Journal of Machine Learning Research 10 (2009) 1391-1445 Submitted 3/08; Revised 4/09; Published 7/09

A Least-squares Approach to Direct Importance Estimation*

Takafumi Kanamori KANAMORI@IS.NAGOYA-U.AC.JP
Department of Computer Science and Mathematical Informatics
Nagoya University
Furocho, Chikusaku, Nagoya 464-8603, Japan

Shohei Hido HIDO@JP.IBM.COM
IBM Research
Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato-shi, Kanagawa 242-8502, Japan

Masashi Sugiyama SUGI@CS.TITECH.AC.JP
Department of Computer Science
Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo 152-8552, Japan

Editor: Bianca Zadrozny

Abstract

We address the problem of estimating the ratio of two probability density functions, which is often referred to as the *importance*. The importance values can be used for various succeeding tasks such as *covariate shift adaptation* or *outlier detection*. In this paper, we propose a new importance estimation method that has a closed-form solution; the leave-one-out cross-validation score can also be computed analytically. Therefore, the proposed method is computationally highly efficient and simple to implement. We also elucidate theoretical properties of the proposed method such as the convergence rate and approximation error bounds. Numerical experiments show that the proposed method is comparable to the best existing method in accuracy, while it is computationally more efficient than competing approaches.

Keywords: importance sampling, covariate shift adaptation, novelty detection, regularization path, leave-one-out cross validation

1. Introduction

In the context of *importance sampling* (Fishman, 1996), the ratio of two probability density functions is called the *importance*. The problem of estimating the importance is attracting a great deal of attention these days since the importance can be used for various succeeding tasks such as *covariate shift adaptation* or *outlier detection*.

> **Covariate Shift Adaptation:** Covariate shift is a situation in supervised learning where the distributions of inputs change between the training and test phases but the conditional distribution of outputs given inputs remains unchanged (Shimodaira, 2000; Quiñonero-Candela et al., 2008). Covariate shift is conceivable in many real-world

*. A MATLAB® or R implementation of the proposed importance estimation algorithm, *unconstrained Least-Squares Importance Fitting* (uLSIF), is available from `http://sugiyama-www.cs.titech.ac.jp/~sugi/software/uLSIF/`.

applications such as bioinformatics (Baldi and Brunak, 1998; Borgwardt et al., 2006), brain-computer interfaces (Wolpaw et al., 2002; Sugiyama et al., 2007), robot control (Sutton and Barto, 1998; Hachiya et al., 2008), spam filtering (Bickel and Scheffer, 2007), and econometrics (Heckman, 1979). Under covariate shift, standard learning techniques such as maximum likelihood estimation or cross-validation are biased and therefore unreliable—the bias caused by covariate shift can be compensated by weighting the loss function according to the importance (Shimodaira, 2000; Zadrozny, 2004; Sugiyama and Müller, 2005; Sugiyama et al., 2007; Huang et al., 2007; Bickel et al., 2007).

Outlier Detection: The outlier detection task addressed here is to identify irregular samples in a validation data set based on a model data set that only contains regular samples (Schölkopf et al., 2001; Tax and Duin, 2004; Hodge and Austin, 2004; Hido et al., 2008). The values of the importance for regular samples are close to one, while those for outliers tend to be significantly deviated from one. Thus the values of the importance could be used as an index of the degree of outlyingness.

Below, we refer to the two sets of samples as the *training* set and the *test* set.

A naive approach to estimating the importance is to first estimate the training and test density functions from the sets of training and test samples separately, and then take the ratio of the estimated densities. However, density estimation is known to be a hard problem particularly in high-dimensional cases if we do not have simple and good parametric density models (Vapnik, 1998; Härdle et al., 2004). In practice, such an appropriate parametric model may not be available and therefore this naive approach is not so effective.

To cope with this problem, direct importance estimation methods which do not involve density estimation have been developed recently. The *kernel mean matching* (KMM) method (Huang et al., 2007) directly gives estimates of the importance at the training inputs by matching the two distributions efficiently based on a special property of *universal reproducing kernel Hilbert spaces* (Steinwart, 2001). The optimization problem involved in KMM is a convex quadratic program, so the unique global optimal solution can be obtained using a standard optimization software. However, the performance of KMM depends on the choice of tuning parameters such as the kernel parameter and the regularization parameter. For the kernel parameter, a popular heuristic of using the median distance between samples as the Gaussian width could be useful in some cases (Schölkopf and Smola, 2002; Song et al., 2007). However, there seems no strong justification for this heuristic and the choice of other tuning parameters is still open.

A probabilistic classifier that separates training samples from test samples can be used for directly estimating the importance, for example, a *logistic regression* (LogReg) classifier (Qin, 1998; Cheng and Chu, 2004; Bickel et al., 2007). Maximum likelihood estimation of LogReg models can be formulated as a convex optimization problem, so the unique global optimal solution can be obtained. Furthermore, since the LogReg-based method only involves a standard supervised classification problem, the tuning parameters such as the kernel width and the regularization parameter can be optimized based on the standard cross-validation procedure. This is a very useful property in practice.

The *Kullback-Leibler importance estimation procedure* (KLIEP) (Sugiyama et al., 2008b; Nguyen et al., 2008) also directly gives an estimate of the importance function by matching the two distributions in terms of the Kullback-Leibler divergence (Kullback and Leibler, 1951). The optimization

problem involved in KLIEP is convex, so the unique global optimal solution—which tends to be sparse—can be obtained, when linear importance models are used. In addition, the tuning parameters in KLIEP can be optimized based on a variant of cross-validation.

As reviewed above, LogReg and KLIEP are more advantageous than KMM since the tuning parameters can be objectively optimized based on cross-validation. However, optimization procedures of LogReg and KLIEP are less efficient in computation than KMM due to high non-linearity of the objective functions to be optimized—more specifically, exponential functions induced by the LogReg model or the log function induced by the Kullback-Leibler divergence. The purpose of this paper is to develop a new importance estimation method that is equipped with a build-in model selection procedure as LogReg and KLIEP and is computationally more efficient than LogReg and KLIEP.

Our basic idea is to formulate the direct importance estimation problem as a least-squares function fitting problem. This formulation allows us to cast the optimization problem as a convex quadratic program, which can be efficiently solved using a standard quadratic program solver. Cross-validation can be used for optimizing the tuning parameters such as the kernel width or the regularization parameter. We call the proposed method *least-squares importance fitting* (LSIF). We further show that the solutions of LSIF is piecewise linear with respect to the ℓ_1-regularization parameter and the entire regularization path (that is, all solutions for different regularization parameter values) can be computed efficiently based on the *parametric optimization technique* (Best, 1982; Efron et al., 2004; Hastie et al., 2004). Thanks to this regularization path tracking algorithm, LSIF is computationally efficient in model selection scenarios. Note that in the regularization path tracking algorithm, we can trace the solution path without a quadratic program solver—we just need to compute matrix inverses.

LSIF is shown to be efficient in computation, but it tends to share a common weakness of regularization path tracking algorithms, that is, *accumulation of numerical errors* (Scheinberg, 2006). The numerical problem tends to be severe if there are many change points in the regularization path. To cope with this problem, we develop an approximation algorithm in the same least-squares framework. The approximation version of LSIF, which we call *unconstrained LSIF* (uLSIF), allows us to obtain the closed-form solution that can be computed just by solving a system of linear equations. Thus uLSIF is numerically stable when regularized properly. Moreover, the leave-one-out cross-validation score for uLSIF can also be computed analytically (cf. Wahba, 1990; Cawley and Talbot, 2004), which significantly improves the computational efficiency in model selection scenarios. We experimentally show that the accuracy of uLSIF is comparable to the best existing method while its computation is faster than other methods in covariate shift adaptation and outlier detection scenarios.

Our contributions in this paper are summarized as follows. A proposed density-ratio estimation method, LSIF, is equipped with cross-validation (which is an advantage over KMM) and is computationally efficient thanks to regularization path tracking (which is an advantage over KLIEP and LogReg). Furthermore, uLSIF is computationally even more efficient since its solution and leave-one-out cross-validation score can be computed analytically in a stable manner. The proposed methods, LSIF and uLSIF, are similar in spirit to KLIEP, but the loss functions are different: KLIEP uses the log loss while LSIF and uLSIF use the squared loss. The difference of the log functions allows us to improve computational efficiency significantly.

The rest of this paper is organized as follows. In Section 2, we propose a new importance estimation procedure based on least-squares fitting (LSIF) and show its theoretical properties. In

Section 3, we develop an approximation algorithm (uLSIF) which can be computed efficiently. In Section 4, we illustrate how the proposed methods behave using a toy data set. In Section 5, we discuss the characteristics of existing approaches in comparison with the proposed methods and show that uLSIF could be a useful alternative to the existing methods. In Section 6, we experimentally compare the performance of uLSIF and existing methods. Finally in Section 7, we summarize our contributions and outline future prospects. Those who are interested in practical implementation may skip the theoretical analyses in Sections 2.3, 3.2, and 3.3.

2. Direct Importance Estimation

In this section, we propose a new method of direct importance estimation.

2.1 Formulation and Notation

Let $\mathcal{D} \subset (\mathbb{R}^d)$ be the data domain and suppose we are given independent and identically distributed (i.i.d.) training samples $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ from a training distribution with density $p_{\mathrm{tr}}(x)$ and i.i.d. test samples $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ from a test distribution with density $p_{\mathrm{te}}(x)$:

$$\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}} \overset{i.i.d.}{\sim} p_{\mathrm{tr}}(x),$$
$$\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}} \overset{i.i.d.}{\sim} p_{\mathrm{te}}(x).$$

We assume that the training density is strictly positive, that is,

$$p_{\mathrm{tr}}(x) > 0 \text{ for all } x \in \mathcal{D}.$$

The goal of this paper is to estimate the *importance* $w(x)$ from $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$:

$$w(x) = \frac{p_{\mathrm{te}}(x)}{p_{\mathrm{tr}}(x)},$$

which is non-negative by definition. Our key restriction is that we want to avoid estimating densities $p_{\mathrm{te}}(x)$ and $p_{\mathrm{tr}}(x)$ when estimating the importance $w(x)$.

2.2 Least-squares Approach to Direct Importance Estimation

Let us model the importance $w(x)$ by the following linear model:

$$\widehat{w}(x) = \sum_{\ell=1}^{b} \alpha_\ell \varphi_\ell(x), \tag{1}$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_b)^\top$ are parameters to be learned from data samples, $^\top$ denotes the transpose of a matrix or a vector, and $\{\varphi_\ell(x)\}_{\ell=1}^b$ are basis functions such that

$$\varphi_\ell(x) \geq 0 \text{ for all } x \in \mathcal{D} \text{ and for } \ell = 1, 2, \ldots, b.$$

Note that b and $\{\varphi_\ell(x)\}_{\ell=1}^b$ could be dependent on the samples $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, for example, *kernel* models are also allowed. We explain how the basis functions $\{\varphi_\ell(x)\}_{\ell=1}^b$ are chosen in Section 2.5.

We determine the parameters $\{\alpha_\ell\}_{\ell=1}^b$ in the model $\widehat{w}(x)$ so that the following squared error J_0 is minimized:

$$\begin{aligned}
J_0(\alpha) &= \frac{1}{2}\int \left(\widehat{w}(x) - w(x)\right)^2 p_{\mathrm{tr}}(x)dx \\
&= \frac{1}{2}\int \widehat{w}(x)^2 p_{\mathrm{tr}}(x)dx - \int \widehat{w}(x)w(x)p_{\mathrm{tr}}(x)dx + \frac{1}{2}\int w(x)^2 p_{\mathrm{tr}}(x)dx \\
&= \frac{1}{2}\int \widehat{w}(x)^2 p_{\mathrm{tr}}(x)dx - \int \widehat{w}(x)p_{\mathrm{te}}(x)dx + \frac{1}{2}\int w(x)^2 p_{\mathrm{tr}}(x)dx,
\end{aligned}$$

where in the second term the probability density $p_{\mathrm{tr}}(x)$ is canceled with that included in $w(x)$. The squared loss $J_0(\alpha)$ is defined as the expectation under the probability of training samples. In covariate shift adaptation (see Section 6.2) and outlier detection (see Section 6.3), the importance values on the training samples are used. Thus, the definition of $J_0(\alpha)$ well agrees with our goal.

The last term of $J_0(\alpha)$ is a constant and therefore can be safely ignored. Let us denote the first two terms by J:

$$\begin{aligned}
J(\alpha) &= \frac{1}{2}\int \widehat{w}(x)^2 p_{\mathrm{tr}}(x)dx - \int \widehat{w}(x)p_{\mathrm{te}}(x)dx \\
&= \frac{1}{2}\sum_{\ell,\ell'=1}^{b} \alpha_\ell \alpha_{\ell'} \left(\int \varphi_\ell(x)\varphi_{\ell'}(x)p_{\mathrm{tr}}(x)dx\right) - \sum_{\ell=1}^{b} \alpha_\ell \left(\int \varphi_\ell(x)p_{\mathrm{te}}(x)dx\right) \\
&= \frac{1}{2}\alpha^\top H\alpha - h^\top \alpha, \qquad (2)
\end{aligned}$$

where H is the $b \times b$ matrix with the (ℓ,ℓ')-th element

$$H_{\ell,\ell'} = \int \varphi_\ell(x)\varphi_{\ell'}(x)p_{\mathrm{tr}}(x)dx, \qquad (3)$$

and h is the b-dimensional vector with the ℓ-th element

$$h_\ell = \int \varphi_\ell(x)p_{\mathrm{te}}(x)dx.$$

Approximating the expectations in J by empirical averages, we obtain

$$\begin{aligned}
\widehat{J}(\alpha) &= \frac{1}{2n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}})^2 - \frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}} \widehat{w}(x_j^{\mathrm{te}}) \\
&= \frac{1}{2}\sum_{\ell,\ell'=1}^{b} \alpha_\ell \alpha_{\ell'} \left(\frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}})\varphi_{\ell'}(x_i^{\mathrm{tr}})\right) - \sum_{\ell=1}^{b} \alpha_\ell \left(\frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}})\right) \\
&= \frac{1}{2}\alpha^\top \widehat{H}\alpha - \widehat{h}^\top \alpha,
\end{aligned}$$

where $\widehat{H}$ is the $b \times b$ matrix with the (ℓ,ℓ')-th element

$$\widehat{H}_{\ell,\ell'} = \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}})\varphi_{\ell'}(x_i^{\mathrm{tr}}), \qquad (4)$$

and $\widehat{h}$ is the b-dimensional vector with the ℓ-th element

$$\widehat{h}_\ell = \frac{1}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}}). \tag{5}$$

Taking into account the non-negativity of the importance function $w(x)$, we can formulate our optimization problem as follows.

$$\min_{\alpha \in \mathbb{R}^b} \left[\frac{1}{2} \alpha^\top \widehat{H} \alpha - \widehat{h}^\top \alpha + \lambda 1_b^\top \alpha \right] \quad \text{subject to } \alpha \geq 0_b, \tag{6}$$

where 0_b and 1_b are the b-dimensional vectors with all zeros and ones, respectively; the vector inequality $\alpha \geq 0_b$ is applied in the element-wise manner, that is, $\alpha_\ell \geq 0$ for $\ell = 1, 2, \ldots, b$. In Eq. (6), we included a penalty term $\lambda 1_b^\top \alpha$ for regularization purposes, where λ (≥ 0) is a regularization parameter. The above is a convex quadratic programming problem and therefore the unique global optimal solution can be computed efficiently by a standard optimization package. We call this method *Least-Squares Importance Fitting* (LSIF).

We can also use the ℓ_2-regularizer $\alpha^\top \alpha$ instead of the ℓ_1-regularizer $1_b^\top \alpha$ without changing the computational property. However, using the ℓ_1-regularizer would be more advantageous since the solution tends to be sparse (Williams, 1995; Tibshirani, 1996; Chen et al., 1998). Furthermore, as shown in Section 2.6, the use of the ℓ_1-regularizer allows us to compute the entire regularization path efficiently (Best, 1982; Efron et al., 2004; Hastie et al., 2004). The ℓ_2-regularization method will be used for theoretical analysis in Section 3.3.

2.3 Convergence Analysis of LSIF

Here, we theoretically analyze the convergence property of the solution $\widehat{\alpha}$ of the LSIF algorithm; practitioners may skip this theoretical analysis.

Let $\widehat{\alpha}(\lambda)$ be the solution of the LSIF algorithm with regularization parameter λ, and let $\alpha^*(\lambda)$ be the optimal solution of the 'ideal' problem:

$$\min_{\alpha \in \mathbb{R}^b} \left[\frac{1}{2} \alpha^\top H \alpha - h^\top \alpha + \lambda 1_b^\top \alpha \right] \quad \text{subject to } \alpha \geq 0_b. \tag{7}$$

Below, we theoretically investigate the *learning curve* (Amari et al., 1992) of LSIF, that is, we elucidate the relation between $J(\widehat{\alpha}(\lambda))$ and $J(\alpha^*(\lambda))$ in terms of the expectation over all possible training and test samples as a function of the number of samples.

Let $\mathbb{E}$ be the expectation over all possible training samples of size n_{tr} and all possible test samples of size n_{te}. Let $\mathcal{A} \subset \{1, 2, \ldots, b\}$ be the set of *active* indices (Boyd and Vandenberghe, 2004), that is,

$$\mathcal{A} = \{\ell \mid \alpha_\ell^*(\lambda) = 0,\ \ell = 1, 2, \ldots, b\}.$$

For the active set $\mathcal{A} = \{j_1, j_2, \ldots, j_{|\mathcal{A}|}\}$ with $j_1 < j_2 < \cdots < j_{|\mathcal{A}|}$, let E be the $|\mathcal{A}| \times b$ indicator matrix with the (i, j)-th element

$$E_{i,j} = \begin{cases} 1 & j = j_i, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, let $\widehat{\mathcal{A}}$ be the active set of $\widehat{\alpha}(\lambda)$:

$$\widehat{\mathcal{A}} = \{\ell \mid \widehat{\alpha}_\ell(\lambda) = 0,\ \ell = 1,2,\ldots,b\}.$$

For the active set $\widehat{\mathcal{A}} = \{\widehat{j}_1, \widehat{j}_2, \ldots, \widehat{j}_{|\widehat{\mathcal{A}}|}\}$ with $\widehat{j}_1 < \widehat{j}_2 < \cdots < \widehat{j}_{|\widehat{\mathcal{A}}|}$, let $\widehat{E}$ be the $|\widehat{\mathcal{A}}| \times b$ indicator matrix with the (i,j)-th element similarly defined by

$$\widehat{E}_{i,j} = \begin{cases} 1 & j = \widehat{j}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

First, we show the optimality condition of (6) which will be used in the following theoretical analyses. The *Lagrangian* of the optimization problem (6) is given as

$$L(\alpha, \xi) = \frac{1}{2}\alpha^\top \widehat{H}\alpha - \widehat{h}^\top \alpha + \lambda 1_b^\top \alpha - \xi^\top \alpha,$$

where ξ is the b-dimensional *Lagrange multiplier* vector. Then the *Karush-Kuhn-Tucker (KKT) conditions* (Boyd and Vandenberghe, 2004) are expressed as follows:

$$\widehat{H}\alpha - \widehat{h} + \lambda 1_b - \xi = 0_b, \tag{9}$$
$$\alpha \geq 0_b,$$
$$\xi \geq 0_b,$$
$$\xi_\ell \alpha_\ell = 0 \text{ for } \ell = 1,2,\ldots,b. \tag{10}$$

Let $\widehat{\xi}'(\lambda)$ be the $|\widehat{\mathcal{A}}|$-dimensional vector with the i-th element being the $\widehat{j}_i$-th element of $\widehat{\xi}(\lambda)$:

$$\widehat{\xi}'_i(\lambda) = \widehat{\xi}_{\widehat{j}_i}(\lambda), \qquad i = 1,\ldots,|\widehat{\mathcal{A}}|. \tag{11}$$

We assume that $\widehat{\xi}'(\lambda)$ only contains non-zero elements of $\widehat{\xi}(\lambda)$. Let $\widehat{G}$ be

$$\widehat{G} = \begin{pmatrix} \widehat{H} & -\widehat{E}^\top \\ -\widehat{E} & O_{|\widehat{\mathcal{A}}| \times |\widehat{\mathcal{A}}|} \end{pmatrix},$$

where $O_{|\widehat{\mathcal{A}}| \times |\widehat{\mathcal{A}}|}$ is the $|\widehat{\mathcal{A}}| \times |\widehat{\mathcal{A}}|$ matrix with all zeros. Then Eqs. (9) and (10) are together expressed in a matrix form as

$$\widehat{G}\begin{pmatrix} \widehat{\alpha}(\lambda) \\ \widehat{\xi}'(\lambda) \end{pmatrix} = \begin{pmatrix} \widehat{h} - \lambda 1_b \\ 0_{|\widehat{\mathcal{A}}|} \end{pmatrix}. \tag{12}$$

Regarding the matrix $\widehat{G}$, we have the following lemma:

Lemma 1 *The matrix $\widehat{G}$ is invertible if $\widehat{H}$ is invertible.*

The proof of the above lemma is given in Appendix A. Below, we assume that $\widehat{H}$ is invertible. Then the inverse of $\widehat{G}$ exists and multiplying $\widehat{G}^{-1}$ from the left-hand side of Eq. (12) yields

$$\begin{pmatrix} \widehat{\alpha}(\lambda) \\ \widehat{\xi}'(\lambda) \end{pmatrix} = \widehat{G}^{-1}\begin{pmatrix} \widehat{h} - \lambda 1_b \\ 0_{|\mathcal{A}|} \end{pmatrix}. \tag{13}$$

The following inversion formula holds for block matrices (Petersen and Pedersen, 2007):

$$\begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix}^{-1} = \begin{pmatrix} M_1^{-1} + M_1^{-1}M_2M_0^{-1}M_3M_1^{-1} & -M_1^{-1}M_2M_0^{-1} \\ -M_0^{-1}M_3M_1^{-1} & M_0^{-1} \end{pmatrix}, \tag{14}$$

where

$$M_0 = M_4 - M_3M_1^{-1}M_2.$$

Applying Eq. (14) to Eq. (13), we have

$$\widehat{\alpha}(\lambda) = \widehat{A}(\widehat{h} - \lambda 1_b), \tag{15}$$

where $\widehat{A}$ is defined by

$$\widehat{A} = \widehat{H}^{-1} - \widehat{H}^{-1}\widehat{E}^\top(\widehat{E}\widehat{H}^{-1}\widehat{E}^\top)^{-1}\widehat{E}\widehat{H}^{-1}. \tag{16}$$

When the Lagrange multiplier vector satisfies

$$\xi_\ell^*(\lambda) > 0 \text{ for all } \ell \in \mathcal{A}, \tag{17}$$

we say that the *strict complementarity condition* is satisfied (Bertsekas et al., 2003). An important consequence of strict complementarity is that the optimal solution and the Lagrange multipliers of convex quadratic problems are uniquely determined. Then we have the following theorem.

