Statistical Outlier Detection Using Direct Density Ratio Estimation

Shohei Hido
IBM Research, Tokyo Research Laboratory, Japan.
Department of Systems Science, Kyoto University, Japan
hido@jp.ibm.com

Yuta Tsuboi
IBM Research, Tokyo Research Laboratory, Japan.
yutat@jp.ibm.com

Hisashi Kashima*
IBM Research, Tokyo Research Laboratory, Japan.
hkashima@jp.ibm.com

Masashi Sugiyama
Department of Computer Science, Tokyo Institute of Technology, Japan.
PRESTO, Japan Science and Technology Agency, Japan.
sugi@cs.titech.ac.jp http://sugiyama-www.cs.titech.ac.jp/~sugi

Takafumi Kanamori
Department of Computer Science and Mathematics Informatics,
Nagoya University, Japan.
kanamori@is.nagoya-u.ac.jp

Abstract

We propose a new statistical approach to the problem of inlier-based outlier detection, i.e., finding outliers in the test set based on the training set consisting only of inliers. Our key idea is to use the *ratio* of training and test data densities as an outlier score. This approach is expected to have better performance even in high-dimensional problems since methods for directly estimating the density ratio without going through density estimation are available. Among various density ratio estimation methods, we employ the method called unconstrained least-squares importance fitting (uLSIF) since it is equipped with natural cross-validation procedures, allowing us to objectively optimize the value of tuning parameters such as the regularization parameter and the kernel width. Furthermore, uLSIF offers a closed-form solution as well as a closed-form formula for the leave-one-out error, so

*Currently with Department of Mathematical Informatics, The University of Tokyo, Japan.

it is computationally very efficient and is scalable to massive datasets. Simulations with benchmark and real-world datasets illustrate the usefulness of the proposed approach.

Keywords

outlier detection, density ratio, importance, unconstrained least-squares importance fitting (uLSIF).

1 Introduction

The goal of *outlier detection* (a.k.a. *anomaly detection*, *novelty detection*, or *one-class classification*) is to find uncommon instances ('outliers') in a given dataset. Outlier detection has been used in various applications such as defect detection from behavior patterns of industrial machines (Fujimaki, Yairi and Machida, 2005; Idé and Kashima, 2004), intrusion detection in network systems (Yamanishi, Takeuchi, Williams and Milne, 2004), and topic detection in news documents (Manevitz and Yousef, 2002). Recent studies include finding unusual patterns in time-series (Yankov, Keogh and Rebbapragada, 2008), discovery of spatio-temporal changes in time-evolving graphs (Chan, Bailey and Leckie, 2008), self-propagating worm detection in information systems (Jiang and Zhu, 2009), and identification of inconsistent records in construction equipment data (Fan, Zaïane, Foss and Wu, 2009). Since outlier detection is useful in various applications it has been a active research topic in statistics, machine learning, and data mining communities for decades (Hodge and Austin, 2004).

A standard outlier detection problem falls into the category of *unsupervised learning* due to lack of prior knowledge on the 'anomalous data'. In contrast, Gao, Cheng and Tan (2006*a*) and Gao, Cheng and Tan (2006*b*) addressed the problem of *semi-supervised* outlier detection where some examples of outlier and inlier are available as a training set. The semi-supervised outlier detection methods could perform better than unsupervised methods thanks to additional label information, but such outlier samples for training are not always available in practice. Furthermore, the type of outliers may be diverse and thus the semi-supervised methods—learning from *known* types of outliers—are not necessarily useful in detecting *unknown* types of outliers.

In this paper, we address the problem of *inlier-based* outlier detection where examples of inlier are available. More formally, the inlier-based outlier detection problem is to find outlier instances in the test set based on the training set consisting only of inlier instances. The setting of inlier-based outlier detection would be more practical than the semi-supervised setting since inlier samples are often available abundantly. For example, in defect detection of industrial machines, we already know that there is no outlier (i.e., a defect) in the past since no failure has been observed in the machinery. Therefore, it is reasonable to separate the measurement data into a training set consisting only of inlier samples observed in the past and the test set consisting of recent samples from which we try to find outliers.

As opposed to supervised learning, the outlier detection problem is vague and it is not possible to universally define what the outliers are. In this paper, we consider a statistical framework and regard instances with low probability densities as outliers. In light of inlier-based outlier detection, outliers may be identified via density estimation of inlier samples. However, density estimation is known to be a hard problem particularly in high dimensions, so outlier detection via density estimation may not work well in practice.

