Semi-Supervised Learning with Support Isolation by Small-Paced Self-Training

Zheng Xie, Hui Sun, Ming Li
Nanjing University,
Nanjing 210023, China
{xiez,sunh,lim}@lamda.nju.edu.cn

Abstract

In this paper, we address a special scenario of semi-supervised learning, where the label missing is caused by a preceding filtering mechanism, i.e., an instance can enter a subsequent process in which its label is revealed *if and only if* it passes the filtering mechanism. The rejected instances are prohibited to enter the subsequent labeling process due to economical or ethical reasons, making the support of the labeled and unlabeled distributions isolated from each other. In this case, semi-supervised learning approaches which rely on certain coherence of the labeled and unlabeled distribution would suffer from the consequent distribution mismatch, and hence result in poor prediction performance. In this paper, we propose a Small-Paced Self-Training framework, which iteratively discovers labeled and unlabeled instance subspaces with bounded Wasserstein distance. We theoretically prove that such a framework may achieve provably low error on the pseudo labels during learning. Experiments on both benchmark and pneumonia diagnosis tasks show that our method is effective.

1 Introduction

Semi-supervised learning [1, 2], which aims to alleviate the huge cost of collecting labeled data by exploiting the relatively large amount of unlabeled data, is raised from real-world demands. Existing approaches utilize the unlabeled data for better modeling the data distribution through different ways, increasing the capability of semi-supervised learning in various scenarios [3–7].

Differing from semi-supervised learning, which usually considers the labeled data is sampled from the population distribution with no or neglectable shift, in specific situations, labeled data may be sampled from a different distribution other than the test distribution, and unlabeled data sampled from the test distribution can be used for improving learning. For example, domain adaptation [8–10] aims to build models with labeled and unlabeled data from two different but related (source and target) domains. Learning under sample selection bias [11, 12] aims to build models from the data where the selection of labeled data subjects to some bias or preference. The shift between labeled and unlabeled distributions is usually assumed to be of certain types, e.g., label shift, covariate shift, concept shift, etc.

In this paper, we focus on a specific type of problem, where the labeling process is not executed on random instances, but determined by some preceding filtering mechanism. Such a mechanism can be regarded as a deterministic classification model for predicting if an instance is qualified to enter the subsequent process, which includes the observation or production of its ground truth label. If the instance is rejected by the filtering model, the label remains unrevealed and will never be known. The filtering mechanism can be a set of rules, a group of experts, or a machine learning model, depending on the situation. Such situations can occur in various fields including financial, medical, marketing, etc., here we take the lung nodule diagnosis as an example:

Preprint. Under review.

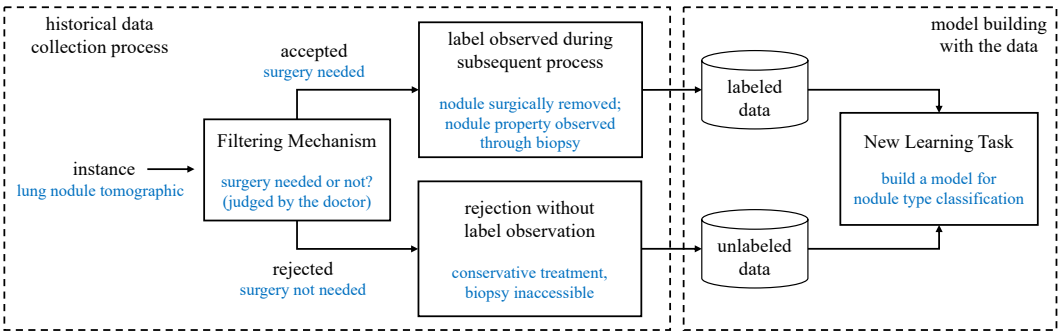
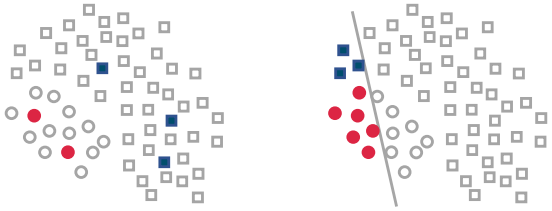


Figure 1: Demonstration of the labeling process governed by a filter.



(a) No support isolation (b) Support isolation

Figure 2: Demonstration of a semi-supervised dataset with support isolation.

When a lung nodule is detected during a CT lung scan, the doctors decide if the nodule has to be surgically removed based on some treatment rules. If the nodule needs surgical removal, the nodule tissue can be collected during the surgery and then analyzed by the pathologists under a microscope. In such a case, the property of the tumor is observed. If the nodule is decided against surgical removal, the patient will have a conservative treatment, and the nodule property remains unrevealed since the nodule tissue is not available for biopsy. The doctors "filter" the data to be "labeled" according to the treatment rules, which makes the data suffer label missing for building machine learning models for other tasks like nodule classification.

Such a filtering mechanism makes the distribution of the labeled data different from the overall data distribution. A hard filtering boundary isolates the labeled distribution and the unlabeled distribution, making it difficult to learn the decision boundary for the target task on the unlabeled side, as shown in Figure 2. Here we remark that the problem we face can be regarded as some sort of sample selection bias, but the mainstream of research in this direction does not interested in such exceptional case that the labeled and unlabeled distribution do not overlap in their support. Instead, the condition that the support of biased labeled data distribution covers the support of the population distribution is generally required, and the distribution density ratio has to be bounded [13, 12, 11, 14]. Such an issue also prevents us from solving it as a domain adaptation problem by regarding the labeled and unlabeled side as two domains, as it has been shown that learning invariant representations can be unhelpful under label shift and shift in the support [15–18].

In this paper, we try to tackle the problem under the framework of semi-supervised learning. We propose Small-Paced Self-Training framework to address the problem. Self-training methods recently show great power on tasks including semi-supervised learning, domain adaptation, and unsupervised learning [19–23]. Some recent theoretical results reveal the insights of self-training algorithms on specific scenarios, including gradual domain adaptation [24], and self-training with consistency regularization [25]. We modify self-training by adding a subset selection mechanism to ensure the pseudo-labeler models make provably low error. To be concrete, Small-Paced Self-Training framework selects a pseudo-labeled subset for training the pseudo-labeler models, and produces pseudo labels only on an unlabeled subset whose Wasserstein distance with the training set is bounded. Intuitively, this strategy learns the concept in a 'small-paced' way to avoid the performance degradation caused by the distribution mismatch. Our theoretical analysis shows that by restricting

the Wasserstein distance of the training distribution and pseudo-labeled distribution, Small-Paced Self-Training produces pseudo-labels with low error provably. Our contributions are two folds:

1. We propose Small-Paced Self-Training framework for semi-supervised learning with support isolation problem, and theoretically show that our Small-Paced Self-Training helps the learning under support isolation.
2. We provide a practical algorithm of the Small-Paced Self-Training framework, and empirically show the effectiveness of our algorithm on benchmark and real-world tasks.

