Safe Deep Semi-Supervised Learning for Unseen-Class Unlabeled Data

Lan-Zhe Guo [1] **Zhen-Yu Zhang** [1] **Yuan Jiang** [1] **Yu-Feng Li** [1] **Zhi-Hua Zhou** [1]

Abstract

Deep semi-supervised learning (SSL) has been recently shown very effectively. However, its performance is seriously decreased when the class distribution is mismatched, among which a common situation is that unlabeled data contains some classes not seen in the labeled data. Efforts on this issue remain to be limited. This paper proposes a simple and effective safe deep SSL method to alleviate the harm caused by it. In theory, the result learned from the new method is never worse than learning from merely labeled data, and it is theoretically guaranteed that its generalization approaches the optimal in the order $O(\sqrt{d\ln(n)/n})$, even faster than the convergence rate in supervised learning associated with massive parameters. In the experiment of benchmark data, unlike the existing deep SSL methods which are no longer as good as supervised learning in 40% of unseen-class unlabeled data, the new method can still achieve performance gain in more than 60% of unseen-class unlabeled data. Moreover, the proposal is suitable for many deep SSL algorithms and can be easily extended to handle other cases of class distribution mismatch.

1. Introduction

Deep neural networks have been reported to achieve competitive or even better performance than human beings in certain supervised learning tasks (LeCun et al., 2015). These tasks, however, all meet a basic condition, that is, have a large number of labeled training data. In many practical tasks, such condition is difficult to meet, as the acquisition of labeled data comes at a cost, which requires huge human and financial costs (Zhou, 2017; Oliver et al., 2018), limiting deep neural network in a broader field.

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. Correspondence to: Yu-Feng Li <liyf@lamda.nju.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

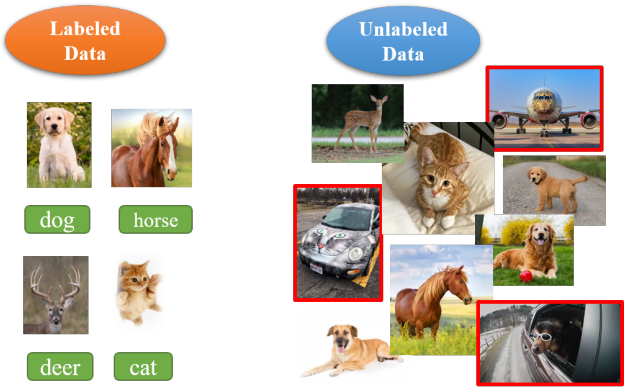

Figure 1. One example of class distribution mismatch. Unlabeled data contains classes that are not seen in the labeled data (indicated with red bounding boxes).

Deep semi-supervised learning (SSL) is proposed to utilize a large number of cheap unlabeled data to help deep neural networks improve performance, reducing the demand for labeled data. Deep SSL has been reported that it achieves highly competitive performance to the supervised learning model, which saves a lot of labeling costs, by exploring the structure of unlabeled data, such as introducing entropy minimization (Grandvalet & Bengio, 2005; Lee, 2013), consistency regularization (Sajjadi et al., 2016; Laine & Aila, 2017; Tarvainen & Valpola, 2017), adversarial training (Miyato et al., 2018) and other interesting techniques (Berthelot et al., 2019).

All of the above positive results, however, are based on a basic assumption that labeled data and unlabeled data come from the same distribution. Such an assumption is difficult to hold in many practical applications, among which one common case is that unlabeled data contains classes that are not seen in the labeled data. For example, in web page classification (Yang et al., 2011), unlabeled web pages crawled from the Internet according to keywords usually contain many categories that have not been seen before. In medical diagnosis (Yang et al., 2015), unlabeled medical images often contain different foci from the diseases to be diagnosed. In image classification, as shown in Figure 1, unlabeled images crawled from Internet/social networking sites according to keywords usually contain broader category concepts than labeled data. Faced with this type of

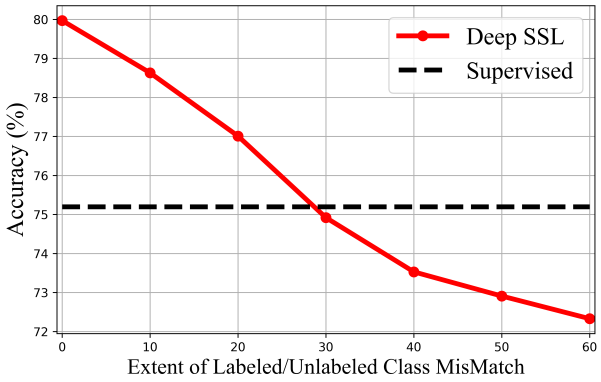

Figure 2. The performance of deep SSL decreases significantly as class mismatches between labeled and unlabeled data increase.

real data, deep SSL no longer works well and may even be accompanied by severe performance degradation (Oliver et al., 2018; Chen et al., 2020). That is, deep SSL is even worse than a simple supervised learning model, as illustrated in Figure 2. Such phenomena undoubtedly go against the expectation of deep SSL and limit its effectiveness in a large number of practical tasks. However, to our best knowledge, the efforts on this aspect remain to be limited.

Building a safe SSL, that is to say, SSL using extra unlabeled data will not be inferior to a simple supervised learning model, is the Holy Grail of SSL (Chapelle et al., 2006; Li & Zhou, 2015; Zhou, 2017). Since the problem was mentioned in (Cozman et al., 2003), there are some attempts (Singh et al., 2009; Li & Zhou, 2015; Loog, 2015; Li et al., 2017; Krijthe & Loog, 2017; Guo & Li, 2018). For example, (Li & Zhou, 2015) builds safe semi-supervised SVMs through optimizing the worst-case performance gain given a set of candidate low-density separators. (Loog, 2015) proposes to maximize the likelihood gain over a supervised model in the worst-case for generative models. (Balsubramani & Freund, 2015) proposes to learn a robust prediction given that the ground-truth label assignment is restricted to a specific candidate set. More introductions to safe SSL can be found in some recent summaries (Li & Liang, 2019; Mey & Loog, 2019; Li et al., 2019). The existing safe SSL, however, is unsuitable to the problem studied in the paper, because i) current safe SSL studies typically assume that labeled data and unlabeled data share the same distribution; ii) it works on shallow models such as SVM, Logistic Regression. In order to alleviate the performance degradation of deep SSL caused by class distribution mismatch, new proposals are desired.

