A General Class of Transfer Learning Regression without Implementation Cost

Shunya Minami,[1] **Song Liu,** [2] **Stephen Wu,** [1,3] **Kenji Fukumizu,** [1,3] **Ryo Yoshida** [1,3,4]
[1] The Graduate University for Advanced Studies (SOKENDAI)
[2] University of Bristol
[3] The Institute of Statistical Mathematics
[4] National Institute for Materials Science
{mshunya, stewu, fukumizu, yoshidar}@ism.ac.jp, song.liu@bristol.ac.uk

Abstract

We propose a novel framework that unifies and extends existing methods of transfer learning (TL) for regression. To bridge a pretrained source model to the model on a target task, we introduce a density-ratio reweighting function, which is estimated through the Bayesian framework with a specific prior distribution. By changing two intrinsic hyperparameters and the choice of the density-ratio model, the proposed method can integrate three popular methods of TL: TL based on cross-domain similarity regularization, a probabilistic TL using the density-ratio estimation, and fine-tuning of pretrained neural networks. Moreover, the proposed method can benefit from its simple implementation without any additional cost; the regression model can be fully trained using off-the-shelf libraries for supervised learning in which the original output variable is simply transformed to a new output variable. We demonstrate its simplicity, generality, and applicability using various real data applications.

Introduction

Transfer learning (TL) (Pan and Yang 2009; Yang et al. 2020) is an increasingly popular machine learning framework that covers a broad range of techniques of repurposing a set of pretrained models on source tasks for another task of interest. It is proven that TL has the potential to improve the prediction performance on the target task significantly, in particular, given a limited supply of training data in which the learning from scratch is less effective. To date, the most outstanding successes of TL have been achieved by refining and reusing specific layers of deep neural networks (Yosinski et al. 2014). One or more layers in the pretrained neural networks are refined according to the new task using a limited target dataset. The remaining layers are either frozen (frozen featurizer) or almost unchanged (fine-tuning) during the cross-domain adaptation.

In this study, we aim to establish a new class of TL, which is applicable to any regression models. The proposed class unifies different classes of existing TL methods for regression. To model the transition from a pretrained model to a new model, we introduce a density-ratio reweighting function. The density-ratio function is estimated by conducting a Bayesian inference with a specific prior distribution while keeping the given source model unchanged. Two hyperparameters and the choice of the density-ratio model characterize the proposed class. It can integrate and extend three popular methods of TL within a unified framework, including TL based on the cross-domain similarity regularization (Jalem et al. 2018; Marx et al. 2005; Raina, Ng, and Koller 2006; Kuzborskij and Orabona 2013, 2017), probabilistic TL using the density-ratio estimation (Liu and Fukumizu 2016; Sugiyama, Suzuki, and Kanamori 2012), and the fine-tuning of pretrained neural networks (Hinton, Vinyals, and Dean 2015; Kirkpatrick et al. 2017; Yosinski et al. 2014).

In general, the model transfer operates through a regularization scheme to leverage the transferred knowledge between different tasks. A conventional regularization aims to retain similarity between the pretrained and transferred models. This natural idea is what we referred to as the cross-domain similarity regularization. On the other hand, the density-ratio method operates with an opposite learning objective that we call the cross-domain dissimilarity regularization; the discrepancy between two tasks is modeled and inferred, and the transferred model is a weighted sum of the pretrained source model and the newly trained model on the discrepancy. These totally different methods can be unified within the proposed framework.

To summarize, the features and contributions of our method are as follows:

- The method can operate with any kinds of regression models.
- The proposed class, which has two hyperparameters, can unify and hybridize three existing methods of TL, including the regularization based on cross-domain similarity and dissimilarity.
- The two hyperparameters and a model for the density-ratio function are selected through cross-validation. With this unified workflow, an ordinary supervised learning without transfer can also be chosen if the previous learning experience interferes with learning in the new task.
- The proposed method can be implemented with no extra cost. With a simple transformation of the output variable, the model can be trained using off-the-shelf libraries for regression that implement the ℓ_2-loss minimization with any regularization scheme. In addition, the method is applicable in scenarios where only the source model is accessible

but not the source data, for example, due to privacy reasons.

Practical benefits of bridging totally different methods in the unified workflow are tested on a wide range of prediction tasks in science and engineering applications.

Proposed Method

We are given a pretrained model $y = f_s(x)$ on the source task, which defines the mapping between any input x to a real-valued output $y \in \mathbb{R}$. The objective is to transform the given $f_s(x)$ into a target model $y = f_t(x)$ by using n instances from the target domain, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$.

Inspired by (Liu and Fukumizu 2016), we apply the probabilistic modeling for the transition from $f_s(x)$ to $f_t(x)$. With the conditional distribution $p_s(y|x)$ of the source task, the one on the target can be written as

$$p_t(y|x) = w(y, x)p_s(y|x)$$

where $w(y, x) = p_t(y|x)/p_s(y|x)$. Consider that the source distribution is modelled by $p_s(y|x, f_s)$ which involves the pretrained $f_s(x)$. In addition, the density-ratio function $w(y, x)$ is separately modeled as $w(y, x|\theta_w)$ with an unknown parameter θ_w, which will be associated with a regression model $f_{\theta_w}(x)$. The target model $p_t(y|x, \theta_w)$ is then

$$p_t(y|x, \theta_w) = w(y, x|\theta_w)p_s(y|x, f_s) \quad (1)$$
$$\text{such that } \forall x : \int w(y, x|\theta_w)p_s(y|x, f_s)\mathrm{d}y = 1,$$

where the normalization constraint is due to the fact that the conditional probability needs to be normalized to 1 over its domain.

