77	64.77±0.58	47.73±2.32	39.47±2.14
Joint	76.89±0.37	73.89±0.34	69.03±0.79	54.75±5.98	44.72±7.72
DMI	75.02±0.45	69.89±0.33	61.88±0.64	51.23±1.18	41.45±1.97
Forward	73.45±0.23	68.99±0.62	60.21±0.75	47.17±2.96	40.75±2.09
Reweight	74.55±0.23	68.42±0.75	62.58±0.46	50.12±0.96	41.08±2.45
T-Revision	74.61±0.39	69.32±0.64	64.09±0.37	50.38±0.87	42.57±3.27
CRUST	76.20±0.90	76.41±1.56	71.06±2.21	64.59±3.32	52.50±0.81
CausalINL	79.08±0.40	75.65±1.04	67.70±0.82	49.19±0.82	47.83±1.58
PTD	78.71±0.22	75.02±0.73	71.86±0.42	56.15±0.45	49.07±2.56
Bayes	80.17±1.32	79.51±1.21	76.43±1.99	69.53±3.24	57.42±4.37
TCP	**82.49±0.51**	**80.88±0.90**	**79.32±1.59**	**76.03±2.58**	**60.66±5.71**

Table 4.2: Means and stds of classification accuracy on *F-MNIST* with different label noise rates.

	IDN-0.1	IDN-0.2	IDN-0.3	IDN-0.4	IDN-0.5
CE	88.54±0.31	88.38±0.42	84.22±0.35	68.86±0.78	51.42±0.66
Decoupling	89.27±0.31	86.50±0.35	85.33±0.47	78.54±0.53	57.32±2.11
MentorNet	90.00±0.34	87.02±0.41	86.02±0.82	80.12±0.76	58.62±1.36
Co-teaching	90.82±0.33	87.89±0.41	86.88±0.32	82.78±0.95	63.22±1.58
Co-teaching+	90.92±0.51	89.77±0.45	88.52±0.45	83.57±1.77	59.32±2.77
Joint	70.24±0.99	56.83±0.45	51.27±0.67	44.24±0.78	30.45±0.45
DMI	91.98±0.62	90.33±0.21	84.81±0.44	69.01±1.87	51.64±1.78
Forward	89.05±0.43	88.61±0.43	84.27±0.46	70.25±1.28	57.33±3.75
Reweight	90.33±0.27	89.70±0.35	87.04±0.35	80.29±0.89	65.27±1.33
T-Revision	91.56±0.31	90.68±0.66	89.46±0.45	84.01±1.24	68.99±1.04
CRUST	89.53±0.55	89.20±0.58	86.68±0.92	83.48±1.55	69.59±3.60
CausalINL	90.14±0.31	88.83±0.37	85.38±1.49	83.82±2.29	69.55±4.11
PTD	91.01±0.22	90.03±0.32	87.68±0.42	84.03±0.52	72.43±1.76
Bayes	92.01±0.22	91.42±0.71	89.64±0.41	81.21±1.13	74.62±2.47
TCP	**92.64±0.22**	**92.15±0.38**	**91.62±0.59**	**90.56±0.79**	**77.49±2.88**

multiple baselines using the same network architecture. Table 4.1 show the results on CIFAR10 with different rates of IDN from 0.1 to 0.5, respectively. TCP outperforms baselines across all datasets and noise rates. The improvement is significant when the noise rate is large. Tables 4.2, 4.3,

Table 4.3: Means and stds of classification accuracy on *SVHN* with different label noise rates.

	IDN-0.1	IDN-0.2	IDN-0.3	IDN-0.4	IDN-0.5
CE	90.77±0.45	90.23±0.62	86.33±1.34	65.66±1.65	48.01±4.59
Decoupling	90.49±0.15	90.47±0.66	85.27±0.34	82.57±1.45	42.56±2.79
MentorNet	90.28±0.12	90.37±0.37	86.49±0.49	83.75±0.75	40.27±3.14
Co-teaching	91.33±0.31	90.56±0.67	88.93±0.78	85.47±0.64	45.90±2.31
Co-teaching+	93.05±1.20	91.05±0.82	85.33±2.71	57.24±3.77	42.56±3.65
Joint	86.01±0.34	78.58±0.72	76.34±0.56	65.14±1.72	46.78±3.77
DMI	93.51±1.09	93.22±0.62	91.78±1.54	69.34±2.45	48.93±2.34
Forward	90.89±0.63	90.65±0.27	87.32±0.59	78.46±2.58	46.27±3.90
Reweight	92.49±0.44	91.09±0.34	90.25±0.77	84.48±0.86	45.46±3.56
T-Revision	94.24±0.53	94.00±0.88	93.01±0.83	88.63±1.37	49.02±4.33
CRUST	93.22±1.32	91.55±0.36	88.64±1.43	80.75±2.78	58.30±2.77
CausalINL	92.38±0.44	91.40±0.86	90.23±1.60	84.50±1.71	68.06±5.12
PTD	93.21±0.45	92.36±0.68	90.57±0.42	86.78±0.63	55.88±3.73
Bayes	94.71±0.44	94.02±1.32	91.38±1.94	85.55±3.17	75.46±3.79
TCP	**94.90±0.11**	**94.60±0.20**	**93.92±1.37**	**94.09±0.34**	**84.92±8.40**

and 4.4 show the results on F-MNIST, SVHN, and CIFAR100. Table 4.5 shows the results on real-world noisy datasets CIFAR10N-1/2/3/W, CIFAR100N, and Clothing1M. Overall, TCP-D achieves the best test accuracy on real-world noisy datasets. Note that the results of baselines on CIFAR10N and CIFAR100N are taken from the official leaderboard `http://www.noisylabels.com/`.

Comparison on the transition matrix estimation error. We compare the transition matrix estimation error of our method with the instance-independent method Forward [88], and two instance-dependent methods PTD [124] and Bayes [129]. As shown in Figure 4.8, our method achieves the consistent best estimation error on CIFAR10 with different noise rates.

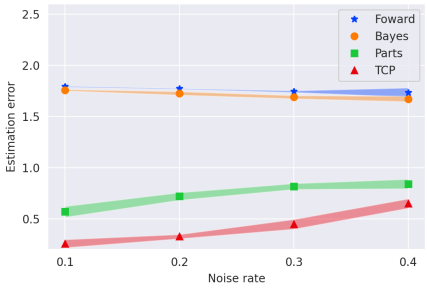


Figure 4.8: Transition matrix estimation errors of four methods on CIFAR10 with noise from IDN-0.1 to IDN-0.4.

Ablation study. We study the effect of removing different components of our methods to provide insights into what makes TCP successful

Table 4.4: Means and stds of classification accuracy on *CIFAR100* with different label noise rates. Note that PTD is not applicable to CIFAR100 which has large classes due to its matrix factorization component.

	IDN-0.1	IDN-0.2	IDN-0.3	IDN-0.4	IDN-0.5
CE	36.80±1.62	31.64±1.04	30.67±2.67	24.00±1.76	20.24±1.49
Decoupling	37.16±0.86	33.01±1.61	31.65±2.62	24.72±2.51	20.13±2.72
MentorNet	37.95±0.93	33.72±1.03	32.04±1.97	26.93±2.35	21.86±2.30
Co-teaching	38.57±0.95	35.60±1.49	33.77±1.91	26.17±2.35	21.96±2.51
Co-teaching+	37.92±1.04	34.51±1.43	33.13±2.04	25.98±2.12	21.88±2.43
Joint	38.96±0.73	35.91±1.22	34.23±1.47	28.75±3.69	23.89±3.93
DMI	37.60±0.84	34.72±1.38	32.87±1.60	28.60±2.16	23.25±2.81
Forward	37.00±1.55	32.72±2.67	31.60±2.84	27.24±2.89	21.13±2.46
Reweight	37.11±0.98	33.98±1.68	32.60±1.22	27.83±1.27	22.01±3.26
T-Revision	38.03±1.05	34.42±2.32	33.60±1.98	28.15±3.69	22.12±3.67
CRUST	43.96±1.25	41.75±1.32	38.60±2.01	32.42±5.23	24.41±2.12
CausalINL	38.02±0.78	36.31±1.23	32.23±9.23	27.63±4.38	22.42±2.16
Bayes	40.76±1.98	36.56±1.20	29.26±1.67	24.38±1.39	17.66±0.94
TCP	**49.65±0.43**	**46.28±2.56**	**44.12±1.92**	**39.88±0.62**	**29.45±2.35**

Table 4.5: Means and stds of classification accuracy on real-world noisy datasets.

	CIFAR10N-1	CIFAR10N-2	CIFAR10N-3	CIFAR10N-W	CIFAR100N	Clothing1M
PES (semi)	95.06±0.15	95.19±0.23	95.22±0.13	92.68±0.22	70.36±0.33	74.29
DivideMix	95.16±0.19	95.23±0.07	95.21±0.14	92.56±0.42	**71.13±0.48**	74.30
CORES	94.45±0.14	94.88±0.31	94.74±0.03	91.66±0.09	61.15±0.73	73.24
ELR+	94.43±0.41	94.20±0.24	94.34±0.22	91.09±1.60	66.72±0.07	74.31
JoCoR	90.30±0.20	90.21±0.19	90.11±0.21	83.37±0.30	59.97±0.24	70.30
CAL	90.93±0.31	90.75±0.30	90.74±0.24	85.36±0.16	61.73±0.42	74.21
TCP-D	**95.51±0.06**	**95.37±0.08**	**95.43±0.04**	**93.36±0.09**	70.09±0.13	**74.41**

in Table 4.6. TCP w/o $\mathcal{D}_{\mathrm{c}}$ indicates that we do not select high clean-TCP data $\mathcal{D}_{\mathrm{c}}$ to learn the clean classifier while TCP w/o $\mathcal{D}_{\mathrm{t}}$ indicates that we do not select high noisy-TCP data $\mathcal{D}_{\mathrm{t}}$ to learn the transition matrix and use it to fine-tune the clean classifier. Results show that the performances of both reduced methods decrease. Without $\mathcal{D}_{\mathrm{c}}$, the primary clean classifier cannot be learned, and thus the transition matrix cannot be learned well. Without $\mathcal{D}_{\mathrm{t}}$, the transition matrix is not learned, and thus the whole noisy data cannot be fully exploited to build a consistent classifier. To sum up, the learning of the clean classifier and the transition matrix benefit and boost each other.

Table 4.6: Ablation study results on CIFAR10 and CIFAR100.

	CIFAR10		CIFAR10	
	IDN-0.2	IDN-0.4	IDN-0.2	IDN-0.4
TCP	**80.88±0.90**	**76.03±2.58**	**46.28±2.56**	**39.88±0.62**
TCP w/o $\mathcal{D}_c$	79.94±0.86	75.28±2.26	45.26±0.50	35.99±0.71
TCP w/o $\mathcal{D}_t$	79.08±0.70	74.98±1.64	43.02±0.56	35.50±1.53

Training cost. Due to the light-weight and simple network architecture, our method is more time-efficient and scalable than those methods, which employs dual networks or requires data augmentations for semi-supervised learning. We report the time costs below to demonstrate this advantage.

Table 4.7: The average time of training each component on CIFAR10 and CIFAR100 with ResNet34 on NVIDIA 3090.

	Standard training	Fine-tune	Estimating T
CIFAR10	38.42s	46.19s	1.34s
CIFAR100	37.37s	47.06s	1.57s

The additional cost caused by estimating T and fine-tune is small. For each epoch, the additional time cost of estimating-T part is neglectable when compared with the cost of one standard training epoch minimizing the cross-entropy loss. This is because estimating T only updates the parameters of a $c \times c$ linear layer, where c is the number of classes. The time cost of fine-tune part is slightly bigger than one standard training epoch. Fortunately, both parts are not necessary to be applied in every epoch. In our experiments, we only apply them at the last 50 epochs. Moreover, since traning clean classifier part (line 7 in Algorithm 3) and estimating T part only involves the high clean-TCP and high noisy-TCP data rather than the whole data, which save plenty of time for the fine-tune part. Therefore, in practice, our method can easily adapt and scale to meet realistic settings.

4.5 Summary

In this chapter, we study the instance-dependent label noise (IDN) problem, which is a more general and practical setting than the previously addressed instance-independent label noise problem. Targeting the main challenges, we propose a novel time-consistency metric, i.e., TCP for the IDN problem. Based on TCP, we can detect examples with clean labels or correct pseudo labels better than the existing measures, and allocate reliable triplets for learning the transition matrix. Then we design an assumption-free curriculum that learns the clean classifier, as well as the transition matrix simultaneously. Through extensive experiments, we empirically demonstrate that the proposed method remarkably surpasses the baselines on many datasets with both synthetic noise and real-world noise, and achieves the smallest transition matrix estimation error than existing methods.

Chapter 5

LNL incorporating Fairness Concerns

With the widespread use of machine learning systems in our daily lives, it is important to consider fairness as a basic requirement when designing these systems, especially when the systems make life-changing decisions, e.g., *COMPAS* algorithm helps judges decide whether to release an offender. For another thing, due to the cheap but imperfect data collection methods, such as crowdsourcing and web crawling, label noise is ubiquitous, which unfortunately makes fairness-aware algorithms even more prejudiced than fairness-unaware ones, and thereby harmful. To tackle these problems, we provide general frameworks for learning fair classifiers with *instance-dependent label noise*. For statistical fairness notions, we rewrite the classification risk and the fairness metric in terms of noisy data and thereby build robust classifiers. For the causality-based fairness notion, we exploit the internal causal structure of data to model the label noise and *counterfactual fairness* simultaneously. Experimental results demonstrate the effectiveness of the proposed methods on real-world datasets with controllable synthetic label noise.

5.1 Motivations and Contributions

Machine learning systems have been widely adopted in our daily life. The overwhelming advantages of these systems are that they never get tired, and they approach (and sometimes surpass) human-level benchmarks on a wide array of tasks [24, 100]. Thereby, they are entrusted with important

tasks, i.e., making high-stakes decisions in loan applications [80], dating and hiring [14, 21], and even parole [26]. Nevertheless, machine learning algorithms are very sensitive to biases which render their decisions *unfair* [74, 3, 85]. One canonical example is a decision support tool used by U.S. courts to assess the likelihood of a defendant becoming a recidivist, called COMPAS [26]. A bias against African-Americans was found with this software in an analysis performed by the news organization ProPublica: COMPAS is more likely to assign a higher risk score to African-American offenders than to Caucasians with the same profile.

To mitigate the bias in machine learning algorithms, plenty of fairness metrics and methods have been proposed. However, label noise degenerates these fairness metrics and could make some fairness-aware algorithms even more prejudiced than fairness-unaware ones. To see this, first, we add two types of label noise, i.e., class-dependent label noise (CDLN) and instance-dependent label noise (IDLN), onto a benchmark dataset *ADULT*[1] [27]. For class-dependent label noise, given clean label Y, the noisy label $\tilde{Y}$ is conditionally independent of the instance X, i.e., $P(\tilde{Y} \mid Y, X) = P(\tilde{Y} \mid Y)$. Instance-dependent label noise is more complex and can capture the true structure of real-world datasets better. The noise rates are set to 0.3 and 0.4:

$$P(\tilde{Y} = -1 \mid Y = 1, X) = P(\tilde{Y} = 1 \mid Y = -1, X) = 0.3 \ (0.4). \tag{5.1}$$

Then we implement the algorithm (p-Fair) in [136] to learn a fair classifier. [136] considered two distinct notions: disparate treatment and disparate impact [7], and employed p%-rule:

$$\min \left(\frac{P\left(\hat{Y} = 1 \mid A = 1\right)}{P\left(\hat{Y} = 1 \mid A = 0\right)}, \frac{P\left(\hat{Y} = 1 \mid A = 0\right)}{P\left(\hat{Y} = 1 \mid A = 1\right)} \right) \geq \frac{p}{100}, \tag{5.2}$$

as a constraint in the objective function, where $\hat{Y}$ is the predicted label and A is the protected attribute. As shown in Table 5.1, the fairness-aware method (p-Fair) gives more unfair and misleading decisions than the vanilla

[1]The ADULT dataset is from UCI ML Repository with gender as the sensitive attribute

Table 5.1: Means and Stds of classification accuracy and fairness score (p value. The higher the value, the better the fairness.) on ADULT dataset with two kinds of label noise over 5 trials. UC denotes the method which optimizes the training loss unconstrainedly.

	CDLN-0.3		CDLN-0.4		IDLN-0.3		IDLN-0.4	
	Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness
p-Fair	80.46±0.35	28.37±4.28	79.90±0.33	31.44±7.42	80.35±0.26	24.18±6.14	79.86±0.29	25.07±9.87
UC	83.47±0.30	28.89±4.14	82.76±0.42	32.03±6.20	83.38±0.24	25.93±5.46	82.84±0.45	26.37±8.23

unconstrained method (UC) under the influence of both kinds of label noise. At the same noise rate, IDLN is more harmful and thus more challenging.

For fairness-aware algorithms employing causality-based fairness notions, the fairness metrics could be robust to label noise to some extent. For example, counterfactual fairness [56] requires that changing the value of protected attribute A, while holding things that are not causally dependent on A constant, will not change the distribution of the predicted label. One straightforward strategy to achieve counterfactual fairness is to build a classifier only consisting of the non-descendants of A. From Figure 5.2, we can see that the label noise does not change the internal causal structure of instances. The original non-descendant Z is still the non-descendant of A, which means the classifier built only with Z is robust to label noise with respect to the counterfactual fairness. Although the fairness is maintained, the decline in accuracy is unavoidable. Figure 5.1 shows that the classification accuracy of the classifier only using non-descendants Z decreases as the noise rate increases. Especially when the data are clean, the gap between the counterfactually fair classifier and the unconstrained classifier is huge, indicating there is a huge information loss of the counterfactually fair classifier.

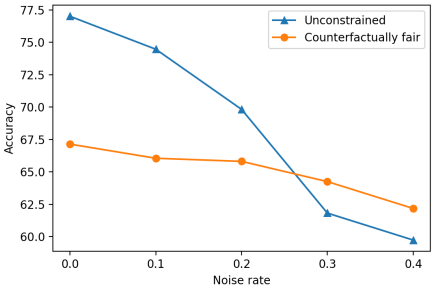


Figure 5.1: Means of classification accuracy on ADULT dataset over 5 trials.

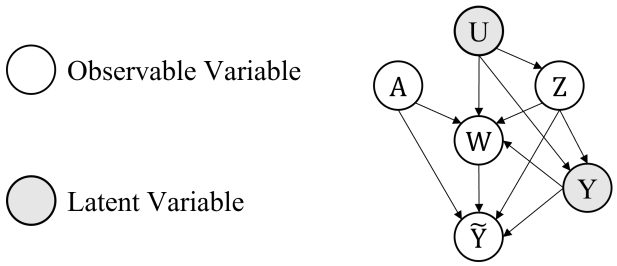


Figure 5.2: Postulated causal graph. Label noise does not change the internal structure of (A, W, Z).

In this chapter, we provide general frameworks for learning fair classifiers with instance-dependent label noise. For statistical fairness notions, we rewrite the classification risk and the fairness metric in terms of noisy data and thereby build robust classifiers. For the causality-based fairness notion, we exploit the internal causal structure of data to model the label noise and counterfactual fairness [56] simultaneously. Specifically, we postulate a general causal graph as shown in Figure[2] 5.2 and employ the variational autoencoder (VAE) framework [52] to make full use of the causal graph which can infer latent variables U and Y by maximizing the joint likelihood of observable variables. In this way, our method also compensates for the information loss, because W contains information from its parents A and U, and we extract the U-part information in W by reconstructing W with U.

5.2 Related Work

To mitigate the bias in machine learning algorithms, plenty of methods, that can be roughly divided into two broad groups, have been proposed. The first group of methods focuses on the statistical fairness notions, which discover the discrepancy of statistical metrics between individuals or subpopulations, e.g., statistical parity [28], equalized odds [38], and predictive parity [20]. This group of methods only considers the correlation but ignores causal effect relations within the data, which can hardly assess the fairness sufficiently [46]. The second group of methods focuses on the causality-based fairness notions, which additionally employs *causal graphs*

[2]We will take benchmark dataset *ADULT* [27] as an example to demonstrate how this causal graph interprets the data in Section 5.3.2.

to take knowledge about the structure of real-world datasets into consideration [71], e.g., fair on average causal effect [50], counterfactual fairness [56], and counterfactual error rates [141].

5.3 Methodology

We consider the binary fair classification problem. Let $\mathcal{D}$ be the distribution of a pair of random variables $(X, Y) \in \mathcal{X} \times \{-1, 1\}$, where $\mathcal{X} \subset \mathbb{R}^d$ and d represents the feature dimension. X can be denoted in detail as a triple (A, Z, W), where A is a protected attribute; Z is a non-descendant variable of A, denoting some root-level attributes; W is the low-level attributes. In real-world datasets, the clean label cannot be observed. Instead, we can only observe the noisy label $\tilde{Y}$. In this case, we have a sample $\{(a_1, z_1, w_1, \tilde{y}_1), \ldots, (a_n, z_n, w_n, \tilde{y}_n)\}$ drawn from a noisy distribution $\mathcal{D}_\rho$ of the random variables $(A, Z, W, \tilde{Y})$ as shown in Figure 5.2.

We follow [89] to use the Directed Acyclic Graph (DAG) with arrows pointing from the parent (direct cause) node to the child (direct effect) node as a formalism to represent causal relationships. Based on the DAG, we use structural causal models (SCMs) to represent the causal mechanism underlying the data distribution: variables can be expressed by a function of their parents with exogenous noise. For Figure 5.2, the corresponding structural causal model can be written as

$$Z = f(U, \varepsilon_Z), Y = f(U, Z, \varepsilon_Y), W = f(U, A, Z, Y, \varepsilon_W), \tilde{Y} = f(A, Z, Y, W, \varepsilon_{\tilde{Y}}). \tag{5.3}$$

Each equation captures a conditional distribution of the term on the left side, conditioned on terms on the right side (excluding the exogenous variable). Note that the last equation is exactly representing the transition relationship $P(\tilde{Y}|A, Z, Y, W)$ we want to identify.

5.3.1 Statistically Fair Classification with Instance-dependent Label Noise

Almost all statistically fair classification problems can be formulated by a constrained optimization problem. Generally, we minimize the classification error $L(\cdot)$ on training data subject to a specific statistical fairness constraint Fair$(\cdot)$:

$$\text{minimize} \quad \sum_{n=1}^{N} L(f(x_n), y_n) \qquad \text{subject to Fair}(X, Y, f) = 0. \tag{5.4}$$

The clean optimization problem can be statistically linked to the noisy optimization problem with the transition relationship. Next, we propose two general methods and for illustration, we specialize them for two representative fairness notions: equalized odds and p-Fair, respectively. Methods designed for them can be easily extended to equal opportunity and demographic parity [38, 28, 112].

Equalized Odds [38]. The definition of equalized odds states that "A predictor $\hat{Y}$ satisfies equalized odds with respect to protected attribute A and outcome Y, if $\hat{Y}$ and A are independent conditional on Y: $P(\hat{Y} = 1|A = 0, Y) = P(\hat{Y} = 1|A = 1, Y), y \in 0, 1$".

For the classification error part, we show how to use the importance reweighting technique [15, 64] to consistently estimate it:

$$\begin{aligned}
&\mathbb{E}_{(X,Y)\sim\mathcal{D}}[L(f(X), Y)] \\
&= \int P_{\mathcal{D}}(X, Y) L(f(X), Y) \mathrm{dX}\ \mathrm{dY} \\
&= \int P_{\mathcal{D}_\rho}(X, Y) \frac{P_{\mathcal{D}}(X, Y)}{P_{\mathcal{D}_\rho}(X, Y)} L(f(X), Y) \mathrm{dX}\ \mathrm{dY} \\
&= \mathbb{E}_{(X,Y)\sim\mathcal{D}_\rho} \left[\frac{P_{\mathcal{D}}(X, Y)}{P_{\mathcal{D}_\rho}(X, Y)} L(f(X), Y) \right] \\
&= \mathbb{E}_{(X,Y)\sim\mathcal{D}_\rho}[\beta(X, Y) L(f(X), Y)],
\end{aligned} \tag{5.5}$$

where $\beta(x,y) = \frac{P_{\mathcal{D}}(X=x,Y=y)}{P_{\mathcal{D}_\rho}(X=x,Y=y)}$. To calculate $\beta(x,y)$, we only need noisy data and the noise rate. Let

$$T_{-1}(x) = P(\tilde{Y} = +1 \mid Y = -1, X = x),\ \ T_{+1}(x) = P(\tilde{Y} = -1 \mid Y = +1, X = x),$$

then we have

$$P(\tilde{Y} = y \mid X = x) = (1 - T_{-1}(x) - T_{+1}(x))\, P(Y = y \mid X = x) + T_{-y}(x)$$

and

$$\beta(x,y) = \frac{P(\tilde{Y} = y \mid X = x) - T_{-y}(x)}{(1 - T_{-1}(x) - T_{+1}(x))\, P(\tilde{Y} = y \mid X = x)}. \tag{5.6}$$

For the equalized odds constraint part, the original one is $|\gamma_0\left(\hat{Y}\right) - \gamma_1\left(\hat{Y}\right)| = 0$, where $\gamma_a(\hat{Y}) \triangleq \{P(\hat{Y} = 1 \mid A = a, Y = 1), P(\hat{Y} = 1 \mid A = a, Y = 0)\}$. Now we rewrite the first term of $\gamma_0\left(\hat{Y}\right)$:

$$\begin{aligned}
&P(\hat{Y} = 1 \mid A = a, Y = 1) \\
&= \frac{P(\hat{Y} = 1, A = a, Y = 1)}{P(A = a, Y = 1)} \\
&= \frac{P(\hat{Y} = 1, A = a, Y = 1) P(A = a, \hat{Y} = 1)}{P(A = a, Y = 1) P(A = a, \hat{Y} = 1)} \\
&= \frac{P(Y = 1 \mid A = a, \hat{Y} = 1) P(A = a, \hat{Y} = 1)}{P(A = a, Y = 1)} \\
&= \frac{\left(P(\tilde{Y} = 1 \mid A = a, \hat{Y} = 1) - T_{-1}(a)\right) P(A = a, \hat{Y} = 1)\, (1 - T_{-1}(a) - T_{+1}(a))}{(1 - T_{-1}(a) - T_{+1}(a)) \left(P(\tilde{Y} = 1, A = a) - T_{-1}(a)\right)} \\
&= \frac{\left(P(\tilde{Y} = 1 \mid A = a, \hat{Y} = 1) - T_{-1}(a)\right) P(A = a, \hat{Y} = 1)}{P(\tilde{Y} = 1, A = a) - T_{-1}(a)},
\end{aligned} \tag{5.7}$$

where all variables are accessible, either observable or learnable, and

$$T_{-1}(a) = P(\tilde{Y} = +1 \mid Y = -1, A = a),\ \ T_{+1}(a) = P(\tilde{Y} = -1 \mid Y = +1, A = a).$$

Note that this group transition relation $T_y(a)$ can be derived from the

individual one $T_y(x)$. The detailed derivation process is provided in Appendix C.1.