2 Problem Setup

Consider the problem of binary classification, let $P_{X,Y}$ denote the underlying data joint distribution over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is the feature space and $\mathcal{Y} = \{+1, -1\}$. Let P_X be the marginal distribution of X. In ordinary semi-supervised learning, the labeled data and unlabeled data

$$D_L = \{(x_i, y_i)\}_{i=1,\dots,n} \sim P_{X,Y}\,, \text{ and}$$
$$D_U = \{(x_j)\}_{j=n+1,\dots,n+m} \sim P_X\,,$$

are considered sampled i.i.d. from the same distribution P.

The problem we face is that a filtering mechanism governs the labeling process, making the sampling of labeled data and unlabeled data no longer i.i.d., but conditioned on the filter's decision. For an instance $X \sim P_X$, random variable $S = g(X)$ where $g : \mathcal{X} \to \{0, 1\}$ is the deterministic filter function which indicates if we are able to observe the label of X. Such filter function defines a partition $\{\mathcal{X}_L, \mathcal{X}_U\}$ of $\mathcal{X}$. The labels of the instances in subspace $\mathcal{X}_L = \{X \in \mathcal{X} | g(X) = 1\}$ are all observed and the instances in subspace $\mathcal{X}_U = \{X \in \mathcal{X} | g(X) = 0\}$ are all unlabeled. In this situation, the labeled set D_L and unlabeled set D_U are sampled from different conditional distributions:

$$D_L = \{(x_i, y_i)\}_{i=1,\dots,n} \sim P_{X,Y|S=1}\,, \text{ and}$$
$$D_U = \{(x_j)\}_{j=n+1,\dots,n+m} \sim P_{X|S=0}\,,$$

and the support of the two distributions are disjoint, or isolated. Particularly, in this paper, we consider the filtering function $g(\cdot)$ as a deterministic machine learning model, or a rule-based decision process. We do not require the knowledge of $g(\cdot)$ other than the dataset and labels.

In general, since $\mathcal{X}_L$ and $\mathcal{X}_U$ are disjoint, we obtain no knowledge on conditional probability $P_{Y|X}$ on $\mathcal{X}_U$ by learning on D_L, and thus the model cannot generalize to $\mathcal{X}_U$. However, the underlying structure of the data distribution may be captured to enable learning. In this paper, we assume the data distribution satisfies some assumptions on its connectivity and separability. These assumptions are commonly used in recent theory analysis for semi-supervised learning, unsupervised learning, and domain adaptation approaches [25, 26], and have been shown to hold for common data distribution including mixtures of isotropic Gaussians and mixtures of manifolds [25].

Definition 2.1 ((a, c)-expansion)**.** The class-conditional distribution P_i satisfies (a, c)-expansion if for all $V \subseteq \mathcal{X}$ with $P_i(V) \leq a$, the following holds:

$$P_i(\mathcal{N}(V)) \geq \min\{cP_i(V), 1\}\,, \tag{1}$$

where

$$\mathcal{N}(x) = \{x' | d(x, x') \leq r\}\,, \tag{2}$$
$$\mathcal{N}(V) = \cup_{x \in V} \mathcal{N}(x)\,. \tag{3}$$

And if P_i satisfies (a, c)-expansion for both classes, then we say P satisfies (a, c)-expansion.

Definition 2.2 ((r, μ)-separation)**.** For a distribution P, an instance $x \sim P$, with at most probability μ, there exists $x' \sim P$ belongs to the different class of x and $x' \in \mathcal{N}(x)$.

Assumption 2.3 (Expansion)**.** We assume that the population distribution P satisfies $(0.5, c)$-expansion on $\mathcal{X}$ for some $c > 1$.

Assumption 2.4 (Separation)**.** We assume that the population distribution P satisfies (r, μ)-separation for some small $\mu > 0$.

Remark 2.5. The above problem setting belongs to a special case of *sample selection bias*, in which the sampling process of the labeled data subjects to some bias that can be described as a random rejection variable S. Let $S = 1$ represents the label is revealed and $S = 0$ represents the label is unrevealed. Then, the training set is sampled from $P_{X,Y|S=1}$ and the unlabeled set is sampled from P_X. Existing researches on this topic require the selection condition:

$$\sup_{(X,Y)\in\mathcal{X}\times\mathcal{Y}} \frac{P_{X,Y|S=1}(X,Y)}{P_{X,Y}(X,Y)} < \infty\,,$$

or equivalently, for any measurable set A, it holds $P_{X,Y|S=1}(A) = 0 \Rightarrow P_{X,Y}(A) = 0$ [27, 28, 11]. With this condition, the propensity score function $s(X) = P(S = 1|X)$ can be modeled and used to reweight the instances. In our setup, $P(S = 1|X) = g(X)$ is deterministic and ranging discrete in $\{0, 1\}$, and the goal is to build a model of $P(Y|X, S = 0)$ with the help of the unlabeled data. The existing line of research cannot be adapted to solve the problem of this paper.

Remark 2.6. Our problem setup is close to *domain adaptation*, where we have labeled data D_S from the source domain and unlabeled D_T from the target domain. Researches on this topic generally assume the class prior remains consistent in the two domains (or at least not vary significantly), and try to find a mapping function to map the source domain and the target domain into one same feature space. The problem we face is different to domain adaptation in two aspects: 1) in our setup, the class prior on the labeled and unlabeled side may vary arbitrarily, and their support does not overlap; and 2) there is not any conceptual reasonable invariant representation of the labeled and the unlabeled data. These differences make it difficult to adapt theories and practices to our setup [18, 29].

3 Small-Paced Self-Training

Standard self-training algorithms iteratively make predictions on the unlabeled data, and then add the confident predictions into the labeled set to refine the model. It is prone to failure when the labeled and unlabeled data distributions change significantly. We point out that 1) standard self-training typically uses all (pseudo-)labeled data, with or without weighting, to build the model for pseudo-labeling more instances, and 2) all the unlabeled instances are candidates for pseudo-labeling. Our proposed small-paced self-training differs from standard self-training in these two aspects.

3.1 Small-Paced Self-Training Framework

The main challenge of the problem is that the labeled distribution and the unlabeled distribution have disjoint support, hence when producing pseudo labels, the distribution discrepancy of $P_{X,Y|S=0}$ and $P_{X,Y|S=1}$ can be large, and the class prior between the two distributions can change arbitrarily.

