Learning under Non-Stationarity: Covariate Shift and Class-Balance Change

Masashi Sugiyama
Tokyo Institute of Technology, Japan.
sugi@cs.titech.ac.jp
http://sugiyama-www.cs.titech.ac.jp/~sugi

Makoto Yamada
Yahoo! Labs, USA
makotoy@yahoo-inc.com

Marthinus Christoffel du Plessis
Tokyo Institute of Technology, Japan.
christo@sg.cs.titech.ac.jp

Abstract

One of the fundamental assumptions behind many supervised machine learning algorithms is that training and test data follow the same probability distribution. However, this important assumption is often violated in practice, for example, because of an unavoidable sample selection bias or non-stationarity of the environment. Due to violation of the assumption, standard machine learning methods suffer a significant estimation bias. In this article, we consider two scenarios of such distribution change — the *covariate shift* where input distributions differ and *class-balance change* where class-prior probabilities vary in classification — and review semi-supervised adaptation techniques based on *importance weighting*.

Keywords

Covariate shift, class-balance change, importance weighting, density ratio estimation, divergence approximation

1 Introduction

The goal of supervised learning such as regression and classification is to learn an input-output dependency from input-output paired training samples so that test output y' for unseen test input $\boldsymbol{x}'$ can be accurately estimated. Various supervised learning algorithms were developed thus far, and they have been demonstrated to be useful in a wide range of applications. Most of the popular machine learning algorithms assume that training and test data follow the same probability distribution, based on which learning machines can generalize to unseen test data from training data [63, 25, 10]. However, this fundamental assumption is often violated in practice, and this causes standard supervised learning algorithms suffer significant estimation bias.

In this article, we consider two scenarios. The first setup is the *covariate shift* [44, 48], where training and test input data follow different distributions but the input-output relation does not change between training and test phases. The other setup is called *class-balance change* in classification [42, 17], where the class-prior probabilities are different in training and test phases but the input distribution of each class does not change. For these two scenarios, we review semi-supervised adaptation techniques, where *importance weighting* plays an essential role.

More specifically, we consider the semi-supervised learning problem where input-output training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ and input-only test samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are available. In the standard semi-supervised learning setup, training and test samples are regarded as being drawn from the same probability distribution [13]. In contrast, in this article, we suppose that they are drawn from different distributions: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ are drawn independently from a joint probability distribution with density $p(\boldsymbol{x}, y)$ and $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are drawn independently from a marginal probability distribution with density $\int p'(\boldsymbol{x}, y)\mathrm{d}y$, where $p(\boldsymbol{x}, y)$ and $p'(\boldsymbol{x}, y)$ are different:

$$p(\boldsymbol{x}, y) \neq p'(\boldsymbol{x}, y).$$

Our goal is to learn the input-output relation for test samples. The situation where training and test samples follow different distributions is also referred to as *non-stationarity adaptation*, *dataset-shift adaptation*, *transfer learning*, and *domain adaptation*. The semi-supervised learning setup with differing training and testing distributions is sometimes called *unsupervised transfer* or *unsupervised adaptation* in literature because no supervision is available from the test domain.

2 Adaptation Techniques for Covariate Shift

The *covariate shift* [44, 48] is the situation where input distributions change but the conditional distribution of outputs given inputs remains unchanged:

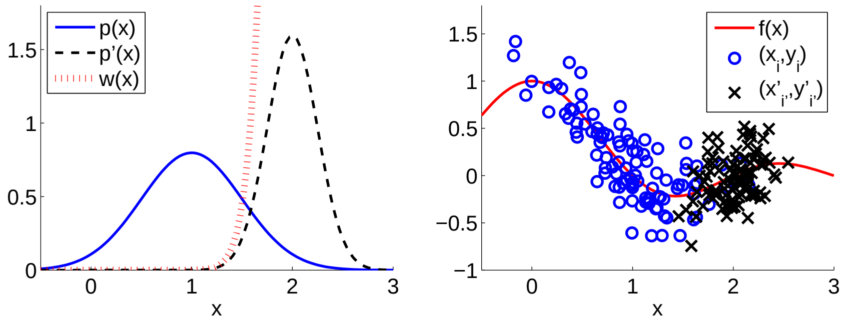
$$p(\boldsymbol{x}) \neq p'(\boldsymbol{x}) \quad \text{and} \quad p(y|\boldsymbol{x}) = p'(y|\boldsymbol{x}).$$

Figure 1 illustrates an example of covariate shift regression: Training input samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ are drawn from the left-hand side of the domain, whereas test input samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are drawn from the right-hand side. This problem is similar to *extrapolation* since the prediction is made in a low density region of the training set.

2.1 Importance-Weighted Learning

For this covariate-shift regression problem, let us use a simple linear model,

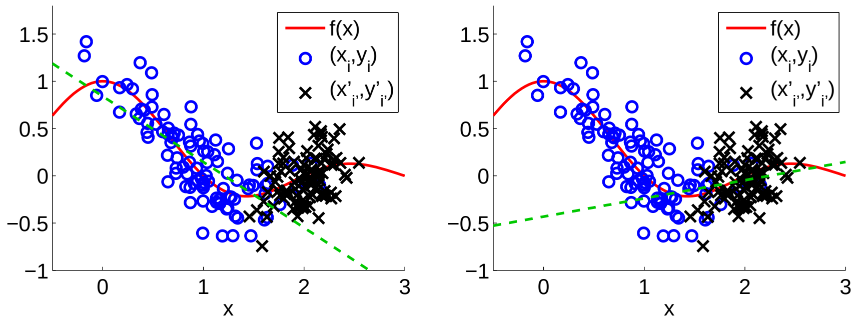
$$f_{\boldsymbol{\theta}}(x) = \theta_1 + \theta_2 x,$$



(a) Input densities and importance

(b) Learning target function, training samples, and test samples

Figure 1: Covariate shift. Input distributions change but the conditional distribution of outputs given inputs does not changed.



(a) Ordinary least-squares

(b) Importance-weighted least-squares

Figure 2: Regression under covariate shift. Dashed lines denote learned functions.

and train this model by *ordinary least-squares*:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i \Big)^2 .$$

The learned result illustrated in Figure 2(a) shows that the obtained function fits the training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ very well, but it does not give good prediction of outputs for the test input samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ (i.e., samples denoted by "×").

Under the covariate shift, it is expected that only training samples whose input points are close to test input samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ are useful. This intuitive idea can be realized by weighting the training loss according to the *importance*, which is the ratio between $p'(\boldsymbol{x})$ and $p(\boldsymbol{x})$.

$$w(\boldsymbol{x}) := \frac{p'(\boldsymbol{x})}{p(\boldsymbol{x})}.$$

In Figure 2(b), the learned result obtained by *importance-weighted least-squares* [44],

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} w(\boldsymbol{x}_i)\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\Big)^2,$$

is illustrated. This shows that importance weighting can improve the accuracy of predicting outputs for the test input samples $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$.

