International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-023-00414-8

REGULAR PAPER

Boosting for regression transfer via importance sampling

Shrey Gupta[1] · **Jianzhao Bi**[2] · **Yang Liu**[3] · **Avani Wildani**[1]

Received: 27 April 2022 / Accepted: 15 June 2023

Abstract
Current instance transfer learning (ITL) methodologies use domain adaptation and sub-space transformation to achieve successful transfer learning. However, these methodologies, in their processes, sometimes overfit on the target dataset or suffer from negative transfer if the test dataset has a high variance. Boosting methodologies have been shown to reduce the risk of overfitting by iteratively re-weighing instances with high-residual. However, this balance is usually achieved with parameter optimization, as well as reducing the skewness in weights produced due to the size of the source dataset. While the former can be achieved, the latter is more challenging and can lead to negative transfer. We introduce a simpler and more robust fix to this problem by building upon the popular boosting ITL regression methodology, two-stage TrAdaBoost.R2. Our methodology, S- TrAdaBoost.R2, is a boosting-based ensemble methodology that utilizes importance sampling to reduce the skewness due to the source dataset. We show that S- TrAdaBoost.R2 performs better than competitive transfer learning methodologies 63% of the time. It also displays consistency in its performance over diverse datasets with varying complexities, as opposed to the sporadic results observed for other transfer learning methodologies.

Keywords Instance transfer learning · Negative transfer · Domain adaptation

1 Introduction

While semi-supervised learning and unsupervised learning methodologies work well for partially labeled or unlabelled datasets [5, 47], they fall short for instances where the sample size is small [25, 48, 65–67]. Instance transfer learning (ITL) [19, 21, 28, 46, 48, 66, 67], a sub-class of data-based transfer learning approaches [73], is designed for limited and labeled samples, shared feature space, and independent and identically distributed (i.i.d) data-distributions [49, 62], making it ideal for real-world datasets [12, 14, 35, 39, 43, 51]. It stands apart from its counterparts, such as feature-transfer learning and parameter-transfer learning, as it allows data adjustment and transformation of domain instances, making it ideal for dissimilarly distributed source and target domains. Moreover, ITL methodologies are as statistically interpretable [13] as they are powerful [6, 66], which increases their usability for domain experts [64] who avoid complex, black-box methodologies [4, 29, 34]. Therefore, these methodologies have the advantage of being less complex but equally reliable when compared to deep transfer methodologies. Another reason for leaning toward ITL methodologies is because it is easier to transfer the source domain by applying adaptation methodologies [32, 57] as well as using techniques involving reduction of distribution difference between the source and the target domain [15, 27, 57]. The accuracy of prediction does not just depend on the transfer learning methodology but also involves the nature of the distribution. Real-world datasets suffer from collecting data that is complete, high-resolution, and evenly sampled. This is due to the dependence on the cost of equipment which can result in hardware limitations. This leads to the resulting dataset varying in resolution as well as the quality [39]. Hence, a robust transfer learning methodology

✉ Avani Wildani
avani@mathcs.emory.edu

Shrey Gupta
shrey.gupta@emory.edu

Jianzhao Bi
jbi6@uw.edu

Yang Liu
yang.liu@emory.edu

[1] Department of Computer Science, Emory University, Atlanta, GA, USA

[2] Department of Environmental and Occupational Health Sciences, University of Washington, Seattle, WA, USA

[3] Gangarosa Department of Environmental Health, Emory University, Atlanta, GA, USA

Published online: 21 July 2023

should perform consistently well for data distributions with varying complexities.

Among the ITL methodologies, we employ ensemble methodology, especially the boosting methodology [13] as it aggregates the results from multiple learners. Similarly, the transfer boosting methodology TrAdaBoost.R2 [50] is regularized and uses domain adaptation for iteratively re-weighing the source instances with respect to the target dataset for knowledge transfer [61]. The underlying architecture is AdaBoost [26], which focuses on misclassified training instances, leading to contextual learning. However, boosting methodologies suffer from negative transfer [53] when the source dataset size is large compared to the target dataset, leading to a skewed final model. To address the problem of negative transfer, we introduce S- TrAdaBoost.R2, a successor to two-stage TrAdaBoost.R2 (TTR2) that uses importance sampling [36, 48, 72] to improve the alignment of source instances with the target values, and applies a balanced weight update strategy to mitigate the skewness generated due to the large sample size of source datasets. We test S- TrAdaBoost.R2 across a range of standard regression datasets with limited target instances and varying complexities, and find that it outperforms other ITL methodologies 63% of the times and the baseline TTR2 more than 75% of the times. Notably, it has consistent performance (RMSE and R-squared score) for both the regular comparative study and the Ablation study (Fig. 2 and Table 2), as opposed to fluctuating results observed for other methodologies.

The primary contributions of this paper are given as follows:

1. We introduce S- TrAdaBoost.R2, complexity-tolerant, domain-agnostic boosting-based transfer learning algorithm that uses importance sampling and a balanced weight update strategy to outperform its predecessor TTR2 and other competitive ITL methodologies.
2. We discuss the complexity measures, i.e., metrics to quantify the complexity of distribution. They categorize the distribution based on correlation, linearity, and smoothness, to provide a numerical estimate of its simplicity.
3. We demonstrate that S- TrAdaBoost.R2 outperforms competitive ITL methodologies when measured in terms of accuracy and loss, for high-complexity datasets. We also provide the ablation analysis for Importance Sampling, which demonstrates the modularity and commutability of the technique.

2 Background

Previous work on transfer learning [19, 67] provides methodologies for measuring the shared information content between multiple domains in transfer learning [42, 44, 59]. These models attempt to find common structural representations of source instances to gauge the quantity as well as the quality of the transfer. However, for highly dissimilar source and target domain instances, a reduction of prediction accuracy for transfer learning algorithms when compared to non-transfer learning algorithms i.e., *negative transfer* is commonplace [53]. Figure 1 shows negative transfer when TTR2 and AdaBoost.R2 are fitted over the concrete dataset from UCI machine learning repository [3]. We observe a decline in TTR2's performance as the target sample size increases. This shows a trade-off in the performance of transfer learning algorithms to the sample size of the target distribution. Hence, transfer learning algorithms perform better when the sample size of a target dataset is small.

