Assignment 2

Tutors: Yu Yao; Jazlyn Lin
Group members: Ke Wang 490538133 kwan7498; Hui-Yun Huang 520172904 hhua9634; Zijie Zhao 490576560 zzha9494

Abstract

Label noise is one of the most critical problems in classification due to its potential negative impacts. Some solutions focus on filtering label noise, while others try to model it without modification on the dataset. In this study, three label noise robust classifiers are implemented with two transition matrix estimators. We first introduce the problems related to label noise as well as the significance in applications. Then we briefly illustrate different methods that can deal with label noise. Specifically, we choose the anchor points method and the Dual-T method to estimate the transition matrix as an important component in three algorithms: FFNN, CNN and ResNet. Their performances are evaluated on three datasets. Finally, we encapsulate the results and add some comments for future work.

1 Introduction

Classification has been extensively researched in machine learning, where the typical method includes learning a model fitted by a labelled dataset to predict the class of a sample sight unseen [1]. The reliability of a model significantly depends on the authenticity of the dataset with labels. However, in the real world, obtaining or affirming the accuracy of a labelled dataset is sometimes unachievable [2]. This is mainly because of the size of the dataset required for training and the limitation of crowdsourcing with insufficient expertise for labelling [3]. As a result, noisy labels are ubiquitous and degenerate the model learned by label noise-contaminated dataset [4]. In addition, label noise is considered to be a critical issue in practice. For example, a gold standard for a medical diagnosis requires nearly absolutely accurate results in medical applications [5].

Therefore, it is needful to introduce techniques that eliminate or reduce noise and its consequences. Two directions for learning with noisy labels are widely used: algorithms with statistically inconsistent or consistent classifiers [6]. The latter is to address the main limitation of the former one by training models using noisy data. As the training progress does not require modification on the labels and the models can also asymptotically converge to the optimal one trained by clean data, it is considered to be statistically consistent. The transition matrix is the key to these classifiers and the performance mostly relies on the transition matrix. In this study, two transition matrix estimators (anchor points method and dual-T method) and three classification algorithms that are robust to the label noise are implemented, using three datasets that contain class-conditional random label noise.

2 Related work

The impacts of label noise cannot be underestimated with the issues such as the degeneration of the performance, the restriction of the learning requirements and the distortion of feature distribution. Thus, addressing label noise in machine learning has been researched intensively. In the literature, statistically inconsistent or statistically consistent can be further extended to three main methods.

The main idea of the first one is to enhance the quality of the dataset. It is naturally intuitively easy to understand that mislabeled examples are either removed or relabeled by some certain filter. The second one is to use algorithms that are directly robust to the label noise. In this case, such algorithms are not sensitive and are more likely to perform well than ones that are vulnerable to label noise. The third one tries to model label noise during training or uses a model embedded with noise information.

2.1 Data Cleansing Methods

Similar to dealing with outliers, the most intuitive way is to cleanse the dataset using some filters, and then train the model with the clean dataset. The framework was inspired by Brodley and Friedl in 1999 [7]. Many filtering methods have seen success recently.

The first group of methods are based on ad hoc measures. The main idea is to remove examples when a measure is beyond a predefined threshold. For the probabilistic classifier, one measure is using the entropy of the conditional distribution [8]. In this case, low entropy means relatively confident and accurate classifications. Therefore, a relabelling process can be done if the label is not consistent with the label predicted by the model.

The other methods involve the predictions of classifiers. Khoshgoftaar and Rebours suggest these algorithms can be categorized into classification, voting and partition filtering [9]. The predictions of a classifier can be used to identify examples with noise labels. For example, SVM [10] and neural networks [11] are used to removed misclassified instances. However, one disadvantage is that these classifiers are easy to remove lots of examples that are on the other side of the decision boundary. To address this dangerous problem, voting by ensembles is introduced. The k-fold cross-validation scheme is first used to split different training and validation sets that are used to create classifiers. Then, the voting filtering for the example is performed [12]. The majority vote and the consensus vote are two main ways. The last one is partition filtering which is usually used for distributed datasets with large sizes. This involves partition and comparison among different subsets.

