Towards Safe Weakly Supervised Learning

Yu-Feng Li, Lan-Zhe Guo, and Zhi-Hua Zhou, *Fellow, IEEE*

Abstract—In this paper, we study weakly supervised learning where a large amount of data supervision is not accessible. This includes i) *incomplete* supervision, where only a small subset of labels is given, such as semi-supervised learning and domain adaptation; ii) *inexact* supervision, where only coarse-grained labels are given, such as multi-instance learning and iii) *inaccurate* supervision, where the given labels are not always ground-truth, such as label noise learning. Unlike supervised learning which typically achieves performance improvement with more labeled examples, weakly supervised learning may sometimes even degenerate performance with more weakly supervised data. Such deficiency seriously hinders the deployment of weakly supervised learning to real tasks. It is thus highly desired to study *safe* weakly supervised learning, which never seriously hurts performance. To this end, we present a generic ensemble learning scheme to derive a safe prediction by integrating multiple weakly supervised learners. We optimize the worst-case performance gain and lead to a maximin optimization. This brings multiple advantages to safe weakly supervised learning. First, for many commonly used convex loss functions in classification and regression, it is guaranteed to derive a safe prediction under a mild condition. Second, prior knowledge related to the weight of the base weakly supervised learners can be flexibly embedded. Third, it can be globally and efficiently addressed by simple convex quadratic or linear program. Finally, it is in an intuitive geometric interpretation with the least square loss. Extensive experiments on various weakly supervised learning tasks, including semi-supervised learning, domain adaptation, multi-instance learning and label noise learning demonstrate our effectiveness.

Index Terms—Weakly supervised learning, safe, semi-supervised learning, domain adaptation, multi-instance learning, label noise learning

1 Introduction

Machine learning has achieved great success in numerous tasks, particularly in supervised learning such as classification and regression. But most successful techniques, such as deep learning [1], require ground-truth labels to be given for a big training data set. It is noteworthy that in many tasks, however, it can be difficult to attain strong supervision due to the fact that the hand-labeled data sets are time-consuming and expensive to collect. Thus, it is desirable for machine learning techniques to be able to work well with weakly supervised data [2].

Compared to the data in traditional supervised learning, weakly supervised data does not have a large amount of precise label information. Weakly supervised data is important in machine learning and commonly appear in many real applications. More specifically, three types of weakly supervised data commonly exist [2].

- *Incomplete* supervised data, i.e., only a small subset of training data is given with labels whereas the other data remain unlabeled. For example, in image categorization [3], it is easy to get a huge number of images from the Internet, whereas only a small subset of images can be annotated due to the annotation cost. Representative techniques for this situation are *semi-supervised learning* [4] which aims to learn a prediction model by leveraging a number of unlabeled data and *domain adaptation* [5] which aims to exploit further supervision information from other related domains.
- *Inexact* supervised data, i.e., only coarse-grained labels are given. Reconsider the image categorization task, it is desirable to have every object in the images annotated; however, usually we only have image-level labels rather than object-level labels. One representative technique for this scenario is *multi-instance learning* [6], which aims to improve the performance by considering the coarse-grained label information.
- *Inaccurate* supervised data, i.e., the given labels have not always been ground-truth. Such a situation occurs in various tasks such as image categorization, when the annotator is careless or weary, or the annotator is not an expert. For this type of label information, *label noise learning* techniques are one main paradigm to learn a promising prediction from noisy label [7].

In traditional machine learning, it is often expected that machine learning techniques such as supervised learning with the usage of more data will be able to improve learning performance. Such observation, however, no longer holds for weakly supervised learning. There are many studies [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] reporting that the usage of weakly supervised data may sometimes lead to performance degradation, that is, the learning performance is even worse than that of baseline methods without using weakly supervised data. Fig. 1 illustrates the intuition. More specifically,

- Semi-supervised learning using unlabeled data may be worse than supervised learning with only limited labeled data [4], [8], [9], [10].

- *The authors are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210023, China. E-mail: {liyf, guolz, zhouzh}@lamda.nju.edu.cn.*

Manuscript received 9 Nov. 2018; revised 17 Apr. 2019; accepted 29 May 2019. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Yu-Feng Li.)
Recommended for acceptance by Y. Guo.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2019.2922396

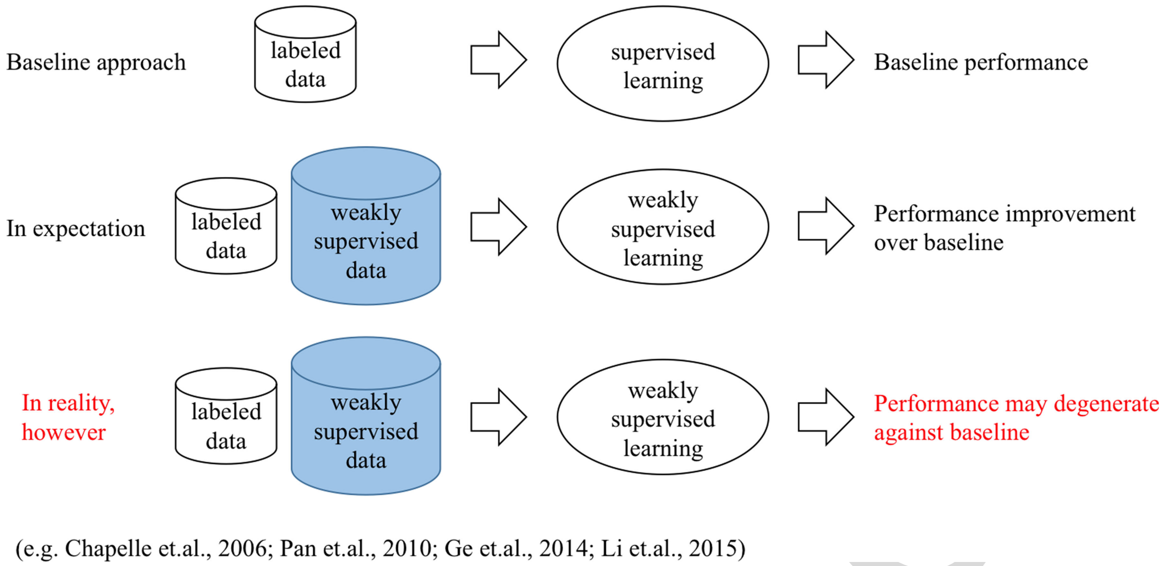

Fig. 1. In practice weakly supervised learning may be not safe, i.e., it may degenerate the performance with the usage of weakly supervised data.

- Domain adaptation has the phenomenon of *negative transfer* [5], [11], [12], [13], [14] that the source domain data contributes to the reduced performance of learning in the target domain.
- Multi-instance learning may be outperformed by the naive learning methods which simply assign the coarse-grained label to a bag of instances [6].
- Label noise learning may be worse than that of learning from only a small amount of high-quality labeled data [7], [15], [16].

Such observations obviously stray from the principle of weakly supervised learning. It is desired to study *safe* weakly supervised learning [17], so that the performance will not be significantly hurt. There is just a little amount of effort on this aspect recently, e.g., [9], [13], [18], whereas they typically work on one concrete scenario. The proposal suitable for various weakly supervised learning scenarios, to our best knowledge, has not been thoroughly studied yet.

1.1 Our Contribution

In this paper, we present a general ensemble learning scheme, SafeW (SAFE Weakly supervised learning), which learns the final prediction by integrating multiple weakly supervised learners. Specifically, we propose a maximin framework, which maximizes the performance gain in the worst case. The framework brings multiple advantages to safe weakly supervised learning. i) It can be shown that the proposal is probably safe for many loss functions (e.g., square loss, hinge loss) in classification and regression, as long as the ground-truth label assignment can be expressed as a convex combination of base learners. ii) Prior knowledge related to the weight of base learners can be easily embedded in our framework. iii) The proposed formulation can be globally and efficiently addressed via a simple convex quadratic program or linear program. iv) It has an intuitive interpretation with the square loss function.

Extensive experimental results on multiple weakly supervised learning scenarios, i.e., semi-supervised learning, domain adaptation, multi-instance learning and label noise learning clearly demonstrate the effectiveness of our proposal.

1.2 Organization

This paper is organized as follows. We first introduce preliminaries in Section 2 and then present our generic framework in Section 3, in which we provide theoretical analysis and study the setup of the weight of base learners. Moreover, we show how to optimize the proposed formulation in Section 4 and relate to some existing work in Section 5. Finally, we report the experimental results in Section 6 and conclude the paper in Section 7.

2 Preliminaries

In weakly supervised learning, due to the lack of sufficient precise label information, ensemble learning that integrates multiple base learners [19] is known as a popular learning technology for weakly supervised data to derive robust performance. Specifically, suppose we have obtained b predictions $\{\mathbf{f}_1, \ldots, \mathbf{f}_b\}$ of unlabeled instances from multiple weakly supervised base learners, where $\mathbf{f}_i \in \mathbb{H}^u$, $i = 1, \ldots, b$ and u is the number of unlabeled instances. Here both classification and regression tasks for weakly supervised data are considered. For classification task $\mathbb{H} = \{+1, -1\}$ and for regression task $\mathbb{H} = \mathbb{R}$. We summarize the main notations appeared in our paper in Table 1.

Many strategies have been employed to generate multiple weakly supervised learners, such as through different learning algorithms, different sampling methods, different model parameters, etc [19]. Previous studies typically focus on deriving good performance from multiple base learners, whereas failing to take the *safeness* of performance into account. In fact, the *good* performance of multiple base learners needs to compare with the baseline approach, and should not suffer from performance degradation.

We let $\mathbf{f}_0 \in \mathbb{H}^u$ denote the prediction of baseline approaches, e.g., directly supervised learning with only limited labeled data. Our ultimate goal is here to derive a safe prediction $\mathbf{f} = g(\{\mathbf{f}_1, \ldots, \mathbf{f}_b\}, \mathbf{f}_0)$, which often outperforms the baseline $\mathbf{f}_0$, meanwhile it would not be worse than $\mathbf{f}_0$. In other words, we would like to maximize the performance gain between our prediction and the baseline prediction.