To this end, this paper proposes a simple and effective safe deep SSL framework DS^{3}L (**D**eep **S**afe **S**emi-**S**upervised **L**earning). Unlike the existing deep SSL, DS^{3}L does not directly use all unlabeled data, but uses it selectively, and tracks the effect of the supervised learning model to prevent performance hazards. Specifically, on the one hand, DS^{3}L weakens unlabeled data with unseen classes, so as to improve the distribution matching to maintain strong generalization ability; on the other hand, it strengthens the labeled data to prevent performance degradation. The above considerations are cast as a whole into bi-level optimization (Bard, 2013) with efficient algorithms. The effectiveness of our proposal is demonstrated both theoretically and empirically. In theory, the result learned from the new method is never worse than learning from merely labeled data, and it is theoretically guaranteed that its generalization approaches the optimal in the order $O(\sqrt{d\ln(n)/n})$, even faster than the convergence rate in supervised learning associated with massive parameters. In the experiment of benchmark data, while the existing deep SSL methods are no longer as good as supervised learning in 40% of unseen-class unlabeled data, the new method can still achieve performance gain in more than 60% of unseen-class unlabeled data. Moreover, the proposal is suitable for any deep SSL algorithm and can be easily extended to handle other cases of data distribution mismatch.

2. Brief Introduction to Deep SSL

We first give a brief review to the standard deep SSL in this section. In the deep SSL task, we are given a set of training data from an unknown distribution, which includes n labeled instances $\mathcal{D}_l = \{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$ and m unlabeled instances $\mathcal{D}_u = \{\mathbf{x}_{n+1}, \cdots, \mathbf{x}_{n+m}\}$. Usually, $m \gg n$, $\mathbf{x} \in \mathcal{X} \in \mathbb{R}^D$, $\mathbf{y} \in \mathcal{Y} = \{1, \cdots, C\}$ where D is the number of input dimension and C is the number of output class in labeled data. The goal of deep SSL is to learn a model $h(\mathbf{x}; \theta) : \{\mathcal{X}; \Theta\} \to \mathcal{Y}$ parameterized by $\theta \in \Theta$ from training data to minimize the generalization risk $R(h) = \mathbb{E}_{(X,Y)}[\ell(h(X; \theta), Y)]$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ refers to certain loss function, e.g., mean squared error or cross entropy loss.

Generalization risk is hard to compute as the data distribution is unknown. The most classical approach is to approximate the generalization risk by minimizing the empirical risk on labeled data, i.e., $\min_{\theta\in\Theta} \hat{R}(h) = \sum_{i=1}^{n} \ell(h(\mathbf{x}_i; \theta), \mathbf{y}_i)$, however, obviously this way ignores the useful structure of unlabeled data. The way deep SSL utilizes unlabeled data structures is usually through the introduction of regularization, which is typically formulated as the following objective.

$$\min_{\theta\in\Theta} \sum_{i=1}^{n} \ell(h(\mathbf{x}_i; \theta), \mathbf{y}_i) + \Omega(\mathbf{x}; \theta) \quad \text{s.t. } \mathbf{x} \in \mathcal{D}_l \cup \mathcal{D}_u. \quad (1)$$

where $\Omega(\mathbf{x}; \theta)$ refers to the regularization term.

The design of the regularization term is the key (Oliver et al., 2018). Early SSL studies mainly used large margin regularization, Laplacian regularization, etc (Chapelle

et al., 2006). As the data augmentation continued to obtain practical effects, the adjustment of hyperparameters and network structure became increasingly critical, *consistency regularization* that forces the predictive results to have consistency under various disturbances, became more and more popular, acting as one of the most important regularization items in deep SSL. Generally, consistency regularization is characterized as

$$\Omega(\mathbf{x};\theta) = \|h(\text{perturb}(\mathbf{x});\theta) - h(\mathbf{x};\theta)\|_2^2 \tag{2}$$

perturb$(\mathbf{x})$ refers to certain stochastic operation. The implementation includes various domain-specific data perturbation strategies, such as rotation, shearing, and Gaussian noise, as well as various model-specific operations, such as dropout (Laine & Aila, 2017). To facilitate the computation and enhance the robustness, Mean-Teacher (Tarvainen & Valpola, 2017) replaces Eq.(2) with the output of an ensemble model using an exponential moving average of model parameters. VAT (Miyato et al., 2018) further improves it by computing an adversarial perturbation that maximally changes the output class distribution to the input.

On the other hand, as the predictive results are required to be closer to a priori, much attention has been paid to the *minimum entropy regularization* (Grandvalet & Bengio, 2005; Lee, 2013), which aims to prevent the class distribution of predictive results from being too flat and has no tendency. Formally, it is cast as following

$$\Omega(\mathbf{x};\theta) = -\sum_{c=1}^{C} h(\mathbf{x};\theta)_c \log(h(\mathbf{x};\theta)_c) \tag{3}$$

Obviously, minimum entropy regularization and consistent regularization could be further combined, and larger performance gain might be expected (Miyato et al., 2018).

Under the same distribution, that is, the unlabeled data and labeled data share the same distribution, deep SSL methods obtain significant performance gains for many benchmark tasks, reducing considerable labeling overhead. However, once the data distribution turns out to be different, such as the class distribution does not match as illustrated in Figure 1, deep SSL can easily fail and even severe performance degradation may occur. Its effect may be even worse than a simple supervised learning model (Oliver et al., 2018).