The above importance-weighted least-squares can be regarded as an application of *importance weighting* to approximating the generalization error (or the expected test loss):

$$G := \iint \mathrm{loss}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))p'(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}\mathrm{d}y,$$

where $\mathrm{loss}(y, \widehat{y})$ denotes a point-wise loss when y is predicted by $\widehat{y}$. More specifically, the generalization error G can be approximated by the importance-weighted average of the training loss:

$$\begin{aligned} G &= \iint \mathrm{loss}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))p'(y|\boldsymbol{x})p'(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}y \\ &= \iint \mathrm{loss}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))p'(y|\boldsymbol{x})\frac{p'(\boldsymbol{x})}{p(\boldsymbol{x})}p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}y \\ &= \iint \mathrm{loss}(y, f_{\boldsymbol{\theta}}(\boldsymbol{x}))w(\boldsymbol{x})p(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}\mathrm{d}y \\ &\approx \frac{1}{n}\sum_{i=1}^{n} \mathrm{loss}(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))w(\boldsymbol{x}_i). \end{aligned}$$

Note that this importance weighting idea can be applied to *any* likelihood/loss-based learning algorithms, including *Fisher discriminant analysis*, *logistic regression*, the *support vector machine*, *boosting*, and the *conditional random field*, and it also plays an important role for reducing the estimation bias in active learning and experimental design scenarios [65, 28, 47, 26, 54, 52]. See [48] for more thorough discussion on importance-weighted learning.

To implement importance-weighted learning, importance values $\{w(\boldsymbol{x}_i)\}_{i=1}^{n}$ are necessary. However, training and test input densities $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ are unknown in practice, and thus the importance values should be estimated from data. A naive approach is to estimate $p(\boldsymbol{x})$ from $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $p'(\boldsymbol{x})$ from $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ separately and then take their ratio. However, such a two-step procedure is not accurate because the error incurred in the estimation of $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ can be increased when their ratio is computed in the second stage. Thus, *directly* estimating the ratio $w(\boldsymbol{x})$ without estimating $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ is more preferable.

Following this idea, various methods of importance estimation have been developed, for example, based on density estimation of $p'(\boldsymbol{x})$ after uniformization of $p(\boldsymbol{x})$ [16, 14], logistic regression for discriminating data from $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ [36, 15, 7], moment matching between $p'(\boldsymbol{x})$ and $p(\boldsymbol{x})w(\boldsymbol{x})$ [36, 18, 29], integral equations between $p'(\boldsymbol{x})$ and $p(\boldsymbol{x})w(\boldsymbol{x})$ [64, 37], density matching between $p'(\boldsymbol{x})$ and $p(\boldsymbol{x})w(\boldsymbol{x})$ under the Kullback-Leibler divergence [58, 34, 61, 67, 70], least-squares importance fitting of $w(\boldsymbol{x})$ to $p'(\boldsymbol{x})/p(\boldsymbol{x})$ [27, 29], and importance fitting of $w(\boldsymbol{x})$ to $p'(\boldsymbol{x})/p(\boldsymbol{x})$ under the Bregman divergence [56].

Among them, the least-squares importance fitting method has various practical advantages, for example, an analytic-form solution that can be computed efficiently is available, cross-validation is available for hyperparameter tuning, the optimal convergence rate is achieved both in parametric and non-parametric settings [27, 29], and the highest numerical stability in terms of condition numbers is achieved among a class of importance estimators [30]. Furthermore, dimensionality reduction methods for improving the accuracy of importance estimation in high-dimensional problems have been developed [49, 59, 68]. See [55] for more comprehensive discussion on direct importance estimation.

2.2 Relative Importance-Weighted Learning

Let us continue using the illustrative example described in Figure 1 and Figure 2. The true importance function $w(x)$ is plotted in Figure 1(a). This shows that, among many training samples, only a small number of samples at around $x = 2$ have large importance weights and other samples have almost zero weights. This implies that importance-weighted learning in this example is rather unreliable because the learned function is essentially obtained from only a few training samples.

Such unreliable behavior is caused by the fact that the importance function $w(\boldsymbol{x})$ can take very large values. To cope with this problem, the *relative importance weight* is useful [71]:

$$w^{(\beta)}(\boldsymbol{x}) = \frac{p'(\boldsymbol{x})}{\beta p'(\boldsymbol{x}) + (1-\beta)p(\boldsymbol{x})},$$

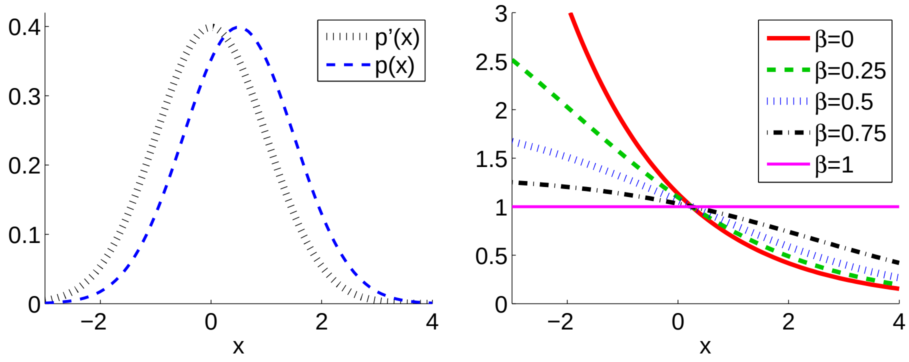
where $\beta \in [0, 1]$ is the relativity parameter. The relative importance weight $w^{(\beta)}(\boldsymbol{x})$ is reduced to the ordinary importance weight $w(\boldsymbol{x})$ when $\beta = 0$. As β is increased, the relative importance weight gets flatter and is reduced to the uniform weight $w^{(\beta)}(\boldsymbol{x}) = 1$ when $\beta = 1$ (Figure 3). The non-negativity of the importance function, $p'(\boldsymbol{x})/p(\boldsymbol{x}) \geq 0$, assures that the relative importance weight is bounded from above by $1/\beta$:

$$w^{(\beta)}(\boldsymbol{x}) = \frac{1}{\beta + (1-\beta)\frac{p(\boldsymbol{x})}{p'(\boldsymbol{x})}} \leq \frac{1}{\beta}.$$

The least-squares method combined with the relative importance weight is called *relative importance-weighted least-squares*:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i=1}^{n} w^{(\beta)}(\boldsymbol{x}_i)\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i\Big)^2,$$

where the relativity parameter β controls the trade-off between bias and variance.



(a) Probability densities (b) Relative importance $w^{(\beta)}(x)$

Figure 3: Relative importance. $p'(x)$ is the normal distribution with mean 0 and variance 1, and $p(x)$ is the normal distribution with mean 0.5 and variance 1.