The rest three terms can be rewritten in a similar way. At this point, we can use noisy data to learn a robust classifier with equalized odds.

p-Fair [136]. For the classification error part, we show how to employ a transition matrix to learn a consistent classifier [88]. Let $f(X)$ output the posterior of $Y \in \{-1, 1\}$, i.e., $f(X) = P(Y \mid X)$, then $P(\tilde{Y} \mid X) = T^\top(X)f(X)$. Therefore, by minimizing $L(T^\top(X)f(X), \tilde{Y})$, the learned f is consistent with the one learned on clean data.

For the p-Fair fairness constraint, the original one is Demographic Parity $|P(\hat{Y}|A = 0) - P(\hat{Y}|A = 1)| = 0$. In practice, they use a soft one $|\frac{1}{N}\sum_{i=1}^{N}(a_i - \overline{a})\, f(x)| \leq c$, where c is a threshold. Note that the consistent classifier f is for clean data, which means we can directly substitute it to the constraint. We name this modified method *Robust-p-Fair* (R-p-Fair).

In practical implementation, we employ the Lagrange multipliers method [11] to transfer a constraint to a regularization term. If the instance-dependent transition matrix is not given, we can approximate it for one instance by a combination of the transition matrices for the parts of the instance. Estimating the transition matrix will be much easier if a small clean set is given [125, 124].

5.3.2 Counterfactually Fair Classification with Instance-dependent Label Noise

In this section, we consider the causality-based fairness notion. We elaborate on how to make full use of the causal graph to design a robust and counterfactually fair classifier. Then we showcase how to implement the algorithm in practice.

Counterfactual Fairness

Intuitively, counterfactual fairness requires that changing A, while holding things that are not causally dependent on A constant, will not change the

distribution of the predictor h:

Definition 5.3.1. *[56] Predictor h is counterfactually fair if under any context $X = x$ and $A = a$,*

$$P\left(h_{A\leftarrow a}(U) = y \mid X = x, A = a\right) = P\left(h_{A\leftarrow a'}(U) = y \mid X = x, A = a\right)$$

for all y and for any value a a' attainable by A.

One straightforward strategy to achieve counterfactual fairness is the following:

Lemma 5.3.1. *[56] Let $\mathcal{G}$ be the causal graph of the given model. Then classifier will be counterfactually fair if it is a function of the non-descendants of A.*

A major concern of this strategy is that it totally discards W and loses much information. W inherently contains information from its parents A and U, and we extract the U-part information in W by reconstructing W with U.

How the Causal Graph Interprets the Data

Here we take the benchmark dataset *ADULT* [27] as an example to demonstrate how this causal graph (Figure 5.2) interprets the data:

- A represents the protected attribute 'Gender'.
- Z represents the other root-level attributes, i.e., 'Age', 'Race', and 'Native country', which are not affected by the protected attribute A.
- W represents the low-level attributes, e.g., 'Workclass', 'Capital-gain', and 'Marital-status', which are caused by the background variable U and root-level attributes A and Z: $U \rightarrow W \leftarrow (A, Z)$.
- U is a latent variable and can be seen as 'Background' of people, which causes those non-protected attributes, making U a confounder of W and Z: $W \leftarrow U \rightarrow Z$.
- Y represents the clean but latent label, the annual income, which is influenced by the background and root-level attributes of people:

$U, Z \rightarrow Y$. Note that, to fulfill counterfactual fairness, we intentionally block the path from A to Y. Meanwhile, annual income (not the salary of a job) acts as a cause of the low-level attributes: $Y \rightarrow W$. For example, people with lower annual income are less willing to do 'Without-pay' work, which is one kind of 'Workclass'. For another example, people with higher annual income pay more attention to investment and wealth management and thereby have a larger 'Capital-gain'. Besides, annual income can obviously affect 'Marital-status'.

- $\tilde{Y}$ represents the noisy label, which is a common child of the observable variables and clean label: $(A, Z, W, Y) \rightarrow \tilde{Y}$.

Based on this causal graph, we can only feed U and Z to the classifier f to infer the clean label Y, which, according to the Lemma 5.3.1, makes f counterfactually fair. Specifically, we employ the variational autoencoder (VAE) framework [52] to make full use of the causal graph which can infer latent variables U and Y by maximizing the joint likelihood of observable variables. Moreover, exploiting the causal graph contributes to the identifiability of the transition relationship between clean and noisy labels [131].

VAE based Causal Inference

The joint distribution $p(U, A, Z, W, \tilde{Y}, Y)$ specified by the causal graph in Figure 5.2 and the structural causal model Eq. (5.3) can be factorized as follows:

$$\begin{aligned} &p(U, A, Z, W, \tilde{Y}, Y) \\ &= p(A)p(U)p(Z \mid U, A)p(Y \mid U, A, Z)p(W \mid U, A, Z, Y)p(\tilde{Y} \mid U, A, Z, Y, W) \\ &= p(A)p(U)p(Z \mid U)p(Y \mid U, Z)p(W \mid U, A, Z, Y)p(\tilde{Y} \mid A, Z, Y, W). \end{aligned} \tag{5.8}$$

Note that although A is involved in the reconstruction of W and $\tilde{Y}$, it does not causally affect how U and Z infer Y. Namely, the counterfactual fairness still holds.

In the encoding phase, we infer the latent variable U and Y from observable variables Z. Without loss of generality, we choose prior $p(U)$ to be simple, i.e., Gaussian. We use an encoder with a learnable parameter ϕ

to model the distribution $p(U, Y \mid A, Z, W, \tilde{Y})$. Since A and its descendant W are not allowed to build the classifier, and given its all parents U and Z, Y is independent on $(A, W, \tilde{Y})$, the encoder can be simplified as:

$$\begin{aligned} q_\phi(U, Y \mid A, Z, W, \tilde{Y}) &= q_{\phi_1}(U \mid A, Z, W, \tilde{Y}) q_{\phi_2}(Y \mid U, A, Z, W, \tilde{Y}) \\ &= q_{\phi_1}(U \mid Z) q_{\phi_2}(Y \mid U, Z), \end{aligned} \tag{5.9}$$

where $q_{\phi_2}(Y \mid U, Z)$ can be employed as a counterfactually fair classifier f.

In the decoding phase, given that $p(U)$ is Gaussian, we need four decoders corresponding to the rest four terms on the right side of Eq. (5.8), as:

$$\begin{aligned} &p_\theta(U, A, Z, W, \tilde{Y}, Y) = \\ &p(A)p(U)p_{\theta_1}(Z \mid U)p_{\theta_2}(Y \mid U, Z)p_{\theta_3}(W \mid U, A, Z, Y)p_{\theta_4}(\tilde{Y} \mid A, Z, Y, W). \end{aligned} \tag{5.10}$$

We denote $\Theta = \{\phi_1, \phi_2, \theta_1, \theta_2, \theta_3, \theta_4\}$ the parameter set of this VAE network. In the evaluation phase, we first sample U from $q_{\phi_1}(U \mid Z)$ and then use (U, Z) to infer Y with $q_{\phi_2}(Y \mid U, Z)$. Note that ϕ_2 and θ_2 are the same, which both model the generation process of Y. It is because Y is a latent intermediate variable such that modeling Y can be treated as either encoding or decoding. Hereinafter we refer to them collectively using classifier f.

Then, because the data likelihood $p_\Theta(A, Z, W, \tilde{Y})$ is intractable, instead of maximizing the data likelihood, we learn Θ by minimizing the negative evidence lower bound (ELBO) [52]. ELBO is a lower bound of the likelihood, which is preferred for optimization because it can be calculated efficiently.

Starting with maximizing the data likelihood $p_\Theta(A, Z, W, \tilde{Y})$, we can derive the negative ELBO as follows (the detailed derivation process is provided in Appendix C.2):

$$-\text{ELBO} \triangleq -\mathrm{E}_{(u,y)\sim q_\phi(u,y|z)}\left[\log p_{\theta_1}(z \mid u)\right] - \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)}\left[\log p_{\theta_3}(w \mid u, a, z, y)\right] \tag{5.11}$$

$$- \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)}\left[\log p_{\theta_4}(\tilde{y} \mid a, z, y, w)\right] + \mathrm{D}_{\mathrm{KL}}(q_{\phi_1}(u \mid z) \| p(u)), \tag{5.12}$$

where D_{KL} is the Kullback–Leibler divergence function. Although the above ELBO does not explicitly involve the counterfactually fair classifier f, the prediction Y plays an important role in the second and third terms of ELBO, which pushes f to be optimized.

So far, the classifier outputs a counterfactually fair prediction Y, which can be treated as cluster numbers but not clean class labels. Since Y is a latent intermediate variable, the map between the value of Y ($+1$ or -1) to the semantic class (*positive* or *negative*) is lost. To map Y to semantic clean labels, noisy labels $\tilde{Y}$ are the only thing we have that could help. In case f is severely misled by $\tilde{Y}$, we introduce a data augmentation technique *Mixup* [139], which generates a weighted combination of random instance pairs from the training data:

$$\hat{x} = \lambda x_i + (1-\lambda)x_j, \tag{5.13}$$

$$\hat{y} = \lambda y_i + (1-\lambda)y_j, \tag{5.14}$$

where weights λ are independently sampled from a Beta distribution for each augmented example. Mixup prevents f from overfitting noisy labels in two aspects. First, it increases the complexity of the training data, which makes it difficult for a network to learn. Second, by combining different features (labels) with one another, a network does not get overconfident about the relationship between the features and their labels.

Practical Implementation

The proposed algorithm is summarized in Algorithm 3.

For the negative ELBO part, the first three terms are exactly reconstruction errors [52]. Therefore, in practice, we use mean squared error to

Algorithm 3 Robust Counterfactually Fair Classification (RCFC).

Input: A training sample of observable variables $(A, Z, W, \tilde{Y})$.
Encode U:
$\mu, \sigma = q_{\phi_1}(Z)$ ▷reparameterization trick
$U = \mu + \sigma\epsilon$ ▷where ϵ is an auxiliary noise variable $\epsilon \sim \mathcal{N}(0, 1)$
Encode Y:
$Y = q_{\phi_2}(U, Z)$
Decode (Reconstruct) $Z, W, \tilde{Y}$:
$\hat{Z} = p_{\theta_1}(U)$ ▷where $\hat{Z}$ is the predicted value of Z
$\hat{W} = p_{\theta_3}(U, A, Z, Y)$ ▷where $\hat{W}$ is the predicted value of W
$\tilde{Y}^{\diamond} = p_{\theta_4}(A, Z, Y, W)$ ▷where $\tilde{Y}^{\diamond}$ is the predicted value of $\tilde{Y}$
Update parameter set Θ by minimizing $-$ELBO and the Mixup loss.
Output: Encoder $q_{\phi_1}(Z)$; Classifier f (Encoder $q_{\phi_2}(U, Z)$).

measure the reconstruction errors for $(\hat{Z}, \hat{W})$ with respect to (Z, W), and we use cross-entropy loss to measure the reconstruction errors for $\hat{\tilde{Y}}$ with respect to $\tilde{Y}$. As for the last $\mathrm{D_{KL}}$ term, first we use the reparameterization trick [52] to sample U once from $q_{\phi_1}(u \mid z)$, and μ, σ are continuous variables with gradients. Note that U can also be the average value of several sampling results to decrease the variance. Then, we calculate $\mathrm{D_{KL}}$ term with the closed-form solution provided by [52]:

$$\mathrm{D_{KL}}(q_{\phi_1}(u \mid z) \| p(u)) = -\frac{1}{2} \sum_{j=1}^{J} \left(1 + \log \left(\left(\sigma_j^{(i)} \right)^2 \right) - \left(\mu_j^{(i)} \right)^2 - \left(\sigma_j^{(i)} \right)^2 \right), \tag{5.15}$$

where J is the dimension of U.

For the mixup loss part, we first concatenate U and Z as input I, and then apply the mixup technique to classifier f with pairs $(I, \tilde{Y})$. Here, we use cross-entropy loss.

5.4 Experiments

Dataset. We employ two widely used benchmark datasets:

- *ADULT* [27]. The prediction task is to determine whether a person makes over \$50K a year, with gender as the protected attribute. The

detailed information for this dataset and how it complies with the causal graph have been elaborated in Section 5.3.2.

- *BANK* [27]. The prediction task is to determine whether a client subscribes to a term deposit, with gender as the protected attribute. We select personal attributes except gender and the credit history attributes as the other root-level attribute Z. Those loan-relevant and property-relevant attributes are selected as the low-level attributes W. We drop social and economic context attributes because they are irrelevant.

For all datasets, of which 10% are split as test data. The rest 90% is for training, of which 10% are split as validation data. We use validation data for model selection. The final output model is selected with the highest validation accuracy.

Noisy labels generation. For clean datasets, we artificially corrupt the class labels of training and validation sets following the instance-dependent label noise generalization method in [124]. We generate noisy datasets of $\{0.1, 0.2, 0.3, 0.4\}$ four noise levels.

Network structure and optimization. For a fair comparison, all experiments are conducted on NVIDIA GeForce RTX 2080 Ti, and all methods are implemented by PyTorch. The dimension of background variable U is set to 2. We employ a three-layer MLP with the Softsign activation function for every single model. The batch size is set to 128. We use SGD optimizer with momentum 0.9 and an initial learning rate 0.001. Learning rate is updated by *ReduceLROnPlateau*, which reduces learning rate when a metric (here we choose training loss as the metric) has stopped improving.

Baselines. We compare our methods R-p-Fair and RCFC with six baselines of four types:

- Standard supervised learning (SSL). It takes all the features as input and noisy labels as the target, which is not fair.
- p-Fair [136]. It takes all the features except A as input and noisy labels as the target, which is softly fair with fairness metric p value. We

reimplement this method with Pytorch.

- Ablation-U (Ab-U). It postulates background variable U but does not model the label noise.
- Ablation-N (Ab-N). It models the label noise but does not postulate background variable U.
- Counterfactual fairness learning (CFL) [56] . It only uses non-descendants to make predictions, which is counterfactually fair.
- Counterfactual fairness learning with Mixup (CFL-M). Based on CFL, it additionally applies mixup technique, which is both counterfactually fair and kind of robust to label noise.

Table 5.2: Means and Standard deviations of classification accuracy and fairness score (p value) on ADULT dataset over 5 trials.

ADULT	IDLN-0.1		IDLN-0.2		IDLN-0.3		IDLN-0.4	
	Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness
SSL	71.63±5.16	14.22±5.48	63.68±5.02	21.24±6.87	58.80±4.17	25.90±5.09	51.20±8.13	34.45±6.05
p-Fair	69.46±6.83	30.08±7.19	61.89±4.96	32.85±5.92	58.50±4.42	35.10±5.53	49.85±7.83	40.43±4.97
R-p-Fair	69.97±7.18	41.78±1.02	63.27±3.94	38.35±3.69	59.46±4.67	39.74±5.08	51.42±8.78	41.02±4.47

Table 5.3: Means and Standard deviations of classification accuracy on ADULT dataset over 5 trials.

ADULT	0.1	0.2	0.3	0.4
Ab-U	65.13±1.97	65.16±2.39	64.10±3.98	61.03±9.97
Ab-N	73.07±2.46	73.30±2.08	70.61±2.44	61.41±9.27
CFL	66.17±1.22	67.20±3.24	63.27±5.67	61.11±8.69
CFL-M	69.07±2.86	64.65±3.79	64.42±4.04	63.90±4.29
RCFC	**74.69±0.42**	**74.65±0.42**	**74.30±0.44**	**72.96±1.91**

Results. The results in Table 5.2 and Table 5.4 show that there is a steady lift of our method on the classification accuracy and fairness score, compared with baseline methods. However, all statistical methods suffer from label noise to a great extent. It is because IDLN is ill-defined and handling it with purely statistical relations is not sufficient.

The results in Table 5.3 and Table 5.5 demonstrate that our method achieves distinguished classification accuracy and is counterfactually fair. Compared with counterfactually fair methods CFL-M and CFL, our method

Table 5.4: Means and Standard deviations of classification accuracy and fairness score (p value) on BANK dataset over 5 trials.

BANK	IDLN-0.1		IDLN-0.2		IDLN-0.3		IDLN-0.4	
	Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness	Accuracy	Fairness
SSL	84.54±4.23	35.49±15.27	66.60±12.87	19.49±5.63	58.69±4.92	18.51±5.95	56.31±4.90	17.80±5.64
p-Fair	87.22±1.80	43.08±21.69	67.00±13.00	18.65±5.34	58.62±5.04	18.53±5.69	57.16±5.20	17.90±5.54
R-p-Fair	87.18±2.33	49.57±27.26	76.02±7.08	25.20±7.08	60.43±4.90	18.54±4.88	60.71±6.36	19.32±5.02

Table 5.5: Means and Standard deviations of classification accuracy on BANK dataset over 5 trials.

BANK	0.1	0.2	0.3	0.4
Ab-U	87.19±1.78	85.86±2.19	81.48±4.93	68.43±11.35
Ab-N	87.51±0.71	86.75±1.37	85.52±4.08	76.74±9.81
CFL	84.09±7.23	72.09±14.92	58.29±16.08	55.42±14.11
CFL-M	80.40±7.70	74.09±12.35	64.98±8.41	58.85±8.32
RCFC	**88.77±0.61**	**88.76±0.62**	**88.72±0.61**	**86.40±1.88**

additionally extracts as much as possible knowledge from the data in the precondition of satisfying a fairness requirement. These credits should go to the postulated causal graph which captures the data structure well. The results of ablation studies Ab-U and Ab-N show that exploiting causal relations and modeling label noise are both significant.

As the noise rate increases, the accuracy of all baselines decreases significantly while there is just a slight drop for our method. Even for challenging noise rates of 0.4, our method achieves good accuracy, uplifting about 15 and 30 points on ADULT and BANK, respectively. CFL-M and CFL have similar performances on both datasets, which means the mixup technique itself does not handle the instance-dependent label noise effectively. It also reflects the improvements of our method are mainly benefited from the proposed causal model which contributes to the identifiability of the transition relationship between clean and noisy labels.

5.5 Summary

This chapter proposed general frameworks for learning fair classifiers with instance-dependent label noise. We notice that label noise not only degenerates the classification accuracy but misleads the fairness-aware algorithms even more prejudiced than fairness-unaware ones. We adapt statistically fair methods to the label noise setting and build consistent classifiers. Then we postulate a general causal graph, which can interpret the real-world datasets well. By exploiting the causal graph, we design an algorithm that both strictly achieves counterfactual fairness and identifies the transition relationship between clean and noisy labels. Experiments conducted on benchmark datasets demonstrate the effectiveness of our method.

Chapter 6

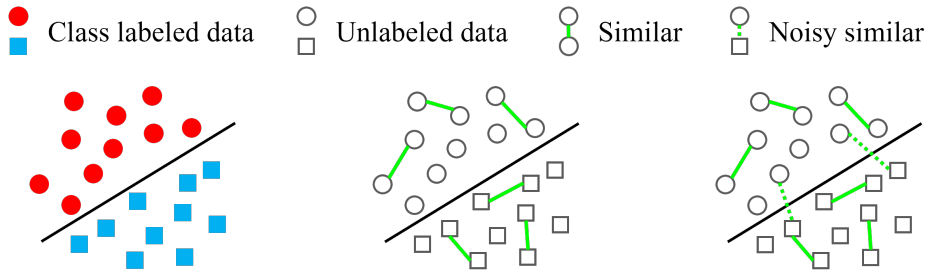
LNL incorporating Privacy Concerns

SU classification employs similar (S) data pairs (two examples belong to the same class) and unlabeled (U) data points to build a classifier, which can serve as an alternative to the standard supervised trained classifiers requiring data points with class labels. SU classification is advantageous because in the era of big data, more attention has been paid to data privacy. Datasets with specific class labels are often difficult to obtain in real-world classification applications regarding privacy-sensitive matters, such as politics and religion, which can be a bottleneck in supervised classification. Fortunately, similarity labels do not reveal the explicit information and inherently protect the privacy, e.g., collecting answers to "With whom do you share the same opinion on issue $\mathcal{I}$?" instead of "What is your opinion on issue $\mathcal{I}$?". Nevertheless, SU classification still has an obvious limitation: respondents might answer these questions in a manner that is viewed favorably by others instead of answering truthfully. Therefore, there exist some dissimilar data pairs labeled as similar, which significantly degenerates the performance of SU classification. In this chapter, we study how to learn from noisy similar (nS) data pairs and unlabeled (U) data, which is called *nSU classification*. Specifically, we model the similarity noise and estimate the noise rate by using the *mixture proportion estimation* technique. Then, a clean classifier can be learned by minimizing a denoised and unbiased classification risk estimator. Moreover, we further derive a theoretical generalization error bound for the proposed method.

6.1 Motivations and Contributions

When it comes to privacy-sensitive matters, such as politics and religion, people often hesitate to directly answer questions like "What is your opinion on issue $\mathcal{I}$?" and prefer to answer questions like "With whom do you share the same opinion on issue $\mathcal{I}$?". Questions in this form can be regarded as one type of randomized response technique, which is a commonly used indirect questioning survey method that reduces the social desirability bias and increases data reliability [114, 29, 5]. To some degree, similarity information avoids embarrassment and protects personal privacy. Nevertheless, in practice, respondents might answer these questions in a manner that is viewed favorably by others instead of answering truthfully [86]. As a result, the collected similarity data not only contain similar but also dissimilar data pairs. This kind of noise is random noise rather than adversarial noise because adversarial noise is often designed by considering the properties of algorithms so that it can confuse the algorithm [105, 70]; however, the noise induced by humans is independent of machine learning algorithms. In this case, if we directly employ the existing algorithms for clean similarity learning to deal with noisy similarity supervision, the classification performance will inevitably degenerate because the model will overfit the noisy data [138]. For example, if we directly employ the estimator designed for SU classification [5], an estimation bias would be introduced, and the learned classifier would thereby no longer be optimal.

In this chapter, we study the problem of how to learn a robust consistent classifier from noisy similar data pairs and unlabeled (nSU) data, which is called *nSU classification*. As shown in Figure 6.1, there is no class supervision in SU or nSU classification. In addition, there are some wrong links of dissimilar data pairs in nSU classification, which makes the problem more difficult. To this end, we propose an empirical risk minimization (ERM) [110] framework to learn classifiers from only nSU data. Although the unbiased classification risk estimator for SU data has been studied, how to properly model the noise is a significant bottleneck for solving the nSU problem. There are two widely-used noise models, i.e., the class-conditional label noise (CCN) model [2] and mutually contaminated distributions (MCD) model [98]. We employ MCD to model the noise: the



(a) Supervised Classification (b) SU Classification (c) nSU Classification

Figure 6.1: The illustration of supervised classification, SU classification, and nSU classification. SU classification learns from similar data pairs and unlabeled data while nSU classification learns from noisy similar data pairs and unlabeled data.

distribution of noisy similar data pairs is a mixture proportion of the similar and dissimilar data pairs, where the noise rate is defined as the proportion of dissimilar data pairs in noisy similar data pairs. We choose MCD because CCN is a noise model for labeling noise and MCD is a noise model for sampling noise, which is why MCD is more suitable for the scenario of surveying sensitive topics with indirect questioning (more discussion can be found in Appendix D.1). To estimate the noise rate, we decompose the nSU data distributions and convert them to a standard mixture proportion estimation (MPE)[1] format. We prove that our MPE problem is well defined such that the noise rate is identifiable and can be consistently estimated using MPE methods. Then we theoretically build an unbiased estimator for the classification risk with respect to the fully and accurately labeled data.

6.2 Related Work

There are a few works regarding the label noise issue on similarity learning. However, the employed noise model, the essential parameter estimation method, and the classifier models of our work are all different from the related works.

[1]Let F, G, and H be distributions on $(\mathcal{X}, \mathfrak{S})$ such that $F = (1-\kappa)G + \kappa H$, where $0 \leq \kappa \leq 1$. MPE is to estimate κ, given i.i.d. samples from both F and H [13].

First, [23] studied a similar but different problem of how to learn a binary classifier from noisy similar data pairs and dissimilar data pairs. The noise model [23] employed is CCN, where the clean labels are assumed to flip into other classes with a certain probability. Specifically, in [23], similar data pairs are corrupted into dissimilar data pairs with probability α, and dissimilar data pairs are corrupted into similar data pairs with probability β:

$$\begin{bmatrix} p(\bar{S}=0|\boldsymbol{x},\boldsymbol{x}') \\ p(\bar{S}=1|\boldsymbol{x},\boldsymbol{x}') \end{bmatrix} = \begin{bmatrix} 1-\beta & \alpha \\ \beta & 1-\alpha \end{bmatrix} \begin{bmatrix} p(S=0|\boldsymbol{x},\boldsymbol{x}') \\ p(S=1|\boldsymbol{x},\boldsymbol{x}') \end{bmatrix}, \tag{6.1}$$

where S and $\bar{S}$ denote the similarity label and noisy similarity label[2].

This model is different from the MCD model, where the noise distribution is a mixture proportion of the clean distributions. Specifically, for the MCD model, in $\tilde{p}_{\mathrm{s}}$ ($\tilde{p}_{\mathrm{d}}$), a proportion ρ_{d} (ρ_{s}) of data pairs are contaminated by dissimilar (similar) data pairs, while the remaining $1-\rho_{\mathrm{d}}$ ($1-\rho_{\mathrm{s}}$) proportion remains similar (dissimilar):

$$\begin{bmatrix} p(\boldsymbol{x},\boldsymbol{x}'|\bar{S}=0) \\ p(\boldsymbol{x},\boldsymbol{x}'|\bar{S}=1) \end{bmatrix} = \begin{bmatrix} 1-\rho_{\mathrm{s}} & \rho_{\mathrm{s}} \\ \rho_{\mathrm{d}} & 1-\rho_{\mathrm{d}} \end{bmatrix} \begin{bmatrix} p(\boldsymbol{x},\boldsymbol{x}'|S=0) \\ p(\boldsymbol{x},\boldsymbol{x}'|S=1) \end{bmatrix}. \tag{6.2}$$

It is notable that the middle matrix in Eq. (6.1) is column normalized while the middle matrix in Eq. (6.2) is row normalized. Moreover, the CCN model is a strict special case of the MCD model [76]. It has been studied in [66] that $p(\bar{S})$ is fixed in the CCN model once $p(\bar{S}|\boldsymbol{x},\boldsymbol{x}')$ is specified while $p(\bar{S})$ is free in the MCD model after $p(\boldsymbol{x},\boldsymbol{x}'|\bar{S})$ is specified. Furthermore, for $\tilde{p}(\boldsymbol{x})$ being the distribution of noisy $\boldsymbol{x}$, $\tilde{p}(\boldsymbol{x}) = p(\boldsymbol{x})$ holds in the CCN model but $\tilde{p}(\boldsymbol{x}) \neq p(\boldsymbol{x})$ holds in the MCD model. Due to this covariate shift [104], CCN methods do not fit the MCD problem setting, while the MCD methods fit the CCN problem setting conversely.

For our nSU contamination model, we only have the noisy similar data pairs, and Eq. (6.9) is consistent with part of Eq. (6.2). Besides, there is no restriction on the noisy dissimilar data pairs. Namely, we can

[2][23] called this noise model *pairing corruption*. They also discussed another noise model called *labeling corruption*, where labels Y and Y' are corrupted in an instance-independent manner.

set the imaginary noisy dissimilar data pairs to obey the distribution in Eq. (6.2). Moreover, the distribution of the unlabeled data is free to label noise. Therefore, the MCD model, as well as the CCN model, is a special case of the nSU contamination model.

