A General Formulation for Safely Exploiting Weakly Supervised Data*

Lan-Zhe Guo and Yu-Feng Li
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
\{guolz, liyf\}@lamda.nju.edu.cn

Abstract
Weakly supervised data is an important machine learning data to help improve learning performance. However, recent results indicate that machine learning techniques with the usage of weakly supervised data may sometimes cause performance degradation. Safely leveraging weakly supervised data is important, whereas there is only very limited effort, especially on a general formulation to help provide insight to guide safe weakly supervised learning. In this paper we present a scheme that builds the final prediction results by integrating several weakly supervised learners. Our resultant formulation brings two advantages. i) For the commonly used convex loss functions in both regression and classification tasks, safeness guarantees exist under a mild condition; ii) Prior knowledge related to the weights of base learners can be embedded in a flexible manner. Moreover, the formulation can be addressed globally by simple convex quadratic or linear program efficiently. Experiments on multiple weakly supervised learning tasks such as label noise learning, domain adaptation and semi-supervised learning validate the effectiveness.

Introduction
Weakly supervised data is commonly appear in real applications (Zhou 2017). Compared to the data in traditional supervised learning, weakly supervised data does not require a large amount of precise label information. Examples includes label noise learning (Fréñay and Verleysen 2014) where label information contains noise; domain adaptation (Pan and Yang 2010) where label information in target domain is not sufficient and one needs to exploit further label information from other domains; semi-supervised learning (Chapelle et al. 2006) where label information is scarce and one needs to leverage a number of additional unlabeled data. Because weakly supervised data loosens the constraint for the label information in learning tasks, it has a broad application prospect, such as image classification (Krishna et al. 2017), natural language processing (Alfonseca et al. 2012) and so on. Taking advantage of weakly supervised data to help build effective learning methods has gained extensive attention and obtained a lot of results.

It is often expected that, machine learning techniques exploiting weakly supervised data are able to improve learning performance. However, recent studies show that machine learning techniques with the use of weakly supervised data may sometimes lead to performance degradation. That is, the learning performance is even worse than that of baseline method without using weakly supervised data. For example, label noise learning may be worse than learning from only a small number of high-quality labeled data (Fréñay and Verleysen 2014); domain adaptation methods may have the phenomenon of negative transfer (Pan and Yang 2010) that the source domain data contribute to the reduced performance of learning in the target domain; semi-supervised learning using unlabeled data may degenerate learning performance, which has been reported in a number of studies (Chapelle et al. 2006; Chawla and Karakoulas 2005; Li and Zhou 2015). How to safely exploit weakly supervised data so that machine learning technology often outperforms and never be worse than the simple baseline, has become an important yet unsolved problem. Recently there is a few effort, but they typically work on a specific scenario of weakly supervised learning (Li and Zhou 2015; Balsubramani and Freund 2015; Li, Zha, and Zhou 2017; Wei et al. 2017). The proposal on generic formulation for various weakly supervised data, to our best knowledge, has not been thoroughly studied.

In this paper, we present a scheme that builds the final prediction results by integrating several weakly supervised learners. The resultant formulation brings some advantages. Firstly, for multiple commonly used convex loss functions (e.g., square loss, hinge loss) in both regression and classification tasks of weakly supervised learning, it has safeness guarantees under a mild condition. Secondly, it can flexibly embed uncertain prior knowledge about the weights of weakly supervised learners in regression and classification tasks. Moreover, our formulation can be addressed globally via simple convex quadratic program or linear program in an efficient manner. Experiments on multiple weakly supervised learning tasks such as label noise learning, domain adaptation and semi-supervised learning validate the effectiveness of our proposed algorithms.

This paper is organized as follows. We first review related

---

*This research was supported by the National Natural Science Foundation of China (61772262, 61403186, 61422304). Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
works and then present the proposed formulation. Next we
discuss the incorporation of prior knowledge, finally we present
the optimization algorithms.

Problem Setting and Formulation

In weakly supervised learning, due to the lack of sufficient
and accurate label information, ensemble learning (Zhou
2012) was recognized as a popular learning technology to
derive robust performance. Many ways can be employed
to generate multiple weakly supervised learners, such as
taking different learning models, different sampling strategies,
different model parameters, etc. Although previous
studies typically work on deriving good performance from
multiple learners, they may suffer from unsafeness. One
underlying reason is that the good performance derived by
previous studies is not explicitly compared with the baseline
method, and may sometimes mislead the learning process.
These motivate us to derive a new formulation.

