

Information Geometrically Generalized Covariate Shift Adaptation

Masanari Kimura¹, Hideitsu Hino^{†, 2, 3}

¹SOKENDAI, Graduate University for Advanced Studies.

Shonan Village, Hayama, Kanagawa 240-0193 Japan

²The Institute of Statistical Mathematics.

10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan

³Center for Advanced Intelligence Project, RIKEN.

1-4-1 Nihonbashi, Chuo-ku, Tokyo 103-0027, Japan

[†]corresponding author: mkimura@ism.ac.jp

Keywords: Information Geometry, Domain Adaptation, Covariate Shift

Abstract

Many machine learning methods assume that the training and test data follow the same distribution. However, in the real world, this assumption is very often violated. In particular, the phenomenon that the marginal distribution of the data changes is called covariate shift, one of the most important research topics in machine learning. We show that the well-known family of covariate shift adaptation methods is unified in the framework of information geometry. Furthermore, we show that parameter search for geometrically generalized covariate shift adaptation method can be achieved efficiently. Numerical experiments show that our generalization can achieve better performance than the existing methods it encompasses.

1 Introduction

When considering supervised learning methods, it is often assumed that the training and test data follow the same distribution (Bishop, 1995; Duda et al., 2006; Hastie et al., 2009; Vapnik, 2013; Mohri et al., 2018). However, this common assumption is violated in the real world in most cases (Huang et al., 2007; Zadrozny, 2004; Cortes et al., 2008; Quionero-Candela et al., 2009; Jiang, 2008).

Covariate shift (Shimodaira, 2000) is a prevalent setting for supervised learning in the real world, where the input distribution differs in the training and test phases, but the conditional distribution of the output variable given the input variable remains unchanged. Covariate shift is a commonly observed phenomenon in real-world machine learning applications, such as emotion recognition (Hassan et al., 2013; Jirayucharoensak et al., 2014), 3D pose estimation (Yamada et al., 2012), brain computer interfaces (Li et al., 2010; Raza et al., 2016), spam filtering (Bickel et al., 2009), and human activity recognition (Hachiya et al., 2012). In addition, there has been recent discussion on the relationship between covariate shift and the robustness of deep learning (Ioffe and Szegedy, 2015; Arpit et al., 2016; Santurkar et al., 2018; Nado et al., 2020; Huang and Yu, 2020; Awais et al., 2020).

Ordinary empirical risk minimization (ERM) (Vapnik, 1998, 2013) may not generalize well to the test data under covariate shift because of the difference between the training and test distributions. However, importance weighting for training examples has been shown to be effective in mitigating the effect of covariate shift (Shimodaira, 2000; Sugiyama and Müller, 2005b,a; Zadrozny, 2004). The main idea of these strategies is weighting the training loss terms according to their importance, which is the ratio of the training input density to the test input density. The importance weighting is widely adopted even in modern covariate shift studies with deep neural networks (DNN) (Fang et al., 2020; Zhang et al., 2021).

In this paper, we consider the generalization of these methods in the framework of information geometry (Amari, 1985; Amari and Nagaoka, 2007), a tool that allows us to deal with probability distributions on Riemannian manifolds. This generalization makes it possible to search for good weighting without searching for a large number of

parameters. Our contributions is summarized as follows:

- (Section 4.1 and 4.2) We generalize existing methods of covariate shift adaptation in the framework of information geometry. By our information geometrical formulation, geometric biases of conventional methods are elucidated.
- (Section 4.3) We show that our geometrically generalized covariate shift adaptation method has a much larger solution space than existing methods controlled by only two parameters. Efficient weighting is obtained by searching for parameters using an information criterion or Bayesian optimization.
- (Section 5) Numerical experiments show that our generalization can achieve better performance than the existing methods it encompasses.

2 Preliminaries

2.1 Problem formulation

First, we formulate the problem of supervised learning. We denote by $\mathcal{X} \subset \mathbb{R}^d$ the input space. The output space is denoted by $\mathcal{Y} \subset \mathcal{R}$ (regression) or $\mathcal{Y} \subset \{1, \dots, K\}$ (K -class classification). We assume that training examples $\{(\mathbf{x}_i^{tr}, y_i^{tr})\}_{i=1}^{n_{tr}}$ are independently and identically distributed (i.i.d.) according to some fixed but unknown distribution $p_{tr}(\mathbf{x}, y)$, which can be decomposed into the marginal distribution and the conditional probability distribution, i.e., $p_{tr}(\mathbf{x}, y) = p_{tr}(\mathbf{x})p_{tr}(y|\mathbf{x})$. We also denote the test examples by $\{(\mathbf{x}_i^{te}, y_i^{te})\}_{i=1}^{n_{te}}$ drawn from a test distribution $p_{te}(\mathbf{x}, y) = p_{te}(\mathbf{x})p_{te}(y|\mathbf{x})$.

Let \mathcal{H} be a hypothesis class. The goal of supervised learning is to obtain a hypothesis $h : \mathcal{X} \rightarrow \mathbb{R}$ ($h \in \mathcal{H}$) with the training examples that minimizes the expected loss over the test distribution:

$$\mathcal{R}(h) := \mathbb{E}_{(\mathbf{x}^{te}, y^{te}) \sim p_{te}(\mathbf{x}, y)} \left[\ell(h(\mathbf{x}^{te}), y^{te}) \right], \quad (1)$$

where $\ell : \mathbb{R} \times \mathcal{Y} \rightarrow \mathbb{R}$ is the loss function that measures the discrepancy between the true output value y and the predicted value $\hat{y} := h(\mathbf{x})$. In this paper, we assume that ℓ is bounded from above, i.e., $\ell(y, y') < \infty$ ($\forall y, y' \in \mathcal{Y}$).

Definition 2.1 (Covariate shift assumption) We consider that the two distributions $p_{tr}(\mathbf{x}, y)$ and $p_{te}(\mathbf{x}, y)$ satisfy the covariate shift assumption if the following three conditions hold: 1) $p_{tr}(\mathbf{x}) \neq p_{te}(\mathbf{x})$, 2) $\text{supp}(p_{tr}(\mathbf{x})) \supset \text{supp}(p_{te}(\mathbf{x}))$ and 3) $p_{tr}(y|\mathbf{x}) = p_{te}(y|\mathbf{x})$.

Under the covariate shift assumption, the goal of covariate shift adaptation is still to obtain a hypothesis h that minimizes the expected loss (1) by utilizing both labeled training examples $\{(\mathbf{x}_i^{tr}, y_i^{tr})\}_{i=1}^{n_{tr}}$ and unlabeled test examples $\{(\mathbf{x}_i^{te})\}_{i=1}^{n_{te}}$.

2.2 Previous works

Ordinary empirical risk minimization (ERM) (Vapnik, 1998, 2013), a standard approach in supervised learning, may fail under the covariate shift because it assumes that the training and test data follow the same distribution. Importance weighting has been shown to be effective in mitigating the effect of covariate shift (Shimodaira, 2000; Sugiyama and Müller, 2005b; Sugiyama et al., 2007; Zadrozny, 2004):

$$\min_{h \in \mathcal{H}} \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} w(\mathbf{x}_i^{tr}) \ell(h(\mathbf{x}_i^{tr}), y_i^{tr}), \quad (2)$$

where $w : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ is a certain weighting function.

Definition 2.2 (IWERM (Shimodaira, 2000)) If we choose the density ratio $p_{te}(\mathbf{x})/p_{tr}(\mathbf{x})$ as the weighting function, ERM according to

$$\min_{h \in \mathcal{H}} \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \frac{p_{te}(\mathbf{x}_i^{tr})}{p_{tr}(\mathbf{x}_i^{tr})} \ell(h(\mathbf{x}_i^{tr}), y_i^{tr}) \quad (3)$$

has consistency.

This is called importance weighted ERM (IWERM). However, IWERM tends to produce an estimator with high variance. We can reduce the variance by flattening the importance weights, which is called adaptive IWERM (AIWERM):

Definition 2.3 (AIWERM (Shimodaira, 2000)) Let $\lambda \in [0, 1]$. If we choose $(p_{te}(\mathbf{x})/p_{tr}(\mathbf{x}))^\lambda$ as the weighting function, we can obtain the variance-reduced estimator:

$$\min_{h \in \mathcal{H}} \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \left(\frac{p_{te}(\mathbf{x}_i^{tr})}{p_{tr}(\mathbf{x}_i^{tr})} \right)^\lambda \ell(h(\mathbf{x}_i^{tr}), y_i^{tr}). \quad (4)$$

Relative IWERM (RIWERM), a stable version of AIWERM, has also been proposed:

Definition 2.4 (RIWERM (Yamada et al., 2011)) *Let $\lambda \in [0, 1]$. If we choose $p_{te}(\mathbf{x})/\lambda p_{tr}(\mathbf{x}) + (1 - \lambda)p_{te}(\mathbf{x})$ as the weighting function, we can directly estimate a flattened version of the importance weight:*

$$\min_{h \in \mathcal{H}} \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \frac{p_{te}(\mathbf{x}_i^{tr})}{\lambda p_{tr}(\mathbf{x}_i^{tr}) + (1 - \lambda)p_{te}(\mathbf{x}_i^{tr})} \ell(h(\mathbf{x}_i^{tr}), y_i^{tr}). \quad (5)$$

All of the above methods are considered as different weighting methods for each point of the training data. More generally, the method of covariate shift adaptation can be essentially rephrased as a weighting strategy for training data.

3 Statistical Model and Exponential Family

Information geometry (Amari, 1985; Amari and Nagaoka, 2007) is a powerful framework that allows us to deal with statistical models on Riemannian manifolds. For theoretical investigation, we need the notion of dual connection and curvature tensor associated with Fisher metric, but these details are deferred to the Appendix 6 and we here present minimum required definitions and notations. We note that the assumption on the parametric family is only required for the information geometric analysis in Section 4.2. The algorithmic framework of the proposed method is independent of the parametric model.

Since $p_{tr}(y|\mathbf{x}) = p_{te}(y|\mathbf{x}) = p(y|\mathbf{x})$ from the assumption of Definition 2.1, what we are interested in is the model manifold $(\mathcal{M}, g(\boldsymbol{\theta}))$ to which the marginal distribution $p(\mathbf{x}; \boldsymbol{\theta})$ belongs:

$$\mathcal{M} = \left\{ p(\mathbf{x}; \boldsymbol{\theta}) ; \boldsymbol{\theta} \in \Theta \right\}. \quad (6)$$

Here, $p_{tr}(\mathbf{x}; \boldsymbol{\theta}), p_{te}(\mathbf{x}; \boldsymbol{\theta}) \in \mathcal{M}$. We note that elements in \mathcal{M} is specified by its parameter $\boldsymbol{\theta}$ and we identify the parameter vector $\boldsymbol{\theta}$ to the density function $p(\mathbf{x}; \boldsymbol{\theta})$ and write $p(\mathbf{x}; \boldsymbol{\theta}) \simeq \boldsymbol{\theta}$ if necessary. In this paper, we assume that \mathcal{M} is an exponential family and the probability density function can be written as

$$p(\mathbf{x}; \boldsymbol{\theta}) = \exp \left\{ \theta^i T_i(\mathbf{x}) + k(\mathbf{x}) - \psi(\boldsymbol{\theta}) \right\}, \quad (7)$$

where \mathbf{x} is a random variable, $\boldsymbol{\theta} = (\theta^1, \dots, \theta^p)$ is an p -dimensional vector parameter to specify a distribution, $\mathbf{T}(\mathbf{x}) = (T_1(\mathbf{x}), \dots, T_p(\mathbf{x}))$ are sufficient statistics of \mathbf{x} , $k(\mathbf{x})$ is a function of \mathbf{x} and ψ corresponds to the normalization factor. In Eq. (7), and hereafter the Einstein summation convention will be assumed, so that summation will be automatically taken over indices repeated twice in the term, e.g., $\mathbf{a}^i \mathbf{b}_i = \sum_i \mathbf{a}^i \mathbf{b}_i$.

In the exponential family, the natural parameter $\boldsymbol{\theta}$ forms the affine coordinate system, i.e.,

$$\boldsymbol{\theta}(t) = (1-t)\boldsymbol{\theta}_1 + t\boldsymbol{\theta}_2 \quad (\forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta, \forall t \in [0, 1]) \quad (8)$$

is a geodesic on \mathcal{M} . As a dual coordinate of $\boldsymbol{\theta}$, the expectation parameter $\boldsymbol{\eta}$ is defined by the Legendre transformation

$$\begin{aligned} \boldsymbol{\eta} &= \nabla \psi(\boldsymbol{\theta}), \quad \boldsymbol{\theta} = \nabla \varphi(\boldsymbol{\eta}), \\ \text{where } \varphi(\boldsymbol{\eta}) &= \max_{\boldsymbol{\theta}'} \left\{ \boldsymbol{\theta}' \cdot \boldsymbol{\eta} - \psi(\boldsymbol{\theta}') \right\}. \end{aligned}$$

Existing weights for covariate shift adaptation are geometrically characterized, then a generalized weight function is designed based on this geometric formulation.

4 Geometrical Generalization of Covariate Shift Adaptation

4.1 Information Geometrically Generalized IWERM

In order to derive a generalized covariate shift adaptation method, we prepare the following function.

