Importance-Weighted Least-Squares Probabilistic Classifier for Covariate Shift Adaptation with Application to Human Activity Recognition

Hirotaka Hachiya
Tokyo Institute of Technology, Tokyo, Japan
hachiya@sg.cs.titech.ac.jp

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

Naonori Ueda
NTT Communication Science Laboratories, Kyoto, Japan
ueda@cslab.kecl.ntt.co.jp

Abstract

Human activity recognition from accelerometric data (e.g., obtained by smart phones) is gathering a great deal of attention since it can be used for various purposes such as remote health-care. However, since collecting labeled data is bothersome for new users, it is desirable to utilize data obtained from existing users. In this paper, we formulate this adaptation problem as learning under covariate shift, and propose a computationally efficient probabilistic classification method based on adaptive importance sampling. The usefulness of the proposed method is demonstrated in real-world human activity recognition.

1 Introduction

Human activity recognition from accelerometric data (e.g., obtained by smart phones) is gathering a great deal of attention recently [1, 2, 11], since it can be used for various purposes such as *remote health-care* [10, 25, 17] and *worker behavior monitoring* [34]. To construct a good classifier for activity recognition, users are required to prepare accelerometric data with activity labels for various types of actions such as walking, running, and bicycle riding. However, since gathering labeled data is costly, this initial data-collection phase prevents new users from using the activity recognition system. Thus, overcoming such a new user problem is an important challenge for increasing the practical usability of the human activity recognition system.

Since unlabeled data are relatively easy to gather, we can typically use labeled data obtained from existing users and unlabeled data obtained from a new user for developing the new user's activity classifier. Such a situation is commonly called *semi-supervised learning*, and various learning methods that utilize unlabeled samples have been proposed so far [5]. However, such semi-supervised learning methods tend to perform poorly if unlabeled test data have a significantly different distribution from the labeled training data. Unfortunately, this is a typical situation in human activity recognition since motion patterns (and thus distributions of motion data) depend heavily on users.

To cope with the differing distributions, several approaches have been explored [21, 19]. Popular lines of research include re-weighting samples according to the importance [23, 12, 30, 13, 6, 27] and learning feature representation that is common to training and test data [4, 8, 3, 9, 18]. In this paper, we focus on the sample re-weighting approach, and propose a new probabilistic classification method that is computationally very efficient. Our proposed approach combines a probabilistic classification method called *least-squares probabilistic classifier* [26, 33] with the sample re-weighting approach [23, 13]. Through experiments on real-world human activity recognition, we demonstrate the usefulness of our proposed approach.

The rest of this paper is organized as follows. In Section 2, we formulate a classification problem, and describe our proposed method called the *importance-weighted least-squares probabilistic classifier* (IWLSPC). In Section 3, we discuss the relation between proposed and existing approaches. Experimental results are reported in Section 4, demonstrating the effectiveness of the proposed IWLSPC algorithm in real-world human activity recognition. Finally, we conclude in Section 5 by summarizing our contributions.

2 Learning under Covariate Shift

In this paper, we consider the classification problem under *covariate shift* [23, 27], i.e., the distributions of input points change between the training and test phases, but the conditional distribution of class labels given input points remains unchanged. In this section, we first formulate the classification problem under covariate shift, and then give our proposed method called the *importance-weighted least-squares probabilistic classifier* (IWLSPC).

2.1 Problem Formulation

Suppose we are given labeled training samples $\{(\boldsymbol{x}_n^{\mathrm{tr}}, y_n^{\mathrm{tr}})\}_{n=1}^{N_{\mathrm{tr}}}$, where $\boldsymbol{x}_n^{\mathrm{tr}} \in \mathbb{R}^d$ (d denotes the input dimensionality) is a training input point drawn independently from a probability distribution with density $p_{\mathrm{tr}}(\boldsymbol{x})$, and $y_n^{\mathrm{tr}} \in \{1, \dots, c\}$ (c denotes the number of classes) is a training label following a conditional probability distribution with density $p(y|\boldsymbol{x} = \boldsymbol{x}_n^{\mathrm{tr}})$.

In addition to the labeled training samples, suppose we are given unlabeled test input points $\{\boldsymbol{x}_n^{\mathrm{te}}\}_{n=1}^{N_{\mathrm{te}}}$, where $\boldsymbol{x}_n^{\mathrm{te}}$ $(\in \mathbb{R}^d)$ is a test input point drawn independently from a probability distribution with density $p_{\mathrm{te}}(\boldsymbol{x})$. Note that $p_{\mathrm{te}}(\boldsymbol{x}) \neq p_{\mathrm{tr}}(\boldsymbol{x})$ in general, and thus the input distributions are different between the training and test phases.

Our goal is to learn a classifier that predicts a class label y^{te} for a test input point $\boldsymbol{x}^{\mathrm{te}}$.

2.2 Importance-Weighted Least-Squares Probabilistic Classifier

Here, we describe our proposed method called the *importance-weighted least-squares probabilistic classifier* (IWLSPC), which combines a probabilistic classification method called *least-squares probabilistic classifier* [26, 33] with the covariate shift adaptation technique [23, 13].

The goal of probabilistic classification is to estimate the class-posterior probability $p(y|\boldsymbol{x})$. Let us model the class-posterior probability $p(y|\boldsymbol{x})$ by

$$p(y|\boldsymbol{x};\boldsymbol{\theta}_y) \equiv \sum_{n=1}^{N_{\mathrm{te}}} \theta_{y,n} K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}),$$

where $\boldsymbol{\theta}_y = (\theta_{y,1}, \ldots, \theta_{y,N_{\mathrm{te}}})^\top$ is the parameter vector and $K(\boldsymbol{x}, \boldsymbol{x}')$ is a kernel function. Below, we focus on the Gaussian kernel:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right),$$

where σ denotes the Gaussian kernel width. We determine the parameter $\boldsymbol{\theta}_y$ so that the following squared error J_y is minimized:

$$\begin{aligned}
J_y(\boldsymbol{\theta}_y) &\equiv \frac{1}{2}\int \Big(p(y|\boldsymbol{x};\boldsymbol{\theta}_y) - p(y|\boldsymbol{x})\Big)^2 p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\
&= \frac{1}{2}\int p(y|\boldsymbol{x};\boldsymbol{\theta}_y)^2 p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\
&\quad - \int p(y|\boldsymbol{x};\boldsymbol{\theta}_y) p(y|\boldsymbol{x}) p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + C \\
&= \frac{1}{2}\boldsymbol{\theta}_y^\top \boldsymbol{Q}\boldsymbol{\theta}_y - \boldsymbol{q}_y^\top \boldsymbol{\theta}_y + C,
\end{aligned}$$

where C is a constant independent of the parameter $\boldsymbol{\theta}_y$, and $\boldsymbol{Q}$ is the $N_{\mathrm{te}} \times N_{\mathrm{te}}$ matrix and $\boldsymbol{q}_y = (q_{y,1}, \ldots, q_{y,N_{\mathrm{te}}})^\top$ is the N_{te}-dimensional vector defined as

$$\begin{aligned}
Q_{n,n'} &\equiv \int K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) K(\boldsymbol{x}, \boldsymbol{x}_{n'}^{\mathrm{te}}) p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \\
q_{y,n} &\equiv \int K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) p(y|\boldsymbol{x}) p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}.
\end{aligned}$$