Now let us consider the problem of estimating the relative importance weight $w^{(\beta)}(\boldsymbol{x})$ from $\{\boldsymbol{x}_i\}_{i=1}^{n}$ and $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$. We use the following linear-in-parameter model $w_{\boldsymbol{\alpha}}(\boldsymbol{x})$ for learning the relative importance weight $w^{(\beta)}(\boldsymbol{x})$:

$$w_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{j=1}^{b} \alpha_j \psi_j(\boldsymbol{x}) = \boldsymbol{\alpha}^\top \boldsymbol{\psi}(\boldsymbol{x}),$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_b)^\top$ is the parameter vector and $\boldsymbol{\psi}(\boldsymbol{x}) = (\psi_1(\boldsymbol{x}), \ldots, \psi_b(\boldsymbol{x}))^\top$ is the basis function vector. As basis functions, we may use, for example, the Gaussian kernels:

$$w_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{j=1}^{n'} \alpha_j \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'_j\|^2}{2\sigma^2}\right),$$

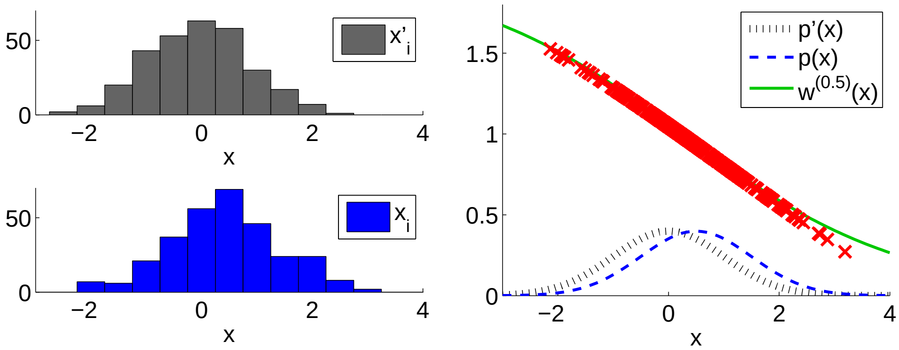
where σ^2 denotes the Gaussian width.

Then the parameter $\boldsymbol{\alpha}$ is learned so that the following criterion $J(\boldsymbol{\alpha})$ is minimized:

$$\begin{aligned} J(\boldsymbol{\alpha}) &= \int \Big(w_{\boldsymbol{\alpha}}(\boldsymbol{x}) - w^{(\beta)}(\boldsymbol{x})\Big)^2 \Big(\beta p'(\boldsymbol{x}) + (1-\beta)p(\boldsymbol{x})\Big)\mathrm{d}\boldsymbol{x} \\ &= \int \boldsymbol{\alpha}^\top \boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x})^\top \boldsymbol{\alpha}\Big(\beta p'(\boldsymbol{x}) + (1-\beta)p(\boldsymbol{x})\Big)\mathrm{d}\boldsymbol{x} \\ &\quad - 2\int \boldsymbol{\alpha}^\top \boldsymbol{\psi}(\boldsymbol{x}) p'(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + C, \end{aligned}$$

where the third term,

$$C = \int w^{(\beta)}(\boldsymbol{x}) p'(\boldsymbol{x})\mathrm{d}\boldsymbol{x},$$



(a) Training and test data (b) Relative importance weight ($\beta = 0.5$)

Figure 4: Illustration of RuLSIF. "×" in Figure 4(b) denotes an estimated relative importance value at $\boldsymbol{x}_i$.

is a constant irrelevant to the parameter $\boldsymbol{\alpha}$ and thus can be ignored. Approximating the expectations in the first and second terms by sample averages and adding the ℓ_2-regularizer, we have the following training criterion:

$$\min_{\boldsymbol{\alpha}} \left[\boldsymbol{\alpha}^\top \widehat{\boldsymbol{G}}_\beta \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \widehat{\boldsymbol{h}} + \lambda \|\boldsymbol{\alpha}\|^2 \right],$$

where $\widehat{\boldsymbol{G}}_\beta$ and $\widehat{\boldsymbol{h}}$, a $b \times b$ matrix and a b-dimensional vector, are defined as

$$\widehat{\boldsymbol{G}}_\beta = \frac{\beta}{n'} \sum_{i'=1}^{n'} \boldsymbol{\psi}(\boldsymbol{x}'_{i'}) \boldsymbol{\psi}(\boldsymbol{x}'_{i'})^\top + \frac{1-\beta}{n} \sum_{i=1}^{n} \boldsymbol{\psi}(\boldsymbol{x}_i) \boldsymbol{\psi}(\boldsymbol{x}_i)^\top \quad \text{and} \quad \widehat{\boldsymbol{h}} = \frac{1}{n'} \sum_{i'=1}^{n'} \boldsymbol{\psi}(\boldsymbol{x}'_{i'}).$$

This training criterion is a convex quadratic function of $\boldsymbol{\alpha}$ and its minimizer $\widehat{\boldsymbol{\alpha}}$ can be obtained analytically as

$$\widehat{\boldsymbol{\alpha}} = \left(\widehat{\boldsymbol{G}}_\beta + \lambda \boldsymbol{I} \right)^{-1} \widehat{\boldsymbol{h}}.$$

This method is called *relative unconstrained least-squares importance fitting* (RuLSIF) [71]. Tuning parameters such as the regularization parameter λ and the Gaussian width σ^2 can be optimized via cross-validation with respect to J.

An example of relative importance estimation by RuLSIF is illustrated in Figure 4.

2.3 Importance-Weighted Model Selection

Choice of the relativity parameter β as well as other tuning parameters such as basis functions and regularization parameters is crucial for obtaining better performance in practice. For model selection, various methods such as the *Akaike information criterion* [2], the *subspace information criterion* [53], and *cross-validation* [46] are available.

However, under the covariate shift, these model selection techniques based on training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ do not give valid evaluation of the prediction accuracy of outputs for test inputs $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$.

Under the covariate shift, importance-weighted variants of such model selection methods are useful [44, 51, 50]. The simplest model selection method called *importance-weighted cross-validation* is given as follows:

1. Randomly split training samples $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ into m disjoint subsets $\{\mathcal{T}_i\}_{i=1}^{m}$ of (approximately) the same size.

2. Repeat for $i = 1, \ldots, m$;

   (a) Obtain a learned function f_i from $\mathcal{T}\backslash\mathcal{T}_i$ (i.e., all samples without $\mathcal{T}_i$).

   (b) Evaluate the generalization error using hold-out samples $\mathcal{T}_i$ as

$$\widehat{G}_i = \begin{cases} \dfrac{1}{|\mathcal{T}_i|} \displaystyle\sum_{(\boldsymbol{x},y)\in\mathcal{T}_i} w(\boldsymbol{x})\Big(f_i(\boldsymbol{x}) - y\Big)^2 & \text{(Regression)}, \\ \dfrac{1}{|\mathcal{T}_i|} \displaystyle\sum_{(\boldsymbol{x},y)\in\mathcal{T}_i} \dfrac{w(\boldsymbol{x})}{2}\Big(1 - \mathrm{sign}\big(f_i(\boldsymbol{x})y\big)\Big) & \text{(Classification)}, \end{cases}$$

   where $|\mathcal{T}_i|$ denotes the number of elements in the set $\mathcal{T}_i$.

3. Output the average of $\widehat{G}_1, \ldots, \widehat{G}_m$ as the final evaluation $\widehat{G}$ of the generalization error:

$$\widehat{G} = \frac{1}{m}\sum_{i=1}^{m} \widehat{G}_i.$$

2.4 Applications

Importance-weighted learning has been successfully applied to various real-world problems, including brain-computer interface [50, 33], robot control [19, 3, 20, 72], speaker identification [69], age prediction from face images [62], activity recognition from accelerometers [21], natural language processing [61], spam filtering [9], targeted advertising [8], HIV therapy screening [6], and wafer alignment in semiconductor exposure apparatus [52]. Below, we describe application of covariate shift adaptation in 3D human-pose estimation from monocular videos [66].