One advantage of filtering label noise is that the model inference step is not affected by the removed examples [13]. This means that compared to relabelling, removing a mislabelled example is much easier. However, this may result in over-cleansing which can decrease the performance of the model [14]. On the other hand, certain examples are more likely to be removed and it contributes to the difficulty of training. This imbalance of the dataset can in turn reduce the efficiency of data cleansing [15]. Another advantage is it can reduce the complexity of the model. However, the reduced complexity is hard to be defined by the reduced size of the dataset.

2.2 Label Noise-Robust Models

Models are said to be robust to label noise if they are still effective even though the label noise is neither modelled nor cleansed. Theoretically, the natural way to measure the loss is 0-1 loss that is neither convex nor differentiable, which means hard to optimize. Thus, a group of other losses are proposed, some of the surrogates can be used to minimize the risk if the probability of misclassification is identical and has no relation to the label noise [16]. Thathachar and Sastry suggest that the 0-1 loss is robust to the label noise when the label noise is in the uniform distribution [17]. Another label noise robust loss is the least square loss. However, other losses such as the exponential loss, the log loss and the hinge loss, which represent AdaBoost, logistic regression and SVM respectively, are not robust to label noise. Therefore, most of the well-known algorithms do not have this robustness property. Research shows that ensemble methods and decision trees have presented robustness to label noise [18]. This is because the inaccurate information gain can decrease the size of trees, and then reduce the impact of label noise [19].

Although losses in algorithms are hard to achieve robustness to label noise, other methods such as regularization can be a remedy. However, these techniques make the model less compact and less accurate [20]. Thus, the only suitable scenario is that case of label noise is simple with methods to prevent overfitting. Correspondingly, the accuracy decreases.

2.3 Label Noise-Tolerant Learning Algorithms

Label noise-tolerant models are also considered to be statistically consistent. The main idea is to utilize the information of label noise and learn the noise model along with the classifier. It can be split into two groups: probabilistic methods and model-based methods.

Methods that are probabilistic are widely used in applications, such as Bayesian methods, Frequentist Methods, clustering methods and belief functions. The Bayesian method reckons that a probability distribution presenting the uncertainty of their value is in all unknown parameters, and it can include the prior knowledge of the unknown parameters [21]. In the frequentist methods, it is assumed that a label consists of two variables: the true label Y and the observed label Y . In the Class-Dependent Noise assumption, the Y is only dependent on Y, and the relationship can be represented using a labelling matrix [22]. The model has been extended to complex and multiclass datasets [23]. Considering the asymmetric label noise, conditional probabilities are changed by label noise and can be solved by label noise modelling [24]. In Clustering-Based Methods, the main idea is to use the clusters of the dataset to build classifiers [25]. In the belief function theory, a belief value is assigned to every example, which corresponds to the number of evidence supporting the subset of classes [26]. The belief mass can be inferred from the data.

In model-based methods, label noise is handled by semi-supervised learning. A set of variants of popular algorithms generated by specific strategies such as SVMs, neural networks, and decision trees, have been proved effective.

Because the probabilistic methods are developed in a more theoretical way than the dataset cleansing and model robust method, they can be used with prior knowledge of label noise. In addition, information obtained by analysing the label noise can also be helpful to build the classifiers. However, it shows that the complexity of the algorithms increased and overfitting is also more likely to happen.

3 Methods

Given the flip rates are known, loss correction is one of the most widely used approaches to construct classifiers that are robust to the label noise. However, estimation of the transition matrix is required when the flip rate is unknown, which generally requires more knowledge and skills to achieve this. In this study, two methods, anchor point assumption and dual-T estimator, were adopted to estimate the transition matrix and evaluated based on their effectiveness.

3.1 Method with known flip rates

3.1.1 Loss correction

Transition matrix [27][28] plays an essential role in designing a classifier robust to label noise and is defined as

$$T_{ij} = P(\tilde{Y} = i|Y = j, X), \tag{1}$$

where the ij-th entry represents the flip rate that class j is labelled by a noisy label i given X. $P(\cdot)$ denotes the probability, and the label space is denoted as $\mathcal{C} = \{1, 2, \ldots, C\}$.