To avoid density estimation, we may use *One-class Support Vector Machine (OSVM)* (Schölkopf, Platt, Shawe-Taylor, Smola and Williamson, 2001) or *Support Vector Data Description (SVDD)* (Tax and Duin, 2004), which finds an inlier region containing a certain fraction of training instances; samples outside the inlier region are regarded as outliers. However, these methods cannot make use of inlier information available in the inlier-based settings. Furthermore, the solutions of OSVM and SVDD depend heavily on the choice of tuning parameters (e.g., the Gaussian kernel width) and there seems to be no reasonable method to appropriately determine the values of the tuning parameters.

To overcome the weakness of the existing methods, we propose a new approach to inlier-based outlier detection. Our key idea is not to directly model the training and test data densities, but only to estimate the *ratio* of training and test data densities. Among existing methods of density ratio estimation (Qin, 1998; Cheng and Chu, 2004; Huang, Smola, Gretton, Borgwardt and Schölkopf, 2007; Bickel, Brückner and Scheffer, 2007; Sugiyama, Nakajima, Kashima, von Bünau and Kawanabe, 2008; Nguyen, Wainwright and Jordan, 2008; Sugiyama, Suzuki, Nakajima, Kashima, von Bünau and Kawanabe, 2008; Kanamori, Hido and Sugiyama, 2009*a*; Kanamori, Hido and Sugiyama, 2009*b*), we employ an algorithm called *unconstrained Least-Squares Importance Fitting (uLSIF)* (Kanamori, Hido and Sugiyama, 2009*a*; Kanamori, Hido and Sugiyama, 2009*b*) for outlier detection. The reason for this choice is that uLSIF is equipped with a variant of cross-validation (CV), so the values of tuning parameters such as the regularization parameter can be objectively determined without subjective trial and error. Furthermore, uLSIF-based outlier detection allows us to compute the outlier score just by solving a system of linear equations—the leave-one-out cross-validation (LOOCV) error can also be computed analytically. Thus, uLSIF-based outlier detection is computationally very efficient and therefore is scalable to massive datasets. Through experiments using benchmark datasets and real-world datasets of failure detection in hard disk drives and financial risk management in loan business, our approach is shown to compare favorably with existing outlier detection methods and other density ratio estimation methods both in accuracy and scalability.

This paper is an extended version of our earlier conference paper presented at IEEE ICDM 2008 (Hido, Tsuboi, Kashima, Sugiyama and Kanamori, 2008), with more details and additional results. The rest of this paper is organized as follows. In Section 2, we mathematically formulate the inlier-based outlier detection problem as a density ratio estimation problem. In Section 3, we give a comprehensive review of existing density ratio estimation methods. In Section 4, we discuss the characteristics of the existing density ratio estimation methods and propose a practical outlier detection procedure based on uLSIF; illustrative numerical examples of the proposed method are also shown. In

Section 5, we discuss the relation between the proposed uLSIF-based method and existing outlier detection methods. In Section 6, we experimentally compare the performance of the proposed and existing algorithms using benchmark and real-world datasets. Finally, in Section 7, we conclude by summarizing our contributions.

2 Outlier Detection via Direct Importance Estimation

In this section, we propose a new statistical approach to outlier detection.

Suppose we have two sets of samples—training samples $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ and test samples $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$ in a domain $\mathcal{D}$ ($\subset \mathbb{R}^d$). The training samples $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ are all inliers, while the test samples $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$ can contain some outliers. The goal of outlier detection here is to identify outliers in the test set based on the training set consisting only of inliers. More formally, we want to assign a suitable *inlier score* for the test samples—the smaller the value of the inlier score is, the more plausible the sample is an outlier.

Let us consider a statistical framework of the inlier-based outlier detection problem: suppose training samples $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ are independent and identically distributed (i.i.d.) following a training data distribution with density $p_{\mathrm{tr}}(\boldsymbol{x})$ and test samples $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$ are i.i.d. following a test data distribution with strictly positive density $p_{\mathrm{te}}(\boldsymbol{x})$. Within this statistical framework, test samples with low training data densities are regarded as outliers. However, $p_{\mathrm{tr}}(\boldsymbol{x})$ is not accessible in practice and density estimation is known to be a hard problem. Therefore, merely using the training data density as an inlier score may not be promising in practice.

In this paper, we propose to use the ratio of training and test data densities, called the *importance*, as an inlier score:

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

If there exists no outlier sample in the test set (i.e., the training and test data densities are equivalent), the value of the importance is one. The importance value tends to be small in the regions where the training data density is low and the test data density is high. Thus samples with small importance values are plausible to be outliers.