3. The Proposed DS^{3}L Framework

To alleviate the performance degradation caused by class distribution mismatch, we propose an effective safe deep SSL framework DS^{3}L. Different from the existing deep SSL which uses all unlabeled data, DS^{3}L uses it selectively and keeps tracking the effect of the supervised learning model to prevent performance hazards. Meanwhile, DS^{3}L uses beneficial unlabeled data as much as possible to improve generalization performance, preventing performance gains from being too conservative. In this section, we first give the DS^{3}L framework with an efficient algorithm and its complexity analysis, then the effectiveness of our proposal is demonstrated theoretically.

3.1. Framework Formulation

On one hand, DS^{3}L uses the unlabeled selectively. The main methodology is to design a weighting function $w : \mathbb{R}^D \to \mathbb{R}$ parameterized by $\alpha \in \mathbb{B}^d$ that maps an instance to a weight. Then, DS^{3}L tries to find the optimal $\hat{\theta}(\alpha)$ that minimizes the corresponding weighted empirical risk,

$$\hat{\theta}(\alpha) = \min_{\theta\in\Theta} \sum_{i=1}^{n} \ell(h(\mathbf{x}_i;\theta), \mathbf{y}_i) + \sum_{i=n+1}^{n+m} w(\mathbf{x}_i;\alpha)\Omega(\mathbf{x}_i;\theta) \tag{4}$$

where $\hat{\theta}(\alpha)$ is denoted as the model trained with the weight function parameterized by α.

On the other hand, DS^{3}L keeps tracking supervised performance to prevent performance degradation. Specifically, DS^{3}L requires that the model returned by the weighted empirical risk process should maximize the generalization performance, i.e.,

$$\alpha^* = \operatorname*{argmin}_{\alpha\in\mathbb{B}^d} \mathbb{E}_{(X,Y)}[\ell(h(X;\hat{\theta}(\alpha)), Y)] \tag{5}$$

In real practice the distribution is unknown, similar to the empirical risk minimization, DS^{3}L tries to find the optimal parameters $\hat{\alpha}$ such that the model returned by optimizing the weighted instance loss, should also have good performance on the labeled data which acts as a unbiased and reliable estimation of the underlying distribution, i.e.,

$$\hat{\alpha} = \operatorname*{argmin}_{\alpha\in\mathbb{B}^d} \sum_{i=1}^{n} \ell(h(\mathbf{x}_i;\hat{\theta}(\alpha)), \mathbf{y}_i) \tag{6}$$

To simplify the notation, we denote $\hat{\theta}(\alpha)$ as $\hat{\theta}$. Taking both the Eq.(4) and Eq.(6) into consideration, the objective of our framework can be formulated as the following bi-level optimization problem,

$$\begin{aligned} &\min_{\alpha\in\mathbb{B}^d} \sum_{i=1}^{n} \ell(h(\mathbf{x}_i;\hat{\theta}), \mathbf{y}_i) \\ &\text{s.t.} \\ &\hat{\theta} = \operatorname*{argmin}_{\theta\in\Theta} \sum_{i=1}^{n} \ell(h(\mathbf{x}_i;\theta), \mathbf{y}_i) + \sum_{i=n+1}^{n+m} w(\mathbf{x}_i;\alpha)\Omega(\mathbf{x}_i;\theta) \end{aligned} \tag{7}$$

Eq.(7) can be understood by two stages: first, DS^{3}L seeks the optimal model parameter $\hat{\theta}$ via the weighted empirical risk minimization, then evaluates it on n labeled instances and optimizes the weight function parameter α to make the learned $\hat{\theta}$ to achieve a better reliable performance. Figure 3 illustrates the DS^{3}L framework.

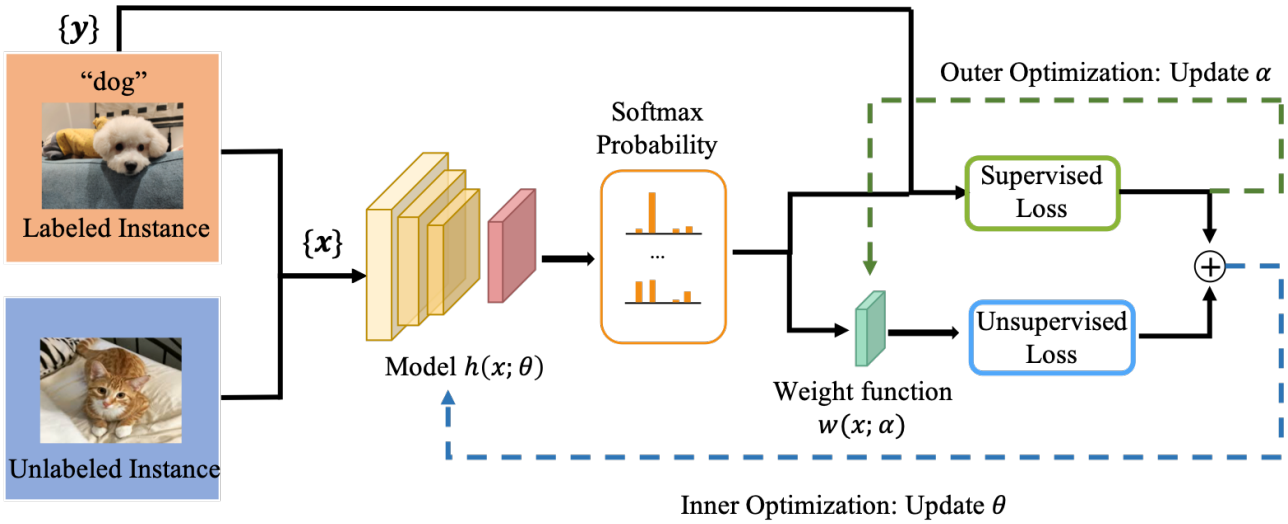

Figure 3. Illustration of the DS^{3}L framework.