For class-conditional label noise (CCN), flip rates only depend on the clean label Y, so the transition matrix in Eq. (1) is independent on the instance X, i.e., $T(X) = X$. Given noisy label posterior $P(\tilde{Y}|X) = [P(\tilde{Y} = 1|X), ..., P(\tilde{Y} = C|X)]^\top$ and clean label posterior $P(Y|X) = [P(Y = 1|X), ..., P(Y = C|X)]^\top$, the relationship between the noisy label posterior and the clean label posterior can be modelled by the transition matrix T by

$$P(\tilde{Y}|X) = TP(Y|X). \tag{2}$$

This equation reveals that the clean label posterior can then be inferred from the noisy label posterior if the transition matrix is known.

Many approaches had been proposed to construct classifiers with label noise. So far, the existing literature has primarily relied on the correction of loss. Note that this kind of approach only works if the transition matrix T is given. There are two different approaches: backward and forward

correction [28][29]. In this study, we would only focus on the backward correction approach due to its ease of implementation.

Backward correction introduces an unbiased estimator of the loss function such that the corrected loss calculated under the label noise is equal to the one obtained from the clean data. Let $f(X)$ denotes the network function and l be the loss, the backward corrected loss l' is defined as

$$l'(f(X), \tilde{Y}) = [T^{-1} l_{Y|f(X)}]_{\tilde{Y}} \tag{3}$$

where $l_{Y|f(X)}$ is a vector with the form of $(l(f(X), 1), ..., l(f(X), C))$.

The corrected loss is an unbiased estimator of the target clean label posterior because

$$\begin{aligned} \mathbb{E}_{\tilde{Y}|X} l'(f(X), \tilde{Y}) &= \mathbb{E}_{Y|X} T l'(f(X), Y) \\ &= \mathbb{E}_{Y|X} T T^{-1} l(f(X), Y) \\ &= \mathbb{E}_{Y|X} l(f(X), Y) \end{aligned} \tag{4}$$

where the first equation holds because $P(\tilde{Y}|X) = TP(Y|X)$ according to Eq. (2), and the second equation holds because of Eq. (3). Hence, by adopting the corrected loss, the estimation of the clean label posterior using the model trained with noisy labels is equal to the one trained with clean labels as the optimizers are identical, i.e.,

$$\arg \min_{\hat{P}(Y|X)} \mathbb{E}_{\tilde{Y}|X} l'(f(X), \tilde{Y}) = \arg \min_{\hat{P}(Y|X)} \mathbb{E}_{Y|X} l(f(X), Y) \tag{5}$$

As a result, we can use the noisy label posterior estimated by the network trained with noisy data to approximate the clean label posterior by

$$\hat{P}(Y|X) = T^{-1} \hat{P}(\tilde{Y}|X). \tag{6}$$

Note that although the backward correction is an unbiased estimator of the clean label posterior, practical implementation requires the inverse of the transition matrix. This may become an issue when the transition matrix grows larger with the increasing number of classes within the dataset or the inverse is unavailable.

3.2 Methods with unknown flip rates

3.2.1 Anchor points assumption

Anchor points assumption is a widely adopted approach to estimate the transition matrix for label noise [27][29][30] . The estimation of the transition matrix relies on the anchor points defined in the clean data domain. An instance x^j is said to be an anchor point of class j if it belongs to class j without uncertainty, i.e., $P(Y = j|X = x^j) = 1$. Based on this property, suppose that noisy label posterior and anchor points are given, the transition matrix T of CCN can then be obtained via

$$\begin{aligned} P(\tilde{Y} = i|X = x^j) &= \sum_{k=1}^{C} P(\tilde{Y} = i|Y = k) P(Y = k|X = x^j) \\ &= P(\tilde{Y} = i|Y = j) P(Y = j|X = x^j) \\ &= P(\tilde{Y} = i|Y = j) \\ &= T_{ij}, \end{aligned} \tag{7}$$

where the ij-th entry of the transition matrix is equal to the probability that an anchor point of class j is labelled with a noisy label i. In other words, given the anchor point of class j, the j-th column of the transition matrix, i.e., $[P(\tilde{Y} = 1|X = x^i), ..., P(\tilde{Y} = C|X = x^i)]^\top$, can be learned with its estimated noisy label posterior.

However, anchor points are usually hard to identify as the clean class posterior is unknown. An alternative approach had been proposed to obtain a good estimate of anchor points by approximating them using the instance that has the highest probability belonging to the target class, i.e., $x^j = \arg\max_x P(Y = j|X = x)$. However, since clean label posterior is usually unknown, we can use

the anchor points found by the following equation to estimate the transition matrix [27] when the noise rate is upper bounded.

$$x^j = \arg\max_x \hat{P}(\tilde{Y} = j|X = x), \tag{8}$$

where $\hat{P}(\tilde{Y} = j|X = x)$ is the prediction probability of noisy label $\tilde{Y} = j$ given that $X = x$.