Theorem 2 *Let P be the probability over all possible training samples of size n_{tr} and test samples of size n_{te}. Let $\xi^*(\lambda)$ be the Lagrange multiplier vector of the problem* (7) *and suppose $\xi^*(\lambda)$ satisfies the strict complementarity condition* (17)*. Then, there exists a positive constant $c > 0$ and a natural number N such that for $\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\} \geq N$,*

$$P(\widehat{\mathcal{A}} \neq \mathcal{A}) < e^{-c\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}}.$$

The proof of the above theorem is given in Appendix B. Theorem 2 shows that the probability that the active set $\widehat{\mathcal{A}}$ of the empirical problem (6) is different from the active set $\mathcal{A}$ of the ideal problem (7) is exponentially small. Thus we may regard $\widehat{\mathcal{A}} = \mathcal{A}$ in practice.

Let A be the 'ideal' counterpart of $\widehat{A}$:

$$A = H^{-1} - H^{-1}E^\top(EH^{-1}E^\top)^{-1}EH^{-1},$$

and let $C_{w,w'}$ be the $b \times b$ covariance matrix with the (ℓ, ℓ')-th element being the covariance between $w(x)\varphi_\ell(x)$ and $w'(x)\varphi_{\ell'}(x)$ under $p_{\mathrm{tr}}(x)$. Let

$$w^*(x) = \sum_{\ell=1}^{b} \alpha_\ell^*(\lambda)\varphi_\ell(x),$$

$$v(x) = \sum_{\ell=1}^{b} [A1_b]_\ell \varphi_\ell(x).$$

Let

$$f(n) = \omega(g(n))$$

denote that $f(n)$ asymptotically dominates $g(n)$; more precisely, for all $C > 0$, there exists n_0 such that

$$|Cg(n)| < |f(n)| \quad \text{for all } n > n_0.$$

Then we have the following theorem.

Theorem 3 *Assume that*

(a) *The optimal solution of the problem* (7) *satisfies the strict complementarity condition* (17)*.*

(b) n_{tr} *and* n_{te} *satisfy*

$$n_{\mathrm{te}} = \omega(n_{\mathrm{tr}}^2). \tag{18}$$

Then, for any $\lambda \geq 0$*, we have*

$$\mathbb{E}[J(\widehat{\alpha}(\lambda))] = J(\alpha^*(\lambda)) + \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(A(C_{w^*,w^*} - 2\lambda C_{w^*,v})) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{19}$$

The proof of the above theorem is given in Appendix C. This theorem elucidates the learning curve of LSIF up to the order of n_{tr}^{-1}. In Section 2.4.1, we discuss practical implications of this theorem.

2.4 Model Selection for LSIF

The practical performance of LSIF depends on the choice of the regularization parameter λ and basis functions $\{\varphi_\ell(x)\}_{\ell=1}^b$ (which we refer to as a *model*). Since our objective is to minimize the cost function J defined in Eq. (2), it is natural to determine the model such that J is minimized.

However, the value of the cost function J is inaccessible since it includes the expectation over unknown probability density functions $p_{\mathrm{tr}}(x)$ and $p_{\mathrm{te}}(x)$. The value of the empirical cost $\widehat{J}$ may be regarded as an estimate of J, but this is not useful for model selection purposes since it is heavily biased—the bias is caused by the fact that the same samples are used twice for learning the parameter α and estimating the value of J. Below, we give two practical methods of estimating the value of J in more precise ways.

2.4.1 INFORMATION CRITERION

In the same way as Theorem 3, we can obtain an asymptotic expansion of the empirical cost $\mathbb{E}\left[\widehat{J}(\widehat{\alpha}(\lambda))\right]$ as follows:

$$\mathbb{E}[\widehat{J}(\widehat{\alpha}(\lambda))] = J(\alpha^*(\lambda)) - \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(A(C_{w^*,w^*} + 2\lambda C_{w^*,v})) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{20}$$

Combining Eqs. (19) and (20), we have

$$\mathbb{E}[J(\widehat{\alpha}(\lambda))] = \mathbb{E}[\widehat{J}(\widehat{\alpha}(\lambda))] + \frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(AC_{w^*,w^*}) + o\left(\frac{1}{n_{\mathrm{tr}}}\right).$$

From this, we can immediately obtain an *information criterion* (Akaike, 1974; Konishi and Kitagawa, 1996) for LSIF:

$$\widehat{J}^{(\mathrm{IC})} = \widehat{J}(\widehat{\alpha}(\lambda)) + \frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(\widehat{A}\widehat{C}_{\widehat{w},\widehat{w}}),$$

where $\widehat{A}$ is defined by Eq. (16). $\widehat{E}$ is defined by Eq. (8) and $\widehat{C}_{w,w'}$ is the $b \times b$ covariance matrix with the (ℓ,ℓ')-th element being the covariance between $w(x)\varphi_\ell(x)$ and $w'(x)\varphi_{\ell'}(x)$ over $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. Since $\widehat{A}$ and $\widehat{C}_{\widehat{w},\widehat{w}}$ are consistent estimators of A and C_{w^*,w^*}, the above information criterion is unbiased up to the order of n_{tr}^{-1}.

Note that the term $\mathrm{tr}(\widehat{A}\widehat{C}_{\widehat{w},\widehat{w}})$ may be interpreted as the *effective dimension* of the model (Moody, 1992). Indeed, when $\widehat{w}(x)=1$, we have $\widehat{H}=\widehat{C}_{\widehat{w},\widehat{w}}$ and thus

$$\mathrm{tr}(\widehat{A}\widehat{C}_{\widehat{w},\widehat{w}})=\mathrm{tr}(I_b)-\mathrm{tr}(E\widehat{C}_{\widehat{w},\widehat{w}}^{-1}E^\top(E\widehat{C}_{\widehat{w},\widehat{w}}^{-1}E^\top)^{-1})=b-|\widehat{\mathcal{A}}|,$$

which is the dimension of the *face* on which $\widehat{\alpha}(\lambda)$ lies.

2.4.2 CROSS-VALIDATION

Although the information criterion derived above is more accurate than just a naive empirical estimator, its accuracy is guaranteed only asymptotically. Here, we employ cross-validation for estimating $J(\widehat{\alpha})$, which has an accuracy guarantee for finite samples.

First, the training samples $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and test samples $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ are divided into R disjoint subsets $\{\mathcal{X}_r^{\mathrm{tr}}\}_{r=1}^R$ and $\{\mathcal{X}_r^{\mathrm{te}}\}_{r=1}^R$, respectively. Then an importance estimate $\widehat{w}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}(x)$ is obtained using $\{\mathcal{X}_j^{\mathrm{tr}}\}_{j\neq r}$ and $\{\mathcal{X}_j^{\mathrm{te}}\}_{j\neq r}$ (that is, without $\mathcal{X}_r^{\mathrm{tr}}$ and $\mathcal{X}_r^{\mathrm{te}}$), and the cost J is approximated using the held-out samples $\mathcal{X}_r^{\mathrm{tr}}$ and $\mathcal{X}_r^{\mathrm{te}}$ as

$$\widehat{J}^{(\mathrm{CV})}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}=\frac{1}{2|\mathcal{X}_r^{\mathrm{tr}}|}\sum_{x^{\mathrm{tr}}\in\mathcal{X}_r^{\mathrm{tr}}}\widehat{w}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}(x^{\mathrm{tr}})^2-\frac{1}{|\mathcal{X}_r^{\mathrm{te}}|}\sum_{x^{\mathrm{te}}\in\mathcal{X}_r^{\mathrm{te}}}\widehat{w}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}(x^{\mathrm{te}}).$$

This procedure is repeated for $r=1,2,\ldots,R$ and its average $\widehat{J}^{(\mathrm{CV})}$ is used as an estimate of J:

$$\widehat{J}^{(\mathrm{CV})}=\frac{1}{R}\sum_{r=1}^R\widehat{J}^{(\mathrm{CV})}_{\mathcal{X}_r^{\mathrm{tr}},\mathcal{X}_r^{\mathrm{te}}}.$$

We can show that $\widehat{J}^{(\mathrm{CV})}$ gives an almost unbiased estimate of the true cost J, where the 'almost'-ness comes from the fact that the number of samples is reduced in the cross-validation procedure due to data splitting (Luntz and Brailovsky, 1969; Wahba, 1990; Schölkopf and Smola, 2002).

Cross-validation would be more accurate than the information criterion for finite samples. However, it is computationally more expensive than the information criterion since the learning procedure should be repeated R times.

2.5 Heuristics of Basis Function Design for LSIF

A good model may be chosen by cross-validation or the information criterion, given that a family of promising model candidates is prepared. As model candidates, we propose using a Gaussian kernel model centered at the *test* points $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, that is,

$$\widehat{w}(x)=\sum_{\ell=1}^{n_{\mathrm{te}}}\alpha_\ell K_\sigma(x,x_\ell^{\mathrm{te}}),$$

where $K_\sigma(x,x')$ is the Gaussian kernel with kernel width σ:

$$K_\sigma(x,x')=\exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right). \tag{21}$$

The reason why we chose the test points $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ as the Gaussian centers, not the training points $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$, is as follows (Sugiyama et al., 2008b). By definition, the importance $w(x)$ tends to take

large values if the training density $p_{\mathrm{tr}}(x)$ is small and the test density $p_{\mathrm{te}}(x)$ is large; conversely, $w(x)$ tends to be small (that is, close to zero) if $p_{\mathrm{tr}}(x)$ is large and $p_{\mathrm{te}}(x)$ is small. When a function is approximated by a Gaussian kernel model, many kernels may be needed in the region where the output of the target function is large; on the other hand, only a small number of kernels would be enough in the region where the output of the target function is close to zero. Following this heuristic, we allocate many kernels at high *test* density regions, which can be achieved by setting the Gaussian centers at the test points $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$.

Alternatively, we may locate $(n_{\mathrm{tr}}+n_{\mathrm{te}})$ Gaussian kernels at both $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$. However, in our preliminary experiments, this did not further improve the performance, but just slightly increased the computational cost. When n_{te} is large, just using all the test points $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ as Gaussian centers is already computationally rather demanding. To ease this problem, we practically propose using a subset of $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ as Gaussian centers for computational efficiency, that is,

$$\widehat{w}(x) = \sum_{\ell=1}^{b} \alpha_\ell K_\sigma(x, c_\ell), \tag{22}$$

where $c_\ell,\ \ell = 1,2,\ldots,b$ are template points randomly chosen from $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ without replacement and $b\ (\leq n_{\mathrm{te}})$ is a prefixed number. In the rest of this paper, we usually fix the number of template points at

$$b = \min(100, n_{\mathrm{te}}),$$

and optimize the kernel width σ and the regularization parameter λ by cross-validation with grid search.

2.6 Entire Regularization Path for LSIF

We can show that the LSIF solution $\widehat{\alpha}$ is piecewise linear with respect to the regularization parameter λ (see Appendix D). Therefore, the *regularization path* (that is, solutions for all λ) can be computed efficiently based on the *parametric optimization technique* (Best, 1982; Efron et al., 2004; Hastie et al., 2004).

A basic idea of regularization path tracking is to check violation of the KKT conditions—which are necessary and sufficient for optimality of convex programs—when the regularization parameter λ is changed. The KKT conditions of LSIF are summarized in Section 2.3. The strict complementarity condition (17) assures the uniqueness of the optimal solution for a fixed λ, and thus the uniqueness of the regularization path. A pseudo code of the regularization path tracking algorithm for LSIF is described in Figure 1—its detailed derivation is summarized in Appendix D. Thanks to the regularization path algorithm, LSIF is computationally efficient in model selection scenarios.

The pseudo code implies that we no longer need a quadratic programming solver for obtaining the solution of LSIF—just computing matrix inverses is enough. Furthermore, the regularization path algorithm is computationally more efficient when the solution is sparse, that is, most of the elements are zero since the number of change points tends to be small for such sparse solutions.

Even though the regularization path tracking algorithm is computationally efficient, it tends to be numerically unreliable, as we experimentally show in Section 4. This numerical instability is caused by near singularity of the matrix $\widehat{G}$. When $\widehat{G}$ is nearly singular, it is not easy to accurately obtain the solutions u, v in Figure 1, and therefore the change point $\lambda_{\tau+1}$ cannot be accurately computed. As a result, we cannot accurately update the active set of the inequality constraints and thus

Input: $\widehat{H}$ and $\widehat{h}$ % see Eqs. (4) and (5) for the definitions
Output: entire regularization path $\widehat{\alpha}(\lambda)$ for $\lambda \geq 0$

$\tau \longleftarrow 0$;
$k \longleftarrow \mathrm{argmax}_i\{\widehat{h}_i \mid i = 1,2,\ldots,b\}$;
$\lambda_\tau \longleftarrow \widehat{h}_k$;
$\widehat{\mathcal{A}} \longleftarrow \{1,2,\ldots,b\}\backslash\{k\}$;
$\widehat{\alpha}(\lambda_\tau) \longleftarrow 0_b$; % the vector with all zeros
While $\lambda_\tau > 0$
  $\widehat{E} \longleftarrow O_{|\widehat{\mathcal{A}}|\times b}$; % the matrix with all zeros
  For $i = 1,2,\ldots,|\widehat{\mathcal{A}}|$
    $\widehat{E}_{i,\widehat{j}_i} \longleftarrow 1$; % $\widehat{\mathcal{A}} = \{\widehat{j}_1, \widehat{j}_2, \ldots, \widehat{j}_{|\widehat{\mathcal{A}}|} \mid \widehat{j}_1 < \widehat{j}_2 < \cdots < \widehat{j}_{|\widehat{\mathcal{A}}|}\}$
  end
  $\widehat{G} \longleftarrow \begin{pmatrix} \widehat{H} & -\widehat{E}^\top \\ -\widehat{E} & O_{|\widehat{\mathcal{A}}|\times|\widehat{\mathcal{A}}|} \end{pmatrix}$;
  $u \longleftarrow \widehat{G}^{-1}\begin{pmatrix} \widehat{h} \\ 0_{|\widehat{\mathcal{A}}|} \end{pmatrix}$;
  $v \longleftarrow \widehat{G}^{-1}\begin{pmatrix} 1_b \\ 0_{|\widehat{\mathcal{A}}|} \end{pmatrix}$;
  If $v \leq 0_{b+|\widehat{\mathcal{A}}|}$ % the final interval
    $\lambda_{\tau+1} \longleftarrow 0$;
    $\widehat{\alpha}(\lambda_{\tau+1}) \longleftarrow (u_1, u_2, \ldots, u_b)^\top$;
  else % an intermediate interval
    $k \longleftarrow \mathrm{argmax}_i\{u_i/v_i \mid v_i > 0,\ i = 1,2,\ldots,b+|\widehat{\mathcal{A}}|\}$;
    $\lambda_{\tau+1} \longleftarrow \max\{0, u_k/v_k\}$;
    $\widehat{\alpha}(\lambda_{\tau+1}) \longleftarrow (u_1, u_2, \ldots, u_b)^\top - \lambda_{\tau+1}(v_1, v_2, \ldots, v_b)^\top$;
    If $1 \leq k \leq b$
      $\widehat{\mathcal{A}} \longleftarrow \widehat{\mathcal{A}} \cup \{k\}$;
    else
      $\widehat{\mathcal{A}} \longleftarrow \widehat{\mathcal{A}}\backslash\{\widehat{j}_{k-b}\}$;
    end
  end
  $\tau \longleftarrow \tau + 1$;
end
$$\widehat{\alpha}(\lambda) \longleftarrow \begin{cases} 0_b & \text{if } \lambda \geq \lambda_0 \\ \frac{\lambda_{\tau+1}-\lambda}{\lambda_{\tau+1}-\lambda_\tau}\widehat{\alpha}(\lambda_\tau) + \frac{\lambda-\lambda_\tau}{\lambda_{\tau+1}-\lambda_\tau}\widehat{\alpha}(\lambda_{\tau+1}) & \text{if } \lambda_{\tau+1} \leq \lambda \leq \lambda_\tau \end{cases}$$

Figure 1: Pseudo code for computing the entire regularization path of LSIF. When the computation of $\widehat{G}^{-1}$ is numerically unstable, we may add small positive diagonals to $\widehat{H}$ for stabilization purposes.

the obtained solution $\widehat{\alpha}(\lambda)$ becomes unreliable; furthermore, such numerical error tends to be accumulated through the path-tracking process. This instability issue seems to be a common pitfall of solution path tracking algorithms in general (see Scheinberg, 2006).

When the Gaussian width σ is very small or very large, the matrix $\widehat{H}$ tends to be nearly singular and thus the matrix $\widehat{G}$ also becomes nearly singular. On the other hand, when the Gaussian width σ is not too small or too large compared with the dispersion of samples, the matrix $\widehat{G}$ is well-conditioned and therefore the path-following algorithm would be stable and reliable.

3. Approximation Algorithm

Within the quadratic programming formulation, we have proposed a new importance estimation procedure LSIF and showed its theoretical properties. We also gave a regularization path tracking algorithm that can be computed efficiently. However, as we experimentally show in Section 4, it tends to suffer from a numerical problem and therefore is not practically reliable. In this section, we give a practical alternative to LSIF which gives an approximate solution to LSIF in a computationally efficient and reliable manner.

3.1 Unconstrained Least-squares Formulation

The approximation idea we introduce here is very simple: we ignore the non-negativity constraint of the parameters in the optimization problem (6). This results in the following unconstrained optimization problem.

$$\min_{\beta\in\mathbb{R}^b}\left[\frac{1}{2}\beta^\top\widehat{H}\beta-\widehat{h}^\top\beta+\frac{\lambda}{2}\beta^\top\beta\right]. \tag{23}$$

In the above, we included a quadratic regularization term $\beta^\top\beta/2$, instead of the linear one $1_b^\top\beta$ since the linear penalty term does not work as a regularizer without the non-negativity constraint. Eq. (23) is an unconstrained convex quadratic program, so the solution can be analytically computed as

$$\widetilde{\beta}(\lambda)=(\widehat{H}+\lambda I_b)^{-1}\widehat{h},$$

where I_b is the b-dimensional identity matrix. Since we dropped the non-negativity constraint $\beta\geq 0_b$, some of the learned parameters could be negative. To compensate for this approximation error, we modify the solution by

$$\widehat{\beta}(\lambda)=\max(0_b,\widetilde{\beta}(\lambda)),$$

where the 'max' operation for a pair of vectors is applied in the element-wise manner. This is the solution of the approximation method we propose in this section.

An advantage of the above unconstrained formulation is that the solution can be computed just by solving a system of linear equations. Therefore, its computation is stable when λ is not too small. We call this method *unconstrained LSIF* (uLSIF). Due to the ℓ_2 regularizer, the solution tends to be close to 0_b to some extent. Thus, the effect of ignoring the non-negativity constraint may not be so strong—later, we analyze the approximation error both theoretically and experimentally in more detail in Sections 3.3 and 4.5.

Note that LSIF and uLSIF differ only in parameter learning. Thus, the basis design heuristic of LSIF given in Section 2.5 is also valid for uLSIF.

3.2 Convergence Analysis of uLSIF

Here, we theoretically analyze the convergence property of the solution $\widehat{\beta}(\lambda)$ of the uLSIF algorithm; practitioners may skip Sections 3.2 and 3.3.

Let $\beta^\circ(\lambda)$ be the optimal solution of the 'ideal' version of the problem (23):

$$\min_{\beta\in\mathbb{R}^b}\left[\frac{1}{2}\beta^\top H\beta - h^\top\beta + \frac{\lambda}{2}\beta^\top\beta\right].$$

Then the ideal solution $\beta^*(\lambda)$ is given by

$$\begin{aligned}\beta^*(\lambda) &= \max(0_b, \beta^\circ(\lambda)),\\ \beta^\circ(\lambda) &= B_\lambda^{-1}h,\\ B_\lambda &= H + \lambda I_b.\end{aligned} \tag{24}$$

Below, we theoretically investigate the learning curve of uLSIF.

Let $\mathcal{B}\subset\{1,2,\ldots,b\}$ be the set of negative indices of $\beta^\circ(\lambda)$, that is,

$$\mathcal{B} = \{\ell \mid \beta^\circ_\ell(\lambda) < 0,\ \ell = 1,2,\ldots,b\},$$

and $\widetilde{\mathcal{B}}\subset\{1,2,\ldots,b\}$ be the set of negative indices of $\widetilde{\beta}(\lambda)$, that is,

$$\widetilde{\mathcal{B}} = \{\ell \mid \widetilde{\beta}_\ell(\lambda) < 0,\ \ell = 1,2,\ldots,b\}.$$

Then we have the following theorem.

Theorem 4 *Assume that $\beta^\circ_\ell(\lambda)\neq 0$ for $\ell = 1,2,\ldots,b$. Then, there exists a positive constant c and a natural number N such that for $\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}\geq N$,*

$$P(\mathcal{B}\neq\widetilde{\mathcal{B}}) < e^{-c\min\{n_{\mathrm{tr}},n_{\mathrm{te}}\}}.$$

The proof of the above theorem is given in Appendix E. The assumption that $\beta^\circ_\ell(\lambda)\neq 0$ for $\ell = 1,2,\ldots,b$ corresponds to the strict complementarity condition (17) in LSIF. Theorem 4 shows that the probability that $\widetilde{\mathcal{B}}$ is different from $\mathcal{B}$ is exponentially small. Thus we may regard $\widetilde{\mathcal{B}} = \mathcal{B}$ in practice.

Let D be the b-dimensional diagonal matrix with the ℓ-th diagonal element

$$D_{\ell,\ell} = \begin{cases} 0 & \ell\in\mathcal{B},\\ 1 & \text{otherwise.}\end{cases}$$

Let

$$\begin{aligned} w^\circ(x) &= \sum_{\ell=1}^b \beta^\circ_\ell(\lambda)\varphi_\ell(x),\\ u(x) &= \sum_{\ell=1}^b [B_\lambda^{-1}D(H\beta^*(\lambda)-h)]_\ell\varphi_\ell(x).\end{aligned}$$

Then we have the following theorem.

Theorem 5 *Assume that*

(a) $\beta^{\circ}_{\ell}(\lambda) \neq 0$ *for* $\ell = 1, 2, \ldots, b$.

(b) n_{tr} *and* n_{te} *satisfy Eq.* (18).