Here, we approximate $\boldsymbol{Q}$ and $\boldsymbol{q}_y$ using the *adaptive importance sampling* technique [23] as follows. First, using the *importance weight* defined as

$$w(\boldsymbol{x}) \equiv \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}, \tag{1}$$

we express $\boldsymbol{Q}$ and $\boldsymbol{q}_y$ in terms of the training distribution as

$$\begin{aligned} Q_{n,n'} &= \int K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) K(\boldsymbol{x}, \boldsymbol{x}_{n'}^{\mathrm{te}}) p_{\mathrm{tr}}(\boldsymbol{x}) w(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \\ q_{y,n} &= \int K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) p(y|\boldsymbol{x}) p_{\mathrm{tr}}(\boldsymbol{x}) w(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ &= p(y) \int K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) p_{\mathrm{tr}}(\boldsymbol{x}|y) w(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \end{aligned}$$

where $p_{\mathrm{tr}}(\boldsymbol{x}|y)$ denotes the training input density for class y. Then, based on the above expressions, $\boldsymbol{Q}$ and $\boldsymbol{q}_y$ are approximated using the training samples $\{(\boldsymbol{x}_n^{\mathrm{tr}}, y_n^{\mathrm{tr}})\}_{n=1}^{N_{\mathrm{tr}}}$ as follows[1]:

$$\begin{aligned} \widehat{Q}_{n,n'} &\equiv \frac{1}{N_{\mathrm{tr}}} \sum_{n''=1}^{N_{\mathrm{tr}}} K(\boldsymbol{x}_{n''}^{\mathrm{tr}}, \boldsymbol{x}_n^{\mathrm{te}}) K(\boldsymbol{x}_{n''}^{\mathrm{te}}, \boldsymbol{x}_{n'}^{\mathrm{te}}) w(\boldsymbol{x}_{n''}^{\mathrm{tr}})^{\nu}, \\ \widehat{q}_{y,n} &\equiv \frac{1}{N_{\mathrm{tr}}} \sum_{n':y_{n'}^{\mathrm{tr}}=y} K(\boldsymbol{x}_{n'}^{\mathrm{tr}}, \boldsymbol{x}_n^{\mathrm{te}}) w(\boldsymbol{x}_{n'}^{\mathrm{tr}})^{\nu}, \end{aligned}$$

where the class-prior probability $p(y)$ was estimated by $N_{\mathrm{tr}}^{(y)}/N_{\mathrm{tr}}$, and $N_{\mathrm{tr}}^{(y)}$ denotes the number of training samples with label y. ν $(0 \leq \nu \leq 1)$ is the *flattening parameter*, which controls the bias-variance trade-off in importance sampling [23]. More specifically, if ν is close to 1, the importance weights are used as they are; then the bias gets smaller, but the variance tends to be larger. On the other hand, if ν is close to 0, the importance weights tend to be one (i.e., flat). Then the bias is larger, but the variance is smaller.

Consequently, we arrive at the following optimization problem:

$$\widehat{\boldsymbol{\theta}}_y \equiv \mathop{\mathrm{arg\,min}}_{\boldsymbol{\theta}_y} \left[\frac{1}{2} \boldsymbol{\theta}_y^{\top} \widehat{\boldsymbol{Q}} \boldsymbol{\theta}_y - \widehat{\boldsymbol{q}}_y^{\top} \boldsymbol{\theta}_y + \frac{\lambda}{2} \boldsymbol{\theta}_y^{\top} \boldsymbol{\theta}_y \right],$$

where $\frac{\lambda}{2}\boldsymbol{\theta}_y^{\top}\boldsymbol{\theta}_y$ is a regularization term to avoid over-fitting and λ (≥ 0) is the regularization parameter. Then, the IWLSPC solution is given *analytically* as

$$\widehat{\boldsymbol{\theta}}_y = (\widehat{\boldsymbol{Q}} + \lambda \boldsymbol{I}_{N_{\mathrm{te}}})^{-1} \widehat{\boldsymbol{q}}_y,$$

[1] When $\nu = 1$, $\boldsymbol{Q}$ may be approximated directly using the test input samples $\{\boldsymbol{x}_n^{\mathrm{te}}\}_{n=1}^{N_{\mathrm{te}}}$ as

$$\widehat{Q}_{n,n'} \equiv \frac{1}{N_{\mathrm{te}}} \sum_{n''=1}^{N_{\mathrm{te}}} K(\boldsymbol{x}_{n''}^{\mathrm{te}}, \boldsymbol{x}_n^{\mathrm{te}}) K(\boldsymbol{x}_{n''}^{\mathrm{te}}, \boldsymbol{x}_{n'}^{\mathrm{te}}).$$

where $\boldsymbol{I}_{N_{\mathrm{te}}}$ denotes the N_{te}-dimensional identity matrix. Since the class-posterior probability is non-negative by definition, we modify the solution as

$$\widehat{p}(y|\boldsymbol{x}) \equiv \frac{1}{Z} \max\left(0, \sum_{n=1}^{N_{\mathrm{te}}} \widehat{\theta}_{y,n} K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}})\right),$$

if $Z \equiv \sum_{y=1}^{c} \max\left(0, \sum_{n=1}^{N_{\mathrm{te}}} \widehat{\theta}_{y,n} K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}})\right) > 0$; otherwise, $\widehat{p}(y|\boldsymbol{x}) \equiv 1/c$.

The learned class-posterior probability $\widehat{p}(y|\boldsymbol{x})$ allows us to predict the class label y^{te} of a new test sample $\boldsymbol{x}^{\mathrm{te}}$ by

$$\widehat{y}^{\mathrm{te}} \equiv \arg\max_{y} \widehat{p}(y|\boldsymbol{x}^{\mathrm{te}}),$$

with *confidence* $\widehat{p}(\widehat{y}^{\mathrm{te}}|\boldsymbol{x}^{\mathrm{te}})$.

When $p_{\mathrm{te}}(\boldsymbol{x}) = p_{\mathrm{tr}}(\boldsymbol{x})$, the above IWLSPC is reduced to the original LSPC [26, 33]. Thus, IWLSPC can be regarded as a natural extension of LSPC to covariate shift situations.

The performance of IWLSPC depends on the choice of the regularization parameter λ, the Gaussian kernel width σ, and the flattening parameter ν. Thus, model selection is critical in practice, and *cross-validation* (CV) is a popular choice for this purpose. However, under covariate shift, ordinary CV is highly biased due to the differing distributions. To cope with this problem, a variant of CV called *importance-weighted CV* (IWCV) has been proposed [28], which is still almost unbiased even under covariate shift. We will use IWCV for model selection of IWLSPC in our experiments. A brief review of IWCV is provided in A.