The concept of translating knowledge and model across domains has been much researched upon and hence, transfer learning, similar to machine learning, is observed for both classical transfer learning [10, 15, 27, 50] and deep transfer learning methodologies [4, 29, 34, 40, 60, 69, 71, 74]. While deep networks can often improve transfer accuracy, they sacrifice model interpretability, generalizability, adaptability, and flexibility for more diverse tasks [9, 52]. Whereas, ITL algorithms have more transparency compared to deep transfer models as they do not suffer from obscurity in showing intermediary steps and learned concepts. Even for unrelated source and target domains, the source instances adapt to the target instances by either re-weighting [10, 27] or transforming to the target space [15], indicative of the adapt-

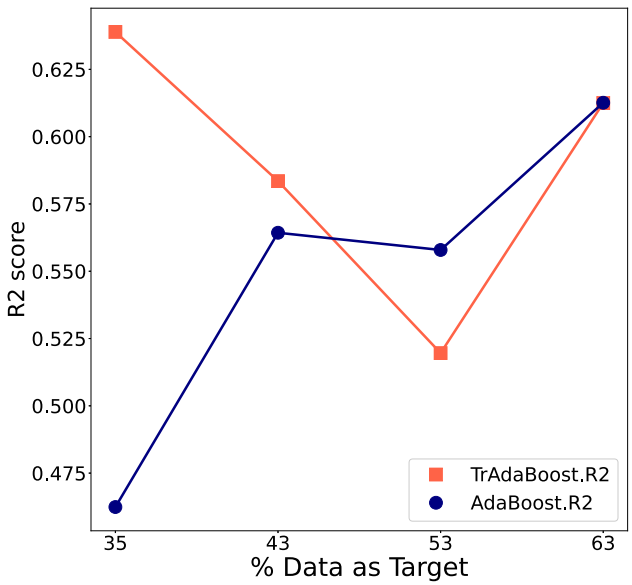

Fig. 1 Negative transfer in TTR2 is induced as a result of increasing the target sample size from 35 to 63% of the total training data. The baseline algorithm is AdaBoost.R2. For a larger target sample size, the baseline performs better than TTR2

ability of ITL methodologies. The current ITL methodologies can be vaguely divided into two types based on how they apply the weighing strategy to the source domain instances. The first one involves re-weighing all the source instances at once using techniques such as Kernel Mean Matching (KMM) [15, 32], Weighted-Kernel Ridge Regression [27], Kullback–Leibler Importance Estimation [57], translating training instances to an Invariant Hilbert Space [30], or learning source domain instance weights based on the conditional distribution difference from the target domain [11]. The second type of methodology is the ensemble learning methodology, primarily including boosting techniques.

2.1 Boosting

Boosting [26] is an ensemble technique that builds a classifier by using a set of weak learners, whereby the weights of the training samples are updated over a chosen number of iterations, and finally these weak learners are combined to generate a strong learner. Popular boosting methodologies such as AdaBoost.R2 [20] typically assume that the test and training datasets have a similar distribution and hence do not require domain adaptation. They do not suffer from overfitting [58] and have a robust prediction over diverse datasets.

2.1.1 Boosting for transfer learning

TrAdaBoost [16] is a classification boosting framework that applies transfer learning to compensate for a lack of training instances for the target dataset. The source and target data instances are merged to form the training data for the TrAdaBoost, and in each iteration, the weights of the instances are adjusted such that the misclassified target instances have their weights increased, whereas the misclassified source instances have their weights reduced, in order to reduce their impact toward the model learning. However, this may lead to model over-fitting, and reduction in the variance of the training model, therefore negatively affecting the model generalizability [63].

2.1.2 Boosting for regression transfer

TrAdaBoost.R2 [50] builds upon TrAdaBoost [20] for regression problems, using adjusted error over residuals and reweighing of the instances. The improved version, called two-stage TrAdaBoost.R2 (TTR2), is divided into two stages. The first stage involves gradually reducing the weights of the source instances until a certain cross-validation threshold is achieved. In the second stage, weights of the source instances are frozen while the weights of the target instances are updated as in AdaBoost.R2. The bi-update methodology for TTR2 helps reduce the skewness produced due to source instances. This mostly happens in the cases when source sample size is very large compared to the target sample size, which consequently makes the model learning biased toward the source domain.

2.2 Variants of regression transfer

Pardoe et al. [50] introduced two categories of transfer learning algorithms. The first category contains algorithms that choose the best hypothesis from a set of experts, each representing the models for the corresponding source dataset. This category includes algorithms such as *ExpBoost.R2* and *Transfer Stacking*. Algorithms in the second category, which include TrAdaBoost.R2 and TTR2, use the grouped source and target datasets to perform boosting. Since boosting methodologies involve instance reweighing, they fall under the category of transfer learning algorithms that use domain adaptation. This is especially useful and applicable for real-world datasets with dissimilar domain distributions. Hence, such domain adaptation transfer methodologies help in reducing the burden of maintaining expert systems [53]. Apart from the boosting methodologies, the varying domain adaptation approaches include using a kernel-employing Gaussian process [10] for source instance modification or kernel ridge regression, and discrepancy minimization for domain adaptation [15]. Similar to importance sampling [48], several studies [27, 45] have used importance weighting of source instances to improve inference for transferring knowledge. Transfer methodologies using approaches similar to active learning, such as [18] (employing modeling structure with second-order Markov chains), as well as the burgeoning variety of deep learning approaches [5, 17], are indicative of the usefulness of active learning in the form of importance sampling as a viable technique to be picked up by ITL methodologies.

2.3 Importance sampling

Importance sampling is a methodology based on the concept that certain instances of the source dataset are more similarly distributed to the instances in the target dataset and thus should be sampled for learning optimal transfer models. The core tenet of importance sampling is that models should be trained with some cognizance of a multi-domain transfer, in order to avoid stale training data [36, 48, 68]. Zhao et al. [72] introduce stochastic optimization for importance sampling of non-transfer learning problems, to reduce variance and improve convergence. Elvira et al. [22, 23] utilize gradient-based learning whereas Bullago et al. [8] and Schuster et al. [55] apply Monte Carlo methods to apply adaptive importance sampling. Salaken et al. [54] present a seeded sampling technique for transfer learning that we extend to form the variance sampling component used by our algorithm, S- TrAdaBoost.R2. Their work introduces

an algorithm to cluster the source domain instances which are then translated to limited target domain instances for knowledge/domain adaptation. In the following section, we describe how we utilize the concept used by seeded sampling for cherry picking instances from the source domain for the purpose of introducing variance in the target dataset.

3 Methodology

Problem definition Given source and target datasets, such that their instances are denoted by x^T and x^S respectively. Hence, the target dataset is denoted as $X^T = \{x_1^T, x_2^T, \ldots, x_m^T\}$ for m instances and source dataset is denoted as $X^S = \{x_1^S, x_2^S, \ldots, x_n^S\}$ for n instances. Similarly, the target output dataset is denoted as $Y^T = \{y_1^T, y_2^T, \ldots, y_m^T\}$ and the source output dataset is denoted as $Y^S = \{y_1^S, y_2^S, \ldots, y_n^S\}$. The target domain suffers from significant data deficiency and dissimilarity of distribution compared to the source domain. Our goal is to find a transfer learning approach that can use the source domain instances as leverage for building the prediction model as well as avoiding negative transfer. The transfer learning algorithm should perform consistently well on varying domain distributions with differing complexities.