We use the HumanEva-I dataset [45], which contains synchronized multi-view videos and motion-capture data for 3 subjects performing multiple activities: Walking, jogging, boxing, throwing and catching, and gesturing. As input $\boldsymbol{x}$, we extract the *histogram-of-oriented-gradient (HoG) feature* [11] of 270 dimensions from videos taken by 3 color cameras with 9630 image-pose frames for each camera. Output $\boldsymbol{y}$ is a corresponding pose vector, which means that we consider a multi-dimensional regression problem. We randomly select n samples from the set of $3 \times 4815 = 14445$ frames for training and use the remaining 14445 frames for testing.

We consider the following scenarios:

Selection bias: The training set contains data from all 3 subjects, whereas the test set only contains data from a single subject.

Subject transfer: The training set contains data from 2 subjects, whereas the test set contains data from the remaining subject not included in the training set.

As regression algorithms, we use *kernel regression* (KR) [1], *twin Gaussian processes regression* (TGP) [11], and the *weighted k-nearest neighbor* (WkNN) method [43]. See [66] for the details of these algorithms. For KR and TGP, we consider their importance-weighted variants which are referred to as IWKR and IWTGP.

Each pose is represented by 20 3D-joint markers: $\boldsymbol{y} = [\boldsymbol{y}^{(1)\top}, \dots, \boldsymbol{y}^{(20)\top}]^\top \in \mathbb{R}^{60}$, where $\boldsymbol{y}^{(m)} \in \mathbb{R}^3$ for $m = 1, \dots, 20$. Error between true pose $\boldsymbol{y}^*$ and its estimate $\widehat{\boldsymbol{y}}$ is measured by the average Euclidean distance:

$$\mathrm{Error}(\boldsymbol{y}^*, \widehat{\boldsymbol{y}}) = \frac{1}{20} \sum_{m=1}^{20} \|\widehat{\boldsymbol{y}}^{(m)} - \boldsymbol{y}^{*(m)}\|.$$

Figure 5 shows the pose estimation error as a function of the training sample size n averaged over all motions and 10 runs. The graphs clearly show that IWTGP and IWKR outperform their non-adaptive counterparts and the baseline WkNN method.

3 Adaptation Techniques for Class-Balance Change

Class-balance change [42, 17] is the classification problem where class-prior probabilities change but the conditional distribution of input $\boldsymbol{x}$ given class y remains unchanged:

$$p(y) \neq p'(y) \quad \text{and} \quad p(\boldsymbol{x}|y) = p'(\boldsymbol{x}|y). \tag{1}$$

Figure 6 illustrates an example of classification under class-balance change. When the class balances are different in the training and test phases, naive training of a classifier yields significant estimation bias even if the class-conditional input density is unchanged.

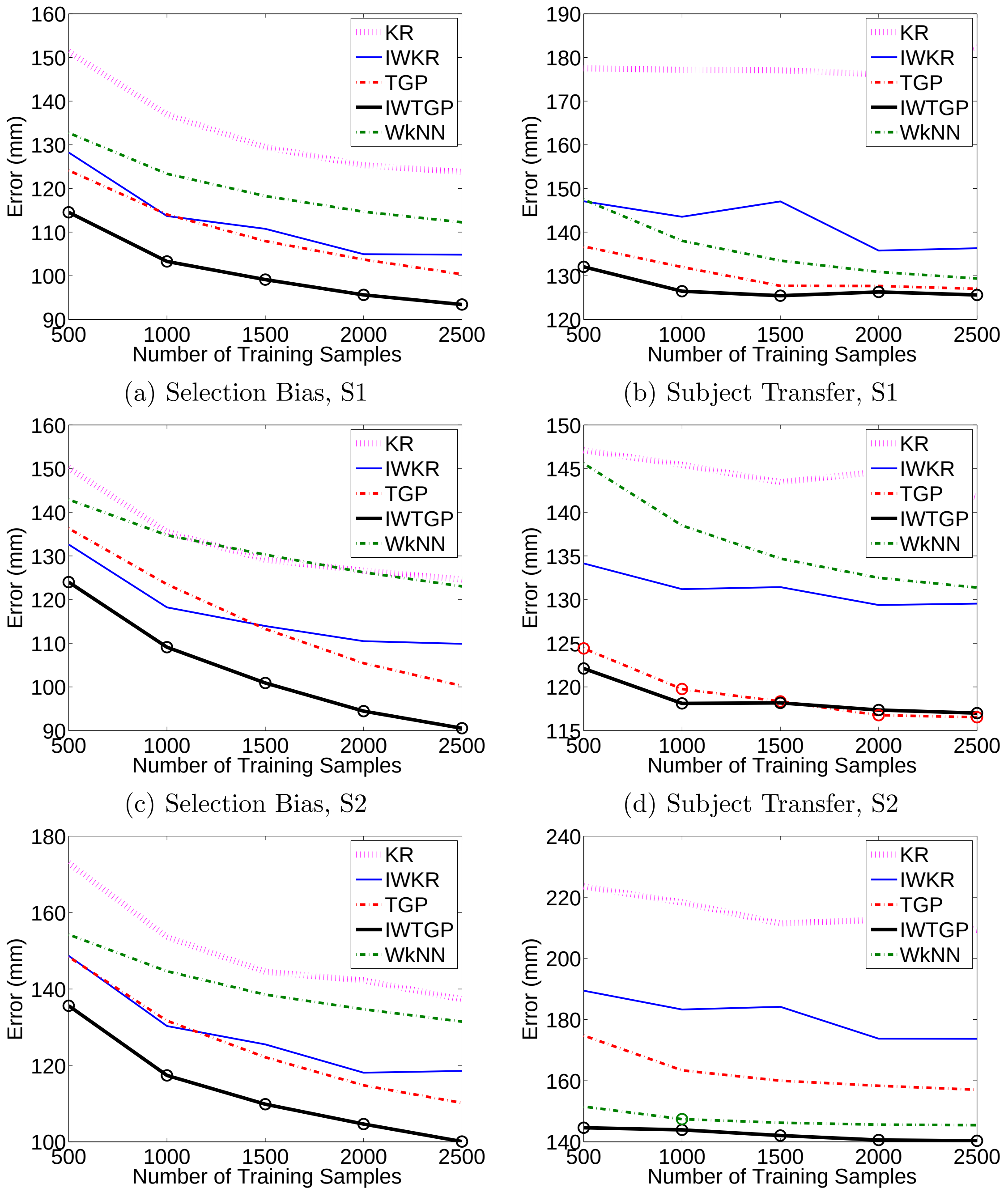
In the same way as covariate shift adaptation, estimation bias caused by class-balance change can be canceled by weighting the training loss according to the *class-balance ratio*:

$$w(y) = \frac{p'(y)}{p(y)}.$$

Below, we focus on binary classification where label y takes either $+1$ or -1 for simplicity.