IWLSPC (and IWCV) requires the values of the importance $\{w(\boldsymbol{x}_n^{\mathrm{tr}})\}_{n=1}^{N_{\mathrm{tr}}}$, which are unknown in practice. So far, various methods for estimating the importance has been developed [24, 7, 20, 12, 30, 13, 29]. Among them, the method called *unconstrained least-squares importance fitting* (uLSIF) [13] was shown to be superior since it achieves the optimal convergence rate [14], it possesses the optimal numerical stability [15], and its solution can be computed analytically. For this reason, we will employ uLSIF in our experiments. A brief review of uLSIF is provided in B.

A MATLAB implementation of IWLSPC is available from

‘`http://sugiyama-www.cs.titech.ac.jp/~hachiya/software/IWLSPC/`’.

3 Related Works

In this section, we review two related works: *Laplacian regularized least-squares* (LapRLS) and *importance-weighted kernel logistic regression* (IWKLR). These two methods will be regarded as baselines for experimental performance comparison in Section 4.

3.1 Laplacian Regularized Least-Squares

Laplacian regularized least-squares (LapRLS) is a standard semi-supervised learning method which tries to impose smoothness over non-linear data manifold [5]. Let us consider a binary classification problem where $y \in \{+1, -1\}$. LapRLS uses a kernel model for class prediction[2]:

$$k(\boldsymbol{x}; \boldsymbol{\theta}) \equiv \sum_{n=1}^{N_{\mathrm{te}}} \theta_n K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}).$$

The parameter $\boldsymbol{\theta}$ is determined as

$$\min_{\boldsymbol{\theta}} \left[\frac{1}{N_{\mathrm{tr}}} \sum_{n=1}^{N_{\mathrm{tr}}} \left(k(\boldsymbol{x}_n^{\mathrm{tr}}; \boldsymbol{\theta}) - y_n^{\mathrm{tr}} \right)^2 + \lambda \boldsymbol{\theta}^\top \boldsymbol{\theta} + \eta \sum_{n,n'=1}^{N} \boldsymbol{L}_{n,n'} k(\boldsymbol{x}_n; \boldsymbol{\theta}) k(\boldsymbol{x}_{n'}; \boldsymbol{\theta}) \right],$$

where the first term is the goodness of fit, the second term is the ℓ_2-regularizer to avoid over-fitting, and the third term is the Laplacian regularizer to impose smoothness over data manifold. $N \equiv N_{\mathrm{tr}} + N_{\mathrm{te}}$, $\boldsymbol{L} \equiv \boldsymbol{D} - \boldsymbol{W}$ is the $N \times N$ graph Laplacian matrix, $\boldsymbol{W}$ is an affinity matrix defined by

$$W_{n,n'} \equiv \exp\left(-\frac{\|\boldsymbol{x}_n - \boldsymbol{x}_{n'}\|^2}{2\tau^2} \right),$$
$$(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \equiv (\boldsymbol{x}_1^{\mathrm{tr}}, \ldots, \boldsymbol{x}_{N_{\mathrm{tr}}}^{\mathrm{tr}}, \boldsymbol{x}_1^{\mathrm{te}}, \ldots, \boldsymbol{x}_{N_{\mathrm{te}}}^{\mathrm{te}}),$$

τ is an affinity-controlling parameter, and $\boldsymbol{D}$ is the diagonal matrix given by $\boldsymbol{D}_{n,n} \equiv \sum_{n'=1}^{N} \boldsymbol{W}_{n,n'}$.

The solution of LapRLS can be analytically computed since the optimization problem is an unconstrained quadratic program. However, covariate shift is not taken into account in LapRLS, and thus it will not perform well if training and test distributions are significantly different.

3.2 Importance-Weighted Kernel Logistic Regression

Kernel logistic regression (KLR) is a popular probabilistic classifier, which learns the class-posterior probability $p(y|x)$ by a kernel logistic model:

$$p(y|\boldsymbol{x}; \boldsymbol{\theta}) \equiv \frac{\exp\left(\sum_{n=1}^{N_{\mathrm{te}}} \theta_{y,n} K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) \right)}{\sum_{y'=1}^{c} \exp\left(\sum_{n=1}^{N_{\mathrm{te}}} \theta_{y',n} K(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) \right)},$$

where $\boldsymbol{\theta} = (\theta_{1,1}, \ldots, \theta_{c,N_{\mathrm{te}}})^\top$. The importance-weighted KLR (IWKLR) method determines the parameter $\boldsymbol{\theta}$ by maximizing the penalized importance-weighted log-likelihood

[2] Note that, in the original LapRLS, kernel basis functions are located at training input points $\{\boldsymbol{x}_n^{\mathrm{tr}}\}_{n=1}^{N_{\mathrm{tr}}}$. Here, we locate them at test input points $\{\boldsymbol{x}_n^{\mathrm{te}}\}_{n=1}^{N_{\mathrm{te}}}$ for being consistent with other methods.

[32, 35]:

$$\max_{\boldsymbol{\theta}} \left[\sum_{n=1}^{N_{\mathrm{tr}}} w(\boldsymbol{x}_n^{\mathrm{tr}})^{\nu} \log p(y_n^{\mathrm{tr}} | \boldsymbol{x}_n^{\mathrm{tr}}; \boldsymbol{\theta}) - \lambda \boldsymbol{\theta}^{\top} \boldsymbol{\theta} \right].$$

The above objective function is concave with respect to $\boldsymbol{\theta}$, and thus the unique maximizer can be obtained by standard optimization techniques such as gradient methods and (quasi-) Newton methods. Although sophisticated optimization toolboxes are readily available, training a large-scale IWKLR model is still computationally challenging since it requires to optimize the all-class parameter $\boldsymbol{\theta}$ of dimension $c \times N_{\mathrm{te}}$ at once. On the other hand, IWLSPC optimizes the class-wise parameter $\boldsymbol{\theta}_y$ of dimension N_{te} separately c times.

4 Experiments

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

4.1 Setup

We compare the performance of the following six classification methods:

- LapRLS+IWCV: a semi-supervised learning method described in Section 3.1. The hyper-parameters σ, τ, λ, and η are selected by IWCV (see A).

- LapRLS+CV: a semi-supervised learning method described in Section 3.1. The hyper-parameters σ, τ, λ, and η are selected by ordinary CV (i.e., no importance weighting).

- IWKLR+IWCV: a probabilistic classifier described in Section 3.2. The hyper-parameters σ, λ, and ν are selected by IWCV. We use a MATLAB implementation of a limited-memory BFGS (Broyden-Fletcher-Goldfarb-Shanno) quasi-Newton method included in the *minFunc* package [22] for optimization.

- KLR+CV: IWKLR+IWCV with all importance weights set to 1, i.e., $p_{\mathrm{tr}}(\boldsymbol{x}) = p_{\mathrm{te}}(\boldsymbol{x})$ is assumed.

- IWLSPC+IWCV: our proposed method described in Section 2.2. The hyper-parameters σ, λ, and ν are selected by IWCV.

- LSPC+CV: IWLSPC+IWCV with all importance weights set to 1, i.e., $p_{\mathrm{tr}}(\boldsymbol{x}) = p_{\mathrm{te}}(\boldsymbol{x})$ is assumed.