Approach S- TrAdaBoost.R2 is a transfer regression boosting algorithm which builds a model, $h_f : X \rightarrow Y$, such that h_f is the final learned hypothesis of the ensemble of hypotheses over the learning iterations, using the training data which is a combination of source and target datasets that share a similar feature space but have dissimilar distributions. Hence by this definition, the combined training dataset (source + target) can be denoted as $\{(x, y) \| x \in X^T \cup X^S, y \in Y^T \cup Y^S$ and $X^T, X^S, Y^T, Y^S \in R^d\}$ where d represents the feature space of the source and target domain.

Algorithm 1: k-Center Sampling

Input: X^T, Y^T, X^S, Y^S
Output: Labeled dataset X^{VT} (size k).
1 Find $X^C \subset X^S$ such that $X^C = \{x_1^C, x_2^C, ..., x_k^C\}$ has k samples, obtained using k-means clustering on X^S.
2 Initialize $X^E = \phi$ (Empty-set)
3 **for** $x^C \in X^C$ **do**
4 | Find x^T such that $\forall x^T \in X^T \ \min(\|X^C - x^T\|)$
| $X^E \cup \{x^T\}$
5 **end for**
6 Repeat steps 3 to 5 and obtain set $X^{VT} \subset X^S$ closest to instances in set X^E.
7 **return** X^{VT}

3.1 S-TrAdaBoost.R2

To improve the performance of TTR2, we present S-TrAdaBoost.R2 as shown in Algorithm 2. There are two main areas where S- TrAdaBoost.R2 diverges from its predecessor, TTR2; the first is applying importance sampling, and the second is the weight update strategy for S- TrAdaBoost.R2, which differs from the TTR2. In the following subsections, we elaborate upon these differences as well as determine the time complexity of S-TrAdaBoost.R2.

3.1.1 Sampling

In order to improve the prediction accuracy, S- TrAdaBoost.R2 initially samples the source dataset, X^S, to obtain optimal representative instances, i.e. similar instances to the target dataset, X^T. Hence, before merging the source domain and target domain samples, we apply importance sampling to carefully select favorable source domain instances. We utilize a greedy approach for calculating the distance between the source and the target instances. Such an importance sampling can be achieved by utilizing distance measures (Euclidean, Manhattan, and more) as well as alternative methodologies utilizing gradient-based and similarity-based sample selection [8, 22, 23, 55]. For our experiments, we use the Euclidean distance (L2 norm). Hence, we find the set $X^{ES} \subset X^S$ such that,

$$X^{ES} = \|\mathbf{x_i^S} - \bar{\mathbf{x}}^{\mathbf{T}}\| \quad \forall x_i \in X^S$$

where $\bar{x}^T$ is the mean of target instances, $\|\cdot\|$ is the Euclidean distance, and $|X^{ES}| = |X^S|$, i.e. they share the same cardinality. We select the top p instances from X^{ES} for the source dataset, which reduces the source dataset size to $X^K = \{x_1^K, x_2^K, \ldots, x_p^K\}$ such that $p \ll n$ and discard the remaining $(n - p)$ instances since they failed the similarity testing threshold.

Furthermore, to improve the generalizability of the prediction model, we also induce variance in the target dataset whereby source instances most similar to the target instances are added using the k-center sampling, an approach presented in Algorithm 1. Including the most similarly distributed source samples in the target dataset improves the fit for the regressor since S- TrAdaBoost.R2 focuses more on target instances than the source instances. These similarly distributed source samples act as noise for the target distribution and thereby improve the generalization error. Even though TTR2 tries to mitigate this using its two-stage source instance penalizing process, we found that reducing the source sample size using importance sampling, as well as performing variance sampling, allows S- TrAdaBoost.R2 to perform better compared to its predecessor.

k-center sampling k-center sampling is an unsupervised approach that returns k centroids, where k is equal to the number of source instances in the set, X^S (Algorithm 1). We employ k-center sampling in our methodology to introduce noise in the target dataset, in order to increase its variability. After the selection of centroids, the target instances closest to these centroids are selected as the representative target set, X^C. The source instances most similar to the representative target set are chosen as the final subset, X^{VT}, for inclusion into the target dataset. The k-center sampling methodology is presented in Algorithm 1. The final size of the target dataset is given as follows: $q = n + k$. For the k-center sampling, the time complexity is $O(N^2)$ as a result of using the k-means clustering for calculating the closeness. Hence, the sampling pipeline produces a new source dataset (due to Importance Sampling) and a new target dataset (due to Variance Sampling) as X^{ES} and X^{VT} respectively.

Algorithm 2: S- TrAdaBoost.R2

Input: The labeled data sets, X^S (size n) and X^T (size m)
The number of estimators, N
The number of cross-validation folds, F
Number of Steps/Iterations, S
The base learning algorithm, $learner$
Learning rate, α

Output: Final hypothesis, h_f

Importance Sampling
Get X^{ES} (updated source dataset) containing p instances (from X^S) most similar to X^T.

Variance Sampling
Get X^{VT} (updated target dataset) containing q instances, obtained using k-Center Sampling on set X^T.

Initialize
Initial weight $w^1 = 1/(p+q)$

for $t \leftarrow 1$ *to* S **do**
  Call AdaBoost.R2 with N estimators and $learner$ to obtain hypothesis h_t.
  Calculate the adjusted error using the hypothesis h_t over F folds as,

$$e_i = |y(x_i) - h(x_i)|/J$$

  where $J = \max_{i=1}^{(p+q)} |e_i|$
  Set $\bar{\beta}_t = \eta_t/1 - \eta_t$ where $\eta_t = \sum_{i=1}^{p+q} w_i^t e_i^t$ and $\beta_t = \frac{q}{(p+q)} + \frac{t}{(S-1)}(1 - \frac{q}{(p+q)})$.
  Update the weights as:

$$w_i^{t+1} = \begin{cases} \frac{w_i^t \bar{\beta}_t^{e_i^t} \alpha}{Z_t}, & 1 \le i \le p \\ \frac{w_i^t \beta_t^{1-e_i^t} \alpha}{Z_t}, & p \le i \le (p+q) \end{cases}$$

  where Z_t is sum of sample weights
end for
return h_f *where* $f = argmin_i error_i$