To tackle this problem, the main idea of small-paced self-training is to 'break' the unlabeled distribution into small component distributions, so that at each iteration, we can find some labeled instances that the unlabeled ones are 'closed' to, and the model trained on the labeled instances can be provably adapted to the unlabeled ones. We use large margin models as the base model in each iteration. Intuitively, when unlabeled distribution drifts away from the labeled distribution only at a small pace, the unlabeled instances get close to the classification border but are unlikely to get across the border if the classifier has a large margin to the labeled instances.

Formally, at iteration t, we denote the subspace remaining unlabeled as $\mathcal{X}_U^{(t)}$ and the pseudo-labeled subspace as $\mathcal{X}_L^{(t)}$. Instead of training model with instances in $\mathcal{X}_L^{(t)}$ and generating pseudo-labels on instances in $\mathcal{X}_U^{(t)}$, we instead select a subset of $\mathcal{X}_L^{(t)}$ and a subset of $\mathcal{X}_U^{(t)}$, namely $\tilde{\mathcal{X}}_L^{(t)}$ and $\tilde{\mathcal{X}}_U^{(t)}$. We define the component distributions $P^{(t)} = P(x|x \in \tilde{\mathcal{X}}_L^{(t)})$ and $Q^{(t)} = P(x|x \in \tilde{\mathcal{X}}_U^{(t)})$. If all unlabeled data are pseudo-labeled in T iterations, we have:

$$P_{S=0} = P(x|x \in \bigcup_{t=1}^{T} \tilde{\mathcal{X}}_U^{(t)})\,. \tag{4}$$

Distribution distance. To make the pseudo-labels on $Q^{(t)}$ reliable, in iteration t, we want to find some labeled instances from some component distribution $P^{(t)}$, such that the distributional distance between $P^{(t)}$ and $Q^{(t)}$ is small enough, and consequently the model f trained on $P^{(t)}$

can produce predictions on $Q^{(t)}$ with guaranteed performance. However, given the labeled and unlabeled distribution isolated in their supports, distribution distance measures like KL-divergence and JS-divergence cannot be defined. A reasonable choice here to consider is Wasserstein distance. In this paper, we use Wasserstein-infinity distance of the distributions:

$$W_\infty(P, Q) = \inf_T(\sup_x \|T(x) - x\|_2)\,, \tag{5}$$

$$T : \mathbb{R}^d \to \mathbb{R}^d, T_\# P = Q\,, \tag{6}$$

where $T_\# P$ denotes the push-forward measure of P by some measurable mapping T such that $T_\# P(A) = P(T^{-1}(A))$ for every set $A \subseteq \mathbb{R}^d$. Intuitively, W_∞ gives the upper bound of the distance of moving points to match the distribution P and Q. Let $\rho(P, Q)$ be the maximum W_∞ on the class conditional distributions:

$$\rho(P, Q) = \max_{y\in\{+1,-1\}}(W_\infty(P_{X|Y=y}, Q_{X|Y=y}))\,. \tag{7}$$

Such measure bounds the maximum moving distance of conditional mapping from Q to P.

Base models. We consider both linear and deep large margin models for classifying the component distributions. For linear base models $f(x) = w^\top x + b$, we optimize ramp loss with ℓ_2 regularization, which produces large margin classifiers and is shown robust to the outliers. Notice that although the entire dataset is not linearly separable, our algorithm breaks the whole dataset into separable subsets and produces reliable pseudo labels gradually, so that the linear base models do not restrict our algorithm to simple tasks. The ramp loss is formalized as:

$$\ell_r(z) = \min(\max(1 - z, 0), 1)\,. \tag{8}$$

For deep models, we use Large Margin Deep Networks [30] as the base models, which penalize the decision boundary going through the neighborhood of instance within distance δ:

$$\ell_m(x, y) = \max(0, \delta + yd_{f,x})\,, \tag{9}$$

where $d_{f,x}$ is the distance of instance x from the decision boundary:

$$d_{f,x} = \min_\xi \|\xi\|_2 \tag{10}$$

$$\text{s.t. } f(x + \xi) = 0\,. \tag{11}$$

Both linear and deep models we choose here enlarge the classification margin. For the linear model, by regularizing the $\|w\| \le R$ for some $\frac{1}{R} > \delta$, it penalizes the instances within the margin of distance at least δ from the classification border. For the deep model, the large margin loss also pushes the decision boundary away from the instances by a distance of at least δ.

Overall framework. Our framework requires the constructed component distributions at each step to meet the following conditions:

1. class prior consistent: $P_Y^{(t)} = Q_Y^{(t)}$;
2. small shifting: the distributional distance of labeled and unlabeled distributions is bounded, i.e., $\rho(P^{(t)}, Q^{(t)}) \le \delta$;
3. α^*-separation: $P^{(t)}$ and $Q^{(t)}$ satisfy α^*-separation for some small $\alpha^* > 0$, i.e., there exists some classifier f^* which can achieve low ramp loss on $P^{(t)}$ and $Q^{(t)}$.

The following theorem states that as long as the remaining unlabeled distribution contains instances from two classes, the component distributions that meet the above requirements exist.

Theorem 3.1. *Suppose the population distribution satisfies $(0.5, c)$-expansion and (r, μ)-separation. If $0 < P_Y(X|X \in \mathcal{X}_U^{(t)}) < 1$ for $Y \in \{+1, -1\}$, then $P^{(t)}$ and $Q^{(t)}$ that satisfy the class balanced, small shifting, and α^*-separation conditions exist.*

The proof of the Theorem 3.1 is provided in Appendix A. Based on this Theorem we can self-train the model till we cannot find any component distributions, and label the remainder unlabeled instances as some single class.

Practically, instead of explicitly finding the component distributions $P^{(t)}$ and $Q^{(t)}$, we select a pseudo-labeled instance set $\tilde{D}_{PL}^{(t)} \sim P^{(t)}$ and $\tilde{D}_U^{(t)} \sim Q^{(t)}$. Here we give an algorithmic summary of the small-paced self-training framework in Algorithm 1. We will first conduct theoretical analysis in Section 3.2 and then practical implementation of this framework in Section 4.

Algorithm 1 Small-Paced Self-Training Framework

repeat

Choose subset $\tilde{D}_{PL}^{(t)} \sim P^{(t)}$ and $\tilde{D}_{U}^{(t)} \sim Q^{(t)}$.

Train $f^{(t)}$ on $\tilde{D}_{PL}^{(t)}$ with a large margin.

Select $\{x \in \tilde{D}_{U}^{(t)} | f(x) \geq \theta\}$ as the pseudo-labels.

until No unlabeled data remaining.

3.2 Theoretical Analysis

In this section, we conduct theoretical analysis of our small-paced self-training framework.

In step t, we regard the training subset $\tilde{D}_{PL}^{(t)}$ as drawn from $P^{(t)}$, and the generated pseudo-labeled instances $\tilde{D}_{U}^{(t)}$ as drawn from $Q^{(t)}$. Our algorithm makes sure that during the self-training process, $\rho(P^{(t)}, Q^{(t)}) \leq \delta$, and the margin on B' is large, i.e., $f(B') > R$.