Second, to build an unbiased classification risk framework from the observation, there are some essential parameters to estimate. [23] roughly tuned the noise rate parameter by cross-validation on the noisy data. Then, based on the tuned parameter, the prior was empirically estimated. [5] used an MPE method to solve the estimation problem. However, the solution of that MPE problem is not guaranteed to be identifiable. By contrast, we prove that the new MPE problem in this work is *irreducible* [97], and thereby can be solved with a theoretical guarantee (see Section 6.3.4 for details).

Third, [23] used a neural network to approximately learn the classifier. However, we use not only the neural network but also the linear model, of which there exist analytical solutions to the latter model.

6.3 Methodology

We consider the binary classification problem. Let $\mathcal{D}$ be the distribution of a pair of random variables $(x, y) \in \mathcal{X} \times \{-1, 1\}$, where $\mathcal{X} \subset \mathbb{R}^d$ and d represents the dimension. Let $f : \mathbb{R}^d \to \mathbb{R}$ be a *hypothesized classifier* in the predefined *hypothesis class* $\mathcal{F}$, and $\ell : \mathbb{R} \times \{\pm 1\} \to \mathbb{R}$ be the *loss function* measuring how well the true class label Y is estimated by the prediction of a hypothesis. The *optimal classifier* f^* is therefore defined by the hypothesis that minimizes the *expected classification risk*:

$$R(f) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\ell(f(x), y)], \tag{6.3}$$

i.e.,

$$f^* = \arg\min_{f\in\mathcal{F}} R(f). \tag{6.4}$$

Often, the distribution of data is unknown. The standard supervised binary classification method utilizes positive and negative training data drawn

i.i.d. from $\mathcal{D}$ to learn the optimal classifier by minimizing the empirical classification risk, which is an approximation of the expected risk in Eq. (6.3). While in our setting, we only have noisy similar (nS) data pairs, i.e., $\{(\boldsymbol{x}_{\mathrm{S},1}, \boldsymbol{x}'_{\mathrm{S},1}), \ldots, (\boldsymbol{x}_{\mathrm{S},n_{\mathrm{nS}}}, \boldsymbol{x}'_{\mathrm{S},n_{\mathrm{nS}}})\}$, and some unlabeled (U) data, i.e., $\{\boldsymbol{x}_{\mathrm{U},1}, \ldots, \boldsymbol{x}_{\mathrm{U},n_{\mathrm{U}}}\}$, which are called the nSU data. A pair of instances is said to be similar if they are from the same class. Noisy similar data pairs mean that the data may come from different classes but are treated as similar data pairs. Therefore, our research problem in this chapter is to build an empirical classification risk by only employing the nSU data that will approximate the expected risk in Eq. (6.3).

6.3.1 Noisy Pairwise Similarity and Unlabeled Data

Below, we provide detailed definitions and assumptions in the nSU classification.

Assumption 6.3.1 ([5]). *Data independence. Without any assumptions, the pairwise data cannot effectively be used to approximate the risk. We assume that each instance is independently drawn from joint distribution $\mathcal{D}$. For example, for every data pair $(\boldsymbol{x}, \boldsymbol{x}')$, if two instances are similar, they follow the probability density as*

$$\begin{aligned} p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') &= p(\boldsymbol{x}, \boldsymbol{x}' | y = y' = +1 \vee y = y' = -1) \\ &= \frac{\pi_+^2 p_+(\boldsymbol{x}) p_+(\boldsymbol{x}') + \pi_-^2 p_-(\boldsymbol{x}) p_-(\boldsymbol{x}')}{\pi_+^2 + \pi_-^2}, \end{aligned} \tag{6.5}$$

where $p(A \vee B)$ represents the probability density that either A or B occurs, and

- *$\pi_+ = p(y = +1)$ and $\pi_- = p(y = -1)$ are the class-prior probabilities, which satisfy $\pi_+ + \pi_- = 1$,*
- *$p_+(\boldsymbol{x}) = p(\boldsymbol{x}|y = +1)$ and $p_-(\boldsymbol{x}) = p(\boldsymbol{x}|y = -1)$ are the class-condition probability density.*

More discussion about this assumption can be found in Appendix D.2.

Eq. (6.5) shows that two instances are drawn independently following $p(\boldsymbol{x}, y)$, which corresponds to the density of $\mathcal{D}$, and they belong to the

similar data pairs if they have the same label. In contrast, if the two instances of a data pair belong to the different classes, they follow the probability density as

$$\begin{aligned} p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}') &= p(\boldsymbol{x}, \boldsymbol{x}' | (y = +1 \wedge y' = -1) \vee (y = -1 \wedge y' = +1)) \\ &= \frac{\pi_+ \pi_- p_+(\boldsymbol{x}) p_-(\boldsymbol{x}') + \pi_+ \pi_- p_-(\boldsymbol{x}) p_+(\boldsymbol{x}')}{2\pi_+ \pi_-} \\ &= \frac{p_+(\boldsymbol{x}) p_-(\boldsymbol{x}') + p_-(\boldsymbol{x}) p_+(\boldsymbol{x}')}{2}, \end{aligned} \tag{6.6}$$

where $p(A \wedge B)$ represents the probability density that both A and B occur.

For the unlabeled data, we assume that they are independently drawn from the marginal density $p(\boldsymbol{x})$, which can be decomposed as

$$p(\boldsymbol{x}) = \pi_+ p_+(\boldsymbol{x}) + \pi_- p_-(\boldsymbol{x}). \tag{6.7}$$

Lemma 6.3.1. *Assume that Assumption I holds. If two instances $\boldsymbol{x}$ and $\boldsymbol{x}'$ are drawn independently from the unlabeled data density, Eq. (6.7) can easily convert to a pairwise marginal version such that*

$$p(\boldsymbol{x}, \boldsymbol{x}') = \pi_{\mathrm{s}} p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') + \pi_{\mathrm{d}} p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}'), \tag{6.8}$$

where $\pi_{\mathrm{s}} = \pi_+^2 + \pi_-^2$ and $\pi_{\mathrm{d}} = 2\pi_+ \pi_-$.

A detailed proof is provided in Appendix D.3.

Assumption 6.3.2. *Contamination model. To handle the noise, we consider the contamination model proposed by [47], which has been widely used in the label noise learning community [76, 98]. Specifically, the noisy similarity data consist of both similar data pairs and dissimilar data pairs:*

$$\tilde{p}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') = (1 - \rho_{\mathrm{d}}) p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') + \rho_{\mathrm{d}} p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}'), \tag{6.9}$$

where $\rho_{\mathrm{d}} \in [0, 1)$ is regarded as the noise rate and $\tilde{p}_{\mathrm{s}}$ denotes the density of noisy similar data pairs. More specifically, in $\tilde{p}_{\mathrm{s}}$, a proportion ρ_{d} of data pairs are contaminated by dissimilar data pairs, while the $(1-\rho_{\mathrm{d}})$ proportion remains similar.

We employ the contamination model rather than the CCN model because the former is more suitable to describe the noise pattern of nSU data. We take the "opinion-on-issue-$\mathcal{I}$" case as an example: In practice, the data generation procedure is to collect answers to the question "With whom do you share the same opinion on issue $\mathcal{I}$?". In statistics, it is to sample examples of similar data pairs from $p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}')$. However, some people give wrong answers, which makes the selected examples contain dissimilar data pairs from $p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}')$. Overall, the data generation procedure is an imbalanced sample from both $p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}')$ and $p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}')$, which can be exactly formulated by a contamination model in Eq. (6.9).

The above two assumptions overlook the data-dependence in pairwise data (i.e., pairwise data are independently sampled from $p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}')$) and the instance-dependence of noise (i.e., $(1-\rho_{\mathrm{d}})$ is independent of $\boldsymbol{x}$ and $\boldsymbol{x}'$). However, this simplification has been widely accepted in statistical learning theory and the label-noise learning communities, and the empirical results on benchmark datasets verify the efficiency of the assumptions [88, 125]. We could then build a denoised and unbiased estimator for the classification risk with respect to the latent clean data with the nSU data and provide a theoretical error bound for the proposed method.

We denote the noisy similar data pairs and the unlabeled data by $\tilde{D}_{\mathrm{s}}$ and D_{u}, respectively.

$$\tilde{D}_{\mathrm{s}} \triangleq \{(\boldsymbol{x}_{\mathrm{S},1}, \boldsymbol{x}'_{\mathrm{S},1}), \ldots, (\boldsymbol{x}_{\mathrm{S},n_{\mathrm{nS}}}, \boldsymbol{x}'_{\mathrm{S},n_{\mathrm{nS}}})\} \overset{\text{i.i.d.}}{\sim} \tilde{p}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}'), \tag{6.10}$$

$$D_{\mathrm{u}} \triangleq \{\boldsymbol{x}_{\mathrm{U},1}, \ldots, \boldsymbol{x}_{\mathrm{U},n_{\mathrm{U}}}\} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}), \tag{6.11}$$

where n_{nS} is the size of $\tilde{D}_{\mathrm{s}}$ and n_{U} is the size of D_{u}. We show that a consistent classifier could be learned with a theoretical guarantee.

6.3.2 Risk Expression with nSU Data

Theorem 6.3.1. *Assume that $\pi_+ \neq 0.5$. The classification risk in Eq. (6.3) can be equivalently expressed only in terms of nSU data as follows:*

$$R_{\mathrm{nSU},\ell}(f) = \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{x}')\sim\tilde{p}_{\mathrm{s}}}[\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x}, \boldsymbol{x}')] + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}[\mathcal{L}_{\mathrm{U}}(\boldsymbol{x})], \tag{6.12}$$

where

$$\begin{aligned}
\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x}, \boldsymbol{x}') &= \frac{\pi_{\mathrm{s}}(1-\pi_{\mathrm{s}})}{2(1-\rho_{\mathrm{d}}-\pi_{\mathrm{s}})(2\pi_{+}-1)}[\tilde{l}(\boldsymbol{x})+\tilde{l}(\boldsymbol{x}')], \\
\tilde{l}(\boldsymbol{x}) &= \ell(f(\boldsymbol{x}),+1)-\ell(f(\boldsymbol{x}),-1), \\
\mathcal{L}_{\mathrm{U}}(\boldsymbol{x}) &= \frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}}-\pi_{-}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}\ell(f(\boldsymbol{x}),+1) \\
&\quad - \frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}}-\pi_{+}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}\ell(f(\boldsymbol{x}),-1).
\end{aligned}$$

A detailed proof and discussion are provided in Appendix D.4.

Theorem 6.3.1 immediately leads to an unbiased risk estimator:

$$\hat{R}_{\mathrm{nSU},\ell}(f) = \frac{1}{n_{\mathrm{nS}}}\sum_{i=1}^{n_{\mathrm{nS}}}\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x}_{\mathrm{S},i}, \boldsymbol{x}'_{\mathrm{S},i}) + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\mathcal{L}_{\mathrm{U}}(\boldsymbol{x}_{\mathrm{U},i}). \tag{6.13}$$

6.3.3 Practical Implementation

Here, we employ the linear-in-parameter-model $f(x) = \boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x})$, where $\boldsymbol{w}$ and $\boldsymbol{\phi}$ are vectors of parameters and basis functions with the same dimension. Then employing Eq. (6.13) with the ℓ_2 regularization, the nSU classification can be formulated as the following regularized empirical risk minimization problem:

$$\hat{\boldsymbol{w}} = \min_{\boldsymbol{w}} \hat{J}_{\ell}(\boldsymbol{w}), \tag{6.14}$$

where

$$\begin{aligned}
\hat{J}_{\ell}(\boldsymbol{w}) &= \frac{1}{n_{\mathrm{nS}}}\sum_{i=1}^{n_{\mathrm{nS}}}\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x}_{\mathrm{S},i}, \boldsymbol{x}'_{\mathrm{S},i}) + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\mathcal{L}_{\mathrm{U}}(\boldsymbol{x}_{\mathrm{U},i}) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\
&= \frac{A}{2n_{\mathrm{nS}}}\sum_{i=1}^{2n_{\mathrm{nS}}}[\ell(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}),+1)-\ell(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}),-1)] \\
&\quad + \frac{B}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}[\ell(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),+1)] - \frac{C}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}[\ell(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),-1)] + \frac{\lambda}{2}\|\boldsymbol{w}\|^2,
\end{aligned} \tag{6.15}$$

and

$$A = \frac{\pi_{\mathrm{s}}(1-\pi_{\mathrm{s}})}{(1-\rho_{\mathrm{d}}-\pi_{\mathrm{s}})(2\pi_{+}-1)}, \tag{6.16}$$

$$B = \frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}} - \pi_{-}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}, \tag{6.17}$$

$$C = \frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}} - \pi_{+}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}, \tag{6.18}$$

and λ (≥ 0) in Eq. (6.15) is the regularization parameter. Note that since the loss form is symmetric to $\boldsymbol{x}_{\mathrm{S},i}$ and $\boldsymbol{x}'_{\mathrm{S},i}$, we use $\boldsymbol{x}_{\mathrm{S},i}$ uniformly in Eq. (6.15). To solve this optimization problem, we need the knowledge of class-prior π_{+} (π_{s} can be calculated from π_{+}) and the noise rate ρ_{d}. In Section 6.3.4, we discuss how to estimate them from nSU data.

Inspired by [5], [81], and [92], we surprisingly find that adopting certain loss functions, i.e., the margin loss function[3] [79], will result in a convex objective function.

Theorem 6.3.2. *Assume that the loss function $\ell(z,t)$ is a convex margin loss function, and for every fixed $t \in \{\pm 1\}$, $\ell(z,t)$ is twice differentiable with respect to z. If $\ell(z,t)$ satisfies the condition*

$$\ell(z,+1) - \ell(z,-1) = -z,$$

then $\hat{J}_{\ell}(\boldsymbol{w})$ is convex.

A detailed proof and more discussion are provided in Appendix D.5.

Below, we consider the squared loss function, which satisfies the conditions in Theorem 6.3.2.

The squared loss function is defined as $\ell_{\mathrm{SQ}}(z,t) = \frac{1}{4}(tz-1)^2$. Substituting ℓ_{SQ} into Eq. (6.15), we have

$$\hat{J}_{\mathrm{SQ}}(\boldsymbol{w}) = \boldsymbol{w}^{\top}\left(\frac{1}{4n_{\mathrm{U}}}X_{\mathrm{U}}^{\top}X_{\mathrm{U}} + \frac{\lambda}{2}I\right)\boldsymbol{w} - \left(\frac{A}{2n_{\mathrm{nS}}}\mathbf{1}^{\top}X_{\mathrm{S}} + \frac{B+C}{2n_{\mathrm{U}}}\mathbf{1}^{\top}X_{\mathrm{U}}\right)\boldsymbol{w},$$

[3] ℓ is said to be a margin loss function if there exists $\psi : \mathbb{R} \to \mathbb{R}_{+}$ such that $\ell(z,t) = \psi(tz)$.

where I is the identity matrix, $\mathbf{1}$ represents the vector whose elements are all ones, $X_{\mathrm{S}} = [\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{s},1}) \ \cdots \ \boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{s},2n_{\mathrm{nS}}})]^\top$, and $X_{\mathrm{U}} = [\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{u},1}) \ \cdots \ \boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{u},n_{\mathrm{U}}})]^\top$. Then we have the analytical solution of this minimization problem as

$$\boldsymbol{w} = n_{\mathrm{U}} \cdot \left(X_{\mathrm{U}}^\top X_{\mathrm{U}} + 2\lambda n_{\mathrm{U}} I\right)^{-1} \left(\frac{A}{n_{\mathrm{nS}}} X_{\mathrm{S}}^\top \mathbf{1} + \frac{B+C}{n_{\mathrm{U}}} X_{\mathrm{U}}^\top \mathbf{1}\right). \tag{6.19}$$

Besides, we provide a deep learning method where we can obtain an approximation to the optimal solution. Specifically, we employ a deep model f with logistic loss: $\ell_{\mathrm{LG}}(z,t) = \log(1+\exp(-tz))$. Then we directly optimize the objective function Eq. (6.13), into which ℓ_{LG} substituted:

$$\hat{f} = \operatorname*{arg\,min}_{f \in \mathcal{F}} \hat{R}_{\mathrm{nSU},\ell_{\mathrm{LG}}}(f). \tag{6.20}$$

6.3.4 Estimating π_+, π_{s}, and ρ_{d} with the MPE method

In the aforementioned method, similar rate π_{s}, class-prior π_+, and noise rate ρ_{d} are assumed to be given in advance, which is not always true. Here, we thus provide a practical method to estimate them. Mixture proportion estimation (MPE) [13] is the following problem: Let F, G, and H be probability distributions on $(\mathcal{X}, \mathfrak{S})$ such that

$$F = (1-\kappa)G + \kappa H, \tag{6.21}$$

where $0 \leq \kappa \leq 1$. Given random samples from F and H, estimate κ.

Similarly, the distributions of both D_{u} and $\tilde{D}_{\mathrm{s}}$ (see Eqs. (6.10) and (6.11)) have mixture representations according to Eq. (6.8) and Eq. (6.9). By substituting $(1-\pi_{\mathrm{s}})$ for π_{d}, we obtain

$$P(\boldsymbol{x}, \boldsymbol{x}') = (1-\pi_{\mathrm{s}}) P_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}') + \pi_{\mathrm{s}} P_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}'), \tag{6.22}$$

$$\tilde{P}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') = (1-\rho_{\mathrm{d}}) P_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') + \rho_{\mathrm{d}} P_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}'). \tag{6.23}$$

The above equations are similar to the standard MPE format but the accessible samples are not from the corresponding F and H. Therefore, by further calculating $[(6.22)\times(1-\rho_{\mathrm{d}}) - (6.23)\times\pi_{\mathrm{s}}]$, $[(6.22)\times\rho_{\mathrm{d}} - (6.23)\times(1-\pi_{\mathrm{s}})]$

respectively and organizing, we obtain

$$P(\boldsymbol{x}, \boldsymbol{x}') = (1 - \frac{\pi_{\mathrm{s}}}{1-\rho_{\mathrm{d}}})P_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}') + \frac{\pi_{\mathrm{s}}}{1-\rho_{\mathrm{d}}}\tilde{P}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}'), \tag{6.24}$$

$$\tilde{P}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') = (1 - \frac{\rho_{\mathrm{d}}}{1-\pi_{\mathrm{s}}})P_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') + \frac{\rho_{\mathrm{d}}}{1-\pi_{\mathrm{s}}}P(\boldsymbol{x}, \boldsymbol{x}'). \tag{6.25}$$

According to these mixture representations, i.e., Eq. (6.22) and Eq. (6.23), we assume that $1 - \rho_{\mathrm{d}} > \pi_{\mathrm{s}}$, which can easily be held because the proportion of similar data pairs in $\tilde{D}_{\mathrm{s}}$ (denoted by $1 - \rho_{\mathrm{d}}$) which it collects the similar data pairs purposely, is apparently bigger than that in unlabeled data (denoted by π_{s}). More discussion can be found in Appendix D.6.

Based on this reasonable assumption, we have the following lemma:

Lemma 6.3.2. *Assume $1 - \rho_{\mathrm{d}} > \pi_{\mathrm{s}}$. Eq. (6.24) and Eq. (6.25) can be equivalently rewritten as a standard MPE format such that*

$$P(\boldsymbol{x}, \boldsymbol{x}') = (1-\gamma)P_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}') + \gamma\tilde{P}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}'), \tag{6.26}$$

$$\tilde{P}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') = (1-\kappa)P_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') + \kappa P(\boldsymbol{x}, \boldsymbol{x}'). \tag{6.27}$$

where $\gamma = \frac{\pi_{\mathrm{s}}}{1-\rho_{\mathrm{d}}} \in [0, 1)$, $\kappa = \frac{\rho_{\mathrm{d}}}{1-\pi_{\mathrm{s}}} \in [0, 1)$.

Proof. Lemma 6.3.2 directly follows from Eq. (6.24) and Eq. (6.25) under the assumption $1 - \rho_{\mathrm{d}} > \pi_{\mathrm{s}}$. □

Note that since we have no information about G in the original MPE problem, without additional assumptions, MPE is ill-defined and the mixture proportion κ is not identifiable.

The weakest and most common assumption to yield the identifiability of the mixture proportion κ is the *irreducibility assumption* [13]:

Definition 6.3.1 ([97])**.** *Let G, and H be probability distributions. We say that G is irreducible with respect to H if there exists no decomposition of the form $G = \gamma H + (1-\gamma)F'$, where F' is some probability distribution and $0 < \gamma \leq 1$. We say that G and H are mutually irreducible if G is irreducible with respect to H and vice versa.*

The irreducibility assumption states that the maximum proportion of H in G approaches to 0, otherwise there would exist such an F'. Consider $\mathfrak{S}$ is the set of measurable sets in $\mathcal{X}$ and above discussion straightforwardly implies the following fact [98, 13]: G being irreducible with respect to H is equivalent to $\text{supp}(H) \not\subset \text{supp}(G)$, i.e.,

$$\inf_{S\in\mathfrak{S},H(S)>0} \frac{G(S)}{H(S)} = 0.$$

In general, the distributions of positive data and negative data are assumed to be mutually irreducible [98], which leads to the following theorem.

Theorem 6.3.3. *Assume that the positive data distribution P_+ and the negative data distribution P_- are mutually irreducible, then P_d is irreducible with respect to $\tilde{P}_\mathrm{s}$, and P_s is irreducible with respect to P. Thus, the mixture proportions γ and κ in Lemma 6.3.2 is identifiable.*

A detailed proof is provided in Appendix D.7.

Based on Lemma 6.3.2 and Theorem 3, we can effectively estimate γ and κ by the MPE method [93]. After estimating γ and κ, we can reversely calculate π_s, ρ_d, and π_+ according to the definitions of γ and κ in Lemma 6.3.2 as follows:

- Similar rate: $\pi_\mathrm{s} = \frac{\gamma(1-\kappa)}{1-\gamma\kappa}$,
- Noise rate: $\rho_\mathrm{d} = \frac{\kappa(1-\gamma)}{1-\gamma\kappa}$,
- Class-prior[4]: $\pi_+ = \frac{\sqrt{2\frac{\gamma(1-\kappa)}{1-\gamma\kappa}-1}+1}{2}$..

Overall, our method for nSU classification is summarized in Algorithm 4.

6.3.5 Error Bound Analysis

In this section, we derive a generalization error bound for the nSU classification.

[4] $2\pi_\mathrm{s} - 1 = \pi_\mathrm{s} - \pi_\mathrm{d} = (\pi_+ - \pi_-)^2 = (2\pi_+ - 1)^2$, such that $\pi_+ = \frac{\sqrt{2\frac{\gamma(1-\kappa)}{1-\gamma\kappa}-1}+1}{2}$

Algorithm 4 nSU classification.

Input: Noisy similar data pairs $\tilde{D}_{\mathrm{s}}$ and unlabeled data D_{u};
Output: The classifier $\hat{f}$;
Stage 1. Estimate the similar rate π_{s} and noise rate ρ_{d}
Intermediate parameters $(\gamma, \kappa) = MPE(\tilde{D}_{\mathrm{s}}, D_{\mathrm{u}})$;
Compute $(\pi_{\mathrm{s}}, \rho_{\mathrm{d}}, \pi_+, \pi_-)$ from (γ, κ);
Stage 2. Obtain classifier $\hat{f}$
if Squared loss **then**
  Compute the analytical solution $\hat{\boldsymbol{w}}$ by Eq. (6.19);
end if
if Logistic loss **then**
  Approximate the optimal classifier f^* by the SGD;
end if
return $\hat{f}$;

Let $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ be a function class of the linear-in-parameter model, and $f^* = \arg\min_{f \in \mathcal{F}} R(f)$ be the true risk minimizer, and $\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{R}_{\mathrm{nSU},\ell}(f)$ be the empirical risk minimizer. By introducing a Rademacher Complexity bound assumption [5], i.e., for any probability density μ, $\mathfrak{R}(\mathcal{F}; n, \mu) \leq \frac{C_{\mathcal{F}}}{\sqrt{n}}$ for some constant $C_{\mathcal{F}} > 0$, then we can obtain the following theorem.

Theorem 6.3.4. *Assume the loss function ℓ is ρ-Lipschitz with respect to the input instance $\boldsymbol{x}$ ($\rho \in (0, \infty)$), and all functions in the hypothesis class $\mathcal{F}$ are bounded by C_{b}, i.e., $\|f\|_\infty \leq C_{\mathrm{b}}$ for any $f \in \mathcal{F}$. Let $C_\ell = \sup_{t \in \{\pm 1\}} \ell(C_{\mathrm{b}}, t)$. For any $\delta > 0$, with probability at least $1 - \delta$,*

$$R(\hat{f}) - R(f^*) \leq \frac{4A\rho C_{\mathcal{F}} + A\sqrt{2C_\ell^2 \log \frac{4}{\delta}}}{\sqrt{2n_{\mathrm{nS}}}} + \frac{2(-B-C)\rho C_{\mathcal{F}} + (-B-C)\sqrt{\frac{1}{2}C_\ell^2 \log \frac{4}{\delta}}}{\sqrt{n_{\mathrm{U}}}}, \tag{6.28}$$

where A, B, and C are defined in Eqs. (6.16), (6.17), and (6.18).

A detailed proof is provided in Appendix D.8.

Theorem 6.3.4 implies that the expected risk of the classifier learned from nSU data is consistent with that of the classifier learned from standard positive and negative data if we have π_+ and ρ_{d} in advance. The

convergence rate is $\mathcal{O}_p(1/\sqrt{n_{\mathrm{nS}}}+1/\sqrt{n_{\mathrm{U}}})$, which achieves the optimal parametric rate for the empirical risk minimization without additional assumptions [75].

6.4 Experiments

6.4.1 Data Generation and Common Setup

To obtain nSU data, first, we collected raw binary classification datasets which consist of positive and negative data, while leaving 10% of the data as test data. Then we converted the labeled data to noisy similar data pairs according to class-prior π_+ and noise rate ρ_{d}. Specifically, we randomly subsampled similar data and dissimilar data pairs following the ratio of $1-\rho_{\mathrm{d}}$ and ρ_{d}. The similar data pairs consisted of positive and negative pairs with a ratio of π_+^2 and π_-^2. After that, we randomly selected unlabeled data samples from positive data and negative data with a ratio of π_+ and π_-.

For all the experiments, the sample size of the noisy similar data pairs was fixed to 4000, while the sample size of the unlabeled data was fixed to 2000. The class-prior π_+ and noise rate ρ_{d} were estimated by the MPE method [93]. For the first MPE for γ, we used the default parameters. For the second MPE for κ, to ensure the term in the square root in the formula $\pi_+ = \frac{\sqrt{2\frac{\gamma(1-\kappa)}{1-\gamma\kappa}-1}+1}{2}$ to be greater than zero, we set $\lambda_{\mathrm{right}} = 2 - 1/\hat{\gamma}$ while kept all other parameters as default, where λ_{right} is a parameter in the MPE method [93] and $\hat{\gamma}$ is the estimated value.

We used the linear basis functions and the regularization parameter λ was fixed to 10^{-4}. For the deep model, we employed a 3-layer MLP (multilayer perceptron) with the softsign active function ($\mathrm{Saf}(x) = x/(1+|x|)$). We used the stochastic gradient descent (SGD) optimizer with an initial learning rate of 0.002, which decays every 40 epochs by a factor of 0.1 with 200 epochs in total.