We employ Bayesian inference to estimate the unknown θ_w in the density-ratio model $w(y, x|\theta_w)$. The target model $p_t(y|x, \theta_w)$ is used as the likelihood for Bayesian inference, and a prior distribution $p(\theta_w|f_s)$ is placed on θ_w, which depends on the given f_s. The posterior distribution is then

$$p(\theta_w|\mathcal{D}) \propto \prod_{i=1}^n p_t(y_i|x_i, \theta_w)p(\theta_w|f_s). \quad (2)$$

We adopt Gaussian models for the likelihood function as

$$w(y, x|\theta_w) \propto \exp\left(-\frac{(y - f_{\theta_w}(x))^2}{\sigma}\right), \quad (3)$$

$$p_s(y|x, f_s) \propto \exp\left(-\frac{(y - f_s(x))^2}{\eta}\right), \quad (4)$$

where $\sigma > 0$ and $\eta > 0$. The normalization constant for the product of the two expressions on the right-hand side of Eq. 3 and Eq. 4 is given as $\exp\left(-(\sigma + \eta)^{-1}(f_s(x) - f_{\theta_w}(x))^2\right)$, which depends on the proximity of $f_{\theta_w}(x)$ to $f_s(x)$. In addition, we regularize the training based on the discrepancy of the two models $f_{\theta_w}(x)$ and $f_s(x)$, which can belong to different classes of regression models. In order to do so, we introduce a prior distribution that implements a function-based regularization as

$$p(\theta_w|f_s) \propto \exp\left(-\sum_{i=1}^m \frac{(f_s(u_i) - f_{\theta_w}(u_i))^2}{\lambda}\right), \quad (5)$$

where $\lambda \in \mathbb{R}\backslash\{0\}$. The discrepancy is measured by the sum of their squared distances over m input values $\mathcal{U} = \{u_i\}_{i=1}^m$. Hereafter, we use the n observed inputs in $\mathcal{D}$ for $\mathcal{U}$. The posterior distribution involves three hyperparameters (σ, η, λ). Note that λ can be either positive or negative and controls the degree of discrepancy, positively or negatively. As described below, this Gaussian-type modeling leads to an analytic workflow that can benefit from less effort on the implementation.

We consider the Maximum a Posteriori (MAP) estimation of θ_w and a class of prediction functions $\hat{y}(x)$ that are characterized by two hyperparameters τ and ρ:

$$\hat{\theta}_w = \arg\min_{\theta_w} \sum_{i=1}^n \left\{(y_i - f_{\theta_w}(x_i))^2 - \tau(f_s(x_i) - f_{\theta_w}(x_i))^2\right\}, \quad (6)$$

$$\hat{y}(x) = \mathrm{argmax}_y\, p_t(y|x, \hat{\theta}_w) = (1 - \rho)f_{\hat{\theta}_w}(x) + \rho f_s(x), \quad (7)$$

$$\tau = \frac{\sigma}{\sigma + \eta} - \frac{\sigma}{\lambda} \in (-\infty, 1), \;\; \rho = \frac{\sigma}{\sigma + \eta} \in (0, 1).$$

In the training objective Eq. 6, the first term measures the goodness-of-fit with respect to $\mathcal{D}$. The second term is derived from the normalization term in Eq. 1 and the prior distribution Eq. 5. It regularizes the training through the discrepancy between $f_{\theta_w}(x)$ and the pretrained $f_s(x)$. The prediction function Eq. 7 corresponds to the mode of the plug-in predictive distribution Eq. 1. Note that the original three hyperparameters are reduced to $\tau \in (-\infty, 1)$ and $\rho \in (0, 1)$. By varying (τ, ρ) and different models on $f_{\theta_w}(x)$ coupled with the learning algorithms, the resulting method can bridge various methods of TL as described later.

Implementation Cost

By completing the square of Eq. 6 with respect to $f_{\theta_w}(x)$, the objective function can be rewritten as a residual sum of squares on a transformed output variable z:

$$\hat{\theta}_w = \arg\min_{\theta_w} \sum_{i=1}^n (z_i - f_{\theta_w}(x_i))^2, \quad z_i = \frac{y_i - \tau f_s(x_i)}{1 - \tau}.$$

Once the original output y_i is simply converted to z_i with a given $f_s(x)$ and τ, the model can be trained by using a common ℓ_2-loss minimization library for regression. Any regularization term, such as ℓ_1- or ℓ_2-regularization, can also be added. Therefore, the proposed method can be implemented at essentially no cost. In the applications shown later, we utilized ridge regression, random forest regression, and neural networks as $f_{\theta_w}(x)$. We simply used the standard libraries of the R language (glmnet, ranger, and MXNet) without any customization or additional coding.

Furthermore, as no source data appear in the objective function, the model is learnable by using only training instances in a target domain as long as a source model is callable. This separately learnable property will be a great advantage in cases, for example, where training the source model from scratch is time-consuming, or the source data can not be disclosed.

Relations to Existing Methods

By adjusting (τ, ρ) coupled with the choice of $f_{\theta_w}(x)$, our method can represent the different types of TL as described

below. The relationship between different methods are visually overviewed in Figure 1.

Regularization Based on Cross-Domain Similarity

One of the most natural ideas for model refinement is to use the similarity to the pretrained $f_s(x)$ as a constraint condition. Many studies have been made so far to incorporate such cross-domain similarity regularization to TL or other related machine learning tasks such as avoiding catastrophic forgetting in continual learning (Kirkpatrick et al. 2017), knowledge distillation to compress complex neural networks to simpler models (Hinton, Vinyals, and Dean 2015).

Here, this type of regularization is described in a Bayesian fashion. We consider a posterior distribution in Eq. 2, but impose the Gaussian distribution on the likelihood $p_t(y|x,\theta_w) = \mathcal{N}(y|f_{\theta_w}(x),\sigma)$ and the same prior to Eq. 5 is imposed to $p(\theta_w|f_s)$. Then, the MAP estimator for θ_w and the mode of the plug-in predictive distribution are of the following form

$$\hat{\theta}_w = \arg\min_{\theta_w} \sum_{i=1}^{n} \{(y_i - f_{\theta_w}(x_i))^2 + \frac{\sigma}{\lambda}(f_s(x_i) - f_{\theta_w}(x_i))^2\}, \tag{8}$$

$$\hat{y}(x) = f_{\hat{\theta}_w}(x). \tag{9}$$

The objective function of our method Eq. 6 can represent the MAP estimation with the objective function in Eq. 8 by restricting the hyperparameter τ (or λ) to be negative, i.e., $\tau = -\sigma/\lambda < 0$. The prediction function in Eq. 9 corresponds to $\rho = 0$ in our method. With a negative τ, the model $f_{\theta_w}(x)$ is estimated to be closer to the pretrained source model. Such a newly trained model $f_{\hat{\theta}_w}(x)$ is directly used as the prediction function without using the source model.