We next show that the small-paced self-training framework helps in learning. We first conduct some lemmas, and then give the main Theorem 3.5. The analysis is based on the linear base model case.

Lemma 3.2. *Given n samples D from a joint distribution P over inputs $\mathbb{R}^d$ and labels $\{-1, +1\}$, and suppose $\mathbb{E}_{X\sim P}[\|X\|_2^2] \leq B^2$. Let $\hat{f}$ and f^* be the empirical and population minimizers of the ramp loss respectively:*

$$\hat{f} = \arg\min\, L(f, D)\,, \tag{12}$$

$$f^* = \arg\min\, L(f, P)\,, \tag{13}$$

where

$$L(f, D) = \sum_{x,y\in D} (\ell_r(yf(x)))\,, \tag{14}$$

$$L(f, P) = \mathbb{E}_{X,Y\sim P}[\ell_r(Yf(X))]\,. \tag{15}$$

Then with probability at least $1 - \delta$,

$$L(\hat{f}, P) - L(f^*, P) \leq \frac{4BR + \sqrt{2\log 2/\delta}}{\sqrt{n}}\,. \tag{16}$$

This lemma bounds the generalization error of a regularized linear classifier. The detailed proof is given in Appendix B.1, which follows the general analysis with Rademacher complexity.

Lemma 3.3. *If f is a linear model with $\|w\| < R$, $\rho(P, Q) = \rho < \frac{1}{R}$, and the class priors on P and Q are the same, i.e., $P(Y) = Q(Y)$, then $Err(f, Q) \leq \frac{2}{1-\rho R} L(f, P)$.*

This lemma tells us that if we train a linear classifier f on P, the error rate on Q can be bounded by the ramp loss on P, even if the ramp loss $L(f, Q)$ can be large. Intuitively, since the shift between P and Q is small, and f is a large margin classifier trained on P, the sample from Q may go into the soft margin of f but is not likely to go across the border. The proof is given in Appendix B.2.

Lemma 3.4. *Given random variables X, Y, Y' with joint distribution P, where X denotes the instance, and Y and Y' denote the ground truth labels and the pseudo labels. If the probability of the pseudo label being incorrect $P(Y \neq Y') \leq \epsilon$, then for any $f(x) = w^\top x + b$, we have that $L(f, P_X P_{Y|X}) < L(f, P_X P_{Y'|X}) + \epsilon$.*

This lemma tells that if the pseudo labels have small error w.r.t. the true labels, then we can learn a classifier with low ramp loss by fitting the pseudo labels. The proof is given in Appendix B.3.

Theorem 3.5. *Given two distributions P, Q with $\rho(P, Q) = \rho < \frac{1}{R}$, and the class priors are the same, i.e., $P(Y) = Q(Y)$. Let f be the pseudo-labeler model which is learned on pseudo-labeled distribution $P_X P_{Y'|X}$ with error probability $P(Y \neq Y') \leq \epsilon$. Then about the error of new pseudo labels on Q we have*

$$Err(f, Q) \leq \frac{2}{1 - \rho R}(\alpha^* + \epsilon + \frac{4BR + \sqrt{2\log 2/\delta}}{\sqrt{n}})\,. \tag{17}$$

This theorem can be easily proved by combining the previous lemmas, and the detailed proof is given in Appendix B.4. This theorem tells us that by training a pseudo-labeler with data on P and pseudo-label with small error, the error rate of the pseudo-labels on Q can be bounded. With this theorem, we can bound the error rate of the pseudo labels at any step as follows.

Corollary 3.6. *Suppose $(P^{(t)}, Q^{(t)})$ for $t = [T]$ are selected component distributions that satisfy class balance, small shift, and linear separation. Letting $\gamma = \frac{2}{1-\rho R}$, then at step t, the new pseudo labels' error rate is bounded:*

$$Err(f^{(t)}, Q^{(t)}) \leq \gamma^t \left(\alpha^* + \frac{4BR + \sqrt{2 \log 2/\delta}}{\sqrt{n}} \right) . \tag{18}$$

4 Practical Implementation

In this section, we give a practical implementation of the framework in the agnostic scenario. The detailed algorithm description is shown in Algorithm 2.

Subset selection. We here describe how to find the instance set $\tilde{D}_{PL}^{(t)} \sim P^{(t)}$ for training and $\tilde{D}_{U}^{(t)} \sim Q^{(t)}$ for pseudo-labeling. The problem of finding such subsets with bounded W_∞ distance from two discrete distributions naturally corresponds to the problem of bipartite matching of instance pairs with a maximum distance limitation. We run bipartite graph matching between the pseudo-labeled and unlabeled data with edges between the nodes (x_L, x_U) that $d(x_L, x_U) < \delta$. The matched instances have empirical Wasserstein distance no larger than δ. If the base models are linear classifiers, an extra linear model $f_0(x) = w_0^\top x + b_0$ is trained on the pseudo-labels of the matched instances. By selecting an equal number of positive and negative instances with margin $\hat{y} f_0(x)$ from large to small while keeping $f_0(x_P) > f_0(x_N)$, we obtain a linear separable, class balanced training set $\tilde{D}_{PL}^{(t)}$. If the base models are deep models, we skip this step as deep models have stronger ability of fitting to separate the selected distribution. The pseudo-labeler model $f^{(t)}$ is then trained on $\tilde{D}_{PL}^{(t)}$ to enforce the margin on the training set. If the deep model is used, the model can be pre-trained on all pseudo-labels and then fine-tuned on the selected $\tilde{D}_{PL}^{(t)}$, as the pre-trainings will not decrease but may increase the generalization of the model on $Q^{(t)}$. The unlabeled instances in $\tilde{D}_{U}^{(t)}$ are fed into $f^{(t)}$, those predictions with large margin $|f^{(t)}(x_U)| > \theta$ will be accepted as pseudo-labeled data $\tilde{D}_{U}^{(t)}$.

Dynamic hyper-parameter choosing. The small-paced self-training requires the setting of hyper-parameter δ. Intuitively, δ controls the extent of the shift of the training $P^{(t)}$ and test distribution $Q^{(t)}$, and the smaller the distribution shifts, the better the model generalizes on $Q^{(t)}$. However, a small δ decreases the size of $\tilde{D}_{PL}^{(t)}$, making the model performance unreliable. To find a proper δ, we search from small to large in some interval $[\delta_-, \delta_+]$ containing δ. We start the algorithm from some small δ, and perform self-training only if the size of $\tilde{D}_{PL}^{(t)}$ reaches a lower limit number of instances. If the training data is too few, the δ is increased by a small step, and then the small-paced self-training algorithm continues. Notice that with δ going up, the geometric margin we require the pseudo labels also goes up correspondingly, thus the risk of introducing error is low. Generally, for datasets with normalized features, we search δ in $[0.1, 0.5]$.