Then, for any $\lambda \geq 0$, *we have*

$$\mathbb{E}[J(\widehat{\beta}(\lambda))] = J(\beta^{*}(\lambda)) + \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(B_{\lambda}^{-1}DHDB_{\lambda}^{-1}C_{w^{\circ},w^{\circ}} + 2B_{\lambda}^{-1}C_{w^{\circ},u}) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{25}$$

The proof of the above theorem is given in Appendix F. Theorem 5 elucidates the learning curve of uLSIF up to the order of n_{tr}^{-1}. An information criterion may be obtained in the same way as Section 2.4.1. However, as shown in Section 3.4, we can have a closed-form expression of the leave-one-out cross-validation score for uLSIF, which would be practically more useful. For this reason, we do not go into the detail of information criterion.

3.3 Approximation Error Bounds for uLSIF

The uLSIF method is introduced as an approximation of LSIF. Here, we theoretically evaluate the difference between the uLSIF solution $\widehat{\beta}(\lambda)$ and the LSIF solution $\widehat{\alpha}(\lambda)$. More specifically, we use the following normalized L_2-norm on the training samples as the difference measure and derive its upper bounds:

$$\mathrm{diff}(\lambda) = \frac{\inf_{\lambda' \geq 0}\sqrt{\frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}}\left(\widehat{w}(x_i^{\mathrm{tr}};\widehat{\alpha}(\lambda')) - \widehat{w}(x_i^{\mathrm{tr}};\widehat{\beta}(\lambda))\right)^2}}{\sum_{i=1}^{n_{\mathrm{tr}}}\widehat{w}(x_i^{\mathrm{tr}};\widehat{\beta}(\lambda))}, \tag{26}$$

where the importance function $\widehat{w}(x;\alpha)$ is given by

$$\widehat{w}(x;\alpha) = \sum_{\ell=1}^{b}\alpha_{\ell}\varphi_{\ell}(x).$$

In the theoretical analysis below, we assume

$$\sum_{i=1}^{n_{\mathrm{tr}}}\widehat{w}(x_i^{\mathrm{tr}};\widehat{\beta}(\lambda)) \neq 0.$$

For $p \in \mathbb{N} \cup \{\infty\}$, let $\|\cdot\|_p$ be the L_p-norm, and let $\|\alpha\|_{\widehat{H}}$ be

$$\|\alpha\|_{\widehat{H}} = \sqrt{\alpha^{\top}\widehat{H}\alpha}, \tag{27}$$

where $\widehat{H}$ is the $b \times b$ matrix defined by Eq. (4). Then we have the following theorem.

Theorem 6 (Norm bound) *Assume that all basis functions satisfy*

$$0 < \varphi_{\ell}(x) \leq 1.$$

Then we have

$$\mathrm{diff}(\lambda) \;\leq\; \frac{\|\widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))} \tag{28}$$

$$\leq\; b^2 \left(1+\frac{b}{\lambda}\right) \frac{1}{\min_\ell \sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}})} \cdot \frac{n_{\mathrm{te}}}{\min_\ell \sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}})}, \tag{29}$$

where b is the number of basis functions. The upper bound (29) *is reduced as the regularization parameter λ increases. For the Gaussian basis function model* (22)*, the upper bound (29) is reduced as the Gaussian width σ increases.*

The proof of the above theorem is given in Appendix G. We call Eq. (28) the *norm bound* since it is governed by the norm of $\widehat{\beta}$. Intuitively, the approximation error of uLSIF would small if λ is large since $\widetilde{\beta} \geq 0$ may not be severely violated due to the strong regularization effect. The upper bound (29) justifies this intuitive claim since the error bound tends to be small if the regularization parameter λ is large. Furthermore, the upper bound (29) shows that for the Gaussian basis function model (22), the error bound tends to be small if the Gaussian width σ is large. This is also intuitive since the Gaussian basis functions are nearly flat when the Gaussian width σ is large—a difference in parameters does not cause a significant change in the learned importance function $\widehat{w}(x)$. From the above theorem, we expect that uLSIF is a nice approximation of LSIF when λ is large and σ is large. In Section 4.5, we numerically investigate this issue.

Below, we give a more sophisticated bound on $\mathrm{diff}(\lambda)$. To this end, let us introduce an intermediate optimization problem defined by

$$\min_{\gamma \in \mathbb{R}^b} \left[\frac{1}{2}\gamma^\top \widehat{H}\gamma - \widehat{h}^\top \gamma + \frac{\lambda}{2}\gamma^\top \gamma\right]$$
$$\text{subject to } \gamma \geq 0_b, \tag{30}$$

which we refer to as *LSIF with quadratic penalty* (LSIFq). LSIFq bridges LSIF and uLSIF since the 'goodness-of-fit' part is the same as LSIF but the 'regularization' part is the same as uLSIF. Let $\widehat{\gamma}(\lambda)$ be the optimal solution of LSIFq (30). Based on the solution of LSIFq, we have the following upper bound.

Theorem 7 (Bridge bound) *For any $\lambda \geq 0$, the following inequality holds:*

$$\mathrm{diff}(\lambda) \;\leq\; \frac{\sqrt{\lambda\left(\|\widehat{\gamma}(\lambda)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty - \|\widehat{\gamma}(\lambda)\|_2^2\right)} + \|\widehat{\gamma}(\lambda) - \widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))}. \tag{31}$$

The proof of the above theorem is given in Appendix H. We call the above bound the *bridge bound* since the bridged estimator $\widehat{\gamma}(\lambda)$ plays a central role in the bound. Note that, in the bridge bound, the inside of the square root is assured to be non-negative due to Hölder's inequality (see Appendix H for detail). The bridge bound is generally much sharper than the norm bound (28), but not always (see Section 4.5 for numerical examples).

3.4 Efficient Computation of Leave-one-out Cross-validation Score for uLSIF

A practically important advantage of uLSIF over LSIF is that the score of leave-one-out cross-validation (LOOCV) can be computed analytically—thanks to this property, the computational complexity for performing LOOCV is the same order as just computing a single solution.

In the current setup, we are given two sets of samples, $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, which generally have different sample size. For simplicity, we assume that $n_{\mathrm{tr}} < n_{\mathrm{te}}$ and the i-th training sample x_i^{tr} and the i-th test sample x_i^{te} are held out at the same time; the test samples $\{x_j^{\mathrm{te}}\}_{j=n_{\mathrm{tr}}+1}^{n_{\mathrm{te}}}$ are always used for importance estimation. Note that this assumption is only for the sake of simplicity; we can change the order of test samples without sacrificing the computational advantages.

Let $\widehat{w}^{(i)}(x)$ be an estimate of the importance obtained without the i-th training sample x_i^{tr} and the i-th test sample x_i^{te}. Then the LOOCV score is expressed as

$$\mathrm{LOOCV} = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left[\frac{1}{2} (\widehat{w}^{(i)}(x_i^{\mathrm{tr}}))^2 - \widehat{w}^{(i)}(x_i^{\mathrm{te}}) \right]. \tag{32}$$

Our approach to efficiently computing the LOOCV score is to use the *Sherman-Woodbury-Morrison formula* (Golub and Loan, 1996) for computing matrix inverses: for an invertible square matrix A and vectors u and v such that $v^\top A^{-1} u \neq -1$,

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1} u v^\top A^{-1}}{1 + v^\top A^{-1} u}. \tag{33}$$

Efficient approximation schemes of LOOCV have often been investigated under asymptotic setups (Stone, 1974; Hansen and Larsen, 1996). On the other hand, we provide the exact LOOCV score of uLSIF, which follows the same line as that of ridge regression (Hoerl and Kennard, 1970; Wahba, 1990).

A pseudo code of uLSIF with LOOCV-based model selection is summarized in Figure 2—its detailed derivation is described in Appendix I. Note that the basis-function design heuristic given in Section 2.5 is used in the pseudo code, but the analytic form of the LOOCV score is available for any basis functions.

4. Illustrative Examples

In this section, we illustrate the behavior of LSIF and uLSIF using a toy data set.

4.1 Setup

Let the dimension of the domain be $d = 1$ and the training and test densities be

$$p_{\mathrm{tr}}(x) = \mathcal{N}(x; 1, (1/2)^2),$$
$$p_{\mathrm{te}}(x) = \mathcal{N}(x; 2, (1/4)^2),$$

where $\mathcal{N}(x; \mu, \sigma^2)$ denotes the Gaussian density with mean μ and variance σ^2. These densities are depicted in Figure 3. The task is to estimate the importance $w(x) = p_{\mathrm{te}}(x)/p_{\mathrm{tr}}(x)$.

Input: $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$
Output: $\widehat{w}(x)$

$b \longleftarrow \min(100, n_{\mathrm{te}})$; $\quad n \longleftarrow \min(n_{\mathrm{tr}}, n_{\mathrm{te}})$;
Randomly choose b centers $\{c_\ell\}_{\ell=1}^{b}$ from $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ without replacement;
For each candidate of Gaussian width σ

$$\widehat{H}_{\ell,\ell'} \longleftarrow \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \exp\left(-\frac{\|x_i^{\mathrm{tr}}-c_\ell\|^2+\|x_i^{\mathrm{tr}}-c_{\ell'}\|^2}{2\sigma^2}\right) \text{ for } \ell,\ell'=1,2,\ldots,b;$$

$$\widehat{h}_\ell \longleftarrow \frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}} \exp\left(-\frac{\|x_j^{\mathrm{te}}-c_\ell\|^2}{2\sigma^2}\right) \text{ for } \ell=1,2,\ldots,b;$$

$$X^{\mathrm{tr}}_{\ell,i} \longleftarrow \exp\left(-\frac{\|x_i^{\mathrm{tr}}-c_\ell\|^2}{2\sigma^2}\right) \text{ for } i=1,2,\ldots,n \text{ and } \ell=1,2,\ldots,b;$$

$$X^{\mathrm{te}}_{\ell,i} \longleftarrow \exp\left(-\frac{\|x_i^{\mathrm{te}}-c_\ell\|^2}{2\sigma^2}\right) \text{ for } i=1,2,\ldots,n \text{ and } \ell=1,2,\ldots,b;$$

For each candidate of regularization parameter λ

$$\widehat{B} \longleftarrow \widehat{H} + \frac{\lambda(n_{\mathrm{tr}}-1)}{n_{\mathrm{tr}}} I_b;$$

$$B_0 \longleftarrow \widehat{B}^{-1}\widehat{h}1_n^\top + \widehat{B}^{-1}X^{\mathrm{tr}}\,\mathrm{diag}\left(\frac{\widehat{h}^\top\widehat{B}^{-1}X^{\mathrm{tr}}}{n_{\mathrm{tr}}1_n^\top - 1_b^\top(X^{\mathrm{tr}}*\widehat{B}^{-1}X^{\mathrm{tr}})}\right);$$

$$B_1 \longleftarrow \widehat{B}^{-1}X^{\mathrm{te}} + \widehat{B}^{-1}X^{\mathrm{tr}}\,\mathrm{diag}\left(\frac{1_b^\top(X^{\mathrm{te}}*\widehat{B}^{-1}X^{\mathrm{tr}})}{n_{\mathrm{tr}}1_n^\top - 1_b^\top(X^{\mathrm{tr}}*\widehat{B}^{-1}X^{\mathrm{tr}})}\right);$$

$$B_2 \longleftarrow \max\left(O_{b\times n}, \frac{n_{\mathrm{tr}}-1}{n_{\mathrm{tr}}(n_{\mathrm{te}}-1)}(n_{\mathrm{te}}B_0 - B_1)\right);$$

$$w_{\mathrm{tr}} \longleftarrow (1_b^\top(X^{\mathrm{tr}}*B_2))^\top; \quad w_{\mathrm{te}} \longleftarrow (1_b^\top(X^{\mathrm{te}}*B_2))^\top;$$

$$\mathrm{LOOCV}(\sigma,\lambda) \longleftarrow \frac{w_{\mathrm{tr}}^\top w_{\mathrm{tr}}}{2n} - \frac{1_n^\top w_{\mathrm{te}}}{n};$$

end
end

$$(\widehat{\sigma},\widehat{\lambda}) \longleftarrow \mathrm{argmin}_{(\sigma,\lambda)}\,\mathrm{LOOCV}(\sigma,\lambda);$$

$$\widetilde{H}_{\ell,\ell'} \longleftarrow \frac{1}{n_{\mathrm{tr}}}\sum_{i=1}^{n_{\mathrm{tr}}} \exp\left(-\frac{\|x_i^{\mathrm{tr}}-c_\ell\|^2+\|x_i^{\mathrm{tr}}-c_{\ell'}\|^2}{2\widehat{\sigma}^2}\right) \text{ for } \ell,\ell'=1,2,\ldots,b;$$

$$\widetilde{h}_\ell \longleftarrow \frac{1}{n_{\mathrm{te}}}\sum_{j=1}^{n_{\mathrm{te}}} \exp\left(-\frac{\|x_j^{\mathrm{te}}-c_\ell\|^2}{2\widehat{\sigma}^2}\right) \text{ for } \ell=1,2,\ldots,b;$$

$$\widehat{\alpha} \longleftarrow \max(0_b, (\widetilde{H}+\widehat{\lambda}I_b)^{-1}\widetilde{h});$$

$$\widehat{w}(x) \longleftarrow \sum_{\ell=1}^{b}\widehat{\alpha}_\ell \exp\left(-\frac{\|x-c_\ell\|^2}{2\widehat{\sigma}^2}\right);$$

Figure 2: Pseudo code of uLSIF algorithm with LOOCV. $B*B'$ denotes the element-wise multiplication of matrices B and B' of the same size, that is, the (i,j)-th element is given by $B_{i,j}B'_{i,j}$. For n-dimensional vectors b and b', $\mathrm{diag}\left(\frac{b}{b'}\right)$ denotes the $n\times n$ diagonal matrix with i-th diagonal element b_i/b'_i. A MATLAB® or R implementation of uLSIF is available from `http://sugiyama-www.cs.titech.ac.jp/~sugi/software/uLSIF/`.

4.2 Importance Estimation

First, we illustrate the behavior of LSIF and uLSIF in importance estimation. We set the number of training and test samples at $n_{\mathrm{tr}} = 200$ and $n_{\mathrm{te}} = 1000$, respectively. We use the Gaussian kernel model (22), and the number of basis functions is set at $b = 100$. The centers of the kernel function are randomly chosen from the test points $\{x_i^{\mathrm{te}}\}_{j=1}^{n^{\mathrm{te}}}$ without replacement (see Section 2.5).

We test different Gaussian widths σ and different regularization parameters λ. The following two setups are examined:

(A) λ is fixed at $\lambda = 0.2$ and σ is changed as $0.1 \leq \sigma \leq 1.0$,

(B) σ is fixed at $\sigma = 0.3$ and λ is changed as $0 \leq \lambda \leq 0.5$.

Figure 4 depicts the true importance and its estimates obtained by LSIF and uLSIF, where all importance functions are normalized so that $\int w(x)dx = 1$ for better comparison. Figures 4(a) and 4(b) show that the estimated importance $\widehat{w}(x)$ tends to be too peaky when the Gaussian width σ is small, while it tends to be overly smoothed when σ is large. If the Gaussian width is chosen appropriately, both LSIF and uLSIF seem to work reasonably well. As shown in Figures 4(c) and 4(d), the solutions of LSIF and uLSIF also significantly change when different regularization parameters λ are used. Again, given that the regularization parameter is chosen appropriately, both LSIF and uLSIF tend to perform well.

From the graphs, we also observe that model selection based on cross-validation works reasonably well for both LSIF (5-fold) and uLSIF (leave-one-out) to choose appropriate values of the Gaussian width or the regularization parameter; this will be analyzed in more detail in Section 4.4.

4.3 Regularization Path

Next, we illustrate how the regularization path tracking algorithm for LSIF behaves. We set the number of training and test samples at $n_{\mathrm{tr}} = 50$ and $n_{\mathrm{te}} = 100$, respectively. For better illustration, we set the number of basis functions at a small value as $b = 30$ in the Gaussian kernel model (22) and use the Gaussian kernels centered at equidistant points in $[0,3]$ as basis functions.

We use the algorithm described in Figure 1 for regularization path tracking. Theoretically, the inequality $\lambda_{\tau+1} < \lambda_\tau$ is assured. In numerical computation, however, the inequality is occasionally violated. In order to avoid this numerical problem, we slightly regularize $\widehat{H}$ for stabilization (see also the caption of Figure 1).

Figure 5 depicts the values of the estimated coefficients $\{\alpha_\ell\}_{\ell=1}^{b}$ as functions of $\|\alpha\|_1$ for $\sigma = 0.1, 0.3$, and 0.5. Note that small $\|\alpha\|_1$ corresponds to large λ. The figure indicates that the regularization parameter λ works as a sparseness controlling factor of the solution, that is, the larger (smaller) the value of λ ($\|\alpha\|_1$) is, the sparser the solution is.

The path following algorithm is computationally efficient and therefore practically very attractive. However, as the above experiments illustrate, the path following algorithm is numerically rather unstable. Modification of $\widehat{H}$ can ease to solve this problem, but this in turn results in accumulating numerical errors through the path tracking process. Consequently, the solutions for small λ tend to be inaccurate. This problem becomes prominent if the number of change points in the regularization path is large.

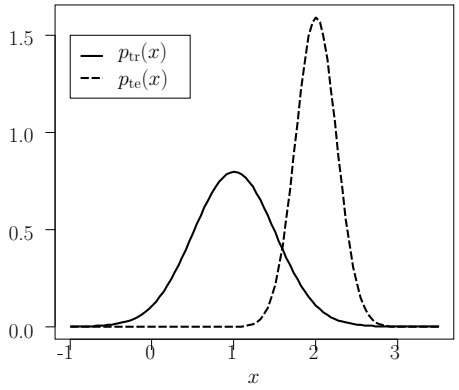
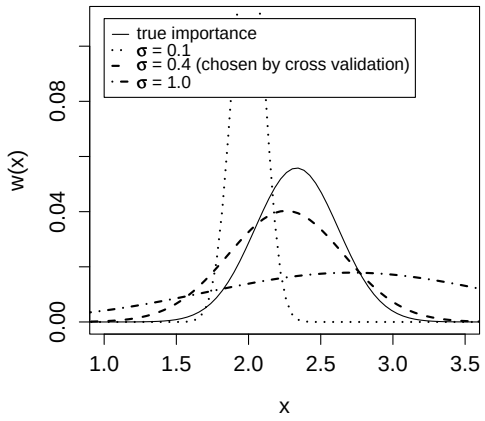
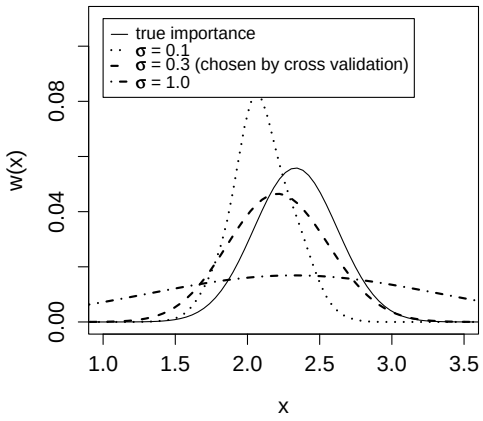


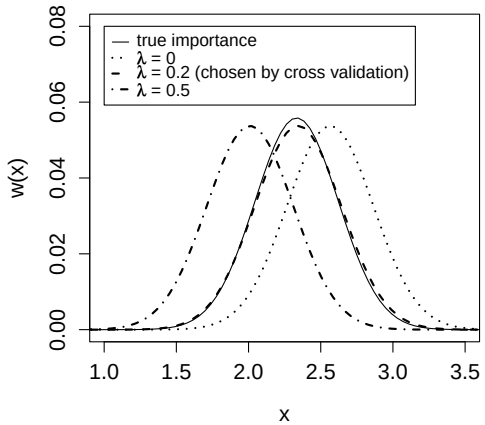
Figure 3: The solid line is the probability density of training data, and the dashed line is the probability density of test data.



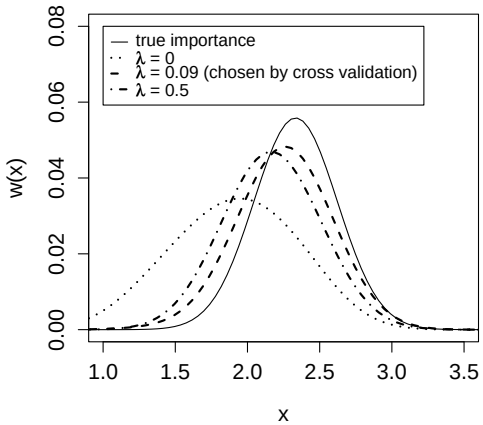
(a) LSIF for $\lambda = 0.2$, $\sigma = 0.1, 0.4, 1.0$.



(b) uLSIF for $\lambda = 0.2$, $\sigma = 0.1, 0.3, 1.0$.

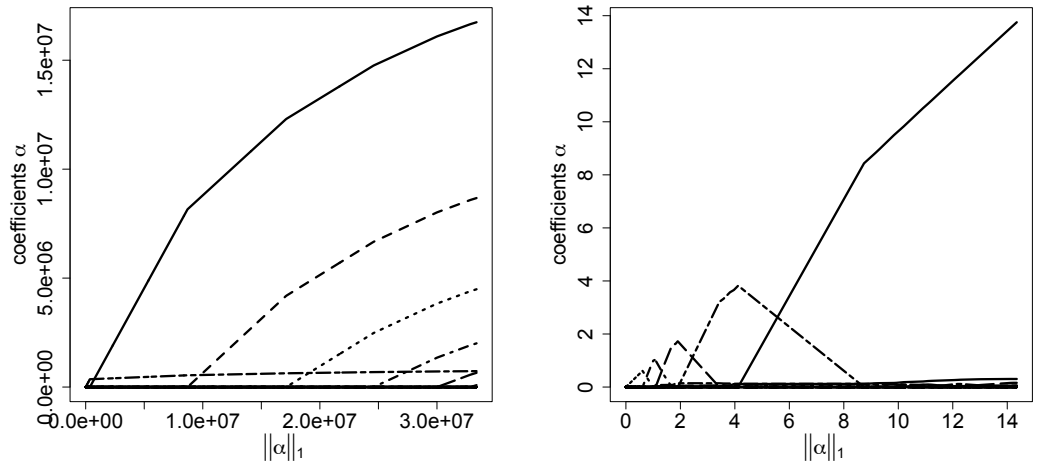


(c) LSIF for $\sigma = 0.3$, $\lambda = 0, 0.2, 0.5$.