One may suspect that this importance-based approach is not suitable when there exist only a small number of outliers—since a small number of outliers cannot increase the values of $p_{\mathrm{te}}(\boldsymbol{x})$ significantly. However, outliers are drawn from a region with small $p_{\mathrm{tr}}(\boldsymbol{x})$ and therefore a small change in $p_{\mathrm{te}}(\boldsymbol{x})$ *significantly* reduces the importance value. For example, let the increase of $p_{\mathrm{te}}(\boldsymbol{x})$ be $\epsilon = 0.01$; then $\frac{1}{1+\epsilon} \approx 1$, but $\frac{0.001}{0.001+\epsilon} \ll 1$. Thus the importance $w(\boldsymbol{x})$ would be a suitable inlier score (see Section 4.3 for illustrative examples).

3 Direct Importance Estimation Methods

The values of the importance are unknown in practice, so we need to estimate them from data samples. If one estimates the training and test densities separately from the data samples, one will be able to estimate the importance just by taking the ratio of the two estimated densities. However, this naive approach can easily suffer from the *curse of dimensionality*, particularly when the data has neither low dimensionality nor a simple distribution. As advocated by Vapnik (1998), density ratio estimation is crucial in statistical learning, but often unnecessarily more difficult than the target problem that one would like to solve. In our case, we would like to *directly* estimate the importance values without going through density estimation. In this section, we review such direct importance estimation methods which could be used for inlier-based outlier detection. In fact, it has been shown that direction estimation methods work better than the two-step approach of separately estimating the two densities and then taking their ratio (Sugiyama, Suzuki, Nakajima, Kashima, von Bünau and Kawanabe, 2008). Therefore these methods are expected to also work well on the inlier-based outlier detection problems.

3.1 Kernel Mean Matching (KMM)

The KMM method (Huang et al., 2007) avoids density estimation and directly gives an estimate of the importance at test points (i.e., data points drawn from the denominator of the ratio).

The basic idea of KMM is to find $\widehat{w}(\boldsymbol{x})$ such that the mean discrepancy between nonlinearly transformed samples drawn from $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$ is minimized in a *universal reproducing kernel Hilbert space* (Steinwart, 2001). The Gaussian kernel

$$K_\sigma(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right) \tag{1}$$

is an example of kernels that induce a universal reproducing kernel Hilbert space. It has been shown that the solution of the following optimization problem agrees with the true importance:

$$\begin{aligned} &\min_{w} \left\| \int K_\sigma(\boldsymbol{x}, \cdot) p_{\mathrm{tr}}(\boldsymbol{x}) d\boldsymbol{x} - \int K_\sigma(\boldsymbol{x}, \cdot) w(\boldsymbol{x}) p_{\mathrm{te}}(\boldsymbol{x}) d\boldsymbol{x} \right\|_{\mathcal{F}}^2 \\ &\text{s.t.} \int w(\boldsymbol{x}) p_{\mathrm{te}}(\boldsymbol{x}) d\boldsymbol{x} = 1 \quad \text{and} \quad w(\boldsymbol{x}) \geq 0, \end{aligned}$$

where $\|\cdot\|_{\mathcal{F}}$ denotes the norm in the Gaussian reproducing kernel Hilbert space.

An empirical version of the above problem is reduced to the following quadratic pro-

gram:

$$\min_{\{w_i\}_{i=1}^{n_{\mathrm{te}}}} \left[\frac{1}{2} \sum_{i,i'=1}^{n_{\mathrm{te}}} w_i w_{i'} K_\sigma(\boldsymbol{x}_i^{\mathrm{te}}, \boldsymbol{x}_{i'}^{\mathrm{te}}) - \sum_{i=1}^{n_{\mathrm{te}}} w_i \kappa_i \right]$$
$$\text{s.t.} \left| \sum_{i=1}^{n_{\mathrm{te}}} w_i - n_{\mathrm{te}} \right| \leq n_{\mathrm{te}} \epsilon \quad \text{and} \quad 0 \leq w_1, \ldots, w_{n_{\mathrm{te}}} \leq B,$$

where

$$\kappa_i = \frac{n_{\mathrm{te}}}{n_{\mathrm{tr}}} \sum_{j=1}^{n_{\mathrm{tr}}} K_\sigma(\boldsymbol{x}_i^{\mathrm{te}}, \boldsymbol{x}_j^{\mathrm{tr}}).$$

σ (≥ 0), B (≥ 0), and ϵ (≥ 0) are tuning parameters. The solution $\{\widehat{w}_i\}_{i=1}^{n_{\mathrm{te}}}$ is an estimate of the importance at the test points $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$.

