PAPER

Computationally Efficient Class-Prior Estimation under Class Balance Change Using Energy Distance

Hideko KAWAKUBO[†a)], **Marthinus Christoffel DU PLESSIS**[††b)], *Nonmembers*, *and* **Masashi SUGIYAMA**[††c)], *Member*

SUMMARY In many real-world classification problems, the class balance often changes between training and test datasets, due to sample selection bias or the non-stationarity of the environment. Naive classifier training under such changes of class balance systematically yields a biased solution. It is known that such a systematic bias can be corrected by weighted training according to the test class balance. However, the test class balance is often unknown in practice. In this paper, we consider a semi-supervised learning setup where labeled training samples and unlabeled test samples are available and propose a class balance estimator based on the *energy distance*. Through experiments, we demonstrate that the proposed method is computationally much more efficient than existing approaches, with comparable accuracy.

key words: *class balance change, class-prior estimation, energy distance*

1. Introduction

A fundamental assumption in supervised machine learning is that training and test data follow the same probability distribution. However, in real-world data, this assumption does not necessarily hold due to intrinsic sample selection bias and non-stationarity of the environment [1], and naive training yields a biased solution [2]. In this paper, we consider the situation called the *class balance change* in classification [3], where only the class-prior probabilities change between the training and test phases. In principle, the bias caused by the class balance change can be corrected by weighted training according to the class ratio of the test data. However, in practice, the test class balance is often unknown and thus it needs to be estimated from data.

So far, semi-supervised class balance estimators from labeled training samples and unlabeled test samples have been developed, which are based on fitting a mixture of class-wise training input distributions to the test input distribution. A seminal method [4] adopts the *expectation-maximization* (EM) algorithm [5] to estimate the class ratio. Another earlier paper [3] showed that the EM-based method can be interpreted as *indirectly* fitting a mixture of class-wise training input distributions to the test input distribution under the *Kullback-Leibler* (KL) divergence [6], and the EM-based method was improved by directly estimating the KL divergence without density estimation [7], [8]. Furthermore, to overcome the high sensitivity of the KL divergence to outliers [9], robust variants based on the *Pearson* divergence [10] and the L_2 distance were developed [3], [11].

Another line of research uses the *maximum mean discrepancy* (MMD) [12] for the mixture model fitting, which measures the distance between embeddings of probability distributions in a *reproducing kernel Hilbert space* (RKHS) [13]. A sophisticated implementation was proposed recently that combines MMD with *multiple kernel learning* (MKL) [14].

The divergence-based methods reviewed above [3], [11] are equipped with cross-validation (CV), and therefore all tuning parameters can be objectively optimized. Thanks to this property, the divergence-based methods work very well in practice, although CV is computationally rather expensive. On the other hand, choosing a kernel function in the MMD-based method is not straightforward because changing the kernel function corresponds to changing the error metric and thus CV cannot be employed. Using the median distance of samples as the Gaussian kernel width is a popular heuristic in MMD [12], but this can cause significant performance degradation in practice [15]. Using MKL for MMD is potentially powerful, but this implementation is computationally highly demanding and thus less practical [14].

In this paper, we propose a novel class balance estimator based on *energy distance* [16]. Energy distance may be interpreted as a special case of MMD with a particular kernel function [17], and thus our contribution in this paper can be regarded as providing a practical choice of the kernel function to the MMD-based method. Since the proposed method does not have any tuning parameter, it is extremely simple and computationally highly efficient. Through experiments, we demonstrate the practical usefulness of the proposed method.

2. Problem Formulation

In this section, we formulate the problem of class-prior estimation under semi-supervised leaning setup.

Suppose that we are given a set of training input-output paired samples,

Manuscript received May 30, 2015.
Manuscript revised September 7, 2015.
Manuscript publicized October 6, 2015.

[†]The author is with Tokyo Institute of Technology, Tokyo, 152–8550 Japan.
[††]The authors are with The University of Tokyo, Tokyo, 113–0033 Japan.
a) E-mail: kawakubo@ms.k.u-tokyo.ac.jp
b) E-mail: christo@ms.k.u-tokyo.ac.jp
c) E-mail: sugi@k.u-tokyo.ac.jp

DOI: 10.1587/transinf.2015EDP7212

$$\{(\boldsymbol{x}_i, y_i) \mid \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{1, \ldots, c\}\}_{i=1}^{n},$$

where d denotes the dimensionality of input vector $\boldsymbol{x}_i$ and c denotes the number of classes. The training samples are assumed to be independent and identically distributed to a probability distribution with density $p(\boldsymbol{x}, y)$. Let n_y be the number of training samples in class y, which satisfies $\sum_{y=1}^{c} n_y = n$.

In addition to the training samples, suppose that we are given a set of input-only test samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ which are independent and identically distributed to another probability distribution with density

$$p'(\boldsymbol{x}) = \sum_{y=1}^{c} p'(\boldsymbol{x}, y).$$

Note that test output samples $\{y'_{i'}\}_{i'=1}^{n'}$ for $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are not provided, i.e., we are considering the *semi-supervised learning* setup [18].

We assume that the class-conditional input densities are common between the training and test samples, but the class-prior probabilities are different:

$$p(\boldsymbol{x}|y) = p'(\boldsymbol{x}|y) \quad \text{and} \quad p(y) \neq p'(y).$$

Note that, under this setup, the training and test joint densities $p(\boldsymbol{x}, y)$ and $p'(\boldsymbol{x}, y)$ as well as the training and test input densities $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ are generally different. Our goal is to estimate $p'(y)$ from the labeled training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ and the unlabeled test samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$.

The basic strategy to directly estimate $p'(y)$, proposed in [3], is to fit a mixture of class-wise training input densities,

$$p^{\theta}(\boldsymbol{x}) = \sum_{y=1}^{c} \theta_y p'(\boldsymbol{x}|y) = \sum_{y=1}^{c} \theta_y p(\boldsymbol{x}|y),$$

to a test input density $p'(\boldsymbol{x})$ under some divergence measure. Here $\{\theta_y\}_{y=1}^{c}$ are parameters that satisfy

$$\forall y \;\; 0 \leq \theta_y \leq 1 \quad \text{and} \quad \sum_{y=1}^{c} \theta_y = 1,$$

which correspond to $\{p'(y)\}_{y=1}^{c}$. A naive approach to solving this fitting problem is the two-step procedure of first estimating densities $p(\boldsymbol{x}|y)$ and $p'(\boldsymbol{x})$ from training and test samples and then approximately computing a divergence between $p^{\theta}(\boldsymbol{x})$ and $p'(\boldsymbol{x})$. However, since density estimation is known to be a hard statistical inference problem [19], avoiding density estimation in divergence estimation is more sensible [20]. In the next section, we review existing class-prior estimators that do not involve density estimation under various divergence measures.

3. Existing Class-Prior Estimators

In this section, we review existing class-prior estimators under various divergences.

3.1 Kullback-Leibler Divergence

The *Kullback-Leibler* (KL) divergence [6] is one of the standard divergence measures in statistics and machine learning, and the KL divergence from p^{θ} to p' is defined as follows:

$$\mathrm{KL}(p^{\theta} \| p') = \int p^{\theta}(\boldsymbol{x}) \log \frac{p^{\theta}(\boldsymbol{x})}{p'(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}.$$

The idea of direct KL divergence estimation without density estimation [7], [8] is to directly approximate the density ratio function $\frac{p^{\theta}(\boldsymbol{x})}{p'(\boldsymbol{x})}$ by a model $r_{\boldsymbol{\beta}}(\boldsymbol{x})$, parameterized with $\boldsymbol{\beta}$, by minimizing the generalized KL divergence from p^{θ} to $r_{\boldsymbol{\beta}} p'$:

$$\mathrm{gKL}(p^{\theta} \| r_{\boldsymbol{\beta}} p') = \int p^{\theta}(\boldsymbol{x}) \log \frac{p^{\theta}(\boldsymbol{x})}{r_{\boldsymbol{\beta}}(\boldsymbol{x}) p'(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} - 1 + \int r_{\boldsymbol{\beta}(\boldsymbol{x})} p'(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}.$$

As the density ratio model, let us employ the Gaussian kernel model,

$$r_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{l=0}^{b} \beta_l \psi_l(\boldsymbol{x}), \tag{1}$$

where

$$\psi_0(\boldsymbol{x}) = 1 \quad \text{and} \quad \psi_l(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_l\|^2}{2\sigma^2}\right). \tag{2}$$

Then the regularized empirical optimization problem is given by

$$\max_{\boldsymbol{\beta}} \left[\sum_{y=1}^{c} \frac{\theta_y}{n_y} \sum_{i:y_i=y} \log\left(\sum_{l=0}^{b} \beta_l \psi_l(\boldsymbol{x}_i) \right) - \frac{1}{n'} \sum_{i'=1}^{n'} \sum_{l=0}^{b} \beta_l \psi_l(\boldsymbol{x}'_{i'}) - \lambda \sum_{l=1}^{b} \beta_l^2 \right],$$

where the second term corresponds to the normalization $\int r_{\boldsymbol{\beta}}(\boldsymbol{x}) p'(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1$, the third term is the quadratic regularizer and $\lambda \geq 0$ is the regularization parameter. Note that tuning parameters such as σ and λ can be optimized by cross-validation.