TABLE 1
Summary of Notations Used in This Paper

Notation	Meaning
u	number of unlabeled instances
b	number of weakly supervised base learners
$\mathbb{H}$	output space, for classification $\mathbb{H} = \{+1, -1\}$; for regression $\mathbb{H} = \mathbb{R}$
$\mathbf{f}_1, \ldots, \mathbf{f}_b \in \mathbb{H}^u$	prediction of weakly supervised learners for unlabeled instances
$\mathbf{f}_0 \in \mathbb{H}^u$	prediction of baseline approach, e.g., supervised learning with labeled data only
$\mathbf{f}^* \in \mathbb{H}^u$	ground-truth prediction for unlabeled instances
$\hat{\mathbf{f}} \in \mathbb{H}^u$	final prediction for unlabeled instances
$\ell(\cdot,\cdot)$	loss function
$\boldsymbol{\alpha}$	weights of weakly supervised base learners
$\mathcal{M}$	a convex set of weights $\boldsymbol{\alpha}$
$\mathbf{C}^{clf}$	covariance matrix of b weakly supervised learners for classification task
$\mathbf{C}^{reg}$	covariance matrix of b weakly supervised learners for regression task

3 The Proposed Framework

We first consider a simpler case that the ground-truth label assignment on unlabeled instances is known. Specifically, let $\mathbf{f}^*$ denote the ground-truth label assignment. Remind that our goal is to find a prediction $\mathbf{f}$ that maximizes the performance gain against the baseline $\mathbf{f}_0$. One can easily have the objective function as

$$\max_{\mathbf{f}\in\mathbb{H}^u} \ \ell(\mathbf{f}_0, \mathbf{f}^*) - \ell(\mathbf{f}, \mathbf{f}^*)$$

Here $\ell(\cdot,\cdot)$ refers to a loss function, e.g., the square loss, the hinge loss, etc. Table 2 summarizes some commonly used loss functions for classification and regression. The smaller the value of the loss function is, the better the performance becomes.

However, obviously $\mathbf{f}^*$ is unknown. To alleviate it, inspired by [20], we assume that $\mathbf{f}^*$ is realized as a convex combination of base learners. Specifically, $\mathbf{f}^* = \sum_{i=1}^b \alpha_i \mathbf{f}_i$ where $\boldsymbol{\alpha} = [\alpha_1; \alpha_2; \ldots; \alpha_b] \geq \mathbf{0}$ be the weight of base learners and $\sum_{i=1}^b \alpha_i = 1$. Then we have the following objective instead by replacing the definition of $\mathbf{f}^*$,

$$\max_{\mathbf{f}\in\mathbb{H}^u} \ \ell\left(\mathbf{f}_0, \sum_{i=1}^b \alpha_i \mathbf{f}_i\right) - \ell\left(\mathbf{f}, \sum_{i=1}^b \alpha_i \mathbf{f}_i\right).$$

In practice, however, one may still be hard to know about the precise weight of base learners. We further assume that $\boldsymbol{\alpha}$ is from a convex set $\mathcal{M}$ to make our proposal more practical, where $\mathcal{M}$ captures the prior knowledge about the importance of base learners and we will discuss the setup of $\mathcal{M}$ in the later section. Without any further information to locate the weight of base learners, to guarantee the safeness, we aim to optimize the worst-case performance gain, since, intuitively, the algorithm would be robust as long as the good performance is guaranteed in the worst case. Then we can obtain a general formulation for weakly supervised data with respect to classification and regression tasks as,

$$\max_{\mathbf{f}\in\mathbb{H}^u} \min_{\boldsymbol{\alpha}\in\mathcal{M}} \ \ell\left(\mathbf{f}_0, \sum_{i=1}^b \alpha_i \mathbf{f}_i\right) - \ell\left(\mathbf{f}, \sum_{i=1}^b \alpha_i \mathbf{f}_i\right). \quad (1)$$

3.1 Analysis

We in this section show that Eq. (1) has safeness guarantees for the commonly used convex loss functions as listed in Table 2 in the classification and regression tasks of weakly supervised learning. To achieve that, we first introduce a result as follows.

Theorem 1. *Suppose the ground-truth $\mathbf{f}^*$ can be constructed by base learners, i.e., $\mathbf{f}^* \in \{\mathbf{f} | \sum_{i=1}^b \alpha_i \mathbf{f}_i, \boldsymbol{\alpha} \in \mathcal{M}\}$. Let $\hat{\mathbf{f}}$ and $\hat{\boldsymbol{\alpha}}$ be the optimal solution to Eq. (1). We have $\ell(\hat{\mathbf{f}}, \mathbf{f}^*) \leq \ell(\mathbf{f}_0, \mathbf{f}^*)$ and $\hat{\mathbf{f}}$ has already achieved the maximal performance gain against $\mathbf{f}_0$.*

Proof. First, we define,

$$L(\mathbf{f}, \boldsymbol{\alpha}) = \ell\left(\mathbf{f}_0, \sum_{i=1}^b \alpha_i \mathbf{f}_i\right) - \ell\left(\mathbf{f}, \sum_{i=1}^b \alpha_i \mathbf{f}_i\right).$$

Since Eq. (1) is a max-min formulation, the following inequality holds for any feasible $\mathbf{f}$ and $\boldsymbol{\alpha}$:

$$L(\mathbf{f}, \hat{\boldsymbol{\alpha}}) \leq L(\hat{\mathbf{f}}, \hat{\boldsymbol{\alpha}}) \leq L(\hat{\mathbf{f}}, \boldsymbol{\alpha}).$$

Let $\boldsymbol{\alpha}^*$ make $\mathbf{f}^* = \sum_{i=1}^b \alpha_i^* \mathbf{f}_i$. By setting $\mathbf{f}$ and $\boldsymbol{\alpha}$ to be $\mathbf{f}_0$ and $\boldsymbol{\alpha}^*$, we have,

$$\ell\left(\mathbf{f}_0, \sum_{i=1}^b \hat{\alpha}_i \mathbf{f}_i\right) - \ell\left(\mathbf{f}_0, \sum_{i=1}^b \hat{\alpha}_i \mathbf{f}_i\right) \leq \ell\left(\mathbf{f}_0, \sum_{i=1}^b \alpha_i^* \mathbf{f}_i\right) - \ell\left(\hat{\mathbf{f}}, \sum_{i=1}^b \alpha_i^* \mathbf{f}_i\right)$$

Thus,

$$\ell(\hat{\mathbf{f}}, \mathbf{f}^*) \leq \ell(\mathbf{f}_0, \mathbf{f}^*).$$

TABLE 2
Commonly Used Loss Functions $\ell(\mathbf{p}, \mathbf{q})$ for Classification and Regression Tasks

Loss function	Definition of $\ell(\mathbf{p}, \mathbf{q})$	Task	η
Hinge loss	$\frac{1}{u}\sum_{i=1}^u \max\{1 - p_i q_i, 0\}$	Classification	1
Cross entropy loss	$\frac{1}{u}\sum_{i=1}^u -p_i \ln(q_i) - (1-p_i)\ln(1-q_i)$	Classification	1
Mean square loss	$\frac{1}{u}\sum_{i=1}^u (p_i - q_i)^2 = \frac{1}{u}(1 - \mathbf{pq})^2$	Classification	4
Mean square loss	$\frac{1}{u}\sum_{i=1}^u (p_i - q_i)^2 = \frac{1}{u}\Vert\mathbf{p} - \mathbf{q}\Vert_2^2$	Regression	2 + M
Mean absolute loss	$\frac{1}{u}\sum_{i=1}^u \vert p_i - q_i\vert = \frac{1}{u}\Vert\mathbf{p} - \mathbf{q}\Vert_1$	Regression	1
Mean ϵ-insensitive loss	$\frac{1}{u}\sum_{i=1}^u \max\{\vert p_i - q_i\vert - \epsilon, 0\}$	Regression	1

The prediction $\mathbf{q} = [q_1; \ldots; q_u] \in \mathbb{R}^u$ and the label $\mathbf{p} = [p_1; \ldots; p_u] \in \mathbb{H}^u$ where $\mathbb{H}^u = \{+1, -1\}^u$ is for classification and $\mathbb{H}^u = \mathbb{R}^u$ is for regression. η is the Lipschitz constant and $M = \max\{|a|, |b|\}$ for regression tasks where the prediction value is in $[a, b]$.

Moreover, since we have already maximized the performance gain in the worst case, $\hat{\mathbf{f}}$ has already achieved the maximal performance gain against $\mathbf{f}_0$. □

According to Theorem 1, we can see that Eq. (1) is a reasonable formulation for our purpose, that is, the derived optimal solution $\hat{\mathbf{f}}$ from Eq. (1) often outperforms $\mathbf{f}_0$ and it would not get any worse than $\mathbf{f}_0$. In comparison to previous studies in [9], [18], [20], the formulation in Eq.(1) brings multiple advantages. In contrast to [9] which requires that the ground-truth is one of the base learners, the condition in Theorem 1 is looser and more practical. In contrast to [18], we explicitly consider to maximize the performance gain over baseline in Eq. (1). In contrast to [20] that focuses on regression, our work is readily applicable for both regression and classification tasks.

Assume that the loss function $\ell(\cdot,\cdot)$ is η-Lipschitz, i.e., $\|\ell(\mathbf{f}_1,\mathbf{f}_2)-\ell(\mathbf{f}_1,\mathbf{f}_3)\| \le \eta\|\mathbf{f}_2-\mathbf{f}_3\|_1$ for any $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3 \in [-1,1]$. Most of commonly used loss functions satisfy this property, and we summarize the η of commonly used loss functions [21] in Table 2. Let $\boldsymbol{\beta}^* = [\beta_1^*,\cdots,\beta_b^*] \in \mathcal{M}$ be the optimal solution to the objective,

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}\in\mathcal{M}}{\arg\min}\ \ell\left(\sum_{i=1}^{b}\beta_i\mathbf{f}_i, \mathbf{f}^*\right),$$

and $\boldsymbol{\epsilon}$ be the residual, i.e., $\boldsymbol{\epsilon} = \mathbf{f}^* - \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i$. We have the following result,

Theorem 2. *The performance gain of* $\hat{\mathbf{f}}$ *against* $\mathbf{f}_0$, *i.e.,* $\ell(\mathbf{f}_0,\mathbf{f}^*)-\ell(\hat{\mathbf{f}},\mathbf{f}^*)$, *has a lower-bound* $-2\eta||\boldsymbol{\epsilon}||_1$.

Proof. Note that $\sum_{i=1}^{b}\beta_i^*\mathbf{f}_i \in \{\mathbf{f}|\sum_{i=1}^{b}\alpha_i\mathbf{f}_i, \boldsymbol{\alpha}\in\mathcal{M}\}$. According to Theorem 1, we have

$$\ell\left(\mathbf{f}_0, \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i\right) - \ell\left(\hat{\mathbf{f}}, \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i\right) \ge 0.$$

Since $\mathbf{f}^* = \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i + \boldsymbol{\epsilon}$,

$$|\ell(\hat{\mathbf{f}},\mathbf{f}^*) - \ell\left(\hat{\mathbf{f}}, \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i\right)| \le \eta||\boldsymbol{\epsilon}||_1.$$

The inequality holds for the reason that the loss function is η-Lipschitz continuous. Similarly, we have, $|\ell(\mathbf{f}_0,\mathbf{f}^*) - \ell(\mathbf{f}_0, \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i)| \le \eta||\boldsymbol{\epsilon}||_1$, which means,

$$-\eta||\boldsymbol{\epsilon}||_1 \le \ell(\hat{\mathbf{f}},\mathbf{f}^*) - \ell\left(\hat{\mathbf{f}}, \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i\right) \le \eta||\boldsymbol{\epsilon}||_1$$
$$-\eta||\boldsymbol{\epsilon}||_1 \le \ell(\mathbf{f}_0,\mathbf{f}^*) - \ell\left(\mathbf{f}_0, \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i\right) \le \eta||\boldsymbol{\epsilon}||_1.$$

Using the above two inequalities,

$$\begin{aligned}&\ell(\mathbf{f}_0,\mathbf{f}^*) - \ell(\hat{\mathbf{f}},\mathbf{f}^*)\\ &\ge \left(\ell\left(\mathbf{f}_0, \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i\right) - \eta||\boldsymbol{\epsilon}||_1\right) - \left(\ell\left(\hat{\mathbf{f}}\sum_{i=1}^{b}\beta_i^*\mathbf{f}_i\right) + \eta||\boldsymbol{\epsilon}||_1\right)\\ &\ge -2\eta||\boldsymbol{\epsilon}||_1.\end{aligned}$$

The second inequality holds due to $\ell(\mathbf{f}_0, \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i) - \ell(\hat{\mathbf{f}}, \sum_{i=1}^{b}\beta_i^*\mathbf{f}_i) \ge 0$. □

Theorem 2 discloses that the worst-case performance is only related to the quality of base learners and has nothing to do with the quantity of base learners.