3.1 Class-Balance Estimation

The training class-balance $p(y)$ can be naively estimated by n_y/n if n_y samples belong to class y in the training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. The test class-balance $p'(y)$ can also be estimated

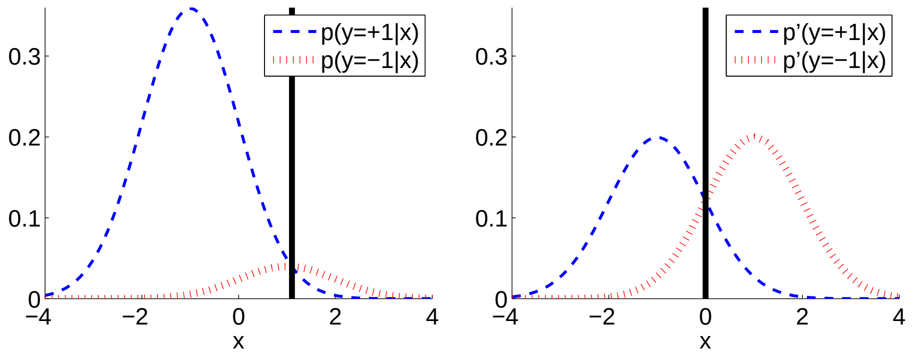


(a) Selection Bias, S1 (b) Subject Transfer, S1

(c) Selection Bias, S2 (d) Subject Transfer, S2

(e) Selection Bias, S3 (f) Subject Transfer, S3

Figure 5: 3D human-pose estimation error as a function of the number of training samples averaged over all motions for each subject. The best method and comparable ones in terms of the average error according to the paired *t-test* at the significance level 5% are specified by '$\circ$'.



(a) Training data

(b) Test data

Figure 6: Change in class balances shifts the optimal classification boundary. Class-conditional input density is the same between the training and test phases (i.e., $p(\boldsymbol{x}|y) = p'(\boldsymbol{x}|y)$), but class-prior probabilities are different (i.e., $p(y) \neq p'(y)$).

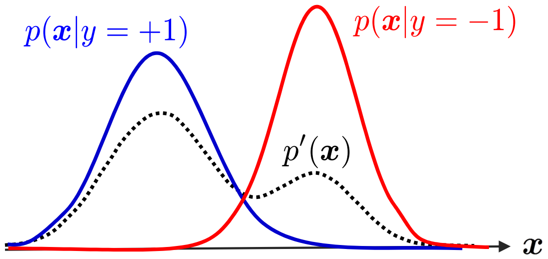


Figure 7: $p'(y)$ can be estimated by fitting a mixture of training class-wise densities $p(\boldsymbol{x}|y)$ to test input density $p'(\boldsymbol{x})$.

in the same way if a labeled test set $\{(\boldsymbol{x}'_{i'}, y'_{i'})\}_{i'=1}^{n'}$ is available. However, we are considering a semi-supervised learning setup where only an unlabeled test set $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$ is available. Thus, $p'(y)$ cannot be estimated naively.

In the semi-supervised learning setup under Eq.(1), $p'(y)$ can be estimated by fitting a mixture $q_\pi(\boldsymbol{x})$ of training class-wise densities $p(\boldsymbol{x}|y)$ to test input density $p'(\boldsymbol{x})$ (see Figure 7):

$$q_\pi(\boldsymbol{x}) = \pi p(\boldsymbol{x}|y=+1) + (1-\pi)p(\boldsymbol{x}|y=-1).$$

The value of the parameter π corresponds to $p'(y=+1)$, whereas $1-\pi$ corresponds to $p'(y=-1)$.

For the fitting of q_π to p', we may use the *Kullback-Leibler (KL) divergence* [32] or the *Pearson (PE) divergence* [35]:

$$\mathrm{KL}(p'\|q_\pi) = \int p'(\boldsymbol{x}) \log \frac{p'(\boldsymbol{x})}{q_\pi(\boldsymbol{x})} \mathrm{d}\boldsymbol{x},$$

$$\mathrm{PE}(p'\|q_\pi) = \int q_\pi(\boldsymbol{x}) \left(\frac{p'(\boldsymbol{x})}{q_\pi(\boldsymbol{x})} - 1 \right)^2 \mathrm{d}\boldsymbol{x}.$$

These divergences can be accurately approximated from samples by directly estimating the density ratio $p'(\boldsymbol{x})/q_\pi(\boldsymbol{x})$ without density estimation of $p'(\boldsymbol{x})$ and $q_\pi(\boldsymbol{x})$ [55]. However, the density ratio function $p'(\boldsymbol{x})/q_\pi(\boldsymbol{x})$ is sensitive to small variation, and therefore it is not robust against outliers.

Here we consider the L^2-*distance* between p' and q_π:

$$L^2(p', q_\pi) = \int \Big(p'(\boldsymbol{x}) - q_\pi(\boldsymbol{x})\Big)^2 \mathrm{d}\boldsymbol{x}.$$

The L^2-distance can also be accurately approximated from samples by directly estimating the density difference $p'(\boldsymbol{x}) - q_\pi(\boldsymbol{x})$, without density estimation of $p'(\boldsymbol{x})$ and $q_\pi(\boldsymbol{x})$ [57].

Historically, non-parametric estimation of mixture proportion π under the L^2-distance was first investigated in [22], which uses empirical distribution functions. Following this seminal work, its variant based on kernel density estimation has been developed [60], and this is further extended to choosing the kernel bandwidths jointly [23]. In the related context of two-sample homogeneity testing under the L^2-distance, the use of kernel density estimators with fixed and equal bandwidths has been investigated [4].

3.2 L^2-Distance Approximation

Here, we explain how the L^2-distance can be directly approximated from data via direct density-difference estimation [31, 57]. For simplicity, we consider the approximation problem of the L^2-distance between p and p',

$$L^2(p, p') = \int f(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x}, \quad \text{where } f(\boldsymbol{x}) = p(\boldsymbol{x}) - p'(\boldsymbol{x}), \tag{2}$$

from $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{x}'_{i'}\}_{i'=1}^{n'}$.

We use the following Gaussian density-difference model:

$$g(\boldsymbol{x}) = \sum_{j=1}^{n+n'} \alpha_j \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_j\|^2}{2\sigma^2}\right),$$

where

$$(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{c}_{n+1}, \ldots, \boldsymbol{c}_{n+n'}) = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_{n'})$$

are Gaussian centers. The parameter $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{n+n'})^\top$ in the density-difference model is learned so that the following criterion $J(\boldsymbol{\alpha})$ is minimized:

$$\begin{aligned} J(\boldsymbol{\alpha}) &= \int \Big(g(\boldsymbol{x}) - f(\boldsymbol{x})\Big)^2 \mathrm{d}\boldsymbol{x} \\ &= \int g(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x} - 2\int g(\boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + C, \end{aligned}$$

where the third term,

$$C = \int f(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x},$$

is a constant irrelevant to the parameter $\boldsymbol{\alpha}$ and thus can be ignored. The first term can be computed analytically as

$$\int g(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x} = \boldsymbol{\alpha}^\top \boldsymbol{U} \boldsymbol{\alpha},$$

where $\boldsymbol{U}$ is the $(n+n') \times (n+n')$ matrix with the (j, j')-th element defined by

$$\begin{aligned} U_{j,j'} &= \int \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_j\|^2}{2\sigma^2}\right) \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_{j'}\|^2}{2\sigma^2}\right) \mathrm{d}\boldsymbol{x} \\ &= (\pi\sigma^2)^{d/2} \exp\left(-\frac{\|\boldsymbol{c}_j - \boldsymbol{c}_{j'}\|^2}{4\sigma^2}\right). \end{aligned}$$