6.4.2 Baselines

We implemented our method with two models. In Stage 1, we employed the KM2 algorithm [93] to estimate the class-prior and noise rate. In Stage

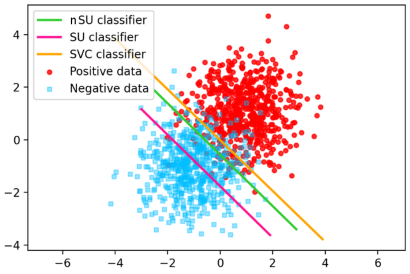
2, for the linear model, we obtained the analytical solution by Eq. (6.19) (denoted by -LC (Linear Classifier)); for the deep neural network, we used the SGD to optimize the model (denoted by -DC (Deep Classifier)). We compared our proposed method with state-of-the-art methods:

- SU classification [5], which is the state-of-the-art method for learning from similarity and unlabeled data. This method was also implemented with two models, i.e., linear and deep models.
- nSD classification [23], which learns a classifier from noisy similarity and dissimilarity data pairs. To make it compatible with the nSU data, we treated the unlabeled data as the noisy dissimilarity data and fed the ground-truth class-prior and noises rates to the nSD algorithm.
- Information-theoretic metric learning (ITML) [25], which utilizes pairwise similarity and dissimilarity as constraints to learn a metric. Then, k-means clustering is applied on test data with the learned metric.

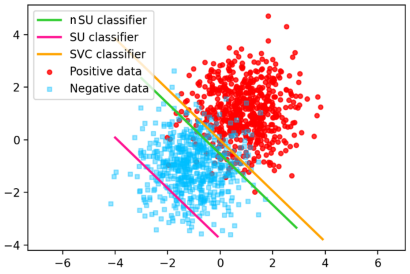
Moreover, we also implemented some unsupervised learning methods, i.e., k-means (KM) [69] and hierarchical clustering schemes (HC) [49]. For unsupervised learning methods, we directly used the implementations on scikit-learn [90]. We also employed a support vector classifier (SVC) [22] with a linear kernel learned from fully-supervised data as a benchmark. Note that learning a classifier without class information will lose the mapping between the cluster nodes and the semantic classes. The semantic classes can be identified with some prior knowledge, e.g., the exact π_+ or the sign of $(\pi_+ - \pi_-)$. Here, we employed the Hungarian algorithm [55], which is a commonly used method for evaluating the clustering accuracy, to assign the output nodes to the dominant semantic classes by using some training examples with class labels.

6.4.3 Experiments on Synthetic Datasets

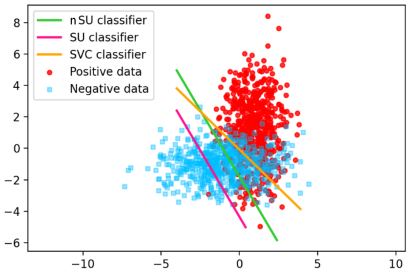
We generated synthetic data drawn from two-dimension normal distributions. Settings with various combinations of parameters were tested. The results are shown in Figures 6.2 and 6.3. For the first case, the positive



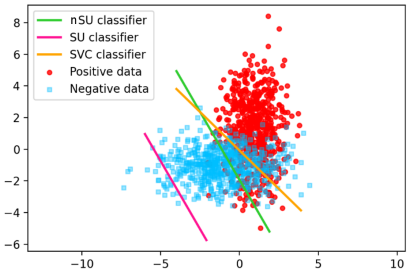
(a) syn1: $\{\pi_+ = 0.7, \rho_{\mathrm{d}} = 0.2\}$.



(b) syn1: $\{\pi_+ = 0.7, \rho_{\mathrm{d}} = 0.3\}$.



(c) syn2: $\{\pi_+ = 0.7, \rho_{\mathrm{d}} = 0.2\}$.

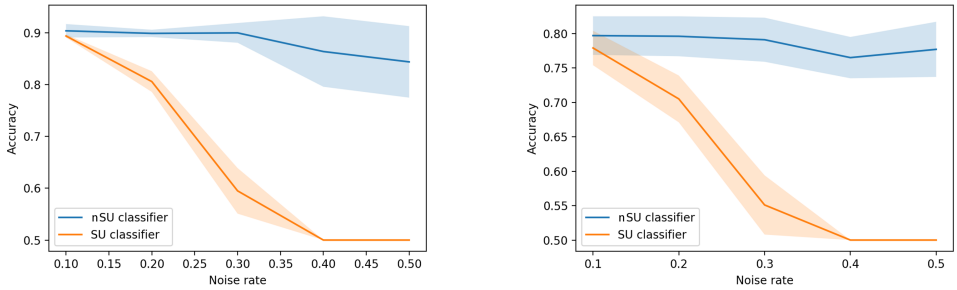


(d) syn2: $\{\pi_+ = 0.7, \rho_{\mathrm{d}} = 0.3\}$.

Figure 6.2: Illustrations based on a single trail of the four setups. The green and pink lines are decision boundaries learned by nSU and SU classification from nSU data respectively. The orange boundary is obtained by SVC with a linear kernel using the fully-supervised data.

data follows the Gaussian distribution $\mathcal{N}\left(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right)$, and the negative data follows $\mathcal{N}\left(\begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\right)$, which we called the syn1 dataset. For another case, the positive data follows $\mathcal{N}\left(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 2\end{pmatrix}\right)$, and the negative data follows $\mathcal{N}\left(\begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}2 & 0\\0 & 1\end{pmatrix}\right)$, having more overlap, which we called the syn2 dataset. From such two binary datasets, we generated the nSU data with $\{\pi_+ = 0.7, \rho_{\mathrm{d}} = 0.2\}$ (Figures 2.a & 2.c) and $\{\pi_+ = 0.7, \rho_{\mathrm{d}} = 0.3\}$ (Figures 2.b & 2.d). Here we employed an SVC with a linear kernel learned from fully-supervised data as a benchmark. The class-prior and noise rate were assumed to be known.

In Figure 6.2, we see that nSU classifier is closer to SVC classifier than



(a) syn1 with stepwise noise.

(b) syn2 with stepwise noise.

Figure 6.3: Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on syn1 and syn2 with stepwise noise. The class-prior π_+ is fixed at 0.7. When the noise rate is higher than 0.3, the SU classifier even loses the identification ability, and assigns all the test instances the same label, which leads to a steady 0.5 accuracy.

SU classifier in all four setups. As the noise rate increases from 0.2 to 0.3, the SU classifier moves further away from the SVC classifier, while the nSU classifier is hardly affected. From Figure 6.3, we see that the accuracy of the SU classifier drops dramatically with the increased noise rate on both datasets. Meanwhile, there is only a slight fluctuation of the nSU classifier. The gap between the accuracy of the nSU classifier and the SVC classifier trained on clean data is small, i.e., within two percentage points.

6.4.4 Experiments on UCI and LIBSVM Datasets

Here datasets were obtained from the LIBSVM data [16] and UCI Machine Learning Repository [27]. Then we generated the corresponding nSU data following the data generation method. Tables 6.1 and 6.2 demonstrate the remarkable superiority of the nSU classifier over the SU classifier and nSD classifier on all the benchmark-simulated datasets. For some datasets, ITML also achieved comparable accuracy. It is because ITML is a cluster-based method, whose performance is closely associated with the natural structure of the dataset, i.e., whether the *low density separation*[5] condition holds.

[5]The decision boundary should lie in a low-density region.

Table 6.1: Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on the UCI and LIBSVM datasets with $\{\pi_+ = 0.7, \rho_{\mathrm{d}} = 0.2\}$. The best (except SVC) and comparable methods (paired t-test at significance level 5%) are highlighted in bold.

Dataset	nSU-LC	nSU-DC	SU-LC	SU-DC	nSD	KM	ITML	HC	SVC
australian	**83.1±4.4**	68.0±8.8	63.7±5.4	54.3±2.9	60.0±2.9	69.1±4.2	**82.2±4.4**	75.4±11.3	84.9±4.1
breast-cancer	**93.6±1.7**	85.8±7.2	84.1±3.8	79.4±4.0	93.0±1.2	**95.9±2.4**	64.1±1.3	**95.7±2.0**	96.2±1.3
fourclass	**71.5±6.7**	64.8±6.2	63.7±1.5	63.4±2.1	64.7±4.8	64.8±5.0	**80.6±5.8**	63.4±15.8	98.2±1.0
magic	**76.2±1.5**	66.2±2.6	69.5±3.5	64.5±4.1	70.5±3.2	61.8±2.3	68.8±5.9	64.7±1.1	80.3±1.0
cod-rna	**89.8±2.7**	**85.6±3.6**	62.1±12.0	63.3±3.7	72.3±7.9	76.8±0.5	83.2±0.3	76.5±2.8	77.2±0.6
adult	67.9±1.0	59.1±6.8	63.6±4.2	58.6±3.0	75.7±0.3	71.1±0.6	**84.4±4.2**	72.6±2.3	70.5±1.0
banknote	**98.0±1.1**	62.2±8.2	**94.9±5.6**	60.6±8.9	62.6±3.9	62.9±2.9	59.1±7.0	66.1±2.0	100.0±0.0
heart	**83.3±3.7**	61.5±12.7	58.3±7.6	56.3±4.8	70.4±6.4	63.7±6.6	**80.0±10.0**	62.2±8.4	60.0±5.5
svmguide1	**76.1±7.0**	56.1±4.3	52.8±3.4	56.6±6.6	71.1±0.6	72.6±1.0	**80.3±1.3**	75.5±12.0	93.5±1.2
htru_2	**96.4±1.3**	81.1±8.4	**96.1±1.2**	86.3±5.7	92.0±1.6	73.4±2.3	86.0±5.0	71.0±8.5	97.1±0.2

Table 6.2: Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on the UCI and LIBSVM datasets with $\{\pi_+ = 0.8, \rho_{\mathrm{d}} = 0.1\}$. The best (except SVC) and comparable methods (paired t-test at significance level 5%) are highlighted in bold.

Dataset	nSU-LC	nSU-DC	SU-LC	SU-DC	nSD	KM	ITML	HC	SVC
australian	**85.4±3.1**	55.1±2.4	82.6±3.6	54.6±2.9	64.6±6.7	69.1±4.2	80.8±5.2	78.9±4.1	86.7±4.2
breast-cancer	**94.8±1.7**	66.7±3.7	93.0±2.4	64.9±2.6	89.0±3.3	**95.9±2.4**	67.7±8.0	94.8±1.9	95.4±2.4
fourclass	**70.8±5.9**	61.8±5.2	57.5±5.3	62.1±5.1	64.3±4.9	62.3±6.4	**80.9±5.7**	55.2±5.2	97.5±1.0
magic	**75.6±2.2**	63.2±7.4	71.4±1.6	65.9±1.4	68.5±3.0	58.3±0.6	**71.7±19.2**	63.4±3.6	75.9±0.7
cod-rna	**90.1±0.9**	58.7±10.9	71.9±7.0	66.7±0.1	66.7±7.8	77.2±0.5	83.2±0.3	77.8±0.3	59.2±0.3
adult	74.0±2.2	60.9±5.7	73.2±2.2	57.7±3.4	71.1±0.6	75.8±0.1	**84.4±4.2**	65.3±8.2	65.5±0.8
banknote	**97.9±0.7**	66.7±6.4	**96.7±2.3**	63.2±9.5	94.1±7.6	63.8±3.4	83.4±13.5	64.1±4.3	100.0±0.0
heart	**77.0±8.4**	63.0±9.4	61.5±10.0	55.6±2.6	63.7±4.1	63.7±6.6	**80.7±8.8**	63.0±10.1	56.6±0.0
svmguide1	73.5±2.5	65.4±1.8	67.3±4.1	57.0±4.6	67.8±3.5	73.3±1.0	**81.5±2.3**	70.6±13.1	91.1±1.4
htru_2	**97.8±0.2**	76.4±9.5	97.1±0.5	84.0±7.2	87.5±3.8	77.2±1.5	81.3±13.2	82.2±7.3	97.2±0.3

Table 6.3: Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on text datasets with $\{\pi_+ = 0.7, \rho_{\mathrm{d}} = 0.2\}$. The best (except SVC) and comparable methods (paired t-test at significance level 5%) are highlighted in bold.

Dataset	nSU-LC	nSU-DC	SU-LC	SU-DC	nSD	KM	ITML	HC	SVC
SMS Spam	72.4±2.3	71.1±5.3	56.5±4.4	70.4±2.9	86.5±0.1	69.3±0.7	**93.1±4.4**	58.6±9.4	84.2±1.5
News_05	**82.9±2.0**	59.4±4.9	72.4±5.3	54.3±1.9	56.6±2.3	62.7±3.7	72.9±8.0	61.0±3.6	92.4±1.9
News_16	**77.7±1.9**	56.8±4.6	64.7±2.0	52.2±1.4	60.0±9.4	**75.6±3.1**	71.3±9.2	69.3±4.8	90.2±2.3
News_27	**82.9±2.8**	52.1±1.1	78.1±6.3	54.3±1.8	57.6±8.0	62.8±2.9	**80.7±14.5**	61.3±3.7	96.5±1.7
News_38	**75.7±3.3**	55.3±5.0	65.4±5.1	51.9±1.5	55.5±7.0	59.2±1.7	**76.6±4.3**	59.3±5.5	85.4±3.3
News_49	**84.0±2.2**	53.4±4.4	74.0±5.7	54.3±3.9	54.1±3.8	67.6±5.1	75.8±9.4	66.3±9.9	92.4±1.0

6.4.5 Experiments on Text Datasets

SMS Spam [1] is a public set of short message service (SMS) labeled messages that have been collected for mobile phone spam research, which is

Table 6.4: Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on text datasets with $\{\pi_+ = 0.8, \rho_d = 0.1\}$. The best (except SVC) and comparable methods (paired t-test at significance level 5%) are highlighted in bold.

Dataset	nSU-LC	nSU-DC	SU-LC	SU-DC	nSD	KM	ITML	HC	SVC
SMS Spam	75.9±2.2	63.0±6.4	58.1±3.1	70.3±1.5	86.5±0.1	70.0±1.0	**93.3±1.4**	74.3±3.7	79.0±1.4
News_05	**88.5±1.7**	54.9±3.6	83.4±3.5	56.6±3.8	59.2±5.0	63.4±3.3	76.0±3.6	63.1±2.9	89.2±2.4
News_16	**83.2±2.4**	53.6±3.6	74.7±3.2	55.9±7.0	53.5±5.9	74.5±3.1	71.4±5.4	67.0±5.4	85.3±1.7
News_27	**92.8±3.0**	64.6±13.6	88.3±1.7	60.6±8.8	64.4±9.5	62.6±2.9	74.0±9.4	61.5±9.2	94.7±1.0
News_38	**80.6±4.0**	55.3±2.4	75.9±3.5	55.4±4.1	55.8±9.2	58.2±1.8	71.7±5.9	53.4±2.3	81.8±2.0
News_49	**87.1±3.2**	58.1±8.6	83.8±4.1	51.7±1.1	59.8±8.7	67.7±4.4	74.1±4.5	65.2±5.8	87.8±2.6

Table 6.5: Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on image datasets with $\{\pi_+ = 0.7, \rho_d = 0.2\}$. The best (except SVC) and comparable methods (paired t-test at significance level 5%) are highlighted in bold.

Dataset	nSU-LC	nSU-DC	SU-LC	SU-DC	nSD	KM	ITML	HC	SVC
Cifar_03	**70.4±2.9**	57.8±6.5	64.8±4.1	51.9±1.5	58.6±8.2	67.1±1.1	64.9±2.6	68.7±2.1	84.7±1.0
Cifar_14	**75.0±3.0**	65.0±3.1	73.2±3.3	54.0±2.2	56.3±7.6	63.1±1.9	67.3±9.8	69.3±4.8	88.7±1.0
Cifar_25	**72.6±1.6**	56.6±3.9	68.0±3.4	54.0±2.6	62.6±5.4	62.8±2.9	**70.1±9.1**	61.3±3.7	87.3±0.8

composed of 5,574 English, real, and non-encoded messages, tagged according to being legitimate or spam. News20 is a collection of approximately 20,000 newsgroup documents, partitioned nearly evenly across 20 different newsgroups. We selected ten newsgroups and paired them into five datasets, i.e., News_05, ..., News_49. The specific class information is provided in Appendix D.9. For these two datasets, we used GloVe [91] to extract vector representations from raw text data. Tables 6.3 and 6.4 show the consistent superiority of the nSU classifier over the SU classifier and nSD classifier on all the benchmark-simulated datasets. Due to the natural structure of the dataset, the clustering methods occasionally achieved the best performance. However, the clustering methods were not stable, with larger standard deviations.

6.4.6 Experiments on Image Datasets

CIFAR-10 [54] has $32 \times 32 \times 3$ color images including 50,000 training images and 10,000 test images of 10 classes. Similarly, we selected six classes and paired them into three datasets, i.e., Cifar_03, Cifar_14, and Cifar_25. The specific class information is provided in Appendix D.9. Since the raw

Table 6.6: Means and Standard Deviations (Percentage) of Classification Accuracy over 5 trials on image datasets with $\{\pi_+ = 0.8, \rho_{\mathrm{d}} = 0.1\}$. The best (except SVC) and comparable methods (paired t-test at significance level 5%) are highlighted in bold.

Dataset	nSU-LC	nSU-DC	SU-LC	SU-DC	nSD	KM	ITML	HC	SVC
Cifar_03	**72.7±2.2**	59.0±6.6	69.6±2.5	51.2±0.7	57.0±4.4	66.2±0.9	65.6±1.4	67.2±3.5	77.2±1.4
Cifar_14	**76.7±1.4**	60.2±7.2	**75.6±2.6**	54.7±2.1	59.5±4.7	62.0±1.4	**71.3±9.2**	57.4±4.3	85.2±1.8
Cifar_25	**76.9±1.9**	64.8±4.0	75.1±2.1	53.6±2.1	65.1±6.3	66.4±2.0	66.8±3.2	66.1±3.3	84.1±0.6

feature of *CIFAR-10* is far from a good representation, we extracted the 32-dimensional features of images by a deep variational autoencoder [52]. Namely, both linear methods (-LC) and deep methods (-DC) in our experiment are fitted on top of the same pre-trained embedding, and take the same advantage of *deep models* extracting good representations. Besides, the linear methods (-LC) have the analytical solution for the objective Eq (6.14) while the deep methods (-DC) use the SGD method to obtain an approximation to the optimal solution, which introduces additional optimization errors. Therefore, the linear methods (-LC) could outperform the deep methods (-DC) in our experiment, which is not conflict with the fact that generally *deep models* do better than purely *linear models* on image datasets like *CIFAR-10*. Tables 6.5 and 6.6 show the consistent superiority of the nSU classifier over the SU classifier, the nSD classifier, and the clutering methods on all the benchmark-simulated datasets.

6.5 Summary

In this chapter, we proposed a novel weakly supervised learning (WSL) problem named nSU classification, which considers the case where the privacy-preserve data, i.e., similar data pairs are corrupted with the mutually contaminated distributions model. To tackle this problem, nSU classification provided a robust risk-consistent estimator for learning from nSU data. The mixture proportion estimation method was employed to estimate the noise rate and the class-prior probabilities. When utilizing proper models and loss functions, we showed that our optimization problem becomes convex. Specifically, there exists a closed-form solution to the objective function with a linear-in-parameter model combined with the

squared loss. We also established a generalization error bound for the proposed method. Experiments conducted on benchmark datasets demonstrated that our method can excellently solve the aforementioned WSL problem and showed superiority over the baseline methods.

Chapter 7

Conclusion

In this thesis, motivated by the phenomenon that machine learning systems have been widely adopted and entrusted with important tasks in our daily life yet acquiring high-quality data is challenging, we investigate the problem of learning with noisy labels incorporating fairness and privacy concerns. First, We propose a method that transforms data points with noisy class labels to data pairs with noisy similarity labels, which reduces the noise rate with a theoretical guarantee and thus makes the noise easier to handle. Second, we design an assumption-free curriculum that learns the clean classifier, as well as the transition matrix simultaneously by allocating reliable triplets in the training curriculum based on the novel TCP metric. Third, we provide general frameworks for learning fair classifiers with noisy labels. For statistical fairness notions, we rewrite the classification risk and the fairness metric in terms of noisy data and thereby build robust classifiers. For the causality-based fairness notion, We exploit the internal causal structure of data to effectively model both the label noise and counterfactual fairness. Finally, we propose a denoised and unbiased estimator for the classification risk with respect to the accurately labeled data by employing the noisy data with indirect supervision and then learn the optimal model under the empirical risk minimization framework.

Appendix A

Supplementary for Chapter 3

A.1 Proof of Theorem 1

Theorem A.1.1. *Assume that the dataset is balanced (each class has the same amount of instances, and c classes in total), and the noise is class-dependent. Given a class transition matrix T_c, such that $T_{c,ij} = P(\tilde{Y} = j|Y = i)$. The elements of the corresponding similarity transition matrix T_s can be calculated as*

$$T_{s,00} = \frac{c^2 - c - \left(\sum_j (\sum_i T_{c,ij})^2 - ||T_c||_{\text{Fro}}^2\right)}{c^2 - c}, \qquad T_{s,10} = \frac{c - ||T_c||_{\text{Fro}}^2}{c},$$
$$T_{s,01} = \frac{\sum_j (\sum_i T_{c,ij})^2 - ||T_c||_{\text{Fro}}^2}{c^2 - c}, \qquad T_{s,11} = \frac{||T_c||_{\text{Fro}}^2}{c}.$$

Proof. Assume each class has n samples. $n^2 T_{c,ij} T_{c,i'j'}$ represents the number the kind of data pairs composed by points of $(\tilde{Y} = j|Y = i)$ and $(\tilde{Y} = j'|Y = i')$. For the first element $T_{s,00}$, $n^2 \sum_{i \neq i'} T_{c,ij} T_{c,i'j'}$ is the number of data pairs with clean similarity labels $H = 0$, while $n^2 \sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}$ is the number of data pairs with clean similarity labels $H = 0$ and noisy similarity labels $\tilde{H} = 0$. Thus the proportion of these two terms is exact the $T_{s,00} = P(\tilde{H} = 0|H = 0)$. The remaining three elements can be represented in the same way. The primal representations are as follows,

$$T_{s,00} = \frac{\sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}}{\sum_{i \neq i'} T_{c,ij} T_{c,i'j'}}, \qquad T_{s,10} = \frac{\sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'}}{\sum_{i = i'} T_{c,ij} T_{c,i'j'}},$$
$$T_{s,01} = \frac{\sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'}}{\sum_{i \neq i'} T_{c,ij} T_{c,i'j'}}, \qquad T_{s,11} = \frac{\sum_{i = i', j = j'} T_{c,ij} T_{c,i'j'}}{\sum_{i = i'} T_{c,ij} T_{c,i'j'}}.$$

Further, note that

$$\sum_{i=i'} T_{c,i,j}T_{c,i',j'} = \sum_{i,j,j'} T_{c,i,j}T_{c,i,j'} = \sum_{i}(\sum_{j} T_{c,i,j})(\sum_{j'} T_{c,i,j'}) = c,$$
$$\sum_{i\neq i'} T_{c,i,j}T_{c,i',j'} = \sum_{i\neq i',j,j'} T_{c,i,j}T_{c,i',j'} = \sum_{i\neq i'}(\sum_{j} T_{c,i,j})(\sum_{j'} T_{c,i,j'}) = (c-1)c,$$
$$\sum_{i=i',j=j'} T_{c,ij}T_{c,i'j'} = ||T_c||^2_{\text{Fro}},$$
$$\sum_{i\neq i',j=j'} T_{c,ij}T_{c,i'j'} = \sum_{j}(\sum_{i} T_{c,ij})^2 - ||T_c||^2_{\text{Fro}}.$$

Substituting above equations to the primal representations, we have the Theorem 1 proved. □

A.2 Pointwise implies pairwise

For an invertible T_c, denote by $\mathbf{v}_j$ the j-th column of T_c and $\mathbf{1}$ the all-one vector. Then,

$$\sum_{j}(\sum_{i} T_{c,ij})^2 = \sum_{j}\langle \mathbf{v}_j, \mathbf{1}\rangle^2 \leq \sum_{j} ||\mathbf{v}_j||^2||\mathbf{1}||^2 = c||T_c||^2_{\text{Fro}},$$

where we use the Cauchy–Schwarz inequality [102] in the second step. Further, we have

$$\begin{aligned}
T_{s,11} + T_{s,00} &= \frac{||T_c||^2_{\text{Fro}}}{c} + \frac{c^2 - c - \left(\sum_j(\sum_i T_{c,ij})^2 - ||T_c||^2_{\text{Fro}}\right)}{c^2 - c} \\
&= \frac{(c-1)||T_c||^2_{\text{Fro}} + c^2 - c - \left(\sum_j(\sum_i T_{c,ij})^2 - ||T_c||^2_{\text{Fro}}\right)}{c^2 - c} \\
&= \frac{(c-1)||T_c||^2_{\text{Fro}} + c^2 - c - \left(\sum_j\langle \mathbf{v}_j, \mathbf{1}\rangle^2 - ||T_c||^2_{\text{Fro}}\right)}{c^2 - c} \\
&\geq \frac{(c-1)||T_c||^2_{\text{Fro}} + c^2 - c - \left(c||T_c||^2_{\text{Fro}} - ||T_c||^2_{\text{Fro}}\right)}{c^2 - c} \\
&= 1.
\end{aligned}$$

Thus the learnability of the pointwise classification implies the learnability of the reduced pairwise classification.

A.3 Proof of Theorem 2

Theorem A.3.1. *Assume that the dataset is balanced (each class has the same amount of samples), and the noise is class-dependent. When the number of classes $c \geq 8$, the noise rate of noisy similarity labels is lower than that of the noisy class labels.*

Proof. Assume each class has n points. As we state in the proof of Theorem A.1.1, the number of data pairs with clean similarity labels $H = 0$ and noisy similarity labels $\tilde{H} = 0$ is $n^2 \sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}$. We denote it by N_{00}. Similarly, we have,

$$N_{00} = n^2 \sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}, \qquad N_{10} = n^2 \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'},$$
$$N_{01} = n^2 \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'}, \qquad N_{11} = n^2 \sum_{i = i', j = j'} T_{c,ij} T_{c,i'j'}.$$

The noise rate is the proportion of the number of noisy labels to the number of total labels. Assume that the number of classes is c. We have

$$S_{noise} = \frac{N_{01} + N_{10}}{N_{00} + N_{01} + N_{10} + N_{11}} = \frac{N_{01} + N_{10}}{c^2 n^2},$$
$$C_{noise} = \frac{n \sum_{i \neq j} T_{c,ij}}{cn}.$$

Let S_{noise} minus C_{noise}, we have

$$\begin{aligned} S_{noise} - C_{noise} &= \frac{n^2 \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} + n^2 \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'}}{c^2 n^2} - \frac{n \sum_{i \neq j} T_{c,ij}}{cn} \\ &= \frac{\sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} + \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'} - c \sum_{i \neq j} T_{c,ij}}{c^2}. \end{aligned}$$

Let $A = \sum_{i\neq i',j=j'} T_{c,ij}T_{c,i'j'} + \sum_{i=i',j\neq j'} T_{c,ij}T_{c,i'j'} - c\sum_{i\neq j} T_{c,ij}$, we have

$$\begin{aligned}
A &= \sum_{i\neq i',j=j'} T_{c,ij}T_{c,i'j'} + \sum_{i=i',j\neq j'} T_{c,ij}T_{c,i'j'} - c\sum_{i\neq j} T_{c,ij} \\
&= \sum_{i\neq i',j=j'} T_{c,ij}T_{c,i'j'} + \sum_{i=i',j\neq j'} T_{c,ij}T_{c,i'j'} - c(\sum_{i,j} T_{c,ij} - \sum_{i=j} T_{c,ij}) \\
&= \sum_{i\neq i',j=j'} T_{c,ij}T_{c,i'j'} + \sum_{i=i',j\neq j'} T_{c,ij}T_{c,i'j'} - c^2 + c\sum_{i=j} T_{c,ij}.
\end{aligned}$$

The second equation holds because the row sum of T_c is 1.