It is worth mentioning that Theorem 1 only gives a sufficient condition for safeness, rather than necessary conditions. Similarly, Theorem 2 only gives the lower bound of performance, not the exact performance. In other words, even if the condition of Theorem 2 is not valid, our method can still achieve robust performance. Our experimental results clearly confirm this observation.

3.2 Weight the Base Learners

The question remained is that how to set up $\mathcal{M}$ which is assumed as a convex set in previous sections. We can simply set $\mathcal{M}$ as a simplex, i.e., $\mathcal{M} = \{\boldsymbol{\alpha}|\sum_{i=1}^{b}\alpha_i = 1, \boldsymbol{\alpha}\ge 0\}$ as [9], [10], [20], but this strategy is too conservative. Obviously, the setup of $\mathcal{M}$ can be easily embedded with a variety of prior knowledge. For example, suppose that base learner $\mathbf{f}_i$ is more reliable than $\mathbf{f}_j$ and the set of all such indexes (i,j) is denoted as $\mathcal{S}$, $\mathcal{M}$ could be set to $\{\boldsymbol{\alpha}|\alpha_i - \alpha_j \ge 0, (i,j)\in\mathcal{S}; \boldsymbol{\alpha}^\top\mathbf{1} = 1; \boldsymbol{\alpha}\ge\mathbf{0}\}$ where $\mathbf{1}$ ($\mathbf{0}$) refers to the all-one (all-zero) vector, respectively; suppose that the importance values of base learners are known, denoted by $\{r_1,\ldots,r_b\}$, one could set up $\mathcal{M}$ as $\{\boldsymbol{\alpha}| -\gamma \le \alpha_i - r_i \le \gamma, \forall i = 1,\ldots,b; \boldsymbol{\alpha}^\top\mathbf{1} = 1; \boldsymbol{\alpha}\ge\mathbf{0}\}$ where γ is a small constant. All of these require precise prior knowledge. One could also set $\mathcal{M}$ via cross validation. However, that is time consuming and in weakly supervised learning, labeled data is too few to afford a reliable cross validation. For this reason, we present a method that learns the weights of base learners from data.

3.3 Regression

Let $\mathbf{C}^{reg}$ be the $b\times b$ covariance matrix of the b base learners $\{f_1,\ldots,f_b\}$ with elements

$$C_{ij}^{reg} = \mathbb{E}[(f_i(X)-\mu_i)^\top(f_j(X)-\mu_j)],$$

where X refers to the set of unlabeled instances and $\mu_i = \mathbb{E}[f_i(X)]$. Let $\boldsymbol{\rho}^{reg} = [\rho_1^{reg};\ldots;\rho_b^{reg}]$ be the vector of covariances between the base learners and the ground-truth label assignment $f^*(X)$, i.e.,

$$\rho_i^{reg} = \mathbb{E}[(f^*(X)-\theta)^\top(f_i(X)-\mu_i)],$$

where $\theta = \mathbb{E}[f^*(X)]$. We minimize the residual w.r.t the ground-truth for $\boldsymbol{\alpha}$ as,

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\arg\min}\ \mathbb{E}[\mathrm{MSE}\left(\sum_{i=1}^{b}\alpha_i f_i(X), f^*(X)\right)], \tag{2}$$

where MSE refers to the Mean Squared Error. Eq. (2) has a closed-form solution [22].

Theorem 3. *(Bates and Granger, 1969) The optimal weight* $\boldsymbol{\alpha}^*$ *satisfies that*

$$\boldsymbol{\rho}^{reg} = \mathbf{C}^{reg}\boldsymbol{\alpha}^*.$$

We need to estimate $\mathbf{C}^{reg}$ and $\boldsymbol{\rho}$. For $\mathbf{C}^{reg}$, it is evident that $(\mathbf{f}_i-\mu_i)^\top(\mathbf{f}_j-\mu_j)$ is an unbiased estimation of C_{ij}^{reg}. Therefore, one could easily have $\hat{\mathbf{C}}^{reg}$ with elements

$$\hat{C}_{ij}^{reg}=(\mathbf{f}_i-\mu_i)^\top(\mathbf{f}_j-\mu_j),$$

be the unbiased estimation of $\mathbf{C}^{reg}$. For $\boldsymbol{\rho}$, the following proposition shows that it is closely related to the performance of base learners.

Proposition 1. *Assume that $\{f_i(X)\}_{i=1}^{i=b}$ is normalized to the mean $\mu_i=0,\forall i=1,\ldots n$ and the standard deviation that equals to 1. Consider mean squared error as the measurement, we have, the bigger the value ρ_i^{reg}, the smaller the loss of f_i.*

Proof. For $\boldsymbol{\rho}^{reg}$, we have,

$$\rho_i^{reg}=\mathbb{E}_{(\mathbb{X},\mathbb{Y})}[(\mathbf{f}^*-\theta)^\top(\mathbf{f}_i-\mu_i)]=\mathbb{E}[(\mathbf{f}^*)^\top\mathbf{f}_i].$$

For MSE, we have,

$$\begin{aligned}\mathrm{MSE}(\mathbf{f}_i,\mathbf{f}^*)&=\mathbb{E}[(\mathbf{f}^*-\mathbf{f}_i)^2]\\&=\mathbb{E}[||\mathbf{f}^*||^2+||\mathbf{f}_i||^2-2(\mathbf{f}^*)^\top\mathbf{f}_i]\\&=2-2\mathbb{E}[(\mathbf{f}^*)^\top\mathbf{f}_i]\\&=2-2\rho_i^{reg}.\end{aligned}$$

Hence, the bigger the value ρ_i^{reg}, the smaller the mean square loss of $\mathbf{f}_i$. □

Therefore, we set $\mathcal{M}$ as $\{\boldsymbol{\alpha}|\hat{\mathbf{C}}^{reg}\boldsymbol{\alpha}\geq\mathbf{1}\delta,\boldsymbol{\alpha}^\top\mathbf{1}=1,\boldsymbol{\alpha}\geq\mathbf{0}\}$, where δ is a constant, indicating that the base learners have a low-bound performance (e.g., better than random-guess) [18]. It is easy to verify that $\mathcal{M}$ is a convex set.

3.4 Classification

Similar to regression tasks, let $\mathbf{C}^{clf}$ be the $b\times b$ matrix representing the agreement between base learners with elements $C_{ij}^{clf}=\mathbb{E}[f_i(X)^\top f_j(X)]$. Let $\boldsymbol{\rho}^{clf}=[\rho_1^{clf};\rho_2^{clf};\ldots;\rho_b^{clf}]$ be the vector that represents the agreement between the base learner and the ground-truth,

$$\rho_i^{clf}=\mathbb{E}[f^*(X)^\top f_i(X)].$$

Taking classification accuracy as the performance measure, it can be shown that,

Theorem 4. *The optimal weight $\boldsymbol{\alpha}^*$ in classification satisfies that $\boldsymbol{\rho}^{clf}=\mathbf{C}^{clf}\boldsymbol{\alpha}^*$.*

Similarly, we set $\mathcal{M}$ as $\{\boldsymbol{\alpha}|\hat{\mathbf{C}}^{clf}\boldsymbol{\alpha}\geq\mathbf{1}\delta,\boldsymbol{\alpha}^\top\mathbf{1}=1,\boldsymbol{\alpha}\geq\mathbf{0}\}$ where $\hat{\mathbf{C}}^{clf}$ is the unbiased estimation of $\mathbf{C}^{clf}$, with elements $\hat{C}_{ij}^{clf}=\mathbf{f}_i^\top\mathbf{f}_j$. $\mathcal{M}$ is also a convex set.

In summary, on one hand, our formulation is able to directly absorb the precise prior knowledge about the importance of learners if available. On the other hand, it is also capable of incorporating with the estimation obtained by covariance matrix analysis on regression and classification tasks when the precise prior knowledge is unavailable.

4 Optimization

Another question unclear in our formulation is that, how can we derive the optimal solution of Eq.(1). Eq. (1) is the subtraction of two loss functions, which is often non-convex and not trivial to derive the global optima [23]. Fortunately, we find that for a class of commonly used convex loss function, Eq. (1) could be equivalently rewritten as a convex optimization problem and thus the global optimal solution is achieved. We describe the optimization procedure for regression and classification respectively in this section.

4.1 Regression

For regression, we have the following theorem,

Theorem 5. *For regression, suppose $\ell(\cdot,\sum_{i=1}^b\alpha_i\mathbf{f}_i)$ is convex to $\boldsymbol{\alpha}$ and $\forall\boldsymbol{\alpha}$, and there exists $\mathbf{f}\in\mathbb{R}^u$ such that $\ell(\mathbf{f},\sum_{i=1}^b\alpha_i\mathbf{f}_i)=0$, then Eq.(1) is a convex optimization.*

We first give a lemma before proving Theorem 5.

Lemma 1. *Under the condition in Theorem 5, in optimality, the optimal solution $\hat{\mathbf{f}}$ and $\hat{\boldsymbol{\alpha}}$ have the following relation, i.e., $\ell(\hat{\mathbf{f}},\sum_{i=1}^b\hat{\alpha}_i\mathbf{f}_i)=0$.*

Proof. Assume, to the contrary, $\ell(\hat{\mathbf{f}},\sum_{i=1}^b\hat{\alpha}_i\mathbf{f}_i)\neq0$. According to the condition, there exists $\tilde{\mathbf{f}}$ such that $\ell(\tilde{\mathbf{f}},\sum_{i=1}^b\hat{\alpha}_i\mathbf{f}_i)=0$. Obviously, $0=\ell(\tilde{\mathbf{f}},\sum_{i=1}^b\hat{\alpha}_i\mathbf{f}_i)<\ell(\hat{\mathbf{f}},\sum_{i=1}^b\hat{\alpha}_i\mathbf{f}_i)$. Hence, $\hat{\mathbf{f}}$ is not optimal, a contradiction. □

We then prove Theorem 5.