Since KMM does not require the individual density estimates, it is expected to work well. However, the performance of KMM is dependent on the tuning parameters B, ϵ, and σ and they cannot be simply optimized, e.g., by cross-validation (CV) since the estimates of the importance are available only at the test points.

3.2 Logistic Regression (LogReg)

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

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

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

$$w(\boldsymbol{x}) = \frac{p(\eta = -1)}{p(\eta = 1)} \frac{p(\eta = 1|\boldsymbol{x})}{p(\eta = -1|\boldsymbol{x})}. \tag{2}$$

The probability ratio of test and training samples $p(\eta = -1)/p(\eta = 1)$ may be simply estimated by the ratio of the numbers of samples $n_{\mathrm{te}}/n_{\mathrm{tr}}$. The conditional probability $p(\eta|\boldsymbol{x})$ could be approximated by discriminating test samples from training samples using a LogReg classifier, where η plays the role of a class variable. Thus, using the LogReg classifier, we can estimate $w(\boldsymbol{x})$ without going through of estimation of $p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$. Below we briefly explain the LogReg method.

The LogReg classifier based on kernel logistic regression employs the following parametric model for expressing the conditional probability $p(\eta|\boldsymbol{x})$:

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

where m is the number of basis functions and $\{\phi_\ell(\boldsymbol{x})\}_{\ell=1}^m$ are fixed basis functions. By substituting this equation into Eq.(2), an importance estimator is given by

$$\begin{aligned}\widehat{w}(\boldsymbol{x}) &= \frac{\widehat{p}(\eta=-1)}{\widehat{p}(\eta=1)}\frac{\widehat{p}(\eta=1|\boldsymbol{x})}{\widehat{p}(\eta=-1|\boldsymbol{x})} = \frac{n_{\mathrm{te}}}{n_{\mathrm{tr}}}\frac{1+\exp\left(\sum_{\ell=1}^m \zeta_\ell\phi_\ell(\boldsymbol{x})\right)}{1+\exp\left(-\sum_{\ell=1}^m \zeta_\ell\phi_\ell(\boldsymbol{x})\right)} \\ &= \frac{n_{\mathrm{te}}}{n_{\mathrm{tr}}}\exp\left(\sum_{\ell=1}^m \zeta_\ell\phi_\ell(\boldsymbol{x})\right).\end{aligned}$$

The parameters $\{\zeta_\ell\}_{\ell=1}^m$ are learned by minimizing the negative regularized log-likelihood:

$$\begin{aligned}\min_{\boldsymbol{\zeta}}\Bigg[&\sum_{i=1}^{n_{\mathrm{te}}}\log\left(1+\exp\left(\sum_{\ell=1}^m \zeta_\ell\phi_\ell(\boldsymbol{x}_i^{\mathrm{te}})\right)\right) \\ &+\sum_{j=1}^{n_{\mathrm{tr}}}\log\left(1+\exp\left(-\sum_{\ell=1}^m \zeta_\ell\phi_\ell(\boldsymbol{x}^{\mathrm{tr}})\right)\right)+\lambda\sum_{\ell=1}^m\zeta_\ell^2\Bigg].\end{aligned}$$

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

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

3.3 Kullback-Leibler Importance Estimation Procedure (KLIEP)

KLIEP (Sugiyama, Nakajima, Kashima, von Bünau and Kawanabe, 2008; Sugiyama, Suzuki, Nakajima, Kashima, von Bünau and Kawanabe, 2008) also directly gives an estimate of the importance function without going through density estimation by implicitly matching the true and estimated distributions under the Kullback-Leibler divergence.

Let us model the importance $w(\boldsymbol{x})$ by the following linear model:

$$\widehat{w}(\boldsymbol{x}) = \sum_{\ell=1}^b \alpha_\ell\varphi_\ell(\boldsymbol{x}), \tag{3}$$

where $\{\alpha_\ell\}_{\ell=1}^b$ are parameters and $\{\varphi_\ell(\boldsymbol{x})\}_{\ell=1}^b$ are basis functions such that $\varphi_\ell(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \mathcal{D}$ and for $\ell = 1, \ldots, b$. Then an estimator of the training data density $p_{\mathrm{tr}}(\boldsymbol{x})$ is given by