After finding the anchor points, the j-th column of the transition matrix T^{ap} estimated by the anchor points method can be estimated as

$$[\hat{P}(\tilde{Y} = 0|X = x^j), \ldots, \hat{P}(\tilde{Y} = C|X = x^j)]^\top. \tag{9}$$

As a result, based on the equation above, the transition matrix can then be estimated by the approximated anchor points without any foreknowledge about the clean data but only the noisy data we have.

3.2.2 Dual-T estimator

Generally, the estimation error of the noisy label posterior is larger than the clean label posterior as label noise is introduced by the randomness specified by the transition matrix. This makes the learned mapping that fits label noise prone to overfitting and thus will lead to a large estimation error for the noisy class posterior. As discussed above, using the anchor point assumption requires the estimation of noisy label posterior, so a large estimation error of noisy label posterior can also lead to an inaccurate estimation for the transition matrix.

Dual-T estimator [6] is an approach that factorizes the transition matrix into two parts to reduce the estimation error of T by introducing an intermediate label. Given the intermediate label Y', the transition matrix could be factorized as

$$\begin{aligned} T^{dt}_{ij} &= P(\tilde{Y} = i|Y = j) \\ &= \sum_{l=1}^{C} P(\tilde{Y} = i|Y' = l, Y = j)P(Y' = l|Y = j) \\ &= \sum_{l=1}^{C} T^A_{il}(Y = j)T^B_{lj}, \end{aligned} \tag{10}$$

where $T^A = P(\tilde{Y}|Y', Y)$ represents the transition matrix from clean and intermediate label to noisy label, and $T^B = P(Y'|Y)$ represents the transition from clean label to intermediate label. T^A_{il} is the component of T^A at i-th row and l-th column, while T^B_{lj} is the component of T^B at l-th row and j-th column. Therefore, the transition matrix by dual-T estimator is

$$T^{dt} = T^A T^B. \tag{11}$$

Since T^B is in a form similar to the original transition matrix T, i.e., $P(\tilde{Y}|Y)$, it can be estimated by using the anchor point approach that was developed for estimating the original transition matrix T. However, the anchor point approach requires $P(Y'|X)$ to estimate T^B. Without a proper choice of the intermediate label, it could be hard to find $P(Y'|X)$ or even introduce extra error for the estimation of the transition matrix. One way to address this problem is to choose a specific intermediate label Y' such that $P(Y'|X) = \hat{P}(\tilde{Y}|X)$. In this way, T^B can then be easily derived from the estimated noisy class posterior by exploiting the noisy data at hand. Since we have direct access to $\hat{P}(\tilde{Y}|X)$, the estimation error of T^B would be zero if the anchor points are given. The estimation of T^B is

$$\hat{T}^B = \hat{T}^{ap}. \tag{12}$$

The estimation of T^A is much more complicated than T^B because three different labels, including clean, intermediate, and noisy labels, are involved in this matrix. Given the noisy label, the intermediate label can be assigned by using $P(Y'|X)$, but the clean class label is usually unavailable. However, assume the intermediate label is identical to the noisy label, then the information of the

clean class label is considered to be less informative than the intermediate label, i.e., it cannot provide further information to predict $\tilde{Y}$ if Y' is given. Hence, we can simply use the dependence only on the intermediate label to predict the noisy label.

$$T^A_{il}(Y=j) = P(\tilde{Y}=i|Y'=l, Y=j) = P(\tilde{Y}=i|Y'=l). \tag{13}$$

Based on this equation, T^A can then be estimated by counting the distribution of the labels. More formally,

$$\hat{T}^A_{il}(Y=j) = \hat{P}(\tilde{Y}=i|Y'=l) = \frac{\sum_m 1_{\{[\arg\max_k P(Y'=k|X=x_m)=l] \wedge [\tilde{Y}_m=j|X=x_i]\}}}{\sum_m 1_{\{\arg\max_k P(Y'=k|X=x_m)=l\}}}, \tag{14}$$

where 1 is an indicator function which equals 1 when the condition denoted by the subscript holds and 0 otherwise. x_m denotes the m-th data sample with the noisy label $\tilde{Y}_m$.