Proof. Because of Lemma 1, the form of Eq. (1) for regression task is thus rewritten as,

$$\min_{\boldsymbol{\alpha}\in\mathcal{M}}\ \ell\left(\mathbf{f}_0,\sum_{i=1}^b\alpha_i\mathbf{f}_i\right).$$

Remind that $\ell(\cdot,\sum_{i=1}^b\alpha_i\mathbf{f}_i)$ is convex to $\boldsymbol{\alpha}$, therefore, Eq. (1) is a convex optimization. □

It is worth noting that the condition in Theorem 5 is rather mild. Many regression loss functions, for example, mean square loss, mean absolute loss [24] and mean ϵ-insensitive loss [25], all satisfy such a mild condition in Theorem 5.

Depending on Lemma 1 and Theorem 5, the formulation in Eq. (3) can be globally and efficiently addressed for regression. We adopt mean square loss (MSE) as an example to show the optimization procedure since MSE is one of the most popular loss functions for regression. With MSE, Eq. (1) can be written as the following equivalent form which only relates to $\boldsymbol{\alpha}$.

$$\min_{\boldsymbol{\alpha}\in\mathcal{M}}\left\|\sum_{i=1}^b\alpha_i\mathbf{f}_i-\mathbf{f}_0\right\|^2.\qquad(3)$$

It is evident that Eq. (3) turns out to be a simple convex quadratic program. Moreover, specifically, by expanding the quadratic form in Eq. (3), it can be rewritten as,

$$\min_{\boldsymbol{\alpha}\in\mathcal{M}}\boldsymbol{\alpha}^\top\mathbf{F}\boldsymbol{\alpha}-\mathbf{v}^\top\boldsymbol{\alpha},\qquad(4)$$

where $\mathbf{F}\in\mathbb{R}^{b\times b}$ is a linear kernel matrix of $\mathbf{f}_i$'s, i.e, $F_{ij}=\mathbf{f}_i^\top\mathbf{f}_j$ and $\mathbf{v}=[2\mathbf{f}_1^\top\mathbf{f}_0;\cdots;2\mathbf{f}_b^\top\mathbf{f}_0]$. Since $\mathbf{F}$ is positive semi-definite, Eq. (4) is a convex quadratic program [26] and can be efficiently addressed by off-the shelf optimization packages, such as the MOSEK package.[1]

1. https://www.mosek.com/resources/downloads

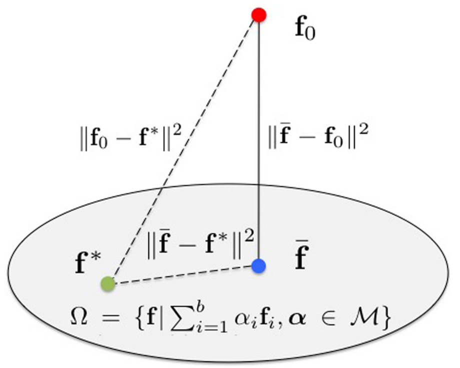

Fig. 2. Intuition of our proposal via the projection viewpoint. Intuitively, the proposal learns a projection of $\mathbf{f}_0$ onto a convex feasible set Ω.

After solving the optimal solution $\boldsymbol{\alpha}^*$, the optimal $\bar{\mathbf{f}} = \sum_{i=1}^b \alpha_i^* \mathbf{f}_i$ is obtained. Algorithm 1 summarizes the pseudo code of the proposed method for regression task.

Algorithm 1. Optimization Procedure for Regression

Input: multiple base learner predictions $\{\mathbf{f}_i\}_{i=1}^b$ and certain direct supervised regression prediction $\mathbf{f}_0$

Output: the learned prediction $\bar{\mathbf{f}}$

1: Construct a linear kernel matrix $\mathbf{F}$ where $F_{ij} = \mathbf{f}_i^\top \mathbf{f}_j$, $\forall 1 \leq i, j \leq b$

2: Derive a vector $\mathbf{v} = [2\mathbf{f}_1^\top \mathbf{f}_0; \ldots; 2\mathbf{f}_b^\top \mathbf{f}_0]$

3: Solve the convex quadratic optimization Eq.(4) and obtain the optimal solution $\boldsymbol{\alpha}^* = [\alpha_1^*, \ldots, \alpha_b^*]$

4: Return $\bar{\mathbf{f}} = \sum_{i=1}^b \alpha_i^* \mathbf{f}_i$

It is not hard to realize that Eq. (3) meets a geometric projection problem. Specifically, let $\Omega = \{\mathbf{f} | \sum_{i=1}^b \alpha_i \mathbf{f}_i, \boldsymbol{\alpha} \in \mathcal{M}\}$, Eq. (3) can be rewritten as,

$$\bar{\mathbf{f}} = \arg\min_{\mathbf{f} \in \Omega} \|\mathbf{f} - \mathbf{f}_0\|^2, \tag{5}$$

which learns a projection of $\mathbf{f}_0$ onto the convex set Ω.

Fig. 2 illustrates the intuition of our proposed method via the viewpoint of geometric projection.

According to Pythagorean Theorem (theorem 2.4.1 in [27]), the distance between $\|\bar{\mathbf{f}} - \mathbf{f}^*\|$ should be smaller than $\|\mathbf{f}_0 - \mathbf{f}^*\|$ if $\mathbf{f}^* \in \Omega$. Such an observation is consistent with Theorem 1. The viewpoint of geometric projection provide an intuitive insight to help understand safe weakly supervised learning.

4.2 Classification

Due to the noncontinuous feasible field of $\mathbf{f}$, it could not simply apply the lemma 1 in regression task to classification. We now show that for the hinge loss, the optimal solution of Eq. (1) can be achieved. For the cross entropy loss, a popular loss function, it can be solved by convex optimization, which only needs a simple convex relaxation technique. Similar tricks could be possibly applicable for additional convex classification losses.

We first have the following lemma,

Lemma 2. *For classification task, the optimal* $\hat{\mathbf{f}}$ *and* $\hat{\boldsymbol{\alpha}}$ *meet the relation that* $\hat{\mathbf{f}} = sign(\sum_{i=1}^b \hat{\alpha}_i \mathbf{f}_i))$ *where* $sign(s)$ *is the sign of value* s.

Proof. Assume, to the contrary, $\hat{\mathbf{f}} \neq sign(\sum_{i=1}^b \hat{\alpha}_i \mathbf{f}_i)$. According to the condition, there exist $\tilde{\mathbf{f}}$ such that $\tilde{\mathbf{f}} = sign(\sum_{i=1}^b \hat{\alpha}_i \mathbf{f}_i)$. Obviously, $\ell(\tilde{\mathbf{f}}, \sum_{i=1}^b \hat{\alpha}_i \mathbf{f}_i) < \ell(\hat{\mathbf{f}}, \sum_{i=1}^b \hat{\alpha}_i \mathbf{f}_i)$. Hence, $\hat{\mathbf{f}}$ is not optimal, a contradiction. □

We then have the following theorem,

Theorem 6. *Suppose that* $\mathbf{f}_i \in \{+1, -1\}^u, \forall i = 1, \ldots, b$. *Eq. (1) is a convex optimization when* $\ell(\cdot, \cdot)$ *is the hinge loss.*

Proof. With Lemma 2, Eq. (1) is thus rewritten as,

$$\min_{\boldsymbol{\alpha} \in \mathcal{M}} \ell\left(\mathbf{f}_0, \sum_{i=1}^b \alpha_i \mathbf{f}_i\right) - \ell\left(sign\left(\sum_{i=1}^b \alpha_i \mathbf{f}_i\right), \sum_{i=1}^b \alpha_i \mathbf{f}_i\right). \tag{6}$$

Since $\mathbf{f}_i \in \{+1, -1\}^u, \forall i = 1, \ldots, b$ and $\ell(\cdot, \sum_{i=1}^b \alpha_i \mathbf{f}_i)$ satisfies the linearity to predictive results, the form $\ell(sign(\sum_{i=1}^b \alpha_i \mathbf{f}_i), \sum_{i=1}^b \alpha_i \mathbf{f}_i)$ can be equivalently rewritten as $\ell(\| \sum_{i=1}^b \alpha_i \mathbf{f}_i \|_1)$. Therefore, Eq.(6) is equal to,

$$\min_{\boldsymbol{\alpha} \in \mathcal{M}} \ell\left(\mathbf{f}_0, \sum_{i=1}^b \alpha_i \mathbf{f}_i\right) + \ell\left(\left\| \sum_{i=1}^b \alpha_i \mathbf{f}_i \right\|_1\right). \tag{7}$$

Eq.(7) is convex and a linear program. Let $\tilde{\mathbf{f}}$ be $\sum_{i=1}^b \alpha_i \mathbf{f}_i$, then, Eq.(7) can be written as,

$$\min_{\boldsymbol{\alpha} \in \mathcal{M}} \ell(\mathbf{f}_0, \tilde{\mathbf{f}}) + \ell(\|\tilde{\mathbf{f}}\|_1) \;\; \text{s.t.} \;\; \tilde{\mathbf{f}} = \sum_{i=1}^b \alpha_i \mathbf{f}_i. \tag{8}$$

By introducing two auxiliary variables $\mathbf{z} = \frac{|\tilde{\mathbf{f}}| + \tilde{\mathbf{f}}}{2}$, $\mathbf{w} = \frac{|\tilde{\mathbf{f}}| - \tilde{\mathbf{f}}}{2}$, then, Eq. (8) can be transformed into,

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathcal{M}, \mathbf{z}, \mathbf{w}} \quad & \ell(\mathbf{f}_0, \tilde{\mathbf{f}}) + \ell(\mathbf{1}^\top(\mathbf{z} + \mathbf{w})) \\ \text{s.t.} \quad & \tilde{\mathbf{f}} = \sum_{i=1}^b \alpha_i \mathbf{f}_i \\ & \tilde{\mathbf{f}} + \mathbf{z} - \mathbf{w} = \mathbf{0}; \mathbf{z} \geq \mathbf{0}, \mathbf{w} \geq \mathbf{0}, \end{aligned} \tag{9}$$

Furthermore, the loss function $\ell(\cdot, \tilde{\mathbf{f}})$ is linear function to $\tilde{\mathbf{f}}$. Therefore, the objective and constraint are linear to $\boldsymbol{\alpha}, \mathbf{z}, \mathbf{w}$, thus, Eq. (9) is a linear program. □

Eq. (9) can be globally addressed in an efficient manner via the MOSEK package as well. After solving the optimal solution $\boldsymbol{\alpha}^*$, the optimal $\bar{\mathbf{f}} = \sum_{i=1}^b \alpha_i^* \mathbf{f}_i$ is obtained. Algorithm 2 summarizes the pseudo code of the proposed method for classification task.