Formally, suppose we have obtained \( n \) predictve results
\( \{y_1, \ldots, y_n\} \) of unlabeled instances from multiple weakly
supervised learners \( \{f_1, \ldots, f_n\} \), where \( y_i \in \mathbb{H}^n, i = 1, \ldots, n \) and \( n \) is the number of unlabeled instances. We let
\( \mathbb{H} = \mathbb{R} \) for regression task and \( \mathbb{H} = \{-1, 1\} \) for classification
result. Meanwhile, we let \( y_0 \in \mathbb{H}^n \) denote the baseline
result, e.g., obtained by training a supervised model with
only limited labeled data. Our goal is to derive a safe
prediction \( y = g(\{y_1, \ldots, y_n\}, y_0) \), which often outperforms,
meanwhile would not be worse than \( y_0 \).

We first consider a simpler case that the ground-truth label
assignment on unlabeled instances, denoted by \( y^* \), is known.
In this case, one can easily have the objective function that
maximizes the performance gain against the baseline \( y_0 \), as
\[
\max_{y \in \mathbb{H}^n} \ell(y_0, y^*) - \ell(y, y^*)
\]

Here \( \ell(\cdot, \cdot) \) is a loss function, e.g., mean square loss, hinge
loss, etc. The smaller the value of the loss function, the bet-
ter the performance. Table 1 summarizes some commonly
used loss functions for regression and classification. Obvi-
ously \( y^* \) is unknown and otherwise the solution is trivial.
To alleviate it, we consider that \( y^* \) is a convex combina-
tion of base learners. Specifically, \( y^* = \sum_{i=1}^{n} \alpha_i y_i \) where
\( \alpha = [\alpha_1; \alpha_2; \ldots; \alpha_n] \geq 0 \) be the non-negative weights of
base learners and \( \sum_{i=1}^{n} \alpha_i = 1 \).

We then have the following objective by replacing the definition of \( y^* \),
\[
\max_{y \in \mathbb{H}^n} \ell(y_0, \sum_{i=1}^{n} \alpha_i y_i) - \ell(y, \sum_{i=1}^{n} \alpha_i y_i)
\]

In practice, however, one may still be hard to know the
precise weights of base learners. We further consider that \( \alpha \)
is from a convex set \( \mathcal{M} \) and make our proposal more prac-
tical, where \( \mathcal{M} \) reflects the prior knowledge for the impor-
tance of base learners. The setup of \( \mathcal{M} \) will be discussed in
the later section. Without further information, we aim to opti-
mize the worst-case performance gain. We then can have a
general formulation with respect to regression as well as
classification task as,
\[
\max_{y \in \mathbb{H}^n} \min_{\alpha \in \mathcal{M}} \ell(y_0, \sum_{i=1}^{n} \alpha_i y_i) - \ell(y, \sum_{i=1}^{n} \alpha_i y_i)
\]
Table 1: Some commonly used loss functions $\ell(p, q)$ for regression and classification tasks. The prediction $q = [q_1; \ldots; q_n] \in \mathbb{R}^n$ and the label $p = [p_1; \ldots; p_n] \in \mathbb{H}^n$ where $\mathbb{H}^n = \mathbb{R}^n$ is for regression and $\mathbb{H}^n = \{+1, -1\}^n$ is for classification.

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Definition</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean square loss</td>
<td>$\ell(p, q) = \frac{1}{n} \sum_{i=1}^{n} (p_i - q_i)^2$</td>
<td>Regression &amp; Classification</td>
</tr>
<tr>
<td>Mean absolute loss</td>
<td>$\ell(p, q) = \frac{1}{n} \sum_{i=1}^{n}</td>
<td>p_i - q_i</td>
</tr>
<tr>
<td>Mean $\epsilon$-insensitive loss</td>
<td>$\ell(p, q) = \frac{1}{n} \sum_{i=1}^{n} \max{</td>
<td>p_i - q_i</td>
</tr>
<tr>
<td>Hinge loss</td>
<td>$\ell(p, q) = \frac{1}{n} \sum_{i=1}^{n} \max{0,</td>
<td>p_i - q_i</td>
</tr>
</tbody>
</table>

Study the Safeness

We show that for the commonly used convex loss functions as listed in Table 1 in both regression and classification tasks, safeness guarantees exist for Eq.(1) under a mild condition. We first introduce a result as follows.