Importance weights are estimated by uLSIF (see B), and the number of kernel basis functions is fixed to 200 by random sampling from N_{te} kernels. Training and test input

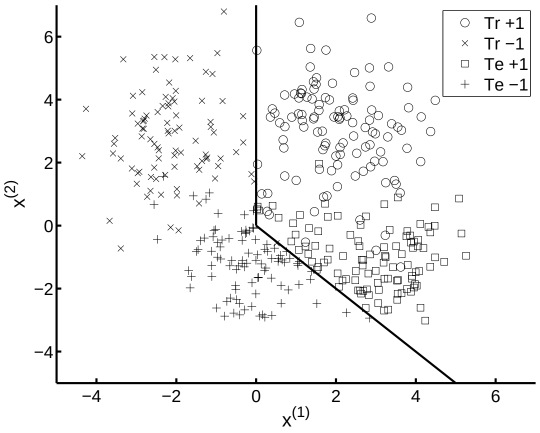


Figure 1: Toy data. The optimal decision boundary is described by the solid line. '○' and '×' denote positive and negative training samples, while '□' and '+' denote positive and negative test samples. Note that the test samples are treated as unlabeled when training a classifier, and their labels are used only for evaluating the test performance.

samples are normalized in the element-wise manner so that each element has mean zero and unit variance. The hyper-parameters are chosen from

$$\begin{aligned}\sigma, \tau &\in \{0.1m, 0.2m, 0.5m, m, 2m, 3m\},\\ \lambda, \eta &\in \{10^{-2}, 10^{-1.5}, 10^{-1}, 10^{-0.5}, 10^{0}\},\\ \nu &\in \{0, 0.2, 0.4, 0.6, 0.8, 1\},\end{aligned}$$

where m is the median distance among test input samples:

$$m \equiv \mathrm{median}\Big(\{\|\boldsymbol{x}_n^{\mathrm{te}} - \boldsymbol{x}_{n'}^{\mathrm{te}}\|\}_{n,n'=1}^{N_{\mathrm{te}}}\Big).$$

The configuration of the computer used in experiments is as follows: *Ubuntu 9.10, Intel(R) Xeon(R) 2.93GHz, 2 CPUs and 24GB RAM.*

4.2 Toy Classification Problem

Let us consider a binary classification problem on the two-dimensional input space. We define the class-posterior probabilities given input $\boldsymbol{x}$ by

$$\begin{aligned}p(y = +1|\boldsymbol{x}) &= \frac{1}{2} + \frac{1}{2}\tanh\Big(x^{(1)} + \min(0, x^{(2)})\Big),\\ p(y = -1|\boldsymbol{x}) &= 1 - p(y = +1|\boldsymbol{x}),\end{aligned}$$

where $\boldsymbol{x} = (x^{(1)}, x^{(2)})^\top$. The optimal decision boundary is the set of all $\boldsymbol{x}$ such that

$$p(y = +1|\boldsymbol{x}) = p(y = -1|\boldsymbol{x}) = \frac{1}{2}.$$

Table 1: Mean misclassification rates [%] and standard deviation (in parentheses) averaged over 50 trials for toy data with 2 dimensional input space. A number in bold face indicates the fact that the method is the best or comparable to the best one in terms of the mean misclassification rate by the *t-test* at the significance level 5%.

N_{tr}	LapRLS +CV	LapRLS +IWCV	KLR +CV	IWKLR +IWCV	LSPC +CV	IWLSPC +IWCV
1000	18.6(6.4)	**16.5**(6.8)	17.5(3.7)	**15.2**(5.4)	16.6(3.4)	**15.2**(5.5)
2000	17.6(6.3)	15.7(7.0)	17.7(4.1)	**14.1**(6.3)	17.2(3.1)	**13.3**(5.8)

Table 2: Mean computation time [s] and standard deviation (in parentheses) averaged over 50 trials for toy data with 2 dimensional input space. A number in bold face indicates the fact that the method is the best or comparable to the best one in terms of the mean computation time by the *t-test* at the significance level 5%.

N_{tr}	LapRLS +CV	LapRLS +IWCV	KLR +CV	IWKLR +IWCV	LSPC +CV	IWLSPC +IWCV
1000	2.5(0.6)	3.5(2.1)	3.5(1.1)	3.0(1.1)	**0.6**(0.2)	**0.5**(0.3)
2000	4.2(0.9)	4.9(3.0)	9.1(2.4)	7.6(2.3)	**1.2**(0.3)	**1.3**(0.4)

Let the training and test input densities be

$$p_{\mathrm{tr}}(\boldsymbol{x}) = \frac{1}{2}N\left(\boldsymbol{x}; \begin{bmatrix}-2\\3\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 2\end{bmatrix}\right) + \frac{1}{2}N\left(\boldsymbol{x}; \begin{bmatrix}2\\3\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 2\end{bmatrix}\right),$$

$$p_{\mathrm{te}}(\boldsymbol{x}) = \frac{1}{2}N\left(\boldsymbol{x}; \begin{bmatrix}0\\-1\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right) + \frac{1}{2}N\left(\boldsymbol{x}; \begin{bmatrix}3\\-1\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right),$$

where $N(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate normal density with respect to $\boldsymbol{x}$, with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

We create training input points $\{\boldsymbol{x}_n^{\mathrm{tr}}\}_{n=1}^{N_{\mathrm{tr}}}$ following $p_{\mathrm{tr}}(\boldsymbol{x})$ and training labels $\{y_n^{\mathrm{tr}}\}_{n=1}^{N_{\mathrm{tr}}}$ following $p(y|\boldsymbol{x}_n^{\mathrm{tr}})$. Similarly, we create $N_{\mathrm{te}} = 1000$ test input points $\{\boldsymbol{x}_n^{\mathrm{te}}\}_{n=1}^{N_{\mathrm{te}}}$ following $p_{\mathrm{te}}(\boldsymbol{x})$ and test labels $\{y_n^{\mathrm{te}}\}_{n=1}^{N_{\mathrm{te}}}$ following $p(y|\boldsymbol{x}_n^{\mathrm{te}})$. Examples of training and test samples, and the optimal decision boundary are illustrated in Figure 1. The graph shows that the optimal decision boundary sharply turns at the origin. This is an extrapolation problem where training samples are distributed in the upper half of the graph and test samples are distributed in the lower half. If the decision boundary is naively estimated from training data, test data may not be classified correctly. This is a typical example of covariate shift.

The experiments are repeated 50 times with different random seeds. Table 1 depicts the experimental results for $N_{\mathrm{tr}} = 1000$ and 2000. The table shows that the use of importance weights highly contributes to reducing the misclassification rate for KLR and LSPC. Table 2 depicts the computation time for training classifiers. The table shows that the LSPC-based methods are computationally much more efficient than the KLR-based methods.

Table 3: Mean misclassification rates [%] and standard deviation (in parentheses) averaged over 50 trials for toy data with 10 dimensional input space. A number in bold face indicates the fact that the method is the best or comparable to the best one in terms of the mean misclassification rate by the *t-test* at the significance level 5%.