With the solution $\widehat{\boldsymbol{\beta}}$ of the above optimization problem, the KL divergence $\mathrm{KL}(p^{\theta} \| p')$ can be estimated as

$$\widehat{\mathrm{KL}}(p^{\theta} \| p') = \sum_{y=1}^{c} \frac{\theta_y}{n_y} \sum_{i:y_i=y} \log r_{\widehat{\boldsymbol{\beta}}}(\boldsymbol{x}_i).$$

The class-prior $\{\theta_y\}_{y=1}^{c}$ that minimizes the above KL divergence is typically chosen by searching from a set of candidate values [3].

3.2 Pearson Divergence

The *Pearson* (PE) divergence [10] is defined as

$$\mathrm{PE}(p^{\theta}\|p') = \int p'(\boldsymbol{x})\Big(\frac{p^{\theta}(\boldsymbol{x})}{p'(\boldsymbol{x})} - 1\Big)^{2}\mathrm{d}\boldsymbol{x}.$$

An advantage of the PE divergence over the KL divergence is that it does not include the log function, which is highly non-linear around zero, and thus estimation with the PE divergence would be more robust against outliers.

Furthermore, the PE divergence can be directly estimated without density estimation *analytically* [21]. More specifically, the density ratio function $\frac{p^{\theta}(\boldsymbol{x})}{p'(\boldsymbol{x})}$ is modeled in the same way as Eq. (1), and the parameter $\boldsymbol{\beta}$ is learned to minimize the squared error to the true density ratio:

$$\min_{\boldsymbol{\beta}} \int p'(\boldsymbol{x})\Big(\frac{p^{\theta}(\boldsymbol{x})}{p'(\boldsymbol{x})} - r_{\boldsymbol{\beta}}(\boldsymbol{x})\Big)^{2}\mathrm{d}\boldsymbol{x}.$$

With empirical approximation and ℓ_2 regularization, the solution $\widehat{\boldsymbol{\beta}}$ can be obtained analytically as

$$\begin{aligned}\widehat{\boldsymbol{\beta}} &= \min_{\boldsymbol{\beta}}\Big[\boldsymbol{\beta}^{\top}\widehat{\boldsymbol{G}}\boldsymbol{\beta} - 2\boldsymbol{\beta}^{\top}\widehat{\boldsymbol{H}}\boldsymbol{\theta} + \lambda\boldsymbol{\beta}^{\top}\boldsymbol{R}\boldsymbol{\beta}\Big]\\ &= (\widehat{\boldsymbol{G}} + \lambda\boldsymbol{R})^{-1}\widehat{\boldsymbol{H}}\boldsymbol{\theta},\end{aligned}$$

where $\lambda \geq 0$ is the regularization parameter, $\boldsymbol{R}$ is the identity matrix with the first element zero, $\widehat{\boldsymbol{G}}$ and $\widehat{\boldsymbol{H}}$ are defined as

$$\begin{aligned}\widehat{G}_{l,l'} &= \frac{1}{n'}\sum_{i'=1}^{n'}\psi_l(\boldsymbol{x}'_{i'})^{\top}\psi_{l'}(\boldsymbol{x}'_{i'}),\\ \widehat{H}_{l,y} &= \frac{1}{n_y}\sum_{i:y_i=y}\psi_l(\boldsymbol{x}_i),\end{aligned}$$

and $\psi_l(\boldsymbol{x})$ is a basis function defined in Eq. (2). Note that tuning parameters such as σ and λ can be optimized by cross-validation.

With the solution $\widehat{\boldsymbol{\beta}}$, the PE divergence $\mathrm{PE}(p^{\theta}\|p')$ can be estimated as

$$\widehat{\mathrm{PE}}(p^{\boldsymbol{\theta}}\|p') = \boldsymbol{\beta}^{\top}\widehat{\boldsymbol{H}}\boldsymbol{\theta} - \frac{1}{2}\boldsymbol{\beta}^{\top}\widehat{\boldsymbol{G}}\boldsymbol{\beta} - \frac{1}{2}.$$

The class-prior $\{\theta_y\}_{y=1}^{c}$ that minimizes the above PE divergence is typically chosen by searching from a set of candidate values [3].

3.3 L_2 Distance

The KL and PE divergences are members of the f-divergence class [22], [23], containing the density ratio function $\frac{p^{\theta}(\boldsymbol{x})}{p'(\boldsymbol{x})}$. Another class of distance measures is the L_t distance for $t \geq 0$, which contains the density difference function $p^{\theta}(\boldsymbol{x}) - p'(\boldsymbol{x})$:

$$L_t(p^{\theta}, p') = \Big(\int\big|p^{\theta}(\boldsymbol{x}) - p'(\boldsymbol{x})\big|^{t}\mathrm{d}\boldsymbol{x}\Big)^{\frac{1}{t}}.$$

Although density ratio function $\frac{p^{\theta}(\boldsymbol{x})}{p'(\boldsymbol{x})}$ can be unbounded (e.g., the ratio of Gaussian densities with the same variance and different means), density difference function $p^{\theta}(\boldsymbol{x}) - p'(\boldsymbol{x})$ is always bounded as long as $p^{\theta}(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ are both bounded. Thus, divergence measures based on the density difference are expected to be more stable. Note that f-divergences are invariant under transformation of $\boldsymbol{x}$, while the L_t distance is symmetric, i.e., $L_t(p^{\theta}, p') = L_t(p', p^{\theta})$.

The idea of direct L_2 distance estimation without density estimation [11] is to directly approximate the density difference function $p^{\theta}(\boldsymbol{x}) - p'(\boldsymbol{x})$ by a model $f_{\boldsymbol{\alpha}}(\boldsymbol{x})$ with parameter $\boldsymbol{\alpha}$ estimated by minimizing the squared error:

$$\int\Big(f_{\boldsymbol{\alpha}}(\boldsymbol{x}) - (p^{\theta}(\boldsymbol{x}) - p'(\boldsymbol{x}))\Big)^{2}\mathrm{d}\boldsymbol{x}.$$

Let us employ a Gaussian kernel density difference model,

$$f_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{l=1}^{n+n'}\alpha_l\psi_l(\boldsymbol{x}),$$

where

$$\psi_l = \exp\Big(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_l\|^{2}}{2\sigma^{2}}\Big).$$

$\boldsymbol{c}_l$ denotes $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{n+n'}) = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}'_1, \ldots \boldsymbol{x}'_{n'})$. With empirical approximation and ℓ_2 regularization, the solution $\widehat{\boldsymbol{\alpha}}$ can be obtained analytically as

$$\begin{aligned}\widehat{\boldsymbol{\alpha}} &= \min_{\boldsymbol{\alpha}}\Big[\boldsymbol{\alpha}^{\top}\boldsymbol{U}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^{\top}\widehat{\boldsymbol{v}} + \lambda\boldsymbol{\alpha}^{\top}\boldsymbol{\alpha}\Big]\\ &= (\boldsymbol{U} + \lambda\boldsymbol{I})^{-1}\widehat{\boldsymbol{v}},\end{aligned}$$

where, for d being the dimensionality of $\boldsymbol{x}$,

$$\begin{aligned}U_{l,l'} &= (\pi\sigma^{2})^{\frac{d}{2}}\exp\Big(-\frac{\|\boldsymbol{c}_l - \boldsymbol{c}_{l'}\|^{2}}{4\sigma^{2}}\Big),\\ \widehat{v}_l &= \frac{1}{n'}\sum_{i'=1}^{n'}\psi_l(\boldsymbol{x}'_{i'}) - \sum_{y=1}^{c}\frac{\theta_y}{n_y}\sum_{i:y_i=y}\psi_l(\boldsymbol{x}_i),\end{aligned}$$

$\lambda \geq 0$ is the regularization parameter and $\boldsymbol{I}$ is the identity matrix. Note that tuning parameters such as σ and λ can be optimized by cross-validation.