Transfer Learning Based on Neural Networks

To our best knowledge, the most powerful and widely used method of TL relies on deep neural networks (Yosinski et al. 2014). When neural networks are put on both $f_{\theta_w}(x)$ and $f_s(x)$ in the objective function Eq. 8, the pretrained $f_s(x)$ is fine-tuned to $f_{\theta_w}(x)$ by retaining the cross-domain similarity between their output layers.

Transfer Learning Based on the Density-Ratio Estimation

The density-ratio TL of (Liu and Fukumizu 2016) was designed to minimize the conditional Kullback-Leibler divergence $\mathbb{E}_{x\sim q(x)}[\mathrm{KL}(q(y|x)||p_t(y|x,\theta_w))]$ between the true density $q(y|x)$ and the transferred model $p_t(y|x,\theta_w)$ based on the density-ratio reweighting as in Eq. 1. As detailed in Supplementary Note A[1], if the transfer model is paramterized in the same way as Eq. 3, the learning objective derived from an empirical risk on the training set $\mathcal{D}$ takes the form

$$\hat{\theta}_w = \arg\min_{\theta_w} \sum_{i=1}^{n} \left\{(y_i - f_{\theta_w}(x_i))^2 - \rho(f_s(x_i) - f_{\theta_w}(x_i))^2\right\}, \tag{10}$$

$$\rho = \frac{\sigma}{\sigma + \eta} \in (0,1).$$

[1]All supplementary notes can be found in the arXiv version of the paper.

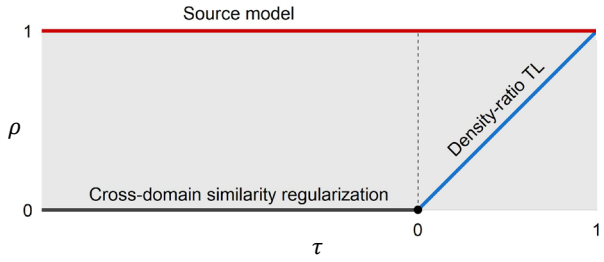


Figure 1: Existing methods mapped onto the hyperparameter space (τ,ρ). The cross-domain similarity regularization corresponds to $\tau < 0$ and $\rho = 0$ (black line). If neural networks are put on both $f_{\theta_w}(x)$ and $f_s(x)$, this region corresponds to the fine-tuning of neural networks. If $\tau = \rho$ (blue line), the class represents the density-ratio TL. The region with $\tau = \rho = 0$ (black dot) or $\rho = 1$ (red line) represents an ordinal regression without transfer or the case where a source model is directly used as the target, respectively.

The second term represents the discrepancy between the density-ratio model and the source model in which the degree of regularizaion is controlled by $\rho \in (0,1)$. For the prediction function, as with Eq. 7, we consider $\hat{y}(x) = (1-\rho)f_{\hat{\theta}_w}(x) + \rho f_s(x)$ that corresponds to the plug-in estimator $\mathrm{argmax}_y\, p_t(y|x,\hat{\theta}_w)$.

In terms of the proposed class of TL, the method in (Liu and Fukumizu 2016) can be considered as a specific choice of $\tau = \rho \in (0,1)$ (the blue line in Figure 1). This corresponds to the case where λ in Eq. 5 is sufficiently large, i.e., the prior distribution for the parameters of the density-ratio function is uniformly distributed and non-informative. It is noted that the objective function in Eq. 10 resembles Eq. 8 in the cross-domain similarity regularization. These two methods are regularized based on the discrepancy between $f_{\theta_w}(x)$ and $f_s(x)$, but their regularization mechanisms work in the opposite directions: the regularization parameter τ takes a positive value for the method in (Liu and Fukumizu 2016), which we call cross-domain dissimilarity regularization, while a negative value for cross-domain similarity regularization.

Learning without Transfer

The proposed family of methods contains two learning schemes without transfer. If the hyperparameters are selected to be $\tau = 0$ and $\rho = 0$ (the black dot in Figure 1), the density-ratio model $\hat{f}_{\theta_w}(x)$ is estimated without using the source model, and the resulting prediction model becomes $\hat{y}(x) = f_{\hat{\theta}_w}(x)$. This corresponds to an ordinary regression procedure. When negative transfer occurs i.e., the previous learning experience interferes with learning in the new task, the desirable hyperparameters would be around $\tau = 0$ and $\rho = 0$. In addition, setting $\rho = 1$ (the red line in Figure 1), the source model alone gives the prediction model as $\hat{y}(x) = f_s(x)$ regardless of $f_{\theta_w}(s)$. By cross-validating the hyperparameters, the proposed framework will automatically determine when not to transfer without using a separate pipelines.

Selection of Hyperparameters and Preference to Bias and Variance

As described above, our method can hybridize various mechanisms of TL by adjusting τ and ρ. The values of the hyperparameters are adjusted through cross-validation. Clearly, the optimal combination of the hyperparameters will differ depending on between-task relationships and the choice for the target model.

Here, we show an expression of the mean squared error (MSE) based on the bias-variance decomposition. For simplicity, we restrict $f_{\hat{\theta}_w}(x)$ to be in the set of all linear predictions taking the form of $f_{\hat{\theta}_w}(x) = x^\mathsf{T}\mathrm{S}\mathbf{z}$. The $n \times n$ smoothing matrix S depends on n samples of p input feature $\phi(x_i) \in \mathbb{R}^p$ $(i = 1, \ldots, n)$ with a predefined basis set ϕ, and $\mathbf{z}$ is a vector of n transformed outputs z_i $(i = 1, \ldots, n)$. For example, this class of prediction includes the kernel ridge regression.