Definition 4.1 (*f*-interpolation (Kimura and Hino, 2021)) *For any $a, b, \in \mathbb{R}$, some $\lambda \in [0, 1]$ and some $\alpha \in \mathbb{R}$, we define *f*-interpolation as*

$$m_f^{(\lambda, \alpha)}(a, b) = f_\alpha^{-1} \left\{ (1-\lambda)f_\alpha(a) + \lambda f_\alpha(b) \right\}, \quad (9)$$

where

$$f_\alpha(a) = \begin{cases} a^{\frac{1-\alpha}{2}} & (\alpha \neq 1) \\ \log a & (\alpha = 1) \end{cases} \quad (10)$$

is the function that defines the *f*-mean (Hardy et al., 1952).

We can easily see that this family includes various known weighted means including the e -mixture and m -mixture for $\alpha = \pm 1$ in the literature of information geometry (Amari, 2016):

$$\begin{aligned} m_f^{(\lambda,1)}(a, b) &= \exp\{(1 - \lambda) \log a + \lambda \log b\}, \\ m_f^{(\lambda,-1)}(a, b) &= (1 - \lambda)a + \lambda b, \\ m_f^{(\lambda,0)}(a, b) &= \left((1 - \lambda)\sqrt{a} + \lambda\sqrt{b} \right)^2, \\ m_f^{(\lambda,3)}(a, b) &= \frac{1}{(1 - \lambda)\frac{1}{a} + \lambda\frac{1}{b}}. \end{aligned}$$

Also, for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ ($d > 0$), we write

$$\mathbf{m} = m_f^{(\lambda,\alpha)}(\mathbf{u}, \mathbf{v}), \text{ where } \mathbf{m}_i = m_f^{(\lambda,\alpha)}(\mathbf{u}_i, \mathbf{v}_i).$$

Using this function, we generalize the existing methods of covariate shift adaptation.

Lemma 4.1 (f -representation of AIWERM) *The marginal positive measures generated by the weighting of AIWERM can be expressed by using the f -interpolation function as*

$$p_A^{(\lambda)}(\mathbf{x}) = m_f^{(\lambda,1)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x})). \quad (11)$$

Proof 1 *From Eq.(4), we consider its expectation as*

$$\begin{aligned} \hat{h} &= \min_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} \left(\frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})} \right)^\lambda \ell(h(\mathbf{x}), y) p_{tr}(\mathbf{x}, y) d\mathbf{x}dy \\ &= \min_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} \ell(h(\mathbf{x}), y) p_A^{(\lambda)}(\mathbf{x}) p_{tr}(y|\mathbf{x}) d\mathbf{x}dy. \end{aligned}$$

Here,

$$\begin{aligned} p_A^{(\lambda)}(\mathbf{x}) &= \left(\frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})} \right)^\lambda p_{tr}(\mathbf{x}) \\ \log p_A^{(\lambda)}(\mathbf{x}) &= \alpha(\log p_{te}(\mathbf{x}) - \log p_{tr}(\mathbf{x})) + \log p_{tr}(\mathbf{x}) \\ &= (1 - \lambda) \log p_{tr}(\mathbf{x}) + \lambda \log p_{te}(\mathbf{x}) \\ p_A^{(\lambda)}(\mathbf{x}) &= \exp\{(1 - \lambda) \log p_{tr}(\mathbf{x}) + \lambda \log p_{te}(\mathbf{x})\} \\ &= m_f^{(\lambda,1)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x})). \end{aligned}$$

Lemma 4.2 (*f*-representation of RIWERM) *The marginal positive measures generated by the weighting of RIWERM can be expressed by using the *f*-interpolation function as*

$$p_R^{(\lambda)}(\mathbf{x}) = m_f^{(\lambda,3)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x})). \quad (12)$$

Proof 2 *From Eq. (5),*

$$\begin{aligned} p_R^{(\lambda)}(\mathbf{x}) &= \frac{p_{te}(\mathbf{x})p_{tr}(\mathbf{x})}{\lambda p_{tr}(\mathbf{x}) + (1 - \lambda)p_{te}(\mathbf{x})} \\ &= \frac{1}{\lambda \frac{1}{p_{te}(\mathbf{x})} + (1 - \lambda) \frac{1}{p_{tr}(\mathbf{x})}} = m_f^{(\lambda,3)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x})). \end{aligned}$$

From the above discussion, the following generalized method of covariate shift adaptation is derived using the *f*-representation.

Theorem 4.1 (Geometrically generalized IWERM) *For $\lambda \in [0, 1]$ and $\alpha \in \mathbb{R}$, AIW-ERM and RIWERM is generalized as*

$$\hat{h} = \min_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} w^{(\lambda, \alpha)}(\mathbf{x}) \ell(h(\mathbf{x}), y) p_{tr}(\mathbf{x}, y) d\mathbf{x} dy, \quad (13)$$

where

$$w^{(\lambda, \alpha)}(\mathbf{x}) = \frac{m_f^{(\lambda, \alpha)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x}))}{p_{tr}(\mathbf{x})}. \quad (14)$$

See the Appendix 6 for the proof. From Definition 4.1, we can confirm that

$$\begin{aligned} m_f^{(0, \alpha)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x})) &= p_{tr}(\mathbf{x}), \quad \text{and} \\ m_f^{(1, \alpha)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x})) &= p_{te}(\mathbf{x}), \end{aligned}$$

for all $\alpha \in \mathbb{R}$, and this means that we can obtain the set of all curves that connect $p_{tr}(\mathbf{x})$ and $p_{te}(\mathbf{x})$.

We note that Zhang et al. (2021) proposed a method based on basis expansion to estimate a flexible importance weight. It is similar to our proposal in the sense that improves the degree of freedom for designing the importance weight. However, our method considers the parametric form of weight, which enables us to achieve information geometric insight.

In many studies of covariate shift problems using the density ratio weighting including Yamada et al. (2011), the direct estimation of the density ratio is often employed (Sugiyama et al., 2012). Our proposed weight function in (14) is also represented as density ratio:

$$\begin{aligned} w^{(\lambda, \alpha)}(\mathbf{x}) &= \frac{\left[(1 - \lambda) p_{tr}(\mathbf{x})^{\frac{1-\alpha}{2}} + \lambda p_{te}(\mathbf{x})^{\frac{1-\alpha}{2}} \right]^{\frac{2}{1-\alpha}}}{p_{tr}(\mathbf{x})} \\ &= \left[1 - \lambda + \lambda \left(\frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})} \right)^{\frac{1-\alpha}{2}} \right]^{\frac{2}{1-\alpha}}, \quad (\alpha \neq 1). \end{aligned}$$

It is then also possible to apply the direct estimation of the density ratio using, for example kernel expansion. In our implementation, we simply used the given $p_{tr}(\mathbf{x})$ and $p_{te}(\mathbf{x})$ separately because they are explicitly known by the construction of the training and the test datasets as explained in Section 5.1. In the practical application of the proposed method in which the generative processes of the covariates of training and test data are unknown, direct density estimation would be a promising approach.

4.2 Geometric Bias

AIWERM and RIWERM connects two distributions p_{tr} and p_{te} in different ways. Statistical bias and variance of IWERM, AIWERM, and RIWERM are discussed in the respective papers. In this subsection, we study the geometric bias of these methods to have a deeper understanding of these methods from the geometric viewpoint.

The proposed generalization of IWERM is independent from a specific parametrization of density functions. In this subsection, for theoretical treatment, the exponential model manifold which contains $p_{tr}(\mathbf{x}; \boldsymbol{\theta})$ and $p_{te}(\mathbf{x}; \boldsymbol{\theta})$ are considered, hence geodesics can be described by a linear combination of parameters as explained in Appendix 6. With this assumption, specifying λ and α is equivalent to selecting a point on the geodesic connecting p_{tr} and p_{te} .

Definition 4.2 (α -divergence (Amari, 1985)) *Let α be a real parameter. The α -divergence between two probability vectors \mathbf{p} and \mathbf{q} is defined as*

$$D_\alpha[\mathbf{p} : \mathbf{q}] = \frac{4}{1 - \alpha^2} \left(1 - \sum_i p_i^{\frac{1-\alpha}{2}} q_i^{\frac{1+\alpha}{2}} \right). \quad (15)$$

Definition 4.3 (α -representation (Amari, 2009)) For some positive measure $m_i^{\frac{1-\alpha}{2}}$, the coordinate system $\theta = (\theta^i)$ derived from the α -divergence is $\theta^i = m_i^{\frac{1-\alpha}{2}} = f_\alpha(m_i)$ and denote by θ^i the α -representation of a positive measure $m_i^{\frac{1-\alpha}{2}}$.

Definition 4.4 (α -geodesic (Amari, 2016)) The α -geodesic connecting two probability vectors $p(\mathbf{x})$ and $q(\mathbf{x})$ is defined as

$$\begin{aligned} r_i(\lambda) &= c(\lambda) f_\alpha^{-1} \left\{ (1 - \lambda) f_\alpha(p(x_i)) + \lambda f_\alpha(q(x_i)) \right\}, \\ c(\lambda) &= \left(\sum_{i=1}^p r_i(\lambda) \right)^{-1}. \end{aligned} \quad (16)$$

Let $\psi_\alpha(\theta) = \frac{1-\alpha}{2} \sum_{i=1}^p m_i$, the dual coordinate system η is given by $\eta = \nabla \psi_\alpha(\theta)$ as

$$\eta_i = (\theta^i)^{\frac{1+\alpha}{1-\alpha}} = f_{-\alpha}(m_i), \quad (17)$$

which is the $-\alpha$ -representation of m_i .

From Definitions 4.1 and 4.4, we see that f -interpolation is the unnormalized version of the α -geodesic. We write $\tilde{m}_f^{(\lambda, \alpha)}$ for a suitably normalized f -interpolation. The important properties of α -geodesics are

- the α -geodesic is a geodesic in the α -coordinate system derived from α -divergence,
- the $-\alpha$ -geodesic is linear in the $-\alpha$ -representation.

Let γ_c be the geodesic connecting two distributions parameterized by θ_{tr} and θ_{te} . Now, we define two types of geometric biases to characterize the dispersion of θ_{tr} from θ_{te} with respect to the direction along the α -geodesic and to the direction orthogonal to the α -geodesic.

Definition 4.5 (Geodesic bias and curvature bias) If we write the unit vector along the α -geodesic direction as e_1 and any unit vector in the orthogonal direction to e_1 as e_2 , the bias relative to the test distribution due to weighting can be decomposed as follows:

- geodesic bias: $b_g = (1 - \lambda)e_1$,
- curvature bias: $b_c = (1 - \lambda)tr_g(\text{Ric})e_2$,

Algorithm 1 Bayesian optimization for IGIWERM

Input: acquisition function $a(\lambda, \alpha|D)$, target function $L(h; \lambda, \alpha)$, initial points D_{init} compose of a set of parameters $\Xi = \{(\lambda, \alpha)\}$ and corresponding values of the target function

Output: (λ^*, α^*) that minimizes $\min_{h \in \mathcal{H}} L(h; \lambda, \alpha)$

Initialize $D = D_{init}$

while Not converge **do**

$$\hat{\lambda}, \hat{\alpha} = \arg \min_{\lambda, \alpha} a(\lambda, \alpha|D), \quad \Xi = \Xi \cup \{(\hat{\lambda}, \hat{\alpha})\}$$

$$\hat{e} = L(h; \hat{\lambda}, \hat{\alpha}), \quad D = D \cup \{(\hat{\lambda}, \hat{\alpha}, \hat{e})\}$$

end while

$$(\lambda^*, \alpha^*) = \arg \min_{(\lambda, \alpha) \in \Xi} \{\min_{h \in \mathcal{H}} L(h; \lambda, \alpha)\}$$

where tr_g is the trace operation on the metric tensor g and Ric is the Ricci curvature of the curve connecting the two points generated by the weighting:

$$Ric = R_{ikj} d\theta^i \otimes d\theta^j. \quad (18)$$

Here, R_{ikj} is the Riemannian curvature tensor.

For more detail on the geometric concepts, see textbooks on Riemannian manifolds (Jost, 2017). This definition of geometric biases is consistent with the fact that IWERM, which corresponds to $\lambda = 1$, leads to an unbiased estimator of the risk in the test dataset.

Proposition 4.2 For AIWERM, the geometric bias $b_A(\lambda)$ is computed as

$$b_A(\lambda) = (1 - \lambda)e_1. \quad (19)$$

Proposition 4.3 For RIWERM, the geometric bias $b_R(\lambda)$ is computed as

$$b_R(\lambda) = (1 - \lambda) \left\{ e_1 + tr_g \left(-4\Lambda_{ikj} d\theta^i \otimes d\theta^j \right) e_2 \right\}. \quad (20)$$

Here, Λ is a tensor that depends on the connection.