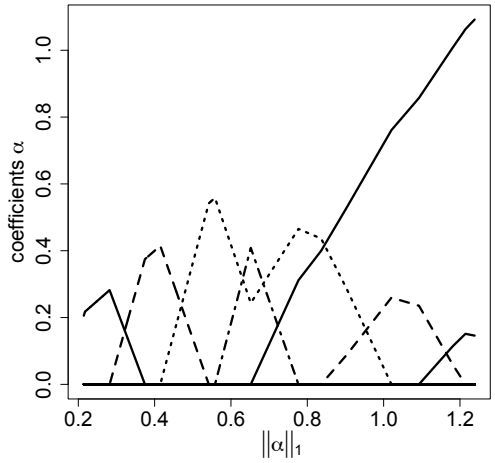


(d) uLSIF for $\sigma = 0.3$, $\lambda = 0, 0.09, 0.5$.

Figure 4: True and estimated importance functions obtained by LSIF and uLSIF for various different Gaussian widths σ and regularization parameters λ.



(a) $\sigma = 0.1$. (b) $\sigma = 0.3$.

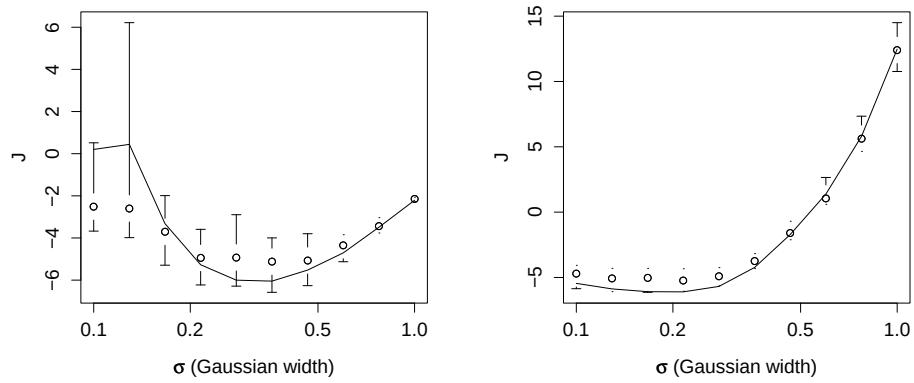


(c) $\sigma = 0.5$.

Figure 5: Regularization path of LSIF: the values of the estimated coefficients $\{\alpha_\ell\}_{\ell=1}^b$ are depicted as functions of the L_1-norm of the estimated parameter vector for $\sigma = 0.1, 0.3$, and 0.5. Small $\|\alpha\|_1$ corresponds to large λ.

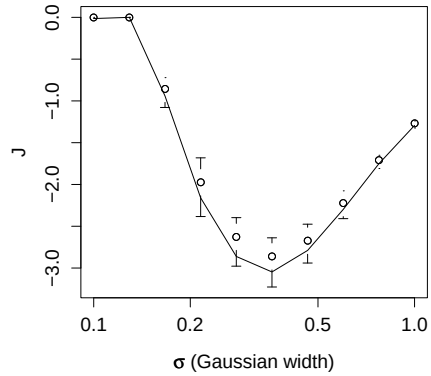
4.4 Cross-validation

Here we illustrate the behavior of the cross-validation scores of LSIF and uLSIF. We set the number of training and test samples at $n_{\mathrm{tr}} = 200$ and $n_{\mathrm{te}} = 1000$, respectively. The number of template points is $b = 100$ and the Gaussian kernel model (22) is used. The centers of the kernel functions are randomly chosen from the test points as described in Section 4.2. The left column of Figure 6 depicts the expectation of the true cost $J(\widehat{\alpha})$ over 50 runs for LSIF and its estimate by 5-fold CV (25, 50, and 75 percentiles are plotted in the figure) as functions of the Gaussian width σ for $\lambda = 0.2$, 0.5, and 0.8. We used the regularization path tracking algorithm for computing the solutions of LSIF.

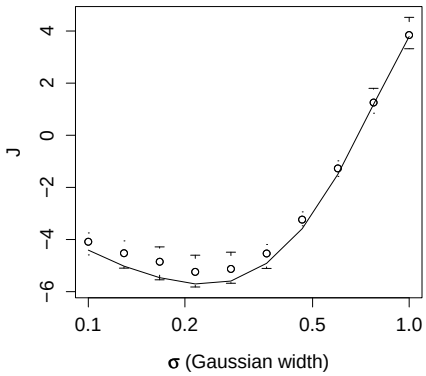


(a) LSIF with 5CV ($\lambda = 0.2$).

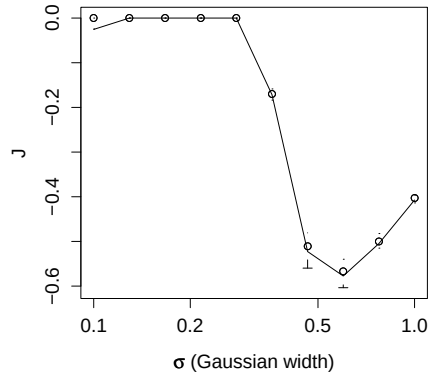
(b) uLSIF with LOOCV ($\lambda = 0.2$).



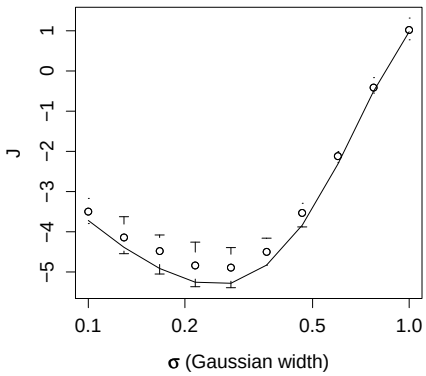
(c) LSIF with 5CV ($\lambda = 0.5$).



(d) uLSIF with LOOCV ($\lambda = 0.5$).



(e) LSIF with 5CV ($\lambda = 0.8$).



(f) uLSIF with LOOCV ($\lambda = 0.8$).

Figure 6: The true cost J and its cross-validation estimate as functions of Gaussian width σ for different values of λ. The solid line denotes the expectation of the true cost J over 50 runs, while '$\circ$' and error bars denote the 25, 50, and 75 percentiles of the cross-validation score.

The right column shows the expected true cost and its LOOCV estimates for uLSIF in the same manner.

The graphs show that overall CV gives reasonably good approximations of the expected cost, although CV for LSIF with small λ and small σ is rather inaccurate due to numerical problems—the solution path of LSIF is computed from $\lambda = \infty$ to $\lambda = 0$, and the numerical error is accumulated as the tracking process approaches to $\lambda = 0$. This phenomenon seems problematic when σ is small.

4.5 Difference between LSIF and uLSIF

In Section 3.3, we analyzed the approximation error of uLSIF against LSIF. Here we numerically investigate the behavior of the approximation error (26) as well as the norm bound (28) and the bridge bound (31). We set the number of training and test samples at $n_{\mathrm{tr}} = 200$ and $n_{\mathrm{te}} = 1000$, respectively. The number of template points in the Gaussian kernel model (22) is set at $b = 100$. The centers of the kernel functions are randomly chosen from the test points (see Section 4.2).

Figure 7 depicts the true approximation error as well as its upper bounds as functions of the regularization parameter λ; λ is varied from 0.001 to 10 and the three Gaussian widths $\sigma = 0.1, 0.5, 1.0$ are tested. The graphs show that when λ and σ are large, the approximation error tends to be small; this is in good agreement with the theoretical analysis given in Section 3.3. The bridge bound is fairly tight in the entire range and is sharper than the norm bound except when σ is small and λ is large.

4.6 Summary

Through the numerical examples, we overall found that LSIF and uLSIF give qualitatively similar results. However, the computation of the solution-path tracking algorithm for LSIF tends to be numerically unstable, which can result in unreliable model selection performance. On the other hand, only a system of linear equations needs to be solved in uLSIF, which turned out to be much more stable than LSIF. Thus, uLSIF would be practically more reliable than LSIF.

Based on the above findings, we will focus on uLSIF in the rest of this paper.

5. Relation to Existing Methods

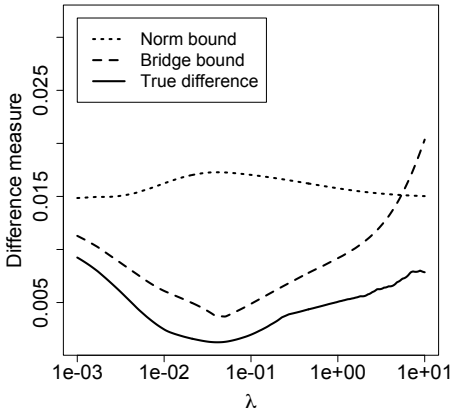
In this section, we discuss the characteristics of existing approaches in comparison with the proposed methods.

5.1 Kernel Density Estimator

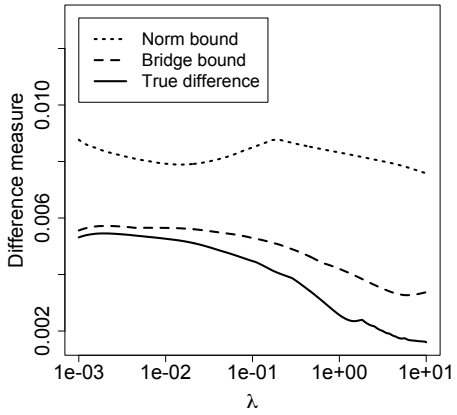
The *kernel density estimator* (KDE) is a non-parametric technique to estimate a probability density function $p(x)$ from its i.i.d. samples $\{x_k\}_{k=1}^{n}$. For the Gaussian kernel (21), KDE is expressed as

$$\widehat{p}(x) = \frac{1}{n_{\mathrm{tr}}(2\pi\sigma^2)^{d/2}} \sum_{k=1}^{n} K_\sigma(x, x_k).$$

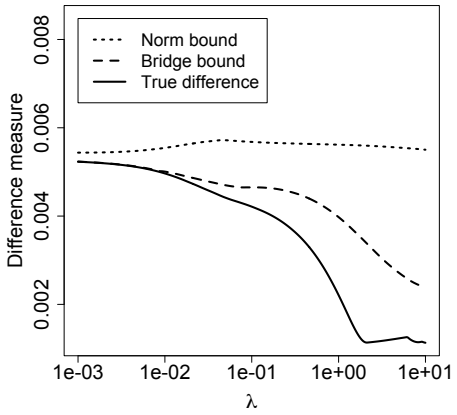
The performance of KDE depends on the choice of the kernel width σ. The kernel width σ can be optimized by *likelihood cross-validation* (LCV) as follows (Härdle et al., 2004): First, divide the samples $\{x_i\}_{i=1}^{n}$ into R disjoint subsets $\{\mathcal{X}_r\}_{r=1}^{R}$. Then obtain a density estimate $\widehat{p}_{\mathcal{X}_k}(x)$ from



(a) $\sigma = 0.1$.



(b) $\sigma = 0.5$.



(c) $\sigma = 1.0$.

Figure 7: The approximation error of uLSIF against LSIF as functions of the regularization parameter λ for different Gaussian width σ. Its upper bounds are also plotted in the graphs.

$\{\mathcal{X}_r\}_{r\neq k}$ (i.e., without $\mathcal{X}_k$) and compute its log-likelihood for $\mathcal{X}_k$:

$$\frac{1}{|\mathcal{X}_k|}\sum_{x\in\mathcal{X}_k}\log\widehat{p}_{\mathcal{X}_k}(x).$$

Repeat this procedure for $r = 1,2,\ldots,R$ and choose the value of σ such that the average of the above hold-out log-likelihood over all r is maximized. Note that the average hold-out log-likelihood is an almost unbiased estimate of the Kullback-Leibler divergence from $p(x)$ to $\widehat{p}(x)$, up to an irrelevant constant.

KDE can be used for importance estimation by first obtaining density estimators $\widehat{p}_{\mathrm{tr}}(x)$ and $\widehat{p}_{\mathrm{te}}(x)$ separately from $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$, and then estimating the importance by $\widehat{w}(x) = \widehat{p}_{\mathrm{te}}(x)/\widehat{p}_{\mathrm{tr}}(x)$. A potential limitation of this approach is that KDE suffers from the *curse of dimensionality* (Vapnik, 1998; Härdle et al., 2004), that is, the number of samples needed to maintain the

same approximation quality grows exponentially as the dimension of the domain increases. This is critical when the number of available samples is limited. Therefore, the KDE-based approach may not be reliable in high-dimensional problems.

5.2 Kernel Mean Matching

The *kernel mean matching* (KMM) method allows us to directly obtain an estimate of the importance values at training points without going through density estimation (Huang et al., 2007). The basic idea of KMM is to find $\widehat{w}(x)$ such that the mean discrepancy between nonlinearly transformed samples drawn from $p_{\mathrm{te}}(x)$ and $p_{\mathrm{tr}}(x)$ is minimized in a *universal reproducing kernel Hilbert space* (Steinwart, 2001). The Gaussian kernel (21) is an example of kernels that induce universal reproducing kernel Hilbert spaces and it has been shown that the solution of the following optimization problem agrees with the true importance:

$$\min_{w(x)} \left\| \int K_\sigma(x,\cdot) p_{\mathrm{te}}(x) dx - \int K_\sigma(x,\cdot) w(x) p_{\mathrm{tr}}(x) dx \right\|_{\mathcal{H}}^2$$
$$\text{subject to } \int w(x) p_{\mathrm{tr}}(x) dx = 1 \text{ and } w(x) \geq 0,$$

where $\|\cdot\|_{\mathcal{H}}$ denotes the norm in the Gaussian reproducing kernel Hilbert space and $K_\sigma(x,x')$ is the Gaussian kernel (21).

An empirical version of the above problem is reduced to the following quadratic program:

$$\min_{\{w_i\}_{i=1}^{n_{\mathrm{tr}}}} \left[\frac{1}{2} \sum_{i,i'=1}^{n_{\mathrm{tr}}} w_i w_{i'} K_\sigma(x_i^{\mathrm{tr}}, x_{i'}^{\mathrm{tr}}) - \sum_{i=1}^{n_{\mathrm{tr}}} w_i \kappa_i \right]$$
$$\text{subject to } \left| \sum_{i=1}^{n_{\mathrm{tr}}} w_i - n_{\mathrm{tr}} \right| \leq n_{\mathrm{tr}} \varepsilon \text{ and } 0 \leq w_1, w_2, \ldots, w_{n_{\mathrm{tr}}} \leq B,$$

where

$$\kappa_i = \frac{n_{\mathrm{tr}}}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} K_\sigma(x_i^{\mathrm{tr}}, x_j^{\mathrm{te}}).$$

B (≥ 0) and ε (≥ 0) are tuning parameters that control the regularization effects. The solution $\{\widehat{w}_i\}_{i=1}^{n_{\mathrm{tr}}}$ is an estimate of the importance at the training points $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$.

Since KMM does not involve density estimation, it is expected to work well even in high dimensional cases. However, the performance is dependent on the tuning parameters B, ε, and σ, and they cannot be simply optimized, for example, by CV since estimates of the importance are available only at the training points. A popular heuristic is to use the median distance between samples as the Gaussian width σ, which is shown to be useful (Schölkopf and Smola, 2002; Song et al., 2007). However, there seems no strong justification for this heuristic. For the choice of ε, a theoretical result given in Huang et al. (2007) could be used as guidance, although it is still hard to determine the best value of ε in practice.

5.3 Logistic Regression

Another approach to directly estimating the importance is to use a probabilistic classifier. Let us assign a selector variable $\eta = -1$ to training samples and $\eta = 1$ to test samples, that is, the training

and test densities are written as

$$p_{\mathrm{tr}}(x) = p(x|\eta = -1),$$
$$p_{\mathrm{te}}(x) = p(x|\eta = 1).$$

Note that η is regarded as a random variable.

Application of the Bayes theorem yields that the importance can be expressed in terms of η as follows (Qin, 1998; Cheng and Chu, 2004; Bickel et al., 2007):

$$w(x) = \frac{p(\eta = -1)}{p(\eta = 1)} \frac{p(\eta = 1|x)}{p(\eta = -1|x)}.$$

The probability ratio of test and training samples may be simply estimated by the ratio of the numbers of samples:

$$\frac{p(\eta = -1)}{p(\eta = 1)} \approx \frac{n_{\mathrm{tr}}}{n_{\mathrm{te}}}.$$

The conditional probability $p(\eta|x)$ could be approximated by discriminating test samples from training samples using a *logistic regression* (LogReg) classifier, where η plays the role of a class variable. Below we briefly explain the LogReg method.

The LogReg classifier employs a parametric model of the following form for expressing the conditional probability $p(\eta|x)$:

$$\widehat{p}(\eta|x) = \frac{1}{1 + \exp\left(-\eta \sum_{\ell=1}^{m} \zeta_\ell \phi_\ell(x)\right)},$$

where m is the number of basis functions and $\{\phi_\ell(x)\}_{\ell=1}^{m}$ are fixed basis functions. The parameter ζ is learned so that the negative regularized log-likelihood is minimized:

$$\widehat{\zeta} = \operatorname*{argmin}_{\zeta} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \log\left(1 + \exp\left(\sum_{\ell=1}^{m} \zeta_\ell \phi_\ell(x_i^{\mathrm{tr}})\right)\right) + \sum_{j=1}^{n_{\mathrm{te}}} \log\left(1 + \exp\left(-\sum_{\ell=1}^{m} \zeta_\ell \phi_\ell(x^{\mathrm{te}})\right)\right) + \lambda \zeta^\top \zeta \right].$$

Since the above objective function is convex, the global optimal solution can be obtained by standard nonlinear optimization methods such as Newton's method, the conjugate gradient method, and the BFGS method (Minka, 2007). Then the importance estimate is given by

$$\widehat{w}(x) = \frac{n_{\mathrm{tr}}}{n_{\mathrm{te}}} \exp\left(\sum_{\ell=1}^{m} \zeta_\ell \phi_\ell(x)\right). \tag{34}$$

An advantage of the LogReg method is that model selection (that is, the choice of the basis functions $\{\phi_\ell(x)\}_{\ell=1}^{m}$ as well as the regularization parameter λ) is possible by standard CV since the learning problem involved above is a standard supervised classification problem.

5.4 Kullback-Leibler Importance Estimation Procedure

The *Kullback-Leibler importance estimation procedure* (KLIEP) (Sugiyama et al., 2008a) also directly gives an estimate of the importance function without going through density estimation by matching the two distributions in terms of the Kullback-Leibler divergence (Kullback and Leibler, 1951).

Let us model the importance $w(x)$ by the linear model (1). An estimate of the test density $p_{\mathrm{te}}(x)$ is given by using the model $\widehat{w}(x)$ as

$$\widehat{p}_{\mathrm{te}}(x) = \widehat{w}(x)p_{\mathrm{tr}}(x).$$

In KLIEP, the parameters α are determined so that the Kullback-Leibler divergence from $p_{\mathrm{te}}(x)$ to $\widehat{p}_{\mathrm{te}}(x)$ is minimized:

$$\begin{aligned}\mathrm{KL}[p_{\mathrm{te}}(x)\|\widehat{p}_{\mathrm{te}}(x)] &= \int_{\mathcal{D}} p_{\mathrm{te}}(x)\log\frac{p_{\mathrm{te}}(x)}{\widehat{w}(x)p_{\mathrm{tr}}(x)}dx \\ &= \int_{\mathcal{D}} p_{\mathrm{te}}(x)\log\frac{p_{\mathrm{te}}(x)}{p_{\mathrm{tr}}(x)}dx - \int_{\mathcal{D}} p_{\mathrm{te}}(x)\log\widehat{w}(x)dx. \qquad (35)\end{aligned}$$

The first term is a constant, so it can be safely ignored. Since $\widehat{p}_{\mathrm{te}}(x)$ $(=\widehat{w}(x)p_{\mathrm{tr}}(x))$ is a probability density function, it should satisfy

$$1 = \int_{\mathcal{D}} \widehat{p}_{\mathrm{te}}(x)dx = \int_{\mathcal{D}} \widehat{w}(x)p_{\mathrm{tr}}(x)dx. \qquad (36)$$

Then the KLIEP optimization problem is given by replacing the expectations in Eqs. (35) and (36) with empirical averages as follows:

$$\begin{aligned}&\max_{\{\alpha_\ell\}_{\ell=1}^b}\left[\sum_{j=1}^{n_{\mathrm{te}}}\log\left(\sum_{\ell=1}^{b}\alpha_\ell\varphi_\ell(x_j^{\mathrm{te}})\right)\right] \\ &\qquad\text{subject to } \sum_{\ell=1}^{b}\alpha_\ell\left(\sum_{i=1}^{n_{\mathrm{tr}}}\varphi_\ell(x_i^{\mathrm{tr}})\right) = n_{\mathrm{tr}} \text{ and } \alpha_1,\alpha_2,\ldots,\alpha_b \geq 0.\end{aligned}$$

This is a convex optimization problem and the global solution—which tends to be sparse (Boyd and Vandenberghe, 2004)—can be obtained, for example, by simply performing gradient ascent and feasibility satisfaction iteratively. Model selection of KLIEP is possible by LCV.

Properties of KLIEP-type algorithms have been theoretically investigated in Nguyen et al. (2008) and Sugiyama et al. (2008b) (see also Qin, 1998; Cheng and Chu, 2004). Note that the importance model of KLIEP is the linear model (1), while that of LogReg is the log-linear model (34). A variant of KLIEP for log-linear models has been studied in Tsuboi et al. (2008).

5.5 Discussions

Table 1 summarizes properties of proposed and existing methods.

KDE is efficient in computation since no optimization is involved, and model selection is possible by LCV. However, KDE may suffer from the curse of dimensionality due to the difficulty of density estimation in high dimensions.

Methods	Density estimation	Model selection	Optimization	Out-of-sample prediction
KDE	Necessary	**Available**	**Analytic**	**Possible**
KMM	**Not necessary**	Not available	Convex quadratic program	Not possible
LogReg	**Not necessary**	**Available**	Convex non-linear	**Possible**
KLIEP	**Not necessary**	**Available**	Convex non-linear	**Possible**
LSIF	**Not necessary**	**Available**	Convex quadratic program	**Possible**
uLSIF	**Not necessary**	**Available**	**Analytic**	**Possible**

Table 1: Relation between proposed and existing methods.

KMM can potentially overcome the curse of dimensionality by directly estimating the importance. However, there is no objective model selection method. Therefore, model parameters such as the Gaussian width need to be determined by hand, which is highly unreliable unless we have strong prior knowledge. Furthermore, the computation of KMM is rather demanding since a quadratic programming problem has to be solved.

LogReg and KLIEP also do not involve density estimation, but different from KMM, they give an estimate the entire importance function, not only the values of the importance at training points. Therefore, the values of the importance at unseen points can be estimated by LogReg and KLIEP. This feature is highly useful since it enables us to employ CV for model selection, which is a significant advantage over KMM. However, LogReg and KLIEP are computationally rather expensive since non-linear optimization problems have to be solved. Note that the LogReg method is slightly different in motivation from other methods, but has some similarity in computation and implementation, for example, the LogReg method also involves a kernel smoother.