We assume that y follows $y = f_t(x) + \epsilon$ where $f_t(x)$ denotes the true model and the observation noise ϵ has mean zero and variance σ_ϵ^2. For the prediction function $\hat{y}(x) = (1-\rho)f_{\hat{\theta}_w}(x) + \rho f_s(x)$, $\mathrm{MSE}(\hat{y}(x)) = \mathbb{E}_{y|x}[y - \hat{y}(x)]^2$ can be expressed as:

$$\mathrm{MSE}(\hat{y}(x)) = \left[\frac{\rho-\tau}{1-\tau}\mathrm{D}(x) + \frac{1-\rho}{1-\tau}\mathrm{B}_1(x) - \frac{\tau(1-\rho)}{1-\tau}\mathrm{B}_2(x)\right]^2 + \left(\frac{1-\rho}{1-\tau}\right)^2 \mathrm{V}(x) + \sigma_\epsilon^2, \quad (11)$$

where

$$\begin{aligned}\mathrm{D}(x) &= f_t(x) - f_s(x),\\ \mathrm{B}_1(x) &= f_t(x) - x^\mathsf{T}\mathrm{S}\mathbf{f}_t,\\ \mathrm{B}_2(x) &= f_s(x) - x^\mathsf{T}\mathrm{S}\mathbf{f}_s,\\ \mathrm{V}(x) &= \sigma_\epsilon^2 x^\mathsf{T}\mathrm{S}\mathrm{S}^\mathsf{T}x.\end{aligned}$$

The first term is the squared bias, which consists of three building blocks. $\mathrm{D}(x)$ represents the discrepancy between $f_t(x)$ and $f_s(x)$. $\mathrm{B}_1(x)$ is a bias of the linear estimator $x^\mathsf{T}\mathrm{S}\mathbf{f_t}$ with respect to the true model $f_t(x)$, assuming that n observations $\mathbf{f}_t = (f_t(x_1), \ldots, f_t(x_n))^\mathsf{T}$ for the unknown $f_t(x)$ are given. Likewise, $\mathrm{B}_2(x)$ is the bias of $x^\mathsf{T}\mathrm{S}\mathbf{f_s}$ with respect to $f_s(x)$. The second term corresponds to the variance of $\hat{y}(x)$. This is proportional to $\mathrm{V}(x) = \sigma_\epsilon^2 x^\mathsf{T}\mathrm{S}\mathrm{S}^\mathsf{T}x$. The third term is the variance of the observation noise.

The relative magnitudes of $\mathbb{E}_x[\mathrm{D}(x)^2]$, $\mathbb{E}_x[\mathrm{B}_1(x)^2]$, $\mathbb{E}_x[\mathrm{B}_2(x)^2]$, and $\mathbb{E}_x[\mathrm{V}(x)]$ determine the optimal hyperparameters to the cross-domain similarity regularization, the density-ratio TL, and the learning without transfer. Let $\mathrm{D} = \mathrm{D}(x)$, $\mathrm{B}_1 = \mathrm{B}_1(x)$, $\mathrm{B}_2 = \mathrm{B}_2(x)$, and $\mathrm{V} = \mathrm{V}(x)$, respectively. Consider the expectation of the MSE in Eq. 11 with respect the marginal distribution of x: $\mathbb{E}_{x\sim q(x)}[\mathrm{MSE}(\hat{y}(x))]$. Because the expected MSE is quadratic with respect to ρ for any τ, the minimum under the inequality constraint $0 \le \rho \le 1$ is achieved by

$$\rho(\tau) = \begin{cases} 0 & \rho_*(\tau) \le 0 \\ \rho_*(\tau) & 0 < \rho_*(\tau) < 1 \\ 1 & \rho_*(\tau) \ge 1 \end{cases}$$

where $\rho_*(\tau)$ denotes the solution for the unconstrained minimization. Taking the derivative of the expected MSE with respect to ρ, we have an equation as

$$\frac{1}{(1-\tau)^2}\mathbb{E}[((\rho-\tau)\mathrm{D}+(1-\rho)\mathrm{B}_1-\tau(1-\rho)\mathrm{B}_2))(\mathrm{D}-\mathrm{B}_1+\tau\mathrm{B}_2)] - \frac{1-\rho}{(1-\tau)^2}\mathbb{E}[\mathrm{V}] = 0. \quad (12)$$

Assuming that $\tau \neq 1$, this leads to an expression for the unconstrained solution as

$$\rho_*(\tau) = \frac{\mathbb{E}[(\tau\mathrm{D} - \mathrm{B}_1 + \tau\mathrm{B}_2)(\mathrm{D} - \mathrm{B}_1 + \tau\mathrm{B}_2)] + \mathbb{E}[\mathrm{V}]}{\mathbb{E}[\mathrm{D} - \mathrm{B}_1 + \tau\mathrm{B}_2]^2 + \mathbb{E}[\mathrm{V}]}. \quad (13)$$

Likewise, taking the derivative of the expected MSE with respect to τ, we have

$$\frac{1-\rho}{(1-\tau)^3}\mathbb{E}[((\rho-\tau)\mathrm{D}+(1-\rho)\mathrm{B}_1-\tau(1-\rho)\mathrm{B}_2))(\mathrm{D}-\mathrm{B}_1+\mathrm{B}_2)] - \frac{(1-\rho)^2}{(1-\tau)^2}\mathbb{E}[\mathrm{V}] = 0. \quad (14)$$

Combining Eq. 12 and Eq. 14 where $\tau \neq 1$ and $\rho \neq 1$, we obtain an equation

$$(1-\tau)\mathbb{E}[\tau(\mathrm{D} + (1-\rho)\mathrm{B}_2)\mathrm{B}_2 - (1-\rho)\mathrm{B}_1\mathrm{B}_2 + \rho\mathrm{D}\mathrm{B}_2] = 0,$$

then yielding an expression for the solution

$$\tau(\rho) = \frac{(1-\rho)\mathbb{E}[\mathrm{B}_1\mathrm{B}_2] + \rho\mathbb{E}[\mathrm{D}\mathrm{B}_2]}{(1-\rho)\mathbb{E}[\mathrm{B}_2^2] + \mathbb{E}[\mathrm{D}\mathrm{B}_2]}. \quad (15)$$

According to the two expressions in Eq. 13 and Eq. 15, we can investigate the preference in the hyperparameter selection in regard to the bias and variance components in the data generation process.