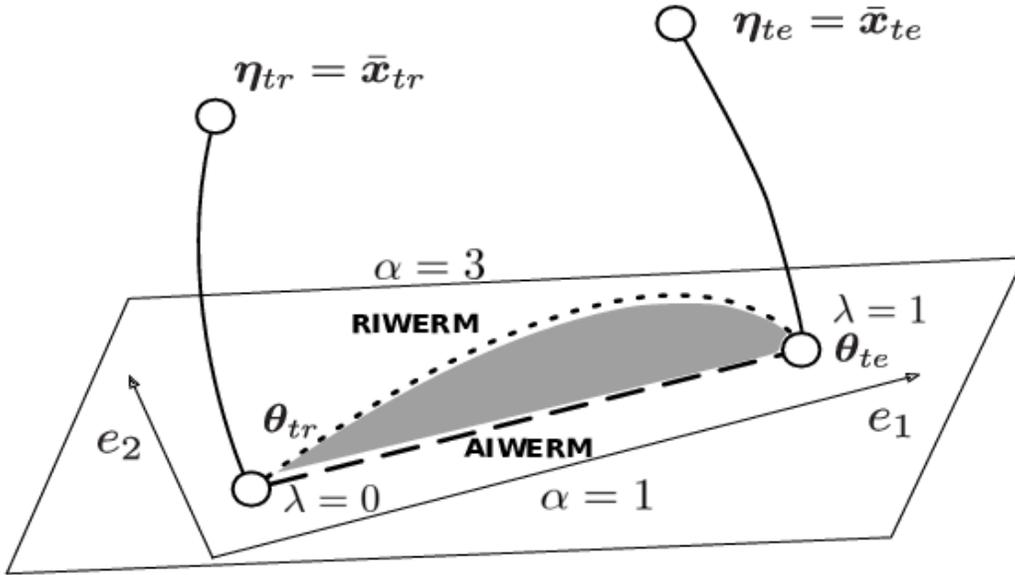


Figure 1: Geometry of covariate shift adaptation methods. In the θ -coordinate system, the dashed line corresponds to AIWERM and the dotted line corresponds to RIWERM. We write unit vector along the α -geodesic direction as e_1 and any unit vector in the orthogonal direction to e_1 as e_2 . Here, $\lambda = 0$ and $\lambda = 1$ correspond to θ_{tr} (ERM) and θ_{te} (IWERM), respectively, and $\alpha = 1$ and $\alpha = 3$ correspond to the AIWERM and RIWERM curves in the figure.

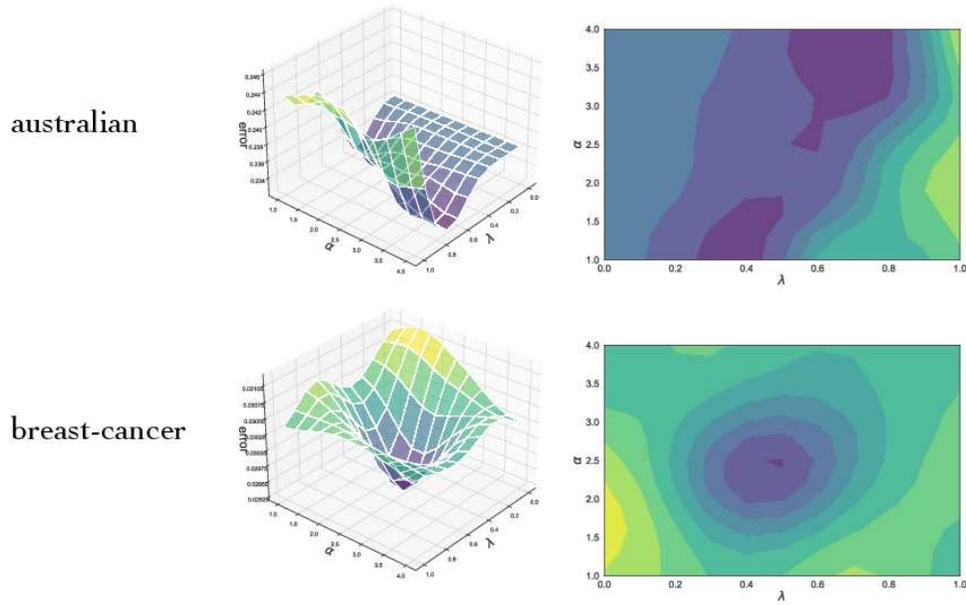


Figure 2: Visualization of grid search for α and λ on LIBSVM dataset.

These propositions are proved by straightforward calculation as detailed in Appendix 6

Figure 1 shows the curves on the manifolds created by AIWERM and RIWERM. Both of them satisfy

- for $\lambda = 0$, it is equivalent to unweighted ERM,
- for $\lambda = 1$, it is equivalent to IWERM.

Note that the curvature bias b_c vanishes for all $\lambda \in [0, 1]$ in AIWERM, while RIWERM does not guarantee the vanishing of the curvature bias for $\lambda \in (0, 1)$.

Intuitively, the geometric bias reveals in which direction the two parameters are misaligned. IWERM, which corresponds to AIWERM and RIWERM with $\lambda = 1$, is optimal when the sample size is large enough, but in real problems with limited sample size, it is often desirable to adopt a point between θ_{tr} and θ_{te} . AIWERM and RIWERM consider distinct curves and specify a point on them by the parameter λ . Our geometric analysis revealed that these curves are included in the set of curves represented by dual f -representation of the parameter coordinate system, and the geometric biases of these particular cases (AIWERM and RIWERM) are identified. The results presented in this subsection do not claim superiority of a particular method and are of importance in their

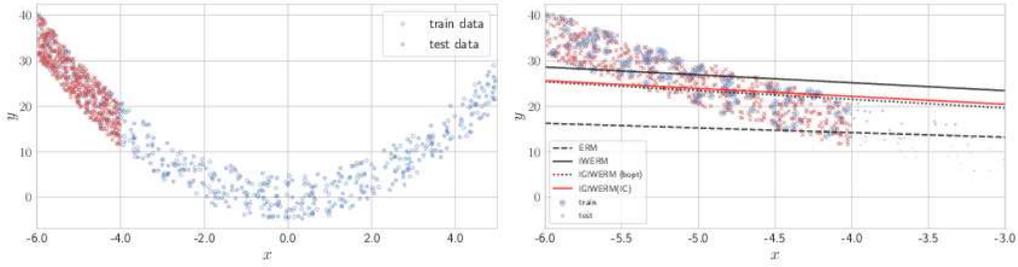


Figure 3: Left: generated data from $y = x^2 + \varepsilon$. We see that $p_{tr}(x)$ and $p_{te}(x)$ are different. Right: results of fitting by ERM, IWERM, and IGIWERM.

Table 1: Mean squared errors of covariate shift adaptation methods in regression problems over 10 trials. Here, IGIWERM (bopt) is the Bayesian optimization based, and IGIWERM (IC) is the information criterion based strategy.

Weighting strategy	MSE
ERM	160.19(± 4.25)
IWERM	33.76(± 3.82)
AIWERM	31.14(± 2.97)
RIWERM	30.03(± 2.74)
IGIWERM (bopt)	28.89(± 2.42)
IGIWERM (IC)	28.38(± 2.12)

own right as a geometric analysis of the covariate shift method.

4.3 Optimization of the generalized IWERM

The existing covariate shift adaptation methods described above can be regarded as having determined a good “weighting direction” in some sense in advance and then the “weighting magnitude” is adjusted according to the parameter λ . This approach is very convenient in terms of computational efficiency since the only optimized parameter is $\lambda \in [0, 1]$.

However, geometrically, these methods only consider certain curves on the manifold as candidate solutions, as can be seen from Figure 1, which means that the solution

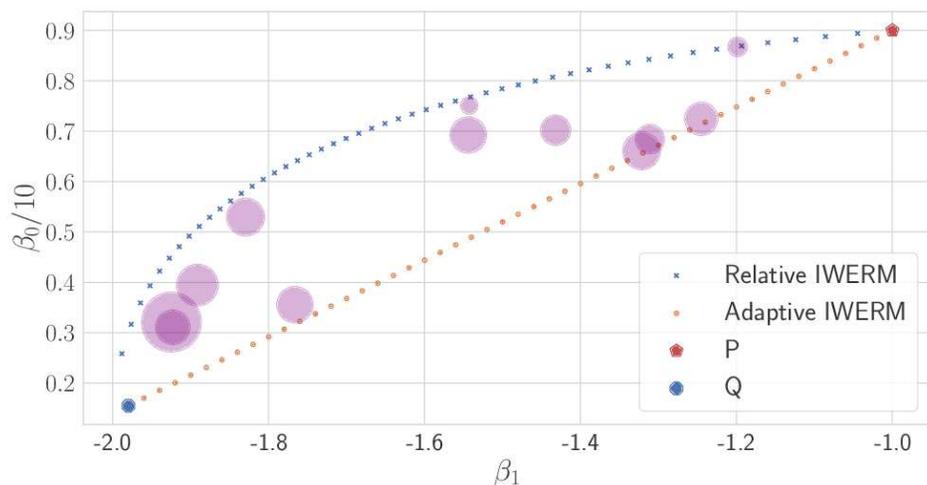


Figure 4: Bayesian optimization for IGIWERM. The coordinates of the purple circles are the parameters explored by Bayesian optimization, and the size of the purple circles indicates the goodness of the parameters (inverse of the MSE).

space is very small.

Our information geometrical IWERM (IGIWERM) can handle all curves $\gamma_\alpha(\lambda)$ in $\Pi_{(p_{tr}, p_{te})}$ that connect $p_{tr}(\mathbf{x})$ and $p_{te}(\mathbf{x})$, by adding only one parameter. For example, by setting $\alpha \in [1, 3]$, shaded area in Figure 1 can be used as the solution space. The problem of how to determine λ and α remains.

4.1 Information criterion

When the predictive model is of a simple parametric form, information criterion derived in (Shimodaira, 2000) is available (see appendix of (Shimodaira, 2000) for the proof.):

Theorem 4.4 (Information criterion for IGIWERM) *Let the information criterion for IWERM be*

$$IC_{GW} := -2L_1(\hat{\boldsymbol{\theta}}) + 2tr(J_w H_w^{-1}), \quad (21)$$

Table 2: Mean misclassification rates averaged over 10 trails on LIBSVM benchmark datasets. The numbers in the brackets are the standard deviations. For the methods with (optimal), the optimal parameters for the test data are obtained by linear search.

Dataset	#features	#data	unweighted	IWERM	AIWERM (optimal)	RIWERM (optimal)	ours
australian	14	690	33.46(±23.65)	22.13(±3.37)	21.98(±3.36)	21.73(±3.82)	18.85(±3.99)
breast-cancer	10	683	38.28(±10.98)	41.23(±15.39)	36.41(±9.68)	36.13(±10.81)	31.65(±8.49)
heart	13	270	45.17(±6.98)	39.94(±8.55)	39.76(±8.49)	39.76(±8.92)	35.37(±6.84)
diabetes	8	768	33.19(±5.69)	37.22(±6.63)	33.11(±6.45)	33.38(±5.74)	32.83(±5.62)
madelon	500	2,000	47.78(±1.53)	47.28(±2.20)	47.10(±2.13)	47.12(±1.65)	46.56(±2.12)

where $L_1(\boldsymbol{\theta}) = \sum_{i=1}^{n_{tr}} dr(\mathbf{x}_i^{tr}) \log p(y_i^{tr} | \mathbf{x}_i^{tr}, \boldsymbol{\theta})$, $dr(\mathbf{x}) = \frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})}$ and

$$J_w = -\mathbb{E}_{p_{tr}} \left[\left. \frac{dr(\mathbf{x}) \frac{\partial \log p(y|\mathbf{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta}_w^*} \times \frac{\partial \left(\frac{m_f^{\lambda, \alpha}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x}))}{p_{tr}(\mathbf{x})} \log p(y|\mathbf{x}, \boldsymbol{\theta}) \right)}{\partial \boldsymbol{\theta}'} \right|_{\boldsymbol{\theta}_w^*} \right]$$

$$H_w = \mathbb{E}_{p_{tr}} \left[\frac{\partial^2 \left(\frac{m_f^{\lambda, \alpha}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x}))}{p_{tr}(\mathbf{x})} \log p(y|\mathbf{x}, \boldsymbol{\theta}) \right)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \right].$$

Here, $\boldsymbol{\theta}_w^*$ is the minimizer of the weighted empirical risk. The matrices J_w and H_w may be replaced by their consistent estimates. Then, $IC_{GW}/2n$ is an unbiased estimator of the expected loss up to $O(n^{-1})$ term:

$$\mathbb{E}_{p_{tr}} [IC_{GW}/2n] = \mathbb{E}_{p_{tr}} [\ell_1(\hat{\boldsymbol{\theta}}_w)] + o(n^{-1}). \quad (22)$$

4.2 Bayesian optimization

This information criterion does not work for complicated nonparametric models. As a method that can be applied in general situations, we consider using Bayesian optimization Snoek et al. (2012); Frazier (2018) to find the optimal weighting by IGIWERM. Bayesian optimization assumes that the target function is drawn from a prior distribution over functions, typically a Gaussian process, updating a posterior as we observe the target function value in new places. We use the validation loss as the target function:

$$L(h; \lambda, \alpha) = \frac{1}{n_{val}} \sum_{i=1}^{n_{val}} \frac{p_{te}(\mathbf{x}_i^{val})}{p_{tr}(\mathbf{x}_i^{val})} \ell(h_{\lambda, \alpha}(\mathbf{x}_i^{val}), y_i^{val}). \quad (23)$$

where n_{val} is the validation sample size and $h_{\lambda, \alpha}$ is given by IWERM with λ and α . This validation procedure is based on the importance weighted cross validation used

in (Sugiyama et al., 2007). In Bayesian optimization, an acquisition function $a(\lambda, \alpha|D)$ is used for measuring goodness of candidate point (λ, α) based on current dataset D . As the acquisition function, we adopt the expected improvement Mockus et al. (1978); Jones et al. (1998). In this strategy, we choose the next query point which has the highest expected improvement over the current minimum target value. See Appendix 6 for more detail. The overall picture is summarized in Algorithm 1.