With the solution $\widehat{\boldsymbol{\alpha}}$, the L_2 distance $L_2(p^{\theta}, p')$ can be estimated as

$$\widehat{L_2}(p^{\theta}, p') = 2\widehat{\boldsymbol{v}}^{\top}\widehat{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}^{\top}\boldsymbol{U}\widehat{\boldsymbol{\alpha}}.$$

The class-prior $\{\theta_y\}_{y=1}^{c}$ that minimizes the L_2 distance is typically chosen by searching from a set of candidate values [11].

3.4 Maximum Mean Discrepancy

Maximum mean discrepancy (MMD) [24] measures the distance between embeddings of probability distributions in a *reproducing kernel Hilbert space* (RKHS) [13].

Let $\boldsymbol{x}$, $\check{\boldsymbol{x}}$ and $\boldsymbol{x}'$, $\check{\boldsymbol{x}}'$ be samples drawn from the probability distributions with densities p^{θ} and p', respectively, and ψ be a *characteristic kernel* [24] such as the Gaussian kernel. Then MMD is defined

$$\begin{aligned}\mathrm{MMD}(p^\theta, p') = {} & \mathbb{E}_{\boldsymbol{x},\check{\boldsymbol{x}}\sim p^\theta}[\psi(\boldsymbol{x},\check{\boldsymbol{x}})] + \mathbb{E}_{\boldsymbol{x}',\check{\boldsymbol{x}}'\sim p'}[\psi(\boldsymbol{x}',\check{\boldsymbol{x}}')] \\ & - 2\mathbb{E}_{\boldsymbol{x}\sim p^\theta,\check{\boldsymbol{x}}'\sim p'}[\psi(\boldsymbol{x},\check{\boldsymbol{x}}')],\end{aligned}$$

where $\mathbb{E}$ denotes the expectation. For any characteristic kernel ψ, $\mathrm{MMD}(p^\theta, p') = 0$ if and only if $p^\theta = p'$.

An advantage of MMD is that it can be immediately estimated from samples. However a practical difficulty of using MMD is that the performance depends on the choice of kernel function ψ, and cross-validation *cannot* be used, because changing the kernel function corresponds to changing the error metric. A popular heuristic in MMD is to use the median distance of samples as the Gaussian kernel width [12], although this does not always work well [15]. Recently, an MMD-based class-prior estimator was proposed, which learns the kernel function by multiple kernel learning [14]. Although this was shown to work well, it is computationally very expensive.

4. Proposed Method

As shown above, the MMD-based class-prior estimator with a single kernel can be computationally more efficient than the methods based on the KL divergence, the PE divergence, and the L_2 distance because no cross-validation is included. However, in practice, the choice of kernel functions is not straightforward. In this section, we introduce another distance measure called the *energy distance* [16], and propose to use it in class-prior estimation.

4.1 Energy Distance

The energy distance is defined as the weighted L_2 distance between characteristic functions, which is the inverse Fourier transform of the density function.

More specifically, the energy distance between p^θ and p' is defined as follows:

$$\begin{aligned}&\mathrm{ED}(p^\theta, p') \\ &= \int_{\mathbb{R}^d} \|\boldsymbol{\phi}_{p^\theta}(\boldsymbol{t}) - \boldsymbol{\phi}_{p'}(\boldsymbol{t})\|^2 \left(\frac{\pi^{\frac{d+1}{2}}}{\Gamma(\frac{d+1}{2})}\|\boldsymbol{t}\|^{d+1}\right)^{-1} \mathrm{d}\boldsymbol{t},\end{aligned}$$

where $\boldsymbol{\phi}_p$ denotes the characteristic function of p, $\|\cdot\|$ denotes the Euclidean distance, and $\Gamma(\cdot)$ is the *gamma function*. The energy distance has the following properties:

- $\mathrm{ED}(p^\theta, p') = \mathrm{ED}(p', p^\theta)$,
- $\mathrm{ED}(p^\theta, p') \geq 0$,
- $\mathrm{ED}(p^\theta, p') = 0$ if and only if $p^\theta = p'$.

An important property of the energy distance is that $\mathrm{ED}(p^\theta, p')$ can be equivalently expressed as

$$\begin{aligned}\mathrm{ED}(p^\theta, p') = {} & 2\mathbb{E}_{\boldsymbol{x}\sim p^\theta,\check{\boldsymbol{x}}'\sim p'}\|\boldsymbol{x}-\check{\boldsymbol{x}}'\| - \mathbb{E}_{\boldsymbol{x},\check{\boldsymbol{x}}\sim p^\theta}\|\boldsymbol{x}-\check{\boldsymbol{x}}\| \\ & - \mathbb{E}_{\boldsymbol{x}',\check{\boldsymbol{x}}'\sim p'}\|\boldsymbol{x}'-\check{\boldsymbol{x}}'\|, \qquad (3)\end{aligned}$$

under the mild assumptions that $\mathbb{E}_{\boldsymbol{x}\sim p^\theta}\|\boldsymbol{x}\| < \infty$ and $\mathbb{E}_{\boldsymbol{x}'\sim p'}\|\boldsymbol{x}'\| < \infty$. Equation (3) allows us to immediately obtain a sample approximation to the energy distance in the same way as MMD. However, unlike MMD, there is no tuning parameter such as the Gaussian kernel width. Below, we propose to use the energy distance in class-prior estimation, which we will demonstrate to be practically useful in the next section.

Actually, the energy distance was shown to be a special case of MMD [17], meaning that MMD with a certain choice of kernels is reduced to the energy distance. Therefore, our contribution in this paper can be regarded as providing a practical choice of the kernel function in the MMD-based method. The resulting proposed method does not contain any tuning parameter, and thus it is extremely simple and computationally highly efficient.

4.2 Class-Prior Estimation under Energy Distance

Here, we describe the procedure of class-prior estimation based on the energy distance, which minimizes an empirical approximation of $\mathrm{ED}(p^\theta, p')$ with respect to $\boldsymbol{\theta}$.

4.2.1 Convexity of $\mathrm{ED}(p^\theta, p')$ as a Function of $\boldsymbol{\theta}$

$\mathrm{ED}(p^\theta, p')$ given by Eq. (3) can be more specifically expressed as

$$\begin{aligned}\mathrm{ED}(p^\theta, p') = {} & 2\sum_{y=1}^{c}\theta_y\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y),\check{\boldsymbol{x}}'\sim p'}\|\boldsymbol{x}-\check{\boldsymbol{x}}'\| \\ & - \sum_{y,y'=1}^{c}\theta_y\theta_{y'}\mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y),\check{\boldsymbol{x}}\sim p(\boldsymbol{x}|y')}\|\boldsymbol{x}-\check{\boldsymbol{x}}\| \\ & - \mathbb{E}_{\boldsymbol{x}',\check{\boldsymbol{x}}'\sim p'}\|\boldsymbol{x}'-\check{\boldsymbol{x}}'\|. \qquad (4)\end{aligned}$$

Equation (4) can be compactly expressed as a function of $\boldsymbol{\theta}$ by

$$J(\boldsymbol{\theta}) = -\boldsymbol{\theta}^\top\boldsymbol{A}\boldsymbol{\theta} + 2\boldsymbol{\theta}^\top\boldsymbol{s},$$

where $\boldsymbol{A}$ is the $c\times c$ symmetric matrix and $\boldsymbol{s}$ is the c-dimensional vector defined as

$$\begin{aligned}A_{y,y'} &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y),\check{\boldsymbol{x}}\sim p(\boldsymbol{x}|y')}\|\boldsymbol{x}-\check{\boldsymbol{x}}\|, \\ s_y &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y),\boldsymbol{x}'\sim p'}\|\boldsymbol{x}-\boldsymbol{x}'\|.\end{aligned}$$

To solve $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$, let us begin with rewriting $J(\boldsymbol{\theta})$ using $\dot{\boldsymbol{\theta}} = (\theta_1,\ldots,\theta_{c-1})^\top$ and $\theta_c = 1-\sum_{y=1}^{c-1}\dot{\theta}_y$ as follows:

$$\dot{J}(\dot{\boldsymbol{\theta}}) = \dot{\boldsymbol{\theta}}^\top\boldsymbol{B}\dot{\boldsymbol{\theta}} - 2\dot{\boldsymbol{\theta}}^\top\boldsymbol{t} + C, \qquad (5)$$

where C is a constant, $\boldsymbol{B}$ is the $(c-1)\times(c-1)$ symmetric matrix and $\boldsymbol{t}$ is the $(c-1)$-dimensional vector defined as

$$\begin{aligned}B_{y,y'} &= -A_{y,y'} + A_{y,c} + A_{c,y'} - A_{c,c} \qquad (6) \\ t_y &= -s_y + A_{y,c} + s_c - A_{c,c}.\end{aligned}$$

For the function $\dot{J}(\dot{\boldsymbol{\theta}})$, we have the following theorem.