Algorithm 2. Optimization Procedure for Classification

Input: multiple base learner predictions $\{\mathbf{f}_i\}_{i=1}^b$ and certain direct supervised regression prediction $\mathbf{f}_0$

Output: the learned prediction $\bar{\mathbf{f}}$

1: Let u equals to the length of $\mathbf{f}_0$

2: Solve the linear optimization Eq.(9) and obtain the optimal solution $\boldsymbol{\alpha}^* = [\alpha_1^*, \ldots, \alpha_b^*]$

3: Return $\bar{\mathbf{f}} = \sum_{i=1}^b \alpha_i^* \mathbf{f}_i$

We further show that convexity is also feasible for the cross entropy loss, a popular loss in deep neural network [28], via a slight convex relaxation. Let

$$\hat{\ell}(p) = \begin{cases} \ln(p) & 0.5 \leqslant p \leqslant 1 \\ \ln(1-p) & 0 \leqslant p < 0.5 \end{cases}. \quad (10)$$

It is easy to show that when $\ell(\cdot,\cdot)$ realizes the cross entropy loss,

$$-\ell\left(sign\left(\sum_{i=1}^{b} \alpha_i \mathbf{f}_i\right), \sum_{i=1}^{b} \alpha_i \mathbf{f}_i \right) = \sum_{j=1}^{u} \hat{\ell}\left(\left(\sum_{i=1}^{b} \alpha_i \mathbf{f}_i \right)_j \right),$$

where $((\sum_{i=1}^{b} \alpha_i \mathbf{f}_i)_j)$ refers to the jth element of $(\sum_{i=1}^{b} \alpha_i \mathbf{f}_i)$. Let

$$g(p) = \begin{cases} (2\ln 2)p - 2\ln 2 & 0.5 \leqslant p \leqslant 1 \\ -(2\ln 2)p & 0 \leqslant p < 0.5. \end{cases} \quad (11)$$

It is not hard to verify that $g(p)$ realizes the convex hull, the tightest convex relaxation of $\hat{\ell}(p)$.

Theorem 7. *Let $\tilde{\mathbf{f}} = \sum_{i=1}^{b} \alpha_i \mathbf{f}_i$. Consider the optimization problem,*

$$\min_{\boldsymbol{\alpha}} \ell(\mathbf{f}_0, \tilde{\mathbf{f}}) + \sum_{j=1}^{u} g(\tilde{f}_j). \quad (12)$$

It can be shown that Eq. (12) is convex and the convex relaxation of Eq. (1) with the cross entropy loss.

Proof. According to Lemma 2, the optimal $\mathbf{f}$ leads to $sign(\sum_{i=1}^{b} \alpha_i \mathbf{f}_i)$, which consequently makes Eq. (1) to equivalently write as

$$\min_{\boldsymbol{\alpha}} \ell(\mathbf{f}_0, \tilde{\mathbf{f}}) + \sum_{j=1}^{u} \hat{\ell}(\tilde{f}_j). \quad (13)$$

Remind that $\ell(\mathbf{f}_0, \tilde{\mathbf{f}})$ is the convex loss and $g(p)$ is the convex hull of $\hat{\ell}(p)$. We conclude that Eq. (12) is convex and the convex relaxation of Eq. (1) with the cross entropy loss. □

Similarly, the optimal $\bar{\mathbf{f}} = \sum_{i=1}^{b} \alpha_i^* \mathbf{f}_i$ is obtained with the optimal solution $\boldsymbol{\alpha}^*$ of Eq. (12). Similar tricks could be applied to cope with other convex classification losses.

5 Related Work

Effectively exploiting weakly supervised data has been attracted much attention from the past decade [2], [6], [7]. Many methods have been developed and there are some discussions on the usefulness of weakly supervised data.

In semi-supervised learning, many methods have developed such as, generative model based approaches [29], graph-based approaches [30], disagreement-based approaches [31] and semi-supervised SVMs [32]. In very recent, efforts on safely using unlabeled data attract increasing attention. Li and Zhou [9] aimed to build safe semi-supervised SVMs by optimizing the worst-case performance gain given a set of candidate low-density separators, showing that the proposal is probably safe given that low-density assumption holds [4]. Balsubramani and Freund [18] learned a robust prediction with the highest accuracy given that the ground-truth label assignment is restricted to a specific candidate set. Li, Kwok and Zhou [10] concerned to build a generic safe semi-supervised classification framework for variants of performance measures, e.g., AUC, F_1 score, Top$_k$ precision. However, these studies are restricted on semi-supervised classification, and the effort on semi-supervised regression has not been thoroughly studied.

In domain adaptation, a number of methods have been developed, e.g., instances transfer based approaches [33], feature representation transfer based approaches [34], parameter transfer based approaches [35], relational knowledge transfer based approaches [36]. However, there are few discussions on how to avoid negative transfer though it is regarded as an important issue in domain adaptation [5]. Rosenstein et al. [11] empirically showed that if two tasks are dissimilar, then brute-force transfer may hurt the performance of the target task. Bakker and Heskes [14] presented a Bayesian method for joint prior distribution of multiple tasks and considered that some of the model parameters should be loosely connected among tasks. Argyriou et al. [12] considered situations that the representations should be different among different groups of tasks and tasks within a group are easier to perform domain adaptation. Ge et al. [13] assigned weight to source domains corresponding to the relatedness to the target domain and constructed the final target learner uses the weight to attenuate the effects of negative transfer.

In multi-instance learning, many effective algorithms have been developed, e.g., density-based approaches [37], k-nearest neighbor based approaches [38], support vector machine based approaches [39], ensemble based approaches [40], kernel based approaches [41] and so on [6]. However, multi-instance learning methods have uncertainty and sometimes even worse than the simple supervised learning methods. Ray and Craven [42] compared the performance of MIL methods against supervised methods on MIL. They found that in many cases, supervised yield the most competitive results and they also noted that, while some methods systematically dominate others, the performance of algorithms was application-dependent. Carbonneau et al. [43] studied the ability to identify witnesses (positive instances) of several MIL methods. They found that being dependent on the nature of the data, some algorithm performs well while others would have difficulty. In this paper, we use the worst-case analysis to overcome the model uncertainty and learn a safe prediction.

In label noise learning, many studies have been proposed, such as data cleaning approaches, probabilistic label noise tolerant approaches, ensemble based approaches. There are also a number of studies indicating that label noise will seriously affect the learning performance [7], [15], [16], [44]. Considerable efforts have been made to enable models to be robust to the presence of label noise. For example, in the aspect of theoretical consideration, Manwani and Satry [45] studied the robustness of loss functions in the empirical risk minimization framework and disclosed that 0-1 loss function is noise tolerant while the other loss functions are not naturally noisy tolerant. In the aspect of practical consideration, ensemble methods, e.g., bagging and boosting are regarded to be robust to label noise [7] and bagging often achieves a better result than boosting in the presence of label noise [46].

6 Experiments

In this section, comprehensive evaluations are performed to verify the effectiveness of the proposed.[2] Experiments are

2. http://lamda.nju.edu.cn/code_SAFEW.ashx

conducted on all the four aforementioned weakly supervised learning tasks: semi-supervised learning (Section 6.1), domain adaptation (Section 6.2), multi-instance learning (Section 6.3) and label noise learning (Section 6.4).

6.1 Semi-Supervised Learning

For semi-supervised learning, we do experiments on regression tasks with a broad range of datasets[3] that cover diverse domains including physical measurements (*abalone*), health (*bodyfat*), economics (*cadata*), activity recognition (*mpg*), etc. The sample size ranges from around 100 (*pyrim*) to more than 20,000 (*cadata*).

We compare the performance of the proposed SafeW with the baseline method and three state-of-the-art semi-supervised regression methods. a) Baseline k-NN method, which is a direct supervised nearest neighbor algorithm trained on the labeled data only. b) COREG [47]: a representative semi-supervised regression method based on co-training [31]. This algorithm uses two k-nearest neighbor regressors with different distance metrics, each of which labels the unlabeled data for the other regressors where the labeling confidence is estimated through consulting the influence of the labeling of unlabeled examples on the labeled ones. c) Self-kNN: Semi-supervised extension of the supervised kNN method based on self-training [48]. It first trains a supervised kNN method based on only labeled instances, and then predict the label of unlabeled instances After that, by adding the predicted labels on the unlabeled data as "ground-truth", another supervised kNN method is trained. This process is repeated until predictions on the unlabeled data no longer change or a maximum number of iteration achieves. d) Self-LS: Semi-supervised extension of the supervised least square method [49] based on self-training, which is similar to Self-kNN except that the supervised method is adapted to the least square regression. e) We also compare with the voting method, which uniformly weights multiple base learners. This approach is found promising in practice [19]. f) We also report the results of the oracle method: OpW (Optimal Weighting) that learns the optimal weight according to the ground-truth which we cannot obtain in real applications.

For the baseline 1NN method, the euclidean distance is used to locate the nearest neighbor. For the Self-kNN method, the euclidean distance is used and k is set to 3. The maximum number of iteration is set to 5 and further increasing it does not improve performance. For the Self-LS method, the parameters related to the importance of the labeled and unlabeled instances are set to 1 and 0.1, respectively. For the COREG method, the parameters are set to the recommended one in the package and the two distance metrics are employed by the euclidean and Mahalanobis distances. For the Voting method and the proposed SafeWmethod, 3 semi-supervised regressors are used where one is from the Self-LS method and the other two are from the Self-kNN methods employing the euclidean and the Cosine distance, respectively. For the proposed SafeW, the parameter δ is set by 5-fold cross validation from the range $[0.5u, 0.7u]$. In our experiments, all the features and labels are normalized into [0,1]. For each data set, 5 and 10 labeled instances are randomly selected and the rest ones are unlabeled data. The experiment is repeated for 30 times, and the average performance (mean±std) on the unlabeled data is reported.

Table 3 shows the Mean Square Error of the compared methods and the proposal on 5 and 10 labeled instances. We have the following observations from Table 3. i) Self-kNN generally improves the performance, however, it causes serious performance degradation in 2 cases. ii) Self-LS is not effective. One possible reason is the performance of supervised LS is not as good as that of kNN in our experimental data sets. iii) COREG achieves good performance, whereas it also will significantly decrease the performance in some cases. iv) The Voting method improves both the average performance of Self-kNN and Self-LS, but in 6 cases it significantly decreases the performance. v) The proposed method achieves significant improvement in 6 and 7 cases, which are the most among all the compared methods on 5 and 10 labeled instances, respectively. It also obtains the best average performance. What is more important, it does not seriously reduce the performance. vi) The OpW method cannot achieve 0 error which means that the assumption in Theorem 1 is usually not satisfied, however, the proposal still achieves safe results. This observation demonstrates that SafeW is robust to the assumption.

Overall the proposal improves the safeness of semi-supervised learning, in addition, obtains highly competitive performance compared with state-of-the-art approaches.