3.2. Optimization Algorithm and Complexity Analysis

Eq.(7) is a bi-level optimization problem (Bard, 2013), where one optimization problem is nested within another problem. The inner-level optimization is to find a weighted empirical risk minimizer model given the training set whereas the outer-level optimization is to minimize the supervised loss given the learned model. For the sake of simplicity, we denote the outer-level objective as $\mathcal{L}^{outer}(\theta)$ and the inner-level objective as $\mathcal{L}^{inner}(\theta, \alpha)$.

Indeed, in general, there is no closed-form expression of θ, so it is not possible to directly optimize the upper-level objective function. The classical approaches for solving bi-level optimization problems can be categorized as single-level reduction methods, descent methods and evolutionary methods (Sinha et al., 2018). However, these methods are usually inefficient to handle the big data and complex learning models. To meet the efficiency requirement of the deep model, we adopt the online approximation based optimization method proposed in (Ren et al., 2018). We further give a more general analysis based on a general weight learning function instead of directly optimizing the instance weight.

In general cases, we adopt gradient descent methods (or one of its variants like momentum, RMSProp, Adam, etc.) to solve the optimal $\hat{\theta}$ approximately. Specifically, the training procedure can be written as:

$$\theta_{t+1} = \theta_t - \eta_\theta \nabla_\theta \mathcal{L}^{inner}(\theta_t, \alpha) \tag{8}$$

η_θ is the learning rate for θ, and t indicates the t-th iteration.

After learned the optimal model $\hat{\theta}$, we compute the supervised loss and then update the weight parameter α:

$$\alpha_{t+1} = \alpha_t - \eta_\alpha \nabla_\alpha \mathcal{L}^{outer}(\hat{\theta}) \tag{9}$$

However, calculating the optimal α requires two nested loops of optimization, i.e., we need to compute the optimal parameter $\hat{\theta}$ for each α_t which needs $T \times T$ round iterations. This is time inefficient and can not handle large-scale data sets and deep models. To further accelerate the optimization, we propose an approximate alternating optimization method by updating α and θ iteratively.

Updating θ. Once given the parameter α_t of weight function w, the updated θ_{t+1} can be simply optimized as the common single-level optimization

$$\theta_{t+1} = \theta_t - \eta_\theta \nabla_\theta \mathcal{L}^{inner}(\theta_t, \alpha_t) \tag{10}$$

Updating α. After receiving the parameter θ_{t+1} (an approximation of $\hat{\theta}$), we can calculate the outer objective, and update α through

$$\alpha_{t+1} = \alpha_t - \eta_\alpha \nabla_\alpha \mathcal{L}^{outer}(\theta_{t+1}) \tag{11}$$

The main difficulty in Eq.(11) is to solve the bi-level gradient $\nabla_\alpha \mathcal{L}^{outer}(\theta_{t+1})$ as α is explicitly beyond the outer objective. According to the chain rule, we have,

$$\begin{aligned} &\nabla_\alpha \mathcal{L}^{outer}(\theta_{t+1}) \\ &= \nabla_\alpha \mathcal{L}^{outer}(\theta_t - \eta_\theta \nabla_\theta \mathcal{L}^{inner}(\theta_t, \alpha_t)) \\ &= \nabla_\theta \mathcal{L}^{outer}(\theta_t)(-\eta_\theta \nabla_\alpha \nabla_\theta \mathcal{L}^{inner}(\theta_t, \alpha_t)) \end{aligned} \tag{12}$$

In real practice, we can leverage automatic differentiation techniques to compute the gradient of $\mathcal{L}^{outer}(\theta_{t+1})$ w.r.t. α_t. The optimization can be easily implemented using popular deep learning frameworks such as Pytorch [1] or Tensorflow [2]. The overall algorithm is summarized in Algorithm 1 and the main computation flowchart is plotted in Figure 4.

[1] https://pytorch.org/
[2] www.tensorflow.org

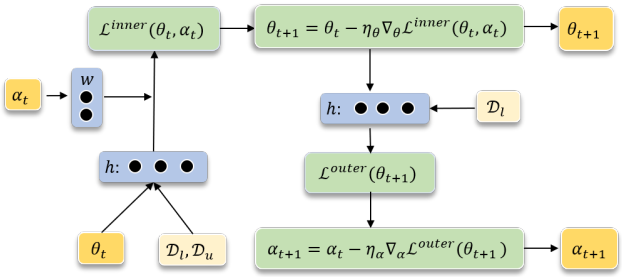

Figure 4. Main flowchart of the proposed DS^{3}L.

Complexity Compared with regular optimization on a single-level problem, the new method can be regarded as requiring an extra forward and backward passes of the classifier network and an extra forward and backward passes of the weight function to compute the bi-level gradient. Therefore, compared with the regular training procedures of deep SSL, DS^{3}L needs approximately 3 $\times$ training time.

We further analyze the convergence of optimization process in DS^{3}L and derive the follow theorem,

Theorem 1. *(Convergence.) Suppose the supervised loss function is Lipschitz-smooth with constant $L \leq 2$, and the supervised loss and unsupervised loss have ρ-bounded gradients, then by following our optimization algorithm, the labeled loss always monotonically decreases along with the iteration t, i.e.,*

$$\mathcal{L}^{outer}(\theta_{t+1}) \leq \mathcal{L}^{outer}(\theta_t) \tag{13}$$

Furthermore, the equality in Eq.(13) holds only when the gradient of the outer objective respect to α becomes 0 at some iteration t, i.e.,

$$\mathcal{L}^{outer}(\theta_{t+1}) = \mathcal{L}^{outer}(\theta_t)$$

if and only if

$$\nabla_\alpha \mathcal{L}^{outer}(\theta_t) = 0$$

Moreover, we can prove the convergence rate of our optimization method to be $\mathcal{O}(1/\epsilon^2)$.