Theorem 1: $\dot{J}(\dot{\boldsymbol{\theta}})$ defined by Eq. (5) is convex with respect to $\dot{\boldsymbol{\theta}}$.

The proof of Theorem 1 is given in Appendix A.

Especially in the binary case where c = 2, $\boldsymbol{B}$ is not a matrix but a scalar given as

$$\begin{aligned}\boldsymbol{B} &= -A_{1,1} + 2A_{1,2} - A_{2,2}\\ &= \mathrm{ED}(p(\boldsymbol{x}|y=1), p(\boldsymbol{x}|y=2)) > 0. \qquad (7)\end{aligned}$$

Thus, $\dot{J}(\dot{\boldsymbol{\theta}})$ is strongly convex when $c = 2$.

On the other hand, for the strong convexity in general multi-class cases where $c > 2$, let us express $\boldsymbol{B}$ defined by Eq. (6) as the following block matrix:

$$\boldsymbol{B} = \boldsymbol{B}_{c-1} = \begin{bmatrix}\boldsymbol{B}_{c-2} & \boldsymbol{b}_{c-2}\\ \boldsymbol{b}_{c-2}^{\top} & B_{c-1,c-1}\end{bmatrix},$$

where $\boldsymbol{B}_{c-2}$ denotes the $(c-2)$-th leading principal minor of $\boldsymbol{B}_{c-1}$ and $\boldsymbol{b}_{c-2} = [B_{1,c-1}, \ldots, B_{c-2,c-1}]^{\top}$. Then we have the following theorem.

Theorem 2: In the multi-class classification cases where $c > 2$, $\dot{J}(\dot{\boldsymbol{\theta}})$ is strongly convex, if and only if the following conditions are satisfied.

$$\begin{cases} B_1 > 0,\\ B_{y,y} - \boldsymbol{b}_{y-1}^{\top}\boldsymbol{B}_{y-1}^{-1}\boldsymbol{b}_{y-1} > 0 \quad (y = 2, \ldots, c-1).\end{cases}$$

A proof of Theorem 2 is given in Appendix B.

Below, we will explain the intuition of the conditions in Theorem 2, in case of $c = 3$. $B_1 > 0$ is derived in the same way as Eq. (7). $\boldsymbol{B}_2$ is defined as

$$\begin{aligned}\boldsymbol{B}_2 &= \begin{bmatrix} B_1 & b_1\\ b_1^{\top} & B_{2,2}\end{bmatrix}\\ &= \frac{1}{2}\begin{bmatrix} 2d_{13} & d_{13} + d_{32} - d_{12}\\ d_{23} + d_{31} - d_{21} & 2d_{23}\end{bmatrix},\end{aligned}$$

where $d_{ij} = \mathrm{ED}(p(\boldsymbol{x}|y=i), p(\boldsymbol{x}|y=j))$. Let us consider what the following condition indicates.

$$\begin{aligned}&B_{2,2} - b_1^{\top}B_1^{-1}b_1\\ &= \frac{1}{4d_{13}}\{4d_{13}d_{32} - (d_{13} + d_{32} - d_{12})^2\}\\ &= \frac{1}{4d_{13}}(2\sqrt{d_{13}d_{32}} + d_{13} + d_{32} - d_{12})\times\\ &\qquad\qquad (2\sqrt{d_{13}d_{32}} - d_{13} - d_{32} + d_{12})\\ &= \frac{1}{4d_{13}}\{(\sqrt{d_{13}} + \sqrt{d_{32}})^2 - d_{12}\}\times\\ &\qquad\qquad \{d_{12} - (\sqrt{d_{13}} - \sqrt{d_{32}})^2\} > 0. \qquad (8)\end{aligned}$$

Equation (8) is equivalent to

$$(\sqrt{d_{13}} - \sqrt{d_{32}})^2 < d_{12} < (\sqrt{d_{13}} + \sqrt{d_{32}})^2,$$

which is equivalent to

$$\left|\sqrt{d_{13}} - \sqrt{d_{32}}\right| < \sqrt{d_{12}} < \sqrt{d_{13}} + \sqrt{d_{32}}.$$

This is satisfied if and only if the following three conditions hold:

$$\begin{cases}\sqrt{d_{12}} + \sqrt{d_{23}} > \sqrt{d_{13}},\\ \sqrt{d_{21}} + \sqrt{d_{13}} > \sqrt{d_{23}},\\ \sqrt{d_{13}} + \sqrt{d_{32}} > \sqrt{d_{12}}.\end{cases}$$

Therefore, the condition $B_{2,2} - b_1^{\top}B_1^{-1}b_1 > 0$ is equivalent to the triangle inequalities which consist of the square root of the energy distance.

4.2.2 Class-prior Estimation by Solving an Optimization Problem

In the binary case where $c = 2$, $\dot{J}(\dot{\boldsymbol{\theta}})$ is strongly convex as shown in 4.2.1. The optimal solution $\boldsymbol{\theta}^*$ can be obtained analytically by

$$\begin{aligned}\theta_1^* &= \begin{cases}\widetilde{\theta}_1 & \text{if } \widetilde{\theta}_1 \in [0,1],\\ 0 & \text{if } \widetilde{\theta}_1 < 0,\\ 1 & \text{if } \widetilde{\theta}_1 > 1,\end{cases}\\ \theta_2^* &= 1 - \theta_1^*,\end{aligned}$$

where

$$\widetilde{\theta}_1 = \frac{-s_1 + A_{1,2} + s_2 - A_{2,2}}{-A_{1,1} + 2A_{1,2} - A_{2,2}}.$$

In practice, we approximate $\boldsymbol{A}$ and $\boldsymbol{s}$ from samples as

$$\begin{aligned}\widehat{A}_{y,y'} &= \frac{1}{n_y n_{y'}}\sum_{i:y_i=y}\sum_{j:y_j=y'}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|,\\ \widehat{s}_y &= \frac{1}{n_y n'}\sum_{i:y_i=y}\sum_{i'=1}^{n'}\|\boldsymbol{x}_i - \boldsymbol{x}'_{i'}\|.\end{aligned}$$

Then $\widetilde{\theta}_1$ can be approximately computed as

$$\widetilde{\theta}_1 \approx \frac{-\widehat{s}_1 + \widehat{A}_{1,2} + \widehat{s}_2 - \widehat{A}_{2,2}}{-\widehat{A}_{1,1} + 2\widehat{A}_{1,2} - \widehat{A}_{2,2}}.$$

In multi-class cases where $c > 2$, the optimal solution $\boldsymbol{\theta}^*$ may be obtained by solving the following quadratic programming problem:

$$\min_{\dot{\boldsymbol{\theta}}} \dot{J}(\dot{\boldsymbol{\theta}}) \quad \text{subject to} \quad \forall y \;\; \dot{\theta}_y \geq 0, \;\; \sum_{y=1}^{c-1}\dot{\theta}_y \leq 1.$$

An empirical approximation to $\mathrm{ED}(p^{\theta}, p')$ given by Eq. (4) can be expressed as

$$\begin{aligned}\widehat{\mathrm{ED}}(p^{\theta}, p') = 2\sum_{y=1}^{c}\frac{\theta_y}{n_y n'}\sum_{i:y_i=y}\sum_{i'=1}^{n'}\|\boldsymbol{x}_i - \boldsymbol{x}'_{i'}\|\\ - \sum_{y,y'=1}^{c}\frac{\theta_y\theta_{y'}}{n_y n_{y'}}\sum_{i:y_i=y}\sum_{j:y_j=y'}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\end{aligned}$$

$$-\frac{1}{n'^2}\sum_{i'=1}^{n'}\sum_{j'=1}^{n'}\|\boldsymbol{x}'_{i'}-\boldsymbol{x}'_{j'}\|.$$