**Theorem 1.** Suppose that the ground-truth $y^*$ can be constructed by the base learners, i.e., $y^* \in \{y_j \mid \sum_{i=1}^{b} \alpha_i y_{ji} \in \mathcal{M}\}. Let \tilde{y}$ and $\tilde{\alpha}$ be the optimal solution to Eq.(1), we then have $\ell(\tilde{y}, y^*) \leq \ell(y_0, y^*)$ and $\tilde{y}$ has already achieved the maximal performance gain against $y_0$.

**Proof.** We define $L(y, \alpha) = \ell(y_0, \sum_{i=1}^{n} \alpha_i y_{i}) - \ell(y, \sum_{i=1}^{n} \alpha_i y_{i})$.

The following inequality holds for any feasible $y$ and $\alpha$:

$L(y, \tilde{\alpha}) \leq L(y, \tilde{\alpha}) \leq L(y, \tilde{\alpha})$

According to the assumption, $y^* \in \{y_j \mid \sum_{i=1}^{b} \alpha_i y_{ji} \in \mathcal{M}\}$ and let $\alpha^*$ makes $y^* = \sum_{i=1}^{n} \alpha^*_i y_{i}$. By setting $y$ and $\alpha$ to be $y_0$ and $\alpha^*$, then we can deduce that

$\ell(y_0, y^*) \geq \ell(\tilde{y}, y^*)$

Moreover, since we already maximize the performance gain in the worst case, $\tilde{y}$ has already achieved the maximal performance gain against $y_0$. □

**Lemma 1.** When $\ell(\cdot, \sum_{i=1}^{n} \alpha_i y_{i})$ is convex to $\alpha$ and there exists $y \in \mathbb{R}^n$ such that $\ell(y, \sum_{i=1}^{n} \alpha_i y_{i}) = 0$, for any $\alpha$. In optimality, the optimal solution $\tilde{y}$ and $\tilde{\alpha}$ have the following relation, i.e., $\ell(\tilde{y}, \sum_{i=1}^{n} \tilde{\alpha}_i y_{i}) = 0$.

**Proof.** Assume, to the contrary, $\ell(\tilde{y}, \sum_{i=1}^{n} \tilde{\alpha}_i y_{i}) \neq 0$. According to the assumption, there exist $y^*$ such that $\ell(y^*, \sum_{i=1}^{n} \tilde{\alpha}_i y_{i}) = 0$. Obviously, $0 = \ell(\tilde{y}, \sum_{i=1}^{n} \tilde{\alpha}_i y_{i}) < \ell(\tilde{y}, \sum_{i=1}^{n} \tilde{\alpha}_i y_{i})$. Hence, $\tilde{y}$ is not optimal, a contradiction. □

**Theorem 2.** Under the same condition in Lemma 1, Eq.(1) is a convex optimization.

**Proof.** With Lemma 1, the form of Eq.(1) for regression task can be rewritten as,

$$\min_{\alpha \in \mathcal{M}} \ell(y_0, \sum_{i=1}^{n} \alpha_i y_{i})$$

(2)

Remind that $\ell(\cdot, \sum_{i=1}^{n} \alpha_i y_{i})$ is convex to $\alpha$, obviously, Eq.(1) is a convex optimization. □

The condition in Theorem 2 is rather mild. Many regression loss functions, for example, mean square loss, mean absolute loss (Willmott and Matsuura 2005) and mean $\epsilon$-insensitive loss (Smola and Schölkopf 2004), satisfy such a mild condition in Theorem 2.

Due to the discontinuous feasible field of $y$, Lemma 1 does not hold for most of the classification loss functions. It could not simply apply or extend the result in regression task to classification. Fortunately, we find that for some particular classification loss function like the hinge loss, the optimal solution of Eq.(1) is still possible.

**Lemma 2.** When $\ell(\cdot, \cdot)$ is realized as the hinge loss, in optimality, the optimal $\tilde{y}$ and $\tilde{\alpha}$ meet a relation $\tilde{y} = \text{sign}(\sum_{i=1}^{n} \tilde{\alpha}_i y_{i})$ where $\text{sign}(s)$ is the sign of value $s$.

The proof is in the supplementary material. We then have,

**Theorem 3.** Suppose $y_i \in \{+1, -1\}^n, \forall i = 1, \ldots, n$, Eq.(1) is a convex optimization when $\ell(\cdot, \cdot)$ is realized as the hinge loss.