Theorem 2. *(Convergence Rate.) Suppose the aforementioned conditions hold, and let the step size η_θ satisfy $\eta_\theta = \min\{1, \frac{k}{T}\}$ for some constant $k > 0$, such that $\frac{k}{T} < 1$ and $\eta_\alpha = \min\{\frac{1}{L}, \frac{C}{\sqrt{T}}\}$ for some constant $C > 0$, such that $\frac{\sqrt{T}}{C} \leq L$. Then, the approximation algorithm can achieve $\mathbb{E}[\|\nabla_\alpha \mathcal{L}^{outer}(\theta_t)\|_2^2] \leq \epsilon$ in $\mathcal{O}(1/\epsilon^2)$. More specifically,*

$$\min_{0 \leq t \leq T} \mathbb{E}[\|\nabla_\alpha \mathcal{L}^{outer}(\theta_t)\|_2^2] \leq \mathcal{O}(\frac{C}{\sqrt{T}})$$

where C is some constant independent to the convergence process.

Compared with the analysis procedure in (Ren et al., 2018), we obtain the same convergence results with a more general function class.

Algorithm 1 The DS^{3}L Learning Framework

Input: Labeled data $\mathcal{D}_l = \{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$, unlabeled data $\mathcal{D}_u = \{\mathbf{x}_{n+1}, \cdots, \mathbf{x}_{n+m}\}$, max iterations T.
Output: Learned weight function parameter α_T and model parameter θ_T.

Initialize weights function parameter α_0 and model parameter θ_0,
for $t = 0$ to $T - 1$ **do**
  $\{\mathbf{x}, \mathbf{y}\} \leftarrow$ SampleBatchLabeledData($\mathcal{D}_l$).
  $\{\mathbf{x}\} \leftarrow$ SampleBatchUnlabeledData($\mathcal{D}_u$).
  Compute training loss: $\mathcal{L}^{inner}(\theta_t, \alpha_t)$.
  Update model: $\theta_{t+1} = \theta_t - \eta_\theta \nabla_\theta \mathcal{L}^{inner}(\theta_t, \alpha_t)$.
  Compute supervised loss: $\mathcal{L}^{outer}(\theta_{t+1})$.
  Compute gradient: $\nabla_\alpha \mathcal{L}^{outer}(\theta_{t+1})$.
  Update weight: $\alpha_{t+1} = \alpha_t - \eta_\alpha \nabla_\alpha \mathcal{L}^{outer}(\theta_{t+1})$.
end for

3.3. Theoretical Studies

We first describe the superiority of DS^{3}L over supervised learning method and previous SSL methods intuitively,

Compared with supervised methods. Supervised methods that simply optimize θ on labeled data can lead to unsatisfactory performance as the labeled data is too few to learn the high-dimensional θ well, whereas, in DS^{3}L, labeled data is sufficient to learn a good α which can be constructed to be low-dimensional.

Compared with previous SSL methods. Previous SSL methods that treat all unlabeled instances equally can lead to performance degradation as unlabeled instances with unseen classes could hurt performance, whereas, in DS^{3}L, the unlabeled data is used selectively according to labeled data performance that can help achieve safe performance.

Then, in order to show the safeness of DS^{3}L, we analyze the empirical risk of DS^{3}L compared with simple supervised method and obtain the following theorem,

Theorem 3. *(Safeness.) Let θ^{SL} be the supervised model, i.e., $\theta^{SL} = \arg\min_{\theta \in \Theta} \sum_{i=1}^{n} \ell(h(\mathbf{x}_i; \theta), \mathbf{y}_i)$. Define the empirical risk as:*

$$\hat{R}(\theta) = \frac{1}{n} \sum_{i=1}^{n} [\ell(h(\mathbf{x}_i; \theta), \mathbf{y}_i)]$$

Then we have the empirical risk of $\hat{\theta}$ returned by DS^{3}L to be never worse than θ^{SL} that is learned from merely labeled data, i.e., $\hat{R}(\hat{\theta}) \leq \hat{R}(\theta^{SL})$.

Theorem 3 reveals that compared with previous deep SSL methods, DS^{3}L can achieve safeness in terms of empirical risk, i.e., the performance is not worse than its supervised counterpart, with the learned α.

We further analyze the generalization risk of DS^{3}L based on (Zhao et al., 2019) to better understand the effect of the parameter dimension and the size of labeled data to α and drive the following theorem,

Theorem 4. *(Generalization.) Assume that the loss function is λ-Lipschitz continuous w.r.t. α. Let $\alpha \in \mathbb{B}^d$ be the parameter of example weighting function w in a d-dimensional unit ball. Let n be the labeled data size. Define the generalization risk as:*

$$R(\theta) = \mathbb{E}_{(X,Y)}[\ell(h(X;\theta), Y)]$$

Let $\alpha^* = \arg\max_{\alpha \in \mathbb{B}^d} R(\hat{\theta}(\alpha))$ be the optimal parameter in the unit ball, and $\hat{\alpha} = \arg\max_{\alpha \in \mathcal{A}} \hat{R}(\hat{\theta}(\alpha))$ be the empirically optima among a candidate set $\mathcal{A}$. With probability at least $1 - \delta$ we have,

$$R(\hat{\theta}(\alpha^*)) \leq R(\hat{\theta}(\hat{\alpha})) + \frac{(3\lambda + \sqrt{4d\ln(n) + 8\ln(2/\delta)})}{\sqrt{n}}$$

Theorem 4 establishes that DS^{3}L approaches the optimal weight in the order $O(\sqrt{d\ln(n)/n})$. It is noteworthy that as stated in Theorem 20.6 in (Shalev-Shwartz & Ben-David, 2014), training an optimal deep supervised model θ on labeled data is in the order $O(\sqrt{D\ln(D)\ln(n)/n})$ (here D denotes the number of parameters in θ). Note that the dimension d (usually in hundreds) is much smaller than D (usually in millions for deep neural networks), which concludes that DS^{3}L enjoys a faster convergence rate than supervised deep learning based on massive parameters.