5 Experiments

In this section, we verify the proposed small-paced self-training algorithm with linear base models (L) and deep base models (D), against several baselines on semi-supervised learning. The baselines are: **Mean Teacher** [6], a deep semi-supervised learning approach that leverages the consistency of model outputs over different timestamp of the whole training. **MixMatch** [7], a holistic deep semi-supervised learning approach that combines multiple components from different SSL diagrams. **FixMatch** [31], another recent holistic approach that combines consistency regularization and pseudo-labeling, which shown great ability on semi-supervised tasks. **Cycle Self-Training** [26], a self-training variety designed for domain adaptation. Last, we compare standard **Self-Training** [19] to show the small-paced restriction helps the learning process in our setup.

Algorithm 2 Small-Paced Self-Training Algorithm

Let $t = 1$.
repeat
  Select $(\tilde{D}_{PL}^{(t)}, \tilde{D}_{U}^{(t)})$ by bipartite matching of $\big(D_{PL}^{(t)}, D_{U}^{(t)}, \{(x_L, x_U) | \|x_L - x_U\| < \delta\}\big)$.
  if $|\tilde{D}_{PL}^{(t)}| < n$ **then** increase δ, continue.
  (Deep Base Model) Pre-train $f^{(t)}$ with all pseudo-labeled data $D_{PL}^{(t)}$.
  (Deep Base Model) Fine-tune $f^{(t)}$ on $\tilde{D}_{PL}^{(t)}$.
  (Linear Base Model) Train $f^{(t)}$ on $\tilde{D}_{PL}^{(t)}$ with ramp loss and regularization $\|w\| < R$.
  Reject unconfident $\{x_U | f(x_U) < \theta\}$ from $\tilde{D}_{U}^{(t)}$, accept the reminder as the pseudo-labels.
  if $|\tilde{D}_{U}^{(t)}| = 0$ **then** decrease θ.
  Let $t \leftarrow t + 1$.
until No unlabeled data remaining.

We compare the methods on `CIFAR10`, `CIFAR100` [32] dataset and real-world X-ray pneumonia identification task [33], which we refer to as `Pneumonia` hereinafter. On `CIFAR10`, we simulate the situation that we want to build a model to identify vehicles against animals, but the label collecting process is affected by a filter model built on automobile and dog images. Such situations commonly occur when we want to build a model for identification of some interesting object in real-world, but only has limited data to train an imperfect model at the start. For the pneumonia identification task, the X-ray images are collected from healthy children and children with pneumonia [33]. The task we face is to identify the virus pneumonia patients, where the labeled data comes from the patients who are formerly diagnosed with bacterial pneumonia.

Performance under support isolation. The experimental results on `CIFAR10` and `Pneumonia` are shown in Tables 1 and 2. The results on `CIFAR100` are reported in Appendix C due to the page limit. It can be observed that Small-Paced Self-Training outperforms the baseline approaches when support isolation occurs in the datasets. The Small-Paced Self-Training algorithm with deep base models achieves better performance than using the linear models, while both algorithms achieve strong performance.

Table 1: Results of the methods on CIFAR10 dataset. Numbers in brackets are the percentage of the degradation, compared to the no-support-isolation case.

Method	ACC	AUC	F1
MeanTeacher	0.902 (5.5%↓)	0.912 (4.3%↓)	0.887 (5.9%↓)
MixMatch	0.809 (9.4%↓)	0.833 (8.1%↓)	0.799 (9.2%↓)
FixMatch	0.966 (0.4%↓)	**0.967** (**0.5%**↓)	0.958 (**0.5%**↓)
CST	0.953 (0.5%↓)	0.953 (0.6%↓)	0.942 (0.6%↓)
Self-Training	0.921 (5.6%↓)	0.923 (5.1%↓)	0.917 (5.5%↓)
Small-Paced Self-Training (L)	0.919 (3.6%↓)	0.925 (2.8%↓)	0.942 (4.0%↓)
Small-Paced Self-Training (D)	**0.973** (**0.2%**↓)	**0.967** (**0.5%**↓)	**0.961** (0.8%↓)

Table 2: Results of the methods on X-ray image classification task. Numbers in brackets are the percentage of the degradation, compared to the no-support-isolation case.

Method	ACC	AUC	F1
MeanTeacher	0.774 (2.5%↓)	0.737 (3.7%↓)	0.650 (6.3%↓)
MixMatch	0.755 (**0.0%**↓)	0.724 (1.2%↓)	0.639 (2.0%↓)
FixMatch	0.783 (0.6%↓)	0.749 (0.7%↓)	0.672 (0.6%↓)
CST	0.763 (0.7%↓)	0.718 (1.5%↓)	0.626 (2.6%↓)
Self-Training	0.744 (0.6%↓)	0.653 (1.5%↓)	0.485 (3.6%↓)
Small-Paced Self-Training (L)	0.778 (0.8%↓)	0.760 (**0.4%**↓)	0.687 (**0.4%**↓)
Small-Paced Self-Training (D)	**0.796** (1.6%↓)	**0.768** (1.1%↓)	**0.697** (1.6%↓)

Impact of the support isolation. Yet the effect of the filtering mechanism can be regarded as some sort of distribution mismatch of the labeled and unlabeled data, the problem we try to address in this paper is one of the most severe cases. The main obstacle is that the support of the labeled data cannot cover the unlabeled data. To demonstrate the difficulty induced by the support isolation, we alter the `CIFAR10` and `Pneumonia` dataset used in the previous experiment by adding labels of 5% unlabeled data selected at random and removing the labels of the identical amount of labeled data. We denote this altered experiment setup as the *no-support-isolation* case. As the label rate remains unchanged and both labeling setup suffers from huge selection bias, ordinary semi-supervised learning approaches are affected hugely by the change of the distribution support. In Tables 1 and 2, we report the performance degradation when support isolation happens, compared to no support isolation case. The full results of the no-support-isolation case is contained in Appendix C. The Small-Paced Self-Training algorithms are shown to be impacted less from the support isolation problem than the standard self-training. Notice that Small-Paced Self-Training algorithms do not leverage image augmentation and consistency regularization techniques like all of the other baselines do, which are shown to be powerful for image-related tasks. The low performance degradation shows the effectiveness of the small-paced restriction.