Then the empirical solution $\widehat{\boldsymbol{\theta}}$ can be obtained by solving the following quadratic programming problem:

$$\min_{\dot{\boldsymbol{\theta}}} \dot{\boldsymbol{\theta}}^\top \widehat{\boldsymbol{B}}\dot{\boldsymbol{\theta}} - 2\dot{\boldsymbol{\theta}}^\top \widehat{\boldsymbol{t}} \quad \text{subject to } \forall y \;\; \dot{\theta}_y \geq 0, \; \sum_{y=1}^{c-1}\dot{\theta}_y \leq 1,$$

where $\widehat{\boldsymbol{B}}$ and $\widehat{\boldsymbol{t}}$ are defined as

$$\begin{aligned}\widehat{B}_{y,y'} &= -\widehat{A}_{y,y'} + \widehat{A}_{y,c} + \widehat{A}_{c,y'} - \widehat{A}_{c,c},\\ \widehat{t}_y &= -\widehat{s}_y + \widehat{A}_{y,c} + \widehat{s}_c - \widehat{A}_{c,c}.\end{aligned}$$

If $\widehat{\dot{\boldsymbol{\theta}}}$ satisfies $\widehat{\dot{\theta}}_y \geq 0$ for all y and $\sum_{y=1}^{c-1}\widehat{\dot{\theta}}_y \leq 1$, then we can simply obtain the solution by

$$\widehat{\dot{\boldsymbol{\theta}}} = \widehat{\boldsymbol{B}}^{-1}\widehat{\boldsymbol{t}}, \quad \widehat{\theta}_c = 1 - \sum_{y=1}^{c-1}\widehat{\dot{\theta}}_y.$$

5. Experiments

In this section, we report experimental results. We compared the performance of the proposed method, denoted by ED, with the following four methods.

PE-DR[†]: The density-ratio method using the PE-divergence estimator [3].

LSDD[††]: The density-difference method using the L_2 distance estimator [11].

MMD: The MMD-based method with the single Gaussian kernel [14], where the median distance of samples is used as the Gaussian kernel width.

MMD-MKL[†††]: The MMD-based method with multiple kernel learning (MKL) [14].

5.1 Binary Cases

First, we conducted experiments with binary classification data. Table 1 shows the list of datasets we used, containing the input dimensionality d, the number of training samples n and the number of test samples n'. The class ratio of training samples was fixed at 1 : 1, while the class ratios of test samples were set according to the selected true class-priors $\theta^* \in \{0.1, 0.2, \ldots, 0.9\}$.

Gauss1, Gauss2 and Gauss3 are artificial datasets. Samples in class 1 of Gauss1, Gauss2 and Gauss3 follow $N(0,1)$, the normal distribution with mean 0 and variance 1. While samples in class 2 of Gauss1, Gauss2 and Gauss3 follow $N(1,1)$, $N(2,1)$ and $N(3,1)$ respectively. Other datasets in Table 1 are benchmark datasets. For each dataset, all methods were run 100 times with random selection of data samples from the original datasets.

Table 1 Specification of binary datasets. ♡ indicates artificial data. ♠, ♢ and ♣ indicate datasets taken from Machine Learning Data Set Repository[††††], LIBSVM Data[†††††] and The Elements of Statistical Learning[††††††], respectively.

	Dataset	d	n	n'	♯ Class 1	♯ Class 2
♡	Gauss1	1	200	200	5,000	5,000
♡	Gauss2	1	200	200	5,000	5,000
♡	Gauss3	1	200	200	5,000	5,000
♠	Banana	2	200	200	2,924	2,376
♠	Image	18	200	200	898	1,188
♠	Waveform	21	200	200	3,353	1,647
♢	Breast cancer	10	200	100	444	239
♢	SVMguide1	4	200	200	3,089	4,000
♣	SAheart	9	100	100	302	160

The average and standard deviation of the squared error between estimated and true class-priors are shown in Fig. 1. This shows that ED works well as a whole and has a stable performance over a wide range of datasets. MMD uses a fixed parameter which coincidentally worked well with the Banana dataset, but failed for other datasets. This is in contrast to the proposed method which scored well with all datasets.

Table 2 summarizes the computation time of all methods, showing that the proposed method ED is faster than other methods in orders of magnitude. MMD is also relatively fast, but compared with ED, the computation of Gaussian kernels themselves is expensive. PE-DR and LSDD are slow due to cross-validation, and MKL requires a huge amount of computation time for learning kernel combinations.

The summary of the experiments for binary cases is as follows. In most cases ED is on par with other methods, though the estimation error of ED is not always lower than that of LSDD and MMD-MKL, such as in the case of SVMguide1 and SAheart. However, the performance of ED is always close to that of LSDD and MMD-MKL in spite of its shortest computation time. Thus we can say that the performances of those methods depend on the datasets while ED is always much faster than LSDD and MMD-MKL.

Next, we trained a Gaussian-kernel support vector machine (SVM) [19] with instance weights based on the estimated class-priors:

$$\begin{aligned}\min_{\boldsymbol{w},\delta}\Big[\widehat{\theta}_1 &\sum_{i:y_i=1}\max(0, 1-(\boldsymbol{w}^\top\psi(\boldsymbol{x}_i)+\delta))\\ &+(1-\widehat{\theta}_1)\sum_{i:y_i=2}\max(0, 1+(\boldsymbol{w}^\top\psi(\boldsymbol{x}_i)+\delta)) + \lambda\|\boldsymbol{w}\|^2\Big],\end{aligned}$$

where $\widehat{\theta}_1$ is an estimated class-prior for class 1, $\boldsymbol{w}$ denotes a coefficient vector, $\psi(\boldsymbol{x})$ denotes a feature vector in the Gaussian reproducing kernel Hilbert space and δ is a bias term. We solved this optimization problem in the dual using LIBSVM [25]. All the hyper-parameters (regularization parameter λ and the Gaussian kernel width σ) were selected

[†] We used the code available from http://www.ms.k.u-tokyo.ac.jp/~christo/pages/classprior-pearson-page.html.

[††] We used the code available from http://www.ms.k.u-tokyo.ac.jp/~christo/pages/classprior-L2-page.html.

[†††] We used the code personally provided by the authors. As a quadratic program solver, we used "Gurobi" instead of "quadprog" in the original code.

[††††] http://mldata.org/repository/data/.

[†††††] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

[††††††] http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/.