4.4 Learning guarantee

Generalization bounds of weighted maximum likelihood estimator for the target domain are derived in Cortes et al. (2010), and our weight function (14) is compatible with their bound. The weight defined by Eq. (14) is bounded when $\alpha \neq 1$ and achieves a standard rate $O(n_{tr}^{-1/2})$. When $\alpha = 1$, the weight is unbounded and its rate is $O(n_{tr}^{-3/8})$. Details are shown in Appendix 6.

5 Numerical Experiments

In this section, we present experimental results of domain adaptation problems under covariate shift using both synthetic and real data¹. Since the main purpose of the experiments is to see the effect of our generalization of the importance weighted ERM and comparison to the proposed and conventional IWERM methods, in all experiments, we assume that p_{tr} and p_{te} are known as detailed in Section 5.1.

5.1 Induction of Covariate Shift

Since each dataset is composed of data points generated from independent and identical distributions, we need to artificially induce covariate shifts. We induce the covariate shift as follows Cortes et al. (2008):

1. As a preprocessing step, we perform Z-score standardization on all input data.

¹Source code to reproduce the results is available from <https://github.com/nocotan/IGIWERM>

2. Then, an example (\mathbf{x}, y) is assigned to the training dataset with probability $\exp(v)/(1 + \exp(v))$ and to the test dataset with probability $1/(1 + \exp(v))$, where $v = 16\mathbf{w}^T \mathbf{x}/\sigma$ with σ being the standard deviation of $\mathbf{w}^T \mathbf{x}$ determined by using the given dataset, and $\mathbf{w} \in \mathbb{R}^d$ is a given projection vector. Here, the projection vector \mathbf{w} is given randomly for each experimental process.

By this construction of the training and test datasets, p_{tr} and p_{te} are explicitly determined as

$$p_{tr}(\mathbf{x}) = \frac{\exp(16\mathbf{w}^T \mathbf{x}/\sigma)}{1 + \exp(16\mathbf{w}^T \mathbf{x}/\sigma)},$$

$$p_{te}(\mathbf{x}) = \frac{1}{1 + \exp(16\mathbf{w}^T \mathbf{x}/\sigma)},$$

when the projection vector $\mathbf{w} \in \mathbb{R}^d$ is given. Although density ratio estimation could be employed in our experiments, we assume that the distribution is known in order to compare the performance of the proposed method without relying on the accuracy of the density or density ratio estimation.

5.2 Illustrative Example in Regression

Here, we predict the response $y \in \mathbb{R}$ using ordinary linear regression: $y = \beta_0 + \beta_1 x + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, where $\mathcal{N}(a, b)$ denotes the normal distribution with mean a and variance b . In the numerical example below, we assume the true $p(y|x)$ given as $y = x^2 + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, 5)$. The $p_{tr}(x)$ and $p_{te}(x)$ of the covariate x are $x^{tr} \sim \mathcal{N}(0, 5)$, $x^{te} \sim \mathcal{N}(-5, 0.5)$. The training sample size is $n_{tr} = 1000$ and the test sample size is $n_{te} = 300$. The left-hand side of Fig. 3 shows the data to be generated. We can see that $p_{tr}(x) \neq p_{te}(x)$.

The right panel of Fig. 3 shows the results of fitting by unweighted ERM, IWERM, and IGIWERM. Here, the parameters of IGIWERM are explored by using Algorithm 1, as shown in Fig. 4. The coordinates of the purple circles are the parameters explored by Bayesian optimization, and the radius of the purple circles is proportional to the goodness $r(\beta)$ of the parameters (inverse of the MSE): $r(\beta) = (\frac{1}{n} \sum_{i=1}^n (y_i - h(x_i, \beta))^2)^{-1}$. By choosing the size $r(\beta)$ of the plot for each point in this manner, the better-evaluated parameters can be plotted in larger circles. From this figure, it can be seen that our

generalized weighting is not restricted to lying just on two curves corresponding to AIWERM and RIWERM.

For the normal linear regression, the information criterion (21) is calculated from

$$IC_{GW}(\lambda, \alpha) = \frac{1}{2} \sum_{i=1}^{n_{tr}} d_r(\mathbf{x}_i^{tr}) \left\{ \frac{\hat{\epsilon}_1^2}{\hat{\sigma}^2 + \log(2\pi\hat{\sigma}^2)} \right\} + \sum_{i=1}^{n_{tr}} d_r(\mathbf{x}_i^{tr}) \left\{ \frac{\hat{\epsilon}_i^2}{\hat{\sigma}^2} \hat{h}_i + \frac{m_f^{\lambda, \alpha}(p_{tr}(\mathbf{x}^{tr}), p_{te}(\mathbf{x}^{tr}))}{2\hat{c}_w p_{tr}(\mathbf{x}^{tr})} \left(\frac{\hat{\epsilon}_i^2}{\hat{\sigma}^2} - 1 \right)^2 \right\}.$$

Here, $\hat{c}_w = \sum_{i=1}^n \frac{m_f^{\lambda, \alpha}(p_{tr}(x_i^{tr}), p_{te}(x_i^{tr}))}{p_{tr}(x_i^{tr})}$, $\hat{\sigma}^2 = \sum_{i=1}^n \frac{m_f^{\lambda, \alpha}(p_{tr}(x_i^{tr}), p_{te}(x_i^{tr}))}{p_{tr}(x_i^{tr})} \hat{\epsilon}_i^2 / \hat{c}_w$ and $\hat{\epsilon}_i$ is the residual. Table 1 shows that the IGIWERM outperforms existing methods.

5.3 Experiments on binary classification problem

We show the results of our experiments on the LIBSVM dataset².

In the experiments, we randomly generate a mapping vector \mathbf{w} for each trial and perform 10 trials for each dataset. We use SVM with Radial Basis Function (RBF) kernel as the base classifier. In this experiment, the parameters λ of AIWERM and RIWERM are chosen optimally by linear search using the test data. The experimental results on benchmark datasets are summarized in Table 2. The table shows that the proposed IGIWERM outperforms the conventional methods even when the parameters of those methods are optimized by using the test dataset. More experimental results on other datasets with various models are reported in Appendix 6.

5.4 Computational Cost

Here, we investigate the computational cost of our IGIWERM. The experimental setup is the same as in Section 5.2. The mean and standard deviation of the computation time obtained in the 10 trials are shown in Table 3. From this table, we can see that our IGIWERM takes constant times longer to compute than the vanilla ERM.

²<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

Table 3: Computational cost of ERM and IGIWERM.

Method	Computation time [sec]
ERM	1.130(± 0.238)
IGIWERM	9.887(± 0.845)

6 Conclusion and Discussion

We generalized existing methods of covariate shift adaptation in the geometrical framework. By our information geometrical formulation, geometric biases of conventional methods are elucidated. Unlike the dominant approaches restricted to a specific curve on a manifold in the literature, our generalization has a much larger solution space with only two parameters. Our experiments highlighted the advantage of our method over previous approaches, suggesting that our generalization can achieve better performance than the existing methods. A drawback of our proposed method is its relatively high computational cost for optimizing parameters α and λ . We used Bayesian optimization for efficient parameter search, and further sophisticated approaches would be explored in our future work.

As mentioned in the introduction, the importance weighting is used with deep neural network models (Fang et al., 2020), in which the importance weight in the feature representation obtained by DNN is considered. It is also worth mentioning that Sakai and Shimizu (2019) used RIWERM in the study of covariate shift on the learning from positive and unlabeled data. Our generalization will be applicable to their methods to improve the performance under a small sample regime. In particular, in a standard approach for optimizing the implicit weight function $w(\mathbf{x})$, it is common to add a regularization term $(w(\mathbf{x}) - p_{tr}(\mathbf{x})/p_{te}(\mathbf{x}))^2$ to the optimization objective. The use of the derived geodesic and curvature biases to regularize the optimal weight function will be investigated in connection with the modern weight learning approach using deep neural network models. Finally, the relation between geometric bias and statistical bias should be explored.

Acknowledgement

Part of this work is supported by JST CREST JPMJCR1761, JPMJCR2015, JSPS KAKENHI 17H01793, JP22H03653 and NEDO (JPNP18002). Finally, we express our special thanks to the anonymous reviewers whose valuable comments helped to improve the manuscript.

Appendix A: Statistical Manifolds and Straight Line in Exponential Family

Let M be a d -dimensional differentiable manifold with a Riemannian metric g . For each $\mathbf{x} \in M$, $T_{\mathbf{x}}M$ is its tangent space.

Definition 6.1 *Let $g_{\mathbf{x}}$ an inner product*

$$g_{\mathbf{x}} : T_{\mathbf{x}}(M) \times T_{\mathbf{x}}(M) \rightarrow \mathbb{R} \quad \forall \mathbf{x} \in M. \quad (24)$$

When, for any $X, Y \in M$, the map $\mathbf{x} \rightarrow g_{\mathbf{x}}(X_{\mathbf{x}}, Y_{\mathbf{x}})$ is differentiable with respect to $\mathbf{x} \in M$, $g_{\mathbf{x}}$ is denoted as the Riemannian metric.

The correspondence $X : M \ni \mathbf{x} \mapsto X_{\mathbf{x}} \in T_{\mathbf{x}}M$ is called a vector field on M . For $\mathbf{x} \in M$, let coordinate expression of $X_{\mathbf{x}}$ be $X_{\mathbf{x}} = (v^1(\mathbf{x}), \dots, v^d(\mathbf{x}))$. Then, $v^i(\mathbf{x})$ defines a real-valued function v^i on M and X is expressed as $X = (v^1, \dots, v^d)$. When a function on M is k times continuously differentiable, it is called the class C^k , and the set of all functions of class C^k on M is denoted as $C^k(M)$. A vector field X is called class C^k when all of v^i , $i = 1, \dots, d$ are class C^k . The set of all class C^∞ vector fields is denoted as $\mathfrak{X}(M)$. A tangent space $T_{\mathbf{x}}(M)$ is a vector space spanned by differentials $\frac{\partial}{\partial x^i}$, namely,

$$T_{\mathbf{x}}(M) = \left\{ a^i \left(\frac{\partial}{\partial x^i} \right)_{\mathbf{x}} \mid \forall a_i \in \mathbb{R} \right\}. \quad (25)$$

Following the notational convention of differential geometry, we use $\partial_i = \frac{\partial}{\partial x^i}$ and the Einstein summation convention. The vector field on a manifold M is then written as

$$\mathfrak{X}(M) = \{ v^i \partial_i \mid v^i \in C^\infty(M) \}. \quad (26)$$

For $X \in \mathfrak{X}(M)$ and $f \in C^\infty(M)$, $fX \in \mathfrak{X}(M)$ is defined by $(fX)_x = f(x)X_x$, ($\mathbf{x} \in M$). Differential of a function f with respect to a vector field X is denoted as $Xf \in C^\infty(M)$ and defined by $(Xf)(\mathbf{x}) = X_x(f)$, ($\mathbf{x} \in M$). When two vector fields are expressed as $X = v^i \partial_i$ and $Y = u^i \partial_i$, we have

$$X(Yf) - Y(Xf) = (v^j \partial_j u^i - u^j \partial_j v^i) \partial_i f. \quad (27)$$

The commutator product of X and Y is defined as $[X, Y] \in \mathfrak{X}(M)$, $[X, Y]f = (XY - YX)f$, and

$$[X, Y] = (v^j \partial_j u^i - u^j \partial_j v^i) \partial_i. \quad (28)$$

Definition 6.2 Consider a map $\nabla : \mathfrak{X}(M) \times \mathfrak{X}(M) \rightarrow \mathfrak{X}(M)$ which assigns a pair of vectors $(X, Y) \in \mathfrak{X}(M) \times \mathfrak{X}(M)$ to a vector $\nabla_Y X \in \mathfrak{X}(M)$. $\nabla_Y X$ is called a covariant derivative of X with respect to Y , and ∇ is called an affine connection when the following conditions hold for any $X, Y, Z \in \mathfrak{X}(M)$ and $f \in C^\infty(M)$:

- $\nabla_{Y+Z} X = \nabla_Y X + \nabla_Z X$
- $\nabla_{fX} X = f \nabla_Y X$
- $\nabla_Z (X + Y) = \nabla_Z X + \nabla_Z Y$
- $\nabla_Y (fX) = (Yf)X + f \nabla_Y X$

Definition 6.3 Let ∇ be an affine connection on M , and define a map

$$T : \mathfrak{X}(M) \times \mathfrak{X}(M) \rightarrow \mathfrak{X}(M)$$

$$(X, Y) \mapsto T(X, Y) = \nabla_X Y - \nabla_Y X - [X, Y]. \quad (29)$$

The map T is called the torsion tensor field of ∇ . When $T = 0$ for all $X, Y \in \mathfrak{X}(M)$, the connection ∇ is called torsion-free.

For an affine connection, the Christoffel symbol $\Gamma_{ij}^k \in C^\infty(M)$ is defined by

$$\nabla_{\partial_i} \partial_j = \Gamma_{ij}^k \partial_k. \quad (30)$$

With this formula, the connection and the Christoffel symbol are often identified. The affine connection ∇ is torsion-free when and only when $\Gamma_{ij}^k = \Gamma_{ji}^k$.