In summary, based on theorem 3 and theorem 4, from both the safeness and generalization, it is reasonable to expect that DS^{3}L achieves better generalization performance compared with baseline supervised learning methods.

4. Experiments

To validate the effectiveness of the proposed method, we conduct experiments on two standard MNIST and CIFAR benchmarks for semi-supervised image classification using deep convolutional neural networks (CNNs).

4.1. MNIST Handwritten Digit Recognition Task

MNIST is a standard dataset for handwritten digit classification tasks which includes $60,000$ training images of size 28×28 and $10,000$ test images. The data set contains 10 classes: digit "1" to digit "10". Specifically, we select 10 images per class from classes 1-6 to construct the labeled data set, i.e., 60 labeled data in total, and $30,000$ images from classes 110 as unlabeled data. We vary the ratio of unlabeled images from 1-6 to modulate class distribution mismatch. For example, when the extent of labeled/unlabeled class mismatch ratio is 0%, all unlabeled data comes from classes 1-6 while the extent is 50% means half of the unlabeled data comes from classes 1-6 and the others come from 7-10.

DS^{3}L is compared with the following state-of-the-art deep SSL methods,

- Pesudo-Labeling (Lee, 2013): Pseudo-labeling proceeds by producing "pseudo-labels" for the unlabeled data using the prediction function itself over the course of training. Pseudo labels with a class probability greater than a predefined threshold are used as the target labels, for the unlabeled data in the standard supervised loss function.
- Π-Model (Laine & Aila, 2017; Sajjadi et al., 2016): Π-Model adopts the consistency regularization and adds a loss term that encourages the distance between the prediction for an unlabeled instance and its stochastic perturbation (e.g., data augmentation, random noise) to be small.
- Temporal Ensembling (Laine & Aila, 2017): Instead of the stochastic perturbation in Π-Model, Temporal ensembling adopts the ensemble of predictions as the target for the unlabeled instances during the training process to produce a more stable performance.
- Mean Teacher (Tarvainen & Valpola, 2017): Mean teacher further improves the target quality for unlabeled instances by setting the target via an exponential moving average of parameters from previous training steps.
- Virtual Adversarial Training (VAT) (Miyato et al., 2018): Instead of relying on the stochastic perturbation of unlabeled instances, VAT aims to find adversarial disturbances that most affect the output of the prediction function.

Moreover, we also compare with the supervised learning method that simply trains a deep neural network on the small labeled data set as the baseline method.

We adopt a two-layer CNN model as our classifier network which contains two conv2ds with size $1 \times 16 \times 3$ and $16 \times 32 \times 3$, and two MaxPool2ds with size 3, stride 2 and padding 1, and adopt ReLU as the activate function (Goodfellow et al., 2016). The networks in all compared methods are trained using SGD with a momentum 0.9, a weight decay 5×10^{-4} and a learning rate 0.01. We train the model for 200,000 updates with a batch size of 100. The base deep SSL algorithm adopted in DS^{3}L is the same as the simplest Π-model. The experiments show that our proposal has already achieved performance gains over compared methods in a clear margin with Π-model. It is worth noting that the

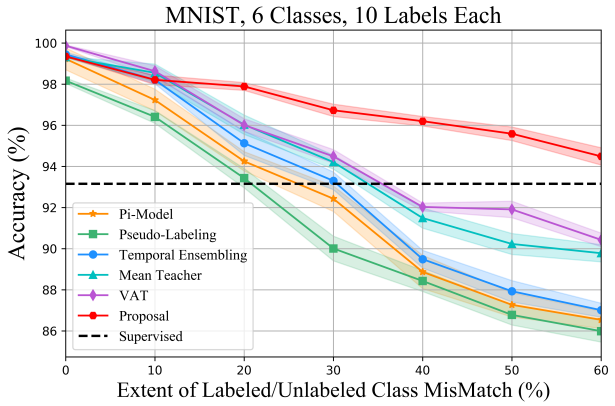

Figure 5. Classification accuracy of compared deep SSL techniques and DS^{3}L on MNIST data set (class $1-6$) with varying class mismatch ratio between labeled and unlabeled data. Shaded regions indicate standard deviation over five runs.

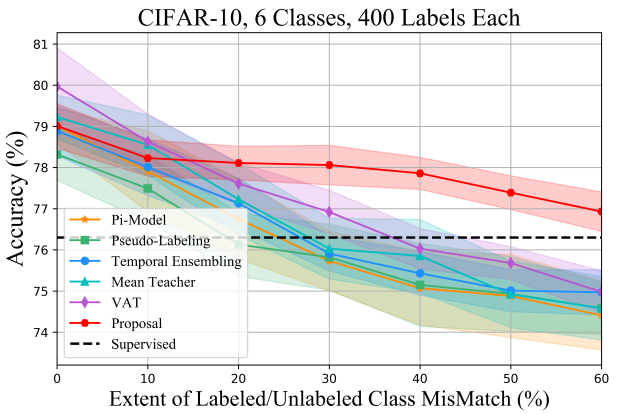

Figure 6. Classification accuracy of compared deep SSL techniques and DS^{3}L on CIFAR-10 data set with varying class mismatch ratio between labeled and unlabeled data. Shaded regions indicate standard deviation over five runs.

performance of our method can be further improved by incorporating more advanced deep SSL algorithms and more refined parameter optimization.

The averaged accuracy of deep SSL methods over 5 runs v.s. the extent of labeled/unlabeled class mismatch is plotted in Figure 5. From Figure 5, we can find that all the methods are clearly better than the baseline supervised learning method with the same class distribution. However, with the aggravation of class distribution mismatch, the performance of the existing deep SSL method decreases rapidly. Many deep SSL techniques are even inferior to the baseline supervised learning method when 40% of the unlabeled instances come from unseen classes, whereas our DS^{3}L can still maintain clear performance improvement in presence of more than 60% of unseen-class unlabeled instances, i.e., the irrelevant unlabeled instances are even more than the relevant ones. These empirical results in line with the theoretical analysis and demonstrate the effectiveness of DS^{3}L.