Consider a case where the source and target models are significantly different by taking the limit $\mathbb{E}[\mathrm{D}^2] \to \infty$. For the expectation of $\mathbb{E}[\mathrm{D}\mathrm{X}]$ for the product of D and any X, it holds that $\mathbb{E}[\mathrm{D}\mathrm{X}]/\mathbb{E}[\mathrm{D}^2] \to 0$ as $\mathbb{E}[\mathrm{D}^2] \to \infty$. This can be seen by considering the Cauchy-Schwarz inequality:

$$\begin{aligned}-\mathbb{E}[\mathrm{D}^2]^{\frac{1}{2}}\mathbb{E}[\mathrm{X}^2]^{\frac{1}{2}} \le \mathbb{E}[\mathrm{D}\mathrm{X}] &\le \mathbb{E}[\mathrm{D}^2]^{\frac{1}{2}}\mathbb{E}[\mathrm{X}^2]^{\frac{1}{2}}\\ \Leftrightarrow -\frac{\mathbb{E}[\mathrm{X}^2]^{\frac{1}{2}}}{\mathbb{E}[\mathrm{D}^2]^{\frac{1}{2}}} \le \frac{\mathbb{E}[\mathrm{D}\mathrm{X}]}{\mathbb{E}[\mathrm{D}^2]} &\le \frac{\mathbb{E}[\mathrm{X}^2]^{\frac{1}{2}}}{\mathbb{E}[\mathrm{D}^2]^{\frac{1}{2}}}.\end{aligned}$$

In the second line, the upper- and the lower-bounds go to zero as $\mathbb{E}[\mathrm{D}^2] \to \infty$. Thus, in Eq. 13, all terms except those having $\mathbb{E}[\mathrm{D}^2]$, which appear in its numerator and denominator, approach asymptotically to zero, which results in

$$\rho_*(\tau) \to \frac{\tau\mathbb{E}[\mathrm{D}^2]}{\mathbb{E}[\mathrm{D}^2]} = \tau \ \text{ as } \ \mathbb{E}[\mathrm{D}^2] \to \infty.$$

Furthermore, noting that $\mathbb{E}[\mathrm{D}\mathrm{X}] = O(\mathbb{E}[\mathrm{D}^2]^{\frac{1}{2}})$, it can been seen that $\tau(\rho)$ in Eq. 15 approaches asymptotically ρ:

$$\tau(\rho) \to \frac{\rho\mathbb{E}[\mathrm{D}\mathrm{B}_2]}{\mathbb{E}[\mathrm{D}\mathrm{B}_2]} = \rho \ \text{ as } \ \mathbb{E}[\mathrm{D}^2] \to \infty.$$

Therefore, when $\mathbb{E}[\mathrm{D}^2]$ dominates the other three quantities, the density-ratio TL ($\tau = \rho$) is preferred. This fact accounts for the experimental observations presented above.

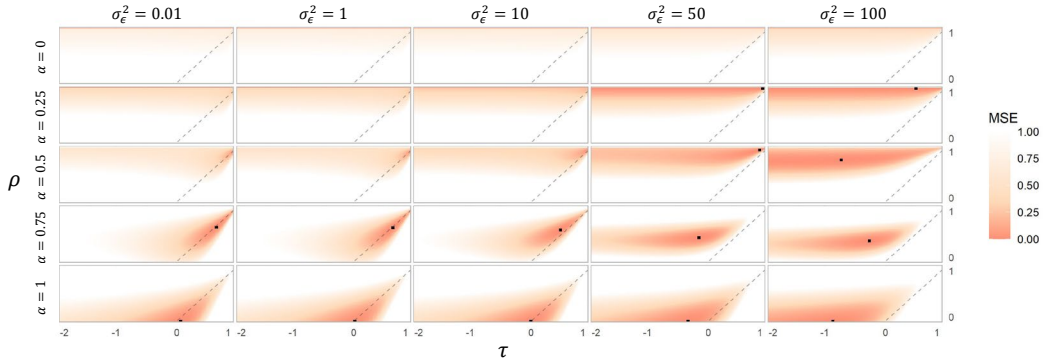


Figure 2: Heatmap display of the MSE landscape on the hyperparameter space (τ, ρ) that changes as a function of the bias (α) and variance (σ_ϵ). With given τ and ρ, the linear ridge regression was used to train $f_{\theta_w}(x)$ on the artificial data. The black dot denotes the lowest MSE.

On the other hand, if the source and target models are completely the same ($\mathbb{E}[\mathrm{D}^2] = 0$), it holds that $\rho_*(\tau) = 1$. Alternatively, if $\mathbb{E}[\mathrm{V}] \to \infty$, $\rho_*(\tau) = 1$. The direct use of the source model as a prediction function tends to be optimal as the source and target tasks get closer or the variance $\mathbb{E}[\mathrm{V}]$ becomes larger. It has not yet been clear when the cross-domain similarity regularization would be preferred, either theoretically or experimentally.

Results

Illustrative Example

Some intrinsic properties of the proposed method are illustrated by presenting numerical examples using artificial data. According to our experience, there is a link between the bias and variance magnitudes and the hyperparameters that minimize the MSE. This will be demonstrated.

We assumed the true functions on the source and target tasks to be linear as $f_t(x) = x^\mathsf{T}\theta_t$ and $f_s(x) = x^\mathsf{T}\theta_s$ where $x \in \mathbb{R}^{300}$. The true parameters were generated as $\theta_t = \alpha\theta_s + (1-\alpha)\theta_w$ where $\theta_s \sim \mathcal{N}(0, \mathrm{I})$ and $\theta_w \sim \mathcal{N}(0, \mathrm{I})$. The output variable was assumed to follow $y = f_t(x) + \epsilon$ where $x \sim \mathcal{N}(0, \mathrm{I})$ and $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. With the given θ_w and θ_s, we generated $\{x_i, y_i\}_{i=1}^n$ with the sample size set to $n = 50$ by randomly sampling x and ϵ. The discrepancy between the source and target models is controlled by the mixing rate $\alpha \in [0, 1]$ for any given θ_w. In particular, if α is set to zero, the source and target models are the same ($\forall x$: $\mathrm{D}(x) = 0$ in Eq. 11). The variance σ_ϵ^2 of the observational noises affects the magnitude of the variance $\mathbb{E}[\mathrm{V}]$ in the model estimation.