6 Related Work

Semi-supervised learning [1, 2] aims to improve learning by utilizing unlabeled data, traditionally can be classified as generative models [34, 4], low density separation based methods [35–37], graph based methods [38, 5], and disagreement based methods [39, 40]. With the rise of deep neural networks in recent years, new approaches are proposed to exploit the power of stronger models for more challenging tasks. Consistency regularization methods are based on the concept that specific types of perturbations applied to an unlabeled instance should not change the model prediction [41, 42]. Entropy minimization methods encourage the models to make confident predictions to avoid the decision boundary going near dense regions [43]. Deep generative models try to recover the data distribution for better feature learning [44, 45]. Graph neural networks exploit the graphical structure of the data with neural networks [46–48]. Holistic models like MixMatch, FixMatch unify multiple strategies and components to achieve strong performance [7, 31].

The problem we address in this paper is also related to, yet different from, domain adaptation [49], sample selection bias [27], and covariate shift [13]. Domain adaptation aims to align source and target domains into one common representation space, so that the labeled source data can be helpful for building a model for the target domain without target label [9, 18, 17]. Literature on sample selection bias and covariate shift problems employ importance reweighting or other techniques to compensate for the distribution density shift [27, 28, 13, 14].

Self-training, also known as pseudo-labeling, is a type of method that trains models according to the previous prediction on the unlabeled data [19, 43]. It is drawing increasing attention, as it shows great effectiveness in semi-supervised learning, domain adaptation, and other related tasks [31, 20, 21]. Though the idea of self-training can date back a very long time, there is little progress in the theoretical understanding of self-training type of algorithms until recent years. Kumar et al. [24] conducted theoretical analysis for self-training of linear models in the scenario of gradual domain adaptation. Wei et al. [25] theoretically analyzed the self-training with input-consistency regularization, provided improved understanding for applying self-training algorithms with deep learning models.

7 Conclusion

In this paper, we address the Semi-Supervised Learning problem where the label missing is caused by a proceeding filtering mechanism. Such filtering mechanism dominated label collecting process leads to the isolation of the support of the labeled and unlabeled data distributions, making the problem more difficult than other scenarios. In this case, the standard self-training approach suffers from overconfidence on instances far away from the current knowledge boundary. We propose Small-Paced Self-Training to tackle this problem, which gradually pushes the knowledge boundary. Theoretical results show that by leveraging such a small-paced restriction, the algorithm can produce reliable pseudo labels on the overall dataset. The algorithm implementation of the framework may not be the only way to enjoy the theoretical guarantee, we hope that there will be more effective algorithms being developed based on the idea of limiting the distributional distance for self-training.

References

[1] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 2006.

[2] Xiaojin Zhu, Andrew B. Goldberg, Ronald Brachman, and Thomas Dietterich. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.

[3] Kristin Bennett and Ayhan Demiriz. Semi-Supervised Support Vector Machines. In *Advances in Neural Information Processing Systems*, volume 11, 1999.

[4] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text Classification from Labeled and Unlabeled Documents Using EM. *Machine Learning*, 39(2):103–134, 2000.

[5] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *Proceedings of the 20th International Conference on Machine Learning*, pages 912–919, 2003.

[6] Antti Tarvainen and Harri Valpola. Mean Teachers are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[7] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[8] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A Theory of Learning from Different Domains. *Machine Learning*, 79(1-2):151–175, 2010.

[9] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1994–2003, 2018.

[10] Xiaoye Qu, Zhikang Zou, Yu Cheng, Yang Yang, and Pan Zhou. Adversarial Category Alignment Network for Cross-domain Sentiment Classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[11] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, editors. *Dataset Shift in Machine Learning*. The MIT Press, 2008.

[12] Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting Sample Selection Bias by Unlabeled Data. In *Advances in Neural Information Processing Systems*, pages 601–608, 2006.

[13] Hidetoshi Shimodaira. Improving Predictive Inference under Covariate Shift by Weighting the Log-Likelihood Function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[14] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate Shift Adaptation by Importance Weighted Cross Validation. *Journal of Machine Learning Research*, 8(35): 985–1005, 2007.

[15] Shai Ben-David and Ruth Urner. On the Hardness of Domain Adaptation and the Utility of Unlabeled Target Samples. In *23rd International Conference on Algorithmic Learning Theory*, volume 7568 of *Lecture Notes in Computer Science*, pages 139–153, 2012.

[16] Fredrik D. Johansson, David A. Sontag, and Rajesh Ranganath. Support and Invertibility in Domain-Invariant Representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, volume 89, pages 527–536, 2019.

[17] Bo Li, Yezhen Wang, Tong Che, Shanghang Zhang, Sicheng Zhao, Pengfei Xu, Wei Zhou, Yoshua Bengio, and Kurt Keutzer. Rethinking Distributional Matching Based Domain Adaptation. 2020.

[18] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J. Gordon. On Learning Invariant Representations for Domain Adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 7523–7532, 2019.

[19] Dong-Hyun Lee. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In *Workshop on challenges in representation learning, ICML*, 2013.

[20] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-Ensembling for Visual Domain Adaptation. In *International Conference on Learning Representations*, 2018.

[21] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning Discrete Representations via Information Maximizing Self-Augmented Training. In *Proceedings of the 34th International Conference on Machine Learning*, page 1558–1567, 2017.

[22] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. SENTRY: Selective Entropy Optimization via Committee Consistency for Unsupervised Domain Adaptation. *CoRR*, abs/2012.11460, 2020.

[23] Sang Michael Xie, Ananya Kumar, Robbie Jones, Fereshte Khani, Tengyu Ma, and Percy Liang. In-N-Out: Pre-Training and Self-Training using Auxiliary Information for Out-of-Distribution Robustness. In *International Conference on Learning Representations*, 2021.

[24] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding Self-Training for Gradual Domain Adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 5468–5479, 2020.

[25] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical Analysis of Self-Training with Deep Networks on Unlabeled Data. In *International Conference on Learning Representations*, 2021.

[26] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle Self-Training for Domain Adaptation. In *Advances in Neural Information Processing Systems*, volume 34, page 14, 2021.

[27] Bianca Zadrozny. Learning and Evaluating Classifiers under Sample Selection Bias. In *Proceedings of the 21st International Conference on Machine Learning*, page 114, 2004.

[28] Anqi Liu and Brian Ziebart. Robust Classification Under Sample Selection Bias. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[29] Rui Shu, Hung H. Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-T Approach to Unsupervised Domain Adaptation. In *International Conference on Learning Representations*, 2018.

[30] Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[31] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, A Colin Raffel, Dogus Ekin Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[32] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

[33] Daniel S. Kermany et al. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5):1122–1131.e9, 2018.

[34] Behzad M. Shahshahani and David A. Landgrebe. The Effect of Unlabeled Samples in Reducing the Small Sample Size Problem and Mitigating the Hughes Phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095, 1994.

[35] Thorsten Joachims. Transductive Inference for Text Classification Using Support Vector Machines. In *Proceedings of the 16th International Conference on Machine Learning*, page 200–209, 1999.