Note that although T^A can be obtained by Eq. (14) without the clean label, in practice, it is hard to find a classifier that can map the intermediate labels (or estimated noisy labels) exactly identical to the noisy labels. By introducing T^A, the estimation error for transition matrix T could be further reduced [6].

Dual-T has a lot of advantages and had been justified to outperform the basic T-estimator constructed under the anchor point assumption on both real-world and synthetic datasets [6]. Overall, it reduces the estimation error of the transition matrix in two ways. First, there is no estimation error for T^B. Second, by factorizing the transition matrix into two parts, the problem of estimating the noisy label posterior $P(\tilde{Y}|Y)$ can be transformed into a problem of fitting the noisy labels $P(\tilde{Y}=i|Y'=l)$. Since the noisy label posterior is a mapping of the features into a value in [0, 1] but the class labels are in the discrete set $\{1, 2, ..., C\}$, it is much easier to learn the class labels than the class posterior.

3.3 Classification models

Three classifiers to predict the noisy class, including the vanilla feed forward neural network (FFNN), convolution neural network (CNN), and residual neural network (ResNet), were adopted to construct the classifiers robust to label noise.

The formulated FFNN model has an architecture of five sequential hidden layers with decaying hidden sizes from 1024 to 32. The activation function is ReLU. Each hidden layer is followed by a batch normalization layer and a dropout layer with a dropout ratio of 0.1.

The formulated CNN has an architecture with three convolution layers and five linear layers. The convolution layers have a varying output channel size, while the kernel size = 3 × 3, stride = 1, padding = 1 or 0. Each convolution layer is followed by the batch normalization layer, ReLU activation and 2 × 2 max pooling with stride = 2. Each of the linear layers is followed by ReLU activation and drop out layer with a dropout ratio of 0.1. The same structure of CNN was used for all three datasets, while a tiny adjustment is applied for the CIFAR dataset with RGB input and different sizes. The input channel is set as 3 for the RGB data, and the input dimension for the first linear layer is adjusted for the data structure.

For the approach with ResNet, we have adopted the well-known ResNet 18 structure with the architecture shown in Figure 1 [31], with the output size of the last linear layer is modified to 3 in order to adapt to the problem setup.

All the models were trained for 4 epochs using the Adam optimizer, with an initial learning rate set to 0.001.

The cost function used is the cross-entropy loss, which is

$$l_{ce}(o) = -\sum_{c=1}^{C} y_{o,c} log(p_{o,c}), \tag{15}$$

where $y_{o,c}$ is a binary indicator in $\{0, 1\}$ and $y_{o,c} = 1$ if class label c is the correct classification for observation o. $p_{o,c}$ is the predicted probability that observation o is of class c.

As $\sum_{c=1}^{C} p_{o,c} = 1$, a higher $p_{o,c}$ for the correct prediction results in a lower cross-entropy loss l_{ce}. Therefore, minimizing l_{ce} leads us to a more accurate prediction from the classifier.

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure 1: The original ResNet 18 architecture. We have modified the output size of the fully connected layer equal to 3.

4 Experiments

4.1 Setup

4.1.1 Datasets

Three datasets were used to compare and evaluate the robustness of the proposed algorithms. The first two datasets were adapted from the Fashion MNIST dataset [32], while the third one was adapted from the CIFAR dataset [33]. For each dataset, the training and validation set were corrupted by class-conditional label noise with different flip rates, whereas the test set was clean. The transition matrices were only given for the first two Fashion MNIST datasets but not for the CIFAR dataset.

Fashion MNIST (Fashion Modified National Institute of Standards and Technology database) is a dataset of images collected from Zalando's article. Each image depicts an item of clothing or fashion accessory and has a size of 28 × 28 under grayscale. In this study, two corrupted versions of it with different flip rates were used. Both of them consist of a training-validation set of 18,000 examples and a test set of 3,000 examples with three labels in total. The transition matrices of the noisy flip rates were given by $T_{.5} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$ and $T_{.6} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.3 \\ 0.3 & 0.3 & 0.4 \end{bmatrix}$, respectively. Since the proportion of label noise for each class is fixed and equals 0.5 and 0.6 for each of the datasets, they are referred to as Fashion MNIST 0.5 and Fashion MNIST 0.6, respectively.