We used the linear ridge regression to estimate f_{θ_w} with the hyperparameter on the ℓ_2-regularization that was fixed at $\lambda = 0.0001$. The true source model was used as $f_s(z)$. We then investigated the change of the MSE landscape as a function of the bias α and the variance σ_ϵ, which are summarized in Figure 2. For any given values of τ and ρ, the MSE was approximately evaluated by averaging the ℓ_2-loss over additionally generated 1,000 samples on (x, y) and rescaled to the range in $[0, 1]$. For $\alpha = 0$ where the source and target models are the same, the MSE became small in the region along $\rho = 1$ that corresponds to the use of the pretrained source model as the target model with no modification. As α increased while keeping σ_ϵ at smaller values, the region where the MSE becomes small was concentrated around $\tau = \rho$, indicating the dominant performance of the density-ratio TL. On the other hand, as both α and σ_ϵ became larger, the region with $\tau < 0$ and $\rho = 0$ tended to be more favored. This region corresponds to the TL with the cross-domain similarity regularization. It was confirmed that the pattern of the MSE landscape varies continuously with respect to the bias and variance components.

In many other applications, we have often observed the same trend on the preference of τ and ρ with respect to the relative magnitude of the bias and variance. Another example assuming nonlinear models for $f_s(x)$ and $f_t(x)$, and random forests for $f_{\theta_w}(x)$ is shown in Supplementary Note B.

Real Data Applications

Task, Data and Analysis Procedure The proposed method was applied to five real data analyses in materials science and robotics applications: (i) multiple properties of organic polymers and inorganic compounds (Yamada et al. 2019), (ii) multiple properties of polymers (Kim et al. 2018) and low-molecular-weight compounds (monomers, unpublished data), (iii) properties of donor molecules in organic solar cells (Paul et al. 2019) obtained from experiments (Lopez et al. 2016) and quantum chemical calculations (Pyzer-Knapp, Li, and Aspuru-Guzik 2015), (iv) formation energies of various inorganic compounds and crystal polymorphisms of SiO_2 and CdI_2 (Jain et al. 2013), and (v) the feed-forward torques required to follow a desired trajectory at seven joints of a SARCOS anthropomorphic robot arm (Williams and Rasmussen 2006). The model transfers were conducted exhaustively between all task pairs within each application, which resulted in a total of 185 pairs of the source and target tasks with 9 different combinations of $f_s(x)$ and $f_{\theta_w}(x)$ (a total of 1,665 cases).

For each task pair, we used three machine learning algorithms; Ridge regression using a linear model (LN), random forests (RF), and neural networks (NN) to estimate $f_s(x)$ and $f_{\theta_w}(x)$. In the source task, the entire dataset was used to train $f_s(x)$ under default settings of software packages without

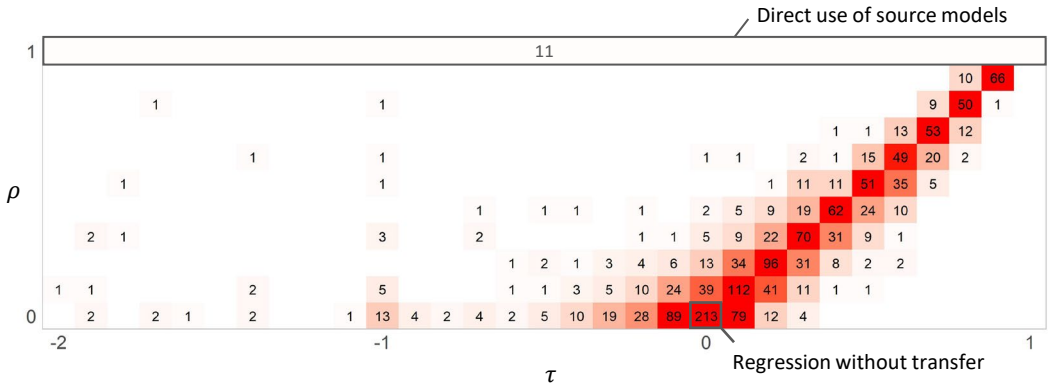


Figure 3: Distribution of (τ, ρ) that delivered the lowest MSE in 1,665 cases (185 task pairs and 3^2 combinations of models for $f_s(x)$ and $f_{\theta_w}(x)$). The number in each pixel denotes the count of cases.

Source task	Target task	$f_s(x)$	$f_{\theta_w}(x)$ LN	$f_{\theta_w}(x)$ RF	$f_{\theta_w}(x)$ NN	Selected hyperparameters LN	Selected hyperparameters RF	Selected hyperparameters NN
Monomer - Dielectric constant	Monomer - HOMO-LUMO gap	LN	0.8292	0.7435	0.8823	(-0.1, 0.1)	(0.6, 0.4)	(0.1, 0.3)
		RF	0.8302	**0.7139**	0.7421	(-0.1, 0.2)	(0.5, 0.3)	(0.8, 0.8)
		NN	0.8250	0.7372	0.7644	(-0.2, 0.2)	(0.2, 0.3)	(0.4, 0.4)
	Monomer - Refractive index	LN	0.0436	0.0424	0.0439	(0.8, 0.9)	(0.8, 0.9)	(0.8, 0.9)
		RF	0.0463	0.0415	0.0415	(0.9, 0.9)	(- , 1.0)	(- , 1.0)
		NN	0.0365	**0.0355**	0.0505	(0.8, 0.9)	(0.8, 0.9)	(0.4, 0.7)
	Polymer - Band gap	LN	1.0881	0.7862	0.8936	(0.3, 0.1)	(0.0, 0.1)	(0.6, 0.6)
		RF	0.8594	0.7477	**0.7130**	(-0.2, 0.4)	(0.4, 0.3)	(0.8, 0.8)
		NN	0.8654	0.8598	0.8908	(-0.5, 0.1)	(0.3, 0.5)	(0.6, 0.5)
	Polymer - Dielectric constant	LN	0.6031	**0.5358**	0.6376	(-0.4, 0.2)	(0.3, 0.2)	(-0.5, 0.0)
		RF	0.5988	0.5786	0.6678	(-0.2, 0.2)	(0.3, 0.2)	(0.0, 0.4)
		NN	0.6143	0.5478	0.7563	(-0.1, 0.2)	(0.2, 0.3)	(-0.2, 0.1)
	Polymer - Refractive index	LN	**0.3269**	0.3906	0.3442	(0.0, 0.0)	(-0.4, 0.0)	(0.2, 0.4)
		RF	**0.3269**	0.3574	0.3312	(0.0, 0.0)	(0.1, 0.1)	(0.1, 0.2)
		NN	**0.3269**	0.3845	0.4254	(0.0, 0.0)	(-0.1, 0.1)	(-1.7, 0.0)