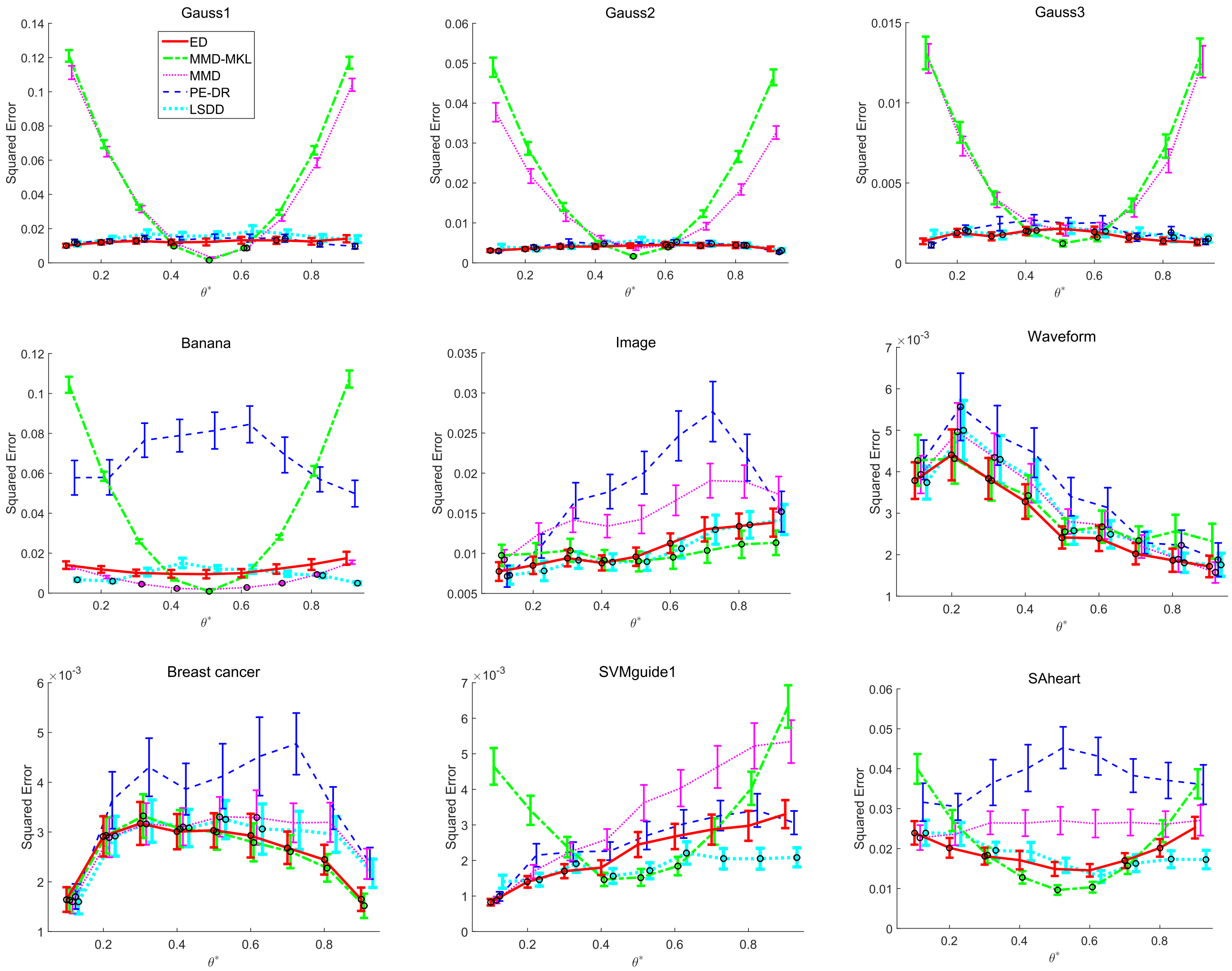

Fig. 1 The average and standard deviation of the squared error between estimated and true class-priors. We applied t-test at significance level 5%, and the best method and methods that do not have significant difference from the best one are marked with '○'.

Table 2 Computation time (sec). $a\{b\} \overset{\text{def}}{=} a \times 10^b$. We used Intel Xeon E5 – 2667 CPU, equipped with 64 GB of memory. B-cancer★ and SVMg1★★ indicate Breast cancer and SVMguide1.

Dataset	ED	MMD -MKL	MMD	PE-DR	LSDD
Gauss1	**3.7**{−3}	7.0{2}	4.9{−2}	1.2{2}	1.7{2}
Gauss2	**3.7**{−3}	8.2{2}	4.9{−2}	9.2{1}	1.7{2}
Gauss3	**6.0**{−3}	5.9{2}	4.9{−2}	9.1{1}	1.7{2}
Banana	**3.5**{−3}	7.2{2}	4.9{−2}	8.3{1}	1.7{2}
Image	**4.4**{−3}	1.2{3}	4.7{−2}	9.6{1}	4.5{1}
Waveform	**4.5**{−3}	1.2{3}	4.9{−2}	1.2{2}	4.5{1}
B-cancer★	**4.0**{−3}	3.7{3}	3.7{−2}	1.0{2}	1.8{2}
SVMg1★★	**4.0**{−3}	8.0{2}	4.7{−2}	9.4{1}	4.4{1}
SAheart	**3.3**{−3}	1.4{2}	4.7{−2}	9.4{1}	4.4{1}
Average	**4.1**{−3}	8.0{2}	4.7{−2}	1.0{2}	1.2{2}

via 5-fold *weighted* cross-validation [26] in terms of the 0/1 loss.

The average of the misclassification rate are shown in Fig. 2, showing that all the weighted methods tend to outperform the non-weighted counterparts and are comparable overall.

In case of Image, SVMguide1 and SAheart, there are low correlations between the squared error of class-prior estimation and the misclassification rate. This may be attributed to the fact that $p(\boldsymbol{x}|y = 1)$ and $p(\boldsymbol{x}|y = 2)$ are somewhat distant. If the two densities have almost no overlap, small differences in the estimated class-prior estimation does not strongly affect the misclassification rate. For example, since the densities of class 1 and class 2 in Gauss3 are more distant than those in Gauss1, the squared error and the misclassification rate of Gauss3 are less correlated than those of Gauss1.

5.2 Multi-Class Cases

Next, we applied class-prior estimation to multi-class classification. Since the compared methods, with the exception of MMD, is prohibitively slow for a large number of classes, we used only three-class datasets for multi-class experiments. Table 3 shows the list of datasets we used.

As training data, n samples were drawn from each of

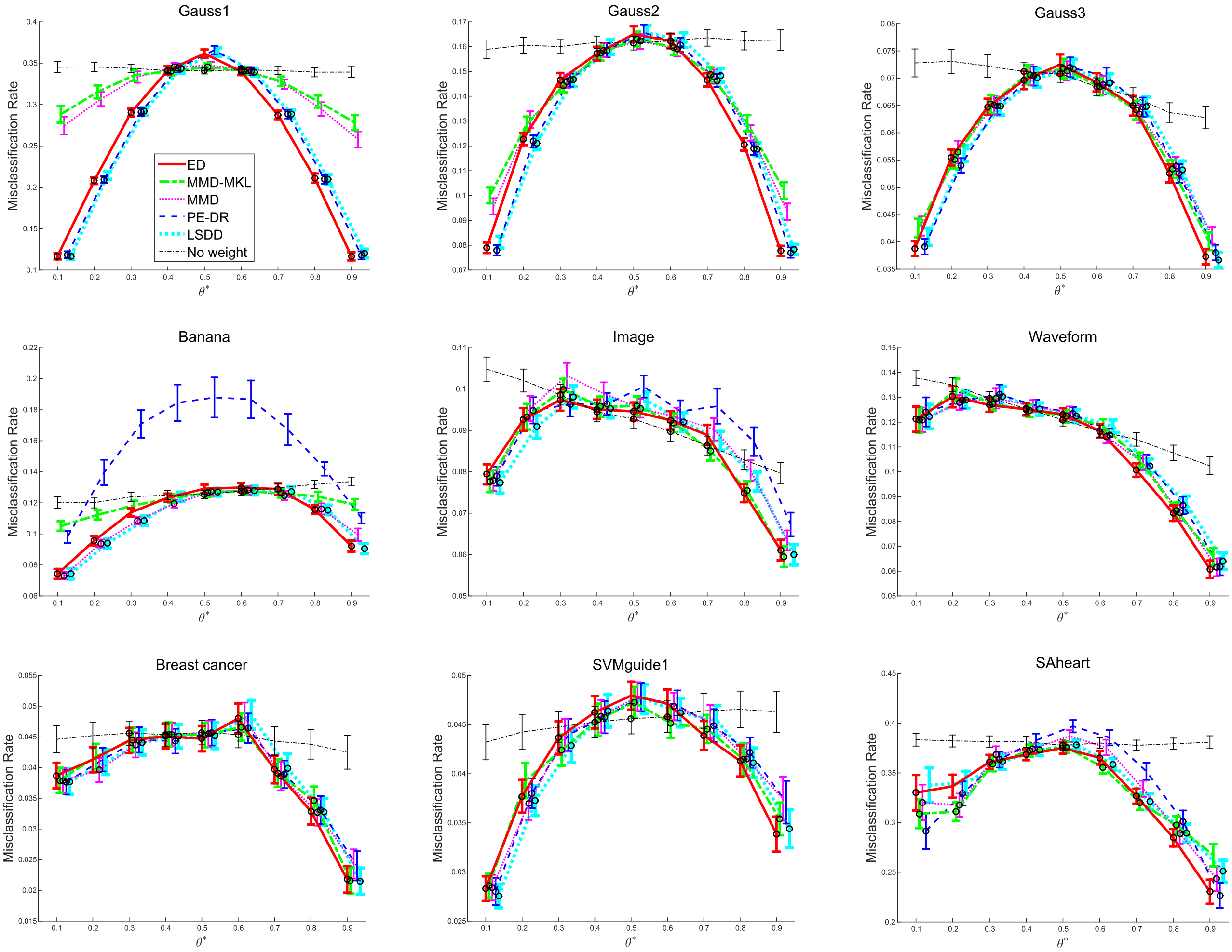

Fig. 2 The average and standard deviation of the misclassification rates. SVM was used for the classification. We applied t-test at significance level 5%, and the best method and significant methods that do not have significant difference from the best one are marked with '∘'.

Table 3 Specification of three-class datasets. All the datasets are taken from LIBSVM Data.

Dataset	d	♯ Class 1	♯ Class 2	♯ Class 3
Combined	100	39,455	18,300	21,068
DNA	180	1,051	464	485
SVMguide2	20	221	53	117

the classes (i.e. the class ratio of training samples was fixed at 1 : 1 : 1), while as the test data, 100 samples were drawn following the probabilities 0.6, 0.1 and 0.3 from each of the classes.