CIFAR (Canadian Institute For Advanced Research) is another dataset which contains a collection of 32 × 32 color images. Compared to the original version, the one used in this study contains only 15,000 examples and 3,000 examples for the training-validation set and test set, respectively. There are 3 different labels in total, but the associated flip rates are unknown.

For each dataset, 80% of the examples were randomly sampled from the training-validation set to train the model, and the rest 20% were for validation to prevent overfitting. The robustness of the model was evaluated using the test data without noisy labels. All the training, validation, and test set were divided into small batches of size 100 to improve the model efficiency.

4.1.2 Evaluation Metric

The performance of each classifier was evaluated using the top-1 accuracy which is defined as

$$\text{Top-1 accuracy} = \frac{\text{number of correctly classified examples}}{\text{total number of test examples}} \times 100\%. \tag{16}$$

An algorithm would have higher top-1 accuracy if it is robust to the label noise as high accuracy indicates a better prediction for the clean label.

To obtain a rigorous performance metric, each classifier would be trained by different training and validation sets using random sampling 10 times, and the resulting distribution would be reported in Section 4.2.

The transition matrix was estimated by the model with the highest validation accuracy under ten random train-validation splits. The anchor points are then estimated on training sets and the mean squared error (MSE) of the components between the estimation $\hat{T}$ and the ground truth T is used to evaluate the estimation error, i.e.,

$$MSE = \frac{1}{C^2}\|T - \hat{T}\|_F^2 = \frac{1}{C^2}\sum_{ij}(T_{ij} - \hat{T_{ij}})^2, \tag{17}$$

where C is the number of distinctive classes given in the dataset, and C^2 denotes the number of entries within the transition matrix.

4.2 Results and discussions

4.2.1 Fashion MNIST 0.5

Table 1 are the testing results of the Fashion MNIST 0.5 dataset. The average accuracy of the three models shows good robustness to the label noise. The small standard deviation of all three cases also implies they achieve a stable performance with different training sets. Among them, CNN has the highest accuracy, followed by ResNet and FFNN. This result is consistent with our expectation as CNN has a relatively better ability to capture the features of images and is less prone to shift and rotation compared to FFNN and ResNet.

Table 1: Testing results of Fashion MNIST 0.5 dataset

	Accuracy (mean)	Accuracy (std)
FFNN	0.8435	0.0195
CNN	0.9332	0.0000
ResNet	0.8699	0.0454

To verify the effectiveness of the transition matrix estimators, two methods proposed in Section 3.2 were applied to Fashion MNIST 0.5. The estimated result obtained from the anchor point assumption is $\hat{T}^{ap} = \begin{bmatrix} 0.6653 & 0.1544 & 0.2115 \\ 0.2097 & 0.5440 & 0.1199 \\ 0.1250 & 0.3016 & 0.6685 \end{bmatrix}$, and the one using dual-T estimator is $\hat{T}^{dt} = \begin{bmatrix} 0.4114 & 0.3011 & 0.2870 \\ 0.2761 & 0.3944 & 0.3001 \\ 0.3082 & 0.3092 & 0.4103 \end{bmatrix}$. The mean squared error (MSE) associated with each method is 0.009751 and 0.006648, respectively. As indicated by low MSE, both of them achieve a good estimation of the flip rates, whereas the dual-T estimator outperforms the anchor point approach with MSE reduced by 31.8%.

4.2.2 Fashion MNIST 0.6

Table 2 shows the testing results of the Fashion MNIST 0.6 dataset. Compared to those obtained in Fashion MNIST 0.5, the accuracy is decreased by 7 - 18% within different models, while CNN still has the highest accuracy among all. According to the transition matrices are given for the two Fashion MNIST datasets, the flip rate of each class in Fashion MNIST 0.6 is slightly higher than that in the Fashion MNIST 0.5 by 0.1. This explains the decrease in the overall accuracy for the former as the proportion of the label noise introduced to the dataset is increased. Even though, the robustness of the three models still remains in stable condition as their standard deviations are small.