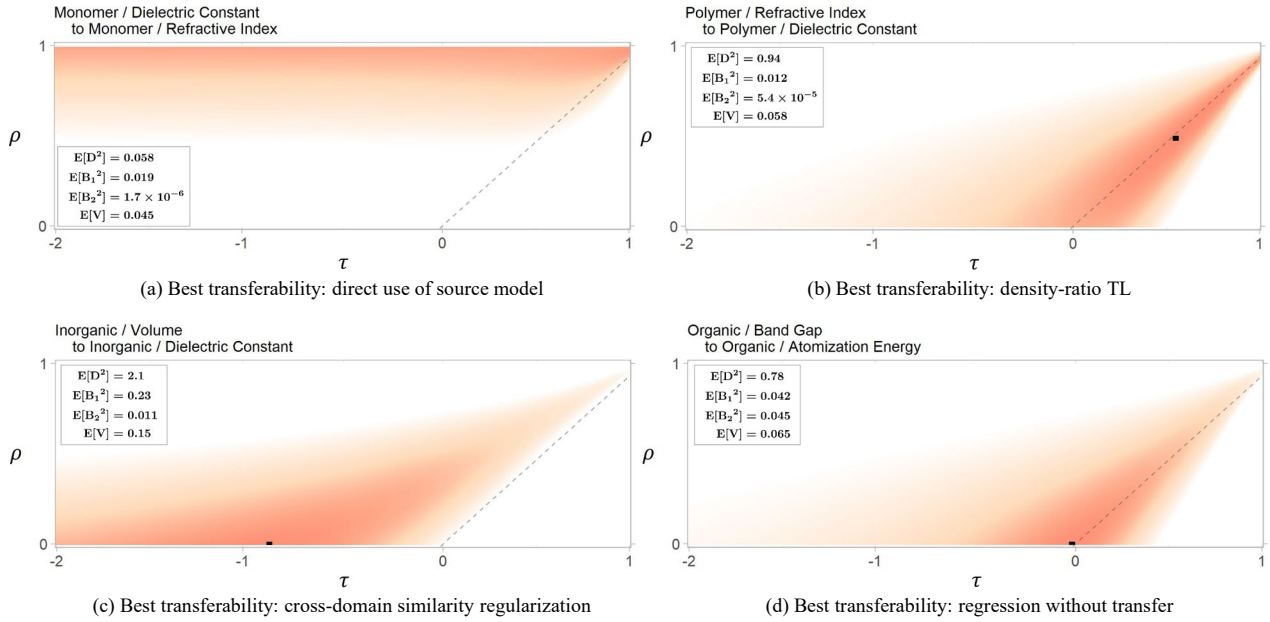
Table 1: Selected hyperparameters (the last three columns, representing hyperparameters τ and ρ) and their corresponding MSEs (the 4-6th columns) for the TL from one source task to five target tasks. Three different models (LN: linear, RF: random forests, and NN: neural networks) were applied to $f_s(x)$ and $f_{\theta_w}(x)$. Supplementary Note C provides full results for all the 1,665 cases.

adjusting hyperparameters. In all cases, 50 randomly selected samples were used to train $f_{\theta_w}(x)$. We choose the best model based on the 5-fold cross validation. The resulting model was used to predict all the remaining data, and the MSE was evaluated. Details of the datasets and analysis procedure are presented in Supplementary Note C.

Results Throughout all the 1,665 cases, we investigated how the hyperparameters selected by the cross-validation are distributed (Figure 3). In many cases, the distribution of the selected hyperparameters was concentrated in the neighboring areas of the density-ratio TL ($\tau = \rho$) and the cross-domain similarity regularization ($\tau < 0, \rho = 0$). The density-ratio TL was selected for 609 cases (36.6%) and the cross-domain similarity regularization was selected for 176 cases (10.6%). In particular, there was a significant bias toward the neighbors of $\tau = \rho$.

The selected hyperparameters and the MSEs for the 1,665 cases are presented in Tables S1-S5 of the Supplementary Note. As an illustrative example, Table 1 shows the result of the TL from one source task (prediction of a dielectric property of small molecules) to five target tasks (prediction of two properties of small molecules and three properties of polymers). This result also indicates the presence of bias toward τ and ρ. It was also observed that in some cases the choice of the density-ratio model significantly affects the prediction performance and in other cases it does not.

We speculate that the four quantities $\mathbb{E}_x[\mathrm{D}^2]$, $\mathbb{E}_x[\mathrm{B}_1^2]$, $\mathbb{E}_x[\mathrm{B}_2^2]$ and $\mathbb{E}_x[\mathrm{V}]$ or their counterparts in general regression, determine the preference of τ and ρ. Figure 4 shows the MSE mapped on the hyperparameter space and the four quantities for four task pairs. They were selected as the typical cases where the four different learning schemes are preferred. The proposed method exhibited the preference to direct use of source models when the difference between the source and target domains ($\mathbb{E}_x[\mathrm{D}^2]$) was small. When



(a) Best transferability: direct use of source model

(b) Best transferability: density-ratio TL

(c) Best transferability: cross-domain similarity regularization

(d) Best transferability: regression without transfer

Figure 4: The MSE landscapes of the hyperparameter space for four different cases that exhibited the best transferability in different hyperparameter sets. Sample estimates on three bias-related quantities ($\mathbb{E}_x[\mathrm{D}^2]$, $\mathbb{E}_x[{\mathrm{B}_1}^2]$, and $\mathbb{E}_x[{\mathrm{B}_2}^2]$) and the mean variance ($\mathbb{E}_x[\mathrm{V}]$) are shown on each plot.