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

$$\min_{\alpha \in \mathcal{M}} \ell(y_0, \sum_{i=1}^{n} \alpha_i y_{i}) - \ell(\text{sign}(\sum_{i=1}^{n} \alpha_i y_{i}), \sum_{i=1}^{n} \alpha_i y_{i})$$

(3)

Since $y_i \in \{+1, -1\}^n, \forall i = 1, \ldots, n$ and $\ell(\cdot, \sum_{i=1}^{n} \alpha_i y_{i})$ is the hinge loss, the form $\ell(\text{sign}(\sum_{i=1}^{n} \alpha_i y_{i}), \sum_{i=1}^{n} \alpha_i y_{i})$
can be equivalently rewritten as $1 - \frac{1}{u} \| \sum_{i=1}^{n} \alpha_i y_i \|_1$ using $\sum_{i=1}^{n} \alpha_i = 1$. Therefore, Eq. (3) is equal to,

$$\min_{\alpha \in \mathcal{M}} \ell(y_0, \sum_{i=1}^{n} \alpha_i y_i) + \frac{1}{u} \| \sum_{i=1}^{n} \alpha_i y_i \|_1 - 1 \quad (4)$$

Evidently, Eq. (4) is a convex optimization. \hfill \Box

Though hinge loss is the only loss function to help derive a global solution of Eq. (1) for classification task, hinge loss is well-known as one powerful loss function in classification. The reason for the particularity of hinge loss mainly lies in its linearity to the predictive results (the term $\frac{1}{u} \| \sum_{i=1}^{n} \alpha_i y_i \|_1$). Such a property, unfortunately, does not hold for other loss functions such as logistic loss, exponential loss, cross-entropy loss, etc.

**Weight the Base Learners**

One question remained is that how to setup $\mathcal{M}$. Obviously, the setup of $\mathcal{M}$ can be easily embedded with a variety of prior knowledge. For example, suppose base learner $f_i$ is more reliable than $f_j$ and the set of all such indexes $(i, j)$ is denoted as $S$, $\mathcal{M}$ could be set to $\{\alpha \mid \alpha_i - \alpha_j \geq 0, (i, j) \in S; \alpha^\top \mathbf{1} = 1; \alpha \geq 0\}$ where $\mathbf{1}$ refer to the all-one and all-zero vectors respectively; suppose the importance values of base learners are known and let $\{r_1, \ldots, r_n\}$ denote the importance values, one could set up $\mathcal{M}$ as $\{\alpha \mid -\gamma \leq \alpha_i - r_i \leq \gamma, \forall i = 1, \ldots, n; \alpha^\top \mathbf{1} = 1; \alpha \geq 0\}$ where $\gamma$ is a small constant, and so on and so forth.

In practice, one may be hard to obtain the precise prior knowledge of base learners, for example, the importance values. In this case we present to learn the weights of base learners. Before presenting the algorithms, we first investigate how the performance of our formulation is affected with the setup of $\mathcal{M}$. Assume that the loss function $\ell(\cdot, \cdot)$ is $\eta$-Lipschitz, i.e., $\|\ell(y_1, y_2) - \ell(y_1, y_3)\| \leq \eta \|y_2 - y_3\|$ for any $y_1, y_2, y_3 \in [-1, 1]$. Most of commonly used loss functions including the ones in Table 1 satisfy such property (Rosasco et al. 2004). Let $\beta^* = [\beta_1^*, \ldots, \beta_n^*] \in \mathcal{M}$ be the optimal solution to the objective,

$$\beta^* = \arg \min_{\beta \in \mathcal{M}} \ell(\sum_{i=1}^{n} \beta_i y_i, y^*)$$

and $\epsilon$ be the residual, i.e., $\epsilon = y^* - \sum_{i=1}^{n} \beta_i^* y_i$. We have the following result.

**Theorem 4.** *The performance gain of $\hat{y}$ against $y_0$, i.e., $\ell(y_0, y^*) - \ell(\hat{y}, y^*)$, has a lower-bound $-2\eta \| \epsilon \|_1$.***

The proof is in the supplementary material. Theorem 4 discloses that the worst-case of performance gain is lower-bounded by the norm of the residual. This motivates us to learn the weights of base learners such that the residual is minimized. We then present the learning approach for regression and classification through the idea of covariance matrix analysis (Bates and Granger 1969).