N_{tr}	LSPC+CV	IWLSPC+IWCV
1000	**20.3**(2.3)	**20.3**(3.7)
2000	**20.1**(2.5)	**19.7**(3.6)
4000	20.0(1.8)	**19.4**(2.8)
6000	19.9(1.5)	**19.3**(1.9)
8000	19.5(1.5)	**18.6**(2.0)
10000	19.3(1.8)	**18.4**(1.9)

To evaluate the performance of the importance-weighting approach in high dimensional input spaces, we added eight dimensions to the above 2-dimensional input space—the extra dimensional data $(x^{(3)}, x^{(4)}, \ldots, x^{(10)})^{\top}$ follow independently normal distribution. Table 3 depicts the misclassification rate for LSPC+CV and IWLSPC+IWCV in various numbers N_{tr} of training samples. The table shows that IWLSPC+IWCV significantly outperforms LSPC+CV specifically when the number of training samples is large ($N_{\mathrm{tr}} \geq 4000$). This implies that importance-weighting can perform well even in high dimensional data.

4.3 Human Activity Recognition using Accelerometric Data

Next, we apply the proposed method to real-world human activity recognition. We use three-axis accelerometric data collected by *iPodTouch* available from `http://alkan.mns.kyutech.ac.jp/web/data.html`. In the data collection procedure, subjects were asked to perform a specific task such as walking, running, and bicycle riding. The duration of each task was arbitrary and the sampling rate was 20 Hz with small variations. An example of three-axis accelerometric data for "walking" is plotted in Figure 2.

To extract features from the accelerometric data, each data stream was segmented in a sliding window manner with window width 5 seconds and sliding step 1 second. Depending on subjects, the position and orientation of iPodTouch was arbitrary—held by hand or kept in a pocket or a bag. For this reason, we decided to take the ℓ_2-norm of the 3-dimensional acceleration vector at each time step, and computed the following 5 orientation-invariant features from each window: *mean*, *standard deviation*, *fluctuation of amplitude*, *average energy* and *frequency-domain entropy* [1, 2].

Let us consider a situation where 2 new users (indicated by u1 and u2) want to use the activity recognition system. However, since the new users do not want to label their accelerometric data, there is no labeled sample for the new users. On the other hand, a large number of unlabeled samples for the new user and a large number of labeled data obtained from existing users are available. Let labeled training data $\{(\boldsymbol{x}_n^{\mathrm{tr}}, y_n^{\mathrm{tr}})\}_{n=1}^{N_{\mathrm{tr}}}$ be the set of labeled accelerometric data for 20 existing users. Each user has at most 100

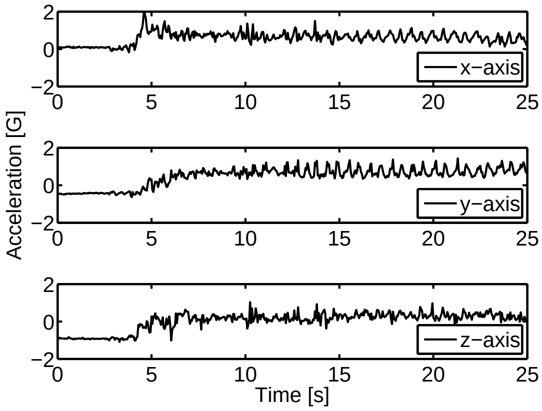


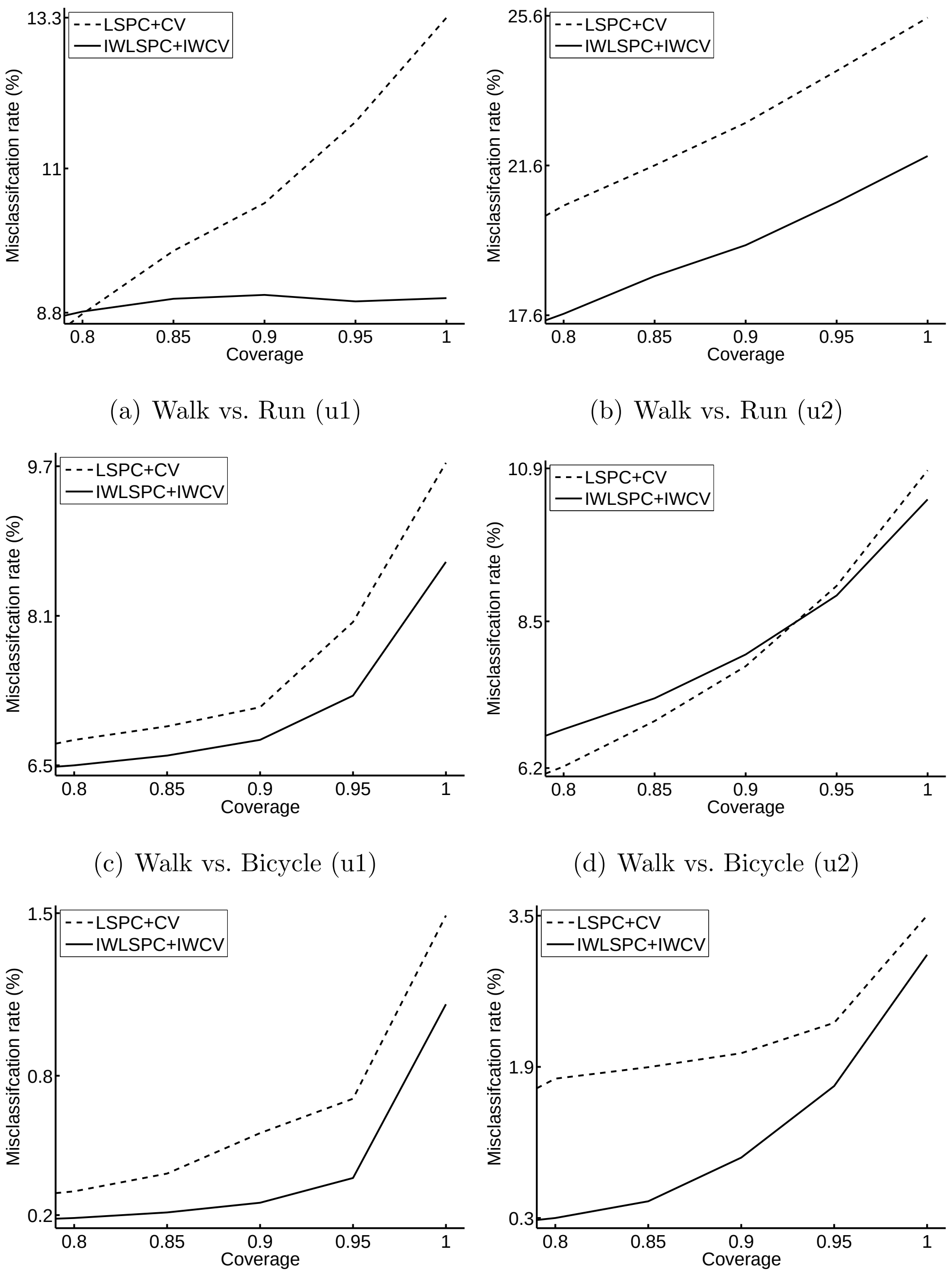
Figure 2: An example of three-axis accelerometric data for "walking".

labeled samples for each action. Let unlabeled test data $\{\boldsymbol{x}_n^{\mathrm{te}}\}_{n=1}^{N_{\mathrm{te}}}$ be $N_{\mathrm{te}} = 800$ unlabeled accelerometric data obtained from a new user. For stabilization purposes, we average out the importance weights over each existing user in IWCV.