The proposed LSIF method is qualitatively similar to LogReg and KLIEP, that is, it can avoid density estimation, model selection is possible, and non-linear optimization is involved. LSIF is advantageous over LogReg and KLIEP in that it is equipped with a regularization path tracking algorithm. Thanks to this, model selection of LSIF is computationally much more efficient than LogReg and KLIEP. However, the regularization path tracking algorithm tends to be numerically unstable.

The proposed uLSIF method inherits good properties of existing methods such as no density estimation involved and a build-in model selection method equipped. In addition to these preferable properties, the solution of uLSIF can be computed in an efficient and numerically stable manner. Furthermore, thanks to the availability of the closed-form solution of uLSIF, the LOOCV score can be analytically computed without repeating hold-out loops, which highly contributes to reducing the computation time in the model selection phase.

In the next section, we experimentally show that uLSIF is computationally more efficient than existing direct importance estimation methods, while its estimation accuracy is comparable to the best existing methods.

6. Experiments

In this section, we compare the experimental performance of the proposed and existing methods.

6.1 Importance Estimation

Let the dimension of the domain be d and

$$p_{\mathrm{tr}}(x) = \mathcal{N}(x; (0,0,\ldots,0)^\top, I_d),$$
$$p_{\mathrm{te}}(x) = \mathcal{N}(x; (1,0,\ldots,0)^\top, I_d).$$

The task is to estimate the importance at training points:

$$w_i = w(x_i^{\mathrm{tr}}) = \frac{p_{\mathrm{te}}(x_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(x_i^{\mathrm{tr}})} \quad \text{for } i = 1,2,\ldots,n_{\mathrm{tr}}.$$

We compare the following methods:

KDE(CV): The Gaussian kernel (21) is used, where the kernel widths of the training and test densities are separately optimized based on 5-fold LCV.

KMM(med): The performance of KMM is dependent on B, ε, and σ. We set $B = 1000$ and $\varepsilon = (\sqrt{n_{\mathrm{tr}}} - 1)/\sqrt{n_{\mathrm{tr}}}$ following the original paper (Huang et al., 2007), and the Gaussian width σ is set at the median distance between samples within the training set and the test set (Schölkopf and Smola, 2002; Song et al., 2007).

LogReg(CV): The Gaussian kernel model (22) are used as basis functions. The kernel width σ and the regularization parameter λ are chosen based on 5-fold CV.[1]

KLIEP(CV): The Gaussian kernel model (22) is used. The kernel width σ is selected based on 5-fold LCV.

uLSIF(CV): The Gaussian kernel model (22) is used. The kernel width σ and the regularization parameter λ are determined based on LOOCV.

All the methods are implemented using the *MATLAB*® environment, where the *CPLEX*® optimizer is used for solving quadratic programs in KMM and the *LIBLINEAR* implementation is used for LogReg (Lin et al., 2007).

We fixed the number of test points at $n_{\mathrm{te}} = 1000$ and consider the following two setups for the number n_{tr} of training samples and the input dimensionality d:

(a) n_{tr} is fixed at $n_{\mathrm{tr}} = 100$ and d is changed as $d = 1,2,\ldots,20$,

(b) d is fixed at $d = 10$ and n_{tr} is changed as $n_{\mathrm{tr}} = 50,60,\ldots,150$.

We run the experiments 100 times for each d, each n_{tr}, and each method, and evaluate the quality of the importance estimates $\{\widehat{w}_i\}_{i=1}^{n_{\mathrm{tr}}}$ by the *normalized mean squared error* (NMSE):

$$\mathrm{NMSE} = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\frac{\widehat{w}_i}{\sum_{i'=1}^{n_{\mathrm{tr}}} \widehat{w}_{i'}} - \frac{w_i}{\sum_{i'=1}^{n_{\mathrm{tr}}} w_{i'}} \right)^2.$$

1. In Sugiyama et al. (2008b) where KLIEP has been proposed, the performance of LogReg has been experimentally investigated in the same setup. In that paper, however, LogReg was not regularized since KLIEP was not also regularized. On the other hand, we use a regularized LogReg method and choose the regularization parameter in addition to the Gaussian kernel width by CV here. Thanks to the regularization effect, the results of LogReg in the current paper tends to be better than that reported in Sugiyama et al. (2008b).

In practice, the scale of the importance is not significant and the relative magnitude among w_i is important. Thus the above NMSE would be a suitable error metric for evaluating the performance of each method.

NMSEs averaged over 100 trials (a) as a function of input dimensionality d and (b) as a function of the training sample size n_{tr} are plotted in log scale in Figure 8. Error bars are omitted for clear visibility—instead, the best method in terms of the mean error and comparable ones based on the t-test at the significance level 1% are indicated by '$\circ$'; the methods with significant difference from the best methods are indicated by '$\times$'.

Figure 8(a) shows that the error of KDE(CV) sharply increases as the input dimensionality grows, while LogReg, KLIEP, and uLSIF tend to give much smaller errors than KDE. This would be the fruit of directly estimating the importance without going through density estimation. KMM tends to perform poorly, which is caused by an inappropriate choice of the Gaussian kernel width. On the other hand, model selection in LogReg, KLIEP, and uLSIF seems to work quite well. Figure 8(b) shows that the errors of all methods tend to decrease as the number of training samples grows. Again LogReg, KLIEP, and uLSIF tend to give much smaller errors than KDE and KMM.

Next we investigate the computation time. Each method has a different model selection strategy, that is, KMM does not involve CV, KDE and KLIEP involve CV over the kernel width, and LogReg and uLSIF involve CV over both the kernel width and the regularization parameter. Thus the naive comparison of the total computation time is not so meaningful. For this reason, we first investigate the computation time of each importance estimation method after the model parameters are fixed.

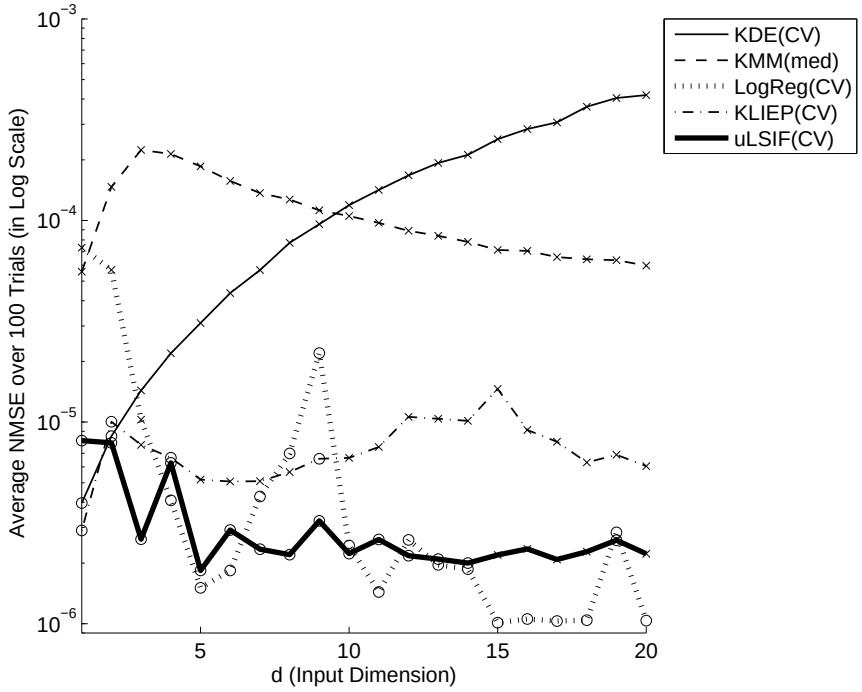
The average CPU computation time over 100 trials are summarized in Figure 9. Figure 9(a) shows that the computation time of KDE, KLIEP, and uLSIF is almost independent of the input dimensionality, while that of KMM and LogReg is rather dependent on the input dimensionality. Note that LogReg for $d \leq 3$ is slow due to some convergence problem of the LIBLINEAR package. Among them, the proposed uLSIF is one of the fastest methods. Figure 9(b) shows that the computation time of LogReg, KLIEP, and uLSIF is nearly independent of the number of training samples, while that of KDE and KMM sharply increase as the number of training samples increases.

Both LogReg and uLSIF have high accuracy and their computation time after model selection is comparable. Finally, we compare the entire computation time of LogReg and uLSIF including CV, which is summarized in Figure 10. We note that the Gaussian width σ and the regularization parameter λ are chosen over the 9×9 grid in this experiment for both LogReg and uLSIF. Therefore, the comparison of the entire computation time is fair. Figures 10(a) and 10(b) show that uLSIF is approximately 5 times faster than LogReg.

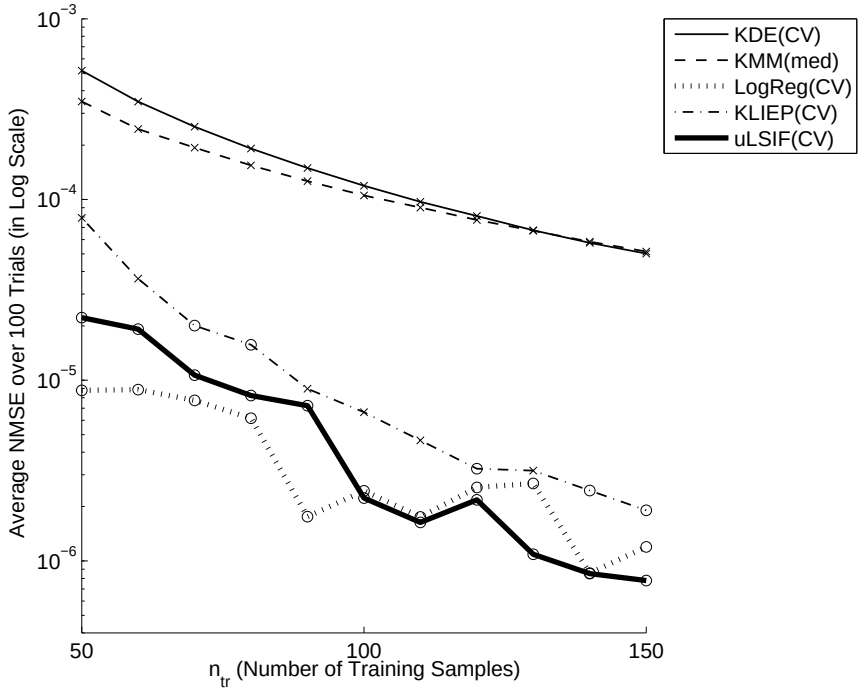
Overall, uLSIF is shown to be comparable to the best existing method (LogReg) in terms of the accuracy, but is computationally more efficient than LogReg.

6.2 Covariate Shift Adaptation in Regression and Classification

Next, we illustrate how the importance estimation methods could be used in *covariate shift adaptation* (Shimodaira, 2000; Zadrozny, 2004; Sugiyama and Müller, 2005; Huang et al., 2007; Bickel and Scheffer, 2007; Bickel et al., 2007; Sugiyama et al., 2007). Covariate shift is a situation in supervised learning where the input distributions change between the training and test phase but the conditional distribution of outputs given inputs remains unchanged. Under covariate shift, standard learning techniques such as maximum likelihood estimation or cross-validation are biased—the bias

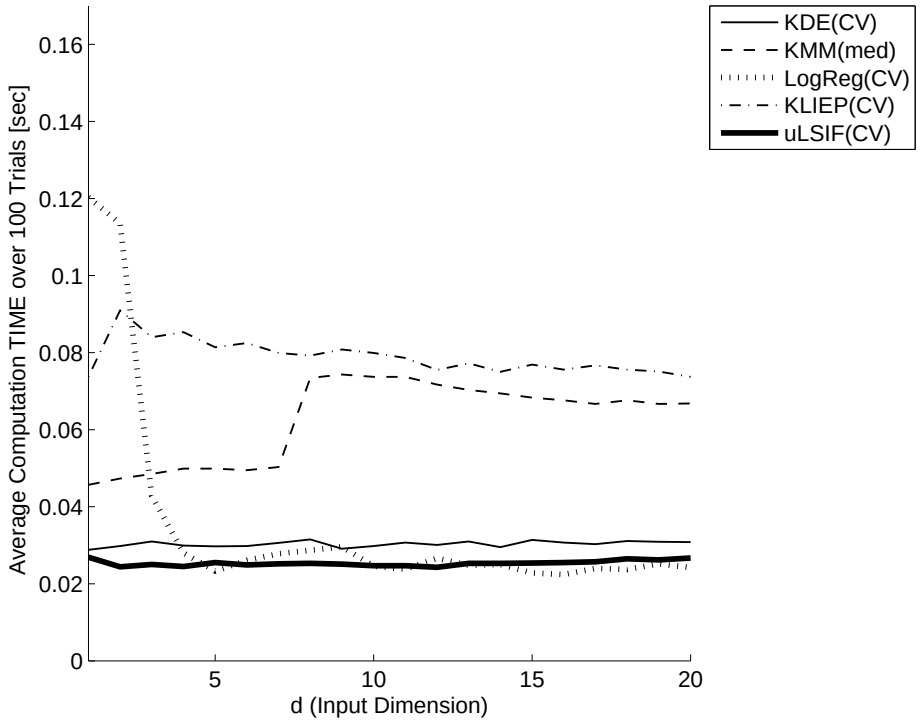


(a) When input dimensionality is changed

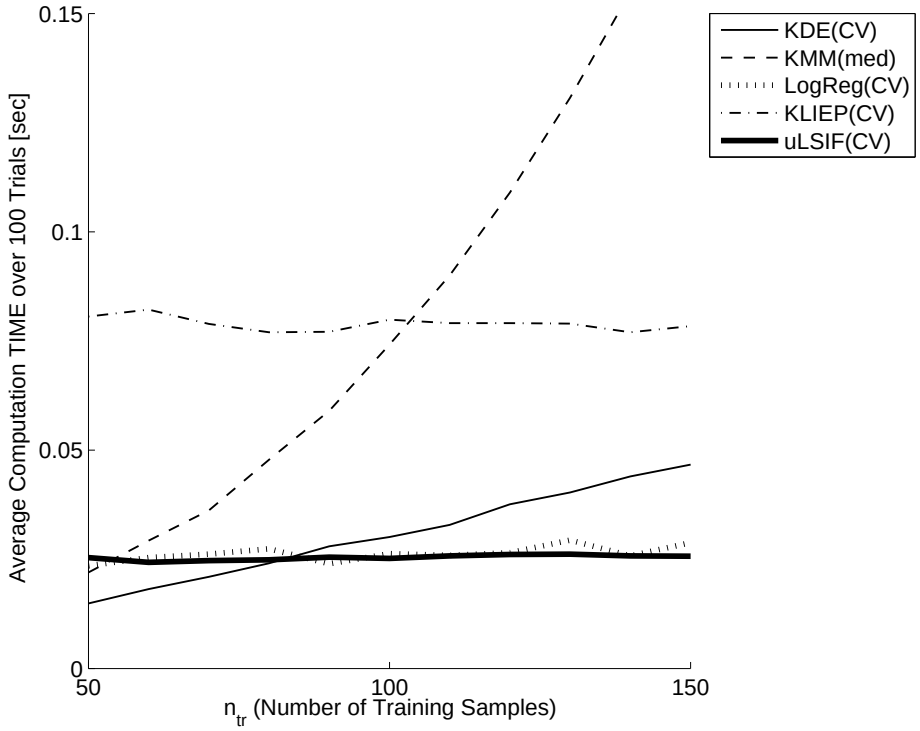


(b) When training sample size is changed

Figure 8: NMSEs averaged over 100 trials in log scale for the artificial data set. Error bars are omitted for clear visibility. Instead, the best method in terms of the mean error and comparable ones based on the *t-test* at the significance level 1% are indicated by '$\circ$'; the methods with significant difference from the best methods are indicated by '$\times$'.

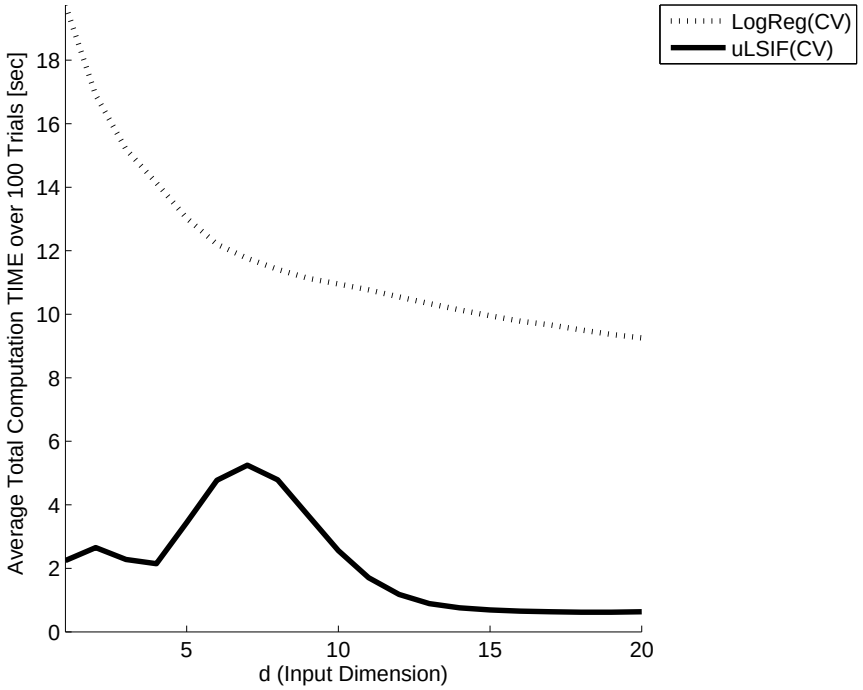


(a) When input dimensionality is changed

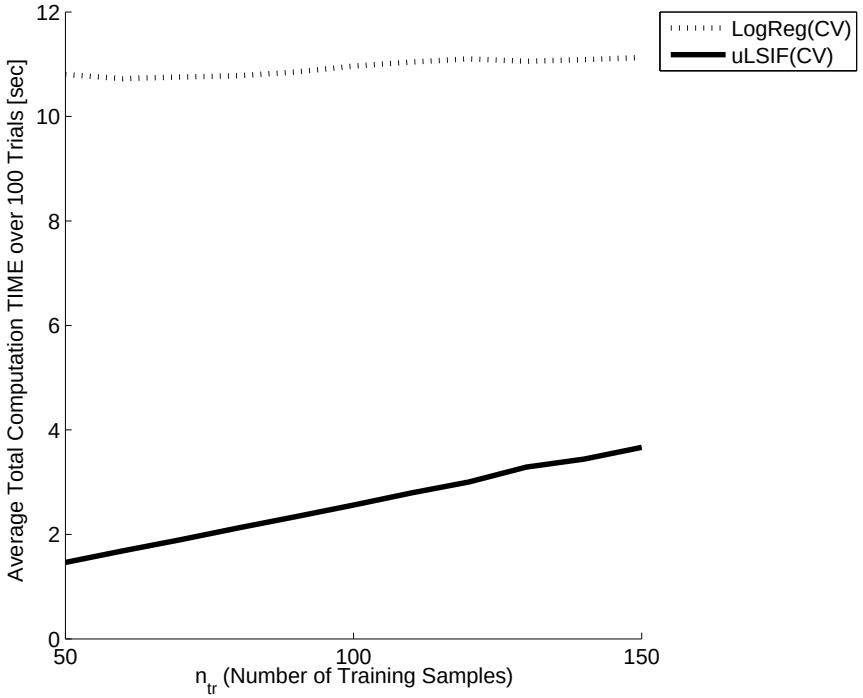


(b) When training sample size is changed

Figure 9: Average computation time (after model selection) over 100 trials for the artificial data set.



(a) When input dimensionality is changed



(b) When training sample size is changed

Figure 10: Average computation time over 100 trials for the artificial data set (including model selection of the Gaussian width σ and the regularization parameter λ over the 9×9 grid).

caused by covariate shift can be asymptotically canceled by weighting the loss function according to the importance.

In addition to training input samples $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ drawn from a training input density $p_{\mathrm{tr}}(x)$ and test input samples $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ drawn from a test input density $p_{\mathrm{te}}(x)$, suppose that we are given training *output* samples $\{y_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ at the training input points $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$. The task is to predict the outputs for test inputs $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ based on the input-output training samples $\{(x_i^{\mathrm{tr}},y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$.

We use the following kernel model for function learning:

$$\widehat{f}(x;\theta)=\sum_{\ell=1}^{t}\theta_\ell K_h(x,m_\ell),$$

where $K_h(x,x')$ is the Gaussian kernel (21) and m_ℓ is a template point randomly chosen from $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ without replacement. We set the number of kernels at $t=50$. We learn the parameter θ by *importance weighted regularized least-squares* (IWRLS) (Evgeniou et al., 2000; Sugiyama and Müller, 2005):

$$\widehat{\theta}_{\mathrm{IWRLS}}\equiv\operatorname*{argmin}_{\theta}\left[\sum_{i=1}^{n_{\mathrm{tr}}}\widehat{w}(x_i^{\mathrm{tr}})\left(\widehat{f}(x_i^{\mathrm{tr}};\theta)-y_i^{\mathrm{tr}}\right)^2+\gamma\|\theta\|^2\right]. \tag{37}$$

It is known that IWRLS is consistent when the true importance $w(x_i^{\mathrm{tr}})$ is used as weights—unweighted RLS is not consistent due to covariate shift, given that the true learning target function $f(x)$ is not realizable by the model $\widehat{f}(x)$ (Shimodaira, 2000).

The solution $\widehat{\theta}_{\mathrm{IWRLS}}$ is analytically given by

$$\widehat{\theta}_{\mathrm{IWRLS}}=(K^\top\widehat{W}K+\gamma I_b)^{-1}K^\top\widehat{W}y^{\mathrm{tr}},$$

where

$$\begin{aligned}K_{i,\ell}&=K_h(x_i^{\mathrm{tr}},m_\ell),\\ \widehat{W}&=\mathrm{diag}\left(\widehat{w}(x_1^{\mathrm{tr}}),\widehat{w}(x_2^{\mathrm{tr}}),\ldots,\widehat{w}(x_{n_{\mathrm{tr}}}^{\mathrm{tr}})\right),\\ y^{\mathrm{tr}}&=(y_1^{\mathrm{tr}},y_2^{\mathrm{tr}},\ldots,y_{n_{\mathrm{tr}}}^{\mathrm{tr}})^\top.\end{aligned}$$

$\mathrm{diag}(a,b,\ldots,c)$ denotes the diagonal matrix with the diagonal elements $a,b,\ldots,c$.