Table 2: Testing results of Fashion MNIST 0.6 dataset

	Accuracy (mean)	Accuracy (std)
FFNN	0.7070	0.0308
CNN	0.8579	0.0278
ResNet	0.6950	0.0988

The transition matrix of Fashion MNIST 0.6 estimated using the anchor point assumption and dual-T estimator are $\hat{T}^{ap} = \begin{bmatrix} 0.4582 & 0.2893 & 0.3220 \\ 0.2666 & 0.4324 & 0.2961 \\ 0.2752 & 0.2783 & 0.3819 \end{bmatrix}$ and $\hat{T}^{dt} = \begin{bmatrix} 0.3453 & 0.3287 & 0.3325 \\ 0.3260 & 0.3437 & 0.3295 \\ 0.3287 & 0.3285 & 0.3375 \end{bmatrix}$, and the corresponding mean square error are 0.0008421 and 0.0016810, respectively. Both of the estimators still have a good estimation of the flip rates even when more label noise is introduced. However, the lower MSE of the dual-T estimator indicates a poorer performance compared to the anchor point approach. This inconsistency with the previous result may be attributed to the limited training examples available for all three datasets. As reported by [6], the superiority of the dual-T estimator does not also hold when the sample size is small. This is because the estimation of T^B is degraded by the decreasing number of examples per class. Therefore, the resulting transition matrix may have a larger estimation error compared to the anchor assumption.

4.2.3 CIFAR

The unknown transition matrix for the CIFAR dataset was estimated by the model with the highest validation of Top-1 accuracy. We use the training set to estimate the anchor points. After obtaining the estimated transition matrix by the two methods proposed in Section 3.2, the testing results using the estimated transition matrix are shown in Table 3.

The estimated transition matrix obtained using the anchor point assumption and dual-T estimators are $\hat{T}^{ap} = \begin{bmatrix} 0.4696 & 0.0958 & 0.3543 \\ 0.3438 & 0.5334 & 0.2170 \\ 0.1866 & 0.3708 & 0.4286 \end{bmatrix}$ and $\hat{T}^{dt} = \begin{bmatrix} 0.3494 & 0.3210 & 0.3274 \\ 0.3232 & 0.3574 & 0.3283 \\ 0.3268 & 0.3156 & 0.3480 \end{bmatrix}$, respectively. We first noticed that there is a gap between the results obtained from the two approaches even though both of them are proved to be able to achieve high accuracy with stable performance in the previous results.

Table 3: Testing results of CIFAR dataset

	Anchor point assumption		Dual-T estimator	
	Accuracy (mean)	Accuracy (std)	Accuracy (mean)	Accuracy (std)
FFNN	0.5182	0.0191	0.5435	0.0282
CNN	0.5736	0.0951	0.6384	0.0931
ResNet	0.5207	0.0877	0.5053	0.0666

As indicated by the accuracy scores shown in 3, models training under the CIFAR dataset are less robust to the label noise. The highest average accuracy is 63.84%, which is achieved by the CNN using a dual-T estimator. This degradation in the performance may result from the increase in data complexity as the CIFAR dataset contains RGB images of 3 input channels. Generally, the accuracy obtained by the dual-T estimator is higher than those of the anchor point approach, especially in CNN where the accuracy is improved by 6%. This indicates a better estimation of the transition matrix using a dual-T estimator as an accurate estimation of the transition matrix is the key to

deriving the target clean label posterior and boosting the classifier performance. However, limited by the sample size, the superiority of the dual-T estimator is not completely stable as indicated by its poorer estimation in ResNet.

5 Conclusion

The transition matrix is essential in dealing with label noise. In this study, we implement two methods to learn the transition matrix given a dataset and evaluate them by implementing 3 different algorithms. In general, with the same transition matrix, CNN performs best due to the characteristic of dealing with images. The poor performance of ResNet may be due to insufficient fine tuning of the ResNet 18 architecture. Additionally, for the same algorithm, the dual-T method sometimes achieves higher performance. However, the performance of this method is affected by the size of the training set. Overall, all algorithms show robustness to label noise, even though they are sensitive to the changes in the flip rates and the complexity of the data.

Currently, the transition matrix estimators work well, but additional requirements may be satisfied. The label noise is modelled under a Class-Dependent condition. In future work, we may compare different estimators and methods, and explore more complex noise models, such as Instance- and Label-Dependent Noise, as well as their applications. Besides, further work could be applied to the analysis of confidence levels on the noisy label. Instead of estimating the transition matrix, using data whose noisy labels are more similar to the clean labels could be a different approach to the noisy label classification problem.