The experiment is repeated 50 times with different sample choice. Table 4 depicts the experimental results for each new user (u1 and u2) in three binary classification tasks: *walk vs. run*, *walk vs. riding a bicycle*, *walk vs. taking a train*. The table shows that IWLSPC+IWCV and IWKLR+IWCV compare favorably with other methods in terms of the classification accuracy. Table 6 depicts the confusion matrices of three binary classification tasks using IWLSPC+IWCV. The table shows that the number of corrected predictions for each task is rather imbalanced between two activities, e.g., there are more corrected predictions for "walking" than for "taking a train". Table 5 depicts the computation time for training classifiers. The table shows that the LSPC-based methods are computationally much more efficient than the KLR-based methods.

Figure 3 depicts the mean misclassification rate for various *coverage* level, which is the ratio of test sample size used for evaluating the misclassification rate. For example, the coverage 0.8 means that 80% of test samples with high confidence level (obtained by an estimated class-posterior probability) are used for evaluating the misclassification rate. The graphs show that, for most of the coverage level, IWLSPC+IWCV outperforms LSPC+CV and the misclassification rate of both IWLSPC+IWCV and LSPC+CV tends to decrease as the coverage level decreases. This implies that the confidence estimation by (IW)LSPC is reliable since the higher the confidence is, the more accurate the learned classifier. This fact leads to a good heuristic to further improve the performance. That is, the prediction with low confidence level is rejected and instead the prediction with high confidence level obtained in the previous time step is used since an action usually continues for a certain amount of duration.

Table 7 depicts the mean misclassification rate for the case that the prediction with lower confidence level than a threshold, i.e., 0.8 or 0.9 is replaced with the previous one with the highest confidence level in last five steps. The table shows that the correction based on the confidence-level can further improve the performance of IWLSPC+IWCV.



(a) Walk vs. Run (u1)

(b) Walk vs. Run (u2)

(c) Walk vs. Bicycle (u1)

(d) Walk vs. Bicycle (u2)

(e) Walk vs. Train (u1)

(f) Walk vs. Train (u2)

Figure 3: Misclassification rate as a function of coverage for each new user (u1 and u2) in human activity recognition.

Table 4: Mean misclassification rates [%] and standard deviation (in parentheses) averaged over 50 trials for each new user (u1 and u2) in human activity recognition. A number in bold face indicates the fact that the method is the best or comparable to the best one in terms of the mean misclassification rate by the *t-test* at the significance level 5%.

Walk vs.	LapRLS +CV	LapRLS +IWCV	KLR +CV	IWKLR +IWCV	LSPC +CV	IWLSPC +IWCV
Run (u1)	21.2(4.7)	11.3(4.7)	15.0(6.7)	**8.9**(0.8)	13.2(4.0)	**8.9**(0.6)
Bicycle (u1)	9.9(1.2)	12.5(1.4)	9.7(0.8)	**8.6**(4.5)	9.9(0.9)	**8.7**(4.7)
Train (u1)	2.2(0.23)	2.2(0.3)	**1.4**(0.4)	**1.2**(1.5)	1.5(0.4)	**1.1**(1.5)
Run (u2)	24.0(4.8)	**19.9**(9.4)	24.6(1.4)	**21.6**(5.8)	24.5(1.0)	**21.0**(5.6)
Bicycle (u2)	8.6(1.5)	9.9(2.5)	6.7(1.4)	6.8(1.7)	**6.3**(1.3)	**6.4**(1.8)
Train (u2)	4.0(1.3)	3.7(1.0)	3.7(0.7)	**2.9**(0.7)	3.5(0.6)	**3.1**(0.5)

Table 5: Mean computation time [s] and standard deviation (in parentheses) averaged over 50 trials for each new user (u1 and u2) in human activity recognition. A number in bold face indicates the fact that the method is the best or comparable to the best one in terms of the mean computation time by the *t-test* at the significance level 5%.

Walk vs.	LapRLS +CV	LapRLS +IWCV	KLR +CV	IWKLR +IWCV	LSPC +CV	IWLSPC +IWCV
Run (u1)	14.1(0.7)	14.5(0.8)	86.8(16.2)	78.8(23.2)	7.3(1.1)	**6.6**(1.3)
Bicycle (u1)	38.8(4.8)	52.8(8.1)	38.8(4.8)	52.8(8.1)	4.2(0.8)	**3.7**(0.8)
Train (u1)	5.5(0.6)	5.4(0.6)	19.8(7.3)	30.9(6.0)	**3.9**(0.8)	**4.0**(0.8)
Run (u2)	12.6(2.1)	12.1(2.2)	70.1(12.9)	128.5(51.7)	**8.2**(1.3)	**7.8**(1.5)
Bicycle (u2)	16.8(7.0)	27.2(5.6)	16.8(7.0)	27.2(5.6)	3.7(0.8)	**3.1**(0.9)
Train (u2)	5.6(0.7)	5.6(0.6)	24.9(10.8)	29.4(10.3)	**4.1**(0.8)	**3.9**(0.8)

5 Conclusions

In human activity recognition, there are several practical issues to be taken into account: a new user is not willing to gather labeled data, motion patterns are significantly different depending on users, and prediction with confidence is useful. In this paper, we formulated the new user problem in human activity recognition as a *covariate shift adaptation* problem [23], and proposed a computationally efficient probabilistic classification method. The proposed method is an importance-weighted variant of the *least-squares probabilistic classifier* [26, 33], which is computationally very efficient thanks to the availability of an analytic-form solution. Experiments on real-world human activity recognition illustrated the usefulness of the proposed method.

A potential limitation of the proposed method is that it relies on the assumption that the class-posterior probability does not change in between training and test phases. That is, when this assumption does not hold, e.g., an existing user has a different decision boundary between "walking" and "running" from the one of new user, the learned classifier

Table 6: Confusion matrix for three binary-classification tasks using IWLSPC+IWCV in human activity recognition. There are 80000 test samples (2 users × 50 trials × 800 test samples) in total for each task. A number in bold face indicates correct predictions.

		Predicted	
		Walk	Run
Actual	Walk	**31931**	8069
	Run	3907	**36093**

		Walk	Bicycle
Actual	Walk	**38474**	1526
	Bicycle	4494	**35506**

		Predicted	
		Walk	Train
Actual	Walk	**39973**	27
	Train	1651	**38349**

Table 7: Mean misclassification rates [%] and standard deviation (in parentheses) averaged over 50 trials for each new user (u1 and u2) using IWLSPC+IWCV with/without correction based on confidence levels in human activity recognition. A number in bold face indicates the fact that the method is the best or comparable to the best one in terms of the mean misclassification rate by the *t-test* at the significance level 5%.