The kernel width h and the regularization parameter γ in IWRLS (37) are chosen by *importance weighted CV* (IWCV) (Sugiyama et al., 2007). More specifically, we first divide the training samples $\{z_i^{\mathrm{tr}}\mid z_i^{\mathrm{tr}}=(x_i^{\mathrm{tr}},y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$ into R disjoint subsets $\{\mathcal{Z}_r^{\mathrm{tr}}\}_{r=1}^{R}$. Then a function $\widehat{f}_r(x)$ is learned using $\{\mathcal{Z}_j^{\mathrm{tr}}\}_{j\neq r}$ by IWRLS and its mean test error for the remaining samples $\mathcal{Z}_r^{\mathrm{tr}}$ is computed:

$$\frac{1}{|\mathcal{Z}_r^{\mathrm{tr}}|}\sum_{(x,y)\in\mathcal{Z}_r^{\mathrm{tr}}}\widehat{w}(x)\mathrm{loss}\left(\widehat{f}_r(x),y\right),$$

where

$$\mathrm{loss}(\widehat{y},y)=\begin{cases}(\widehat{y}-y)^2 & \text{(Regression)},\\ \frac{1}{2}(1-\mathrm{sign}\{\widehat{y}y\}) & \text{(Classification)}.\end{cases}$$

We repeat this procedure for $r=1,2,\ldots,R$ and choose the kernel width h and the regularization parameter γ so that the average of the above mean test error over all r is minimized. We set the number of folds in IWCV at $R=5$. IWCV is shown to be an (almost) unbiased estimator of the

Data	Uniform	KDE (CV)	KMM (med)	LogReg (CV)	KLIEP (CV)	uLSIF (CV)
kin-8fh	1.00(0.34)	1.22(0.52)	1.55(0.39)	1.31(0.39)	**0.95**(**0.31**)	**1.02**(**0.33**)
kin-8fm	1.00(0.39)	1.12(0.57)	1.84(0.58)	1.38(0.57)	**0.86**(**0.35**)	**0.88**(**0.39**)
kin-8nh	**1.00**(**0.26**)	1.09(0.20)	1.19(0.29)	1.09(0.19)	**0.99**(**0.22**)	**1.02**(**0.18**)
kin-8nm	**1.00**(**0.30**)	1.14(0.26)	1.20(0.20)	1.12(0.21)	**0.97**(**0.25**)	1.04(0.25)
abalone	**1.00**(**0.50**)	1.02(0.41)	**0.91**(**0.38**)	**0.97**(**0.49**)	**0.94**(**0.67**)	**0.96**(**0.61**)
image	**1.00**(**0.51**)	0.98(0.45)	1.08(0.54)	**0.98**(**0.46**)	**0.94**(**0.44**)	**0.98**(**0.47**)
ringnorm	1.00(0.04)	0.87(0.04)	**0.87**(**0.04**)	0.95(0.08)	0.99(0.06)	0.91(0.08)
twonorm	1.00(0.58)	1.16(0.71)	**0.94**(**0.57**)	**0.91**(**0.61**)	**0.91**(**0.52**)	**0.88**(**0.57**)
waveform	1.00(0.45)	1.05(0.47)	0.98(0.31)	**0.93**(**0.32**)	**0.93**(**0.34**)	**0.92**(**0.32**)
Average	1.00(0.38)	1.07(0.40)	1.17(0.37)	1.07(0.37)	0.94(0.35)	0.96(0.36)
Comp. time	—	0.82	3.50	3.27	2.23	1.00

Table 2: Mean test error averaged over 100 trials for covariate shift adaptation in regression and classification. The numbers in the brackets are the standard deviation. All the error values are normalized by that of 'Uniform' (uniform weighting, or equivalently no importance weighting). For each data set, the best method in terms of the mean error and comparable ones based on the *Wilcoxon signed rank test* at the significance level 1% are described in bold face. The upper half corresponds to regression data sets taken from DELVE (Rasmussen et al., 1996), while the lower half correspond to classification data sets taken from IDA (Rätsch et al., 2001). All the methods are implemented using the *MATLAB*® environment, where the *CPLEX*® optimizer is used for solving quadratic programs in KMM and the *LIBLINEAR* implementation is used for LogReg (Lin et al., 2007).

generalization error, while unweighted CV with misspecified models is biased due to covariate shift (Zadrozny, 2004; Sugiyama et al., 2007).

The data sets provided by DELVE (Rasmussen et al., 1996) and IDA (Rätsch et al., 2001) are used for performance evaluation. Each data set consists of input/output samples $\{(x_k, y_k)\}_{k=1}^{n}$. We normalize all the input samples $\{x_k\}_{k=1}^{n}$ into $[0,1]^d$ and choose the test samples $\{(x_j^{\mathrm{te}}, y_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}}$ from the pool $\{(x_k, y_k)\}_{k=1}^{n}$ as follows. We randomly choose one sample (x_k, y_k) from the pool and accept this with probability $\min(1, 4(x_k^{(c)})^2)$, where $x_k^{(c)}$ is the c-th element of x_k and c is randomly determined and fixed in each trial of the experiments. Then we remove x_k from the pool regardless of its rejection or acceptance, and repeat this procedure until n_{te} samples are accepted. We choose the training samples $\{(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$ uniformly from the rest. Thus, in this experiment, the test input density tends to be lower than the training input density when $x_k^{(c)}$ is small. We set the number of samples at $n_{\mathrm{tr}} = 100$ and $n_{\mathrm{te}} = 500$ for all data sets. Note that we only use $\{(x_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}}$ and $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ for training regressors or classifiers; the test output values $\{y_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ are used only for evaluating the generalization performance.

We run the experiments 100 times for each data set and evaluate the *mean test error*:

$$\frac{1}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} \mathrm{loss}\left(\widehat{f}(x_j^{\mathrm{te}}), y_j^{\mathrm{te}}\right).$$

The results are summarized in Table 2, where 'Uniform' denotes uniform weights (or equivalently, no importance weight). The numbers in the brackets are the standard deviation. All the error values are normalized so that the mean error of Uniform is one. For each data set, the best method in terms of the mean error and comparable ones based on the *Wilcoxon signed rank test* at the significance level 1% are described in bold face. The upper half of the table corresponds to regression data sets taken from DELVE (Rasmussen et al., 1996), while the lower half correspond to classification data sets taken from IDA (Rätsch et al., 2001). All the methods are implemented using the *MATLAB*® environment, where the *CPLEX*® optimizer is used for solving quadratic programs in KMM and the *LIBLINEAR* implementation is used for LogReg (Lin et al., 2007).

The table shows that the generalization performance of uLSIF tends to be better than that of Uniform, KDE, KMM, and LogReg, while it is comparable to the best existing method (KLIEP). The mean computation time over 100 trials is described in the bottom row of the table, where the value is normalized so that the computation time of uLSIF is one. This shows that the computation time of uLSIF is much shorter than KLIEP. Thus, uLSIF is overall shown to be useful in covariate shift adaptation.

6.3 Outlier Detection

Finally, we apply importance estimation methods in outlier detection.

Here, we consider an outlier detection problem of finding irregular samples in a data set ("evaluation data set") based on another data set ("model data set") that only contains regular samples. Defining the importance over two sets of samples, we can see that the importance values for regular samples are close to one, while those for outliers tend to be significantly deviated from one. Thus the importance values could be used as an index of the degree of outlyingness in this scenario. Since the evaluation data set has wider support than the model data set, we regard the evaluation data set as the training set $\{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}}$ (that is, the denominator in the importance) and the model data set as the test set $\{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}$ (that is, the numerator in the importance). Then outliers tend to have smaller importance values (that is, close to zero).

We again test KMM(med), LogReg(CV), KLIEP(CV), and uLSIF(CV) for importance estimation; in addition, we include native outlier detection methods for comparison purposes. The outlier detection problem that the native methods used below solve is to find outliers in a single data set $\{x_k\}_{k=1}^{n}$—the native methods can be employed in the current scenario just by finding outliers from all samples:

$$\{x_k\}_{k=1}^{n} = \{x_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}} \cup \{x_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}}.$$

One-class support vector machine (OSVM): The *support vector machine* (SVM) (Vapnik, 1998; Schölkopf and Smola, 2002) is one of the most successful classification algorithms in machine learning. The core idea of SVM is to separate samples in different classes by the maximum margin hyperplane in a kernel-induced feature space.

OSVM is an extension of SVM to outlier detection (Schölkopf et al., 2001). The basic idea of OSVM is to separate data samples $\{x_k\}_{k=1}^{n}$ into outliers and inliers by a hyperplane in a Gaussian reproducing kernel Hilbert space. More specifically, the solution of OSVM is given

as the solution of the following convex quadratic programming problem:

$$\min_{\{w_k\}_{k=1}^n} \frac{1}{2}\sum_{k,k'=1}^{n} w_k w_{k'} K_\sigma(x_k, x_{k'})$$
$$\text{subject to } \sum_{k=1}^{n} w_k = 1 \text{ and } 0 \le w_1, w_2, \ldots, w_n \le \frac{1}{\nu n},$$

where ν ($0 \le \nu \le 1$) is the maximum fraction of outliers.

We use the inverse distance of a sample from the separating hyperplane as an outlier score. The OSVM solution is dependent on the outlier ratio ν and the Gaussian kernel width σ, and there seems to be no systematic method to determine the values of these tuning parameters. Here we use the median distance between samples as the Gaussian width, which is a popular heuristic (Schölkopf and Smola, 2002; Song et al., 2007). The value of ν is fixed at the true output ratio, that is, the ideal optimal value. Thus the simulation results below should be slightly in favor of OSVM.

Local outlier factor (LOF): LOF is the score to detect a local outlier which lies relatively far from the nearest dense region (Breunig et al., 2000). For a prefixed natural number k, the LOF value of a sample x is defined by

$$\mathrm{LOF}_R(x) = \frac{1}{k}\sum_{i=1}^{k} \frac{\mathrm{imd}_k(\mathrm{nearest}_i(x))}{\mathrm{imd}_k(x)},$$

where $\mathrm{nearest}_i(x)$ denotes the i-th nearest neighbor of x and $\mathrm{imd}_k(x)$ denotes the inverse mean distance from x to its k nearest neighbors:

$$\mathrm{imd}_k(x) = \frac{1}{\frac{1}{k}\sum_{i=1}^{k} \|x - \mathrm{nearest}_i(x)\|}.$$

If x alone is apart from a cloud of points, $\mathrm{imd}_k(x)$ tends to become smaller than than $\mathrm{imd}_k(\mathrm{nearest}_i(x))$ for all i. Then the LOF value gets large and therefore such a point is regarded as an outlier. The performance of LOF depends on the choice of the parameter k and there seems no systematic way to find an appropriate value of k. Here we test several different values of k.

Kernel density estimator (KDE'): A naive density estimation of all data samples $\{x_k\}_{k=1}^n$ can also be used for outlier detection since the density value itself could be regarded as an outlier score. We use KDE with the Gaussian kernel (21) for density estimation, where the kernel width is determined based on 5-fold LCV.

All the methods are implemented using the R environment—we use the *ksvm* routine in the *kernlab* package for OSVM (Karatzoglou et al., 2004) and the *lofactor* routine in the *dprep* package for LOF (Fernandez, 2005).

The data sets provided by IDA (Rätsch et al., 2001) are used for performance evaluation. These data sets are binary classification data sets consisting of positive/negative and training/test samples. We allocate all positive training samples for the "model" set, while all positive test samples and a fraction ρ $(= 0.01, 0.02, 0.05)$ of negative test samples are assigned in the "evaluation" set. Thus, we regard the positive samples as regular and the negative samples as irregular.

In the evaluation of the performance of outlier detection methods, it is important to take into account both the detection rate (the amount of true outliers an outlier detection algorithm can find) and the detection accuracy (the amount of true inliers that an outlier detection algorithm misjudges as outliers). Since there is a trade-off between the detection rate and the detection accuracy, we adopt the area under the ROC curve (AUC) as our error metric (Bradley, 1997).

The mean AUC values over 20 trials as well as the computation time are summarized in Table 3, showing that uLSIF works fairly well. KLIEP works slightly better than uLSIF, but uLSIF is computationally much more efficient. LogReg overall works reasonably well, but it performs poorly for some data sets and the average AUC performance is not as good as uLSIF or KLIEP. KMM and OSVM are not comparable to uLSIF in AUC and they are computationally inefficient. Note that we also tested KMM and OSVM with several different Gaussian widths and experimentally found that the heuristic of using the median sample distance as the Gaussian kernel width works reasonably well in this experiment. Thus the AUC values of KMM and OSVM are close to optimal. LOF with large k is shown to work well, although it is not clear whether the heuristic of simply using large k is always appropriate or not. The computational cost of LOF is high since nearest neighbor search is computationally expensive. KDE' works reasonably well, but its performance is not as good as uLSIF and KLIEP.

Overall, uLSIF is shown to work well with low computational costs.

7. Conclusions

The importance is useful in various machine learning scenarios such as covariate shift adaptation and outlier detection. In this paper, we proposed a new method of importance estimation that can avoid solving a substantially more difficult task of density estimation. We formulated the importance estimation problem as least-squares function fitting and casted the optimization problem as a convex quadratic program (we referred to it as LSIF). We theoretically elucidated the convergence property of LSIF and showed that the entire regularization path of LSIF can be efficiently computed based on a parametric optimization technique. We further developed an approximation algorithm (we called it uLSIF), which allows us to obtain the closed-form solution. We showed that the leave-one-out cross-validation score can be computed analytically for uLSIF—this makes the computation of uLSIF highly efficient. We carried out extensive simulations in covariate shift adaptation and outlier detection, and experimentally confirmed that the proposed uLSIF is computationally more efficient than existing approaches, while the accuracy of uLSIF is comparable to the best existing methods. Thanks to the low computational cost, uLSIF would be highly scalability to large data sets, which is very important in practical applications.

We have given convergence proofs for LSIF and uLSIF. A possible future direction to pursue along this line is to show the convergence of LSIF and uLSIF in non-parametric cases, for example, following Nguyen et al. (2008) and Sugiyama et al. (2008b). We are currently exploring various possible applications of important estimation methods beyond covariate shift adaptation or outlier detection, for example, feature selection, conditional distribution estimation, independent component analysis, and dimensionality reduction—we believe that importance estimation could be used as a new versatile tool in statistical machine learning.

Data		uLSIF	KLIEP	LogReg	KMM	OSVM	LOF			KDE'
Name	ρ	(CV)	(CV)	(CV)	(med)	(med)	$k=5$	$k=30$	$k=50$	(CV)
banana	0.01	0.851	0.815	0.447	0.578	0.360	0.838	0.915	0.919	0.934
	0.02	0.858	0.824	0.428	0.644	0.412	0.813	0.918	0.920	0.927
	0.05	0.869	0.851	0.435	0.761	0.467	0.786	0.907	0.909	0.923
b-cancer	0.01	0.463	0.480	0.627	0.576	0.508	0.546	0.488	0.463	0.400
	0.02	0.463	0.480	0.627	0.576	0.506	0.521	0.445	0.428	0.400
	0.05	0.463	0.480	0.627	0.576	0.498	0.549	0.480	0.452	0.400
diabetes	0.01	0.558	0.615	0.599	0.574	0.563	0.513	0.403	0.390	0.425
	0.02	0.558	0.615	0.599	0.574	0.563	0.526	0.453	0.434	0.425
	0.05	0.532	0.590	0.636	0.547	0.545	0.536	0.461	0.447	0.435
f-solar	0.01	0.416	0.485	0.438	0.494	0.522	0.480	0.441	0.385	0.378
	0.02	0.426	0.456	0.432	0.480	0.550	0.442	0.406	0.343	0.374
	0.05	0.442	0.479	0.432	0.532	0.576	0.455	0.417	0.370	0.346
german	0.01	0.574	0.572	0.556	0.529	0.535	0.526	0.559	0.552	0.561
	0.02	0.574	0.572	0.556	0.529	0.535	0.553	0.549	0.544	0.561
	0.05	0.564	0.555	0.540	0.532	0.530	0.548	0.571	0.555	0.547
heart	0.01	0.659	0.647	0.833	0.623	0.681	0.407	0.659	0.739	0.638
	0.02	0.659	0.647	0.833	0.623	0.678	0.428	0.668	0.746	0.638
	0.05	0.659	0.647	0.833	0.623	0.681	0.440	0.666	0.749	0.638
satimage	0.01	0.812	0.828	0.600	0.813	0.540	0.909	0.930	0.896	0.916
	0.02	0.829	0.847	0.632	0.861	0.548	0.785	0.919	0.880	0.898
	0.05	0.841	0.858	0.715	0.893	0.536	0.712	0.895	0.868	0.892
splice	0.01	0.713	0.748	0.368	0.541	0.737	0.765	0.778	0.768	0.845
	0.02	0.754	0.765	0.343	0.588	0.744	0.761	0.793	0.783	0.848
	0.05	0.734	0.764	0.377	0.643	0.723	0.764	0.785	0.777	0.849
thyroid	0.01	0.534	0.720	0.745	0.681	0.504	0.259	0.111	0.071	0.256
	0.02	0.534	0.720	0.745	0.681	0.505	0.259	0.111	0.071	0.256
	0.05	0.534	0.720	0.745	0.681	0.485	0.259	0.111	0.071	0.256
titanic	0.01	0.525	0.534	0.602	0.502	0.456	0.520	0.525	0.525	0.461
	0.02	0.496	0.498	0.659	0.513	0.526	0.492	0.503	0.503	0.472
	0.05	0.526	0.521	0.644	0.538	0.505	0.499	0.512	0.512	0.433
twonorm	0.01	0.905	0.902	0.161	0.439	0.846	0.812	0.889	0.897	0.875
	0.02	0.896	0.889	0.197	0.572	0.821	0.803	0.892	0.901	0.858
	0.05	0.905	0.903	0.396	0.754	0.781	0.765	0.858	0.874	0.807
waveform	0.01	0.890	0.881	0.243	0.477	0.861	0.724	0.887	0.889	0.861
	0.02	0.901	0.890	0.181	0.602	0.817	0.690	0.887	0.890	0.861
	0.05	0.885	0.873	0.236	0.757	0.798	0.705	0.847	0.874	0.831
Average		0.661	0.685	0.530	0.608	0.596	0.594	0.629	0.622	0.623
Comp. time		1.00	11.7	5.35	751	12.4	85.5			8.70

Table 3: Mean AUC values for outlier detection over 20 trials for the benchmark data sets. All the methods are implemented using the R environment, where quadratic programs in KMM are solved by the *ipop* optimizer (Karatzoglou et al., 2004), the *ksvm* routine is used for OSVM (Karatzoglou et al., 2004), and the *lofactor* routine is used for LOF (Fernandez, 2005).

Acknowledgments

The authors wish to thank Issei Sato for fruitful discussion and helpful comments. The authors would also like to thank the anonymous referees whose comments helped to improve the paper further. This work was supported by MEXT (20680007), SCAT, and AOARD.

Appendix A. Existence of the Inverse Matrix of $\widehat{G}$

Here we prove Lemma 1.

Let us consider the following system of linear equations:

$$\begin{pmatrix} \widehat{H} & -\widehat{E}^\top \\ -\widehat{E} & O_{|\widehat{\mathcal{A}}|\times|\widehat{\mathcal{A}}|} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0_b \\ 0_{|\widehat{\mathcal{A}}|} \end{pmatrix}, \tag{38}$$

where x and y are b- and $|\widehat{\mathcal{A}}|$-dimensional vectors, respectively. From the upper half of Eq. (38), we have

$$x = \widehat{H}^{-1}\widehat{E}^\top y.$$

Substituting this into the lower half of Eq. (38), we have

$$\widehat{E}\widehat{H}^{-1}\widehat{E}^\top y = 0_{|\widehat{\mathcal{A}}|}.$$

From the definition, the rank of the matrix $\widehat{E}$ is $|\widehat{\mathcal{A}}|$, that is, $\widehat{E}$ is a row-full rank matrix. As a result, the matrix $\widehat{E}\widehat{H}^{-1}\widehat{E}^\top$ is invertible. Therefore, Eq. (38) has the unique solution $x = 0_b$ and $y = 0_{|\widehat{\mathcal{A}}|}$. This implies that $\widehat{G}$ is invertible.

Appendix B. Active Set of LSIF

Here, we prove Theorem 2.

We prove that the active set $\mathcal{A}$ does not change under the infinitesimal shift of H and h if the strict complementarity condition is satisfied. We regard the pair of a symmetric matrix and a vector (H', h') as an element in the $(\frac{b(b+1)}{2} + b)$-dimensional Euclidean space. We consider the following linear equation:

$$\begin{pmatrix} \alpha' \\ \xi' \end{pmatrix} = \begin{pmatrix} H' & -E^\top \\ -E & O_{|\mathcal{A}|\times|\mathcal{A}|} \end{pmatrix}^{-1} \begin{pmatrix} h' - \lambda 1_b \\ 0_{|\mathcal{A}|} \end{pmatrix},$$

where E is the $|\mathcal{A}| \times b$ indicator matrix determined from the active set $\mathcal{A}$ (see Section 2.3 for the detailed definition). If $H' = H$ and $h' = h$ hold, the solution $(\alpha', \xi') = (\alpha^*(\lambda), \xi^*(\lambda))$ satisfies

$$\begin{aligned} &\alpha'_\ell = 0,\ \xi'_\ell > 0, \quad {}^\forall \ell \in \mathcal{A}, \\ &\alpha'_\ell > 0,\ \xi'_\ell = 0, \quad {}^\forall \ell \notin \mathcal{A}, \end{aligned} \tag{39}$$

because of the strict complementarity condition. On the other hand, if the norm of $(H', h') - (H, h)$ is infinitesimal, the solution (α', ξ') also satisfies Eq. (39) because of the continuity of the relation between (H', h') and (α', ξ').

As a result, there exists an ε-ball B_ε in $\mathbb{R}^{\frac{b(b+1)}{2}+b}$ such that the equality $\mathcal{A} = \{\ell \mid \alpha'_\ell = 0\}$ holds for any $(H', h') \in B_\varepsilon$. Therefore, we have $P(\mathcal{A} \neq \widehat{\mathcal{A}}) \leq P((\widehat{H}, \widehat{h}) \notin B_\varepsilon)$. Due to the large deviation principle (Dembo and Zeitouni, 1998), there is a positive constant c such that

$$-\frac{1}{\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}} \log P((\widehat{H}, \widehat{h}) \notin B_\varepsilon) > c > 0,$$

if $\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}$ is large enough. Thus, asymptotically $P(\widehat{\mathcal{A}} \neq \mathcal{A}) < e^{-c \min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}}$ holds.

Appendix C. Learning Curve of LSIF

Here, we prove Theorem 3.