[36] Olivier Chapelle, Mingmin Chi, and Alexander Zien. A Continuation Method for Semi-Supervised SVMs. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 185–192, 2006.

[37] Yu-Feng Li, James T. Kwok, and Zhi-Hua Zhou. Cost-sensitive Semi-Supervised Support Vector Machine. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 500–505, 2010.

[38] Avrim Blum and Shuchi Chawla. Learning from Labeled and Unlabeled Data Using Graph Mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, pages 19–26, 2001.

[39] Avrim Blum and Tom Mitchell. Combining Labeled and Unlabeled Data with Co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[40] Zhi-Hua Zhou and Ming Li. Tri-Training: Exploiting Unlabeled Data Using Three Classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, 2005.

[41] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised Learning with Ladder Networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[42] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*, 2018.

[43] Yves Grandvalet and Yoshua Bengio. Semi-supervised Learning by Entropy Minimization. In *Advances in Neural Information Processing Systems*, volume 17, 2005.

[44] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[45] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised learning with gans: Manifold invariance with improved inference. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[46] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[47] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017.

[48] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph Sequence Neural Networks. In *International Conference on Learning Representations*, 2016.

[49] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[50] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning*. Cambridge University Press, 2009.

A Proof of Theorem 3.1

We first prove the following lemma, and then give the proof of the theorem.

Lemma A.1. *Suppose P_i is some conditional distribution defined on $\mathcal{X} \in \mathbb{R}^d$ satisfies $(0.5, c)$-expansion for some $c > 1$. Suppose A and B are two disjoint subsets of $\mathcal{X}$ such that $\mathcal{X} = A \cup B$. Then there exist $A^* \subseteq A$ and $B^* \subseteq B$ satisfy:*

$$W_\infty(P_i(X|X \in A^*), P_i(X|X \in B^*)) < \delta\,.$$

Proof. Without loss of generality we assume $P_i(A) \geq P_i(B)$. Let $B' = \mathcal{N}(A) \setminus A$, $A' = \mathcal{N}(B) \setminus B$. With $(0.5, c)$-expansion we have A' and B' non-empty. Then for any $x_1 \in A'$, there exists $x_2 \in B'$, $\|x_1 - x_2\| \leq \delta$.

Let C be any hyperball with diameter be δ containing x_1 and x_2. Then $A^* = A \cap C$ and $B^* = B \cap C$ satisfy the condition. □

Proof of Theorem 3.1. By applying Lemma A.1 on P_{+1} and P_{-1} by letting the set A and B be $\mathcal{X}_L^{(t)}$ and $\mathcal{X}_U^{(t)}$, we can easily find for subset $\hat{\mathcal{X}}_{L,+}^{(t)}$, $\hat{\mathcal{X}}_{U,+}^{(t)}$, $\hat{\mathcal{X}}_{L,-}^{(t)}$, $\hat{\mathcal{X}}_{U,-}^{(t)}$, in the feature space such that:

$$W_\infty(P_+(\hat{\mathcal{X}}_{L,+}^{(t)}), P_+(\hat{\mathcal{X}}_{U,+}^{(t)})) < \delta\,,$$
$$W_\infty(P_-(\hat{\mathcal{X}}_{L,-}^{(t)}), P_-(\hat{\mathcal{X}}_{U,-}^{(t)})) < \delta\,.$$

We next prove the separability for linear class. Let C_Y be the hyperball constructed in the proof of Lemma A.1 for $Y = \{+1, -1\}$. Then C_Y covers the support of the selected class conditional distribution: $(\hat{\mathcal{X}}_{L,Y}^{(t)} \cup \hat{\mathcal{X}}_{U,Y}^{(t)}) \subseteq C_Y$. Then due to sap, with probability at least $1 - \mu$, the distance between the centers of two hyperballs is at least δ. Then the two hyperballs are linearly separable.

Finally, as long as the distributions has finite densities, the class prior probabilities can be easily adjusted to be equal by removing some elements from the subsets.

□

B Proof of Theorems in Section 3.2

Here we give the proof omitted in Section 3.2. For some part of the proof we follow the theoretical analysis in [24].

B.1 Proof of Lemma 3.2

We follow the proof in [50]. Let $A = \{l(y, f(x))\}$ where ℓ_r is the ramp loss, f is a linear model with $\|w\| < R$. Since ramp loss ℓ_r is L-Lipschitz function and its Lipschitz constant is 1, we have $\mathcal{R}(A) \leq \mathcal{R}(F)$, where $\mathcal{R}(F)$ is the Rademacher complexity of the linear class. The Rademacher complexity of linear models with $\|w\| < R$ and $\|x\| < B$ is known as:

$$\mathcal{R}(F) < \frac{BR}{\sqrt{n}}\,.$$

By applying the generalization error bound by Rademacher complexity:

$$L(\hat{f}, P) - L(f^*, P) \leq 4\mathcal{R}(A) + \sqrt{\frac{2\log 2/\delta}{n}}\,,$$

we have

$$L(\hat{f}, P) - L(f^*, P) \leq \frac{4BR + \sqrt{2\log 2/\delta}}{\sqrt{n}}\,.$$

B.2 Proof of Lemma 3.3

We first show that if model $f(x) = w^\top x + b$ has low ramp loss $L(f, P)$, then the probability of an instance has a small margin ($yf(x) < \rho R$) is bounded from above.

$$\begin{aligned} L(f,P) &= \mathbb{E}_{X,Y\sim P}[\ell_r(Yf(X))] \\ &\geq \mathbb{E}_{X,Y\sim P}[\ell_r(Yf(X))\mathbb{I}[Yf(X) \leq \rho R]] \\ &\geq \mathbb{E}_{X,Y\sim P}[(1-\rho R)\mathbb{I}[Yf(X) \leq \rho R]] \\ &= (1-\rho R)P[Yf(X) \leq \rho R] \end{aligned}$$

and thus

$$P(Yf(X) \leq \rho R) \leq \frac{1}{1-\rho R} L(f,P)\,. \tag{19}$$

Then we show that as long as the probability of an instance has a small margin is bounded, the error on distribution Q which has small distance with P, is bounded from above too.