6.2 Domain Adaptation

We conduct compared experiments for domain adaptation on two benchmark datasets,[4] i.e., *20Newsgrous* and *Landmine*. The *20Newsgroups* dataset [50] contains 19,997 documents and is partitioned into 20 different newsgroups. Following the setup in [33], [51], we generate six different cross-domain data sets by utilizing its hierarchical structure. Specifically, the learning task is defined as the top-category binary classification, where our goal is to classify documents into one of the top-categories. For each data set, two top-categories are chosen, one as positive and another as negative. Then we select some subcategories under the positive and negative classes respectively to form a domain. In this work, we use documents from four top-categories: *Comp*, *Rec*, *Sci* and *Talk* to generate data sets.

The *Landmine* dataset is a detection dataset which contains 29 domains and 9 features. The data from domain 1 to domain 5 are collected from a leafy area; the data from Domain 20 to domain 24 are collected from a sand area. We use the whole data from domain 1 to domain 5 as the source domain and the data from domain 20 to domain 24 as five target domains. For *20newsgroup*, following [52], we randomly select 10 percent instances in the target domain as the labeled data and use 300 most important features as the representation. For *Landmine*, 5 percent instances in the target domain are used as the labeled data.

We compare the performance of the proposed SafeW with the baseline method and 3 state-of-the-art domain

3. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

4. http://www.cse.ust.hk/TL/

TABLE 3
Mean Square Error (mean±std) for the Compared Methods and SafeW Using 5 and 10 Labeled Instances

Dataset	1NN	Self-*k*NN	Self-LS	COREG	Voting	OpW	SafeW
5 labeled instances							
abalone	.017 ± .007	**.014 ± .003**	**.013 ± .004**	**.013 ± .003**	**.012 ± .003**	**.005 ± .001**	**.013 ± .003**
bodyfat	.024 ± .008	.025 ± .009	$\boxed{.054 \pm .016}$	.026 ± .008	$\boxed{.031 \pm .011}$	**.018 ± .003**	.025 ± .009
cadata	.090 ± .031	**.073 ± .023**	**.067 ± .022**	**.069 ± .028**	**.069 ± .022**	**.039 ± .014**	**.070 ± .023**
cpusmall	.027 ± .012	$\boxed{.031 \pm .008}$	$\boxed{.050 \pm .021}$	$\boxed{.031 \pm .009}$	.024 ± .006	**.014 ± .003**	.028 ± .009
eunite2001	.052 ± .017	**.037 ± .015**	**.024 ± .012**	**.037 ± .011**	**.031 ± .013**	**.018 ± .005**	**.032 ± .010**
housing	.042 ± .007	.043 ± .009	$\boxed{.048 \pm .012}$	.041 ± .008	.042 ± .009	**.024 ± .002**	.041 ± .009
mg	.071 ± .035	**.057 ± .015**	**.053 ± .011**	**.054 ± .019**	**.054 ± .013**	**.028 ± .009**	**.053 ± .013**
mpg	.029 ± .012	.030 ± .012	$\boxed{.040 \pm .014}$	.031 ± .012	.031 ± .012	**.016 ± .002**	.030 ± .012
pyrim	.032 ± .009	**.027 ± .005**	$\boxed{.063 \pm .012}$	.029 ± .011	**.025 ± .007**	**.013 ± .002**	**.025 ± .005**
space_ga	.005 ± .002	.005 ± .003	$\boxed{.030 \pm .005}$	**.004 ± .002**	$\boxed{.008 \pm .002}$	**.001 ± .000**	**.004 ± .002**
Ave. Mse.	.039	.034	.044	.033	.033	.020	.032
Win/Tie/Loss against 1NN		5/4/1	4/0/6	5/4/1	5/3/2	**9/0/0**	**6/4/0**
10 labeled instances							
Dataset	1NN	Self-*k*NN	Self-LS	COREG	Voting	OpW	SafeW
abalone	.020 ± .010	**.014 ± .005**	**.013 ± .004**	**.012 ± .003**	**.012 ± .003**	**.004 ± .001**	**.013 ± .005**
bodyfat	.019 ± .005	.019 ± .007	$\boxed{.041 \pm .013}$	.020 ± .006	$\boxed{.023 \pm .009}$	**.010 ± .002**	.018 ± .007
cadata	.083 ± .029	**.063 ± .012**	**.056 ± .007**	**.054 ± .010**	**.057 ± .009**	**.033 ± .011**	**.060 ± .013**
cpusmall	.024 ± .012	$\boxed{.027 \pm .008}$	$\boxed{.042 \pm .004}$	$\boxed{.028 \pm .008}$	**.020 ± .005**	**.012 ± .003**	.025 ± .008
eunite2001	.044 ± .014	**.037 ± .013**	**.020 ± .006**	**.031 ± .009**	**.029 ± .009**	**.017 ± .002**	**.029 ± .007**
housing	.039 ± .010	.036 ± .009	.036 ± .009	**.035 ± .005**	**.034 ± .008**	**.021 ± .003**	**.035 ± .009**
mg	.062 ± .019	**.046 ± .015**	**.048 ± .011**	**.045 ± .015**	**.043 ± .014**	**.024 ± .004**	**.045 ± .014**
mpg	.022 ± .007	.020 ± .006	$\boxed{.030 \pm .014}$	.021 ± .007	.021 ± .008	**.011 ± .001**	.020 ± .006
pyrim	.023 ± .006	**.021 ± .005**	$\boxed{.052 \pm .014}$	.022 ± .006	**.020 ± .007**	**.009 ± .001**	**.020 ± .006**
space_ga	.004 ± .001	**.003 ± .001**	$\boxed{.028 \pm .002}$	**.003 ± .001**	$\boxed{.006 \pm .001}$	**.000 ± .000**	**.003 ± .001**
Ave. Mse.	.034	.029	.037	.027	.026	.016	.027
Win/Tie/Loss against 1NN		6/3/1	4/1/5	6/3/1	7/1/2	**9/0/0**	**7/3/0**

For the compared methods, if the performance is significantly better/worse than the baseline method LR, the corresponding entries are then bolded/boxed. The average performance is listed for comparison. The win/tie/loss counts against the baseline method are summarized and the method with the smallest number of losses is bolded.

adaptation methods. a) Baseline supervised LR method, which trains a supervised logistic regression model for the labeled data in the target domain only. b) Baseline domain adaptation method which simply combines the data in the source and target domain together to train a supervised model. c) MIDA (**M**aximum **I**ndependence **D**omain **A**daptation) method [53], which is a feature-level transfer learning algorithm that learns a domain-invariant subspace between the source domain and target domain, and trained a supervised model on the learned subspace. d) TCA (**T**ransfer **C**omponent **A**nalysis) method [54], which is also a feature-level transfer learning algorithm, and achieves success in many domain adaptation tasks. e) TrAdaBoost method [33], which uses boosting [55] to select the most useful data in the source domain and has been proved as a powerful transfer learning method. f) The OpW method that has been mentioned previously.

For MIDA and TCA, the kernel type is set to the linear kernel and the dimension of the subspace is set to 30. For MIDA, TCA and the Original method, Logistic Regression model is employed as the supervised model on the feature space. For TrAdaBoost, SVM is adopted as the base learner and the number of iterations is set to 20. MIDA, TCA and the Original method are used as our base learners. Parameter δ is set by 5-fold cross validation from the range $[0.5u, 0.7u]$. Experiments are repeated for 30 times and the average accuracies on the unlabeled instances are reported.

Results are shown in Tables 4. We can see that, Original, MIDA and TCA methods degenerate the performance in many cases, while SafeW does not suffer such a deficiency. Moreover, in terms of average performance, SafeW achieves the best result. Therefore, our proposal achieves highly competitive performance with compared methods while more importantly, unlike previous methods that will hurt performance in some cases, it does not degenerate the performance. Besides, the OpW method still cannot achieve 100 percent accuracy which demonstrates that SafeW is robust to the safeness assumption.

6.3 Multi-Instance Learning

For multi-instance learning task, we evaluate the proposed methods on five benchmark data sets popularly used in the studies of MIL, including *Musk1*, *Musk2*, *Elephant*, *Fox*, *Tiger*.[5] In addition, two commonly used MIL datasets, i.e., *Birds* [56] and *SIVAL* [57] are also being used in experiments.

We compare the performance of the proposed SafeW with 2 baseline methods and 5 state-of-the-art domain adaptation methods. a) Baseline SI-SVM method, which assigns the label of its bag to each instance. The classifier assigns a label to each instance. b) miSVM [39], which is a transductive SVM. Instances inherit their bag label. The SVM is trained and classify each instance in the dataset. It is then retrained using the new label assignments. This procedure is repeated until the labels remain stable. c) C-*k*NN [38], which is an adaptation of *k*NN to MIL problems. The distance between the two bags is measured using the minimum Hausdorff distance. C-*k*NN

5. http://www.uco.es/grupos/kdis/momil/

TABLE 4
Classification Accuracy (mean $\pm$ std) of Domain Adaptation Task for the Compared Methods and SafeW on 20newsgroup and Landmine Datasets