Let us consider the ideal problem (7). Let $\alpha^*(\lambda)$ and $\xi^*(\lambda)$ be the optimal parameter and Lagrange multiplier (that is, the KKT conditions are fulfilled; see Section 2.3) and let $\xi^{*\prime}(\lambda)$ be the vector of non-zero elements of $\xi^*(\lambda)$ defined in the same way as Eq. (11). Then $\alpha^*(\lambda)$ and $\xi^{*\prime}(\lambda)$ satisfy

$$G \begin{pmatrix} \alpha^*(\lambda) \\ \xi^{*\prime}(\lambda) \end{pmatrix} = \begin{pmatrix} h - \lambda 1_b \\ 0_{|\mathcal{A}|} \end{pmatrix}, \tag{40}$$

where

$$G = \begin{pmatrix} H & -E^\top \\ -E & O_{|\mathcal{A}| \times |\mathcal{A}|} \end{pmatrix}.$$

From the central limit theorem and the assumption (18), we have

$$\widehat{h} = h + O_p\left(\frac{1}{\sqrt{n_{\mathrm{te}}}}\right) = h + o_p\left(\frac{1}{n_{\mathrm{tr}}}\right), \tag{41}$$

where O_p and o_p denote the asymptotic order in probability. The assumption (a) implies that the equality

$$\widehat{E} = E \tag{42}$$

holds with exponentially high probability due to Theorem 2. Note that $\widehat{G}$ is the same size as G if $\widehat{E} = E$. Thus we have

$$\widehat{G} = G + \delta G,$$

where

$$\delta G = \begin{pmatrix} \delta H & O_{b \times |\mathcal{A}|} \\ O_{|\mathcal{A}| \times b} & O_{|\mathcal{A}| \times |\mathcal{A}|} \end{pmatrix},$$
$$\delta H = \widehat{H} - H. \tag{43}$$

Combining Eqs. (12), (40), (41), and (42), we have

$$\begin{pmatrix} \widehat{\alpha}(\lambda) \\ \widehat{\xi}'(\lambda) \end{pmatrix} = \widehat{G}^{-1} G \begin{pmatrix} \alpha^*(\lambda) \\ \xi^{*\prime}(\lambda) \end{pmatrix} + o_p\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{44}$$

The matrix Taylor expansion (Petersen and Pedersen, 2007) yields

$$\widehat{G}^{-1} = G^{-1} - G^{-1}\delta G G^{-1} + G^{-1}\delta G G^{-1}\delta G G^{-1} - \cdots, \tag{45}$$

and the central limit theorem asserts that

$$\delta H = O_p\left(\frac{1}{\sqrt{n_{\mathrm{tr}}}}\right). \tag{46}$$

Combining Eqs. (44), (45), (14), and (46), we have

$$\begin{aligned}\delta\alpha &= \widehat{\alpha}(\lambda) - \alpha^*(\lambda) &(47)\\ &= -A\delta H\alpha^*(\lambda) + A\delta HA\delta H\alpha^*(\lambda) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). &(48)\end{aligned}$$

Through direct calculation, we can confirm that

$$AHA = A. \tag{49}$$

Similar to Eq. (15), it holds that

$$\alpha^*(\lambda) = A(h - \lambda 1_b). \tag{50}$$

From Eqs. (49) and (50), we have

$$A(H\alpha^*(\lambda) - h) = -\lambda A 1_b. \tag{51}$$

Eqs. (43), (4), and (3) imply

$$\mathbb{E}[\delta H] = O_{b\times b}. \tag{52}$$

From Eqs. (2) and (47), we have

$$J(\widehat{\alpha}(\lambda)) = J(\alpha^*(\lambda)) + \frac{1}{2}\delta\alpha^\top H\delta\alpha + (H\alpha^*(\lambda) - h)^\top\delta\alpha. \tag{53}$$

From Eqs. (46), (48), and (49), we have

$$\begin{aligned}\mathbb{E}\left[\delta\alpha^\top H\delta\alpha\right] &= \mathrm{tr}(H\,\mathbb{E}\left[\delta\alpha\delta\alpha^\top\right])\\ &= \mathrm{tr}(AHA\,\mathbb{E}\left[(\delta H\alpha^*(\lambda))(\delta H\alpha^*(\lambda))^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right)\\ &= \mathrm{tr}(A\,\mathbb{E}\left[(\delta H\alpha^*(\lambda))(\delta H\alpha^*(\lambda))^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). &(54)\end{aligned}$$

From Eqs. (48), (51), and (52), we have

$$\begin{aligned}\mathbb{E}\left[\delta\alpha^\top(H\alpha^*(\lambda) - h)\right] &= -\,\mathbb{E}\left[(\delta H\alpha^*(\lambda) - \delta HA\delta H\alpha^*(\lambda))^\top A(H\alpha^*(\lambda) - h)\right] + o\left(\frac{1}{n_{\mathrm{tr}}}\right)\\ &= \mathbb{E}\left[(\delta H\alpha^*(\lambda) - \delta HA\delta H\alpha^*(\lambda))^\top\lambda A 1_b\right] + o\left(\frac{1}{n_{\mathrm{tr}}}\right)\\ &= -\,\lambda\,\mathrm{tr}(A\,\mathbb{E}\left[(\delta H\alpha^*(\lambda))(\delta HA1_b)^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). &(55)\end{aligned}$$

Combining Eqs. (53), (54), and (55), we have

$$\mathbb{E}[J(\widehat{\alpha}(\lambda))] = J(\alpha^*(\lambda)) + \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(A\,\mathbb{E}\left[(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))^\top\right]) - \frac{\lambda}{n_{\mathrm{tr}}}\mathrm{tr}(A\,\mathbb{E}\left[(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta HA1_b)^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right).$$

According to the central limit theorem, $\sqrt{n_{\mathrm{tr}}}\delta H_{i,j}$ asymptotically follows the normal distribution with mean zero and variance

$$\int \varphi_i^2(x)\varphi_j^2(x)p_{\mathrm{tr}}(x)dx - H_{i,j}^2,$$

and the asymptotic covariance between $\sqrt{n_{\mathrm{tr}}}\delta H_{i,j}$ and $\sqrt{n_{\mathrm{tr}}}\delta H_{i',j'}$ is given by

$$\int \varphi_i(x)\varphi_j(x)\varphi_{i'}(x)\varphi_{j'}(x)p_{\mathrm{tr}}(x)dx - H_{i,j}H_{i',j'}.$$

Then we have

$$\lim_{n_{\mathrm{tr}}\to\infty}\mathbb{E}\left[(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))^\top\right] = C_{w^*,w^*},$$
$$\lim_{n_{\mathrm{tr}}\to\infty}\mathbb{E}\left[(\sqrt{n_{\mathrm{tr}}}\delta H\alpha^*(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta HA1_b)^\top\right] = C_{w^*,v},$$

where $C_{w,w'}$ is the $b\times b$ covariance matrix with the (ℓ,ℓ')-th element being the covariance between $w(x)\varphi_\ell(x)$ and $w'(x)\varphi_{\ell'}(x)$ under $p_{\mathrm{tr}}(x)$. Then we obtain Eq. (19).

Appendix D. Regularization Path of LSIF

Here, we derive the regularization path tracking algorithm given in Figure 1.

When λ is greater than or equal to $\max_k \widehat{h}_k$, the solution of the KKT conditions (9)–(10) is provided as $\alpha = 0_b$, $\xi = \lambda 1_b - \widehat{h} \geq 0_b$. Therefore, the initial value of λ_0 is $\max_k \widehat{h}_k$, and the corresponding optimal solution is $\widehat{\alpha}(\lambda_0) = 0_b$.

Since $\widehat{\xi}'(\lambda)$ corresponds to non-zero elements of $\widehat{\xi}(\lambda)$ as shown in Eq. (11), we have

$$\widehat{\xi}_j(\lambda) = \begin{cases} \widehat{\xi}'_i(\lambda) & \text{if } j = \widehat{j}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{56}$$

When λ is decreased, the solutions $\widehat{\alpha}(\lambda)$ and $\widehat{\xi}(\lambda)$ still satisfy Eqs. (12) and (56) as long as the active set $\widehat{\mathcal{A}}$ remains unchanged. Change points of the active set can be found by examining the non-negativity conditions of $\widehat{\alpha}(\lambda)$ and $\widehat{\xi}(\lambda)$ as follows. Suppose λ is decreased and the non-negativity condition

$$\begin{pmatrix} \widehat{\alpha}(\lambda) \\ \widehat{\xi}(\lambda) \end{pmatrix} \geq 0_{2b}$$

is violated at $\lambda = \lambda'$. That is, both $\widehat{\alpha}(\lambda') \geq 0_b$ and $\widehat{\xi}(\lambda') \geq 0_b$ hold, and either $\widehat{\alpha}(\lambda'-\varepsilon) \geq 0_b$ or $\widehat{\xi}(\lambda'-\varepsilon) \geq 0_b$ is violated for any $\varepsilon > 0$. If $\widehat{\alpha}_j(\lambda') = 0$ for $j \notin \widehat{\mathcal{A}}$, j should be added to the active set

$\widehat{\mathcal{A}}$; on the other hand, if $\widehat{\xi}_j(\lambda') = 0$ for some $j \in \widehat{\mathcal{A}}$, $\widehat{\alpha}_j(\lambda')$ will take a positive value and therefore j should be removed from the active set $\widehat{\mathcal{A}}$. Then, for the updated active set, we compute the solutions by Eqs. (12) and (56). Iterating this procedure until λ reaches zero, we can obtain the entire regularization path.

Note that we omitted some minor exceptional cases for the sake of simplicity—treatments for all possible exceptions and the rigorous convergence property are exhaustively studied in Best (1982).

Appendix E. Negative Index Set of $\beta^\circ(\lambda)$

Here we prove Theorem 4.

As explained in Appendix B, we regard the pair of a symmetric matrix and a vector (H', h') as an element in the $(\frac{b(b+1)}{2} + b)$-dimensional Euclidean space.

We consider the linear equation

$$\beta' = (H' + \lambda I_b)^{-1} h'.$$

Due to the assumption, for $H' = H$ and $h' = h$, we have

$$\beta'_\ell \neq 0,\ \ell = 1, 2, \ldots, b. \tag{57}$$

On the other hand, if the norm of $(H', h') - (H, h)$ is infinitesimal, the solution β' also satisfies Eq. (57), and the sign of β'_ℓ is same as that of β_ℓ for $\ell = 1, 2, \ldots, b$, because of the continuity of the relation between (H', h') and β'.

As a result, there exists an ε-ball B_ε in $\mathbb{R}^{\frac{b(b+1)}{2}+b}$ such that the equality $\mathcal{B} = \widetilde{\mathcal{B}}$ holds for any $(H', h') \in B_\varepsilon$. Therefore, we have $P(\mathcal{B} \neq \widetilde{\mathcal{B}}) \leq P((\widehat{H}, \widehat{h}) \notin B_\varepsilon)$. Due to the large deviation principle (Dembo and Zeitouni, 1998), there is a positive constant c such that

$$-\frac{1}{\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}} \log P((\widehat{H}, \widehat{h}) \notin B_\varepsilon) > c > 0,$$

if $\min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}$ is large enough. Thus, asymptotically $P(\mathcal{B} \neq \widetilde{\mathcal{B}}) < e^{-c \min\{n_{\mathrm{tr}}, n_{\mathrm{te}}\}}$ holds.

Appendix F. Learning Curve of uLSIF

Here, we prove Theorem 5.

Let

$$\widehat{B}_\lambda = \widehat{H} + \lambda I_b.$$

The matrix Taylor expansion (Petersen and Pedersen, 2007) yields

$$\widehat{B}_\lambda^{-1} = B_\lambda^{-1} - B_\lambda^{-1} \delta H B_\lambda^{-1} + B_\lambda^{-1} \delta H B_\lambda^{-1} \delta H B_\lambda^{-1} - \cdots. \tag{58}$$

Let $\widetilde{\mathcal{B}} \subset \{1, 2, \ldots, b\}$ be the set of negative indices of $\widetilde{\beta}(\lambda)$, that is,

$$\widetilde{\mathcal{B}} = \{\ell \mid \widetilde{\beta}_\ell(\lambda) < 0,\ \ell = 1, 2, \ldots, b\}.$$

Let $\widehat{D}$ be the b-dimensional diagonal matrix with the ℓ-th diagonal element

$$\widehat{D}_{\ell,\ell} = \begin{cases} 0 & \ell \in \widetilde{\mathcal{B}}, \\ 1 & \text{otherwise}. \end{cases}$$

The assumption (a) implies that the equality

$$\widehat{D} = D \tag{59}$$

holds with exponentially high probability due to Theorem 4. Combining Eqs. (59), (41), (58), and (24), we have

$$\begin{aligned}\delta\beta &= \widehat{\beta}(\lambda) - \beta^*(\lambda)\\ &= \widehat{D}\widehat{B}_\lambda^{-1}\widehat{h} - DB_\lambda^{-1}h\\ &= -DB_\lambda^{-1}\delta H\beta^\circ(\lambda) + DB_\lambda^{-1}\delta HB_\lambda^{-1}\delta H\beta^\circ(\lambda) + o\left(\frac{1}{n_{\mathrm{tr}}}\right).\end{aligned} \tag{60}$$

From Eqs. (46) and (60), we have

$$\mathbb{E}\left[\delta\beta^\top H\delta\beta\right] = \mathrm{tr}(B_\lambda^{-1}DHDB_\lambda^{-1}\,\mathbb{E}\left[(\delta H\beta^\circ(\lambda))(\delta H\beta^\circ(\lambda))^\top\right]) + o\left(\frac{1}{n_{\mathrm{tr}}}\right). \tag{61}$$

From Eqs. (52) and (24), we have

$$\begin{aligned}\mathbb{E}\left[\delta\beta^\top(H\beta^*(\lambda)-h)\right] =& \mathbb{E}\left[(-\delta H\beta^\circ(\lambda) + \delta HB_\lambda^{-1}\delta H\beta^\circ(\lambda))^\top B_\lambda^{-1}D(H\beta^*(\lambda)-h)\right]\\ &+ o\left(\frac{1}{n_{\mathrm{tr}}}\right)\\ =& \mathbb{E}\left[\mathrm{tr}(B_\lambda^{-1}(\delta H\beta^\circ(\lambda))(\delta HB_\lambda^{-1}D(H\beta^*(\lambda)-h))^\top)\right]\\ &+ o\left(\frac{1}{n_{\mathrm{tr}}}\right).\end{aligned} \tag{62}$$

Combining Eqs. (53), (61), and (62), we have

$$\begin{aligned}\mathbb{E}\left[J(\widehat{\beta}(\lambda))\right] =& J(\beta^*(\lambda)) + \frac{1}{2n_{\mathrm{tr}}}\mathrm{tr}(B_\lambda^{-1}DHDB_\lambda^{-1}\,\mathbb{E}[(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))^\top)\\ &+ \frac{1}{n_{\mathrm{tr}}}\mathrm{tr}(B_\lambda^{-1}\,\mathbb{E}[(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta HB_\lambda^{-1}D(H\beta^*(\lambda)-h))^\top) + o\left(\frac{1}{n_{\mathrm{tr}}}\right).\end{aligned}$$

According to the central limit theorem, we have

$$\begin{aligned}\lim_{n_{\mathrm{tr}}\to\infty}\mathbb{E}[(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))^\top] &= C_{w^\circ,w^\circ},\\ \lim_{n_{\mathrm{tr}}\to\infty}\mathbb{E}[(\sqrt{n_{\mathrm{tr}}}\delta H\beta^\circ(\lambda))(\sqrt{n_{\mathrm{tr}}}\delta HB_\lambda^{-1}D(H\beta^*(\lambda)-h))^\top] &= C_{w^\circ,u}.\end{aligned}$$

Then we obtain Eq. (25).

Appendix G. 'Norm' Upper Bound of Approximation Error for uLSIF

Here we prove Theorem 6.

Using the weighted norm (27), we can express $\mathrm{diff}(\lambda)$ as

$$\mathrm{diff}(\lambda) = \frac{\inf_{\lambda'\geq 0}\|\widehat{\alpha}(\lambda') - \widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}}\widehat{w}(x_i^{\mathrm{tr}};\widehat{\beta}(\lambda))}.$$

As shown in Appendix D, $\widehat{\alpha}(\lambda') = 0_b$ holds for some large λ'. Then we immediately have

$$\mathrm{diff}(\lambda) \leq \frac{\|\widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))},$$

which proves Eq. (28). Let $\kappa_{\max}$ be the largest eigenvalue of $\widehat{H}$. Then $\|\widehat{\beta}(\lambda)\|_{\widehat{H}}$ can be upper bounded as

$$\|\widehat{\beta}(\lambda)\|_{\widehat{H}} \leq \sqrt{\kappa_{\max}}\|\widehat{\beta}(\lambda)\|_2 \leq \sqrt{\kappa_{\max}}\|\widetilde{\beta}(\lambda)\|_2,$$

where the first inequality may be confirmed by eigen-decomposing $\widehat{H}$ and the second inequality is clear from the definitions of $\widehat{\beta}(\lambda)$ and $\widetilde{\beta}(\lambda)$. Let $\kappa_{\min}$ be the smallest eigenvalue of $\widehat{H}$. Then an upper bound of $\|\widetilde{\beta}(\lambda)\|_2^2$ is given as

$$\|\widetilde{\beta}(\lambda)\|_2^2 = \widehat{h}^\top (\widehat{H} + \lambda I_b)^{-2}\widehat{h} \leq \frac{1}{(\kappa_{\min} + \lambda)^2}\|\widehat{h}\|_2^2 \leq \frac{1}{\lambda^2}\|\widehat{h}\|_2^2,$$

where the last inequality follows from $\kappa_{\min} > 0$.

Now we have

$$\begin{aligned}
\frac{\|\widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} w(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))} &\leq \frac{1}{\sum_{i=1}^{n_{\mathrm{tr}}} w(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))} \frac{\sqrt{\kappa_{\max}}\|\widehat{h}\|_2}{\lambda} \\
&= \frac{1}{\sum_{i=1}^{n_{\mathrm{tr}}} \sum_{\ell=1}^{b} \varphi_\ell(x_i^{\mathrm{tr}})\widehat{\beta}_\ell(\lambda)/\|\widehat{\beta}(\lambda)\|_1} \frac{\sqrt{\kappa_{\max}}\|\widehat{h}\|_2}{\lambda\|\widehat{\beta}(\lambda)\|_1}.
\end{aligned}$$

For the denominator of the above expression, we have

$$\sum_{i=1}^{n_{\mathrm{tr}}} \sum_{\ell=1}^{b} \varphi_\ell(x_i^{\mathrm{tr}}) \frac{\widehat{\beta}_\ell(\lambda)}{\|\widehat{\beta}(\lambda)\|_1} \geq \min_{\ell'}\Big(\sum_{i=1}^{n_{\mathrm{tr}}} \varphi_{\ell'}(x_i^{\mathrm{tr}})\Big) \cdot \sum_{\ell=1}^{b} \frac{\widehat{\beta}_\ell(\lambda)}{\|\widehat{\beta}(\lambda)\|_1} = \min_{\ell} \sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}}),$$

where the last equality follows from the non-negativity of $\widehat{\beta}_\ell(\lambda)$. The reciprocal of $\|\widehat{h}\|_2/\|\widehat{\beta}(\lambda)\|_1$ is lower bounded as follows:

$$\frac{\|\widehat{\beta}(\lambda)\|_1}{\|\widehat{h}\|_2} = \left\|\max\left\{\frac{\widetilde{\beta}(\lambda)}{\|\widehat{h}\|_2}, 0\right\}\right\|_1 \geq \left\|\max\left\{\frac{\widetilde{\beta}(\lambda)}{\|\widehat{h}\|_2}, 0\right\}\right\|_\infty = \max_{\ell} \frac{\widetilde{\beta}_\ell(\lambda)}{\|\widehat{h}\|_2},$$

where the last equality follows from the fact that there is an ℓ such that $\widetilde{\beta}_\ell(\lambda) > 0$; otherwise, we have $\sum_{i=1}^{n_{\mathrm{tr}}} w(x_i^{\mathrm{tr}}; \widehat{\beta}) = 0$ which contradicts to the assumption of the theorem. Let us put

$$\kappa e = \frac{\widetilde{\beta}(\lambda)}{\|\widehat{h}\|_2},$$

where $\kappa > 0$ and $e \in \mathbb{R}^b$ such that $\|e\|_2 = 1$. Then we have

$$(\kappa_{\max} + \lambda)^{-1} \leq \kappa \text{ and } e^\top \widehat{h} > 0.$$

Note that there exists an ℓ such that $e_\ell > 0$. Then, we have

$$\max_\ell \frac{\widetilde{\beta}_\ell(\lambda)}{\|\widehat{h}\|_2} = \max_\ell \kappa e_\ell = \kappa \max_\ell e_\ell \geq \frac{1}{\kappa_{\max}+\lambda} \max_\ell e_\ell$$
$$\geq \frac{1}{\kappa_{\max}+\lambda} \min_e \{\max_\ell e_\ell \mid e^\top e = 1, e^\top \widehat{h}/\|\widehat{h}\|_1 > 0\}.$$

Now we prove the following lemma.

Lemma 8 *Let $p_1, p_2, \ldots, p_b$ $(b \geq 2)$ be positive numbers such that*

$$\sum_{\ell=1}^{b} p_\ell = 1,$$

and let

$$\varepsilon = \frac{1}{\sqrt{b}} \min_\ell \frac{p_\ell}{1-p_\ell}.$$

Then, there exists no $e = (e_1, e_2, \ldots, e_b) \in \mathbb{R}^b$ such that the three conditions,

$$\sum_{\ell=1}^{b} e_\ell^2 = 1, \ \sum_{\ell=1}^{b} p_\ell e_\ell > 0, \text{ and } e_\ell < \varepsilon \text{ for } \ell = 1, 2, \ldots, b$$

are satisfied at the same time.