3.1.2 Weight update strategy

We present S- TrAdaBoost.R2 in Algorithm 2, where we hypothesize that by updating the target weights more aggressively, the prediction model is able to mitigate the source distribution bias. This is especially useful for dissimilar source and target domain distributions, as well as when $|X^S| \gg |X^T|$. We also note that S- TrAdaBoost.R2 does not employ AdaBoost.R2' [50], a modified version of AdaBoost.R2 where the weights of source instances are frozen and the weights of target instances are updated based on the reweighing approach used by AdaBoost.R2. However, applying highly focused domain adaptation by freezing weights of source instances can greatly reduce the generalizability of the model, as performed in the previous technique, TTR2. For this reason, our approach penalizes both the source domain and target domain instances allowing for a balanced weighing. Hence, in S- TrAdaBoost.R2, the hypothesis is obtained by using the AdaBoost.R2 methodology initially. The weights for the instances are then updated iteratively using the following weight equation,

$$w_i^{t+1} = \begin{cases} \frac{w_i^t \bar{\beta}_t^{e_i^t} \alpha}{Z_t}, & 1 \le i \le p \\ \frac{w_i^t \beta_t^{1-e_i^t} \alpha}{Z_t}, & p \le i \le (p+q) \end{cases}$$

In the above equation, $\bar{\beta}_t = \eta_t/1 - \eta_t$ such that $\eta_t = \sum_{k=1}^{(p+q)} w_i^t e_i^t$, and $Z_t = \sum_{k=1}^{(p+q)} w_i \beta_t$ indicates the sum of sample weights. For the above weighing strategy, the source domain instances are penalized more aggressively with both β and e_i depending on instance residual compared to the target domain instances with constant β. This allows for a balanced weighing where both domain instances are penalized with the target instance weighing being slower compared to the source instance weighing to balance the skewness caused by a large number of source instances. Hence, although the source instances are penalized more than target instances, the instance weighing is still not as aggressive as in the predecessor methodology, TTR2 which can lead to overfitting on the dataset.

3.1.3 Time complexity for S-TrAdaBoost.R2

The time complexity of the S- TrAdaBoost.R2 can be divided into *four* parts:

1. Time complexity of importance sampling (O_1)
2. Time complexity of the weak hypothesis (O_2)
3. Time complexity of computing the error rate in S- TrAdaBoost.R2 (O_3)
4. Time complexity of the second stage of S- TrAdaBoost.R2 (O_4).

For S iterations, time complexity can be defined as $O(S*(O_2+O_3+O_4))$. For our experiments, we chose a decision tree as the base learner. The time complexity for creating a decision tree is $O(d*N^2*logN)$ (O_2), where d is the dimension of the dataset, N is the number of samples, and each decision is taken in $O(logN)$ time. The time complexity of computing adjusted error combined with the weight update process (O_3), does not increase more than $O(N^2)$. Finally, the time complexity of computing the second stage of the S-TrAdaBoost.R2 is similar to producing a weak hypothesis (O_4). Hence, the time complexity over S iterations is given as follows:

$$\begin{aligned} &O(S*(d*N^2*logN+N+d*N^2*logN)) \\ &\quad = O(2*S*d*N^2*logN+S*N)) \\ &\quad = O(S*d*N^2*logN)) \end{aligned}$$

For the k-center sampling, the time complexity is $O(N^2)$ for calculating closeness using the k-means clustering, as well as using Manhattan distance for finding the most similar source instances. Hence, the total time complexity for S-TrAdaBoost.R2 can be calculated as follows:

$$O(S*d*N^2*logN+N^2) = O(S*d*N^2*logN)$$

3.2 Complexity of distribution

Domain-agnostic characterizations of dataset complexity are surprisingly uncommon. Fernandez et al. [24] present a characterization based on Shannon entropy, but this does not extend to the continuous, often real-valued domains of many real-world datasets [7]. Other intuitive measures such as sorting datasets by the number of features or self-similarity do not reliably capture types of datasets that we observed as being especially prone to negative transfer. The heterogeneity and complexity of datasets usually determine the model performance. While the heterogeneity of real-world datasets can be outlined as a factor of their multi-source and spatiotemporal character, this might not be true for their complexity. Ho et al. [31] proposed metrics to measure complexity for classification datasets. Maciel et al. [41] extended that work for regression datasets which stemmed from the work done by Lorena et al. [38] that utilizes meta-features as a measure of complexity. In the following sections, we discuss and apply the measures provided by Maciel et al. [41] to characterize the complexity of regression datasets.

3.2.1 Collective feature efficiency (C_{FE}): correlation measure

The correlation measure determines the highly correlated predictor to the target variable and fits a linear regressor to find its residuals. All the instances having residual less than a certain threshold ($\epsilon \leq 0.1$) are discarded and the remaining instances are used to determine the next highly correlated predictor. The process is repeated until the complete feature space has been visited. Maciel et al. [41] describes the measure as the Collective Feature Efficiency (C_{FE}) which is expressed as follows:

$$C_{FE} = 1 - \sum_{k} \frac{N_k}{N}$$

where N_k is the number of instances that are removed (using the set threshold), N is the total number of instances and k is the feature. Higher values for C_{FE} indicate more complex problems.

3.2.2 Distance from linear function (D_L): linearity measure

The linearity measure sums the absolute values of residuals when a multiple linear regressor is used as the learner [41]. It is expressed as a distance measure (D_L) and is quantified as follows:

$$D_L = 1 - \sum_{i=1}^{N} \frac{R_i}{N}$$

where R_i are the residues and N is the sample size. Lower values indicate a simpler distribution.

3.2.3 Input distribution (D_I): smoothness measure

The smoothness measure determines the smoothness of the distribution by ordering the predictor values in ascending order with regard to the output variable. It then finds the distance (L2 Norm) between each pair of instances [41]. Lower values mean a simpler distribution, indicating that the instances in input space are closer to each other, leading to a smooth distribution. It is expressed as follows:

$$D_I = \frac{1}{N} \sum_{i=2}^{N} \|\mathbf{x_i} - \mathbf{x_{i-1}}\|$$

where N is the sample size and $\|.\|$ is the Euclidean distance.

4 Evaluation

For our experiments, we evaluate S- TrAdaBoost.R2 against other competitive transfer learning methodologies such as TTR2 [50], KMM.TL [32], KLIEP.TL [57] and IW-KRR.TL [27] known to perform well for regression-based instance transfer learning problems. Since TTR2 is the predecessor for S- TrAdaBoost.R2, we define it as the baseline

Table 1 Dataset statistics [Tr: Training, Tt: Test, P_C^M: predictor] and complexity (Sect. 3.2)

	Concrete	Housing	Auto	Ailerons	Elevators	Abalone	Kinematics	C.Activity
Shape	(1030, 9)	(506, 14)	(392, 8)	Tr: (7154, 41) Tt: (6596, 41)	Tr: (8572, 19) Tt: (7847, 19)	(4177, 9)	(8192, 9)	(8192, 22)
Target	Strength	medv	mpg	Goal	Goal	Rings	y	usr
P_C^M	Cement	nox	h.power	*None*	*None*	weight	theta7	pgin
C_{FE}	0.66	0.39	0.51	0.47	0.59	0.69	**0.70**	0.36
D_L	0.20	0.29	0.24	0.26	0.32	0.27	0.19	**0.36**
D_I	0.71	0.90	0.58	0.68	0.59	0.51	**1.08**	0.58

Bold represents datasets with the highest complexity

algorithm for comparison. The decision tree regressor was chosen as the base learner for these methodologies. For TTR2 and S- TrAdaBoost.R2, the following values were considered: S (no. of steps) = 30, F (CV-folds) = 10, learning rate = 0.1 and a *squared loss*. Similar values were used by Pardoe et al. [50] for their study on regression boosting. For the remaining algorithms, we used the default values for the parameters. The values were chosen to maintain generalizability of the predictions across the algorithm. They were derived using multiple experiments and iterations involving parameter tuning, and were judged to not be biased toward a single model to the best of our knowledge. The results along with the ablation study are presented in the following sections.