$\mathbb{E}_x[\mathrm{D}^2]$ was large, the relative magnitude of $\mathbb{E}_x[\mathrm{D}^2]$ and the other three quantities $\mathbb{E}_x[{\mathrm{B}_1}^2]$, $\mathbb{E}_x[{\mathrm{B}_2}^2]$ and $\mathbb{E}_x[\mathrm{V}]$ would determine the choice; if $\mathbb{E}_x[\mathrm{V}]$ was small, the density-ratio TL was preferred, and if $\mathbb{E}_x[\mathrm{V}]$ was large, the cross-domain similarity regularization was preferred. Furthermore, when both $\mathbb{E}_x[{\mathrm{B}_1}^2]$ and $\mathbb{E}_x[\mathrm{V}]$ were small, training without transfer was preferred. Such relationships were often observed in other cases as well. However, these are only views derived from partial observations, and there would be more complex factors to work in the learning mechanism. Supplementary Note C shows the results of investigating the magnitudes of the bias and variance and the selected hyperparameters for all cases.

Concluding Remarks

We proposed a new class of TL that is characterized by two hyperparameters which in turn control training and prediction procedure. This new class of TL unifies two different types of existing methods that are based on the cross-domain similarity regularization and the density-ratio estimation. If we use neural networks on the source and target models, the class represents the fine tuning of neural networks. In addition, some specific selection of hyperparameters offers the choice of ordinary regression without transfer or the direct use of a pretrained source model as the target. According to the choice of hyperparameters and models, we can derive various learning methods in which these two methods are hybridized.

The cross-domain similarity regularization and the density-ratio TL follow opposite learning objectives. In the former case, the target model is regularized as being closer to the source model. In the latter case, the difference between the source and target models is estimated to be far away from the source model. Most of the widely used techniques have adopted the former approach that leverages the proximity of the target model to the source model. Interestingly, in many cases, the cross-domain similarity regularization rarely exhibited the best transferability according to our empirical study, and often, the density-ratio estimation or its neighboring areas in the hyperparameter space showed better performances. Although the idea of the cross-domain similarity regularization is more widely adopted, our results indicate that we should further explore the direction based on the opposite idea, such as the density-ratio estimation.

This study focused on the regression setting. In addition, in the Bayesian framework, we assumed the specific type of the likelihood and prior distribution. The empirical risk derived from this assumption takes the sum of the squared loss. With this formulation, we could perform the model training simply by using an existing library for regression. This allows us to keep the implementation cost to practically zero. However, there are also limitations of using the squared loss. We should consider a wide range of loss functions and learning tasks. The treatment of more general loss functions and discriminant problems is one of the future issues.

Acknowledgments

Ryo Yoshida acknowledges the financial support received from a Grant-in-Aid for Scientific Research (A) 19H01132 from the Japan Society for the Promotion of Science (JSPS), JST CREST Grant Number JPMJCR19I1, JPNP16010 commissioned by the New Energy and Industrial Technology Development Organization (NEDO), and JSPS KAKENHI

Grant Number 19H05820. Stephen Wu acknowledges the financial support received from JSPS KAKENHI Grant Number JP18K18017. This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1

References

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .

Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; and Persson, K. A. 2013. The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials* 1(1): 011002. ISSN 2166532X. doi: 10.1063/1.4812323. URL http://link.aip.org/link/AMPADS/v1/i1/p011002/s1\&Agg=doi.

Jalem, R.; Kanamori, K.; Takeuchi, I.; Nakayama, M.; Yamasaki, H.; and Saito, T. 2018. Bayesian-driven first-principles calculations for accelerating exploration of fast ion conductors for rechargeable battery application. *Scientific Reports* 8(1): 1–10.

Kim, C.; Chandrasekaran, A.; Huan, T. D.; Das, D.; and Ramprasad, R. 2018. Polymer Genome: A data-powered polymer informatics platform for property predictions. *The Journal of Physical Chemistry C* 122(31): 17575–17585.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Demis, H.; Claudia, C.; Dharshan, K.; and Raia, H. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* 114(13): 3521–3526.

Kuzborskij, I.; and Orabona, F. 2013. Stability and hypothesis transfer learning. In *International Conference on Machine Learning*, 942–950.

Kuzborskij, I.; and Orabona, F. 2017. Fast rates by transferring from auxiliary hypotheses. *Machine Learning* 106(2): 171–195.

Liu, S.; and Fukumizu, K. 2016. Estimating Posterior Ratio for Classification: transfer Learning from Probabilistic Perspective. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, 747–755.

Lopez, S. A.; Pyzer-Knapp, E. O.; Simm, G. N.; Lutzow, T.; Li, K.; Seress, L. R.; Hachmann, J.; and Aspuru-Guzik, A. 2016. The Harvard organic photovoltaic dataset. *Scientific Data* 3(1): 1–7.

Marx, Z.; Rosenstein, M. T.; Kaelbling, L. P.; and Dietterich, T. G. 2005. Transfer learning with an ensemble of background tasks. In *NIPS Workshop on Inductive Transfer*.

Pan, S. J.; and Yang, Q. 2009. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10): 1345–1359.

Paul, A.; Jha, D.; Al-Bahrani, R.; Liao, W.-k.; Choudhary, A.; and Agrawal, A. 2019. Transfer learning using ensemble neural networks for organic solar cell screening. In *2019 International Joint Conference on Neural Networks*, 1–8.

Pyzer-Knapp, E. O.; Li, K.; and Aspuru-Guzik, A. 2015. Learning from the Harvard clean energy project: the use of neural networks to accelerate materials discovery. *Advanced Functional Materials* 25(41): 6495–6502.

Raina, R.; Ng, A. Y.; and Koller, D. 2006. Constructing informative priors using transfer learning. In *Proceedings of the 23rd International Conference on Machine Learning*, 713–720.

Sugiyama, M.; Suzuki, T.; and Kanamori, T. 2012. *Density Ratio Estimation in Machine Learning*. Cambridge University Press.

Williams, C. K.; and Rasmussen, C. E. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Yamada, H.; Liu, C.; Wu, S.; Koyama, Y.; Ju, S.; Shiomi, J.; Morikawa, J.; and Yoshida, R. 2019. Predicting materials properties with little data using shotgun transfer learning. *ACS Central Science* 5(10): 1717–1730.

Yang, Q.; Zhang, Y.; Dai, W.; and Pan, S. J. 2020. *Transfer Learning*. Cambridge University Press.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 3320–3328.