Proof We suppose that $e \in \mathbb{R}^b$ satisfies the three conditions. If $\min_\ell p_\ell/(1-p_\ell) > 1$, we have $p_\ell > 1/2$ for all ℓ. However, this is contradictory to $\sum_{\ell=1}^b p_\ell = 1$. Therefore, we have

$$\min_\ell p_\ell/(1-p_\ell) \leq 1,$$

from which we have

$$\varepsilon \leq 1/\sqrt{b}.$$

The equality constraint $\sum_{\ell=1}^b e_\ell^2 = 1$ implies the condition that there exists an e_i such that $|e_i| \geq 1/\sqrt{b}$. Moreover, we have $e_1, e_2, \ldots, e_b < \varepsilon \leq 1/\sqrt{b}$, and thus there is an e_i such that $e_i \leq -1/\sqrt{b}$. Hence, we have

$$\frac{p_i}{\sqrt{b}} \leq -p_i e_i < \sum_{\ell \neq i} p_\ell e_\ell < \sum_{\ell \neq i} p_\ell \frac{1}{\sqrt{b}} \min_k \frac{p_k}{1-p_k} = (1-p_i)\frac{1}{\sqrt{b}} \min_k \frac{p_k}{1-p_k} \leq \frac{p_i}{\sqrt{b}}.$$

This results in contradiction. ■

Let $p_\ell = \widehat{h}_\ell/\|\widehat{h}\|_1$ and we use Lemma 8. Note that any element of $\widehat{h}$ is positive. Then, we have

$$\frac{\|\widehat{\beta}(\lambda)\|_1}{\|\widehat{h}\|_2} \geq \frac{1}{\kappa_{\max}+\lambda} \cdot \frac{1}{\sqrt{b}} \min_\ell \frac{p_\ell}{\sum_{i \neq \ell} p_i}.$$

Moreover, we have

$$\min_\ell \frac{p_\ell}{\sum_{i \neq \ell} p_i} \geq \frac{\min_\ell \widehat{h}_\ell}{\sum_{\ell'=1}^b \widehat{h}_{\ell'}} = \frac{\min_\ell \sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}})}{\sum_{\ell'=1}^b \sum_{j=1}^{n_{\mathrm{te}}} \varphi_{\ell'}(x_j^{\mathrm{te}})} \geq \frac{\min_\ell \sum_{j=1}^{n_{\mathrm{te}}} \varphi_\ell(x_j^{\mathrm{te}})}{n_{\mathrm{te}} b},$$

where the last inequality follows from the assumption $0 < \varphi_\ell(x) \leq 1$. Therefore, we have the inequality

$$\begin{aligned}
&\frac{1}{\sum_{i=1}^{n} w(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))} \frac{\sqrt{\kappa_{\max}} \|\widehat{h}\|_2}{\lambda} \\
&\leq b\sqrt{b\,\kappa_{\max}} \left(1 + \frac{\kappa_{\max}}{\lambda}\right) \frac{1}{\min_\ell \sum_{i=1}^{n_{\mathrm{tr}}} \varphi_\ell(x_i^{\mathrm{tr}})} \cdot \frac{n_{\mathrm{te}}}{\min_{\ell'} \sum_{j=1}^{n_{\mathrm{te}}} \varphi_{\ell'}(x_j^{\mathrm{te}})}. \qquad (63)
\end{aligned}$$

An upper bound of $\kappa_{\max}$ is given as follows. For all $a \in \mathbb{R}^b$, the inequality

$$-\sum_{\ell=1}^{b} |a_\ell| \varphi_\ell(x) \leq \sum_{\ell=1}^{b} a_\ell \varphi_\ell(x) \leq \sum_{\ell=1}^{b} |a_\ell| \varphi_\ell(x) \qquad (64)$$

holds because of the positivity of $\varphi_\ell(x)$. Let us define $\overline{a} \in \mathbb{R}^b$ for given $a \in \mathbb{R}^b$ as

$$\overline{a} \;=\; (|a_1|, |a_2|, \ldots, |a_b|)^\top.$$

Note that $\|\overline{a}\|_2 = \|a\|_2$ holds. Then, using Eq. (64), we obtain the inequality

$$a^\top \widehat{H} a = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} a_\ell \varphi_\ell(x_i^{\mathrm{tr}}) \right)^2 \leq \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} |a_\ell| \varphi_\ell(x_i^{\mathrm{tr}}) \right)^2 = \overline{a}^\top \widehat{H} \overline{a},$$

for any $a \in \mathbb{R}^b$. Therefore, we obtain

$$\max_{\|a\|_2=1} a^\top \widehat{H} a \leq \max_{\|a\|_2=1} \overline{a}^\top \widehat{H} \overline{a} = \max_{\|a\|_2=1,\, a \geq 0_b} a^\top \widehat{H} a, \qquad (65)$$

where the last equality is derived from the relation,

$$\{\overline{a} \mid \|a\|_2 = 1,\, a \in \mathbb{R}^b\} \;=\; \{a \mid \|a\|_2 = 1,\, a \geq 0_b,\, a \in \mathbb{R}^b\}.$$

On the other hand, due to the additional constraint $a \geq 0_b$, the inequality

$$\max_{\|a\|_2=1,\, a \geq 0_b} a^\top \widehat{H} a \;\leq\; \max_{\|a\|_2=1} a^\top \widehat{H} a \qquad (66)$$

holds. From Eqs. (65) and (66), we have

$$\kappa_{\max} = \max_{\|a\|_2=1} a^\top \widehat{H} a = \max_{\|a\|_2=1, a \geq 0_b} a^\top \widehat{H} a = \max_{\|a\|_2=1,\, a \geq 0_b} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} a_\ell \varphi_\ell(x_i^{\mathrm{tr}}) \right)^2.$$

Using the assumption $0 < \varphi_\ell(x) \leq 1$, we have

$$\begin{aligned}
\kappa_{\max} \;&=\; \max_{\|a\|_2=1,\, a \geq 0_b} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} a_\ell \varphi_\ell(x_i^{\mathrm{tr}}) \right)^2 \;\leq\; \max_{\|a\|_2=1,\, a \geq 0_b} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \left(\sum_{\ell=1}^{b} a_\ell \right)^2 \\
&=\; \max_{\|a\|_2=1,\, a \geq 0_b} \left(\sum_{\ell=1}^{b} a_\ell \right)^2 \;\leq\; \max_{\|a\|_2=1,\, a \geq 0_b} b \cdot \sum_{\ell=1}^{b} a_\ell^2 \\
&=\; b, \qquad (67)
\end{aligned}$$

where the Schwarz inequality for a and 1_b is used in the last inequality. The inequalities (63) and (67) lead to the inequality (29).

It is clear that the upper bound (29) is a decreasing function of λ (> 0). For the Gaussian basis function, $\varphi_\ell(x)$ is an increasing function with respect to the Gaussian width σ. Thus, Eq. (29) is a decreasing function of σ.

Appendix H. 'Bridge' Upper Bound of Approximation Error for uLSIF

Here we prove Theorem 7.

From the triangle inequality, we obtain

$$\mathrm{diff}(\lambda) \leq \frac{\inf_{\lambda' \geq 0} \|\widehat{\alpha}(\lambda') - \widehat{\gamma}(\lambda)\|_{\widehat{H}} + \|\widehat{\gamma}(\lambda) - \widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))}. \tag{68}$$

We derive an upper bound of the first term.

First, we show that the LSIF optimization problem (6) is equivalently expressed as

$$\min_{\alpha \in \mathbb{R}^b} \left[\frac{1}{2} \alpha^\top \widehat{H} \alpha - \widehat{h}^\top \alpha \right]$$
$$\text{subject to } \alpha \geq 0_b, \ \ 1_b^\top \alpha \leq c,$$

which we refer to as LSIF$'$. The KKT conditions of LSIF (6) are given as

$$\begin{cases} \widehat{H}\alpha - \widehat{h} + \lambda 1_b - \mu = 0_b, \\ \alpha \geq 0_b, \ \ \mu \geq 0_b, \ \ \alpha^\top \mu = 0, \end{cases}$$

where μ is the Lagrange multiplier vector. Similarly, the KKT conditions of LSIF$'$ are given as

$$\begin{cases} \widehat{H}\alpha - \widehat{h} + \mu_0 1_b - \mu = 0_b, \\ \alpha \geq 0_b, \ \ \mu \geq 0_b, \ \ \alpha^\top \mu = 0, \\ 1_b^\top \alpha - c \leq 0, \ \mu_0 \geq 0, \ \ (1_b^\top \alpha - c)\mu_0 = 0, \end{cases} \tag{69}$$

where μ and μ_0 are the Lagrange multipliers. Let $(\widehat{\alpha}(\lambda), \widehat{\mu}(\lambda))$ be the solution of the KKT conditions of LSIF. Then, we find that $(\alpha, \mu, \mu_0) = (\widehat{\alpha}(\lambda), \widehat{\mu}(\lambda), \lambda)$ is the solution of Eq. (69) with $c = 1_b^\top \widehat{\alpha}(\lambda)$. Note that LSIF$'$ is a strictly convex optimization problem, and thus $\widehat{\alpha}(\lambda)$ is the unique optimal solution. Conversely, when the solution of Eq. (69) is provided as $(\widehat{\alpha}, \widehat{\mu}, \mu_0)$, LSIF with $\lambda = \mu_0$ has the same optimal solution $\widehat{\alpha}$.

When the optimal solution of LSIFq is $\widehat{\gamma}(\lambda)$, the KKT conditions of LSIFq (30) are given as

$$\widehat{H}\widehat{\gamma}(\lambda) - \widehat{h} + \lambda \widehat{\gamma}(\lambda) - \widehat{\eta} = 0_b, \tag{70}$$
$$\widehat{\gamma}(\lambda) \geq 0_b, \ \ \widehat{\eta} \geq 0_b, \ \ \widehat{\gamma}(\lambda)^\top \widehat{\eta} = 0, \tag{71}$$

where $\widehat{\eta}$ is the Lagrange multiplier vector.

Let $\widehat{\alpha}(\lambda_1)$ be the optimal solution of LSIF$'$ with $c = 1_b^\top \widehat{\gamma}(\lambda)$, and suppose that the solution $\widehat{\alpha}(\lambda_1)$ coincides with that of LSIF with $\lambda = \lambda_1$. Then, from Eq. (69), we have

$$\widehat{H}\widehat{\alpha}(\lambda_1) - \widehat{h} + \lambda_1 1_b - \widehat{\mu}(\lambda_1) = 0_b, \tag{72}$$
$$\widehat{\alpha}(\lambda_1) \geq 0_b, \ \ \widehat{\mu}(\lambda_1) \geq 0_b, \ \ \widehat{\alpha}(\lambda_1)^\top \widehat{\mu}(\lambda_1) = 0, \tag{73}$$
$$1_b^\top \widehat{\alpha}(\lambda_1) - 1_b^\top \widehat{\gamma}(\lambda) \leq 0, \ \ \lambda_1 \geq 0, \ \ (1_b^\top \widehat{\alpha}(\lambda_1) - 1_b^\top \widehat{\gamma}(\lambda))\lambda_1 = 0. \tag{74}$$

From Eqs. (70) and (72), we obtain

$$\widehat{H}(\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda)) = -\lambda_1 1_b + \lambda \widehat{\gamma}(\lambda) + \widehat{\mu}(\lambda_1) - \widehat{\eta}. \tag{75}$$

Applying Eqs. (71), (73), (74), and (75), we have

$$\begin{aligned}\inf_{\lambda' \geq 0} \|\widehat{\alpha}(\lambda') - \widehat{\gamma}(\lambda)\|_{\widehat{H}}^2 &\leq (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda))^\top \widehat{H} (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda)) \\ &= -\lambda_1 (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda))^\top 1_b + \lambda (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda))^\top \widehat{\gamma}(\lambda) \\ &\quad + (\widehat{\alpha}(\lambda_1) - \widehat{\gamma}(\lambda))^\top (\widehat{\mu}(\lambda_1) - \widehat{\eta}) \\ &= \lambda (\widehat{\alpha}(\lambda_1)^\top \widehat{\gamma}(\lambda) - \|\widehat{\gamma}(\lambda)\|_2^2) - \widehat{\alpha}(\lambda_1)^\top \widehat{\eta} - \widehat{\gamma}(\lambda)^\top \widehat{\mu}(\lambda_1) \\ &\leq \lambda (\widehat{\alpha}(\lambda_1)^\top \widehat{\gamma}(\lambda) - \|\widehat{\gamma}(\lambda)\|_2^2). \end{aligned} \tag{76}$$

From $\widehat{\alpha}(\lambda_1) \geq 0_b$, $\widehat{\gamma}(\lambda) \geq 0_b$, and $1_b^\top \widehat{\alpha}(\lambda_1) \leq 1_b^\top \widehat{\gamma}(\lambda)$, we have

$$\|\widehat{\alpha}(\lambda_1)\|_1 = 1_b^\top \widehat{\alpha}(\lambda_1) \leq 1_b^\top \widehat{\gamma}(\lambda) \leq \|\widehat{\gamma}(\lambda)\|_1.$$

Then we have the following inequality:

$$\begin{aligned}\widehat{\alpha}(\lambda_1)^\top \widehat{\gamma}(\lambda) &\leq \widehat{\alpha}(\lambda_1)^\top (\|\widehat{\gamma}(\lambda)\|_\infty 1_b) \\ &= \|\widehat{\alpha}(\lambda_1)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty \leq \|\widehat{\gamma}(\lambda)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty. \end{aligned} \tag{77}$$

For p and q such that $1/p + 1/q = 1$ and $1 \leq p, q \leq \infty$, Hölder's inequality states that

$$\|\alpha * \beta\|_1 \leq \|\alpha\|_p \cdot \|\beta\|_q,$$

where $\alpha * \beta$ denotes the element-wise product of α and β. Setting $p = 1$, $q = \infty$, and $\alpha = \beta = \widehat{\gamma}(\lambda)$ in Hölder's inequality, we have

$$\|\widehat{\gamma}(\lambda)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty - \|\widehat{\gamma}(\lambda)\|_2^2 \geq 0. \tag{78}$$

Combining Eqs. (68), (76), (77), and (78), we obtain

$$\mathrm{diff}(\lambda) \leq \frac{\sqrt{\lambda(\|\widehat{\gamma}(\lambda)\|_1 \cdot \|\widehat{\gamma}(\lambda)\|_\infty - \|\widehat{\gamma}(\lambda)\|_2^2)} + \|\widehat{\gamma}(\lambda) - \widehat{\beta}(\lambda)\|_{\widehat{H}}}{\sum_{i=1}^{n_{\mathrm{tr}}} \widehat{w}(x_i^{\mathrm{tr}}; \widehat{\beta}(\lambda))}.$$

Appendix I. Closed Form of LOOCV Score for uLSIF

Here we derive a closed form expression of the LOOCV score for uLSIF (see Figure 2 for the pseudo code).

Let

$$\varphi(x) = (\varphi_1(x), \varphi_2(x), \ldots, \varphi_b(x))^\top.$$

Then the matrix $\widehat{H}$ and the vector $\widehat{h}$ are expressed as

$$\widehat{H} = \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \varphi(x_i^{\mathrm{tr}}) \varphi(x_i^{\mathrm{tr}})^\top,$$

$$\widehat{h} = \frac{1}{n_{\mathrm{te}}} \sum_{j=1}^{n_{\mathrm{te}}} \varphi(x_j^{\mathrm{te}}),$$

and the coefficients $\widetilde{\beta}(\lambda)$ can be computed by

$$\widetilde{\beta}(\lambda) = \widehat{B}_{\lambda}^{-1}\widehat{h}.$$

Let $\widehat{\beta}^{(i)}$ be the estimator obtained without the i-th training sample x_i^{tr} and the i-th test sample x_i^{te}. Then the estimator has the following closed form:

$$\begin{aligned}
\widehat{\beta}^{(i)}(\lambda) &= \max(0_b, \widetilde{\beta}^{(i)}(\lambda)), \\
\widetilde{\beta}^{(i)}(\lambda) &= \left(\frac{1}{n_{\mathrm{tr}}-1}(n_{\mathrm{tr}}\widehat{H} - \varphi(x_i^{\mathrm{tr}})\varphi(x_i^{\mathrm{tr}})^{\top}) + \lambda I_b\right)^{-1} \frac{1}{n_{\mathrm{te}}-1}(n_{\mathrm{te}}\widehat{h} - \varphi(x_j^{\mathrm{te}})).
\end{aligned}$$

Let $\widehat{B} = \widehat{H} + \frac{\lambda(n_{\mathrm{tr}}-1)}{n_{\mathrm{tr}}} I_b$ and $\widetilde{\beta} = \widehat{B}^{-1}\widehat{h}$ in the following calculation. Using the Sherman-Woodbury-Morrison formula (33), we can simplify the expression of $\widetilde{\beta}^{(i)}(\lambda)$ as follows:

$$\begin{aligned}
\widetilde{\beta}^{(i)}(\lambda) =& \frac{n_{\mathrm{tr}}-1}{n_{\mathrm{tr}}}\left(\widehat{B} - \frac{1}{n_{\mathrm{tr}}}\varphi(x_i^{\mathrm{tr}})\varphi(x_i^{\mathrm{tr}})^{\top}\right)^{-1}\left(\frac{n_{\mathrm{te}}}{n_{\mathrm{te}}-1}\widehat{h} - \frac{1}{n_{\mathrm{te}}-1}\varphi(x_i^{\mathrm{te}})\right) \\
=& \frac{n_{\mathrm{tr}}-1}{n_{\mathrm{tr}}}\left(\widehat{B}^{-1} + \frac{1}{n_{\mathrm{tr}} - \varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})\varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\right) \\
& \times \left(\frac{n_{\mathrm{te}}}{n_{\mathrm{te}}-1}\widehat{h} - \frac{1}{n_{\mathrm{te}}-1}\varphi(x_i^{\mathrm{te}})\right) \\
=& \frac{(n_{\mathrm{tr}}-1)n_{\mathrm{te}}}{n_{\mathrm{tr}}(n_{\mathrm{te}}-1)}\left(\widetilde{\beta} + \frac{\varphi(x_i^{\mathrm{tr}})^{\top}\widetilde{\beta}}{n_{\mathrm{tr}} - \varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})\right) \\
& - \frac{(n_{\mathrm{tr}}-1)}{n_{\mathrm{tr}}(n_{\mathrm{te}}-1)}\left(\widehat{B}^{-1}\varphi(x_i^{\mathrm{te}}) + \frac{\varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\varphi(x_i^{\mathrm{te}})}{n_{\mathrm{tr}} - \varphi(x_i^{\mathrm{tr}})^{\top}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})}\widehat{B}^{-1}\varphi(x_i^{\mathrm{tr}})\right).
\end{aligned}$$

Thus the matrix inversion required for computing $\widetilde{\beta}^{(i)}(\lambda)$ for all $i = 1, 2, \ldots, n_{\mathrm{tr}}$ is only $\widehat{B}$. Applying this to Eq. (32) and rearrange the formula, we can compute the LOOCV score analytically.

References

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, 1974.

S. Amari, N. Fujita, and S. Shinomoto. Four types of learning curves. *Neural Computation*, 4(4): 605–618, 1992.

P. Baldi and S. Brunak. *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, 1998.

D. Bertsekas, A. Nedic, and A. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Belmont, MA, 2003.

M. J. Best. An algorithm for the solution of the parametric quadratic programming problem. CORR Report 82-24, Faculty of Mathematics, University of Waterloo, 1982.

S. Bickel and T. Scheffer. Dirichlet-enhanced spam filtering based on biased samples. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004.

A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2000.

G. C. Cawley and N. L. C. Talbot. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Networks*, 17(10):1467–75, 2004.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

K. F. Cheng and C. K. Chu. Semiparametric density estimation under a two-sample density ratio model. *Bernoulli*, 10(4):583–604, 2004.

A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*. Springer-Verlag, New York, 1998.

B. Efron, T. Hastie, R. Tibshirani, and I. Johnstone. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

E. A. Fernandez. *The dprep Package*, 2005. URL `http://math.uprm.edu/ edgar/dprep.pdf`.

G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, Berlin, 1996.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 1996.

H. Hachiya, T. Akiyama, M. Sugiyama, and J. Peters. Adaptive importance sampling with automatic model selection in value function approximation. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI2008)*, pages 1351–1356, Chicago, USA, Jul. 13–17 2008. The AAAI Press.

L. K. Hansen and J. Larsen. Linear unlearning for crossvalidation. *Advances in Computational Mathematics*, 5:269–280, 1996.

W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer Series in Statistics. Springer, Berlin, 2004.

T. Hastie, S. Rosset, R. Tibshirani, and J. ZHu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

J. J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47(1):153–162, 1979.

S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Inlier-based outlier detection via direct density ratio estimation. In *Proceedings of IEEE International Conference on Data Mining (ICDM2008)*, Pisa, Italy, Dec. 15–19 2008.

V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3):55–67, 1970.

J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press, Cambridge, MA, 2007.

A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis. An S4 package for kernel methods in R. *Journal of Statistical Planning and Inference*, 11(9):1–20, 2004.

S. Konishi and G. Kitagawa. Generalized information criteria in model selection. *Biometrika*, 83: 875–890, 1996.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region Newton method for large-scale logistic regression. Technical report, Department of Computer Science, National Taiwan University, 2007. URL `http://www.csie.ntu.edu.tw/~cjlin/liblinear/`.

A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of recognition. *Technicheskaya Kibernetica*, 3, 1969. in Russian.

T. P. Minka. A comparison of numerical optimizers for logistic regression. Technical report, Microsoft Research, 2007. URL `http://research.microsoft.com/~minka/papers/logreg/minka-logreg.pdf`.

J. E. Moody. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 847–854. Morgan Kaufmann Publishers, Inc., 1992.

X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functions and the likelihood ratio by penalized convex risk minimization. In *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2008. MIT Press.

K. B. Petersen and M. S. Pedersen. The matrix cookbook. Technical report, Technical University of Denmark, 2007. URL `http://www2.imm.dtu.dk/pubdb/p.php?3274`.

J. Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–639, 1998.

J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, editors. *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA, 2008.

C. E. Rasmussen, R. M. Neal, G. E. Hinton, D. van Camp, M. Revow, Z. Ghahramani, R. Kustra, and R. Tibshirani. The DELVE manual, 1996. URL `http://www.cs.toronto.edu/~delve/`.

G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for adaboost. *Machine Learning*, 42(3): 287–320, 2001.

K. Scheinberg. An efficient implementation of an active set method for SVMs. *Journal of Machine Learning Research*, 7:2237–2257, 2006.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *Proceedings of the 24th International Conference on Machine learning*, pages 823–830, New York, NY, USA, 2007. ACM.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

M. Stone. Cross-validatory choice and assessment of statistical predictors. *Journal of the Royal Statistical Society B*, 32(2):111–147, 1974.

M. Sugiyama and K.-R. Müller. Input-dependent estimation of generalization error under covariate shift. *Statistics & Decisions*, 23(4):249–279, 2005.

M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, May 2007.

M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1433–1440, Cambridge, MA, 2008a. MIT Press.

M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008b.

R. S. Sutton and G. A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. In *Proceedings of 2008 SIAM International Conference on Data Mining (SDM2008)*, Atlanta, Georgia, USA, 2008.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

G. Wahba. *Spline Model for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia and Pennsylvania, 1990.

P. M. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, 7 (1):117–143, 1995.

J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.

B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine Learning*, New York, NY, 2004. ACM Press.