Datasets We chose 8 standard regression datasets from the UCI machine learning repository [3] as shown in Table 1. UCI datasets were divided into *source*, *target*, and *test* sets using the splitting methodology used by Pardoe et al. [50]. The splits were made by identifying the feature moderately correlated with the target variable, which allowed for concepts to be significantly different from each other. The *first* split was considered as the target dataset and the remaining splits as the source dataset. This was done so that the source sample size would be higher than the target sample size. The target dataset was further split into training and testing datasets using a k-fold split over 20 iterations. Our initial study showed that the root mean squared loss (RMSE) on concrete, housing, and automobile datasets were moderately varied for such a division which allowed for robust predictions since it incorporated both generalizability for the models, as well as lesser noise. Hence, we further extended the splitting methodology to other datasets – abalone, kinematics, and computer activity. For ailerons and elevators datasets, the UCI repository already consisted of a testing dataset. We took very few target instances so that the remaining larger dataset could be used as the source dataset, which in turn imitates a real-world transfer learning problem. Table 1 shows the dataset statistics including their size, target variable, and predictor used for correlation splitting. Although Concrete, Housing, and Automobile are small sample datasets, they were used to imitate the study by Pardoe et al. [50]. We compensated for this imbalance using other large sample datasets with varying heterogeneity. The complexity evaluation in Table 1 shows the complexity of dataset distributions based on variance (C_{FE}), smoothness (D_I), and linearity (D_L). For each measure, a higher value indicates a more complex distribution. We observe that *Kinematics* has the highest complexity (2 out of 3 times) when compared to the other datasets.

Ablation study We perform an ablation study where the importance sampling technique is applied individually to each transfer learning methodology. The goal of this study is to induce fairness in comparison, given the modular nature of importance sampling. Sampling is a two-phase methodology that includes variance sampling and importance sampling. The variance Sampling includes sprinkling the target dataset with source instances in order to introduce noise and increase the variance of the distribution. For the concrete, housing, and automobile datasets, variance sampling was not applied due to the low sample size. The importance sampling on the other hand uses similarity measuring to find the source instances most similar (important) to the target instances. The ablation study exploits importance sampling for all the methodologies and variance sampling for larger datasets.

4.1 Results

We implemented the experiments on an HPC cluster with 16 processors and 128 GB RAM. Any required short supplemental processing was performed on personal laptops with half the number of processors and RAM. The number of cross-validation folds was 20 for the datasets. The distribution of prediction values is shown in the box-plot Fig. 2. We observe that S- TrAdaBoost.R2consistently performs well, with low RMSE as well as a high R-squared score. However, this is not true for other methodologies, especially IW- KRR.TL and TTR2 which, although they sometimes outperform S- TrAdaBoost.R2, also fluctuate highly in their performance. Example IW- KRR.TL is the most optimal model for automobile, abalone, and kinematics datasets as observed through its mean RMSE and R-squared values.