Dataset	LR	Original	MIDA	TCA	TrAdaBoost	Voting	OpW	SafeW
20newsgroup								
Comp vs Rec	$.703 \pm .009$	$\mathbf{.749 \pm .014}$	$\mathbf{.796 \pm .020}$	$\mathbf{.794 \pm .016}$	$\mathbf{.808 \pm .016}$	$\mathbf{.796 \pm .014}$	$\mathbf{.889 \pm .010}$	$\mathbf{.796 \pm .017}$
Comp vs Sci	$.823 \pm .066$	$\boxed{.799 \pm .019}$	$\mathbf{.895 \pm .019}$	$.826 \pm .017$	$\mathbf{.858 \pm .020}$	$\mathbf{.855 \pm .024}$	$\mathbf{.924 \pm .019}$	$\mathbf{.893 \pm .021}$
Comp vs Talk	$.842 \pm .069$	$\boxed{.802 \pm .018}$	$\boxed{.823 \pm .016}$	$.843 \pm .011$	$\boxed{.825 \pm .014}$	$\boxed{.823 \pm .017}$	$\mathbf{.893 \pm .015}$	$.845 \pm .016$
Sci vs Talk	$.729 \pm .105$	$.710 \pm .012$	$\mathbf{.746 \pm .016}$	$\boxed{.702 \pm .009}$	$.717 \pm .021$	$.729 \pm .043$	$\mathbf{.824 \pm .010}$	$\mathbf{.747 \pm .015}$
Rec vs Sci	$.801 \pm .076$	$\boxed{.775 \pm .016}$	$.803 \pm .015$	$\mathbf{.844 \pm .012}$	$.802 \pm .015$	$.814 \pm .024$	$\mathbf{.901 \pm .015}$	$\mathbf{.844 \pm .016}$
Rec vs Talk	$.828 \pm .045$	$.828 \pm .012$	$\mathbf{.857 \pm .011}$	$\mathbf{.858 \pm .013}$	$\mathbf{.842 \pm .011}$	$\mathbf{.857 \pm .012}$	$\mathbf{.913 \pm .012}$	$\mathbf{.858 \pm .011}$
Average	.787	.777	.820	.811	.808	.807	.891	.831
Win/Tie/Loss against LR		1/2/3	4/1/1	3/2/1	3/2/1	3/2/1	6/0/0	**5/1/0**
Landmine								
Domain-20	$.922 \pm .017$	$.924 \pm .003$	$\mathbf{.927 \pm .004}$	$.926 \pm .005$	$.918 \pm .003$	$.924 \pm .004$	$\mathbf{.963 \pm .003}$	$\mathbf{.927 \pm .004}$
Domain-21	$.936 \pm .010$	$\boxed{.931 \pm .005}$	$.938 \pm .005$	$\boxed{.930 \pm .005}$	$\boxed{.926 \pm .003}$	$.935 \pm .006$	$\mathbf{.977 \pm .004}$	$.940 \pm .004$
Domain-22	$.959 \pm .005$	$.956 \pm .004$	$\boxed{.951 \pm .007}$	$\mathbf{.965 \pm .002}$	$\boxed{.910 \pm .003}$	$.960 \pm .004$	$\mathbf{.994 \pm .002}$	$\mathbf{.965 \pm .002}$
Domain-23	$.936 \pm .010$	$\boxed{.931 \pm .004}$	$\mathbf{.942 \pm .005}$	$\boxed{.931 \pm .005}$	$\mathbf{.963 \pm .004}$	$\mathbf{.947 \pm .003}$	$\mathbf{.981 \pm .003}$	$\mathbf{.943 \pm .004}$
Domain-24	$.954 \pm .005$	$.952 \pm .003$	$\boxed{.945 \pm .003}$	$\boxed{.943 \pm .003}$	$.954 \pm .003$	$.953 \pm .002$	$\mathbf{.989 \pm .003}$	$.955 \pm .002$
Average	.941	.939	.941	.939	.934	.943	.981	.946
Win/Tie/Loss against LR		0/3/2	2/1/2	1/1/3	1/2/2	1/4/0	**5/0/0**	**3/2/0**

relies on a two-level voting scheme. This algorithm was widely used in instance classification [58]. d) CCE [59], which is based on clustering and classifier ensembles. At first, the feature space is clustered using a fixed number of clusters. The bags are represented as binary vectors in which each bit corresponds to a cluster. The binary codes are utilized to train one of the classifiers in the ensemble. e) MIBoosting [60]: This method is essentially the same as the gradient boosting except that the loss function is based on bag classification error. The instance is classified individually and their labels are combined to obtain bag labels. f) mi-Graph [41]: This method represents each bag by a graph in which instances correspond to nodes. Cliques are identified in the graph to adjust the instances weight. Instances belonging to larger cliques have lower weight so that every concept present in the bag is equally represented when instances are averaged. A graph kernel captures the similarity between bags and is used in an SVM. g) We also compare with the Voting method, which uniformly weight multiple base learners.

For *Birds* and *SIVAL*, we adopt the *Brown Creeper* and *Apple* as the target class, respectively. For C-kNN, we set refs = 1 and citers = 5. For SI-SVM and mi-SVM, we adopt Libsvm as the implementation and use the RBF kernel. For CCE, MIBoosting, and miGraph, we set all the parameters as the recommended one. For the Voting method and SafeW, we adopt SI-SVM, mi-SVM, C-kNN and mi-Graph as the base learners. The parameter δ is set by 5-fold cross validation from the range [$0.3u$, $0.8u$]. Experiment for each dataset is repeated for 10 times and the average accuracy is reported.

Table 5 shows the accuracy of compared methods and the proposal on 7 datasets. From the results, we can see that, CCE, C-kNN, and MIBoosting degenerate the performance in many cases, while SafeW does not suffer such a deficiency. miGraph achieves the best average performance, but the proposed SafeW achieves the smallest number of losses against the baseline method. Besides, compared with the naive ensemble methods, SafeW also achieves better performance. This validates the effectiveness of SafeW.

6.4 Label Noise Learning

We conduct experimental comparison for label noise learning on a number of frequently-used classification datasets,[6] i.e., *Australian*, *Breast-Cancer*, *Diabetes*, *Digit1*, *Heart*, *Ionosphere*, *Splice* and *USPS*. For each data set, 80 percent of instances are used for training and the rest are used for testing. In the training set, 70 percent of instances are randomly selected as the noisy or weakly labeled data and the rest ones are high-quality labeled data. For the noisy labeled data, their labels are randomly reversed with a probability $p\%$ where p ranges from 10 percent to 40 percent with an interval 10 percent.

We compare the performance of the proposed SafeW with the following methods. a) Baseline Sup-SVM method, which is a supervised SVM trained on only high-quality labeled data. b) Bagging, which is regarded as to be robust with label noisy [7]. c) rLR (Robust Logistic Regression) [61], that enhances the logistic regression model to handle label noise. d) 3 classic classification methods: SVM, LR (Logistic Regression), k-NN with regardless of label noise. For LR, the *glmfit* function in Matlab is used. For k-NN method, k is set to 3. For Sup-SVM and SVM method, Libsvm package [62] is adopted and the kernel is set to RBF kernel. For Bagging method, we adopt the decision tree as the base learner. For rLR method, the parameter is set to the recommended one. For SafeW, LR, SVM, and k-NN are invoked as base learners and parameter δ is set by 5-fold cross validation from the range $[0.5u, 0.7u]$. Experiments are repeated for 30 times, and the average classification accuracy is reported.

Fig. 3 shows how the performance varies with the increase of noisy data. From Fig. 3 we can have the following observations. i) As the noise ratio increases, the accuracies of compared methods generally decrease; ii) Compared with the baseline method, all the compared methods

6. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

TABLE 5
Accuracy (mean ± std) for Compared Methods and SafeW on 7 Datasets

	SI-SVM	CCE	miSVM	C-kNN	MIBoosting	miGraph	Voting	SafeW
Musk1	.840 ± .119	.831 ± .027	**.869 ± .120**	.849 ± .143	.837 ± .120	**.889 ± .073**	**.881 ± .079**	**.869 ± .101**
Musk2	.853 ± .101	$\boxed{.723 \pm .019}$	.838 ± .085	**.875 ± .131**	$\boxed{.790 \pm .088}$	**.903 ± .086**	**.879 ± .049**	**.884 ± .082**
Fox	.546 ± .092	**.599 ± .027**	**.582 ± .102**	**.576 ± .016**	**.638 ± .102**	**.616 ± .079**	**.590 ± .034**	**.590 ± .051**
Elephant	.801 ± .088	.793 ± .021	**.825 ± .073**	$\boxed{.785 \pm .016}$	**.827 ± .073**	**.869 ± .078**	**.825 ± .049**	**.819 ± .053**
Tiger	.778 ± .092	$\boxed{.758 \pm .012}$	**.789 ± .089**	$\boxed{.757 \pm .017}$	.784 ± .085	**.801 ± .083**	.779 ± .017	**.790 ± .031**
SIVAL	.761 ± .071	$\boxed{.715 \pm .053}$	.771 ± .110	$\boxed{.735 \pm .151}$	$\boxed{.715 \pm .064}$	.756 ± .035	$\boxed{.737 \pm .029}$	.755 ± .047
Birds	.720 ± .121	$\boxed{.690 \pm .095}$	.720 ± .090	.707 ± .090	$\boxed{.643 \pm .141}$	$\boxed{.663 \pm .084}$	.713 ± .081	.713 ± .090
Average	.757	.730	.771	.755	.748	.785	.772	.774
Win/Tie/Loss against SI-SVM		1/2/4	4/3/0	2/2/3	2/2/3	5/1/1	4/2/1	**5/2/0**

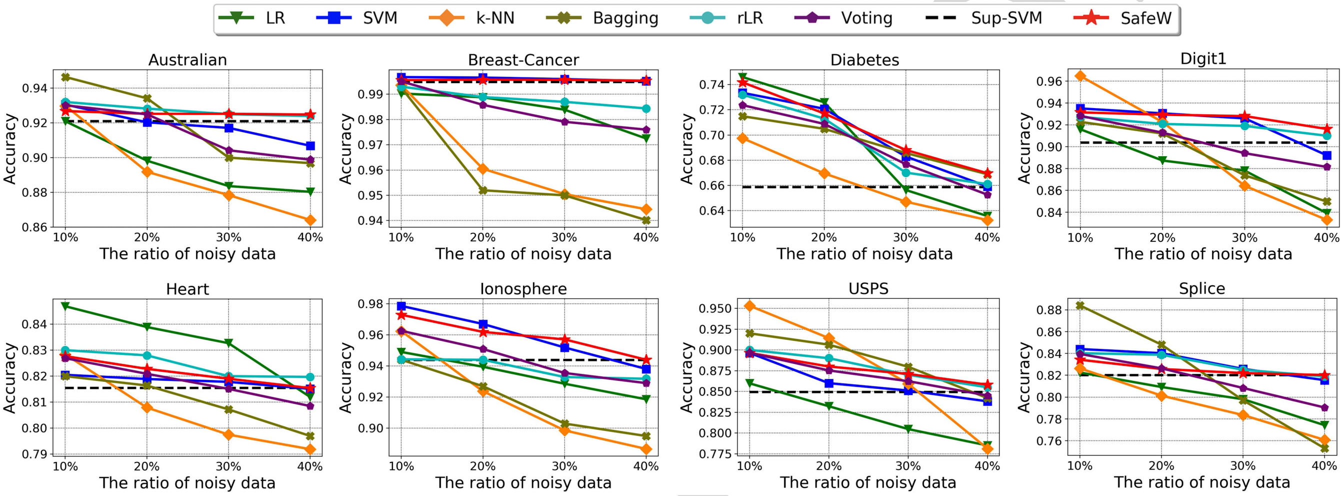

Fig. 3. Classification accuracy of compared methods with different numbers of noise ratio.

perform worse than Sup-SVM in many cases, especially when the noise ratio becomes larger, while our proposed SafeW does not suffer from such deficiency. iii) The proposed SafeW achieves best average performance.

Overall, our proposal achieves highly competitive performance compared with state-of-the-art label noise learning methods and never performs worse than the baseline Sup-SVM method. These demonstrate the effectiveness of the SafeW method.

7 Conclusion

In this paper, we study safe weakly supervised learning that will not hurt performance with the use of weakly supervised data. This problem is crucial whereas has not been extensively studied. Based on our preliminary work [20], [63], in this paper we present a scheme to derive a safe prediction by integrating multiple weakly supervised learners. The resultant formulation has a safeness guarantee for many commonly used convex loss functions in classification and regression. Besides, it is capable of involving prior knowledge about the weight of base learners. Further, it can be globally solved efficiently and extensive experiments validate the effectiveness of our proposed algorithms. In future, it is necessary to study safe weakly supervised learning with adversarial examples.