**Regression** Let $C^{reg}$ be the $n \times n$ covariance matrix of the $n$ base learners $\{f_1, \ldots, f_n\}$ 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 $\rho_n^{reg} = [\rho_1^{reg}; \ldots; \rho_n^{reg}]$ be the vector of covariances between the base learners and the ground-truth label assignment $f^*(X)$,

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

where $\theta = \mathbb{E}[f^*(X)]$. We minimize the residual for $\alpha$ as,

$$\alpha^* = \arg \min_{\alpha} \mathbb{E}[\text{MSE}(\sum_{i=1}^{n} \alpha_i f_i(X), f^*(X))] \quad (5)$$

where MSE refers to the Mean Squared Error. Eq. (5) has a closed-form solution (Bates and Granger 1969).

**Theorem 5.** *(Bates and Granger, 1969)* The optimal weights $\alpha^*$ satisfies

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

With Theorem 5, one need to estimate $C^{reg}$ and $\rho$. For $C^{reg}$, it is evident that $(y_i - \mu_i)^\top (y_j - \mu_j)$ is an unbiased estimation of $C_{ij}^{reg}$. Therefore, one could easily have $C^{reg}$ with elements

$$C_{ij}^{reg} = (y_i - \mu_i)^\top (y_j - \mu_j)$$

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

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

Therefore, we can setup $\mathcal{M}$ as $\{\alpha \mid \hat{C}^{reg} \alpha \geq \delta \alpha^\top \mathbf{1} = 1, \alpha \geq 0\}$, where $\delta$ is a constant, indicating that the base learners have a low-bad performance (e.g., are better than random-guess) (Balasubramani and Freund 2015). It is easy to verify that $\mathcal{M}$ is a convex set.

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

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

With classification accuracy to be the performance measure, it can be shown that,

**Theorem 6.** *The optimal weights $\alpha^*$ in classification satisfies that $\rho^{clf} = C^{clf} \alpha^*$.***

We can setup $\mathcal{M}$ as $\{\alpha \mid \hat{C}^{clf} \alpha \geq \delta \alpha^\top \mathbf{1} = 1, \alpha \geq 0\}$ where $C^{clf}$ is the unbiased estimation of $C^{clf}$, with elements $C_{ij}^{clf} = y_i^\top y_j$, $\mathcal{M}$ is also a convex set.

In summary, our formulation is able to directly absorb the precise prior knowledge about the importance of learners if available. It is also capable of incorporating with the estimation results obtained by covariance matrix analysis on both regression and classification tasks, when the precise prior knowledge is unavailable.
Efficient Optimization Algorithms

The formulation in Eq. (1) can be globally and efficiently addressed. For regression, we adopt mean square loss as the implementation. According to Lemma 1 and Theorem 2, Eq.(1) can be rewritten as,
\[
\min \sum_{i=1}^{n} \alpha_i \|y_i - y_0\|^2
\]
which is equivalent to the following form,
\[
\min \alpha^T F \alpha - v^T \alpha
\]
where \( F \in \mathbb{R}^{n \times n} \) is a linear kernel matrix of \( y_i \)'s, i.e, \( F_{ij} = y_i^T y_j \) and \( v = [2y_1^T y_0; \cdots; 2y_n^T y_0] \). Obviously, Eq.(6) is a simple convex quadratic program (Boyd and Vandenberghe 2004) and can be efficiently addressed by off-the-shelf optimization package, such as MOSEK.

For classification, we adopt the hinge loss as the implementation. According to Lemma 2 and Theorem 3, Eq.(1) can be rewritten as
\[
\min \sum_{i=1}^{n} \alpha_i \|y_i + 1\|_1
\]
which is a simple linear program. The detail derivation is addressed in an efficient manner via MOSEK as well.

Experiments

In this section, we conduct experiments on three weakly supervised learning tasks, i.e., label noise learning, domain adaptation and semi-supervised learning so as to evaluate the effectiveness of our proposed algorithms. We call our proposal as SAFEW (SAFE Weakly supervised learning).

Label Noise Learning Task

We conduct experimental comparison for label noise learning tasks on a number of frequently-used classification datasets\(^1\), i.e., Australian, Breast-Cancer, Diabetes, Digit1, Heart, Ionosphere, USPS and Splice. For each data set, 80% of instances are used for training and the rest ones are used for testing. In the training set, 70% of instances are randomly selected as the noisy or low-quality 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% to 40% with an interval 10%. Experiments are repeated for 30 times, and the average classification accuracy is reported.