Suppose a manifold M is equipped with a Riemannian metric g . When

$$Xg(Y, Z) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z) \quad (31)$$

holds for all $X, Y, Z \in \mathfrak{X}(M)$, the connection ∇ is called a metric connection. In general, an affine connection is not a metric connection, but there uniquely exists an affine connection ∇^* which satisfies

$$Xg(Y, Z) = g(\nabla_X Y, Z) + g(Y, \nabla_X^* Z). \quad (32)$$

The connection ∇^* is called the dual connection of ∇ .

Given a Riemannian metric g , another representation of the Christoffel symbol is given by

$$\Gamma_{ij,k} = g(\nabla_{\partial_i} \partial_j, \partial_k). \quad (33)$$

Definition 6.4 *When an affine connection ∇ is torsion-free and a metric connection with respect to the Riemannian metric g , it is called a Levi-Civita connection with respect to the metric g .*

In general, when a $(0, 3)$ -tensor \bar{T} is given in addition to an affine connection ∇ and a Riemannian metric g , an alternative connection $\tilde{\nabla}$ is defined as

$$g(\tilde{\nabla}_Y X, Z) = g(\nabla_Y X, Z) + \bar{T}(X, Y, Z). \quad (34)$$

Let Ω be a set for which probability measure is defined, and define a d -dimensional statistical model

$$S = \{p(\cdot; \boldsymbol{\xi}) \mid \boldsymbol{\xi} \in \Xi\}, \quad (35)$$

where the parameter space Ξ is isomorphic to \mathbb{R}^d . As a Riemannian metric associated with the statistical model S , we consider the Fisher metric defined as

$$g_{ij}(\boldsymbol{\xi}) = \mathbb{E}_{\boldsymbol{\xi}}[(\partial_i l_{\boldsymbol{\xi}})(\partial_j l_{\boldsymbol{\xi}})], \quad (36)$$

where $\mathbb{E}_{\boldsymbol{\xi}}[\cdot]$ is expectation with respect to a probability density $p(\cdot; \boldsymbol{\xi})$ and $l_{\boldsymbol{\xi}}(x) = \log p(x; \boldsymbol{\xi})$ ($x \in \Omega$) is the log-likelihood. Now, consider a $(0, 3)$ -tensor \bar{T} on S defined by

$$(\bar{T})_{ijk}(\boldsymbol{\xi}) = \sum_{x \in \Omega} (\partial_i l_{\boldsymbol{\xi}}(x)) (\partial_j l_{\boldsymbol{\xi}}(x)) (\partial_k l_{\boldsymbol{\xi}}(x)) p(x; \boldsymbol{\xi}), \quad (37)$$

and based on the Levi-Civita connection ∇ associated with the Fisher metric g on S , we define a affine connection $\nabla^{(\alpha)}$ by

$$g(\nabla_Y^{(\alpha)} X, Z) = g(\nabla_Y X, Z) - \frac{\alpha}{2} \bar{T}(X, Y, Z), \quad (X, Y, Z \in \mathfrak{X}(S)). \quad (38)$$

This connection is called the α -connection. The Christoffel symbols associated with connections ∇ and $\nabla^{(\alpha)}$ are

$$\begin{aligned} \Gamma_{ij,k} &= \mathbb{E}_{\xi} \left[\left\{ \partial_i \partial_j l_{\xi} + \frac{1}{2} (\partial_i l_{\xi}) (\partial_j l_{\xi}) \right\} (\partial_k l_{\xi}) \right], \\ \Gamma_{ij,k}^{(\alpha)} &= \mathbb{E}_{\xi} \left[\left\{ \partial_i \partial_j l_{\xi} + \frac{1-\alpha}{2} (\partial_i l_{\xi}) (\partial_j l_{\xi}) \right\} (\partial_k l_{\xi}) \right]. \end{aligned}$$

From $\Gamma_{ij,k}^{(\alpha)} = \Gamma_{ji,k}^{(\alpha)}$, the α -connection is torsion-free. Note that the dual connection of $\nabla^{(\alpha)}$ is $\nabla^{(-\alpha)}$, and it also holds that

$$\nabla^{(\alpha)} = \frac{1+\alpha}{2} \nabla^* + \frac{1-\alpha}{2} \nabla. \quad (39)$$

Definition 6.5 For an affine connection ∇ of a manifold M , a map

$$\begin{aligned} R : \mathfrak{X}(M) \times \mathfrak{X}(M) \times \mathfrak{X}(M) &\rightarrow \mathfrak{X}(M) \\ (X, Y, Z) &\mapsto R(X, Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X, Y]} Z \end{aligned}$$

is called the curvature tensor field of the connection ∇ .

The curvature tensor is expressed with coordinate and the Christoffel symbol as

$$R(\partial_i, \partial_j) \partial_k = (\partial_i \Gamma_{jk}^l - \partial_j \Gamma_{ik}^l) \partial_l + (\Gamma_{jk}^l \Gamma_{il}^m - \Gamma_{ik}^l \Gamma_{jl}^m) \partial_m. \quad (40)$$

Definition 6.6 When both the torsion and curvature are zero, the connection ∇ is said to be flat.

Let γ be a map from a close interval I to a manifold M . The map γ is parameterized by a real-valued parameter $t \in I$ as $\gamma(t)$ and called a curve on M . When the value of γ at two endpoints of I is fixed, the shortest path between these two points is defined by using the variational principle. The pararell shift of $\frac{d\gamma}{dt}$ along with γ is expressed as

$$\nabla_{\frac{d}{dt}} \frac{d\gamma}{dt} = \left(\frac{d^2 \gamma_k}{dt^2} + (\Gamma_{ij}^k \circ \gamma) \frac{d\gamma_i}{dt} \frac{d\gamma_j}{dt} \right) \partial_k. \quad (41)$$

Definition 6.7 *An equation*

$$\nabla_{\frac{d}{dt}} \frac{d\gamma}{dt} = \mathbf{0} \quad (42)$$

is called the geodesic equation, and the curve satisfying this equation is called a geodesic.

Note that if $\Gamma_{ij,k} = 0 \forall i, j, k$, the geodesic equation is of the form $\frac{d^2\gamma_k}{dt^2} = 0$, hence the geodesic is a straight line.

Definition 6.8 *Let S be a d -dimensional statistical model. When each element of the model in S is represented by*

$$p(x; \boldsymbol{\theta}) = \exp(k(x) + \theta^i F_i(x) - \psi(\boldsymbol{\theta})), \quad (43)$$

by using functions $k, F_1, \dots, F_d : \Omega \rightarrow \mathbb{R}$ and $\psi : \Theta \rightarrow \mathbb{R}$, the statistical model S is called an exponential family, and $\boldsymbol{\theta}$ is called the natural parameter of the model.

Note that in a general statistical model S , ξ , and Ξ are often used as its parameter and the parameter space, while for an exponential family, θ and Θ are often used to represent its parameter and the parameter space. Consider an exponential family with α connection $\nabla^{(\alpha)}$. The Christoffel symbols are

$$\Gamma_{ij,k}^{(\alpha)} = \mathbb{E}_{\boldsymbol{\theta}} \left[\left\{ \partial_i \partial_j l_{\boldsymbol{\theta}} + \frac{1-\alpha}{2} (\partial_i l_{\boldsymbol{\theta}}) (\partial_j l_{\boldsymbol{\theta}}) \right\} (\partial_k l_{\boldsymbol{\theta}}) \right], \quad (44)$$

and

$$\partial_i l_{\boldsymbol{\theta}} = F_i(x) - (\partial_i \psi)(\boldsymbol{\theta}), \quad (\partial_i \partial_j \psi)(\boldsymbol{\theta}). \quad (45)$$

So, when $\alpha = 1$, we have

$$\Gamma_{ij,k}^{(1)} = \mathbb{E}_{\boldsymbol{\theta}} [-(\partial_i \partial_j \psi)(\boldsymbol{\theta}) (\partial_k l_{\boldsymbol{\theta}})] = 0, \quad (46)$$

namely, the exponential family is flat with the Fisher metric and $\alpha = 1$ connection. This implies that in exponential family, for the $\alpha = 1$ -connection $\nabla^{(1)}$ associated with the Fisher metric, the geodesic between two points correspond to natural parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ is of the form $t\boldsymbol{\theta}_1 + (1-t)\boldsymbol{\theta}_2$.

Appendix B: Proofs of main results

Proof 3 (Derivation of the information geometrically generalized IWERM) Let h_A be a hypothesis generated by AIWERM. From Lemma 4.1, we can write

$$\begin{aligned}\hat{h}_A &= \min_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} \ell(h(\mathbf{x}), y) p_A^{(\lambda)}(\mathbf{x}) p_{tr}(y|\mathbf{x}) d\mathbf{x} dy \\ &= \min_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} \ell(h(\mathbf{x}), y) m_f^{(\lambda,1)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x})) p_{tr}(y|\mathbf{x}) d\mathbf{x} dy \\ &= \min_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} \ell(h(\mathbf{x}), y) \frac{m_f^{(\lambda,1)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x}))}{p_{tr}(\mathbf{x})} p_{tr}(\mathbf{x}, y) d\mathbf{x} dy.\end{aligned}\quad (47)$$

From Lemma 4.2, we also have

$$\hat{h}_R = \min_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} \ell(h(\mathbf{x}), y) \frac{m_f^{(\lambda,3)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x}))}{p_{tr}(\mathbf{x})} p_{tr}(\mathbf{x}, y) d\mathbf{x} dy.\quad (48)$$

Then, we consider

$$\hat{h} = \min_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} w^{(\lambda,\alpha)}(\mathbf{x}) \ell(h(\mathbf{x}), y) p_{tr}(\mathbf{x}, y) d\mathbf{x} dy,\quad (49)$$

where

$$w^{(\lambda,\alpha)}(\mathbf{x}) = \frac{m_f^{(\lambda,\alpha)}(p_{tr}(\mathbf{x}), p_{te}(\mathbf{x}))}{p_{tr}(\mathbf{x})}.\quad (50)$$

We can see that AIWERM is a special case when $\alpha = 1$ and RIWERM is a special case when $\alpha = 3$.

Proof 4 (Proofs of Propositions 4.2 and 4.3) Let

$$\boldsymbol{\theta}^{(\lambda,\alpha)} = m_f^{(\lambda,\alpha)}(\boldsymbol{\theta}_{tr}, \boldsymbol{\theta}_{te}),\quad (51)$$

and let $R^{(\alpha)}$ be the Riemann curvature tensor defined in Definition 6.5 with respect to the α -connection $\nabla^{(\alpha)}$.

We define the relative curvature tensor as

$$R^{(\alpha,\beta)}(X, Y, Z) = \left[\nabla_X^{(\alpha)}, \nabla_Y^{(\beta)} \right] Z - \nabla_{[X,Y]}^{(\alpha)} Z\quad (52)$$

and the difference tensor as

$$K(X, Y) = \nabla_X^* Y - \nabla_X Y.\quad (53)$$

For any $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$, we have

$$\begin{aligned}\nabla_X^{(\alpha)} \nabla_Y^{(\beta)} Z &= \left(\frac{1+\alpha}{2} \nabla_X^* + \frac{1-\alpha}{2} \nabla_X \right) \left(\frac{1+\beta}{2} \nabla_Y^* + \frac{1-\beta}{2} \nabla_Y \right) Z \\ &= \frac{(1+\alpha)(1+\beta)}{4} \nabla_X^* \nabla_Y^* Z + \frac{(1+\alpha)(1-\beta)}{4} \nabla_X^* \nabla_Y Z \\ &\quad + \frac{(1-\alpha)(1+\beta)}{4} \nabla_X \nabla_Y^* Z + \frac{(1-\alpha)(1-\beta)}{4} \nabla_X \nabla_Y Z.\end{aligned}\quad (54)$$

$$\begin{aligned}\nabla_Y^{(\beta)} \nabla_X^{(\alpha)} Z &= \left(\frac{1+\beta}{2} \nabla_Y^* + \frac{1-\beta}{2} \nabla_Y \right) \left(\frac{1+\alpha}{2} \nabla_X^* + \frac{1-\alpha}{2} \nabla_X \right) Z \\ &= \frac{(1+\alpha)(1+\beta)}{4} \nabla_Y^* \nabla_X^* Z + \frac{(1-\alpha)(1+\beta)}{4} \nabla_Y^* \nabla_X Z \\ &\quad + \frac{(1+\alpha)(1-\beta)}{4} \nabla_Y \nabla_X^* Z + \frac{(1-\alpha)(1-\beta)}{4} \nabla_Y \nabla_X Z.\end{aligned}\quad (55)$$

$$\nabla_{[X,Y]}^{(\alpha)} Z = \frac{1+\alpha}{2} \nabla_{[X,Y]}^* Z + \frac{1-\alpha}{2} \nabla_{[X,Y]} Z. \quad (56)$$