Acknowledgments

The authors want to thank the associate editor and reviewers for helpful comments and suggestions. This research was supported by the National Key R&D Program of China (2018YFB1004300) and the National Natural Science Foundation of China (61772262). Yu-Feng Li and Lan-Zhe Guo contribute equally to this work.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2017.

[3] R. Krishna, Y. Zhu, O. Groth, J. Johnson, et al., "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 32–73, 2017.

[4] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.

[5] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[6] M.-A. Carbonneau, V. Cheplygina, E. Granger, and G. Gagnon, "Multiple instance learning: A survey of problem characteristics and applications," *Pattern Recognit.*, vol. 77, pp. 329–353, 2018.

[7] B. Frénay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.

[8] N. V. Chawla and G. Karakoulas, "Learning from labeled and unlabeled data: An empirical study across techniques and domains," *J. Artif. Intell. Res.*, vol. 23, pp. 331–366, 2005.

[9] Y.-F. Li and Z.-H. Zhou, "Towards making unlabeled data never hurt," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 175–188, Jan. 2015.

[10] Y.-F. Li, J. T. Kwok, and Z.-H. Zhou, "Towards safe semi-supervised learning for multivariate performance measures," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1816–1822.

[11] M. T. Rosenstein, Z. Marx, and L. P. Kaelbling, "To transfer or not to transfer," in *Proc. NIPS Workshop "Inductive Transfer: 10 Years Later"*, 2005.

[12] A. Argyriou, A. Maurer, and M. Pontil, "An algorithm for transfer learning in a heterogeneous environment," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2008, pp. 71–85.

[13] L. Ge, J. Gao, H. Ngo, K. Li, and A. Zhang, "On handling negative transfer and imbalanced distributions in multiple source transfer learning," *Statistical Anal. Data Mining*, vol. 7, no. 4, pp. 254–271, 2014.

[14] B. Bakker and T. Heskes, "Task clustering and gating for Bayesian multitask learning," *J. Mach. Learn. Res.*, vol. 4, pp. 83–99, 2003.

[15] A. Gaba and R. L. Winkler, "Implications of errors in survey data: A Bayesian model," *Manag. Sci.*, vol. 38, no. 7, pp. 913–925, 1992.

[16] R. J. Hickey, "Noise modelling and evaluating learning from examples," *Artif. Intell.*, vol. 82, no. 1–2, pp. 157–179, 1996.

[17] Y.-F. Li and D.-M. Liang, "Safe semi-supervised learning: A brief introduction," *Frontiers Comput. Sci.*, vol. 13, no. 4, pp. 669–676, 2019.

[18] A. Balsubramani and Y. Freund, "Optimally combining classifiers using unlabeled data," in *Proc. Int. Conf. Learn. Theory*, 2015, pp. 211–225.

[19] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.

[20] Y.-F. Li, H.-W. Zha, and Z.-H. Zhou, "Learning safe prediction for semi-supervised regression," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2217–2223.

[21] L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Comput.*, vol. 16, no. 5, pp. 1063–1076, 2004.

[22] J. M. Bates and C. W. Granger, "The combination of forecasts," *Oper. Res.*, vol. 20, no. 4, pp. 451–468, 1969.

[23] A. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915–936, 2003.

[24] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Res.*, vol. 30, no. 1, pp. 79–82, 2005.

[25] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[27] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*. London, U.K.: Oxford Univ. Press, 1997.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[29] D. J. Miller and H. S. Uyar, "A mixture of experts classifier with learning based on both labelled and unlabelled data," in *Proc. Advances Neural Inf. Process. Syst.*, 1997, pp. 571–577.

[30] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 912–919.

[31] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.

[32] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 1999, pp. 200–209.

[33] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 193–200.

[34] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 759–766.

[35] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Proc. Advances Neural Inf. Process. Syst.*, 2008, pp. 153–160.

[36] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising markov logic networks for transfer learning," in *Proc. AAAI Conf. Artif. Intell.*, 2007, pp. 608–614.

[37] Q. Zhang and S. A. Goldman, "EM-DD: An improved multiple-instance learning technique," in *Proc. Advances Neural Inf. Process. Syst.*, 2001, pp. 1073–1080.

[38] J. Wang and J. Zucker, "Solving the multiple-instance problem: A lazy learning approach," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 1119–1126.

[39] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. Advances Neural Inf. Process. Syst.*, 2002, pp. 561–568.

[40] X. Xu and E. Frank, "Logistic regression and boosting for labeled bags of instances," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2004, pp. 272–281.

[41] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li, "Multi-instance learning by treating instances as non-IID samples," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 1249–1256.

[42] S. Ray and M. Craven, "Supervised versus multiple instance learning: An empirical comparison," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 697–704.

[43] M. Carbonneau, E. Granger, and G. Gagnon, "Witness identification in multiple instance learning using random subspaces," in *Proc. Int. Conf. Pattern Recognit.*, 2016, pp. 3639–3644.

[44] L. Fan, X. Li, Q. Guo, and C. Zhang, "Nonlocal image denoising using edge-based similarity metric and adaptive parameter selection," *Sci. China Inf. Sci.*, vol. 61, no. 4, pp. 049 101:1–049 101:3, 2018.

[45] N. Manwani and P. Sastry, "Noise tolerance under risk minimization," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1146–1151, Jun. 2013.

[46] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, 2000.

[47] Z.-H. Zhou and M. Li, "Semi-supervised regression with co-training," in *Proc. Int. Joint Conf. Artif. Intell.*, 2005, pp. 908–913.

[48] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. Annu. Meet. Assoc. Comput. Linguistics*, 1995, pp. 189–196.

[49] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin, Germany: Springer, 2001.

[50] K. Lang, "Newsweeder: Learning to filter netnews," in *Proc. Int. Conf. Mach. Learn.*, 1995, pp. 331–339.

[51] L. Li, X. Jin, and M. Long, "Topic correlation analysis for cross-domain text classification," in *Proc. AAAI Conf. Artif. Intell.*, 2012, pp. 998–1004.

[52] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-bridged PLSA for cross-domain text classification," in *Proc. Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 627–634.

[53] K. Yan, L. Kou, and D. Zhang, "Learning domain-invariant subspace using domain features and independence maximizationm," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 288–299, 2017.

[54] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[55] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*, 1995, pp. 23–37.

[56] F. Briggs, X. Z. Fern, and R. Raich, "Rank-loss support instance machines for miml instance annotation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 534–542.

[57] R. Rahmani, S. A. Goldman, H. Zhang, J. Krettek, and J. E. Fritts, "Localized content based image retrieval," in *Proc. ACM SIGMM Int. Workshop Multimedia Inf. Retrieval*, 2005, pp. 227–236.

[58] Z.-H. Zhou, X.-B. Xue, and Y. Jiang, "Locating regions of interest in CBIR with multi-instance learning techniques," in *Proc. Australasian Joint Conf. Artif. Intell.*, 2005, pp. 92–101.

[59] Z.-H. Zhou and M.-L. Zhang, "Solving multi-instance problems with classifier ensemble based on constructive clustering," *Knowl. Inf. Syst.*, vol. 11, no. 2, pp. 155–170, 2007.

[60] C. Zhang, J. C. Platt, and P. A. Viola, "Multiple instance boosting for object detection," in *Proc. Advances Neural Inf. Process. Syst.*, 2006, pp. 1417–1424.

[61] J. Bootkrajang and A. Kabán, "Label-noise robust logistic regression and its applications," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2012, pp. 143–158.

[62] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

[63] L.-Z. Guo and Y.-F. Li, "A general formulation for safely exploiting weakly supervised data," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3126–3133.

Yu-Feng Li received the BSc and PhD degrees in computer science from Nanjing University, China, in 2006 and 2013, respectively. He joined the National Key Laboratory for Novel Software Technology at Nanjing University in 2013, and is currently an associate professor. He is a member of the LAMDA group. His research interests include mainly in machine learning. Particularly, he is interested in weakly supervised learning, statistical learning and optimization. He has received outstanding doctoral dissertation award from China Computer Federation (CCF), outstanding doctoral dissertation award from Jiangsu Province and Microsoft Fellowship Award. He has published more than 40 papers in top-tier journals and conferences such as the *Journal of Machine Learning Research*, the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *Artificial Intelligence*, the *IEEE Transactions on Knowledge and Data Engineering*, ICML, NIPS, AAAI, etc. He is/was served as an editorial board member of machine learning journal special issues, co-chair of ACML18 workshop and ACML19 tutorial, and a senior PC member of top-tier conferences such as IJCAI19/17/15, AAAI19.

Lan-Zhe Guo received the BSc degree in 2017. He is currently working toward the PhD degree in the National Key Laboratory for Novel Software Technology at Nanjing University, China. His research interests include in machine learning. Particularly, he is interested in weakly-supervised learning.

Zhi-Hua Zhou (S'00-M'01-SM'06-F'13) received the BSc, MSc, and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honors. He joined the Department of Computer Science & Technology, Nanjing University as an assistant professor in 2001, and is currently professor, head of the Department of Computer Science and Technology, and dean of the School of Artificial Intelligence; he is also the founding director of the LAMDA group. His research interests include in artificial intelligence, machine learning and data mining. He has authored the books *Ensemble Methods: Foundations and Algorithms* and *Machine Learning* (in Chinese), and published more than 150 papers in top-tier international journals or conference proceedings. He has received various awards/honors including the National Natural Science Award of China, the IEEE Computer Society Edward J. McCluskey Technical Achievement Award, the PAKDD Distinguished Contribution Award, the IEEE ICDM Outstanding Service Award, the Microsoft Professorship Award, etc. He also holds 24 patents. He is the editor-in-chief of the *Frontiers of Computer Science*, associate editor-in-chief of the *Science China Information Sciences*, Action or associate editor of the *Machine Learning*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *ACM Transactions on Knowledge Discovery from Data*, etc. He served as associate editor-in-chief for *Chinese Science Bulletin* (2008-2014), associate editor for *IEEE Transactions on Knowledge and Data Engineering* (2008-2012), *IEEE Transactions on Neural Networks and Learning Systems* (2014-2017), *ACM Transactions on Intelligent Systems and Technology* (2009-2017), Neural Networks (2014-2016), etc. He founded ACML (Asian Conference on Machine Learning), served as Advisory Committee member for IJCAI (2015-2016), Steering Committee member for ICDM, PAKDD and PRICAI, and chair of various conferences such as general co-chair of ICDM 2016 and PAKDD 2014, Program co-chair of AAAI 2019 and SDM 2013, and area chair of NIPS, ICML, AAAI, IJCAI, KDD, etc. He is/was the chair of the IEEE CIS Data Mining Technical Committee (2015-2016), the chair of the CCF-AI (2012-), and the chair of the CAAI Machine Learning Technical Committee (2006-2015). He is a foreign member of the Academy of Europe, and a fellow of the ACM, AAAI, AAAS, IEEE, IAPR, IET/IEE, CCF, and CAAI. He is a fellow of the IEEE.