Our proposed algorithm is compared with the following methods, including 1 baseline method Sup-SVM that trains a supervised SVM only on the high-quality labeled data; 2 state-of-the-art label noise learning methods: i) Bagging which is regarded as to be robust with label noisy (Fréau and Verleysen 2014); ii) rLR (Robust Logistic Regression) (Boothkrajang and Kabán 2012) that enhances the logistic regression model to handle label noise; 3 traditional classification methods (i.e., SVM, LR (Logistic Regression), \( k \)-NN (\( k \)-Nearest Neighbor)) with regardless of label noise. For LR, the \texttt{glmfit} function in Matlab is used. For \( k \)-NN method, \( k \) is set to 3. For Sup-SVM and SVM method, Libsvm package (Chang and Lin 2011) is adopted and the kernel is set to RBF kernel. For Bagging method, we adopt 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 used as base learners and parameter \( \delta \) is set by 5-fold cross validation from the range \([0.5u, 0.7u]\).

The results are shown in Figure 1 and 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 performs worse than Sup-SVM in many cases, especially when the noise ratio becomes larger, while our proposed SAFEW does not suffer from such deficiency. Moreover, SAFEW achieves best average performance (see detail results in the supplementary material). These demonstrate the effectiveness of SAFEW method.

Domain Adaptation Task

We conduct compared experiments for domain adaptation on two benchmark datasets\(^2\), i.e., 20newsgroups and Landmine. The 20Newsgroups dataset (Lang 1995) contains 19,997 documents and is partitioned into 20 different newsgroups. Following the setup in (Dai et al. 2007; Li, Jin, and Long 2012), 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 sub-categories under the positive and negative classes respectively to form a domain.

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 6 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 (Xue et al. 2008), we randomly select 10% instances in target domain as the labeled data and use 300 most important features as the representation. For Landmine, 5% instances in the target domain are used as the labeled data. Experiments are repeated for 30 times and the average accuracies on the unlabeled instances are reported.

Our method is compared with one baseline supervised method LR that trains a supervised logistic regression model for the labeled data in target domain, one naive domain adaptation method called as Original method that combines the data in source and target domain to train a supervised model, and three state-of-the-art domain adaptation methods, i) Maximum Independence Domain Adaptation (MIDA) method (Yan, Kou, and Zhang 2016); ii) Transfer Component Analysis (TCA) method (Pan et al. 2011); iii)

\(^1\)http://lamda.nju.edu.cn/code_SAFEW.aspx
\(^2\)https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
\(^3\)http://www.cse.ust.hk/TL/
TrAdaBoost method (Dai et al. 2007). MIDA and TCA are two feature-level learning algorithms that learns a domain-invariant subspace between source domain and target domain. TrAdaBoost method is a transfer learner based on AdaBoost. For MIDA and TCA, the kernel type is set to 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 $\delta$ is set by 5-fold cross validation from the range $[0.5u, 0.7u]$.

Results are shown in Tables 2 and 3. Original, MIDA and TCA methods degenerate the performance in many cases. 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.

**Semi-Supervised Learning Task**

For semi-supervised learning, we do experiments on regression tasks with a broad range of datasets\(^4\) 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). All the features and labels are normalized into $[0, 1]$. For each dataset, we randomly select 10 data as the labeled instances. Experiment for each dataset is repeated for 30 times, and the average performance (mean±std) on the unlabeled data is reported.

The compared methods include 1NN method which is a direct supervised nearest neighbor algorithm with only labeled data, Self-$k$NN method which is a semi-supervised extension of the supervised $k$NN method based on self-training (Yarowsky 1995), Self-LS method which is a semi-supervised extension of the supervised least square method (Hastie, Tibshirani, and Friedman 2001), Average method which is a simple ensemble method, Safer method (Li, Zha, and Zhou 2017) which is a method proposed for semi-supervised regression. For Self-$k$NN, we use two distance measures: Euclidean and Cosine, and $k$ is set to 3, the maximum number of iterations is set to 5. For Self-LS method, the parameters related to the importance for the labeled and unlabeled instances are set to 1 and 0.1. For SafeW, Average and Safer methods, Self-$k$NN(Euclidean), Self-$k$NN(Cosine) and Self-LS are adopted as base learners. Parameter $\delta$ is set by 5-fold cross validation from the range $[0.5u, 0.7u]$.