4.2. CIFAR Image Classification Task

CIFAR-10 is a benchmark for image classification tasks which consists of $60,000$ natural images of size of 32×32 as the training data and $10,000$ as test data. The data set contains 10 categories: "airline", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "trunk". In our experiments, we perform a 6-class classification task on animal classes (bird, cat, deer, dog, frog, horse) and select 400 images per class to construct the labeled data set, i.e., 2,400 labeled examples. Meantime, 20,000 images are randomly selected from all the 10 classes as the unlabeled data set. Again we vary the ratio of unlabeled images from the other four classes to modulate class distribution mismatch, following the experimental setup on MNIST.

For CIFAR-10, we adopt a Wide ResNet-28-10 (Zagoruyko & Komodakis, 2016) as our classifier network. In addition, we apply global contrast normalization and ZCA-normalized the inputs using statistics calculated on the CIFAR-10 training set. ZCA normalization is a widely-used preprocessing step for CIFAR-10. We also adopt data augmentation techniques including random horizontal flipping, random translation by up to 2 pixels, and Gaussian input noise with a standard deviation 0.15. Like the experiment on MNIST, we adopt Π-model as the base SSL algorithm. We train the network for 500,000 updates with a batch size of 100. We adopt Adam as the optimization algorithm with the initial learning rate $3e^{-4}$ and weight decay factor 0.2 after 400,000 iterations.

The experimental results are shown in Figure 6. We can observe a similar result, that is, our method achieves satisfactory performance under different degrees of class distribution mismatch. Unlike many deep SSL techniques that are inferior to baseline supervised learning method with 40% unseen-class unlabeled data, DS^{3}L achieves the best performance with the simple unsupervised regularization term even in more than 60% class mismatch ratio. All these results demonstrate that our proposed DS^{3}L is very effective against the harm caused by class distribution mismatch.

4.3. Universality for Various Deep SSL

Previous results reveal that the proposed DS^{3}L conducted on one simple deep SSL model achieves promising performance with varying class mismatch ratio between labeled and unlabeled data. To further demonstrate the flexibility of DS^{3}L that can be used for any deep SSL model, we report the results of DS^{3}L incorporated with four kinds of deep SSL methods (i.e., Π-Model, Temporal Ensembling, Mean Teacher and VAT) in Figure 7 and Figure 8. The

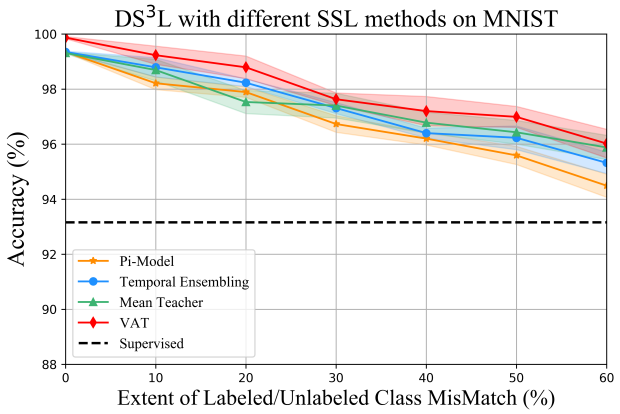

Figure 7. Classification accuracy of DS^{3}L incorporated with four deep SSL methods on MNIST data set.

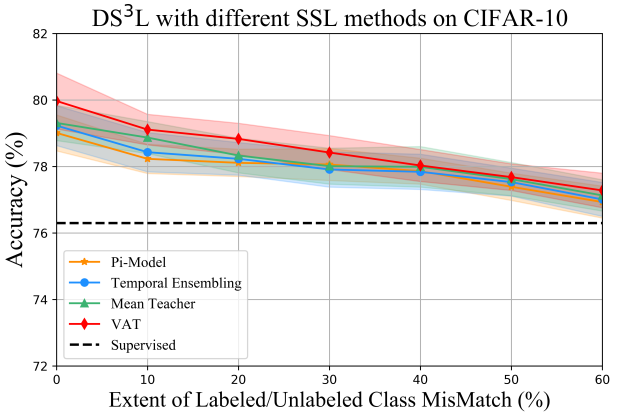

Figure 8. Classification accuracy of DS^{3}L incorporated with four deep SSL methods on CIFAR-10 data set.

experimental setup is the same as we previously describe. We further verify that the DS^{3}L achieves safe performance with all kinds of deep SSL methods, that is, performs superior to the baseline supervised learning method in all cases. This result demonstrates the flexibility of DS^{3}L.

4.4. Unseen-Class Unlabeled Data Identification

To further quantify the identification ability of our method in unseen-class unlabeled data, our proposal is compared with the probability estimation method on MNIST and CIFAR-10 data sets. Similar to Pseudo-Label, the probability estimation method (Hendrycks & Gimpel, 2017) uses the labeled data to get the class distribution of each unlabeled data, and then calculates the probability of belonging to the known classes through softmax. Examples with low predicted probability can be treated as unseen-class unlabeled examples. The AUC value can be used to measure the identification ability, by treating the unseen-class unlabeled data as a negative class and the others as a positive one. Table 1 shows the experimental results under different class mismatch ratios. It can be seen that compared with the probabilities based method our proposal consistently reduces the misclassification rate in unseen-class unlabeled data identification.

Table 1. (1-AUC)% for unseen-class data identification.