Then

$$\begin{aligned}R^{(\alpha,\beta)}(X, Y, Z) &= \nabla_X^{(\alpha)} \nabla_Y^{(\beta)} Z - \nabla_X^{(\beta)} \nabla_Y^{(\alpha)} Z - \nabla_{[X,Y]}^{(\alpha)} Z \\ &= \frac{(1+\alpha)(1+\beta)}{4} (\nabla_X^* \nabla_Y^* - \nabla_Y^* \nabla_X^*) Z \\ &\quad + \frac{(1+\alpha)(1-\beta)}{4} (\nabla_X^* \nabla_Y - \nabla_Y \nabla_X^*) Z \\ &\quad + \frac{(1-\alpha)(1+\beta)}{4} (\nabla_X \nabla_Y^* - \nabla_Y^* \nabla_X) Z \\ &\quad + \frac{(1-\alpha)(1-\beta)}{4} (\nabla_X \nabla_Y - \nabla_Y \nabla_X) Z \\ &\quad - \frac{1+\alpha}{2} \nabla_{[X,Y]}^* Z - \frac{1-\alpha}{2} \nabla_{[X,Y]} Z \\ &= \frac{(1+\alpha)(1+\beta)}{4} \left\{ R^*(X, Y, Z) + \nabla_{[X,Y]}^* Z \right\} \\ &\quad + \frac{(1+\alpha)(1-\beta)}{4} \left\{ R^{(1,-1)}(X, Y, Z) + \nabla_{[X,Y]}^* Z \right\} \\ &\quad + \frac{(1-\alpha)(1+\beta)}{4} \left\{ R^{(-1,1)}(X, Y, Z) + \nabla_{[X,Y]} Z \right\} \\ &\quad + \frac{(1-\alpha)(1-\beta)}{4} \left\{ R^{(-1,-1)}(X, Y, Z) + \nabla_{[X,Y]}^* Z \right\} \\ &\quad - \frac{1+\alpha}{2} \nabla_{[X,Y]}^* Z - \frac{1-\alpha}{2} \nabla_{[X,Y]} Z\end{aligned}\quad (57)$$

$$\begin{aligned}4R^{(\alpha,\beta)} &= (1+\alpha)(1+\beta)R^* + (1-\alpha)(1-\beta)R \\ &\quad + (1+\alpha)(1-\beta)R^{(1,-1)} + (1-\alpha)(1+\beta)R^{(-1,1)}.\end{aligned}\quad (58)$$

We also have

$$\begin{aligned}
K^{(\alpha,\beta)}(X, Y) &= \nabla_X^{(\beta)}Y - \nabla_X^{(\alpha)}Y \\
&= \left\{ \frac{1+\beta}{2} \nabla_X^*Y + \frac{1-\beta}{2} \nabla_X Y \right\} - \left\{ \frac{1+\alpha}{2} \nabla_X^*Y + \frac{1-\alpha}{2} \nabla_X Y \right\} \\
&= \frac{\beta-\alpha}{2} \nabla_X^*Y - \frac{\beta-\alpha}{2} \nabla_X Y = \frac{\beta-\alpha}{2} K(X, Y) \quad (59)
\end{aligned}$$

$$K^{(\alpha,\beta)}\left(X, K^{(\alpha,\beta)}(Y, Z)\right) = \frac{\beta-\alpha}{2} K\left(X, K^{(\alpha,\beta)}(Y, Z)\right) = \frac{(\beta-\alpha)^2}{4} K\left(X, K(Y, Z)\right). \quad (60)$$

Combining them, the following relations hold:

$$K^{(\beta,\alpha)}\left(X, K^{(\beta,\alpha)}(Y, Z)\right) = K^{(\beta,\alpha)}\left(X, \nabla_Y^{(\alpha)}Z - \nabla_Y^{(\beta)}Z\right) \quad (61)$$

$$\begin{aligned}
&= K^{(\beta,\alpha)}\left(X, \nabla_Y^{(\alpha)}Z\right) - K^{(\beta,\alpha)}\left(X, \nabla_Y^{(\beta)}Z\right) \\
&= \nabla_X^{(\alpha)}\nabla_Y^{(\alpha)}Z - \nabla_X^{(\beta)}\nabla_Y^{(\alpha)}Z - \nabla_X^{(\alpha)}\nabla_Y^{(\beta)}Z + \nabla_X^{(\beta)}\nabla_Y^{(\beta)}Z \quad (62)
\end{aligned}$$

$$\frac{(\alpha-\beta)^2}{4} K\left(X, K(Y, Z)\right) = \nabla_X^{(\alpha)}\nabla_Y^{(\alpha)}Z - \nabla_X^{(\beta)}\nabla_Y^{(\alpha)}Z - \nabla_X^{(\alpha)}\nabla_Y^{(\beta)}Z + \nabla_X^{(\beta)}\nabla_Y^{(\beta)}Z. \quad (63)$$

Swapping X and Y , we have

$$\begin{aligned}
&\frac{(\alpha-\beta)^2}{4} \left\{ K\left(X, K(Y, Z)\right) - K\left(Y, K(X, Z)\right) \right\} \\
&= R^{(\alpha)}(X, Y, Z) + R^{(\beta)}(X, Y, Z) - \left\{ \left[\nabla_X^{(\alpha)}, \nabla_Y^{(\beta)} \right] Z - \nabla_{[X,Y]}^{(\alpha)} Z \right\} \\
&\quad - \left\{ \left[\nabla_X^{(\beta)}, \nabla_Y^{(\alpha)} \right] Z - \nabla_{[X,Y]}^{(\beta)} Z \right\} \\
&= R^{(\alpha)}(X, Y, Z) + R^{(\beta)}(X, Y, Z) - R^{(\alpha,\beta)}(X, Y, Z) - R^{(\beta,\alpha)}(X, Y, Z). \quad (64)
\end{aligned}$$

Making $\alpha = \beta$, we have

$$\begin{aligned}
4R^{(\alpha)} &= (1+\alpha)^2 R^* + (1-\alpha)^2 R + (1-\alpha^2)R^{(1,-1)} + (1-\alpha^2)R^{(-1,1)} \\
&= (1+\alpha^2)R^* + (1-\alpha)^2 R + (1-\alpha^2)\left(R^{(1,-1)} + R^{(-1,1)}\right). \quad (65)
\end{aligned}$$

Making $\alpha = 1$ and $\beta = -1$, we also have

$$\begin{aligned}
R^{(1,-1)}(X, Y, Z) + R^{(-1,1)}(X, Y, Z) &= R^*(X, Y, Z) + R(X, Y, Z) \\
&\quad - \left\{ K\left(X, K(Y, Z)\right) - K\left(Y, K(X, Z)\right) \right\}. \quad (66)
\end{aligned}$$

From Eq. (65) and (66), we obtain

$$\begin{aligned}
4R^{(\alpha)} &= (1 + \alpha)^2 R^*(X, Y, Z) + (1 - \alpha)^2 R(X, Y, Z) \\
&\quad + (1 - \alpha^2) R^*(X, Y, Z) + (1 - \alpha^2) \left\{ K(X, K(Y, Z)) - K(Y, K(X, Z)) \right\} \\
&= 2(1 + \alpha) R^*(X, Y, Z) + 2(1 - \alpha) R(X, Y, Z) \\
&\quad + (1 - \alpha^2) \left\{ K(Y, K(X, Z)) - K(X, K(Y, Z)) \right\} \tag{67}
\end{aligned}$$

Since the exponential family is dually flat, that is $R = 0$ and $R^* = 0$, and the Riemann curvature tensor with respect to $\nabla^{(\alpha)}$ is

$$R^{(\alpha)}(X, Y, Z) = \frac{1 - \alpha^2}{4} \Lambda, \tag{68}$$

$$\Lambda = \left(K(Y, K(X, Z)) - K(X, K(Y, Z)) \right). \tag{69}$$

Then, the geometric bias vector of $\theta^{(\lambda, \alpha)}$ is

$$b(\alpha, \lambda) = (1 - \lambda) \left\{ e_1 + \text{tr}_g \left(\frac{1 - \alpha^2}{2} \Lambda_{ikj} d\theta^i \otimes d\theta^j \right) e_2 \right\}, \tag{70}$$

where tr_g is the trace operation on the metric tensor g , and Λ_{ikj} is the element of Λ in Eq. (69). Since AIWERM and RIWERM are two special cases for $\alpha = 1$ and $\alpha = 3$, we have

$$b(1, \lambda) = (1 - \lambda) e_1, \tag{71}$$

$$b(3, \lambda) = (1 - \lambda) \left\{ e_1 + \text{tr}_g \left(-4 \Lambda_{ikj} d\theta^i \otimes d\theta^j \right) e_2 \right\}. \tag{72}$$

Appendix C: Learning guarantee

Generalization bounds of weighted maximum likelihood estimator for the target domain are derived in Cortes et al. (2010), and our weight function (14) is compatible with their bound.

Then, the gap between the expected (with respect to test distribution) loss $\mathcal{R}(h)$ and

empirical risk $L(h; \lambda, \alpha)$ is bounded as

$$\begin{aligned}
|\mathcal{R}(h) - L(h; \lambda, \alpha)| &\leq \left| \mathbb{E}_{p_{tr}} \left[\left\{ \frac{p_{te}(\mathbf{x})}{p_{tr}(\mathbf{x})} - w^{(\lambda, \alpha)}(\mathbf{x}) \right\} \right] \right| \ell(h(\mathbf{x}, y(\mathbf{x}))) \\
&+ 2^{5/4} \max \left(\sqrt{\mathbb{E}_{p_{tr}}(w^{(\lambda, \alpha)}(\mathbf{x}))^2 \ell^2(h(\mathbf{x}, y(\mathbf{x})))}, \sqrt{\mathbb{E}_{\hat{p}_{tr}}(w^{(\lambda, \alpha)}(\mathbf{x}))^2 \ell^2(h(\mathbf{x}, y(\mathbf{x})))} \right) \\
&\times \left(\frac{p \log \frac{2n_{tr}e}{p} + \log \frac{4}{\delta}}{n_{tr}} \right)^{\frac{3}{8}}. \tag{73}
\end{aligned}$$

In the above inequality, p is the pseudo-dimension of the function class $\{w^{\lambda, \alpha}(\mathbf{x})\ell(h(\mathbf{x}), y(\mathbf{x})) \mid h \in \mathcal{H}\}$ where $y(\mathbf{x})$ is the ground truth function of connecting \mathbf{x} and y as $y = y(\mathbf{x})$. The first term of the r.h.s. of the above inequality is the bias introduced by using $w^{\lambda, \alpha}$ instead of a standard density ratio, and the second term reflects the variance. It is worth mentioning that the term $\mathbb{E}_{p_{tr}}(w^{(\lambda, \alpha)}(\mathbf{x}))^2 \ell^2(h(\mathbf{x}, y(\mathbf{x})))$ is further bounded by $d_2(p_{te} \parallel p_{tr}) = \int_{\mathbf{x} \in \mathcal{X}} \frac{p_{te}^2(\mathbf{x})}{p_{tr}(\mathbf{x})} d\mathbf{x}$.

Appendix D: Optimization of the generalized IWERM

In the expected improvement strategy, the $t + 1^{th}$ point $(\lambda, \alpha)_{t+1}$ is selected according to the following equation.

$$(\lambda, \alpha)_{t+1} = \arg \min_{(\lambda, \alpha)} \mathbb{E} \left[\max \left(0, h_{t+1}(\lambda, \alpha) - L(\lambda^\dagger, \alpha^\dagger) \right) \middle| D_t \right],$$

where $L(\lambda^\dagger, \alpha^\dagger)$ is the maximum value of empirical risk that has been encountered so far, $h_{t+1}(\lambda, \alpha)$ is the posterior mean of the surrogate at the $t + 1^{th}$ step and $D_t = \{(\lambda, \alpha)_i, L(\lambda_i, \alpha_i)\}_{i=1}^t$. This equation for Gaussian process surrogate is an analytical expression:

$$\begin{aligned}
a_{EI}(\lambda, \alpha) &= (\mu_t(\lambda, \alpha) - L(\lambda^\dagger, \alpha^\dagger))\Phi(Z) + \sigma_t(\lambda, \alpha)\phi(Z), \\
Z &= \frac{\mu_t(\lambda, \alpha) - L(\lambda^\dagger, \alpha^\dagger)}{\sigma_t(\lambda, \alpha)},
\end{aligned}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are normal cumulative and density functions, respectively, and μ_t and σ_t are mean and standard deviation of $\{(\lambda, \alpha)_i\}_{i=1}^t$.

Appendix E: Additional experimental results

6.1 Experimental results on LIBSVM dataset

We show that for the LIBSVM dataset, IGIWERM is effective even for multiple models. Table 4 shows the results for each model. We use the scikit-learn Pedregosa et al. (2011) implementation of the models, and the hyperparameters of each model are the default values of this library. Figure 5 also shows the relationship between the two parameters of IGIWERM and the errors that can be achieved. For this visualization, we explore the parameter pairs by grid search and evaluate their performance at that time. From this figure, it is seen that the best performance is often achieved when $\alpha \neq 1$ and $\alpha \neq 3$, showing the sub-optimality of conventional methods.

6.2 Experimental results on regression problems

In this section, we present experimental results for the regression problem. All the datasets used in this experiment are available in the scikit-learn Pedregosa et al. (2011) dataset collection. We use SVR with Radial Basis Function (RBF) kernel as the base regressor. Table 5 shows the results of this experiment. In this experiment, we use MSE as a metric, and this table shows that IGIWERM is superior to existing covariate shift adaptation methods. Figure 6 shows the relationship between the two parameters of IGIWERM and the mean squared errors that can be achieved. This figure shows that, as in the case of binary classification, the optimal parameters do not necessarily match those of existing methods.