We computed the L_2 distance between the estimated and true class-priors, and trained a L_2 regularized kernel logistic regression [27] with instance weights based on the estimated class-priors:

$$\min_{\boldsymbol{w}}\Big[\sum_{y=1}^{c}\widehat{\theta}_y\sum_{i:y_i=y}g(\boldsymbol{x_i},y_i,\boldsymbol{w}^{(y)})+\lambda\|\boldsymbol{w}^{(y)}\|^2\Big],$$

where $\widehat{\theta}_y$ is an estimated class-prior for class y. $g(\boldsymbol{x_i},y_i,\boldsymbol{w}^{(y)})$ indicates the logistic loss function defined as

$$g(\boldsymbol{x_i},y_i,\boldsymbol{w}^{(y)})=\log\frac{\exp(y_i\sum_{j=1}^{n}w_j^{(y)}\psi_j(\boldsymbol{x}_i))}{\sum_{y'=1}^{c}\exp(y_i\sum_{j=1}^{n}w_j^{(y')}\psi_j(\boldsymbol{x}_i))},$$

where $\boldsymbol{w}=(w_1^{(1)},\ldots,w_n^{(1)},\ldots,w_1^{(c)},\ldots,w_n^{(c)})^\top$. All the hyper-parameters (regularization parameter λ and the Gaussian kernel width σ) were selected via 5-fold *weighted* cross-validation [26] in terms of the logistic loss.

Since MKL is prohibitively slow, it was only run 50 times on the datasets. All other methods were run 100 times. Figure 3 indicates that the performance of all methods roughly improves as the number of training samples increases, and ED works stably. ED is not the best method on every datasets, however it is not much worse than the best performing algorithm.

Table 4 shows the computation time of each dataset with the largest number of labeled samples. ED is much faster than the other methods. MMD is also fast, however it is not stable especially in the case of DNA. This is an example which indicates that MMD with a fixed parameter does not work well.

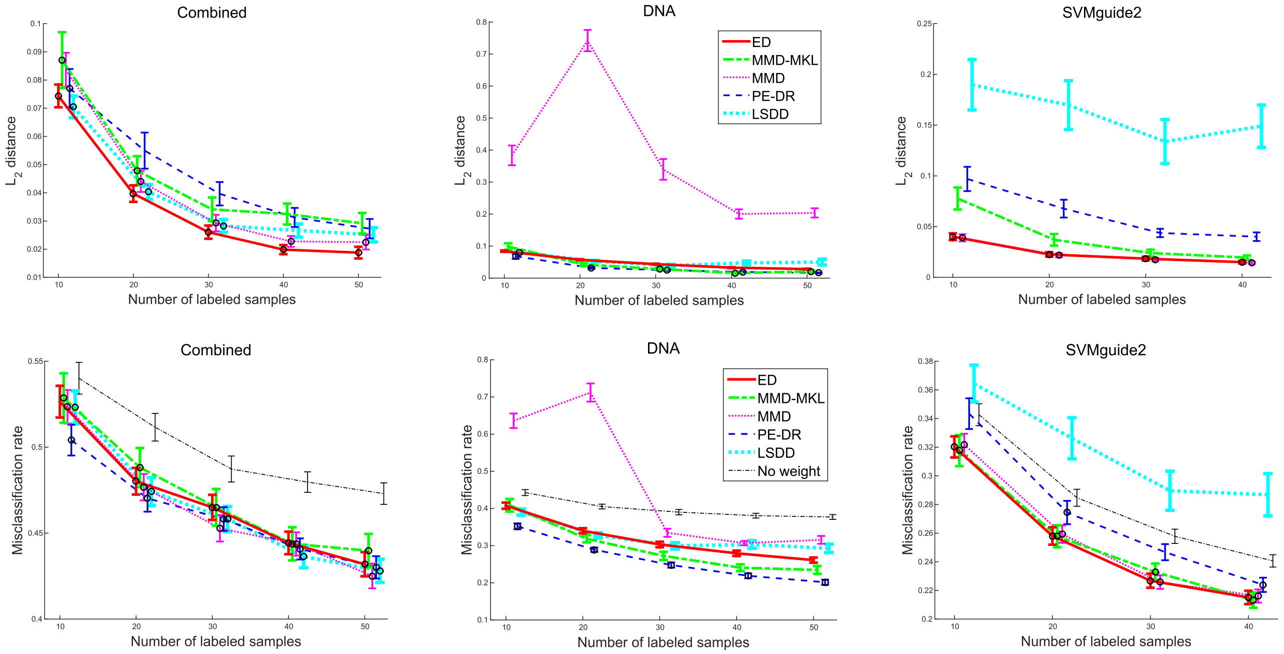

Fig. 3 The upper row: The average and standard deviation of the squared error between estimated and true class-priors. The lower row: The average and standard deviation of the misclassification rates. We applied t-test at significance level 5%, and the best method and methods that do not have significant difference from the best one are marked with ‘∘’.

Table 4 Computation time (sec). $a\{b\} \overset{\text{def}}{=} a \times 10^{b}$. We used Intel Xeon E5 − 2667 CPU, equipped with 64 GB of memory. SVMg2⋆⋆⋆ indicates SVMguide2.

Dataset	ED	MMD -MKL	MMD	PE-DR	LSDD
Combined	**1.3**{−2}	5.3{3}	7.1{−2}	6.7{1}	2.7{3}
DNA	**1.4**{−2}	7.1{3}	8.1{−2}	6.8{1}	2.7{3}
SVMg2⋆⋆⋆	**5.1**{−3}	4.7{3}	2.9{−2}	2.3{1}	8.9{2}
Average	**1.1**{−2}	5.7{3}	6.0{−2}	5.3{1}	2.1{3}

Through both binary and multi-class experiments, we conclude that the proposed method can be a computationally efficient alternative to the existing class-prior estimation methods.

6. Conclusion

In this paper, we proposed a simple and computationally efficient class-prior estimator based on the energy distance, and proved the convexity of the optimization problem of the proposed method. We conducted experiments for both binary and multi-class cases, and the results showed that the proposed method worked well and is stable over a wide range of datasets. Furthermore, the computation time of the proposed method was much faster than the compared method significantly.

Acknowledgements

MCdP was supported by the JST CREST project, and MS was supported by KAKENHI 25700022.

References

[1] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, eds., Dataset Shift in Machine Learning, MIT Press, Cambridge, Massachusetts, USA, 2009.

[2] M. Sugiyama and M. Kawanabe, Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation, MIT Press, Cambridge, Massachusetts, USA, 2012.

[3] M.C. du Plessis and M. Sugiyama, “Semi-supervised learning of class balance under class-prior change by distribution matching,” Neural Networks, vol.50, pp.110–119, 2014.

[4] M. Saerens, P. Latinne, and C. Decaestecker, “Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure,” Neural Computation, vol.14, no.1, pp.21–41, 2002.

[5] A.P. Dempster, N.M. Laird, and D.B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” J. Royal Statistical Society, Series B, vol.39, no.1, pp.1–38, 1977.

[6] S. Kullback and R.A. Leibler, “On information and sufficiency,” The Annals of Mathematical Statistics, vol.22, no.1, pp.79–86, 1951.

[7] 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, vol.60, no.4, pp.699–746, 2008.

[8] X. Nguyen, M.J. Wainwright, and M.I. Jordan, “Estimating divergence functionals and the likelihood ratio by convex risk minimization,” IEEE Transactions on Information Theory, vol.56, no.11, pp.5847–5861, 2010.

[9] A. Basu, I.R. Harris, N.L. Hjort, and M.C. Jones, “Robust and efficient estimation by minimising a density power divergence,” Biometrika, vol.85, no.3, pp.549–559, 1998.

[10] K. Pearson, “On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling,” Philosophical Magazine Series 5, vol.50, no.302, pp.157–175, 1900.