For the first term $\sum_{i\neq i',j=j'} T_{c,ij}T_{c,i'j'}$, notice that:

$$\begin{aligned}
\sum_{i\neq i',j=j'} T_{c,ij}T_{c,i'j'} &= \sum_j \sum_i T_{c,ij}(\sum_{i'\neq i} T_{c,i'j}) \\
&= \sum_j \sum_i T_{c,ij}(\sum_{i'\neq i} T_{c,i'j} + T_{c,ij} - T_{c,ij}) \\
&= \sum_j \sum_i T_{c,ij}(\sum_{i'} T_{c,i'j} - T_{c,ij}) \\
&= \sum_j \sum_i T_{c,ij}(S_j - T_{c,ij}) \\
&\quad (S_j \text{ is the column sum of the } j-th \text{ column}) \\
&= \sum_j \sum_i T_{c,ij}S_j - T_{c,ij}^2 \\
&= \sum_j S_j \sum_i T_{c,ij} - \sum_j \sum_i T_{c,ij}^2 \\
&= \sum_j S_j^2 - \sum_j \sum_i T_{c,ij}^2. \qquad \text{(A.1)}
\end{aligned}$$

Due to the symmetry of i and j, for the second term $\sum_{i=i',j\neq j'} T_{c,ij}T_{c,i'j'}$, we have

$$\begin{aligned}
\sum_{i=i',j\neq j'} T_{c,ij}T_{c,i'j'} &= \sum_j \sum_i T_{c,ij}(R_i - T_{c,ij}) \\
&\quad (R_i \text{ is the row sum of the } i-th \text{ row, and } R_i = 1) \\
&= \sum_j \sum_i T_{c,ij} - T_{c,ij}^2 \\
&= c - \sum_j \sum_i T_{c,ij}^2. \qquad \text{(A.2)}
\end{aligned}$$

Therefore, substituting Equation (A.1) and (A.2) into A, we have

$$A = \sum_j S_j^2 - \sum_j \sum_i T_{c,ij}^2 + c - \sum_j \sum_i T_{c,ij}^2 - c^2 + c \sum_{i=j} T_{c,ij}.$$

To prove $S_{noise} - C_{noise} \leq 0$ is equivalent to prove $A \leq 0$.

Let $M = c^2 - c$, $N = \sum_j S_j^2 - 2\sum_j \sum_i T_{ij}^2 + c\sum_{i=j} T_{ij}$ (we drop the subscript c in $T_{c,ij}$), and $A = N - M$. Now we utilize the Adjustment method [103] to scale N. For every iteration, we denote the original N by N_o, and the adjusted N by N_a.

Since $c \geq 8$, there can not exist three columns with column sum bigger than $c/2 - 1$. Otherwise, the sum of the three columns will be bigger than c, which is impossible because the sum of the whole matrix is c.

Therefore, first, we assume that the $j, k-th$ columns have column sum bigger than $c/2 - 1$. Then, for the row i, we add the elements l, which are not in $j, k - th$ columns, to the diagonal element. We have

$$\begin{aligned}
N_a - N_o &= (S_i + T_{il})^2 + (S_l + T_{il})^2 + cT_{il} - 2(T_{ii} + T_{il})^2 \\
&\quad - S_i^2 - S_l^2 + 2(T_{ii}^2 + T_{il}^2) \\
&= T_{il}(2T_{il} + 2S_i - 2S_l + c - 4T_{ii}) \\
&\geq T_{il}(2T_{il} - 2S_l + c - 2T_{ii}) \qquad (\because S_i \geq T_{ii}) \\
&> T_{il}(2T_{il} - c + 2 + c - 2T_{ii}) \qquad (\because S_l < c/2 - 1) \\
&\geq 0. \qquad (\because T_{ii} \leq 1)
\end{aligned}$$

We do such adjustment to every rows, then N_a is getting bigger and the adjusted matrix will only have values on diagonal elements and the $j, k-th$ columns. Since the diagonal elements are dominant in the row, $S_j + S_k < 2c/3 + 2/3$ (because for $i \neq j, k, T_{ij} + T_{ik} < 2/3$).

Assume that the column sum of $k - th$ column is no bigger than that of the $j - th$ column, and thus $S_k < c/3 + 1/3$. Then, for a row i, we add

the T_{ik} to T_{ii}. We have

$$\begin{aligned} N_a - N_o &= (S_i + T_{ik})^2 + (S_k + T_{ik})^2 + cT_{ik} - 2(T_{ii} + T_{ik})^2 \\ &\quad - S_i^2 - S_k^2 + 2(T_{ii}^2 + T_{ik}^2) \\ &= T_{ik}(2T_{ik} + 2S_i - 2S_k + c - 4T_{ii}) \\ &\geq T_{ik}(2T_{ik} - 2S_k + c - 2T_{ii}) \qquad (\because S_i \geq T_{ii}) \\ &> T_{ik}(2T_{ik} + c/3 - 2/3 - 2T_{ii}) \qquad (\because S_k < c/3 + 1/3) \\ &\geq 0. \qquad (\because c \geq 8, \text{ and } T_{ii} \leq 1) \end{aligned}$$

We do such adjustment to every rows, then N_a is getting bigger and the adjusted matrix will only have values on diagonal elements and the $j-th$ column, which is called final matrix.

Note that if there is only one column with a column sum bigger than $c/2-1$, we can adjust the rest $c-1$ columns as above and then obtain the final matrix as well. If there is no column with a column sum bigger than $c/2-1$, we can adjust all the elements as above and then obtain a *unit matrix*. For the unit matrix, $A = N - M < N_a - M = 0$, the Theorem A.3.1 is proved.

Now we process the final matrix. For simplification, we assume $j = 0$ in the final matrix. We denote the T_{ij} by b_i and T_{ii} by a_i, for $i = \{1, \ldots, c-1\}$. We have

$$\begin{aligned} N_a &= \sum_i a_i^2 + (1 + \sum_i b_i)^2 + c(\sum_i a_i + 1) - 2(\sum_i a_i^2 + \sum_i b_i^2 + 1) \\ &= (1 + \sum_i b_i)^2 + c\sum_i a_i + c - \sum_i a_i^2 - 2\sum_i b_i^2 - 2 \\ &= 1 + (\sum_i b_i)^2 + 2\sum_i b_i + c\sum_i a_i + c - \sum_i a_i^2 - 2\sum_i b_i^2 - 2 \\ &= (\sum_i b_i)^2 + 2\sum_i b_i - 2\sum_i b_i^2 + c\sum_i a_i - \sum_i a_i^2 + c - 1 \\ &= (\sum_i b_i)^2 + 2\sum_i b_i - 2\sum_i b_i^2 + c\sum_i (1 - b_i) - \sum_i (1 - b_i)^2 + c - 1 \\ &= (\sum_i b_i)^2 + 2\sum_i b_i - 2\sum_i b_i^2 + c^2 - c - c\sum_i b_i - \sum_i (1 - 2b_i + b_i^2) + c - 1 \\ &= (\sum_i b_i)^2 + 4\sum_i b_i - 3\sum_i b_i^2 - c\sum_i b_i + c^2 - c. \end{aligned}$$

Now we prove $A = N - M \leq N_a - M \leq 0$. Note that

$$\begin{aligned} N_a - M &= (\sum_i b_i)^2 + 4\sum_i b_i - 3\sum_i b_i^2 - c\sum_i b_i \\ &= (\sum_i b_i)^2 + 3\sum_i b_i - 3\sum_i b_i^2 - (c-1)\sum_i b_i \\ &= (\sum_i b_i)^2 + 3\sum_i b_i - 3\sum_i b_i^2 - (\sum_i (1-b_i) + \sum_i b_i)\sum_i b_i \\ &= 3\sum_i b_i - 3\sum_i b_i^2 - \sum_i (1-b_i)\sum_i b_i \\ &= 3\sum_i b_i(1-b_i) - \sum_i (1-b_i)\sum_i b_i. \end{aligned}$$

According to the rearrangement inequality[39], we have

$$\sum_i (1-b_i)\sum_i b_i \geq (c-1)\sum_i b_i(1-b_i).$$

Note that $c \geq 8$, thus $3\sum_i b_i(1-b_i) - \sum_i (1-b_i)\sum_i b_i \leq 0$, and $A \leq 0$. Therefore $S_{noise} - C_{noise} \leq 0$, and the equation holds if and only if the noise rate is 0 or every instances have the same noisy class label (i.e., there is one column in the T_c, of which every elements are 1, and the rest elements of the T_c are 0). Above two extreme situations are not considered in this paper. Namely, the noise rate of the noisy similarity labels is lower than that of the noisy class labels. Theorem A.3.1 is proved. □

A.4 Implementation of Class2Simi with Reweight

The expected risk for clean pairwise data is

$$R(f) = \mathbb{E}_{(X_i, X_j, H_{ij}) \sim \mathcal{D}}[\ell(\langle f(X_i), f(X_j)\rangle, H_{ij})],$$

where

$$\begin{aligned}\ell(\langle f(X_i), f(X_j)\rangle, H_{ij}) = &- \sum_{i,j} H_{ij} \log(\langle f(X_i), f(X_j)\rangle) \\ &+ (1 - H_{ij}) \log(1 - \langle f(X_i), f(X_j)\rangle), \\ &- \sum_{i,j} H_{ij} \log \hat{S}_{ij} + (1 - H_{ij}) \log(1 - \hat{S}_{ij}).\end{aligned}$$

Here, we employ the *importance reweighting* technique to build a *risk-consistent* algorithms. Specifically,

$$\begin{aligned}R(f) &= \mathbb{E}_{(X_i,X_j,H_{ij})\sim\mathcal{D}}[\ell(\langle f(X_i), f(X_j)\rangle, H_{ij})] \\ &= \int_{(x_i,x_j)} \sum_k P_{\mathcal{D}}(X_i = x_i, X_j = x_j, H_{ij} = k)\ell(\langle f(X_i), f(X_j)\rangle, H_{ij})d(x_i, x_j) \\ &= \int_{(x_i,x_j)} \sum_k P_{\mathcal{D}_\rho}(X_i, X_j, \tilde{H}_{ij} = k)\frac{P_{\mathcal{D}}(X_i, X_j, H_{ij} = k)}{P_{\mathcal{D}_\rho}(X_i, X_j, \tilde{H}_{ij} = k)}\ell(\langle f(X_i), f(X_j)\rangle, H_{ij} = k)d(x_i, x_j) \\ &= \int_{(x_i,x_j)} \sum_k P_{\mathcal{D}_\rho}(X_i, X_j, \tilde{H}_{ij} = k)\frac{P_{\mathcal{D}}(H_{ij} = k|X_i, X_j)}{P_{\mathcal{D}_\rho}(\tilde{H}_{ij} = k|X_i, X_j)}\ell(\langle f(X_i), f(X_j)\rangle, H_{ij} = k)d(x_i, x_j) \\ &= \mathbb{E}_{(X_i,X_j,\tilde{H}_{ij})\sim\mathcal{D}_\rho}[\tilde{\ell}(\langle f(X_i), f(X_j)\rangle, \tilde{H}_{ij})],\end{aligned}$$

where $\mathcal{D}$ denotes the distribution of clean data; $\mathcal{D}$ denotes the distribution of noisy data, and

$$\tilde{\ell}(\langle f(X_i), f(X_j)\rangle, \tilde{H}_{ij}) = \frac{P_{\mathcal{D}}(H_{ij} = \tilde{H}_{ij}|X_i, X_j)}{P_{\mathcal{D}_\rho}(\tilde{H}_{ij}|X_i, X_j)}\ell(\langle f(X_i), f(X_j)\rangle, \tilde{H}_{ij}).$$

Empirically, as shown in Figrue A.1, we use $\hat{S}_{ij} = f(X_i)^\top f(X_j)$ to measure the similarity of two points in a pair. $P(H_{ij} = 1|X_i, X_j)$ and $P(H_{ij} = 0|X_i, X_j)$ are approximated by $\hat{S}_{ij}$ and $1-\hat{S}_{ij}$, respectively. Then $P\left(\tilde{H}_{ij} \mid X_i, X_j\right)$ can be approximated according to $P\left(\tilde{H}_{ij} \mid X_i, X_j\right) = T_s^\top P(H_{ij} \mid X_i, X_j)$. Thus a risk-consistent estimator can be built:

$$R_n(f) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha\ell(\langle f(X_i), f(X_j)\rangle, \tilde{H}_{ij}),$$

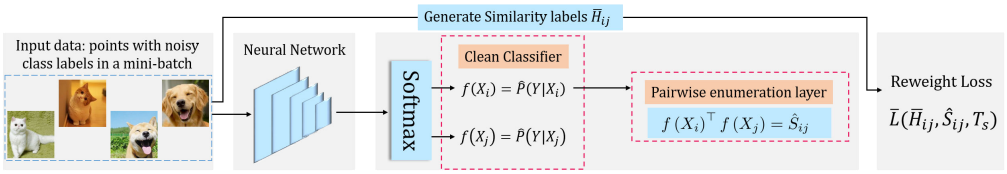


Figure A.1: Pipeline of Class2Simi with *Reweight*.

where

$$\alpha = \left\{ \tilde{H}_{ij} \frac{\hat{S}_{ij}}{T_{s,11}\hat{S}_{ij} + T_{s,01}(1-\hat{S}_{ij})} + (1-\tilde{H}_{ij}) \frac{1-\hat{S}_{ij}}{T_{s,10}\hat{S}_{ij} + T_{s,00}(1-\hat{S}_{ij})} \right\}.$$

A.5 Proof of Theorem 3

Theorem A.5.1. *Assume the parameter matrices $W_1, \ldots, W_d$ have Frobenius norm at most $M_1, \ldots, M_d$, and the activation functions are 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Assume the transition matrix is given, and the instances X are upper bounded by B, i.e., $\|X\| \leq B$ for all X, and the loss function ℓ is upper bounded by M. Then, for any $\delta > 0$, with probability at least $1-\delta$,*

$$R(\hat{f}) - R_n(\hat{f}) \leq \frac{(T_{s,11} - T_{s,01}) 2Bc(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d} M_i}{T_{s,11}\sqrt{n}} + M\sqrt{\frac{\log 1/\delta}{2n}}. \tag{A.3}$$

Proof. We have defined

$$R(f) = \mathbb{E}_{(X_i, X_j, \tilde{Y}_i, \tilde{Y}_j, \tilde{H}_{ij}, T_s) \sim \mathcal{D}_\rho} [\ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij})], \tag{A.4}$$

and

$$R_n(f) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij}), \tag{A.5}$$

where n is training sample size of the noisy data.

First, we bound the generalization error with Rademacher complexity [8].

Theorem A.5.2 ([8]). *Let the loss function be upper bounded by M. Then, for any $\delta > 0$, with the probability $1 - \delta$, we have*

$$\sup_{f \in \mathcal{F}} |R(f) - R_n(f)| \leq 2\mathfrak{R}_n(\ell \circ \mathcal{F}) + M\sqrt{\frac{\log 1/\delta}{2n}}, \tag{A.6}$$

where $\mathfrak{R}_n(\ell \circ \mathcal{F})$ is the Rademacher complexity defined by

$$\mathfrak{R}_n(\ell \circ \mathcal{F}) = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij})\right], \tag{A.7}$$

and $\{\sigma_1, \cdots, \sigma_n\}$ are Rademacher variables uniformly distributed from $\{-1, 1\}$.

Before further upper bound the Rademacher complexity $\mathfrak{R}_n(\ell \circ \mathcal{F})$, we discuss the special loss function and its *Lipschitz continuity* w.r.t $h_k(X_i), k = \{1, \ldots, c\}$.

Lemma A.5.1. *Given similarity transition matrix T_s, the loss function $\ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij})$ is μ-Lipschitz with respect to $h_k(X_i), k = \{1, \ldots, c\}$, and $\mu = (T_{s,11} - T_{s,01})/T_{s,11}$*

$$\left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij})}{\partial h_k(X_i)}\right| < \frac{T_{s,11} - T_{s,01}}{T_{s,11}}. \tag{A.8}$$

Detailed proof of Lemma A.5.1 can be found in Section A.5.1.

Lemma A.5.1 shows that the loss function is μ-Lipschitz with respect to $h_k(X_i), k = \{1, \ldots, c\}$.

Based on Lemma A.5.1, we can further upper bound the Rademacher complexity $\mathfrak{R}_n(\ell \circ \mathcal{F})$ by the following lemma.

Lemma A.5.2. *Given similarity transition matrix T_s and assume that loss function $\ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij})$ is μ-Lipschitz with respect to $h_k(X_i), k =$*

$\{1, \ldots, c\}$, *we have*

$$\begin{aligned}\mathfrak{R}_n(\ell \circ \mathcal{F}) &= \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij})\right] \\ &\leq \mu c \mathbb{E}\left[\sup_{h\in H} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h(X_i)\right], \end{aligned} \tag{A.9}$$

where H is the function class induced by the deep neural network.

Detailed proof of Lemma A.5.2 can be found in Section A.5.2.

The right-hand side of the above inequality, indicating the hypothesis complexity of deep neural networks and bounding the Rademacher complexity, can be bounded by the following theorem.

Theorem A.5.3. *[31] Assume the Frobenius norm of the weight matrices $W_1, \ldots, W_d$ are at most $M_1, \ldots, M_d$. Let the activation functions be 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Let X is upper bounded by B, i.e., for any X, $\|X\| \leq B$. Then,*

$$\mathbb{E}\left[\sup_{h\in H} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h(X_i)\right] \leq \frac{B(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d} M_i}{\sqrt{n}}. \tag{A.10}$$

Combining Lemma A.5.1,A.5.2, and Theorem A.5.2, A.5.3, Theorem A.5.1 is proved. □

A.5.1 Proof of Lemma 1

Recall that

$$\begin{aligned}&\ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij} = 1) \\ &= -\log(\hat{\tilde{S}}_{ij}) \\ &= -\log(\hat{S}_{ij} \times T_{s,11} + (1-\hat{S}_{ij}) \times T_{s,01}) \\ &= -\log(f(X_i)^\top f(X_j) \times T_{s,11} + (1 - f(X_i)^\top f(X_j)) \times T_{s,01}), \end{aligned} \tag{A.11}$$

where

$$f(X_i) = [f_1(X_i), \ldots, f_c(X_i)]^\top = \left[\left(\frac{\exp(h_1(X))}{\sum_{k=1}^{c}\exp(h_k(X))}\right), \ldots, \left(\frac{\exp(h_c(X))}{\sum_{k=1}^{c}\exp(h_k(X))}\right)\right]^\top. \quad \text{(A.12)}$$

Take the derivative of $\ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij} = 1)$ w.r.t. $h_k(X_i)$, we have

$$\frac{\partial \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij} = 1)}{\partial h_k(X_i)} = \frac{\partial \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij} = 1)}{\partial \hat{\tilde{S}}_{ij}} \big[\frac{\partial f(X_i)}{\partial h_k(X_i)}\big]^\top \frac{\partial \hat{\tilde{S}}_{ij}}{\partial f(X_i)},$$

where

$$\frac{\partial \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij} = 1)}{\partial \hat{\tilde{S}}_{ij}} = -\frac{1}{f(X_i)^\top f(X_j) \times T_{s,11} + (1 - f(X_i)^\top f(X_j)) \times T_{s,01}},$$

$$\frac{\partial \hat{\tilde{S}}_{ij}}{\partial f(X_i)} = f(X_j) \times T_{s,11} - f(X_j) \times T_{s,01},$$

$$\frac{\partial f(X_i)}{\partial h_k(X_i)} = f'(X_i) = [f'_1(X_i), \ldots, f'_c(X_i)]^\top.$$

Note that the derivative of the softmax function has some properties, i.e., if $m \neq k$, $f'_m(X_i) = -f_m(X_i)f_k(X_i)$ and if $m = k$, $f'_k(X_i) = (1 - f_k(X_i))f_k(X_i)$.

We denote by $Vector_m$ the $m-th$ element in $Vector$ for those complex vectors. Because $0 < f_m(X_i) < 1, \forall m \in \{1, \ldots, c\}$, we have

$$f'_m(X_i) \leq |f'_m(X_i)| < f_m(X_i), \qquad \forall m \in \{1, \ldots, c\}; \quad \text{(A.13)}$$

$$f'(X_i)^\top f(X_j) < f(X_i)^\top f(X_j). \quad \text{(A.14)}$$

Therefore,

$$\begin{aligned}
&\left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij}=1)}{\partial h_k(X_i)}\right| \\
&= \left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij}=1)}{\partial \hat{\tilde{S}}_{ij}} \left[\frac{\partial f(X_i)}{\partial h_k(X_i)}\right]^\top \frac{\partial \hat{\tilde{S}}_{ij}}{\partial f(X_i)}\right| \\
&= \left|\frac{f'(X_i)^\top f(X_j) \times T_{s,11} - f'(X_i)^\top f(X_j) \times T_{s,01}}{f(X_i)^\top f(X_j) \times T_{s,11} + (1 - f(X_i)^\top f(X_j)) \times T_{s,01}}\right| \\
&< \left|\frac{f(X_i)^\top f(X_j) \times T_{s,11} - f(X_i)^\top f(X_j) \times T_{s,01}}{f(X_i)^\top f(X_j) \times T_{s,11} + (1 - f(X_i)^\top f(X_j)) \times T_{s,01}}\right| \\
&< \left|\frac{T_{s,11} - T_{s,01}}{T_{s,11}}\right| \\
&= \frac{T_{s,11} - T_{s,01}}{T_{s,11}}.
\end{aligned} \tag{A.15}$$

The second inequality holds because of $T_{s,11} > T_{s,01}$ (Detailed proof can be found in Section A.5.1) and Equation (20). The third inequality holds because of $f(X_i)^\top f(X_j) < 1$.

Similarly, we can prove

$$\left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij}=0)}{\partial h_k(X_i)}\right| < \frac{T_{s,11} - T_{s,01}}{T_{s,11}}. \tag{A.16}$$

Combining Equation (A.15) and Equation (A.16), we obtain

$$\left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \tilde{H}_{ij})}{\partial h_k(X_i)}\right| < \frac{T_{s,11} - T_{s,01}}{T_{s,11}}. \tag{A.17}$$

Proof of $T_{s,11} > T_{s,01}$

As we mentioned in Section A.3, we have,

$$
\begin{aligned}
N_{00} &= n^2 \sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}, \qquad N_{01} = n^2 \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'}, \\
N_{10} &= n^2 \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'}, \qquad N_{11} = n^2 \sum_{i = i', j = j'} T_{c,ij} T_{c,i'j'}, \\
T_{s,01} &= \frac{N_{01}}{N_{00} + N_{01}}, \qquad T_{s,11} = \frac{N_{11}}{N_{10} + N_{11}}, \\
T_{s,11} - T_{s,01} &= \frac{N_{11}N_{00} + N_{11}N_{01} - N_{01}N_{10} - N_{01}N_{11}}{(N_{00} + N_{01})(N_{10} + N_{11})}.
\end{aligned}
$$

Let us review the definition of similarity labels: if two instances belong to the same class, they will have similarity label $S = 1$, otherwise $S = 0$. That is to say, for a k-class dataset, only $\frac{1}{k}$ of similarity data has similarity labels $S = 1$, and the rest $1 - \frac{1}{k}$ has similarity labels $S = 0$. We denote the number of data with similarity labels $S = 1$ by N_1, otherwise N_0. Therefore, for the balanced dataset with n samples of each class, $N_1 = cn^2$, and $N_0 = c(c-1)n^2$. Let $A = T_{s,11} - T_{s,01}$, we have

$$
\begin{aligned}
A &= N_{11}N_{00} - N_{01}N_{10} \\
&= N_{11}N_{00} - (N_0 - N_{00})(N_1 - N_{11}) \\
&= N_{11}N_{00} - N_0N_1 - N_{11}N_{00} + N_{11}N_0 + N_1N_{00} \\
&= N_{11}N_0 - N_{01}N_1 \\
&= c(c-1)n^2N_{11} - cn^2N_{01} \\
&> 0.
\end{aligned}
$$

The last equation holds because of $(c-1)N_{11} - N_{01} > 0$ according to the rearrangement inequality [39].

A.5.2 Proof of Lemma 2

$$
\begin{aligned}
&\mathbb{E}\left[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell(f(X_i),f(X_j),T_s,\tilde{H}_{ij})\right]\\
&=\mathbb{E}\left[\sup_{g}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell(f(X_i),f(X_j),T_s,\tilde{H}_{ij})\right]\\
&=\mathbb{E}\left[\sup_{\arg\max\{h_1,\ldots,h_c\}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell(f(X_i),f(X_j),T_s,\tilde{H}_{ij})\right]\\
&=\mathbb{E}\left[\sup_{\max\{h_1,\ldots,h_c\}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell(f(X_i),f(X_j),T_s,\tilde{H}_{ij})\right]\\
&\leq\mathbb{E}\left[\sum_{k=1}^{c}\sup_{h_k\in H}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell(f(X_i),f(X_j),T_s,\tilde{H}_{ij})\right]\\
&=\sum_{k=1}^{c}\mathbb{E}\left[\sup_{h_k\in H}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\ell(f(X_i),f(X_j),T_s,\tilde{H}_{ij})\right]\\
&\leq\mu c\mathbb{E}\left[\sup_{h_k\in H}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h_k(X_i)\right]\\
&=\mu c\mathbb{E}\left[\sup_{h\in H}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h(X_i)\right],
\end{aligned}
$$

where the first three equations hold because given T_s, f and $\max\{h_1,\ldots,h_c\}$ give the same constraint on $h_j(X_i), j=\{1,\ldots,c\}$; the sixth inequality holds because of the Talagrand Contraction Lemma [59].

A.6 Further Details on Experiments

A.6.1 Network Structure and Optimization

Note that for *CIFAR-10*, we use ResNet-26 with shake-shake regularization [30] **except** the experiment on noisy T_c in Figure 4, where we use ResNet-32 with pre-activation [41] for shorter training time. In stage 1, We use the same optimization method as *Forward* to learn the transition matrix $\hat{T}_c$. In stage 2, we use Adam optimizer with an initial learning rate 0.001. On *MNIST*, the batch size is 128 and the learning rate decays every 5 epochs by a factor of 0.1 with 30 epochs in total. On *CIFAR-10*, the

batch size is 512 and the learning rate decays every 40 epochs by a factor of 0.1 with 200 epochs in total. On *CIFAR-100*, the batch size is 512 and the learning rate decays every 40 epochs by a factor of 0.1 with 120 epochs in total. On *News20*, the batch size is 128 and the learning rate decays every 5 epochs by a factor of 0.1 with 30 epochs in total. On *Clothing1M**, the batch size is 32 and the learning rate drops every 5 epochs by a factor of 0.1 with 10 epochs in total.