6.3 Experimental results on multi-class classification

We also introduce the additional experimental results for the multi-class classification problem. All the datasets used in this experiment are available in the scikit-learn Pedregosa et al. (2011) dataset collection. We also use the scikit-learn Pedregosa et al. (2011) implementation of the models, and the hyperparameters of each model are the default values of this library. We note that the number of training sample for `covtype` is so large hence the results with SVM for this dataset are omitted. Table 6 shows the

experimental results, and we see that our proposed generalization outperforms existing methods. Figure 7 shows the relationship between the two parameters of IGIWERM and the errors that can be achieved. This figure also shows the sub-optimality of conventional methods.

6.4 Visualization of covariate shift

In Section 5, we induce the covariate shift by the method of Cortes et al Cortes et al. (2008). Figure 8 shows a plot by PCA of each dataset splitted into the training set and test set. This figure shows that we are able to induce a covariate shift by partitioning the dataset.

References

- Amari, S. (1985). Differential-geometrical methods in statistics. *Lecture Notes on Statistics*, 28:1.
- Amari, S. (2009). α -divergence is unique, belonging to both f -Divergence and bregman divergence classes. *IEEE Trans. Inf. Theory*, 55(11):4925–4931.
- Amari, S. (2016). *Information Geometry and Its Applications*. Springer.
- Amari, S. and Nagaoka, H. (2007). *Methods of Information Geometry*. American Mathematical Soc.
- Arpit, D., Zhou, Y., Kota, B. U., and Govindaraju, V. (2016). Normalization propagation: A parametric technique for removing internal covariate shift in deep networks.
- Awais, M., Iqbal, M. T. B., and Bae, S.-H. (2020). Revisiting internal covariate shift for batch normalization. *IEEE Trans Neural Netw Learn Syst*, PP.
- Bickel, S., Brückner, M., and Scheffer, T. (2009). Discriminative learning under covariate shift. *J. Mach. Learn. Res.*, 10(9).
- Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press.

- Cortes, C., Mansour, Y., and Mohri, M. (2010). Learning bounds for importance weighting. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*.
- Cortes, C., Mohri, M., Riley, M., and Rostamizadeh, A. (2008). Sample selection bias correction theory. In *Algorithmic Learning Theory*, pages 38–53. Springer Berlin Heidelberg.
- Duda, R. O., Hart, P. E., and Others (2006). *Pattern classification*. John Wiley & Sons.
- Fang, T., Lu, N., Niu, G., and Sugiyama, M. (2020). Rethinking importance weighting for deep learning under distribution shift. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Frazier, P. I. (2018). A tutorial on Bayesian optimization.
- Hachiya, H., Sugiyama, M., and Ueda, N. (2012). Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing*, 80:93–101.
- Hardy, G. H., Littlewood, J. E., and Pólya, G. (1952). *Inequalities*. By GH Hardy, JE Littlewood, G. Pólya.. University Press.
- Hassan, A., Damper, R., and Niranjana, M. (2013). On acoustic emotion recognition: Compensating for covariate shift. *IEEE Trans. Audio Speech Lang. Processing*, 21(7):1458–1468.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Science & Business Media.
- Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. J. (2007). Correcting sample selection bias by unlabeled data. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press.

- Huang, Y. and Yu, Y. (2020). An internal covariate shift bounding algorithm for deep neural networks by unitizing layers' outputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8465–8473. openaccess.thecvf.com.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.
- Jiang, J. (2008). A literature survey on domain adaptation of statistical classifiers. *URL: <http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey>*, 3:1–12.
- Jirayucharoensak, S., Pan-Ngum, S., and Israsena, P. (2014). EEG-based emotion recognition using deep learning network with principal component based covariate shift adaptation. *ScientificWorldJournal*, 2014:627892.
- Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *J. Global Optimiz.*, 13(4):455–492.
- Jost, J. (2017). *Riemannian Geometry and Geometric Analysis*. Springer, Cham.
- Kimura, M. and Hino, H. (2021). α -geodesical skew divergence. *Entropy*, 23(5).
- Li, Y., Kambara, H., Koike, Y., and Sugiyama, M. (2010). Application of covariate shift adaptation techniques in Brain–Computer interfaces. *IEEE Transactions on Biomedical Engineering*, 57(6):1318–1324.
- Mockus, J., Tiesis, V., and Zilinskas, A. (1978). The application of Bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of Machine Learning, second edition*. MIT Press.
- Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., and Snoek, J. (2020). Evaluating Prediction-Time batch normalization for robustness under covariate shift.

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. (2009). *Dataset Shift in Machine Learning*. The MIT Press.
- Raza, H., Cecotti, H., Li, Y., and Prasad, G. (2016). Adaptive learning with covariate shift-detection for motor imagery-based brain–computer interface. *Soft Computing*, 20(8):3085–3096.
- Sakai, T. and Shimizu, N. (2019). Covariate shift adaptation on learning from positive and unlabeled data. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4838–4845.
- Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How does batch normalization help optimization?(no, it is not about internal covariate shift). *arXiv preprint arXiv:1805.11604*.
- Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *J. Stat. Plan. Inference*, 90(2):227–244.
- Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 25, pages 2951–2959. Curran Associates, Inc.
- Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *J. Mach. Learn. Res.*, 8(35):985–1005.

- Sugiyama, M. and Müller, K.-R. (2005a). Generalization error estimation under covariate shift. In *Workshop on Information-Based Induction Sciences*, pages 21–26. [researchgate.net](https://www.researchgate.net).
- Sugiyama, M. and Müller, K.-R. (2005b). Input-dependent estimation of generalization error under covariate shift. *Statistics and Decisions-International Journal Stochastic Methods and Models*, 23(4):249–280.
- Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density ratio estimation in machine learning*. Cambridge University Press.
- Vapnik, V. (2013). *The Nature of Statistical Learning Theory*. Springer Science & Business Media.
- Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.
- Yamada, M., Sigal, L., and Raptis, M. (2012). No bias left behind: Covariate shift adaptation for discriminative 3D pose estimation. In *Computer Vision – ECCV 2012*, pages 674–687. Springer Berlin Heidelberg.
- Yamada, M., Suzuki, T., Kanamori, T., Hachiya, H., and Sugiyama, M. (2011). Relative Density-Ratio estimation for robust distribution comparison. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 24, pages 594–602. Curran Associates, Inc.
- Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning, ICML '04*, page 114, New York, NY, USA. Association for Computing Machinery.
- Zhang, T., Yamane, I., Lu, N., and Sugiyama, M. (2021). A one-step approach to covariate shift adaptation. *SN Comput. Sci.*, 2(4):319.

Table 4: Mean misclassification rates averaged over 10 trails on LIBSVM benchmark datasets. The numbers in the brackets are the standard deviations. For the methods with (optimal), the optimal parameters for the test data are obtained by the linear search. The lowest misclassification rates among five methods are shown with bold.

Dataset	model	unweighted	IWERM	AIWERM (optimal)	RIWERM (optimal)	ours
australian	Logistic Regression	22.76(± 9.53)	22.75(± 7.73)	22.19(± 9.35)	22.39(± 7.76)	21.47(± 8.91)
	SVM	33.46(± 23.65)	22.13(± 3.37)	21.98(± 3.36)	21.73(± 3.82)	18.85(± 3.99)
	AdaBoost	10.20(± 5.09)	16.35(± 10.67)	11.23(± 3.49)	15.47(± 2.72)	9.15(± 3.93)
	Naive Bayes	17.04(± 6.02)	14.92(± 5.96)	16.33(± 6.07)	15.49(± 6.07)	14.74(± 5.84)
	Random Forest	8.89(± 3.72)	8.64(± 3.20)	8.29(± 3.21)	8.38(± 3.47)	8.61(± 3.73)
breast-cancer	Logistic Regression	32.32(± 3.08)	32.87(± 3.56)	32.32(± 3.08)	32.32(± 3.08)	32.26(± 3.04)
	SVM	38.28(± 10.98)	41.23(± 15.39)	36.41(± 9.68)	36.13(± 10.81)	31.65(± 8.49)
	AdaBoost	5.09(± 1.65)	5.63(± 1.38)	6.01(± 1.13)	5.95(± 1.50)	4.84(± 1.50)
	Naive Bayes	11.27(± 4.09)	19.65(± 15.14)	10.36(± 5.14)	18.00(± 14.24)	10.03(± 3.53)
	Random Forest	3.32(± 1.23)	3.19(± 1.10)	3.19(± 1.13)	3.18(± 1.04)	3.13(± 1.04)
heart	Logistic Regression	39.68(± 7.90)	40.55(± 9.10)	39.96(± 7.40)	39.94(± 6.93)	36.56(± 8.29)
	SVM	45.17(± 6.98)	39.94(± 8.55)	39.76(± 8.49)	39.74(± 8.92)	35.37(± 6.84)
	AdaBoost	30.87(± 12.04)	29.24(± 6.47)	29.37(± 12.19)	31.27(± 8.37)	26.96(± 13.12)
	Naive Bayes	22.79(± 6.17)	24.78(± 7.99)	22.87(± 6.02)	24.58(± 7.98)	21.97(± 6.41)
	Random Forest	20.87(± 6.64)	20.98(± 6.62)	20.96(± 6.67)	21.96(± 6.70)	19.95(± 6.61)
diabetes	Logistic Regression	37.62(± 4.35)	40.22(± 4.10)	38.38(± 3.85)	40.11(± 3.74)	36.86(± 4.81)
	SVM	33.19(± 5.69)	37.22(± 6.63)	33.11(± 6.45)	33.38(± 5.74)	32.83(± 5.62)
	AdaBoost	37.69(± 4.28)	40.13(± 5.28)	40.76(± 4.31)	41.26(± 5.16)	33.45(± 4.35)
	Naive Bayes	39.29(± 3.98)	39.21(± 3.18)	39.26(± 2.97)	39.35(± 2.85)	38.10(± 4.02)
	Random Forest	30.09(± 3.03)	30.90(± 3.52)	31.07(± 3.10)	30.51(± 3.67)	29.46(± 2.99)
madelon	Logistic Regression	47.31(± 1.80)	47.80(± 1.57)	47.16(± 1.68)	46.81(± 1.56)	46.31(± 1.69)
	SVM	47.78(± 1.53)	47.28(± 2.20)	47.10(± 2.13)	47.12(± 1.65)	46.56(± 2.12)
	AdaBoost	42.92(± 1.40)	42.91(± 1.68)	43.36(± 1.81)	42.90(± 1.40)	40.64(± 7.32)
	Naive Bayes	41.90(± 1.05)	41.43(± 8.76)	41.79(± 8.05)	41.62(± 7.43)	41.03(± 8.32)
	Random Forest	35.90(± 0.83)	35.42(± 1.75)	35.13(± 1.30)	34.79(± 1.75)	35.56(± 2.03)

Table 5: Mean squared errors averaged over 10 trails on scikit-learn Pedregosa et al. (2011) regression benchmark datasets. The numbers in the brackets are the standard deviations. For the methods with (optimal), the optimal parameters for the test data are obtained by the linear search. The lowest mean squared errors are shown with bold.

Dataset	#features	#data	unweighted	IWERM	AIWERM (optimal)	RIWERM (optimal)	ours
boston	13	506	83.22(± 5.72)	69.87(± 2.31)	69.68(± 1.46)	69.96(± 1.84)	68.36(± 1.20)
diabetes	10	442	0.049(± 0.007)	0.0501(± 0.009)	0.049(± 0.008)	0.049(± 0.009)	0.048(± 0.007)
california housing	8	20,640	1.432(± 0.095)	1.3214(± 0.345)	1.260(± 0.125)	1.261(± 0.086)	1.232(± 0.095)