According to the results in Table 4, we can see that SafeW and Safer are always better than the baseline, while the other compared methods will be outperformed by the baseline method in many cases. Moreover, in terms of average performance, SafeW performs better than Safer. The reason owes to a tight set $\mathcal{M}$ learned for base learners. Again, the results validate the effectiveness of SafeW.

**Conclusion**

In this paper, we study to safely exploit weakly supervised data. That is, learning methods with the usage of weakly supervised data could often improve learning performance, meanwhile in the worst case it wont be worse than the base-
Table 2: Classification Accuracy of domain adaptation on 20newsgroup. For the compared methods, if the performance is significantly better/worse than the baseline method, the corresponding entries are then bolded/boxed (paired t-tests at 95% significance level). 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Logistic Regression</th>
<th>Original</th>
<th>MIDA</th>
<th>TCA</th>
<th>TrAdaBoost</th>
<th>SAFE-W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp vs Rec</td>
<td>.9095 ± .022</td>
<td>.9271 ± .0135</td>
<td>.9648 ± .0164</td>
<td>.8077 ± .0155</td>
<td>.9560 ± .0170</td>
<td></td>
</tr>
<tr>
<td>Comp vs Sci</td>
<td>.8225 ± .0662</td>
<td>.7985 ± .0194</td>
<td>.8946 ± .0188</td>
<td>.8255 ± .0172</td>
<td>.8583 ± .0201</td>
<td>.8925 ± .0212</td>
</tr>
<tr>
<td>Comp vs Talk</td>
<td>.8423 ± .0685</td>
<td>.8022 ± .0182</td>
<td>.8231 ± .0164</td>
<td>.8434 ± .0110</td>
<td>.8247 ± .0143</td>
<td>.8451 ± .0158</td>
</tr>
<tr>
<td>Sci vs Talk</td>
<td>.7294 ± .1045</td>
<td>.7100 ± .0121</td>
<td>.7456 ± .0164</td>
<td>.7022 ± .0092</td>
<td>.7166 ± .0213</td>
<td>.7468 ± .0153</td>
</tr>
<tr>
<td>Rec vs Sci</td>
<td>.8006 ± .0758</td>
<td>.7754 ± .0161</td>
<td>.8033 ± .0151</td>
<td>.8440 ± .0118</td>
<td>.8016 ± .0151</td>
<td>.8435 ± .0157</td>
</tr>
<tr>
<td>Rec vs Talk</td>
<td>.8278 ± .0446</td>
<td>.8276 ± .0115</td>
<td>.8566 ± .0105</td>
<td>.8580 ± .0128</td>
<td>.8415 ± .0113</td>
<td>.8579 ± .0105</td>
</tr>
<tr>
<td>Average</td>
<td>.7876</td>
<td>.7805</td>
<td>.8199</td>
<td>.8112</td>
<td>.8084</td>
<td>.8302</td>
</tr>
<tr>
<td>Win/Tie/Loss against LR</td>
<td>1/2/3</td>
<td>4/1/1</td>
<td>3/2/1</td>
<td>3/2/1</td>
<td>5/1/0</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Classification Accuracy of domain adaptation on Landmine data.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Logistic Regression</th>
<th>Original</th>
<th>MIDA</th>
<th>TCA</th>
<th>TrAdaBoost</th>
<th>SAFE-W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain 20</td>
<td>.9215 ± .0173</td>
<td>.9237 ± .0034</td>
<td>.9265 ± .0039</td>
<td>.9255 ± .0045</td>
<td>.9183 ± .0029</td>
<td>.9271 ± .0035</td>
</tr>
<tr>
<td>Domain 21</td>
<td>.9360 ± .0095</td>
<td>.9310 ± .0047</td>
<td>.9384 ± .0045</td>
<td>.9304 ± .0051</td>
<td>.9261 ± .0033</td>
<td>.9396 ± .0038</td>
</tr>
<tr>
<td>Domain 22</td>
<td>.9594 ± .0051</td>
<td>.9555 ± .0038</td>
<td>.9506 ± .0065</td>
<td>.9650 ± .0017</td>
<td>.9095 ± .0026</td>
<td>.9648 ± .0016</td>