[11] M. Sugiyama, T. Suzuki, T. Kanamori, M.C. du Plessis, S. Liu, and I. Takeuchi, “Density-difference estimation,” Neural Computation, vol.25, no.10, pp.2734–2775, 2013.
[12] A. Gretton, K.M. Borgwardt, M. Rasch, B. Schölkopf, and A.J. Smola, “A kernel method for the two-sample-problem,” Advances in Neural Information Processing Systems 19, ed. B. Schölkopf, J. Platt, and T. Hoffman, pp.513–520, MIT Press, Cambridge, MA, USA, 2007.
[13] N. Aronszajn, “Theory of reproducing kernels,” Transactions of the American Mathematical Society, vol.68, no.3, pp.337–404, 1950.
[14] A. Iyer, S. Nath, and S. Sarawagi, “Maximum mean discrepancy for class ratio estimation: Convergence bounds and kernel selection,” Proc. 31st International Conference on Machine Learning (ICML2014), pp.530–538, Beijing, China, 2014.
[15] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. Sriperumbudur, “Optimal kernel choice for large-scale two-sample tests,” Advances in Neural Information Processing Systems, pp.1205–1213, 2012.
[16] G.J. Székely and M.L. Rizzo, “Energy statistics: A class of statistics based on distances,” Journal of Statistical Planning and Inference, vol.143, no.8, pp.1249–1272, 2013.
[17] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu, “Equivalence of distance-based and RKHS-based statistics in hypothesis testing,” The Annals of Statistics, vol.41, no.5, pp.2263–2291, 2013.
[18] O. Chapelle, B. Schölkopf, and A. Zien, eds., “Semi-Supervised Learning,” MIT Press, Cambridge, MA, USA, 2006.
[19] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, NY, USA, 1998.
[20] M. Sugiyama, S. Liu, M.C. du Plessis, M. Yamanaka, M. Yamada, T. Suzuki, and T. Kanamori, “Direct divergence approximation between probability distributions and its applications in machine learning,” J. Computing Science and Engineering, vol.7, no.2, pp.99–111, 2013.
[21] T. Kanamori, S. Hido, and M. Sugiyama, “A least-squares approach to direct importance estimation,” J. Machine Learning Research, vol.10, pp.1391–1445, July 2009.
[22] S.M. Ali and S.D. Silvey, “A general class of coefficients of divergence of one distribution from another,” J. Royal Statistical Society, Series B, vol.28, no.1, pp.131–142, 1966.
[23] I. Csiszár, “Information-type measures of difference of probability distributions and indirect observation,” Studia Scientiarum Mathematicarum Hungarica, vol.2, pp.229–318, 1967.
[24] K. Fukumizu, B.K. Sriperumbudur, A. Gretton, and B. Schölkopf, “Characteristic kernels on groups and semigroups,” Advances in Neural Information Processing Systems 21, ed. D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, pp.473–480, 2009.
[25] C.C. Chang and C.J. Lin, “LIBSVM: A library for support vector machines,” tech. rep., Department of Computer Science, National Taiwan University, 2001. http://www.csie.ntu.edu.tw/˜cjlin/libsvm/.
[26] M. Sugiyama, M. Krauledat, and K.R. Müller, “Covariate shift adaptation by importance weighted cross validation,” J. Machine Learning Research, vol.8, pp.985–1005, May 2007.
[27] J. Friedman, T. Hastie, and R. Tibshirani, The elements of statistical learning, Springer, Berlin, 2001.
[28] F. Zhang, The Schur complement and its applications, Springer Science & Business Media, 2006.

Appendix A: Proof of Theorem 1

If the matrix $\boldsymbol{B}$ is positive semi-definite, $\dot{J}(\dot{\boldsymbol{\theta}})$ is convex with respect to $\dot{\boldsymbol{\theta}}$. So we prove the positive semi-definiteness of $\boldsymbol{B}$ below.

Proof : Let $\mathcal{X}$ be the domain of input vector $\boldsymbol{x}$, $\mathcal{H}_k$ be an RKHS, K be a positive-semidefinite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined as

$$K(\boldsymbol{x}, \check{\boldsymbol{x}}) = -\|\boldsymbol{x} - \check{\boldsymbol{x}}\| + \|\boldsymbol{x}\| + \|\check{\boldsymbol{x}}\|.$$

K is called the *distance kernel* [17]. The map $\varphi : \mathcal{X} \to \mathcal{H}_k$, $\varphi(\boldsymbol{x}) : \boldsymbol{x} \mapsto K(\cdot, \boldsymbol{x})$ is the canonical feature map. Let $\overline{\varphi}_y$ be the true mean of the feature vectors of the y-th class:

$$\overline{\varphi}_y = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y)}\varphi(\boldsymbol{x}).$$

Since $\overline{\varphi}_y{}^\top \overline{\varphi}_{y'} = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y), \check{\boldsymbol{x}}\sim p(\boldsymbol{x}|y')} K(\boldsymbol{x}, \check{\boldsymbol{x}})$, $A_{y,y'}$ can be expressed as

$$\begin{aligned} A_{y,y'} &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y), \check{\boldsymbol{x}}\sim p(\boldsymbol{x}|y')}\left[-K(\boldsymbol{x}, \check{\boldsymbol{x}}) + \|\boldsymbol{x}\| + \|\check{\boldsymbol{x}}\|\right] \\ &= -\overline{\varphi}_y^\top \overline{\varphi}_{y'} + \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x}|y)}\|\boldsymbol{x}\| + \mathbb{E}_{\check{\boldsymbol{x}}\sim p(\boldsymbol{x}|y')}\|\check{\boldsymbol{x}}\|. \end{aligned}$$

Then we can rewrite $B_{y,y'}$ as follows:

$$\begin{aligned} B_{y,y'} &= \overline{\varphi}_y^\top \overline{\varphi}_{y'} - \overline{\varphi}_y^\top \overline{\varphi}_c - \overline{\varphi}_c^\top \overline{\varphi}_{y'} + \overline{\varphi}_c^\top \overline{\varphi}_c \\ &= (\overline{\varphi}_y - \overline{\varphi}_c)^\top (\overline{\varphi}_{y'} - \overline{\varphi}_c) \\ &= (\tilde{\boldsymbol{B}}^\top \tilde{\boldsymbol{B}})_{y,y'}, \end{aligned}$$

where $\tilde{\boldsymbol{B}} = [\overline{\varphi}_1 - \overline{\varphi}_c, \cdots, \overline{\varphi}_{c-1} - \overline{\varphi}_c]$. Since $\boldsymbol{B}$ is a positive-semidefinite matrix, $\dot{J}(\dot{\boldsymbol{\theta}})$ is convex. □

Appendix B: Proof of Theorem 2

If the matrix $\boldsymbol{B}$ is strictly positive definite, $\dot{J}(\dot{\boldsymbol{\theta}})$ is strongly convex with respect to $\dot{\boldsymbol{\theta}}$. So we prove that $\boldsymbol{B}$ is strictly positive definite below.

Proof :

$$\begin{aligned} B_1 &= -A_{1,1} + 2A_{1,c} - A_{c,c} \\ &= \mathrm{ED}(p(\boldsymbol{x}|y = 1), p(\boldsymbol{x}|y = c)) > 0. \end{aligned}$$

Then we can immediately prove that $\boldsymbol{B}$ is strictly positive definite, from the Schur complement condition for positive definiteness [28]. □

Hideko Kawakubo was born in Kanagawa, Japan in 1972. In 1996, she received a Bachelor of Economics degree from Rikkyo University. She then received a Bachelor of Mathematics and a Master of Science degree from Ochanomizu University in 2011 and 2013. She is currently pursuing a Ph.D. in computer science at Tokyo Institute of Technology. Her research interests are statistical machine learning algorithms and their applications.

Marthinus Christoffel du Plessis was born in Pretoria, South Africa in 1984. He received the degrees of Bachelor of Engineering and Master of Engineering from the University of Pretoria in 2007 and 2009. In 2014 he received the degree of Doctor of Engineering from Tokyo Institute of Technology. Currently, he is a project assistant professor at the University of Tokyo. His research interests are machine learning and pattern recognition.

Masashi Sugiyama was born in Osaka, Japan, in 1974. He received the degrees of Bachelor of Engineering, Master of Engineering, and Doctor of Engineering in Computer Science from Tokyo Institute of Technology, Japan in 1997, 1999, and 2001, respectively. In 2001, he was appointed Assistant Professor in the same institute, and he was promoted to Associate Professor in 2003. He moved to the University of Tokyo as Professor in 2014. He received an Alexander von Humboldt Foundation Research Fellowship and researched at Fraunhofer Institute, Berlin, Germany, from 2003 to 2004. In 2006, he received an European Commission Program Erasmus Mundus Scholarship and researched at the University of Edinburgh, Edinburgh, UK. He received the Faculty Award from IBM in 2007 for his contribution to machine learning under non-stationarity, the Nagao Special Researcher Award from the Information Processing Society of Japan in 2011 and the Young Scientists' Prize from the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology Japan for his contribution to the density-ratio paradigm of machine learning. His research interests include theories and algorithms of machine learning and data mining, and a wide range of applications such as signal processing, image processing, and robot control.