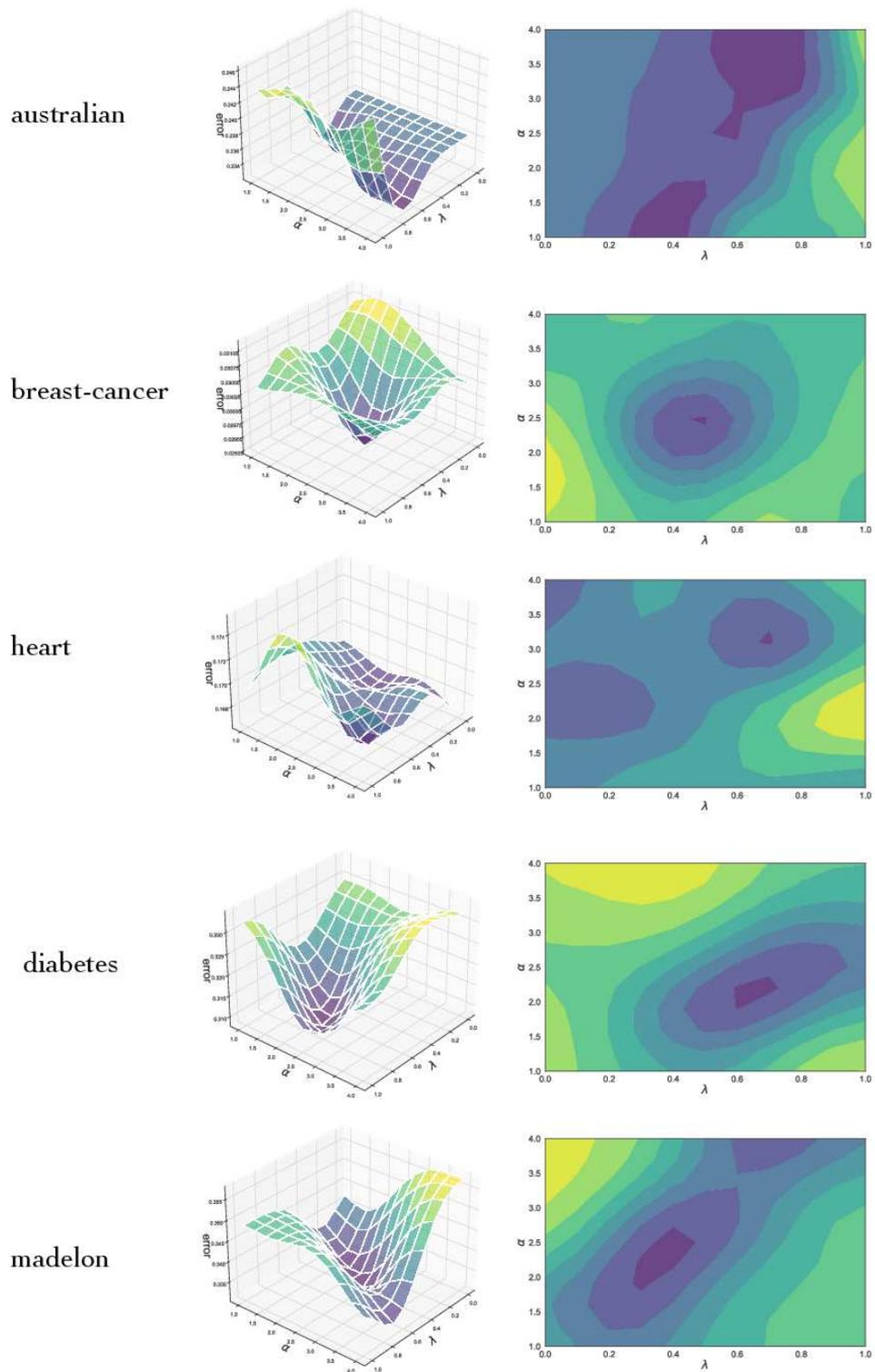


Figure 5: Visualization of grid search for α and λ on LIBSVM dataset. For the sake of clarity, we apply a moving average.

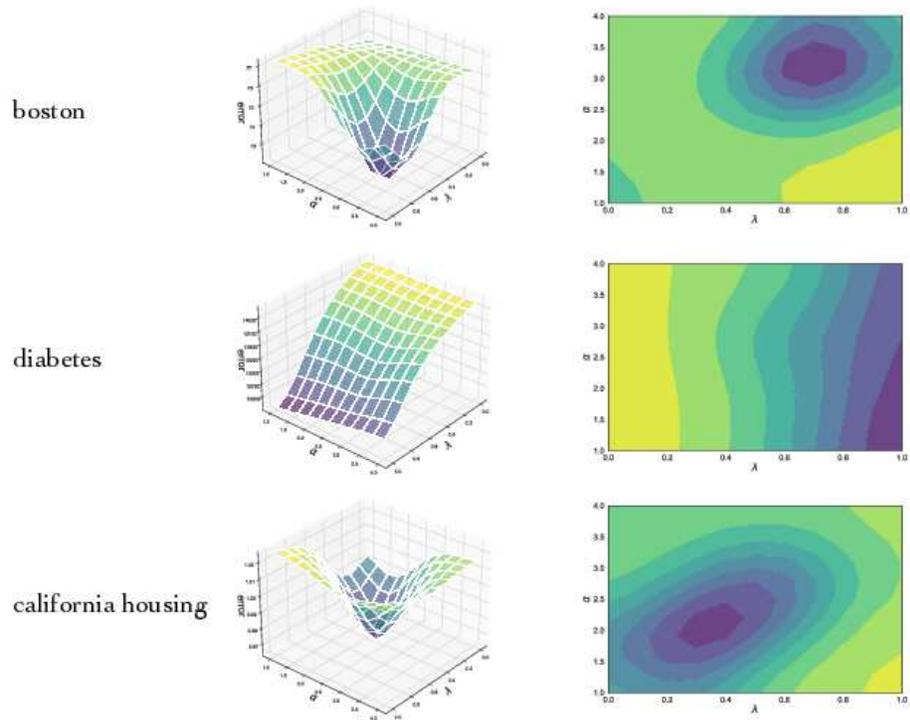


Figure 6: Visualization of grid search for α and λ on scikit-learn regression dataset. For the sake of visualization, we apply a moving average.

Table 6: Mean misclassification rates averaged over 10 trails on scikit-learn Pedregosa et al. (2011) multi-class classification benchmark datasets. The numbers in the brackets are the standard deviations. For the methods with (optimal), the optimal parameters for the test data are obtained by the linear search. The lowest misclassification rates among five methods are shown with bold.

Dataset	model	unweighted	IWERM	AIWERM (optimal)	RIWERM (optimal)	ours
digits	Logistic Regression	6.92(± 2.25)	7.10(± 1.76)	6.90(± 1.81)	6.89(± 1.80)	6.79(± 1.82)
	SVM	4.21(± 2.04)	3.94(± 1.14)	4.20(± 1.22)	4.02(± 1.18)	3.88(± 1.11)
	AdaBoost	67.98(± 12.35)	71.77(± 6.25)	71.26(± 8.21)	70.47(± 6.46)	65.20(± 8.20)
	Naive Bayes	19.07(± 2.45)	18.68(± 2.61)	19.31(± 2.64)	18.78(± 2.68)	18.58(± 2.66)
	Random Forest	6.85(± 2.40)	6.30(± 1.80)	6.23(± 1.53)	6.27(± 1.64)	6.17(± 1.90)
iris	Logistic Regression	54.69(± 20.51)	36.49(± 22.39)	45.78(± 18.09)	35.37(± 23.19)	28.89(± 20.54)
	SVM	55.16(± 22.60)	36.65(± 22.56)	33.21(± 20.25)	30.46(± 21.14)	29.04(± 20.02)
	AdaBoost	27.19(± 22.56)	26.00(± 22.98)	26.00(± 22.98)	26.00(± 22.98)	19.62(± 21.36)
	Naive Bayes	35.88(± 23.24)	35.97(± 26.79)	37.99(± 23.55)	33.84(± 25.64)	27.52(± 21.00)
	Random Forest	26.16(± 22.86)	32.17(± 21.21)	26.00(± 22.98)	28.47(± 22.63)	22.77(± 21.74)
covtype	Logistic Regression	45.36(± 13.08)	32.04(± 5.833)	30.62(± 4.64)	25.20(± 8.99)	19.99(± 5.56)
	SVM	—	—	—	—	—
	AdaBoost	47.53(± 14.51)	25.55(± 12.27)	25.47(± 14.14)	27.86(± 10.37)	18.96(± 7.29)
	Naive Bayes	41.13(± 15.65)	30.66(± 15.09)	28.48(± 15.66)	27.20(± 15.79)	19.64(± 15.62)
	Random Forest	23.51(± 3.31)	18.18(± 2.01)	17.28(± 2.05)	17.13(± 2.26)	16.42(± 2.08)

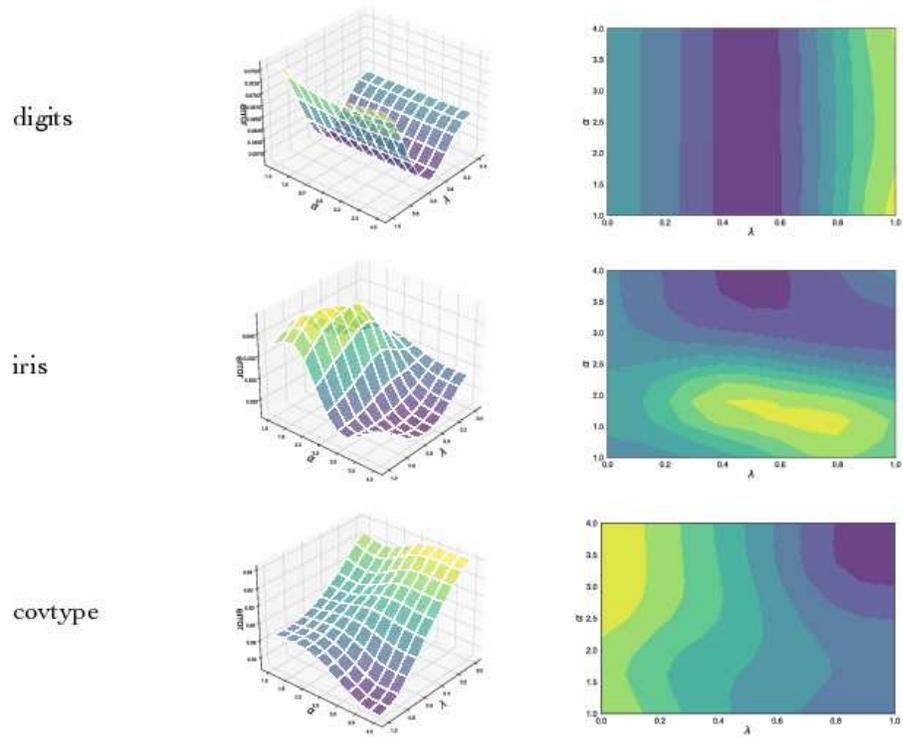


Figure 7: Visualization of grid search for α and λ on scikit-learn multi-class classification dataset. For the sake of visualization, we apply a moving average.

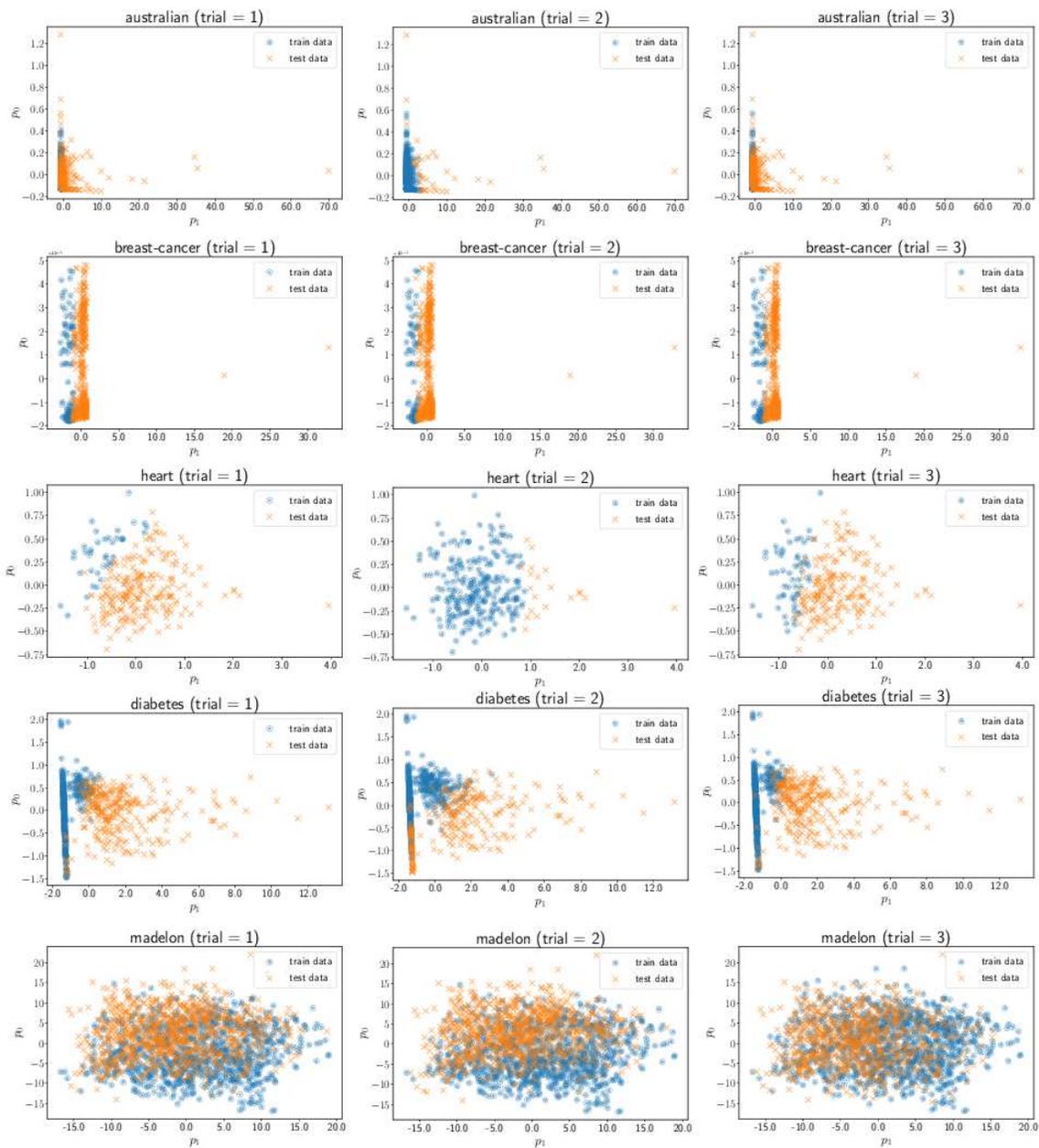


Figure 8: Plot of covariate shifts using the method of Cortes et al Cortes et al. (2008). Each dataset is included in LIBSVM and mapped to two dimensions by PCA.