Approximating the expectations in the second term by sample averages and adding the ℓ_2-regularizer, we have the following training criterion:

$$\min_{\boldsymbol{\alpha}} \left[\boldsymbol{\alpha}^\top \boldsymbol{U} \boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \widehat{\boldsymbol{v}} + \lambda \|\boldsymbol{\alpha}\|^2\right],$$

where $\widehat{\boldsymbol{v}}$ is the $(n+n')$-dimensional vector with the j-th element defined by

$$\widehat{v}_j = \frac{1}{n} \sum_{i=1}^{n} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{c}_j\|^2}{2\sigma^2}\right) - \frac{1}{n'} \sum_{i'=1}^{n'} \exp\left(-\frac{\|\boldsymbol{x}'_{i'} - \boldsymbol{c}_j\|^2}{2\sigma^2}\right).$$

This training criterion is a convex quadratic function of $\boldsymbol{\alpha}$ and its minimizer $\widehat{\boldsymbol{\alpha}}$ can be obtained analytically as

$$\widehat{\boldsymbol{\alpha}} = (\boldsymbol{U} + \lambda \boldsymbol{I})^{-1} \widehat{\boldsymbol{v}}.$$

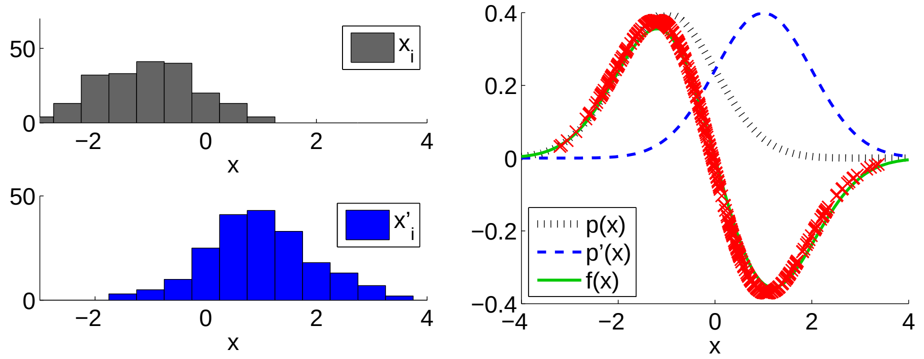
This method is called the *least-squares density-difference* (LSDD) estimator [57]. Tuning parameters such as the regularization parameter λ and basis function $\boldsymbol{\psi}$ can be optimized via cross-validation with respect to J. An example of density-difference estimation by LSDD is illustrated in Figure 8.

If the true density-difference f in Eq.(2) is replaced with the LSDD estimator, we obtain the following L^2-distance estimator:

$$\widehat{\boldsymbol{\alpha}}^\top \boldsymbol{U} \widehat{\boldsymbol{\alpha}}.$$

Similarly, from another expression of the L^2-distance estimator,

$$L^2(p, p') = \int f(\boldsymbol{x}) \Big(p(\boldsymbol{x}) - p'(\boldsymbol{x})\Big) \mathrm{d}\boldsymbol{x},$$



(a) Data (b) Density difference

Figure 8: Illustration of LSDD. "×" in Figure 8(b) denotes an estimated density difference value at $\boldsymbol{x}_i$ and $\boldsymbol{x}'_{i'}$.

we obtain the following L^2-distance estimator:

$$\widehat{\boldsymbol{v}}^\top \widehat{\boldsymbol{\alpha}}.$$

It was shown that the linear combination of these estimators,

$$2\widehat{\boldsymbol{v}}^\top \widehat{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}^\top \boldsymbol{U} \widehat{\boldsymbol{\alpha}},$$

tends to have smaller bias [57], and thus this would be a more reliable L^2-distance estimator in practice.

3.3 Experiments

Here, we use four *UCI benchmark datasets* [5] for experiments, where we randomly choose 10 labeled training samples from each class and 50 unlabeled test samples following true class-prior:

$$\pi^* = 0.1, 0.2, \dots, 0.9.$$

The LSDD method is compared with the following methods:

KDEi: Kernel density estimation (KDE) is used to approximate $p'(\boldsymbol{x})$ and $q_\pi(\boldsymbol{x})$ from data and then the L^2-distance is computed [60]. Two Gaussian widths are *independently* chosen based on 5-fold least-squares cross-validation [24].

KDEj In the KDE-based method, two Gaussian widths are *jointly* chosen based on 5-fold cross-validation in terms of the LSDD criterion [23]. That is, the cross-validated LSDD criterion is computed as a function of two Gaussian widths and the best pair that minimizes the criterion is selected.

EM: The class-prior estimation method based on the expectation-maximization algorithm [42]. This method actually corresponds to distribution matching under the KL divergence.

The left graphs in Figure 9 plot the mean and standard error of the squared difference between true and estimated class-balances π over 1000 runs. These graphs show that LSDD tends to provide better class-balance estimates than alternative approaches.

Next, we use the estimated class balance to train a classifier. We use a weighted ℓ_2-regularized least-squares classifier [41]. That is, a class label $\widehat{y}$ for a test input $\boldsymbol{x}$ is estimated by

$$\widehat{y} = \mathrm{sign}\left(\sum_{\ell=1}^{n} \widehat{\theta}_\ell K(\boldsymbol{x}, \boldsymbol{x}_\ell)\right),$$

where $K(\boldsymbol{x}, \boldsymbol{x}')$ is the Gaussian kernel function with kernel width κ. $\{\widehat{\theta}_\ell\}_{\ell=1}^{n}$ are learned parameters given by

$$(\widehat{\theta}_1, \ldots, \widehat{\theta}_n) := \mathop{\mathrm{argmin}}_{\theta_1,\ldots,\theta_n} \left[\sum_{i=1}^{n} \frac{\pi_{y_i}}{n_{y_i}/n} \left(\sum_{\ell=1}^{n} \theta_\ell K(\boldsymbol{x}_i, \boldsymbol{x}_\ell) - y_i\right)^2 + \delta \sum_{\ell=1}^{n} \theta_\ell^2\right],$$

where $\pi_{+1} = \widehat{\pi}$, $\pi_{-1} = 1 - \widehat{\pi}$, $\widehat{\pi}$ is a class-balance estimate, and δ (≥ 0) is the regularization parameter. The Gaussian width κ and the regularization parameter δ are chosen by 5-fold weighted cross-validation [50] in terms of the misclassification error.

The right graphs in Figure 9 plot the test misclassification error over 1000 runs. The results show the LSDD-based method provides lower classification errors, which would be brought by good estimates of test class-balances.