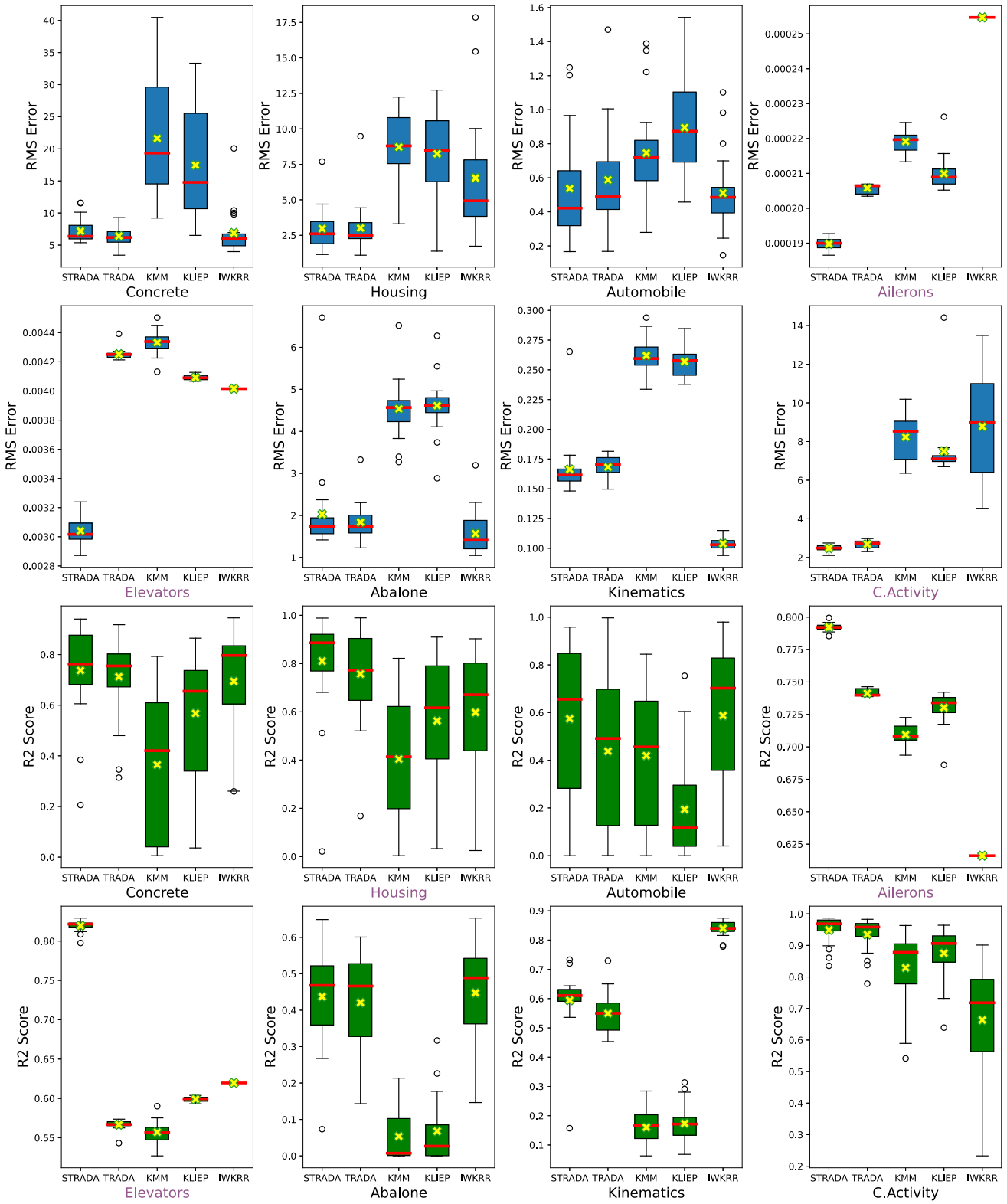

Fig. 2 Comparison of transfer learning algorithms—TRADA: TTR2, STRADA: S- TRADABOOST.R2, KMM: KMM.TL, and KLIEP: KLIEP.TL, IWKRR: IW- KRR.TL, where the RMS error and R-squared score is calculated over 20 iterations. The Interquartile Range (IQR), mean value (marker: yellow "X"), and median value (marker: red line) for each algorithm over the iterations have been highlighted. The datasets for which S- TRADABOOST.R2 performs particularly well are marked as well (marker: purple) (color figure online)

Table 2 Ablation study

	Ailerons		Elevators		Abalone		Kinematics		C.Activity	
	RMSE	R^2	RMS	R^2	RMS	R^2	RMS	R^2	RMS	R^2
TRADA	0.00023	0.65	0.0042	0.38	2.14	0.40	0.18	0.47	2.98	0.92
STRADA	**0.00018**	**0.79**	0.0030	**0.81**	2.02	**0.43**	0.18	0.51	**2.48**	**0.94**
KMM	0.00029	0.46	0.0049	0.31	2.73	0.06	0.27	0.08	11.30	0.17
KLIEP	0.00026	0.58	0.0043	0.42	2.76	0.10	0.26	0.10	11.09	0.22
IWKRR	0.00025	0.63	**0.0021**	**0.81**	**1.99**	0.41	**0.10**	**0.84**	8.77	0.66

Bold represents techniques with the lowest RMSE and highest R^2 score (i.e. best prediction performance)

But it is not consistent in its performance as observed for computer activity, ailerons, and elevators datasets, where it fluctuates highly in its mean and variance over the iterations. However, S- TrAdaBoost.R2 performs consistently well for all of the datasets and comes a close second in the kinematics dataset, where IW- KRR.TL outperforms the competing methodologies by a high margin. Similarly, for TTR2, we observe that it performs well (RMSE score) on concrete and abalone datasets compared to S- TrAdaBoost.R2, but its performance is not consistent as observed for ailerons and elevators datasets. We consider TTR2 to be our baseline algorithm for this study primarily because it is the predecessor of S- TrAdaBoost.R2 and observe that S- TrAdaBoost.R2 outperforms TTR2 75% of the times in the case of loss measure, and 100% when measured for correlation accuracy.

Considering that the importance sampling is a pre-domain adaptation methodology and should not be limited to just S- TrAdaBoost.R2, we conduct an Ablation study as shown in Table 2. We observe minimal improvement in the performance of TTR2 and IW- KRR.TL and find that S- TrAdaBoost.R2 performs consistently well (4 out of 5 times). Table 2 shows that IW- KRR.TL has competitive scores with regard to S- TrAdaBoost.R2; however, it has the same inconsistent performance as observed in the comparative study presented in Fig. 2. Also, TTR2 does not show any improvement except for a similar RMSE score to S- TrAdaBoost.R2 for the kinematics dataset. However, IW- KRR.TL easily outperforms all other methodologies for the kinematics dataset. It should also be noted that in both studies, the remaining algorithms KMM.TL and KLIEP.TL performed quite poorly compared to the other methodologies and showed no apparent sign of improvement in either case. Hence, we can say that S- TrAdaBoost.R2 has shown itself to be consistent among all the measures, adapting more robustly to more complex and varying distribution datasets.

5 Discussion

Since S- TrAdaBoost.R2 is a successor to TTR2, we use TTR2 as the baseline methodology and observe that S- TrAdaBoost.R2 outperforms it 7 out of 8 times during the comparative study. We also note that TTR2 shows no significant improvement during the ablation study. This justifies the steady performance of S- TrAdaBoost.R2, where it consistently has optimal RMSE and R-squared scores during the comparative and ablation studies. The ablation study is used to justify how importance sampling is useful when combined with the learning methodology for S- TrAdaBoost.R2. This is due to the balanced weighing complimenting the source domain sampling methodology. We find that for relatively complex datasets such as concrete, elevators, kinematics, and c.activity (complexity analyzed in Table 1), S- TrAdaBoost.R2 performs well on most of them (3 out of 4 times), falling short only in the case of kinematics dataset when compared to IW- KRR.TL methodology.

It should be noted that both the training error and the generalization error of a similar problem space have been analyzed thoroughly in Freund et al. [26], and this analysis is further known to apply to TrAdaBoost.R2 [50], a predecessor to S- TrAdaBoost.R2. The objective function for transfer learning involves minimizing the loss, $min_h\{\mathcal{L}(h) + \lambda\eta\}$, where η is the regularization function, and λ is the regularization constant for the loss function $\mathcal{L}$. We hypothesize a function $h \in H$ that maps training instances, predictor $x \in X$ to target $y \in Y$ in the target domain T_T. Hence, the instance transfer methodology tries to minimize the weighted loss of target and source domain [63] ($\mathcal{L}(h) = \mathcal{L}_T(h) + \mathcal{L}_S(h)$). Since S- TrAdaBoost.R2 relies on using AdaBoost.R2 unlike TTR2 [50], it has increased generalizability as it avoids overfitting while assigning balanced source and target weights.

While S- TrAdaBoost.R2 has improved generalizability by utilizing balanced reweighing and sampling methodologies, it can however be limited by the computational overhead and poorly strategized implementation of the sampling methodologies. The importance sampling methodology can reduce the performance of transfer learning if the threshold for sampling is high, i.e. very few source domain instances are selected. Furthermore, for large source domain datasets ($> 10^5$), sampling methodologies (importance sampling and variance sampling) cause additional computational overhead. Hence, while these methodologies are simpler to

implement, the initial and sampled instances affect the performance of our approach.

6 Conclusion

We introduce S- TrAdaBoost.R2, which uses importance sampling combined with an unrestricted weight update strategy to improve performance for the domain of instance transfer learning by an average of 12% across all datasets, and 13% in sufficiently complex datasets when compared to its predecessor TTR2. To better characterize the datasets that S- TrAdaBoost.R2 performs well on, we utilize complexity measures [41], C_{FE}, D_L and D_I that employ feature correlation and fitting a linear regressor to compute the complexity for the datasets. Hence, we can conclude that S- TrAdaBoost.R2 would be well suited for complex real-world datasets that range in distributions, as well as uniformity of features. While the functional improvement is large, the additional overhead and physical changes we propose to TTR2 are modest enough that we expect S- TrAdaBoost.R2 as a replacement for TTR2 and other instance transfer methodologies in scientific data analysis pipelines.