Walk vs.	Without correction	With correction	With correction
		threshold=0.8	threshold=0.9
Run (u1)	8.9(0.6)	4.8(2.1)	**4.0**(3.8)
Bicycle (u1)	**8.7**(4.7)	**8.6**(4.6)	**8.6**(5.8)
Train (u1)	1.1(1.5)	**0.8**(1.4)	**0.8**(1.3)
Run (u2)	**21.0**(5.6)	23.3(8.2)	**21.3**(6.1)
Bicycle (u2)	6.4(1.8)	**5.0**(1.7)	5.4(1.7)
Train (u2)	3.1(0.5)	**0.4**(0.6)	**0.5**(0.7)

can be significantly biased. This problem can be overcome if we can eliminate training samples that follow a different class-posterior from the test samples. The change of two class-posterior densities can be detected using the density-ratio $p_{\mathrm{te}}(y|\boldsymbol{x})/p_{\mathrm{tr}}(y|\boldsymbol{x})$ as in *change-point detection* [16] and *two-sample test* [31]. The ratio of class-posterior densities can be in principle estimated similarly as the importance-weight estimation (B). Thus, developing an efficient algorithm for detecting the change of class-posterior would be important future works.

Acknowledgements

This work was supported by AOARD and the FIRST program.

A Model Selection by Importance-Weighted Cross-Validation

The performance of IWLSPC depends on the choice of the regularization parameter λ, the Gaussian kernel width σ, and the flattening parameter ν. *Cross-validation* (CV) is a standard method for model selection. However, under covariate shift, ordinary CV is highly biased due to the differing distributions. To cope with this problem, a variant of CV called *importance-weighted CV* (IWCV) has been proposed [28]. Here, we briefly describe the IWCV procedure.

Let us randomly divide the training set into $\mathcal{D} = \{(\boldsymbol{x}_n^{\mathrm{tr}}, y_n^{\mathrm{tr}})\}_{n=1}^{N_{\mathrm{tr}}}$ into K disjoint non-empty subsets $\{\mathcal{D}_k\}_{k=1}^{K}$ of (approximately) the same size. Let $\widehat{f}_k(\boldsymbol{x})$ be a function learned from $\mathcal{D}\backslash\mathcal{D}_k$ (i.e., without $\mathcal{D}_k$). Then the k-fold IWCV estimate of the generalization error is given by

$$\widehat{G}_{\mathrm{IWCV}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{|\mathcal{D}_k|}\sum_{(\boldsymbol{x}^{\mathrm{tr}},y^{\mathrm{tr}})\in\mathcal{D}_k} w(\boldsymbol{x}^{\mathrm{tr}})\mathrm{loss}(\widehat{f}_k(\boldsymbol{x}^{\mathrm{tr}}), y^{\mathrm{tr}}),$$

where $w(\boldsymbol{x})$ is the importance weight defined by Eq.(1), $\mathrm{loss}(\widehat{y}, y)$ is the *loss* function which measures the discrepancy between the true output value y and its estimate $\widehat{y}$, and $|\mathcal{D}|$ denotes the number of samples in the set $\mathcal{D}$. It was proved that IWCV gives an almost unbiased estimate of the generalization error even under covariate shift [28]. Finally, we select the model (λ, σ, and ν in the case of IWLSPC) that minimizes the above generalization error estimator:

$$(\widehat{\lambda}, \widehat{\sigma}, \widehat{\nu}) = \mathop{\mathrm{arg\,min}}_{(\lambda,\sigma,\nu)} \widehat{G}_{\mathrm{IWCV}}(\lambda, \sigma, \nu).$$

B Importance-Weight Estimation by Unconstrained Least-Squares Importance Fitting

IWLSPC (and IWCV) requires the values of the importance $\{w(\boldsymbol{x}_n^{\mathrm{tr}})\}_{n=1}^{N_{\mathrm{tr}}}$, which are unknown in practice. So far, various methods for estimating the importance has been developed [24, 7, 20, 12, 30, 13]. Among them, the method called *unconstrained least-squares importance fitting* (uLSIF) [13] was shown to be superior since it achieves the optimal convergence rate [14], it possesses the optimal numerical stability [15], and its solution can be computed analytically. Here, we briefly review the uLSIF method.

Suppose we are given training and test input points $\{\boldsymbol{x}_n^{\mathrm{tr}}\}_{n=1}^{N_{\mathrm{tr}}}$ and $\{\boldsymbol{x}_n^{\mathrm{te}}\}_{n=1}^{N_{\mathrm{te}}}$, which independently follow $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$, respectively. The main idea of uLSIF is to directly estimate the importance weight $w(\boldsymbol{x}) = p_{\mathrm{te}}(\boldsymbol{x})/p_{\mathrm{tr}}(\boldsymbol{x})$ without estimating $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$. Let us model the importance $w(\boldsymbol{x})$ by the following model:

$$w(\boldsymbol{x};\boldsymbol{\alpha}) = \sum_{n=1}^{N_{\mathrm{te}}} \alpha_n L(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}),$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{N_{\mathrm{te}}})^\top$ are parameters to be learned from data samples and $L(\boldsymbol{x}, \boldsymbol{x}')$ is a kernel function. Here, we focus on the Gaussian kernel:

$$L(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\kappa^2}\right),$$

where κ denotes the Gaussian kernel width. We determine the parameter $\boldsymbol{\alpha}$ so that the following squared error J is minimized:

$$\begin{aligned} J(\boldsymbol{\alpha}) &\equiv \frac{1}{2}\int \Big(w(\boldsymbol{x};\boldsymbol{\alpha}) - w(\boldsymbol{x})\Big)^2 p_{\mathrm{tr}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2}\int w(\boldsymbol{x};\boldsymbol{\alpha})^2 p_{\mathrm{tr}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \int w(\boldsymbol{x};\boldsymbol{\alpha}) p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + C' \\ &= \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{h}^\top\boldsymbol{\alpha} + C', \end{aligned}$$

where C' is a constant independent of the parameter $\boldsymbol{\alpha}$, and $\boldsymbol{H}$ is the $N_{\mathrm{te}} \times N_{\mathrm{te}}$ matrix and $\boldsymbol{h}$ is the N_{te}-dimensional vector defined as

$$\begin{aligned} H_{n,n'} &\equiv \int L(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) L(\boldsymbol{x}, \boldsymbol{x}_{n'}^{\mathrm{te}}) p_{\mathrm{tr}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \\ h_n &\equiv \int L(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \end{aligned}$$

The expectations in $\boldsymbol{H}$ and $\boldsymbol{h}$ are approximated by the empirical averages using $\{\boldsymbol{x}_n^{\mathrm{tr}}\}_{n=1}^{N_{\mathrm{tr}}}$

and $\{\boldsymbol{x}_n^{\mathrm{te}}\}_{n=1}^{N_{\mathrm{te}}}$ as

$$\widehat{H}_{n,n'} \equiv \frac{1}{N_{\mathrm{tr}}} \sum_{n''=1}^{N_{\mathrm{tr}}} L(\boldsymbol{x}_{n''}^{\mathrm{tr}}, \boldsymbol{x}_n^{\mathrm{te}}) L(\boldsymbol{x}_{n''}^{\mathrm{tr}}, \boldsymbol{x}_{n'}^{\mathrm{te}}),$$

$$\widehat{h}_n \equiv \frac{1}{N_{\mathrm{te}}} \sum_{n'=1}^{N_{\mathrm{te}}} L(\boldsymbol{x}_{n'}^{\mathrm{te}}, \boldsymbol{x}_n^{\mathrm{te}}).$$