References

[1] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[2] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[3] Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 68–83, 2018.

[4] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

[5] Irwin Bross. Misclassification in 2 x 2 tables. *Biometrics*, 10(4):478–486, 1954.

[6] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in neural information processing systems*, 33:7260–7271, 2020.

[7] Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167, 1999.

[8] Jiang-wen Sun, Feng-ying Zhao, Chong-jun Wang, and Shi-fu Chen. Identifying and correcting mislabeled training instances. In *Future generation communication and networking (FGCN 2007)*, volume 1, pages 244–250. IEEE, 2007.

[9] Taghi M Khoshgoftaar and Pierre Rebours. Generating multiple noise elimination filters with the ensemble-partitioning filter. In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004.*, pages 369–375. IEEE, 2004.

[10] Jaree Thongkam, Guandong Xu, Yanchun Zhang, and Fuchun Huang. Support vector machine for outlier detection in breast cancer survivability prediction. In *Asia-Pacific Web Conference*, pages 99–109. Springer, 2008.

[11] Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung. Data cleaning for classification using misclassification analysis. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 14(3):297–302, 2010.

[12] Sofie Verbaeten. Identifying mislabeled training examples in ilp classification problems. In *Proceedings of twelfth Belgian-Dutch conference on machine learning*, pages 1–8, 2002.

[13] Dragan Gamberger, Nada Lavrač, and Sašo Džeroski. Noise elimination in inductive concept learning: A case study in medical diagnosis. In *International Workshop on Algorithmic Learning Theory*, pages 199–212. Springer, 1996.

[14] Nada Matic, Isabelle Guyon, Léon Bottou, J Denker, and Vladimir Vapnik. Computer aided cleaning of large databases for character recognition. In *11th IAPR International Conference on Pattern Recognition. Vol. II. Conference B: Pattern Recognition Methodology and Systems*, volume 1, pages 330–331. IEEE Computer Society, 1992.

[15] Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Andres Folleco. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences*, 259:571–595, 2014.

[16] Naresh Manwani and PS Sastry. Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43(3):1146–1151, 2013.

[17] Mandayam AL Thathachar and Pidaparty S Sastry. *Networks of learning automata: Techniques for online stochastic optimization*. Springer Science & Business Media, 2003.

[18] Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.

[19] Joaquín Abellán and Andrés R Masegosa. Bagging decision trees on data sets with classification noise. In *International Symposium on Foundations of Information and Knowledge Systems*, pages 248–265. Springer, 2010.

[20] Choh Man Teng. Dealing with data corruption in remote sensing. In *International Symposium on Intelligent Data Analysis*, pages 452–463. Springer, 2005.

[21] Marıa-Gloria Basáñez, Clare Marshall, Hélène Carabin, Theresa Gyorkos, and Lawrence Joseph. Bayesian statistics for parasitologists. *TRENDS in Parasitology*, 20(2):85–91, 2004.

[22] Neil Lawrence and Bernhard Schölkopf. Estimating a kernel fisher discriminant in the presence of label noise. In *18th International Conference on Machine Learning (ICML 2001)*, pages 306–306. Morgan Kaufmann, 2001.

[23] Jakramate Bootkrajang and Ata Kabán. Multi-class classification in the presence of labelling errors. In *ESANN*, pages 345–350, 2011.

[24] Mattias Rantalainen and Chris C Holmes. Accounting for control mislabeling in case–control biomarker studies. *Journal of proteome research*, 10(12):5562–5567, 2011.

[25] Charles Bouveyron and Stéphane Girard. Robust supervised classification with mixture models: Learning from data with uncertain labels. *Pattern Recognition*, 42(11):2649–2658, 2009.

[26] Glenn Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

[27] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[28] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

[29] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017.

[30] Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *International conference on machine learning*, pages 125–134. PMLR, 2015.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[32] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[33] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

A Contribution

Work is equally allocated to each member of the group. The coding part was done by Ke Wang who was also responsible for the presentation and analysis of the final results in the report. Zijie did a lot of research work and completed the report writing in abstract, introduction, related works, and conclusion. All the other parts that have not been mentioned, such as methods and experimental setup, were finished by Hui-Yun who also participated in the discussion of the final results in collaboration with Ke Wang.

B Instruction on code

The code can be run sequentially by the cells in the jupyter notebook.