A.6.2 Symmetric and Asymmetric Noise Settings

Symmetric noise setting is defined as follow, where c is the number of classes.

$$\text{Sym-}\rho\text{:}\quad T = \begin{bmatrix} 1-\rho & \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} & \frac{\rho}{C-1} \\ \frac{\rho}{C-1} & 1-\rho & \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} \\ \vdots & & \ddots & & \vdots \\ \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} & 1-\rho & \frac{\rho}{C-1} \\ \frac{\rho}{C-1} & \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} & 1-\rho \end{bmatrix}. \tag{A.18}$$

The asymmetric noise setting is set as follow,

Listing A.1: Asymmetric noise (transition matrix) generation.

```
def AsymTransitionMatrixGenerate(NoiseRate=0.3,
    NumClasses=10, seed=1):
    np.random.seed(seed)
    t = np.random.rand(NumClasses, NumClasses)
    i = np.eye(NumClasses)
    t = t + Coef * NumClasses * i
    for a in range(NumClasses):
    t[a] = t[a] / t[a].sum()
return t
```

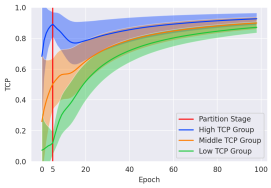
Coef is set to 1.70, 1.20, 0.60, 0.24 at the rate 0.2, 0.3, 0.4, 0.6, respectively.

Appendix B

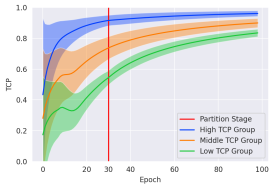
Supplementary for Chapter 4

B.1 More Empirical Studies

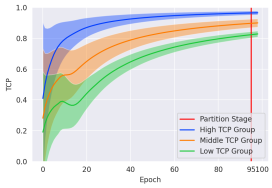
B.1.1 More Empirical Study regarding Figure. 1



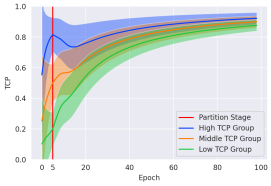
(a) Partitioned at a start stage.



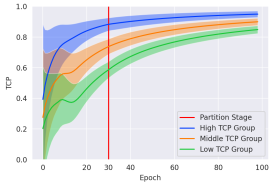
(b) Partitioned at an early stage.



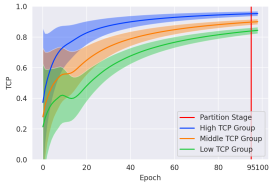
(c) Partitioned at an end stage.



(d) Partitioned at a start stage.



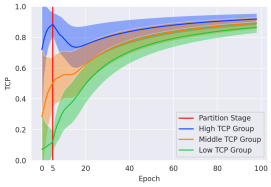
(e) Partitioned at an early stage.



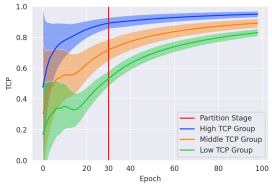
(f) Partitioned at an end stage.

Figure B.1: TCP (mean and std.) of three groups partitioned by TCP calculated at the start stage (epoch 5), early stage (epoch 30), and end stage (epoch 95) during training a ResNet50 on CIFAR100 with IDN-0.4 for 100 epochs. The first row is partitioned by high TCP (10%), middle TCP (80%), and low TCP (10%). The second row is partitioned by high TCP (20%), middle TCP (60%), and low TCP (20%).

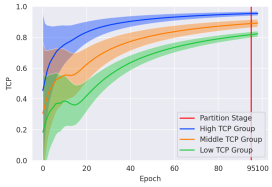
Figures B.1 and B.2 show the results on more practical dataset CIFAR100 and IDN noise 0.4 higher IDN noise 0.6 with different partitions



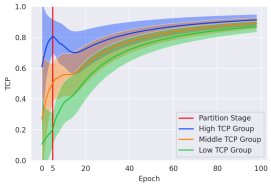
(a) Partitioned at a start stage.



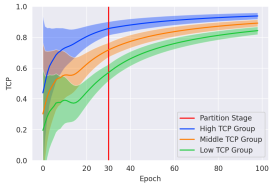
(b) Partitioned at an early stage.



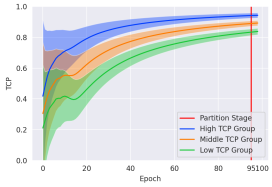
(c) Partitioned at an end stage.



(d) Partitioned at a start stage.



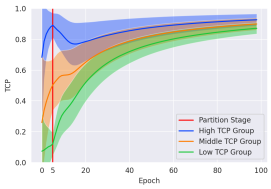
(e) Partitioned at an early stage.



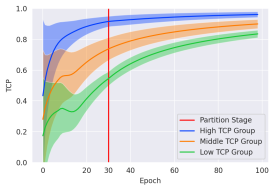
(f) Partitioned at an end stage.

Figure B.2: TCP (mean and std.) of three groups partitioned by TCP calculated at the start stage (epoch 5), early stage (epoch 30), and end stage (epoch 95) during training a ResNet50 on CIFAR100 with IDN-0.6 for 100 epochs. The first row is partitioned by high TCP (10%), middle TCP (80%), and low TCP (10%). The second row is partitioned by high TCP (20%), middle TCP (60%), and low TCP (20%).

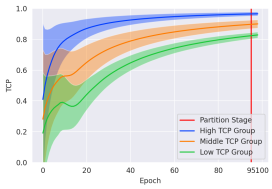
(2:6:2), which demonstrate that our claims hold in general and do not change sensitively with the partition ratios.



(a) Partitioned at a start stage.



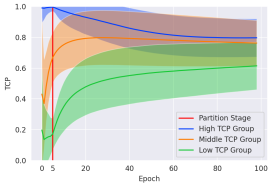
(b) Partitioned at an early stage.



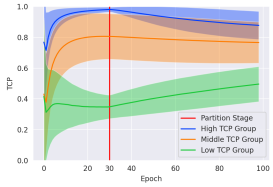
(c) Partitioned at an end stage.

Figure B.3: TCP (mean and std.) of three groups (high TCP (10%), middle TCP (80%), and low TCP (10%)) partitioned by the TCP calculated at the start stage (epoch 5), early stage (epoch 30), and end stage (epoch 95) during training a ResNet50 on CIFAR100 with IDN-0.4 for 100 epochs.

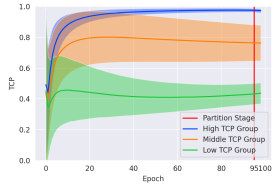
Figures B.3 and B.4 show the results on datasets CIFAR100 and SVHN with different model architectures. Conclusions from Section 3 are based on the memorization effect of overparameterized DNNs. Therefore,



(a) Partitioned at a start stage.

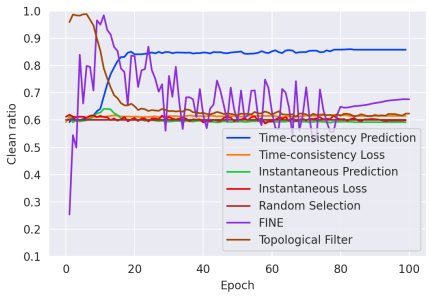


(b) Partitioned at an early stage.

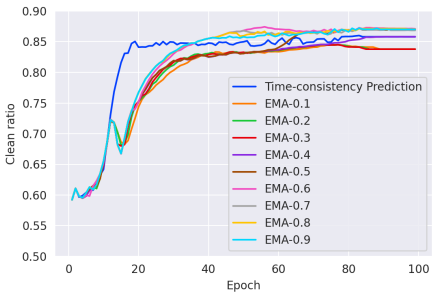


(c) Partitioned at an end stage.

Figure B.4: TCP (mean and std.) of three groups (high TCP (10%), middle TCP (80%), and low TCP (10%)) partitioned by the TCP calculated at the start stage (epoch 5), early stage (epoch 30), and end stage (epoch 95) during training a AlexNet on SVHN with IDN-0.4 for 100 epochs.



(a) Clean ratios of the selected top 5000 instances ranked by six kinds of instance hardness measures.



(b) Clean ratios of the selected top 5000 instances ranked by EMA with fixed discount factors.

Figure B.5: Clean ratios of the selected top 5000 instances ranked by different instance hardness measures, respectively. The clean ratio of randomly selected instances is 0.6 since the noise rate is 0.4.

they hold better for deeper DNNs (ResNet50) than the shallower DNNs (AlexNet). Moreover, for AlexNet on SVHN, the high TCP group partitioned at an early stage has no overlap with the middle TCP group. Overall, the results demonstrate that our conclusions in the paper hold true and generalize to other architectures and datasets.

B.1.2 Comparison with SOTA Selection Methods

Following the same setting as Figure 2, we select 5,000 confident examples at every epoch t according to six types of selection criterion, i.e., instantaneous prediction $\mathrm{InP}_t(x)$, instantaneous loss $\ell(x)$, time-consistency

of prediction $\mathrm{TCP}_{\mathrm{t}}(x)$, time-consistency of loss, and two SOTA confident sample selection methods: FINE [51] and Topological Filter [118]. Besides, we replace $\frac{t}{t+1}$ in Eq. (1) with fixed discount factors λ and name the corresponding measure as $EMA-\lambda$. Then we count the number of instances with clean labels and calculate the clean ratios. As shown in Figure B.5, at the starting stage, when the model just learns the clean data while has not fit the noisy data, FINE and Topological Filter perform well. As the training goes and the model fits the noisy data, our method achieves the best selection clean ratio. Compared with fixed discount EMA measures, our method achieves the best AUC (Area Under Curve).

B.2 Analysis on Introducing Instances with Pseudo Labels

Consider the situation we have a labeled set $\mathcal{L}$ (in practice it can be the selected confident examples set) and one unlabeled instance x'. By training on $\mathcal{L}$ for one step, we have

$$\theta_{t+1}=\theta_t-\eta\sum_{x\in\mathcal{L}}\nabla_\theta\ell\left(x;\theta_t\right),$$

and by training on $\mathcal{L}$ and x' for one step, we have

$$\theta'_{t+1}=\theta_t-\eta\left(\sum_{x\in\mathcal{L}}\nabla_\theta\ell\left(x;\theta_t\right)+\nabla_\theta\ell\left(x';\theta_t\right)\right),$$

where θ_t denotes the network parameters at step t and η denotes the learning rate.

The Taylor expansion of loss $\ell\left(x;\theta\right)$ at the point $\theta=\theta_0$ is:

$$g_{\theta_0}\left(\theta\right)=\left[\sum_{x\in\mathcal{L}}\ell\left(x;\theta_0\right)+\nabla_\theta\ell\left(x;\theta_0\right)\left(\theta-\theta_0\right)\right]+o\left(\left(\theta-\theta_0\right)^2\right). \tag{B.1}$$

Then we evaluate the forgetting effect of introducing instances x' with its pseudo label to the training set by checking the change of loss over the labeled set $\mathcal{L}$ with x' added or not. If adding x' does not cause a vital

change, we can conclude that it does not lead to catastrophic forgetting of the learned examples with correct labels. Therefore, we calculate the change of loss over the labeled set by

$$\begin{aligned}\frac{1}{\eta}\left|\sum_{x\in\mathcal{L}}\left[\ell\left(x;\theta_{t+1}\right)-\ell\left(x;\hat{\theta}_{t+1}\right)\right]\right| =& \frac{1}{\eta}\left|g_{\theta_t}\left(\theta_{t+1}\right)-g_{\theta_t}\left(\hat{\theta}_{t+1}\right)\right| \\ \approx& \left|\nabla_\theta \ell\left(x';\theta_t\right)\sum_{x\in\mathcal{L}}\nabla_\theta \ell\left(x;\theta_t\right)\right| \\ =& \left|\frac{\partial \ell\left(x';\theta_t\right)}{\partial \theta_t}\frac{\partial \theta_t}{\partial t}\right| \\ =& \left|\frac{\partial \ell\left(x';\theta_t\right)}{\partial t}\right| \\ =& \left|\frac{\partial \ell\left(x';\theta_t\right)}{\partial p_t^{\hat{y}'_t}\left(x'\right)}\frac{\partial p_t^{\hat{y}'_t}\left(x'\right)}{\partial t}\right|,\end{aligned}$$

where $p_t^{\hat{y}'_t}(x')$ is the probability of x' belonging to $\hat{y}'_t$ at step t, and $\hat{y}'_t$ is the prediction (pseudo label) of x' at step t. The second line holds because we omit the second and higher order terms of the Taylor expansion in Eq (B.1). Then, with cross-entropy loss employed, we have

$$\frac{\partial \ell\left(x';\theta_t\right)}{\partial p_t^{\hat{y}'_t}\left(x'\right)}=\frac{\partial \log(p_t^{\hat{y}'_t}\left(x'\right))}{\partial p_t^{\hat{y}'_t}\left(x'\right)}=-\frac{1}{p_t^{\hat{y}'_t}\left(x'\right)}.$$

Next, by using $\left(p_{t+1}^{\hat{y}'_t}(x')-p_t^{\hat{y}'_t}(x')\right)$ to approximate $\frac{\partial p_t^{\hat{y}'_t}(x')}{\partial t}$, we have

$$\frac{1}{\eta}\left|\sum_{x\in\mathcal{L}}\left[\ell\left(x;\theta_{t+1}\right)-\ell\left(x;\theta'_{t+1}\right)\right]\right| \approx \left|\frac{p_{t+1}^{\hat{y}'_t}(x')}{p_t^{\hat{y}'_t}(x')}-1\right|. \tag{B.2}$$

Since x' is selected with high clean-TCP, $p_t^{\hat{y}'_t}(x')$ is very close to $p_{t+1}^{\hat{y}'_t}(x')$ because it has been verified in Figure 1 that instances with high clean-TCP in the early stage maintain their high clean-TCP in the future, which means

the loss change can be bounded with a very small value. Therefore, exploiting high clean-TCP instances with pseudo labels helps to correct corrupted labels and learn a clean classifier without causing catastrophic forgetting of the learned examples with correct labels.

Appendix C

Supplementary for Chapter 5

C.1 Derivation of Getting $T_y(a)$ from $T_y(x)$

Here we take $T_{-1}(a)$ as an example. Let $X' \triangleq (Z, W)$ and $X_a \triangleq (A = a, Z, W)$:

$$\begin{aligned} T_{-1}(a) &= P(\tilde{Y} = +1 \mid Y = -1, A = a) \\ &= \frac{\int P(X' = x', \tilde{Y} = +1, Y = -1, A = a)\ \mathrm{d}x'}{\int P(X' = x', Y = -1, A = a)\ \mathrm{d}x'} \\ &= \frac{\int P(\tilde{Y} = +1 \mid X = x_a, Y = -1) P(Y = -1 \mid X = x_a) P(x_a)\ \mathrm{d}x_a}{\int P(Y = -1 \mid X = x_a,) P(x_a)\ \mathrm{d}x_a} \\ &= \frac{\int T_{-1}(x_a) P(Y = -1 \mid X = x_a) P(x_a)\ \mathrm{d}x_a}{\int P(Y = -1 \mid X = x_a) P(x_a)\ \mathrm{d}x_a}, \end{aligned} \tag{C.1}$$

where $P(Y = -1 \mid X = x) = \frac{P(\tilde{Y} = -1 | X = x) - T_{+1}(x)}{(1 - T_{-1}(x) - T_{+1}(x))}$. In practice, we can use the above equation to consistently estimate $T_{-1}(a)$.

C.2 Derivation of the Negative ELBO

To derive the ELBO, we start with maximizing the data likelihood $p_\Theta(A, Z, W, \tilde{Y})$:

$$\log p_\Theta(a, z, w, \tilde{y}) \tag{C.2}$$

$$= \log \int_u \int_y p_\Theta(a, z, w, \tilde{y}, y, u) \mathrm{d}y \ \mathrm{d}u \tag{C.3}$$

$$= \log \int_u \int_y \frac{p_\Theta(a, z, w, \tilde{y}, y, u)}{q_{\phi_1}(u \mid z) q_{\phi_2}(y \mid u, z)} q_{\phi_1}(u \mid z) q_{\phi_2}(y \mid u, z) \mathrm{d}y \ \mathrm{d}u \tag{C.4}$$

$$= \log \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\frac{p_\Theta(a, z, w, \tilde{y}, y, u)}{q_{\phi_1}(u \mid z) q_{\phi_2}(y \mid u, z)} \right] \tag{C.5}$$

$$\geq \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log \frac{p_\Theta(a, z, w, \tilde{y}, y, u)}{q_{\phi_1}(u \mid z) q_{\phi_2}(y \mid u, z)} \right] \triangleq \mathrm{ELBO} \qquad \text{(Jensen's Inequality)} \tag{C.6}$$

$$= \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log \frac{p(a) p(u) p_{\theta_1}(z \mid u) p_{\theta_2}(y \mid u, z) p_{\theta_3}(w \mid u, a, z, y) p_{\theta_4}(\tilde{y} \mid a, z, y, w)}{q_{\phi_1}(u \mid z) q_{\phi_2}(y \mid u, z)} \right] \tag{C.7}$$

$$= \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p_{\theta_1}(z \mid u) \right] + \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p_{\theta_3}(w \mid u, a, z, y) \right] \tag{C.8}$$

$$+ \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p_{\theta_4}(\tilde{y} \mid a, z, y, w) \right] \tag{C.9}$$

$$+ \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log \frac{p(u)}{q_{\phi_1}(u \mid z)} \right] + \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p(a) \right] \tag{C.10}$$

$$= \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p_{\theta_1}(z \mid u) \right] + \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p_{\theta_3}(w \mid u, a, z, y) \right] \tag{C.11}$$

$$+ \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p_{\theta_4}(\tilde{y} \mid a, z, y, w) \right] - \mathrm{D}_{\mathrm{KL}}(q_{\phi_1}(u \mid z) \| p(u)) + \log p(a). \tag{C.12}$$

Since $\log p(a)$ is a constant, we drop it in ELBO. Then, the final negative ELBO can be defined as:

$$-\mathrm{ELBO} \triangleq -\mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p_{\theta_1}(z \mid u) \right] - \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p_{\theta_3}(w \mid u, a, z, y) \right] \tag{C.13}$$

$$- \mathrm{E}_{(u,y)\sim q_\phi(u,y|z)} \left[\log p_{\theta_4}(\tilde{y} \mid a, z, y, w) \right] + \mathrm{D}_{\mathrm{KL}}(q_{\phi_1}(u \mid z) \| p(u)). \tag{C.14}$$

Appendix D

Supplementary for Chapter 6

D.1 Motivation for the Noise Model

The class-conditional noise model is compelling for the setting that the label of a pair of examples $(\boldsymbol{x}, \boldsymbol{x}')$ is annotated by human who can inevitably make mistakes, i.e., the corruption is in $P(S|\boldsymbol{x}, \boldsymbol{x}')$. However, this setting is not very suitable for our problem because we only collect similar data pairs. Therefore, we do not have seeming dissimilar data pairs and only modelling the corruption of $P(S|\boldsymbol{x}, \boldsymbol{x}')$ cannot solve our problem. Besides, as we discuss in the related work, the CCN model is a special case of the MCD model [76]. CCN model does not fit the MCD setting problem, though the MCD model fits the CCN setting problem conversely. Namely, our method, which is developed under the MCD model, can also solve the CCN problem.

In addition to its good generality, we employ the MCD model because CCN is a noise model for labeling noise and MCD is a noise model for sampling noise, which is why MCD is more suitable for the scenario of surveying sensitive topics with indirect questioning. Data reliability is a common concern especially when asking about sensitive topics such as sexual misconduct, or drug and alcohol abuse. Sensitive topics might cause refusals in surveys due to privacy concerns of the subjects [86]. This nonresponse reduces sample size and study power and increases bias. Various indirect questioning methods have been developed to reduce the social desirability bias and increase data reliability. Questions in the form of 'With whom do you share the same opinion on issue $\mathcal{I}$?' can be regarded as one type of

randomized response technique, which is a commonly used indirect questioning survey method [114, 29, 5]. Such questioning is to sample examples of similar data pairs from $p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}')$. Besides, due to the sensitivity of the questions, respondents might answer them in a manner that will be viewed favorably by others instead of answering truthfully [86], which makes the selected examples contain dissimilar data pairs from $p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}')$. These phenomena motivate us to employ the contamination model to describe the noisy similar data pairs, i.e., $\tilde{p}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') = (1-\rho_{\mathrm{d}})p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') + \rho_{\mathrm{d}}p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}')$.

D.2 Discussion about the Data Independence Assumption

Following [5], we assume that each instance is independently drawn from the joint distribution, i.e.,

$$\begin{aligned} p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') &= p(\boldsymbol{x}, \boldsymbol{x}' | y = y' = +1 \vee y = y' = -1) \\ &= \frac{\pi_+^2 p_+(\boldsymbol{x}) p_+(\boldsymbol{x}') + \pi_-^2 p_-(\boldsymbol{x}) p_-(\boldsymbol{x}')}{\pi_+^2 + \pi_-^2}. \end{aligned}$$

However, in real-world surveys, a given positive example might occur very frequently with a few positive examples and very in-frequently with other positive examples. Then the distribution of similar pairs is shifted from the original one. We denote the shifted distribution by $p'_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}')$ and let $p'_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') = w(\boldsymbol{x}, \boldsymbol{x}')\ p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}')$. Then we have

$$p(\boldsymbol{x}, \boldsymbol{x}') = \pi_{\mathrm{s}} p'_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') + \pi_{\mathrm{d}} p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}'), \tag{D.1}$$

$$\tilde{p}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') = (1-\rho_{\mathrm{d}}) p'_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') + \rho_{\mathrm{d}} p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}'), \tag{D.2}$$

$$p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1-\pi_{\mathrm{s}}}{(1-\rho_{\mathrm{d}}-\pi_{\mathrm{s}})w(\boldsymbol{x}, \boldsymbol{x}')} \tilde{p}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') - \frac{\rho_{\mathrm{d}}}{(1-\rho_{\mathrm{d}}-\pi_{\mathrm{s}})w(\boldsymbol{x}, \boldsymbol{x}')} p(\boldsymbol{x}, \boldsymbol{x}'). \tag{D.3}$$

Then, by substituting Eq.(D.3) into Eq.(D.4), the original risk can be

equivalently expressed in terms of data sampled from the shifted distribution $p'_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}')$ and the unlabeled data as

$$\begin{aligned}\mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim p}[\ell(f(\boldsymbol{x}), y)] = &\frac{\pi_{\mathrm{s}}(1-\pi_{\mathrm{s}})}{(1-\rho_{\mathrm{d}}-\pi_{\mathrm{s}})(2\pi_+ - 1)} \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{x}')\sim \tilde{p}_{\mathrm{s}}} \left[\frac{\tilde{l}(\boldsymbol{x}) + \tilde{l}(\boldsymbol{x}')}{2w(\boldsymbol{x}, \boldsymbol{x}')}\right] \\ &- \frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}}}{(1-\rho_{\mathrm{d}}-\pi_{\mathrm{s}})(2\pi_+ - 1)} \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{x}')\sim p} \left[\frac{\tilde{l}(\boldsymbol{x}) + \tilde{l}(\boldsymbol{x}')}{2w(\boldsymbol{x}, \boldsymbol{x}')}\right] \\ &- \frac{\pi_-}{(2\pi_+ - 1)} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}[\ell(f(\boldsymbol{x}), +1)] \\ &+ \frac{\pi_+}{(2\pi_+ - 1)} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}[\ell(f(\boldsymbol{x}), -1)]\,.\end{aligned}$$

Therefore, given the weight $w(\boldsymbol{x}, \boldsymbol{x}')$, we can design an unbiased risk estimator according to the above equation. To validate this setting experimentally, we assign $w(\boldsymbol{x}, \boldsymbol{x}')$ to every pair, and the new data is sampled from $p'_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}')$. We compare the weighted-nSU (wnSU-DC) with the original nSU-DC and SU-DC on UCI and LIBSVM datasets. From Table D.1, we can see that in this shifted setting, wnSU-DC outperforms the original nSU-DC and SU-DC. The weight function $w(\boldsymbol{x}, \boldsymbol{x}')$ must be away from 0, and this gap can affect both the optimization stability and the generalization bound where $1/\min(w)$ will also be in the convergence rate. Another implication of having a weight function is that the unbiased risk estimator should be no longer very good to use in practice because we need to round up too small w which leads to a benign bias (benign in both optimization stability and generalization bound).

Table D.1: Means and Standard Deviations (Percentage) of Classification Accuracy on the UCI and LIBSVM datasets with $\{\pi_+ = 0.7, \rho_{\mathrm{d}} = 0.2\}$ and $\{\pi_+ = 0.8, \rho_{\mathrm{d}} = 0.1\}$.