7 Future work

In the future, we want to expand our methodology for not only instance transfer learning methodologies but also feature-transfer learning [1, 2, 33] as well as parameter-transfer learning methodologies [37, 56]. Although boosting transfer methodologies are simpler to understand than their deep learning counterparts, the user may suffer a trade-off in prediction accuracy for simplicity, which is not always preferred. We also plan to compare boosting-based instance-transfer learning methodologies to deep transfer learning methodologies [70]. We plan to explore a methodology that uses performance gap minimization to improve the boosting in transfer learning, extending on the work of [63]. The complexity of distribution also plays an important part in providing a glimpse of how the distributions and predictions vary. Hence, we plan to investigate other implications of characterizing data by cross-feature complexity, particularly techniques involving correlation to optimal tree depth for network learning models of data.

Funding This research is funded by Laney Graduate School and Rollins Department of Public Health at Emory University. The work of Y. Liu was partially supported by the National Institute of Environmental Health Sciences of the National Institutes of Health (NIH; award 1R01ES032140). The content is solely the responsibility of the authors and does not necessarily represent the o?cial views of NIH.

Code availability The code for the experiments is publicly available on GitHub https://github.com/shrey-gupta/PM25_transfer_learning.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems, vol. 19. MIT Press (2006). https://proceedings.neurips.cc/paper/2006/file/0afa92fc0f8a9cf051bf2961b06ac56b-Paper.pdf
2. Argyriou, A., Pontil, M., Ying, Y., et al.: A spectral regularization framework for multi-task structure learning. In: Advances in Neural Information Processing Systems, vol. 20 (2007)
3. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
4. Bashar, M.A., Nayak, R., Suzor, N.: Regularising LSTM classifier by transfer learning for detecting misogynistic tweets with small training set. Knowl. Inf. Syst. 1–26 (2020)
5. Bengio, Y.: Deep learning of representations for unsupervised and transfer learning texts. In: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, pp. 17–36 (2012)
6. Blanchard, G., Deshmukh, A.A., Dogan, U., et al.: Domain generalization by marginal transfer learning. arXiv preprint arXiv:1711.07910 (2017)
7. Branchaud-Charron, F., Achkar, A., Jodoin, P.M.: Spectral metric for dataset complexity assessment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3215–3224 (2019)
8. Bugallo, M.F., Martino, L., Corander, J.: Adaptive importance sampling in signal processing. Digit. Signal Process. **47**, 36–49 (2015)
9. Camilleri, D., Prescott, T.: Analysing the limitations of deep learning for developmental robotics. In: Conference on Biomimetic and Biohybrid Systems, pp. 86–94. Springer, Berlin (2017)
10. Cao, B., Pan, S.J., Zhang, Y., et al.: Adaptive transfer learning. In: Proceedings of the AAAI Conference on Artificial Intelligence (2010)
11. Chattopadhyay, R., Sun, Q., Fan, W., et al.: Multisource domain adaptation and its application to early detection of fatigue. ACM Trans. Knowl. Discov. Data (TKDD) **6**(4), 1–26 (2012)
12. Chen, L., Cai, Y., Ding, Y., et al.: Spatially fine-grained urban air quality estimation using ensemble semi-supervised learning and pruning. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 1076–1087 (2016)
13. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
14. Cheplygina, V., de Bruijne, M., Pluim, J.P.: Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. Med. Image Anal. **54**, 280–296 (2019)
15. Cortes, C., Mohri, M.: Domain adaptation and sample bias correction theory and algorithm for regression. Theor. Comput. Sci. **519**, 103–126 (2014)
16. Dai, W., Yang, Q., Xue, G.R., et al.: Boosting for transfer learning. In: Proceedings of the 24th International conference on Machine Learning, pp. 193–200 (2007)