Then, we obtain the following optimization problem:

$$\widehat{\boldsymbol{\alpha}} \equiv \operatorname*{arg\,min}_{\boldsymbol{\alpha}} \left[\frac{1}{2} \boldsymbol{\alpha}^\top \widehat{\boldsymbol{H}} \boldsymbol{\alpha} - \widehat{\boldsymbol{h}}^\top \boldsymbol{\alpha} + \frac{\gamma}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha} \right],$$

where $\frac{\gamma}{2}\boldsymbol{\alpha}^\top\boldsymbol{\alpha}$ ($\gamma \geq 0$) is a regularization term. Then the uLSIF solution can be *analytically* computed as

$$\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \gamma \boldsymbol{I}_{N_{\mathrm{te}}})^{-1} \widehat{\boldsymbol{h}},$$

where $\boldsymbol{I}_{N_{\mathrm{te}}}$ denotes the N_{te}-dimensional identity matrix. Finally, since the importance weight is non-negative by definition, we modify the solution as

$$\widehat{w}(\boldsymbol{x}) \equiv \max \left(0, \sum_{n=1}^{N_{\mathrm{te}}} \widehat{\alpha}_n L(\boldsymbol{x}, \boldsymbol{x}_n^{\mathrm{te}}) \right).$$

The performance of uLSIF depends on the choice of the regularization parameter γ and the Gaussian kernel width κ. These tuning parameters can be determined by cross-validation (CV) as follows. Let us randomly divide $\mathcal{X}^{\mathrm{tr}} = \{\boldsymbol{x}_n^{\mathrm{tr}}\}_{n=1}^{N_{\mathrm{tr}}}$ and $\mathcal{X}^{\mathrm{te}} = \{\boldsymbol{x}_n^{\mathrm{te}}\}_{n=1}^{N_{\mathrm{te}}}$ into K disjoint non-empty subsets $\{\mathcal{X}_k^{\mathrm{tr}}\}_{k=1}^{K}$ and $\{\mathcal{X}_k^{\mathrm{te}}\}_{k=1}^{K}$, respectively. Let $\widehat{w}_k(\boldsymbol{x})$ be an importance function learned from $\mathcal{X}^{\mathrm{tr}} \backslash \mathcal{X}_k^{\mathrm{tr}}$ and $\mathcal{X}^{\mathrm{te}} \backslash \mathcal{X}_k^{\mathrm{te}}$ (i.e., without $\mathcal{X}_k^{\mathrm{tr}}$ and $\mathcal{X}_k^{\mathrm{te}}$). Then the k-fold CV estimate of the importance estimation error J is given by

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

where $|\mathcal{X}|$ denotes the number of samples in the set $\mathcal{X}$. Finally, we select the model (i.e., the regularization parameter γ and the Gaussian kernel width κ) that minimizes $\widehat{J}_{\mathrm{CV}}$:

$$(\widehat{\gamma}, \widehat{\kappa}) = \operatorname*{arg\,min}_{(\gamma,\kappa)} \widehat{J}_{\mathrm{CV}}(\gamma, \kappa).$$

References

[1] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of the 2nd IEEE International Conference on Pervasive Computing*, pages 1–17, 2004.

[2] N. B. Bharatula, M. Stager, P. Lukowicz, and G Troster. Empirical study of design choices in multi-sensor context recognition systems. In *Proceedings of the 2nd International Forum on Applied Wearable Computing*, pages 79–93, 2005.

[3] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, 2007.

[4] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspodence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, 2006.

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

[6] C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 442–450. 2010.

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

[8] H. Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th International Annual Meeting of the Association of Computational Linguistics*, pages 256–263, 2007.

[9] H. Daume III, A. Kumar, and A. Saha. Co-regularization based semi-supervised domain adaptation. In *Advances in Neural Information Processing Systems 23*, pages 256–263, 2010.

[10] C. Gao, F. Kong, and J. Tan. Healthaware: Tackling obesity with health aware smart phone systems. In *Proceedings of the 2009 IEEE International Conference on Robotics and Biometics*, pages 1549–1554, 2009.

[11] Y. Hattori, S. Inoue, T. Masaki, G. Hirakawa, and O. Sudo. Gathering large scale human activity information using mobile sensor devices. In *Proceedings of the Second International Workshop on Network Traffic Control, Analysis and Applications*, pages 708–713, 2010.

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

[13] 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.

[14] T. Kanamori, T. Suzuki, and M. Sugiyama. Kernel-based least-squares density-ratio estimation i: Statistical analysis. *Machine Learning*, 2011. submitted.

[15] T. Kanamori, T. Suzuki, and M. Sugiyama. Kernel-based least-squares density-ratio estimation ii: Condition number analysis. *Machine Learning*, 2011. submitted.

[16] Y. Kawahara and M. Sugiyama. Sequential change-point detection based on direct density-ratio estimation. *Statistical Analysis and Data Mining*, 2011. to appear.

[17] P. Leijdekkers and V. Gay. A self-test to detect a heart attack using a mobile phone and wearable sensors. In *Proceedings of the 21st IEEE International Symposium on Computer-Based Medical Systems*, pages 93–98, 2008.

[18] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[19] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

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

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

[22] M. Schmidt. minfunc, 2005. http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html.

[23] 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.

[24] B. W. Silverman. Density ratios, empirical likelihood and cot death. *Journal of the Royal Statistical Society, Series C*, 27(1):26–33, 1978.

[25] E. P. Stuntebeck, J. S. Davis II, G. D. Abowd, and M. Blount. Healthsense: Classification of health-related sensor data through user-assisted machine learning. In *Proceedings of the 9th Workshop on Mobile Computing Systems and Aapplications*, pages 1–5, 2008.

[26] M. Sugiyama. Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting. *IEICE Transactions on Information and Systems*, E93-D:2690–2701, August 2010.

[27] M. Sugiyama and M. Kawanabe. *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation.* MIT Press, Cambridge, MA, USA, 2011. to appear.

[28] 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.

[29] M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning: A Versatile Tool for Statistical Data Processing.* Cambridge University Press, Cambridge, UK, 2012. to appear.

[30] 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.

[31] 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.

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

[33] M. Yamada, M. Sugiyama, G. Wichern, and J. Simm. Improving the accuracy of least-squares probabilistic classifiers. *IEICE Transactions on Information and Systems*, E94-D(6):1337–1340, 2011.

[34] K. Yamagishi, N. Ito, H. Kosuga, N. Yasuda, K. Isogami, and N. Kozuno. A simplified measurement of farm worker's load using an accelerometer. *Journal of the Japanese Society of Agricultural Technology Management*, 9(2):127–132, 2002.

[35] Y. Zhang, X. Hu, and Y. Fang. Logistic regression for transductive transfer learning from multiple sources. In *Proceedings of Advanced Data Mining and Applications*, pages 175–182, 2010.