Data set	Ratio	Probabilities	DS^{3}L
MNIST	0.1	4.33 ± 0.29	$\mathbf{1.67 \pm 0.04}$
	0.2	4.78 ± 0.41	$\mathbf{0.53 \pm 0.23}$
	0.3	4.57 ± 0.33	$\mathbf{1.19 \pm 0.19}$
	0.4	4.73 ± 0.35	$\mathbf{1.50 \pm 0.20}$
	0.5	5.67 ± 0.43	$\mathbf{2.31 \pm 0.13}$
	0.6	7.32 ± 0.51	$\mathbf{3.57 \pm 0.32}$
CIFAR-10	0.1	7.69 ± 0.67	$\mathbf{4.37 \pm 0.48}$
	0.2	7.99 ± 0.63	$\mathbf{5.34 \pm 0.41}$
	0.3	7.67 ± 0.72	$\mathbf{5.33 \pm 0.43}$
	0.4	8.37 ± 0.75	$\mathbf{5.19 \pm 0.47}$
	0.5	9.77 ± 0.88	$\mathbf{6.51 \pm 0.39}$
	0.6	15.03 ± 1.03	$\mathbf{10.47 \pm 0.78}$

5. Conclusion

In this paper we tackle an important problem of deep SSL, that is, performance degradation in the presence of unseen-class in the unlabeled data. We propose a novel safe deep SSL framework DS^{3}L. The effectiveness of our proposal is demonstrated both theoretically and empirically. In theory, the new model is never worse than learning from merely labeled data in term of the empirical risk, and the convergence rate to its optimal generalization is faster than supervised learning with a large number of parameters. Empirical studies show that, unlike many deep SSL methods which are inferior to supervised learning in 40% of the unseen-class unlabeled data, the new method can still achieve performance gain in more than 60% of the unseen-class unlabeled data, which is in line with the theoretical results. The proposal is flexible to various deep SSL algorithms and other cases of class distribution mismatch.

There may be many possible studies in the future, for examples, new data types such as tabular data (Shavitt & Segal, 2018) and new deep models such as deep forest (Zhou & Feng, 2017). Beyond this work, it is also worthwhile to build a more complete theoretical and methodological framework for weakly supervised learning (Zhou, 2017). In addition, the integration of this research with the open environment is also very interesting, consist of many research problems.

The code for the work is readily available and freely downloaded at https://www.lamda.nju.edu.cn/code_DS3L.ashx.

Acknowledgements

This research was supported by the NSFC (61772262, 61673201, 61921006) and the Fundamental Research Funds for the Central Universities (14380061).

References

Balsubramani, A. and Freund, Y. Optimally combining classifiers using unlabeled data. In *Proceedings of the 28th Conference on Learning Theory*, pp. 211–225, 2015.

Bard, J. F. *Practical BiLevel Optimization: Algorithms and Applications*. Springer Science & Business Media, 2013.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 5050–5060, 2019.

Chapelle, O., Scholkopf, B., and Zien, A. *Semi-Supervised Learning*. MIT Press, 2006.

Chen, Y., Zhu, X., Li, W., and Gong, S. Semi-supervised learning under class distribution mismatch. In *The 34th AAAI Conference on Artificial Intelligence*, pp. 3569–3576, 2020.

Cozman, F. G., Cohen, I., and Cirelo, M. C. Semi-supervised learning of mixture models. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 99–106, 2003.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT press, 2016.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pp. 529–536, 2005.

Guo, L.-Z. and Li, Y.-F. A general formulation for safely exploiting weakly supervised data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 3126–3133, 2018.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Krijthe, J. H. and Loog, M. Projected estimators for robust semi-supervised classification. *Machine Learning*, 106 (7):993–1008, 2017.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, pp. 2–8, 2013.

Li, Y.-F. and Liang, D.-M. Safe semi-supervised learning: A brief introduction. *Frontiers Computer Science*, 13(4): 669–676, 2019.

Li, Y.-F. and Zhou, Z.-H. Towards making unlabeled data never hurt. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1):175–188, 2015.

Li, Y.-F., Zha, H.-W., and Zhou, Z.-H. Learning safe prediction for semi-supervised regression. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp. 2217–2223, 2017.

Li, Y.-F., Guo, L.-Z., and Zhou, Z.-H. Towards safe weakly supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. In Press, 2019.

Loog, M. Contrastive pessimistic likelihood estimation for semi-supervised classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):462–475, 2015.

Mey, A. and Loog, M. Improvability through semi-supervised learning: A survey of theoretical results. *arXiv preprint arXiv:1908.09574*, 2019.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (8):1979–1993, 2018.

Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., and Goodfellow, I. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 3235–3246, 2018.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 4331–4340, 2018.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 1163–1171, 2016.

Shalev-Shwartz, S. and Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Shavitt, I. and Segal, E. Regularization learning networks: Deep learning for tabular datasets. In *Advances in Neural Information Processing Systems*, pp. 1386–1396, 2018.

Singh, A., Nowak, R., and Zhu, J. Unlabeled data: Now it helps, now it doesn't. In *Advances in Neural Information Processing Systems*, pp. 1513–1520, 2009.

Sinha, A., Malo, P., and Deb, K. A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2018.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pp. 1195–1204, 2017.

Yang, H., Zhu, S., King, I., and Lyu, M. R. Can irrelevant data help semi-supervised learning, why and how? In *Proceedings of the 20th ACM Conference on Information and Knowledge Management*, pp. 937–946, 2011.

Yang, H., Huang, K., King, I., and Lyu, M. R. Maximum margin semi-supervised learning with irrelevant data. *Neural Networks*, 70:90–102, 2015.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *Proceedings of the British Machine Vision Conference 2016*, 2016.

Zhao, S., Fard, M. M., Narasimhan, H., and Gupta, M. R. Metric-optimized example weights. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 7533–7542, 2019.

Zhou, Z. and Feng, J. Deep forest: Towards an alternative to deep neural networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3553–3559, 2017.

Zhou, Z.-H. A brief introduction to weakly supervised learning. *National Science Review*, pp. 44–53, 2017.