17. Dauphin, G.M.Y., Glorot, X., Rifai, S., et al.: Unsupervised and transfer learning challenge: a deep learning approach. In: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, pp. 97–110 (2012)
18. Davis, J., Domingos, P.: Deep transfer via second-order Markov logic. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 217–224 (2009)
19. Day, O., Khoshgoftaar, T.M.: A survey on heterogeneous transfer learning. J. Big Data **4**(1), 29 (2017)
20. Drucker, H.: Improving regressors using boosting techniques. In: ICML, pp. 107–115 (1997)
21. Du, S.S., Koushik, J., Singh, A., et al.: Hypothesis transfer learning via transformation functions. arXiv preprint arXiv:1612.01020 (2016)
22. Elvira, V., Martino, L., Luengo, D., et al.: A gradient adaptive population importance sampler. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4075–4079. IEEE (2015)
23. Elvira, V., Chouzenoux, E., Akyildiz, Ö.D., et al.: Gradient-based adaptive importance samplers. arXiv preprint arXiv:2210.10785 (2022)
24. Fernández, N., Maldonado, C., Gershenson, C.: Information measures of complexity, emergence, self-organization, homeostasis, and autopoiesis. In: Guided Self-Organization: Inception, pp. 19–51. Springer, Berlin (2014)
25. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning, pp. 1126–1135. PMLR (2017)
26. Freund, Y., Schapire, R., Abe, N.: A short introduction to boosting. J. Jpn. Soc. Artif. Intell. **14**(771–780), 1612 (1999)
27. Garcke, J., Vanck, T.: Importance weighted inductive transfer learning for regression. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 466–481. Springer, Berlin (2014)
28. Guan, Z., Li, A., Zhu, T.: Local regression transfer learning with applications to users' psychological characteristics prediction. Brain Inform. **2**(3), 145–153 (2015)
29. Han, D., Liu, Q., Fan, W.: A new image classification method using CNN transfer learning and web data augmentation. Expert Syst. Appl. **95**, 43–56 (2018)
30. Herath, S., Harandi, M., Porikli, F.: Learning an invariant Hilbert space for domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3845–3854 (2017)
31. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. IEEE Trans. Pattern Anal. Mach. Intell. **24**(3), 289–300 (2002)
32. Huang, J., Gretton, A., Borgwardt, K., et al.: Correcting sample selection bias by unlabeled data. Adv. Neural Inf. Process. Syst. **19**, 601–608 (2006)
33. Jebara, T.: Multi-task feature and kernel selection for SVMs. In: Proceedings of the Twenty-First International Conference on Machine Learning, p. 55 (2004)
34. Jing, M., Ma, X., Huang, W., et al.: Task transfer by preference-based cost learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 2471–2478 (2019)
35. Karpatne, A., Ebert-Uphoff, I., Ravela, S., et al.: Machine learning for the geosciences: challenges and opportunities. IEEE Trans. Knowl. Data Eng. **31**(8), 1544–1554 (2018)
36. Katharopoulos, A., Fleuret, F.: Not all samples are created equal: deep learning with importance sampling. In: International Conference on Machine Learning, pp. 2525–2534. PMLR (2018)
37. Lawrence, N.D., Platt, J.C.: Learning to learn with the informative vector machine. In: Proceedings of the Twenty-First International Conference on Machine Learning, p. 65 (2004)
38. Lorena, A.C., Maciel, A.I., de Miranda, P.B., et al.: Data complexity meta-features for regression problems. Mach. Learn. **107**(1), 209–246 (2018)
39. Lv, M., Li, Y., Chen, L., et al.: Air quality estimation by exploiting terrain features and multi-view transfer semi-supervised regression. Inf. Sci. **483**, 82–95 (2019)
40. Ma, J., Cheng, J.C., Lin, C., et al.: Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques. Atmos. Environ. **214**(116), 885 (2019)
41. Maciel, A.I., Costa, I.G., Lorena, A.C.: Measuring the complexity of regression problems. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 1450–1457. IEEE (2016)
42. Mahmud, M., Ray, S.R.: Transfer learning using Kolmogorov complexity: basic theory and empirical evaluations. Tech. rep (2007)
43. Mei, S., Zhu, H.: Adaboost based multi-instance transfer learning for predicting proteome-wide interactions between salmonella and human proteins. PLoS ONE **9**(10), e110,488 (2014)
44. Mihalkova, L., Huynh, T., Mooney, R.J.: Mapping and revising Markov logic networks for transfer learning. In: AAAI, pp. 608–614 (2007)
45. Ngiam, J., Peng, D., Vasudevan, V., et al.: Domain adaptive transfer learning with specialist models. arXiv preprint arXiv:1811.07056 (2018)
46. Obst, D., Ghattas, B., Cugliari, J., et al.: Transfer learning for linear regression: a statistical test of gain. arXiv preprint arXiv:2102.09504 (2021)
47. Oliver, A., Odena, A., Raffel, C., et al.: Realistic evaluation of deep semi-supervised learning algorithms. arXiv preprint arXiv:1804.09170 (2018)
48. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2009)
49. Pan, W., Xiang, E., Liu, N., et al.: Transfer learning in collaborative filtering for sparsity reduction. In: Proceedings of the AAAI Conference on Artificial Intelligence (2010)
50. Pardoe, D., Stone, P.: Boosting for regression transfer. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, pp. 863–870 (2010)
51. Qi, Z., Wang, T., Song, G., et al.: Deep air learning: interpolation, prediction, and feature analysis of fine-grained air quality. IEEE Trans. Knowl. Data Eng. **30**(12), 2258–2297 (2018)
52. Ramakrishnan, R., Shah, J.: Towards interpretable explanations for transfer learning in sequential tasks. In: AAAI 2016 Spring Symposium (2016)
53. Rosenstein, M.T., Marx, Z., Kaelbling, L.P., et al.: To transfer or not to transfer. In: NIPS 2005 Workshop on Transfer Learning, pp. 1–4 (2005)
54. Salaken, S.M., Khosravi, A., Nguyen, T., et al.: Seeded transfer learning for regression problems with deep learning. Expert Syst. Appl. **115**, 565–577 (2019)
55. Schuster, I.: Gradient importance sampling. arXiv preprint arXiv:1507.05781 (2015)
56. Schwaighofer, A., Tresp, V., Yu, K.: Learning gaussian process kernels via hierarchical Bayes. In: Advances in Neural Information Processing Systems, vol. 17 (2004)
57. Sugiyama, M., Suzuki, T., Nakajima, S., et al.: Direct importance estimation for covariate shift adaptation. Ann. Inst. Stat. Math. **60**(4), 699–746 (2008)
58. Sun, Y., Todorovic, S., Li, J.: Reducing the overfitting of AdaBoost by controlling its data distribution skewness. Int. J. Pattern Recognit. Artif. Intell. **20**(07), 1093–1116 (2006)
59. Swarup, S., Ray, S.R.: Cross-domain knowledge transfer using structured representations. In: AAAI, pp. 506–511 (2006)
60. Tan, C., Sun, F., Kong, T., et al.: A survey on deep transfer learning. In: International Conference on Artificial Neural Networks, pp. 270–279. Springer, Berlin (2018)

61. Tang, D., Yang, X., Wang, X.: Improving the transferability of the crash prediction model using the TrAdaBoost. R2 algorithm. Accid. Anal. Prev. **141**, 105,551 (2020)
62. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: a survey. J. Mach. Learn. Res. **10**(7) (2009)
63. Wang, B., Mendez, J.A., Cai, M.B., et al.: Transfer learning via minimizing the performance gap between domains. In: Advances in Neural Information Processing Systems (2019a)
64. Wang, T., Huan, J., Zhu, M.: Instance-based deep transfer learning. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 367–375. IEEE (2019b)
65. Wang, Y., Yao, Q., Kwok, J.T., et al.: Generalizing from a few examples: a survey on few-shot learning. ACM Comput. Surv. (CSUR) **53**(3), 1–34 (2020)
66. Wei, P., Sagarna, R., Ke, Y., et al.: Uncluttered domain subsimilarity modeling for transfer regression. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 1314–1319. IEEE (2018)
67. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. J. Big data **3**(1), 9 (2016)
68. Xu, X., He, H., Zhang, H., et al.: Unsupervised domain adaptation via importance sampling. IEEE Trans. Circuits Syst. Video Technol. **30**(12), 4688–4699 (2019)
69. Yao, H., Liu, Y., Wei, Y., et al.: Learning from multiple cities: a meta-learning approach for spatial-temporal prediction. In: The World Wide Web Conference, pp. 2181–2191 (2019)
70. Zhang, J., Ding, Z., Li, W., et al.: Importance weighted adversarial nets for partial domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8156–8164 (2018)
71. Zhang, K., Zhang, H., Liu, Q., et al.: Interactive attention transfer network for cross-domain sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 5773–5780 (2019)
72. Zhao, P., Zhang, T.: Stochastic optimization with importance sampling for regularized loss minimization. In: International Conference on Machine Learning, pp. 1–9. PMLR (2015)
73. Zhuang, F., Qi, Z., Duan, K., et al.: A comprehensive survey on transfer learning. Proc. IEEE **109**(1), 43–76 (2020)
74. Zuo, H., Zhang, G., Pedrycz, W., et al.: Fuzzy regression transfer learning in Takagi-Sugeno fuzzy models. IEEE Trans. Fuzzy Syst. **25**(6), 1795–1807 (2016)