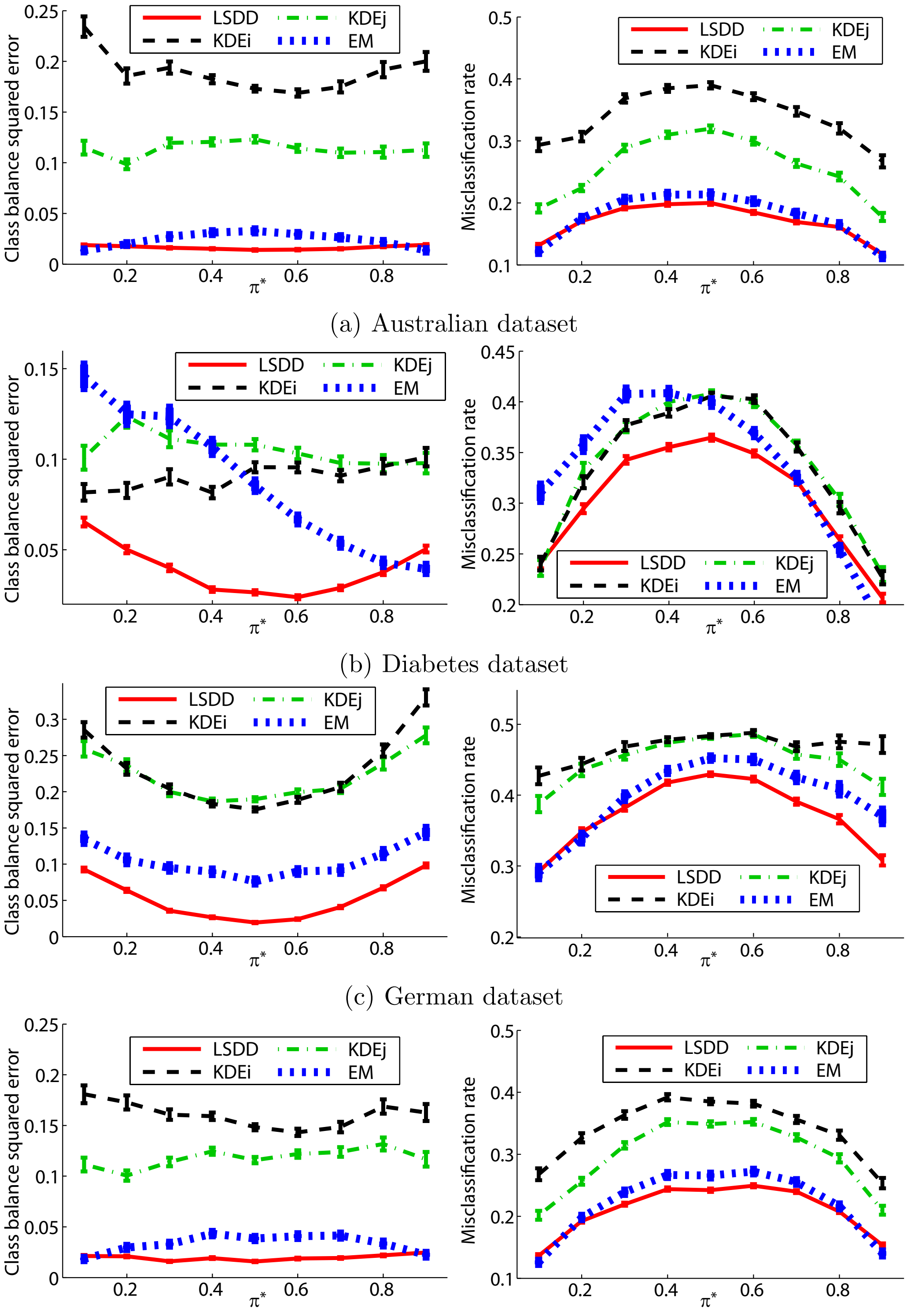
4 Conclusion

In this article, we reviewed semi-supervised adaptive learning techniques for the covariate shift and class-balance change scenarios. In both cases, importance weighting plays an essential role. MATLAB implementations of the algorithms reviewed in this article are available from

“`http://sugiyama-www.cs.titech.ac.jp/~sugi/software/`”.

See [38] for more general discussion on learning under different training and test distributions.

If input-output samples are available from both training and test domains, weighted learning according to the joint importance $p'(\boldsymbol{x}, y)/p(\boldsymbol{x}, y)$ can in principle be used for transferring training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ to the test domain even when $p(\boldsymbol{x}, y)$ and $p'(\boldsymbol{x}, y)$ do not have an explicit link such as the covariate shift and class-balance change [6, 55]. In this situation, not only transferring information from the training domain to the test domain, but also the opposite transfer from the test domain to the training domain is possible simultaneously. This is the idea of *multi-task learning* [12] and is also an important branch of modern machine learning research.



(a) Australian dataset

(b) Diabetes dataset

(c) German dataset

(d) Statlogheart dataset

Figure 9: Results of class-balance adaptation. Left: Squared error of class-balance estimation. Right: Misclassification error by a weighted ℓ_2-regularized least-squares classifier with weighted cross-validation.

Learning from input-output samples has already been studied extensively in statistics and machine learning. However, collecting input-output samples is often expensive and time-consuming in practice. Therefore, learning with side information such as additional input-only samples (semi-supervised learning) and additional related learning tasks (transfer learning and multi-task learning), as well as new models of input-output data collection such as *crowdsourcing* [40] and *self-taught learning* [39], will be important challenges in the arriving *big data* era.

Acknowledgements

MS is supported by MEXT KAKENHI 25700022 and AOARD, MY was supported by the JST PRESTO program and the PLIP program, and MCdP is supported by the MEXT scholarship.

References

[1] A. Agarwal and B. Triggs. Monocular human motion capture with a mixture of regressors. In *Proceedings of IEEE Workshop on Vision for Human Computer Interaction at Computer Vision and Pattern Recognition*, page 72, 2005.

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

[3] T. Akiyama, H. Hachiya, and M. Sugiyama. Efficient exploration through active learning for value function approximation in reinforcement learning. *Neural Networks*, 23(5):639–648, 2010.

[4] N. Anderson, P. Hall, and D. Titterington. Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates. *Journal of Multivariate Analysis*, 50(1):41–54, 1994.

[5] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[6] S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. Multi-task learning for HIV therapy screening. In A. McCallum and S. Roweis, editors, *Proceedings of 25th Annual International Conference on Machine Learning (ICML2008)*, pages 56–63, 2008.

[7] 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 (ICML2007)*, pages 81–88, 2007.

[8] S. Bickel, C. Sawade, and T. Scheffer. Transfer learning by distribution matching for targeted advertising. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 145–152, 2009.

[9] 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*, pages 161–168, Cambridge, MA, 2007. MIT Press.

[10] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA, 2006.

[11] L. Bo and C. Sminchisescu. Twin Gaussian processes for structured prediction. *International Journal of Computer Vision*, 87(1-2):28–52, 2010.

[12] R. Caruana, L. Pratt, and S. Thrun. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[13] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, USA, 2006.

[14] S.-M. Chen, Y.-S. Hsu, and J.-T. Liaw. On kernel estimators of density ratio. *Statistics*, 43(5):463–479, 2009.