Dataset	australian	breast-cancer	fourclass	magic	cod-rna	adult	banknote	heart	svmguide1	htru_2
{0.7, 0.2}										
wnSU-DC	63.4±9.2	93.6±4.7	73.6±8.4	66.2±1.1	78.6±8.8	58.6±6.4	81.2±5.8	72.2±2.6	64.4±10.4	90.8±4.0
nSU-DC	62.9±8.7	82.9±4.3	72.6±8.3	64.2±5.2	74.0±7.0	58.7±7.1	67.6±4.6	70.4±10.5	64.1±8.0	86.8±7.0
SU-DC	55.7±4.8	88.7±7.6	71.0±5.7	64.1±4.4	72.9±6.6	63.8±4.5	68.1±3.1	63.0±5.2	54.9±3.9	90.0±0.7
{0.8, 0.1}										
wnSU-DC	61.4±8.6	90.1±9.6	73.2±6.2	68.1±4.2	78.2±4.0	73.8±2.3	75.8±9.9	71.1±8.0	65.0±12.3	93.7±2.3
nSU-DC	60.9±5.2	71.0±6.2	72.6±6.6	64.7±4.8	77.4±10.2	72.9±3.3	75.1±9.1	65.9±4.1	60.6±4.4	85.9±7.7
SU-DC	58.0±4.5	81.4±12.8	72.6±9.6	66.6±1.8	73.9±8.1	75.0±0.9	75.1±6.4	61.5±4.2	61.3±3.5	85.5±5.1

D.3 Proof of Lemma 1

Proof. Since $\boldsymbol{x}$ and $\boldsymbol{x}'$ are drawn independently, we have

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{x}') &= p(\boldsymbol{x})p(\boldsymbol{x}') \\ &= \pi_+^2 p_+(\boldsymbol{x})p_+(\boldsymbol{x}') + \pi_-^2 p_-(\boldsymbol{x})p_-(\boldsymbol{x}') \\ &\quad + \pi_+\pi_- p_+(\boldsymbol{x})p_-(\boldsymbol{x}') + \pi_+\pi_- p_-(\boldsymbol{x})p_+(\boldsymbol{x}') \\ &= \pi_{\mathrm{s}} p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') + \pi_{\mathrm{d}} p_{\mathrm{d}}(\boldsymbol{x}, \boldsymbol{x}'). \end{aligned}$$

□

D.4 Proof and Discussion about Theorem 1

Proof. To begin with, we introduce the following lemma:

Lemma D.4.1 ([5])**.** *The classification risk* (6.3) *can be equivalently expressed as*

$$R_{\mathrm{SU}}(f) = \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{x}')\sim p_{\mathrm{s}}} [\mathcal{L}_{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{x}')] + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p} [\mathcal{L}'_{\mathrm{U}}(\boldsymbol{x})], \tag{D.4}$$

where

$$\begin{aligned} \mathcal{L}_{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{x}') &= \frac{\pi_{\mathrm{s}}}{2(2\pi_+ - 1)}[\tilde{l}(\boldsymbol{x}) + \tilde{l}(\boldsymbol{x}')], \\ \tilde{l}(\boldsymbol{x}) &= \ell(f(\boldsymbol{x}), +1) - \ell(f(\boldsymbol{x}), -1), \\ \mathcal{L}'_{\mathrm{U}}(\boldsymbol{x}) &= \frac{-\pi_-}{2\pi_+ - 1}\ell(f(\boldsymbol{x}), +1) + \frac{\pi_+}{2\pi_+ - 1}\ell(f(\boldsymbol{x}), -1). \end{aligned}$$

Combining Eq. (6.8) and Eq. (6.9) and organizing, we have

$$p_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1 - \pi_{\mathrm{s}}}{1 - \rho_{\mathrm{d}} - \pi_{\mathrm{s}}}\tilde{p}_{\mathrm{s}}(\boldsymbol{x}, \boldsymbol{x}') - \frac{\rho_{\mathrm{d}}}{1 - \rho_{\mathrm{d}} - \pi_{\mathrm{s}}}p(\boldsymbol{x}, \boldsymbol{x}'). \tag{D.5}$$

Substituting Eq.(D.5) into Eq.(D.4) we have

$$
\begin{aligned}
R(f) &= R_{\mathrm{SU}}(f) \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{x}')\sim\tilde{p}_{\mathrm{s}}}\left[\frac{\pi_{\mathrm{s}}(1-\pi_{\mathrm{s}})[\ell(f(\boldsymbol{x}),+1)-\ell(f(\boldsymbol{x}),-1)+\ell(f(\boldsymbol{x}'),+1)-\ell(f(\boldsymbol{x}'),-1)]}{2(1-\rho_{\mathrm{d}}-\pi_{\mathrm{s}})(2\pi_{+}-1)}\right] \\
&\quad + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}\left[\frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}}-\pi_{-}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}\ell(f(\boldsymbol{x}),+1)-\frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}}-\pi_{+}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}\ell(f(\boldsymbol{x}),-1)\right] \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{x}')\sim\tilde{p}_{\mathrm{s}}}\left[\frac{\pi_{\mathrm{s}}(1-\pi_{\mathrm{s}})}{2(1-\rho_{\mathrm{d}}-\pi_{\mathrm{s}})(2\pi_{+}-1)}[\tilde{l}(\boldsymbol{x})+\tilde{l}(\boldsymbol{x}')]\right] \\
&\quad + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}\left[\frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}}-\pi_{-}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}\ell(f(\boldsymbol{x}),+1)-\frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}}-\pi_{+}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}\ell(f(\boldsymbol{x}),-1)\right] \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{x}')\sim\tilde{p}_{\mathrm{s}}}[\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x},\boldsymbol{x}')] + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}[\mathcal{L}_{\mathrm{U}}(\boldsymbol{x})] \\
&= R_{\mathrm{nSU}}(f),
\end{aligned}
$$

where

$$
\begin{aligned}
\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x},\boldsymbol{x}') &= \frac{\pi_{\mathrm{s}}(1-\pi_{\mathrm{s}})}{2(1-\rho_{\mathrm{d}}-\pi_{\mathrm{s}})(2\pi_{+}-1)}[\tilde{l}(\boldsymbol{x})+\tilde{l}(\boldsymbol{x}')], \\
\tilde{l}(\boldsymbol{x}) &= \ell(f(\boldsymbol{x}),+1)-\ell(f(\boldsymbol{x}),-1), \\
\mathcal{L}_{\mathrm{U}}(\boldsymbol{x}) &= \frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}}-\pi_{-}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}\ell(f(\boldsymbol{x}),+1) \\
&\quad - \frac{\pi_{\mathrm{s}}\rho_{\mathrm{d}}-\pi_{+}(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)}{(\rho_{\mathrm{d}}+\pi_{\mathrm{s}}-1)(2\pi_{+}-1)}\ell(f(\boldsymbol{x}),-1).
\end{aligned}
$$

□

D.4.1 Discussion about $\pi_{+} \neq 0.5$

The direct cause of this requirement $\pi_{+} \neq 0.5$ is that the denominator of the risk contains the term $(2\pi_{+}-1)$, which cannot be zero. The intuitive and ultimate cause is that if $\pi_{+} = 0.5$, the marginal distributions of similarity data and unlabeled data are the same, i.e., $p_{\mathrm{s}}(\boldsymbol{x}) = p(\boldsymbol{x}) = 0.5p_{+}(\boldsymbol{x}) + 0.5p_{-}(\boldsymbol{x})$, but at least 2 marginals with different class priors are required to make comparison and extract contrastive information to make the prediction [66]. Technically, if $\pi_{+} = 0.5$, those terms in the original risk expression, e.g., $\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{x}'\sim p_{+}}\left[\frac{\ell(f(\boldsymbol{x}),+1)+\ell(f(\boldsymbol{x}'),+1)}{2}\right]$ cannot be rewritten as a linear combination of

$\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{x}')\sim p_{\mathrm{s}}}\left[\frac{\ell(f(\boldsymbol{x}),+1)+\ell(f(\boldsymbol{x}'),+1)}{2}\right]$ and $\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}[\ell(f(\boldsymbol{x}),+1)]$.

A π_+ value close to 0.5 makes the marginal distributions of p_s and p more similar and makes p_s and p more entangled. Thus, it becomes more difficult to solve the MPE problem and thereby leads to a poor estimation of the parameters. We investigate the effect of π_+ by conducting experiments on UCI and LIBSVM datasets with $\pi_+ = [0.55, 0.6, 0.7]$ and a constant noise rate $\rho_{\mathrm{d}} = 0.2$. From Table D.2, we can see that the classification accuracy of nSU-LC decreases as the class prior approaches 0.5, indicating the parameters are poorly estimated. Most of SU-LC's results decrease while some of them get increased. The reason could be that the poorly estimated parameters accidentally are close to the true values for SU-LC. Overall, our method nSU-LC performs better than the baseline with a π_+ value close to 0.5.

Table D.2: Means and Standard Deviations (Percentage) of Classification Accuracy on the UCI and LIBSVM datasets with $\pi_+ = [0.55, 0.6, 0.7]$ and $\rho_{\mathrm{d}} = 0.2$.

Dataset	australian	breast-cancer	fourclass	magic	cod-rna	adult	banknote	heart	svmguide1	htru_2
{0.7, 0.2}										
nSU-LC	83.1±4.2	93.6±1.7	71.5±6.7	76.2±1.5	89.8±2.7	67.9±1.0	98.0±1.1	83.3±3.7	76.1±7.0	96.4±1.3
SU-LC	63.7±5.4	84.1±3.8	63.7±1.5	69.5±3.5	62.1±12.0	63.6±4.2	94.9±5.6	58.3±7.6	52.8±3.4	96.1±1.2
{0.6, 0.2}										
nSU-LC	75.1±6.1	91.3±3.4	69.9±7.5	72.4±0.8	87.7±2.4	62.3±3.1	97.3±1.4	81.5±6.8	75.1±6.3	92.5±4.6
SU-LC	59.7±5.5	85.2±3.3	62.4±0.5	69.6±3.2	61.2±3.5	59.8±3.3	93.7±7.5	53.1±2.1	54.6±6.8	95.6±0.5
{0.55, 0.2}										
nSU-LC	66.9±9.2	86.4±4.9	69.4±9.4	67.1±4.3	78.4±5.3	54.4±4.5	90.9±3.6	74.1±6.8	69.7±8.8	86.4±6.3
SU-LC	55.7±0.4	79.4±3.3	64.2±0.1	66.3±3.6	65.4±2.7	52.4±3.0	90.3±5.6	54.6±1.9	58.8±7.7	93.1±1.9

D.5 Proof and Discussion about Theorem 2

Proof. There exists a twice differentiable function $\psi : \mathbb{R} \to \mathbb{R}_+$ such that $\ell(z,t) = \psi(tz)$, because ℓ is a twice differentiable margin loss function.

Taking the derivative of

$$\begin{aligned}\hat{J}_\ell(\boldsymbol{w}) &= \frac{1}{n_{\mathrm{nS}}}\sum_{i=1}^{n_{\mathrm{nS}}}\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x}_{\mathrm{S},i},\boldsymbol{x}'_{\mathrm{S},i}) + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\mathcal{L}_{\mathrm{U}}(\boldsymbol{x}_{\mathrm{U},i}) + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ &= \frac{A}{2n_{\mathrm{nS}}}\sum_{i=1}^{2n_{\mathrm{nS}}}[\ell(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}),+1) - \ell(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}),-1)] \\ &\quad + \frac{B}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}[\ell(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),+1)] - \frac{C}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}[\ell(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),-1)] + \frac{\lambda}{2}\|\boldsymbol{w}\|^2 \\ &= \frac{\lambda}{2}\boldsymbol{w}^\top\boldsymbol{w} - \frac{A}{2n_{\mathrm{nS}}}\sum_{i=1}^{2n_{\mathrm{nS}}}\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}) \\ &\quad + \frac{B}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}[\ell(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),+1)] - \frac{C}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}[\ell(\boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),-1)],\end{aligned}$$

with respect to $\boldsymbol{w}$,

$$\begin{aligned}\frac{\partial}{\partial\boldsymbol{w}}\hat{J}_\ell(\boldsymbol{w}) &= \lambda\boldsymbol{w} - \frac{A}{2n_{\mathrm{nS}}}\sum_{i=1}^{2n_{\mathrm{nS}}}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}) \\ &\qquad + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\left\{B\frac{\partial\ell(\xi_i,+1)}{\partial\xi_i} - C\frac{\partial\ell(\xi_i,-1)}{\partial\xi_i}\right\}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),\end{aligned}$$

where $\xi_i = \boldsymbol{w}^\top\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})$.

Note that the second-order derivative of $\ell(z,t)$ with respect to z is

$$\frac{\partial^2\ell(z,t)}{\partial z^2} = \frac{\partial^2\psi(tz)}{\partial z^2} = \frac{\partial}{\partial z}\left(t\frac{\partial\psi(\xi)}{\partial\xi}\right) = t^2\frac{\partial^2\psi(\xi)}{\partial\xi^2} = \frac{\partial^2\psi(\xi)}{\partial\xi^2},$$

where $\xi = tz$ is employed in the second equality and the last equality holds because $t \in \{+1,-1\}$.

The Hessian matrix is a square matrix of second-order partial derivatives of a scalar-valued function. A twice continuously differentiable function of several variables is convex on a convex set if and only if its Hessian matrix of second partial derivatives is positive semidefinite on the interior

of the convex set. For $\hat{J}_\ell$, its Hessian matrix is

$$\begin{aligned}
\boldsymbol{H}\hat{J}_\ell(\boldsymbol{w}) &= \lambda I + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\left\{B\frac{\partial}{\partial w}\frac{\partial \ell(\xi_i,+1)}{\partial \xi_i} - C\frac{\partial}{\partial w}\frac{\partial \ell(\xi_i,-1)}{\partial \xi_i}\right\}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})^\top \\
&= \lambda I + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\left\{B\frac{\partial^2 \ell(\xi_i,+1)}{\partial \xi_i^2}\frac{\partial \xi_i}{\partial w} - C\frac{\partial^2 \ell(\xi_i,-1)}{\partial \xi_i^2}\frac{\partial \xi_i}{\partial w}\right\}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})^\top \\
&= \lambda I + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\left\{B\frac{\partial^2 \ell(\xi_i,+1)}{\partial \xi_i^2} - C\frac{\partial^2 \ell(\xi_i,-1)}{\partial \xi_i^2}\right\}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})^\top \\
&= \lambda I + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}(B-C)\frac{\partial^2 \psi(\xi)}{\partial \xi^2}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})^\top \\
&= \lambda I + \frac{1}{n_{\mathrm{U}}}\frac{\partial^2 \psi(\xi)}{\partial \xi^2}\sum_{i=1}^{n_{\mathrm{U}}}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})^\top.
\end{aligned}$$

Since ℓ is convex, $\frac{\partial^2 \psi(\xi)}{\partial \xi^2} \geq 0$. Besides, $\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})^\top \succeq 0$. Therefore $\boldsymbol{H}\hat{J}_\ell(\boldsymbol{w}) \succeq 0$, and $\hat{J}_\ell(\boldsymbol{w})$ is convex. □

Examples of marginal loss functions other than the squared loss function that satisfy the condition in Theorem 2 are logistic loss and double hinge loss.

If we focus on the margin loss function ψ, the condition in Theorem 2 can be relaxed to that the corresponding loss ℓ is α-linear-odd: $\ell(x,1) - \ell(x,-1) = \alpha x$, for any $\alpha \in \mathbb{R}$ [87]. This condition is both sufficient and necessary for the composite loss to be convex. The sufficiency can be proved in the same way above. Below we prove this condition is necessary. Without additional assumptions, the objective function can be

decomposed as follows

$$\begin{aligned}
\widehat{J}_{\ell}(\boldsymbol{w}) &\triangleq \frac{\pi_{\mathrm{S}}}{2n_{\mathrm{S}}}\sum_{i=1}^{2n_{\mathrm{S}}}\mathcal{L}_{\mathrm{S},\ell}\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i})\right)+\frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\mathcal{L}_{\mathrm{U},\ell}\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i})\right)+\frac{\lambda}{2}\|\boldsymbol{w}\|^{2}\\
&=\frac{\lambda}{2}\boldsymbol{w}^{\top}\boldsymbol{w}+\frac{\pi_{\mathrm{S}}}{2n_{\mathrm{S}}\left(2\pi_{+}-1\right)}\sum_{i=1}^{2n_{\mathrm{S}}}\left\{\ell\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}),+1\right)-\ell\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}),-1\right)\right\}\\
&\quad+\frac{1}{n_{\mathrm{U}}\left(2\pi_{+}-1\right)}\sum_{i=1}^{n_{\mathrm{U}}}\left\{-\pi_{-}\ell\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),+1\right)+\pi_{+}\ell\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),-1\right)\right\}\\
&=\frac{\lambda}{2}\boldsymbol{w}^{\top}\boldsymbol{w}+\frac{\pi_{\mathrm{S}}}{2n_{\mathrm{S}}\left(2\pi_{+}-1\right)}\sum_{i=1}^{2n_{\mathrm{S}}}\left\{\ell\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}),+1\right)-\ell\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{S},i}),-1\right)\right\}\\
&\quad-\frac{\pi_{-}}{n_{\mathrm{U}}\left(2\pi_{+}-1\right)}\sum_{i=1}^{n_{\mathrm{U}}}\left\{\ell\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),+1\right)-\ell\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),-1\right)\right\}\\
&\quad+\frac{1}{n_{\mathrm{U}}\left(2\pi_{+}-1\right)}\sum_{i=1}^{n_{\mathrm{U}}}\left\{(\pi_{+}-\pi_{-})\ell\left(\boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{U},i}),-1\right)\right\}.
\end{aligned}$$

Notice that the first and the last terms are both convex with respect to w. Since the second and third terms cannot be convex simultaneously unless they are both linear in w. Therefore, to make this objective function convex for arbitrary data and hyper-parameters, $(\ell(x,1)-\ell(x,-1))$ must be linear in x.

D.6 Why Assumption $1-\rho_{\mathrm{d}}>\pi_{\mathrm{s}}$ Generally Holds

Note that the physical meanings of π_{s} and $1-\rho_{\mathrm{d}}$ are the pro portion of similar data pairs in unlabeled data D_{u} and noisy similarity data $\tilde{D}_{\mathrm{s}}$. Generally, even though $\tilde{D}_{\mathrm{s}}$ contains some noise, it still collects similar data pairs purposely, and thereby the noise is not too large. Thus, the proportion of similar data pairs in purposely collected noisy similarity data $\tilde{D}_{\mathrm{s}}$ (i.e., $1-\rho_{\mathrm{d}}$) is generally bigger than in unlabeled data D_{u} (i.e., π_{s}). Besides, if this condition is not satisfied, it becomes a different problem and thus we need a new solution rather than still solving this problem in the proposed way.

D.7 Proof of Theorem 3

Theorem D.7.1. *Assume the positive data distribution P_+ and the negative data distribution P_- are mutually irreducible, then P_{d} is irreducible with respect to $\tilde{P}_{\mathrm{s}}$, and P_{s} is irreducible with respect to P. Thus the mixture proportion γ and κ in Lemma 6.3.2 is identifiable.*

Proof. Since the positive data distribution P_+ and the negative data distribution P_- are mutually irreducible, the following two equations hold:

$$\inf_{S\in\mathfrak{S},P_-(S)>0}\frac{P_+(S)}{P_-(S)}=0, \tag{D.6}$$

$$\inf_{S\in\mathfrak{S},P_+(S)>0}\frac{P_-(S)}{P_+(S)}=0. \tag{D.7}$$

For P_{s} and P_{d},

$$\begin{aligned}\frac{P_{\mathrm{d}}(S,S')}{P_{\mathrm{s}}(S,S')}&=\frac{(\pi_+^2+\pi_-^2)(P_+(S)P_-(S')+P_-(S)P_+(S'))}{2(\pi_+^2P_+(S)P_+(S')+\pi_-^2P_-(S)P_-(S'))}\\&=\frac{(\pi_+^2+\pi_-^2)(P_-(S')/P_+(S')+P_-(S)/P_+(S))}{2(\pi_+^2+\pi_-^2(P_-(S)/P_+(S))(P_-(S')/P_+(S')))}.\end{aligned}$$

According to Eq. (D.7),

$$\inf_{S,S'\in\mathfrak{S},P_{\mathrm{s}}(S,S')>0}\frac{P_{\mathrm{d}}(S,S')}{P_{\mathrm{s}}(S,S')}=0. \tag{D.8}$$

Similarly, we have,

$$\inf_{S,S'\in\mathfrak{S},P_{\mathrm{d}}(S,S')>0}\frac{P_{\mathrm{s}}(S,S')}{P_{\mathrm{d}}(S,S')}=0. \tag{D.9}$$

Thus, P_{s} and P_{d} are mutually irreducible.
For P_{d} and $\tilde{P}_{\mathrm{s}}$,

$$\begin{aligned}\frac{P_{\mathrm{d}}(S,S')}{\tilde{P}_{\mathrm{s}}(S,S')}&=\frac{P_{\mathrm{d}}(S,S')}{(1-\rho_{\mathrm{d}})P_{\mathrm{s}}(S,S')+\rho_{\mathrm{d}}P_{\mathrm{d}}(S,S')}\\&=\frac{P_{\mathrm{d}}(S,S')/P_{\mathrm{s}}(S,S')}{(1-\rho_{\mathrm{d}})+\rho_{\mathrm{d}}P_{\mathrm{d}}(S,S')/P_{\mathrm{s}}(S,S')}.\end{aligned}$$

According to Eq. (D.8),

$$\inf_{S,S'\in\mathfrak{S},\tilde{P}_{\mathrm{s}}(S,S')>0} \frac{P_{\mathrm{d}}(S,S')}{\tilde{P}_{\mathrm{s}}(S,S')} = 0. \tag{D.10}$$

Similarly, we have,

$$\inf_{S,S'\in\mathfrak{S},P(S,S')>0} \frac{P_{\mathrm{s}}(S,S')}{P(S,S')} = 0. \tag{D.11}$$

Therefore, P_{d} is irreducible with respect to $\tilde{P}_{\mathrm{s}}$, and P_{s} is irreducible with respect to P. □

D.8 Proof of Theorem 4

Note that the loss function is symmetric to $\boldsymbol{x}_{\mathrm{S},i}$ and $\boldsymbol{x}'_{\mathrm{S},i}$, such that the expected and empirical risks can be expressed as

$$\begin{aligned}
R_{\mathrm{nSU}}(f) &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\tilde{p}_{\mathrm{s}}}[\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x})] + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}[\mathcal{L}_{\mathrm{U}}(\boldsymbol{x})], \\
\hat{R}_{\mathrm{nSU}}(f) &= \frac{1}{2n_{\mathrm{nS}}}\sum_{i=1}^{2n_{\mathrm{nS}}}\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x}_{\mathrm{S},i}) + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\mathcal{L}_{\mathrm{U}}(\boldsymbol{x}_{\mathrm{U},i}).
\end{aligned}$$

Let $R_{\mathrm{nS}}(f) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\tilde{p}_{\mathrm{s}}}[\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x})]$, $R_{\mathrm{U}}(f) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}[\mathcal{L}_{\mathrm{U}}(\boldsymbol{x})]$,$\hat{R}_{\mathrm{nS}}(f) = \frac{1}{2n_{\mathrm{nS}}}\sum_{i=1}^{2n_{\mathrm{nS}}}\mathcal{L}_{\mathrm{nS}}(\boldsymbol{x}_{\mathrm{S},i})$ and $\hat{R}_{\mathrm{U}}(f) = \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\mathcal{L}_{\mathrm{U}}(\boldsymbol{x}_{\mathrm{U},i})$, we have

$$\begin{aligned}
R(\hat{f}) - R(f^*) &= R_{\mathrm{nSU}}(\hat{f}) - R_{\mathrm{nSU}}(f^*) \\
&= (R_{\mathrm{nSU}}(\hat{f}) - \hat{R}_{\mathrm{nSU}}(\hat{f})) + (\hat{R}_{\mathrm{nSU}}(\hat{f}) - \hat{R}_{\mathrm{nSU}}(f^*)) \\
&\quad + (\hat{R}_{\mathrm{nSU}}(f^*) - R_{\mathrm{nSU}}(f^*)) \\
&\leq (R_{\mathrm{nSU}}(\hat{f}) - \hat{R}_{\mathrm{nSU}}(\hat{f})) + 0 + (\hat{R}_{\mathrm{nSU}}(f^*) - R_{\mathrm{nSU}}(f^*)) \\
&\leq 2\sup_{f\in\mathcal{F}}\left|R_{\mathrm{nSU}}(f) - \hat{R}_{\mathrm{nSU}}(f)\right| \\
&\leq 2\sup_{f\in\mathcal{F}}\left|R_{\mathrm{nS}}(f) - \hat{R}_{\mathrm{nS}}(f)\right| + 2\sup_{f\in\mathcal{F}}\left|R_{\mathrm{U}}(f) - \hat{R}_{\mathrm{U}}(f)\right|.
\end{aligned} \tag{D.12}$$

The third inequality holds because of the definition of f^*.

Here, we introduce the generalization error with Rademacher complexity.

Lemma D.8.1 ([8])**.** *Let the loss function be upper bounded by M. Then, for any $\delta > 0$, with the probability $1 - \delta$, we have*

$$\sup_{f \in \mathcal{F}} |\, \mathbb{E}[f(x)] - \frac{1}{n} \sum_{i=1}^{n} f(x_i)| \leq 2\mathfrak{R}_n(\ell \circ \mathcal{F}) + M\sqrt{\frac{\log 1/\delta}{2n}}, \tag{D.13}$$

where $\mathfrak{R}_n(\ell \circ \mathcal{F})$ is the Rademacher complexity defined by

$$\mathfrak{R}_n(\ell \circ \mathcal{F}) = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell(f(\boldsymbol{x}_i), f(\boldsymbol{x}_{i'}), \bar{S}_{ii'})\right], \tag{D.14}$$

and $\{\sigma_1, \cdots, \sigma_n\}$ are Rademacher variables uniformly distributed from $\{-1, 1\}$.

Now we can bound two terms in Eq.(D.12) with the next two lemmas.

Lemma D.8.2. *Assume the loss function ℓ is ρ-Lipschitz with respect to the first argument ($0 < \rho < \infty$), and all functions in the model class $\mathcal{F}$ are bounded, i.e., there exists a constant C_{b} such that $\|f\|_\infty \leq C_{\mathrm{b}}$ for any $f \in \mathcal{F}$. Let $C_\ell \triangleq \sup_{t \in \{\pm 1\}} \ell(C_{\mathrm{b}}, t)$. For any $\delta > 0$, with probability at least $1 - \frac{\delta}{2}$,*

$$\sup_{f \in \mathcal{F}} \left| R_{N\tilde{S}}(f) - \hat{R}_{N\tilde{S}}(f) \right| \leq \frac{4A\rho C_{\mathcal{F}} + A\sqrt{2C_\ell^2 \log \frac{4}{\delta}}}{\sqrt{2n_{\mathrm{nS}}}}.$$

Proof. By Lemma D.8.1,

$$\begin{aligned}
&\sup_{f\in\mathcal{F}}\left|R_{N\tilde{S}}(f)-\hat{R}_{N\tilde{S}}(f)\right| \\
&\quad= A\sup_{f\in\mathcal{F}}\left|\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\tilde{p}_{\mathrm{s}}}[\ell(f(\boldsymbol{x}),+1)-\ell(f(\boldsymbol{x}),-1)]-\frac{1}{2n_{\mathrm{nS}}}\sum_{i=1}^{2n_{\mathrm{nS}}}[\ell(f(\boldsymbol{x}_{\mathrm{S},i}),+1)-\ell(f(\boldsymbol{x}_{\mathrm{S},i}),-1)]\right| \\
&\quad\le A\left\{\sup_{f\in\mathcal{F}}\left|\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\tilde{p}_{\mathrm{s}}}[\ell(f(\boldsymbol{x}),+1)]-\frac{1}{2n_{\mathrm{nS}}}\sum_{i=1}^{2n_{\mathrm{nS}}}\ell(f(\boldsymbol{x}_{\mathrm{S},i}),+1)\right|\right. \\
&\qquad\left.+\sup_{f\in\mathcal{F}}\left|\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\tilde{p}_{\mathrm{s}}}[\ell(f(\boldsymbol{x}),-1)]-\frac{1}{2n_{\mathrm{nS}}}\sum_{i=1}^{2n_{\mathrm{nS}}}\ell(f(\boldsymbol{x}_{\mathrm{S},i}),-1)\right|\right\} \\
&\quad\le A\left\{4\mathfrak{R}(\ell\circ\mathcal{F};2n_{\mathrm{nS}},\tilde{p}_{\mathrm{s}})+\sqrt{\frac{2C_\ell^2\log\frac{4}{\delta}}{2n_{\mathrm{nS}}}}\right\},
\end{aligned}$$

where $\ell\circ\mathcal{F}$ in the last line means $\{\ell\circ f|f\in\mathcal{F}\}$. The last inequality holds from Lemma D.8.1. By Talagrand's lemma (Lemma 4.2 in [79]),

$$\mathfrak{R}(\ell\circ\mathcal{F};2n_{\mathrm{nS}},\tilde{p}_{\mathrm{s}})\le\rho\mathfrak{R}(\mathcal{F};2n_{\mathrm{nS}},\tilde{p}_{\mathrm{s}}).$$

Together with $\mathfrak{R}(\mathcal{F};n,\mu)\le\frac{C_{\mathcal{F}}}{\sqrt{n}}$, we obtain

$$\begin{aligned}
\sup_{f\in\mathcal{F}}\left|R_{N\tilde{S}}(f)-\hat{R}_{N\tilde{S}}(f)\right| &\le A\left\{4\rho\frac{C_{\mathcal{F}}}{\sqrt{2n_{\mathrm{nS}}}}+\sqrt{\frac{2C_\ell^2\log\frac{4}{\delta}}{2n_{\mathrm{nS}}}}\right\} \\
&=\frac{4A\rho C_{\mathcal{F}}+A\sqrt{2C_\ell^2\log\frac{4}{\delta}}}{\sqrt{2n_{\mathrm{nS}}}}.
\end{aligned}$$

Lemma D.8.3. *Assume the loss function ℓ is ρ-Lipschitz with respect to the first argument ($0<\rho<\infty$), and all functions in the model class $\mathcal{F}$ are bounded, i.e., there exists a constant C_{b} such that $\|f\|_\infty\le C_{\mathrm{b}}$ for any $f\in\mathcal{F}$. Let $C_\ell\triangleq\sup_{t\in\{\pm1\}}\ell(C_{\mathrm{b}},t)$. For any $\delta>0$, with probability at least $1-\frac{\delta}{2}$,*

$$\sup_{f\in\mathcal{F}}\left|R_{\mathrm{U}}(f)-\hat{R}_{\mathrm{U}}(f)\right|\le\frac{2(-B-C)\rho C_{\mathcal{F}}+(-B-C)\sqrt{\frac{1}{2}C_\ell^2\log\frac{4}{\delta}}}{\sqrt{n_{\mathrm{U}}}}.$$

This lemma can be proven similarly to Lemma D.8.2.