</tr>
<tr>
<td>Domain 23</td>
<td>.9361 ± .0095</td>
<td>.9310 ± .0041</td>
<td>.9424 ± .0045</td>
<td>.9314 ± .0051</td>
<td>.9627 ± .0043</td>
<td>.9426 ± .0038</td>
</tr>
<tr>
<td>Domain 24</td>
<td>.9535 ± .0052</td>
<td>.9524 ± .0029</td>
<td>.9447 ± .0025</td>
<td>.9432 ± .0029</td>
<td>.9535 ± .0034</td>
<td>.9550 ± .0024</td>
</tr>
<tr>
<td>Average</td>
<td>.9413</td>
<td>.9387</td>
<td>.9405</td>
<td>.9391</td>
<td>.9340</td>
<td>.9458</td>
</tr>
<tr>
<td>Win/Tie/Loss against LR</td>
<td>0/3/2</td>
<td>2/1/2</td>
<td>1/1/3</td>
<td>1/2/2</td>
<td>3/2/0</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Mean Square Error (mean±std) for the compared methods and SAFE-W on a number of regression data sets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>INN</th>
<th>Self-4NN(Euclidean)</th>
<th>Self-4NN(Cosine)</th>
<th>Self-LS</th>
<th>Average</th>
<th>Safer</th>
<th>SAFE-W</th>
</tr>
</thead>
<tbody>
<tr>
<td>abalone</td>
<td>.020 ± .010</td>
<td>.014 ± .005</td>
<td>.014 ± .003</td>
<td>.013 ± .004</td>
<td>.012 ± .003</td>
<td>.013 ± .005</td>
<td>.013 ± .005</td>
</tr>
<tr>
<td>bodyfat</td>
<td>.019 ± .005</td>
<td>.018 ± .006</td>
<td>.019 ± .005</td>
<td>.041 ± .013</td>
<td>.023 ± .009</td>
<td>.018 ± .007</td>
<td>.017 ± .005</td>
</tr>
<tr>
<td>cadata</td>
<td>.083 ± .029</td>
<td>.063 ± .012</td>
<td>.058 ± .009</td>
<td>.056 ± .007</td>
<td>.057 ± .009</td>
<td>.060 ± .013</td>
<td>.057 ± .005</td>
</tr>
<tr>
<td>cpusmall</td>
<td>.024 ± .012</td>
<td>.027 ± .011</td>
<td>.028 ± .009</td>
<td>.025 ± .010</td>
<td>.024 ± .005</td>
<td>.025 ± .011</td>
<td>.024 ± .009</td>
</tr>
<tr>
<td>housing</td>
<td>.030 ± .010</td>
<td>.036 ± .009</td>
<td>.033 ± .006</td>
<td>.035 ± .009</td>
<td>.043 ± .008</td>
<td>.034 ± .009</td>
<td>.033 ± .005</td>
</tr>
<tr>
<td>mg</td>
<td>.051 ± .009</td>
<td>.039 ± .006</td>
<td>.038 ± .006</td>
<td>.035 ± .015</td>
<td>.038 ± .014</td>
<td>.038 ± .006</td>
<td>.038 ± .006</td>
</tr>
<tr>
<td>mpg</td>
<td>.022 ± .007</td>
<td>.020 ± .006</td>
<td>.018 ± .006</td>
<td>.021 ± .010</td>
<td>.020 ± .006</td>
<td>.019 ± .004</td>
<td>.018 ± .004</td>
</tr>
<tr>
<td>pyrim</td>
<td>.023 ± .006</td>
<td>.021 ± .005</td>
<td>.022 ± .005</td>
<td>.052 ± .014</td>
<td>.020 ± .007</td>
<td>.020 ± .006</td>
<td>.020 ± .006</td>
</tr>
<tr>
<td>Ave. Mse.</td>
<td>.035</td>
<td>.030</td>
<td>.029</td>
<td>.035</td>
<td>.029</td>
<td>.030</td>
<td>.028</td>
</tr>
<tr>
<td>Win/Tie/Loss against INN</td>
<td>4/3/1</td>
<td>3/4/1</td>
<td>3/3/2</td>
<td>5/2/1</td>
<td>6/2/0</td>
<td>6/2/0</td>
<td></td>
</tr>
</tbody>
</table>

There are many interesting future works. For example, our method is a two-stage method and may lose some information, whereas directly one stage method that takes the generation of base learners into account, would be worth studying. Moreover, the study of other weakly supervised setting such as new class detection (Mu, Ming, and Zhou 2017), is an interesting issue in future.

References


Rosenstein, M. T.; Marx, Z.; and Kaelbling, L. P. 2005. To transfer or not to transfer. In a NIPS-05 Workshop on Inductive Transfer: 10 Years Later.