With the assumption that $\rho(P,Q) = \rho < \frac{1}{R}$, there exist mapping functions $T_y : \mathbb{R}^d \to \mathbb{R}^d$ for $y \in \{+1, -1\}$ such that for every measurable set $A \subseteq \mathbb{R}^d$, $P_{Y=y}(T^{-1}(A)) = Q_{Y=y}(A)$, and $\sup_x \|T_y(x) - x\|_2 \leq \rho$.[1]

The error of model f on distribution Q is:

$$Err(f,Q) = \sum_y Q(Y=y)Q(Yf(X) \leq 0|Y=y)\,, \tag{20}$$

by applying the inverse of the mapping,

$$Q(Y=y)Q(Yf(X) \leq 0|Y=y) = P(Y=y)P(Yf(T_y^{-1}(X)) \leq 0|Y=y)\,.$$

Given that $\|T_y^{-1}(X) - X\| \leq \rho$ and $\|w\| \leq R$, we have:

$$\|f(T_y^{-1}(X)) - f(X)\| \leq \rho R\,.$$

Thus,

$$P(Y=y)P(Yf(T_y^{-1}(X)) \leq 0|Y=y) \leq P(Y=y)P(Yf(X) \leq \rho R|Y=y)\,.$$

Substitute the above inequality into Equation (20),

$$Err(f,Q) \leq \sum_y P(Y=y)P(Yf(X) \leq \rho R|Y=y) = P(Yf(X) \leq \rho R)\,.$$

Then by combining with Inequality (19), we have:

$$Err(f,Q) \leq \frac{1}{1-\rho R} L(f,P)\,.$$

B.3 Proof of Lemma 3.4

The loss of model f on the true distribution $P_X P_{Y|X}$ can be rewritten as:

$$L(f, P_X P_{Y|X}) = \mathbb{E}[\ell_r(Yf(X))\mathbb{I}[Y = Y']] + \mathbb{E}[\ell_r(Yf(X))\mathbb{I}[Y \neq Y']]\,.$$

In the case of the pseudo label being correct:

$$\mathbb{E}[\ell_r(Yf(X))\mathbb{I}[Y = Y']] \leq \mathbb{E}[\ell_r(Yf(X))] = L(f, P_X P_{Y|X})\,. \tag{21}$$

Otherwise, recall that the ramp loss is bounded in $[0,1]$, and $P(Y \neq Y') \leq \epsilon$, we have:

$$\mathbb{E}[\ell_r(Yf(X))\mathbb{I}[Y \neq Y']] \leq \mathbb{E}[\mathbb{I}[Y \neq Y']] \leq \epsilon\,. \tag{22}$$

By combining Inequalities (21) and (22), we have:

$$L(f, P_X P_{Y|X}) \leq L(f, P_X P_{Y'|X}) + \epsilon\,.$$

[1] The bipartite matching found in Section 4 can be regarded as a mapping on the empirical distribution.

B.4 Proof of Theorem 3.5

Let f^* be the population minimizer on the latent pseudo-labeled distribution $P_X P_{Y'|X}$, and pseudo-labeler f be the empirical minimizer trained with pseudo labels $\tilde{D}_{PL} \sim P_X P_{Y'|X}$:

$$f^* = \arg\min L(f, P_X P_{Y'|X}) ,$$
$$f = \arg\min L(f, \tilde{D}_{PL}) .$$

By Lemma 3.4 we have:

$$L(f, P) \leq L(f, P_X P_{Y'|X}) + \epsilon .$$

Combining with Lemma 3.2, we know that:

$$L(f, P) \leq L(f^*, P_X P_{Y'|X}) + \epsilon + \frac{4BR + \sqrt{2 \log 2/\delta}}{\sqrt{n}} . \tag{23}$$

Since $P_X P_{Y'|X}$ is α^*-separable, which is guaranteed by our subset selection, we have $L(f^*, P) \leq \alpha^*$, and thus

$$L(f, P) \leq \alpha^* + \epsilon + \frac{4BR + \sqrt{2 \log 2/\delta}}{\sqrt{n}} . \tag{24}$$

By Lemma 3.3 we have:

$$Err(f, Q) \leq \frac{2}{1 - \rho R} L(f, P) .$$

Substituting Inequality (24) into the above inequality we have:

$$Err(f, Q) \leq \frac{2}{1 - \rho R} (\alpha^* + \epsilon + \frac{4BR + \sqrt{2 \log 2/\delta}}{\sqrt{n}}) .$$

C Additional Experiment Results

C.1 Experiment Results on CIFAR100

We evaluate our method on `CIFAR100`, the results are shown in Table 3.

Table 3: Results of the methods on CIFAR100 dataset.

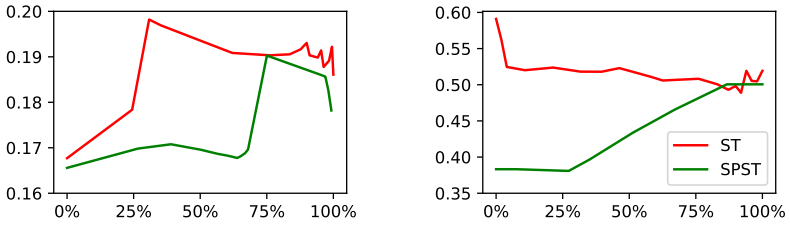
Method	ACC	AUC	F1
MeanTeacher	.649	.708	.643
MixMatch	.722	.752	.682
FixMatch	.831	.839	.782
CST	.798	.749	.670
Self-Training	.682	.752	.684
Small-Paced Self-Training (D)	**.853**	**.858**	**.797**

C.2 Ablation Study

The results of ablation study is shown in Table 4.

Table 4: Ablation study results.

Dataset	Method	ACC	AUC	F1
CIFAR10	ST (non-small-paced)	0.921	0.923	0.917
CIFAR10	SPST (Ours)	0.973	0.967	0.961
Pneumonia	ST (non-small-paced)	0.744	0.653	0.485
Pneumonia	SPST (Ours)	0.796	0.768	0.697



(a) CIFAR10 (b) Pneumonia

Figure 3: Average distance of selected L and U data, w.r.t. the pseudo-labeling rate.

C.3 Subset Selection

We demonstrate the effect of the subset selection mechanism by calculating the average distance of selected labeled and unlabeled instances, on `CIFAR10` dataset and `Pneumonia`. Figure 3 shows the average distance w.r.t. the varying pseudo-labeling rate.

C.4 Complementary Results of the No-support-isolation Case

The experimental results of the no-support-isolation case is listed in Tables 5 and 6. These results are not following the problem setup we study, but used as a reference to calculate the performance degradation in support-isolation case.

Table 5: Results of the methods on CIFAR10 dataset, no support isolation case.

Method	ACC	AUC	F1
MeanTeacher	.954	.953	.943
MixMatch	.893	.906	.880
FixMatch	.970	.972	.963
CST	.958	.959	.948
Self-Training	.976	.973	.970
Small-Paced Self-Training (L)	.953	.952	.942
Small-Paced Self-Training (D)	.975	.972	.969

Table 6: Results of the methods on X-ray image classification task, no support isolation case.

Method	ACC	AUC	F1
MeanTeacher	.794	.766	.694
MixMatch	.755	.733	.652
FixMatch	.788	.754	.676
CST	.768	.729	.643
Self-Training	.749	.663	.690
Small-Paced Self-Training (L)	.784	.763	.690
Small-Paced Self-Training (D)	.809	.776	.708