Combining Lemma D.8.2, Lemma D.8.3 and Eq. (D.12), Theorem 6.3.4 is proven. □

If we further take the MPE error into consideration, there is a gap between the ground-truth empirical risk $\hat{R}_{\mathrm{nSU}}(f)$ and the approximated empirical risk $\hat{\hat{R}}_{\mathrm{nSU}}(f)$, which uses the estimated class prior and noise rate. We have

$$\begin{aligned}
|\hat{R}_{\mathrm{nSU}}(f) - \hat{\hat{R}}_{\mathrm{nSU}}(f)| = |&\frac{A-\hat{A}}{2N_{\mathrm{nS}}}\sum_{i=1}^{2N_{\mathrm{nS}}}[\ell(f(\boldsymbol{x}_{\mathrm{S},i}),+1)-\ell(f(\boldsymbol{x}_{\mathrm{S},i}),-1)] \\
&+\frac{B-\hat{B}}{N_{\mathrm{U}}}\sum_{i=1}^{N_{\mathrm{U}}}[\ell(f(\boldsymbol{x}_{\mathrm{U},i},+1)]-\frac{C-\hat{C}}{N_{\mathrm{U}}}\sum_{i=1}^{N_{\mathrm{U}}}[\ell(f(\boldsymbol{x}_{\mathrm{U},i}),-1)]| \\
\leq &\left|\frac{A-\hat{A}}{2N_{\mathrm{nS}}}\sum_{i=1}^{2N_{\mathrm{nS}}}[\ell(f(\boldsymbol{x}_{\mathrm{S},i}),+1)-\ell(f(\boldsymbol{x}_{\mathrm{S},i}),-1)]\right| \\
&+\left|\frac{B-\hat{B}}{N_{\mathrm{U}}}\sum_{i=1}^{N_{\mathrm{U}}}[\ell(f(\boldsymbol{x}_{\mathrm{U},i}),+1)]\right|+\left|\frac{C-\hat{C}}{N_{\mathrm{U}}}\sum_{i=1}^{N_{\mathrm{U}}}[\ell(f(\boldsymbol{x}_{\mathrm{U},i}),-1)]\right|,
\end{aligned}$$

where $\hat{A},\hat{B},\hat{C}$ are the corresponding estimated ones.

Ideally, if we have the knowledge of the exact class priors and noise rate parameters, those risks can be directly calculated since they are both empirical risks. If not, according to the Theorem 12 in [93], we know that the estimated value $\hat{\lambda}$ converges to the true value λ with a rate $\mathcal{O}(m^{-\frac{1}{2}})$:

$$|\lambda-\hat{\lambda}| \leq |\alpha(\lambda)m^{-\frac{1}{2}}|,$$

where $\alpha(\lambda)$ is the coefficient and m is the smaller number of data from two proportions, i.e., F and H in the MPE. Let $\pi_{\mathrm{s}} = g(\kappa,\gamma) = \frac{\gamma(1-\kappa)}{1-\gamma\kappa}$. The Taylor series of π_{s} at the true value π_{s}^* (κ^*,γ^*) is:

$$\pi_{\mathrm{s}} = g(\kappa,\gamma) = g(\kappa^*,\gamma^*) + (\kappa-\kappa^*)g_{\kappa}^{'}(\kappa^*,\gamma^*) + (\gamma-\gamma^*)g_{\kappa}^{'}(\kappa^*,\gamma^*) + o^n.$$

By omitting the high-order terms, we have:

$$\begin{aligned}|\pi_{\mathrm{s}}-\pi_{\mathrm{s}}^{*}| &\leq |(\kappa-\kappa^{*})g_{\kappa}^{'}(\kappa^{*},\gamma^{*})+(\gamma-\gamma^{*})g_{\kappa}^{'}(\kappa^{*},\gamma^{*})| \\ &\leq \left(|\alpha(\kappa^{*})g_{\kappa}^{'}(\kappa^{*},\gamma^{*})|+|\alpha(\gamma^{*})g_{\kappa}^{'}(\kappa^{*},\gamma^{*})|\right)\left|m^{-\frac{1}{2}}\right|,\end{aligned}$$

which indicates that the convergence rate for π_{s} is $\mathcal{O}(m^{-\frac{1}{2}})$. Likewise, ρ_{d} and π_{+} both have the convergence rate of $\mathcal{O}(m^{-\frac{1}{2}})$. Further, given that π_{s}, ρ_{d}, and π_{+} all converge with a rate of $\mathcal{O}(m^{-\frac{1}{2}})$, similarly, we have that A, B, and C all have the same convergence rate of $\mathcal{O}(m^{-\frac{1}{2}})$.

D.9 Specific Class Information regarding *News20* and *CIFAR-10*

Table D.3: The relationships between the semantic classes in the original News20 and the classes selected in the News_05, $\cdots$, News_49 datasets.

Dataset	Positive	Negative
News_05	alt.atheism	comp.graphics
News_16	misc.forsale	rec.autos
News_27	talk.politics.mideast	comp.sys.ibm.pc.hardware
News_38	comp.os.ms-windows.misc	sci.crypt
News_49	sci.space	sci.med

Table D.4: The relationships between the semantic classes in the original CIFAR-10 and the classes selected in the Cifar_03, Cifar_14, and Cifar_25 datasets.

Dataset	Positive	Negative
Cifar_03	airplane	dog
Cifar_14	cat	ship
Cifar_25	deer	truck

Bibliography

[1] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami. "Contributions to the study of SMS spam filtering: new collection and results". In: *Proceedings of the 11th ACM Symposium on Document Engineering*. 2011.

[2] D. Angluin and P. Laird. "Learning from noisy examples". In: *Machine Learning* 2.4 (1988), pp. 343–370.

[3] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. "Machine bias". In: *ProPublica, May* 23.2016 (2016), pp. 139–159.

[4] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu, and T. Liu. "Understanding and Improving Early Stopping for Learning with Noisy Labels". In: *NeurIPS* 34 (2021).

[5] H. Bao, G. Niu, and M. Sugiyama. "Classification from Pairwise Similarity and Unlabeled Data". In: *ICML*. 2018.

[6] H. Bao, T. Shimada, L. Xu, I. Sato, and M. Sugiyama. "Similarity-based Classification: Connecting Similarity Learning to Binary Classification". In: *arXiv preprint arXiv:2006.06207* (2020).

[7] S. Barocas and A. D. Selbst. "Big data's disparate impact". In: *Calif. L. Rev.* 104 (2016), p. 671.

[8] P. L. Bartlett and S. Mendelson. "Rademacher and Gaussian complexities: Risk bounds and structural results". In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 463–482.

[9] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. "Curriculum learning". In: *ICML*. 2009, pp. 41–48.

[10] A. Berthon, B. Han, G. Niu, T. Liu, and M. Sugiyama. "Confidence scores make instance-dependent label-noise learning possible". In: *ICML*. PMLR. 2021, pp. 825–836.

[11] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[12] B. Biggio, B. Nelson, and P. Laskov. "Support vector machines under adversarial label noise". In: *ACML*. 2011.

[13] G. Blanchard, G. Lee, and C. Scott. "Semi-supervised novelty detection". In: *Journal of Machine Learning Research* 11.Nov (2010), pp. 2973–3009.

[14] M. Bogen and A. Rieke. "Help wanted: An examination of hiring algorithms, equity, and bias". In: (2018).

[15] L. Bruzzone and M. Marconcini. "Domain adaptation problems: A DASVM classification technique and a circular validation strategy". In: *IEEE transactions on pattern analysis and machine intelligence* 32.5 (2009), pp. 770–787.

[16] C.-C. Chang and C.-J. Lin. "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27.

[17] X. Chen and A. Gupta. "Webly supervised learning of convolutional networks". In: *ICCV*. 2015, pp. 1431–1439.

[18] H. Cheng, Z. Zhu, X. Li, Y. Gong, X. Sun, and Y. Liu. "Learning with instance-dependent label noise: A sample sieve approach". In: *ICLR* (2021).

[19] J. Cheng, T. Liu, K. Ramamohanarao, and D. Tao. "Learning with bounded instance and label-dependent label noise". In: *ICML*. PMLR. 2020, pp. 1789–1799.

[20] A. Chouldechova. "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments". In: *Big data* 5.2 (2017), pp. 153–163.

[21] L. Cohen, Z. C. Lipton, and Y. Mansour. "Efficient candidate screening under multiple tests and implications for fairness". In: *1st Symposium on Foundations of Responsible Computing, FORC 2020, June 1-3, 2020, Harvard University, Cambridge, MA, USA (virtual conference)* (2019).

[22] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), pp. 273–297.

[23] S. Dan, H. Bao, and M. Sugiyama. "Learning from Noisy Similar and Dissimilar Data". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 2021.

[24] S. Danziger, J. Levav, and L. Avnaim-Pesso. "Extraneous factors in judicial decisions". In: *Proceedings of the National Academy of Sciences* 108.17 (2011), pp. 6889–6892.

[25] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. "Information-theoretic metric learning". In: *ICML*. 2007.

[26] J. Dressel and H. Farid. "The accuracy, fairness, and limits of predicting recidivism". In: *Science advances* 4.1 (2018), eaao5580.

[27] D. Dua and C. Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[28] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. "Fairness through awareness". In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pp. 214–226.

[29] R. J. Fisher. "Social desirability bias and the validity of indirect questioning". In: *Journal of Consumer Research* 20.2 (1993), pp. 303–315.

[30] X. Gastaldi. "Shake-Shake regularization". In: *arXiv preprint arXiv:1705.07485* (2017).

[31] N. Golowich, A. Rakhlin, and O. Shamir. "Size-independent sample complexity of neural networks". In: *COLT*. 2018.

[32] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu. "Automated curriculum learning for neural networks". In: *ICML*. PMLR. 2017, pp. 1311–1320.

[33] S. Guo, W. Huang, H. Zhang, C. Zhuang, D. Dong, M. R. Scott, and D. Huang. "Curriculumnet: Weakly supervised learning from large-scale web images". In: *ECCV*. 2018, pp. 135–150.

[34] 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". In: *ICML*. PMLR. 2020, pp. 4006–4016.

[35] B. Han, J. Yao, G. Niu, M. Zhou, I. Tsang, Y. Zhang, and M. Sugiyama. "Masking: A new perspective of noisy supervision". In: *Advances in Neural Information Processing Systems*. 2018, pp. 5836–5846.

[36] B. Han, Q. Yao, T. Liu, G. Niu, I. W. Tsang, J. T. Kwok, and M. Sugiyama. "A survey of label-noise representation learning: Past, present and future". In: *arXiv preprint arXiv:2011.04406* (2020).

[37] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. "Co-teaching: Robust training of deep neural networks with extremely noisy labels". In: *NeurIPS*. 2018.

[38] M. Hardt, E. Price, and N. Srebro. "Equality of opportunity in supervised learning". In: *NeurIPS* 29 (2016), pp. 3315–3323.

[39] G. H. Hardy, J. E. Littlewood, G. Pólya, G. Pólya, D. Littlewood, et al. *Inequalities*. Cambridge university press, 1952.

[40] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *CVPR*. 2016, pp. 770–778.

[41] K. He, X. Zhang, S. Ren, and J. Sun. "Identity mappings in deep residual networks". In: *ECCV*. Springer. 2016, pp. 630–645.

[42] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel. "Using trusted data to train deep networks on labels corrupted by severe noise". In: *NeurIPS*. 2018.

[43] Y.-C. Hsu, Z. Lv, and Z. Kira. "Learning to cluster in order to transfer across domains and tasks". In: *International Conference on Learning Representations (ICLR)*. 2018. URL: https://openreview.net/forum?id=ByRWCqvT-.

[44] Y.-C. Hsu, Z. Lv, J. Schlosser, P. Odom, and Z. Kira. "Multi-class classification without multi-class labels". In: *ICLR*. 2019.

[45] W. Hu, Z. Li, and D. Yu. "Simple and Effective Regularization Methods for Training on Noisily Labeled Data with Generalization Guarantee". In: *ICLR*. 2020.

[46] W. Huan, Y. Wu, L. Zhang, and X. Wu. "Fairness through equality of effort". In: *Companion Proceedings of the Web Conference 2020*. 2020, pp. 743–751.

[47] P. J. Huber et al. "Robust Estimation of a Location Parameter". In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101.

[48] 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". In: *ICML*. 2018, pp. 2309–2318.

[49] S. C. Johnson. "Hierarchical clustering schemes". In: *Psychometrika* 32.3 (1967), pp. 241–254.

[50] A. Khademi, S. Lee, D. Foley, and V. Honavar. "Fairness in algorithmic decision making: An excursion through the lens of causality". In: *The World Wide Web Conference*. 2019, pp. 2907–2914.

[51] T. Kim, J. Ko, J. Choi, S.-Y. Yun, et al. "Fine samples for learning with noisy labels". In: *NeurIPS* 34 (2021), pp. 24137–24149.

[52] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". In: *ICLR*. 2014.

[53] J. Kremer, F. Sha, and C. Igel. "Robust Active Label Correction". In: *AISTATS*. 2018, pp. 308–316.

[54] A. Krizhevsky, G. Hinton, et al. *Learning multiple layers of features from tiny images*. Tech. rep. Citeseer, 2009.

[55] H. W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97.

[56] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva. "Counterfactual fairness". In: *NeurIPS* (2017).

[57] Y. LeCun. "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/* (1998).

[58] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[59] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

[60] J. Li, R. Socher, and S. C. Hoi. "Dividemix: Learning with noisy labels as semi-supervised learning". In: *ICLR* (2020).

[61] X. Li, T. Liu, B. Han, G. Niu, and M. Sugiyama. "Provably end-to-end label-noise learning without anchor points". In: *ICML*. PMLR. 2021, pp. 6403–6413.

[62] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li. "Learning from noisy labels with distillation". In: *ICCV*. 2017.

[63] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda. "Early-learning regularization prevents memorization of noisy labels". In: *NeurIPS* 33 (2020), pp. 20331–20342.

[64] T. Liu and D. Tao. "Classification with noisy labels by importance reweighting". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.3 (2016), pp. 447–461.

[65] Y. Liu and H. Guo. "Peer Loss Functions: Learning from Noisy Labels without Knowing Noise Rates". In: *ICML*. 2020.

[66] N. Lu, G. Niu, A. K. Menon, and M. Sugiyama. "On the minimal supervision for training any binary classifier from only unlabeled data". In: *ICLR*. 2019.

[67] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey. "Normalized Loss Functions for Deep Learning with Noisy Labels". In: *ICML*. 2020.

[68] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. M. Erfani, S.-T. Xia, S. Wijewickrema, and J. Bailey. "Dimensionality-Driven Learning with Noisy Labels". In: *ICML*. 2018, pp. 3361–3370.

[69] J. MacQueen. "Classification and analysis of multivariate observations". In: *5th Berkeley Symp. Math. Statist. Probability*. 1967, pp. 281–297.

[70] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. "Towards deep learning models resistant to adversarial attacks". In: *ICLR*. 2018.

[71] K. Makhlouf, S. Zhioua, and C. Palamidessi. "Survey on Causal-based Machine Learning Fairness Notions". In: *arXiv preprint arXiv:2010.09553* (2020).

[72] E. Malach and S. Shalev-Shwartz. "Decoupling" when to update" from" how to update"". In: *NeurIPS*. 2017, pp. 960–970.

[73] N. Manwani and P. Sastry. "Noise tolerance under risk minimization". In: *IEEE Transactions on Cybernetics* (2013).

[74] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. "A survey on bias and fairness in machine learning". In: *ACM Computing Surveys (CSUR)* 54.6 (2021), pp. 1–35.

[75] S. Mendelson. "Lower bounds for the empirical minimization algorithm". In: *IEEE Transactions on Information Theory* 54.8 (2008), pp. 3797–3803.

[76] A. Menon, B. Van Rooyen, C. S. Ong, and B. Williamson. "Learning from corrupted binary labels via class-probability estimation". In: *ICML*. 2015.

[77] A. K. Menon, A. S. Rawat, S. J. Reddi, and S. Kumar. "Can gradient clipping mitigate label noise?" In: *ICLR* (2020).

[78] B. Mirzasoleiman, K. Cao, and J. Leskovec. "Coresets for Robust Training of Neural Networks against Noisy Labels". In: *NeurIPS* 33 (2020).

[79] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.

[80] A. Mukerjee, R. Biswas, K. Deb, and A. P. Mathur. "Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management". In: *International Transactions in operational research* 9.5 (2002), pp. 583–597.

[81] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. "Learning with noisy labels". In: *NeurIPS*. 2013.

[82] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y.Ng. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. 2011.

[83] C. G. Northcutt, T. Wu, and I. L. Chuang. "Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels". In: *UAI*. 2017.

[84] C. G. Northcutt, A. Athalye, and J. Mueller. *Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks*. 2021. arXiv: 2103.14749 [stat.ML].

[85] C. O'neil. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown, 2016.

[86] E. Oral. "Surveying Sensitive Topics with Indirect Questioning". In: *Statistical Methodologies*. IntechOpen, 2019.

[87] G. Patrini, F. Nielsen, R. Nock, and M. Carioni. "Loss factorization, weakly supervised learning and label noise robustness". In: *ICML*. 2016.

[88] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu. "Making deep neural networks robust to label noise: A loss correction approach". In: *CVPR*. 2017.

[89] J. Pearl et al. "Causality: Models, reasoning and inference". In: *Cambridge, UK: CambridgeUniversityPress* 19 (2000).

[90] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al.

"Scikit-learn: Machine learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[91] J. Pennington, R. Socher, and C. D. Manning. "Glove: Global vectors for word representation". In: *EMNLP*. 2014.

[92] M. du Plessis, G. Niu, and M. Sugiyama. "Convex formulation for learning from positive and unlabeled data". In: *ICML*. 2015.

[93] H. Ramaswamy, C. Scott, and A. Tewari. "Mixture proportion estimation via kernel embeddings of distributions". In: *ICML*. 2016.

[94] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. "TRAINING DEEP NEURAL NETWORKS ON NOISY LABELS WITH BOOTSTRAPPING". In: *ICLR*. 2015.

[95] M. Ren, W. Zeng, B. Yang, and R. Urtasun. "Learning to Reweight Examples for Robust Deep Learning". In: *ICML*. 2018, pp. 4331–4340.

[96] N. A. Saxena, K. Huang, E. DeFilippis, G. Radanovic, D. C. Parkes, and Y. Liu. "How do fairness definitions fare? Examining public attitudes towards algorithmic definitions of fairness". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019, pp. 99–106.

[97] C. Scott. "A rate of convergence for mixture proportion estimation, with application to learning from noisy labels". In: *AISTATS*. 2015.

[98] C. Scott, G. Blanchard, and G. Handy. "Classification with asymmetric label noise: Consistency and maximal denoising". In: *COLT*. 2013.

[99] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[100] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), pp. 484–489.

[101] B. C. Stahl and D. Wright. "Ethics and privacy in AI and big data: Implementing responsible research and innovation". In: *IEEE Security & Privacy* 16.3 (2018), pp. 26–33.

[102] J. M. Steele. *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.

[103] Y. Su and B. Xiong. *Methods and Techniques for Proving Inequalities: In Mathematical Olympiad and Competitions*. Vol. 11. World Scientific Publishing Company, 2015.

[104] M. Sugiyama and M. Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT Press, 2012.

[105] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. "Intriguing properties of neural networks". In: *ICLR*. 2014.

[106] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa. "Joint optimization framework for learning with noisy labels". In: *CVPR*. 2018, pp. 5552–5560.

[107] A. Tewari and P. L. Bartlett. "On the consistency of multiclass classification methods". In: *JMLR* 8.May (2007), pp. 1007–1025.

[108] J. Turian, L. Ratinov, and Y. Bengio. "Word representations: a simple and general method for semi-supervised learning". In: *ACL*. 2010, pp. 384–394.

[109] A. Vahdat. "Toward robustness against label noise in training deep discriminative neural networks". In: *NeurIPS*. 2017, pp. 5596–5605.

[110] V. Vapnik. "Principles of risk minimization for learning theory". In: *NeurIPS*. 1991.

[111] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie. "Learning from noisy large-scale datasets with minimal supervision". In: *CVPR*. 2017, pp. 839–847.

[112] S. Verma and J. Rubin. "Fairness definitions explained". In: *2018 ieee/acm international workshop on software fairness (fairware)*. IEEE. 2018, pp. 1–7.

[113] X. Wang, S. Wang, J. Wang, H. Shi, and T. Mei. "Co-mining: Deep face recognition with noisy labels". In: *ICCV*. 2019, pp. 9358–9367.

[114] S. L. Warner. "Randomized response: A survey technique for eliminating evasive answer bias". In: *Journal of the American Statistical Association* 60.309 (1965), pp. 63–69.

[115] H. Wei, L. Feng, X. Chen, and B. An. "Combating Noisy Labels by Agreement: A Joint Training Method with Co-Regularization". In: *CVPR*. June 2020.

[116] H. Wei, L. Feng, X. Chen, and B. An. "Combating noisy labels by agreement: A joint training method with co-regularization". In: *CVPR*. 2020, pp. 13726–13735.

[117] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu. "Learning with Noisy Labels Revisited: A Study Using Real-World Human Annotations". In: *ICLR*. 2022.

[118] P. Wu, S. Zheng, M. Goswami, D. Metaxas, and C. Chen. "A Topological Filter for Learning with Label Noise". In: *NeurIPS*. 2020.

[119] S. Wu, M. Gong, B. Han, Y. Liu, and T. Liu. "Fair Classification with Instance-dependent Label Noise". In: *First Conference on Causal Learning and Reasoning*. 2022.

[120] S. Wu, T. Liu, B. Han, J. Yu, G. Niu, and M. Sugiyama. "Learning from Noisy Pairwise Similarity and Unlabeled Data". In: *Journal of Machine Learning Research* 23.307 (2022), pp. 1–34. URL: http://jmlr.org/papers/v23/21-0946.html.

[121] S. Wu, X. Xia, T. Liu, B. Han, M. Gong, N. Wang, H. Liu, and G. Niu. "Class2simi: A noise reduction perspective on learning with noisy labels". In: *ICML*. 2021.

[122] X. Xia, T. Liu, B. Han, M. Gong, J. Yu, G. Niu, and M. Sugiyama. "Sample selection with uncertainty of losses for learning with noisy labels". In: *ICLR*. 2022.

[123] X. Xia, T. Liu, B. Han, N. Wang, J. Deng, J. Li, and Y. Mao. "Extended T: Learning with Mixed Closed-set and Open-set Noisy Labels". In: *arXiv preprint arXiv:2012.00932* (2020).

[124] X. Xia, T. Liu, B. Han, N. Wang, M. Gong, H. Liu, G. Niu, D. Tao, and M. Sugiyama. "Part-dependent label noise: Towards instance-dependent label noise". In: *NeurIPS*. 2020.

[125] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama. "Are Anchor Points Really Indispensable in Label-Noise Learning?" In: *NeurIPS*. 2019.

[126] H. Xiao, K. Rasul, and R. Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms". In: *arXiv preprint arXiv:1708.07747* (2017).

[127] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. "Learning from massive noisy labeled data for image classification". In: *CVPR*. 2015, pp. 2691–2699.

[128] Y. Xu, P. Cao, Y. Kong, and Y. Wang. "L_dmi: An information-theoretic noise-robust loss function". In: *NeurIPS*. 2019.

[129] S. Yang, E. Yang, B. Han, Y. Liu, M. Xu, G. Niu, and T. Liu. "Estimating instance-dependent label-noise transition matrix using dnns". In: *arXiv preprint arXiv:2105.13001* (2021).

[130] Q. Yao, H. Yang, B. Han, G. Niu, and J. Kwok. "Searching to Exploit Memorization Effect in Learning with Noisy Labels". In: *ICML*. 2020.

[131] Y. Yao, T. Liu, M. Gong, B. Han, G. Niu, and K. Zhang. "Instance-dependent Label-noise Learning under a Structural Causal Model". In: *NeurIPS* (2021).

[132] 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". In: *NeurIPS*. 2020.

[133] X. Yu, B. Han, J. Yao, G. Niu, I. W. Tsang, and M. Sugiyama. "How Does Disagreement Benefit Co-teaching?" In: *ICML*. 2019.

[134] X. Yu, T. Liu, M. Gong, K. Batmanghelich, and D. Tao. "An efficient and provable approach for mixture proportion estimation using linear independence assumption". In: *CVPR*. 2018, pp. 4480–4489.

[135] X. Yu, T. Liu, M. Gong, and D. Tao. "Learning with biased complementary labels". In: *ECCV*. 2018, pp. 68–83.

[136] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi. "Fairness constraints: Mechanisms for fair classification". In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 962–970.

[137] J. H. Zar. "Spearman rank correlation". In: *Encyclopedia of biostatistics* 7 (2005).

[138] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. "Understanding deep learning requires rethinking generalization". In: *ICLR*. 2017.

[139] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. "mixup: Beyond Empirical Risk Minimization". In: *ICLR* (2018).

[140] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. "mixup: Beyond empirical risk minimization". In: (2018).

[141] J. Zhang and E. Bareinboim. "Equality of opportunity in classification: A causal approach". In: *NeurIPS*. 2018, pp. 3675–3685.

[142] Y. Zhang, G. Niu, and M. Sugiyama. "Learning Noise Transition Matrix from Only Noisy Labels via Total Variation Regularization". In: *ICML* (2021).

[143] Z. Zhang and M. Sabuncu. "Generalized cross entropy loss for training deep neural networks with noisy labels". In: *NeurIPS*. 2018, pp. 8778–8788.

[144] S. Zheng, P. Wu, A. Goswami, M. Goswami, D. Metaxas, and C. Chen. "Error-bounded correction of noisy labels". In: *ICML*. 2020, pp. 11447–11457.

[145] T. Zhou, S. Wang, and J. Bilmes. "Robust curriculum learning: from clean label detection to noisy label self-correction". In: *ICLR*. 2021.

[146] T. Zhou, S. Wang, and J. Bilmes. "Curriculum learning by dynamic instance hardness". In: *NeurIPS* 33 (2020), pp. 8602–8613.

[147] Z. Zhu, T. Liu, and Y. Liu. "A second-order approach to learning with instance-dependent label noise". In: *CVPR*. 2021, pp. 10113–10123.