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

[16] J. Ćwik and J. Mielniczuk. Estimating density ratio with application to discriminant analysis. *Communications in Statistics: Theory and Methods*, 18(8):3057–3069, 1989.

[17] M. C. du Plessis and M. Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. In J. Langford and J. Pineau, editors, *Proceedings of 29th International Conference on Machine Learning (ICML2012)*, pages 823–830, Edinburgh, Scotland, Jun. 26–Jul. 1 2012.

[18] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. In J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, editors, *Dataset Shift in Machine Learning*, pages 131–160, Cambridge, MA, USA, 2009. MIT Press.

[19] H. Hachiya, T. Akiyama, M. Sugiyama, and J. Peters. Adaptive importance sampling for value function approximation in off-policy reinforcement learning. *Neural Networks*, 22(10):1399–1410, 2009.

[20] H. Hachiya, J. Peters, and M. Sugiyama. Reward weighted regression with sample reuse. *Neural Computation*, 11(23):2798–2832, 2011.

[21] H. Hachiya, M. Sugiyama, and N. Ueda. Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing*, 80:93–101, 2012.

[22] P. Hall. On the non-parametric estimation of mixture proportions. *Journal of the Royal Statistical Society, Series B*, 43(2):147–156, 1981.

[23] P. Hall and M. P. Wand. On nonparametric discrimination using density differences. *Biometrika*, 75(3):541–547, 1988.

[24] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer, Berlin, Germany, 2004.

[25] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NY, USA, 2001.

[26] T. Kanamori. Pool-based active learning with optimal sampling distribution and its information geometrical interpretation. *Neurocomputing*, 71(1–3):353–362, 2007.

[27] T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, Jul. 2009.

[28] T. Kanamori and H. Shimodaira. Active learning algorithm using the maximum weighted log-likelihood estimator. *Journal of Statistical Planning and Inference*, 116(1):149–162, 2003.

[29] T. Kanamori, T. Suzuki, and M. Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367, 2012.

[30] T. Kanamori, T. Suzuki, and M. Sugiyama. Computational complexity of kernel-based density-ratio estimation: A condition number analysis. *Machine Learning*, 90(3):431–460, 2013.

[31] J. Kim and C. Scott. L_2 kernel classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1822–1831, 2010.

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

[33] Y. Li, H. Kambara, Y. Koike, and M. Sugiyama. Application of covariate shift adaptation techniques in brain computer interfaces. *IEEE Transactions on Biomedical Engineering*, 57(6):1318–1324, 2010.

[34] 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*, 56(11):5847–5861, 2010.

[35] 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*, 50(302):157–175, 1900.

[36] J. Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–630, 1998.

[37] Q. Que and M. Belkin. Inverse density as an inverse problem: The fredholm equation approach. Technical Report 1304.5575, arXiv, 2013.

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

[39] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: Transfer learning from unlabeled data. In Zoubin Ghahramani, editor, *Proceedings of the 24th Annual International Conference on Machine Learning (ICML2007)*, pages 759–766. Omnipress, 2007.

[40] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

[41] R. Rifkin, G. Yeo, and T. Poggio. Regularized least-squares classification. In J. A. K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Models and Applications*, volume 190 of *NATO Science Series III: Computer & Systems Sciences*, pages 131–154. IOS Press, Amsterdam, the Netherlands, 2003.

[42] M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

[43] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proceedings of International Conference on Computer Vision (ICCV2003)*, volume 2, pages 750–757, 2003.

[44] 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.

[45] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University, 2006.

[46] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36:111–147, 1974.

[47] M. Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7:141–166, Jan. 2006.

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

[49] M. Sugiyama, M. Kawanabe, and P. L. Chui. Dimensionality reduction for density ratio estimation in high-dimensional spaces. *Neural Networks*, 23(1):44–59, 2010.

[50] 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.

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

[52] M. Sugiyama and S. Nakajima. Pool-based active learning in approximate linear regression. *Machine Learning*, 75(3):249–274, 2009.

[53] M. Sugiyama and H. Ogawa. Subspace information criterion for model selection. *Neural Computation*, 13(8):1863–1889, 2001.

[54] M. Sugiyama and N. Rubens. A batch ensemble approach to active learning with model selection. *Neural Networks*, 21(9):1278–1286, 2008.

[55] M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, Cambridge, UK, 2012.

[56] M. Sugiyama, T. Suzuki, and T. Kanamori. Density ratio matching under the Bregman divergence: A unified framework of density ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012.

[57] M. Sugiyama, T. Suzuki, T. Kanamori, M. C. du Plessis, S. Liu, and I. Takeuchi. Density-difference estimation. *Neural Computation*, 2013. to appear.

[58] 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, 2008.

[59] M. Sugiyama, M. Yamada, P. von Bünau, T. Suzuki, T. Kanamori, and M. Kawanabe. Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 24(2):183–198, 2011.

[60] D. M. Titterington. Minimum distance non-parametric estimation of mixture proportions. *Journal of the Royal Statistical Society, Series B*, 45(1):37–46, 1983.

[61] Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. *Journal of Information Processing*, 17:138–155, 2009.

[62] K. Ueki, M. Sugiyama, and Y. Ihara. Lighting condition adaptation for perceived age estimation. *IEICE Transactions on Information and Systems*, E94-D(2):392–395, 2011.

[63] V. N. Vapnik. *Statistical Learning Theory.* Wiley, New York, NY, USA, 1998.

[64] V. N. Vapnik, I. Braga, and R. Izmailov. Constructive setting of the density ratio estimation problem and its rigorous solution. Technical Report 1306.0407, arXiv, 2013.

[65] D. P. Wiens. Robust weights and designs for biased regression models: Least squares and generalized M-estimation. *Journal of Statistical Planning and Inference*, 83(2):395–412, 2000.

[66] M. Yamada, L. Sigal, and M. Raptis. No bias left behind: Covariate shift adaptation for discriminative 3D pose estimation. In *Proceedings of European Conference on Computer Vision (ECCV2012)*, pages 674–687, 2012.

[67] M. Yamada and M. Sugiyama. Direct importance estimation with Gaussian mixture models. *IEICE Transactions on Information and Systems*, E92-D(10):2159–2162, 2009.

[68] M. Yamada and M. Sugiyama. Direct density-ratio estimation with dimensionality reduction via hetero-distributional subspace analysis. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI2011)*, pages 549–554, San Francisco, California, USA, Aug. 7–11 2011. The AAAI Press.

[69] M. Yamada, M. Sugiyama, and T. Matsui. Semi-supervised speaker identification under covariate shift. *Signal Processing*, 90(8):2353–2361, 2010.

[70] M. Yamada, M. Sugiyama, G. Wichern, and J. Simm. Direct importance estimation with a mixture of probabilistic principal component analyzers. *IEICE Transactions on Information and Systems*, E93-D(10):2846–2849, 2010.

[71] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama. Relative density-ratio estimation for robust distribution comparison. *Neural Computation*, 25(5):1324–1370, 2013.

[72] T. Zhao, H. Hachiya, V. Tangkaratt, J. Morimoto, and M. Sugiyama. Efficient sample reuse in policy gradients with parameter-based exploration. *Neural Computation*, 25(6):1512–1547, 2013.