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

In KLIEP, the parameters $\{\alpha_\ell\}_{\ell=1}^b$ are determined so that the Kullback-Leibler divergence

from $p_{\mathrm{tr}}(\boldsymbol{x})$ to $\widehat{p}_{\mathrm{tr}}(\boldsymbol{x})$ is minimized:

$$\begin{aligned}\mathrm{KL}[p_{\mathrm{tr}}(\boldsymbol{x})\|\widehat{p}_{\mathrm{tr}}(\boldsymbol{x})] &= \int p_{\mathrm{tr}}(\boldsymbol{x}) \log \frac{p_{\mathrm{tr}}(\boldsymbol{x})}{\widehat{w}(\boldsymbol{x}) p_{\mathrm{te}}(\boldsymbol{x})} d\boldsymbol{x} \\ &= \int p_{\mathrm{tr}}(\boldsymbol{x}) \log \frac{p_{\mathrm{tr}}(\boldsymbol{x})}{p_{\mathrm{te}}(\boldsymbol{x})} d\boldsymbol{x} - \int p_{\mathrm{tr}}(\boldsymbol{x}) \log \widehat{w}(\boldsymbol{x}) d\boldsymbol{x}. \end{aligned} \tag{4}$$

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

$$1 = \int \widehat{p}_{\mathrm{tr}}(\boldsymbol{x}) d\boldsymbol{x} = \int \widehat{w}(\boldsymbol{x}) p_{\mathrm{te}}(\boldsymbol{x}) d\boldsymbol{x}. \tag{5}$$

The KLIEP optimization problem is then given by replacing the expectations in Eqs.(4) and (5) with empirical averages:

$$\begin{aligned} &\max_{\{\alpha_\ell\}_{\ell=1}^b} \left[\sum_{j=1}^{n_{\mathrm{tr}}} \log \left(\sum_{\ell=1}^b \alpha_\ell \varphi_\ell(\boldsymbol{x}_j^{\mathrm{tr}}) \right) \right] \\ &\text{s.t.} \ \frac{1}{n_{\mathrm{te}}} \sum_{\ell=1}^b \alpha_\ell \left(\sum_{i=1}^{n_{\mathrm{te}}} \varphi_\ell(\boldsymbol{x}_i^{\mathrm{te}}) \right) = 1 \ \ \text{and} \ \ \alpha_1, \ldots, \alpha_b \geq 0. \end{aligned}$$

This is a convex optimization problem and the global solution can be obtained, e.g., by simply performing gradient ascent and feasibility satisfaction iteratively. A pseudo code of the KLIEP optimization procedure is described in Figure 1. Note that the solution $\{\widehat{\alpha}_\ell\}_{\ell=1}^b$ tends to be *sparse* (Boyd and Vandenberghe, 2004), which contributes to reducing the computational cost in the test phase. See Nguyen et al. (2008) and Sugiyama, Suzuki, Nakajima, Kashima, von Bünau and Kawanabe (2008) for the convergence proofs.

Model selection of KLIEP is possible by a variant of *likelihood cross-validation* (LCV) (Härdle, Müller, Sperlich and Werwatz, 2004) as follows. We first divide the training samples $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ into a learning part and a validation part, the model is trained based on the learning part, and then its likelihood is verified using the validation part; the model with the largest estimated likelihood is chosen. A pseudo code of LCV for KLIEP is described in Figure 2. Note that this LCV procedure corresponds to choosing the model with the smallest $\mathrm{KL}[p_{\mathrm{tr}}(\boldsymbol{x})\|\widehat{p}_{\mathrm{tr}}(\boldsymbol{x})]$.

A MATLAB® implementation of the entire KLIEP algorithm is available from the following web page:

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

3.4 Least-squares Importance Fitting

KLIEP employed the Kullback-Leibler divergence for measuring the discrepancy between two densities. *Least-squares importance fitting* (LSIF) (Kanamori, Hido and Sugiyama,

Input: $m = \{\varphi_\ell(\boldsymbol{x})\}_{\ell=1}^{b}$, $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$, $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$
Output: $\widehat{w}(\boldsymbol{x})$

$\boldsymbol{A}_{j,\ell} \longleftarrow \varphi_\ell(\boldsymbol{x}_j^{\mathrm{tr}})$ for $j = 1, \ldots, n_{\mathrm{tr}}$ and $\ell = 1, \ldots, b$;
$\boldsymbol{b}_\ell \longleftarrow \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \varphi_\ell(\boldsymbol{x}_i^{\mathrm{te}})$
Initialize $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_b)^\top$ $(> \boldsymbol{0})$ and ε $(0 < \varepsilon \ll 1)$;
Repeat until convergence
    $\boldsymbol{\alpha} \longleftarrow \boldsymbol{\alpha} + \varepsilon \boldsymbol{A}^\top(\boldsymbol{1}./\boldsymbol{A}\boldsymbol{\alpha})$; % Gradient ascent
    $\boldsymbol{\alpha} \longleftarrow \boldsymbol{\alpha} + (1 - \boldsymbol{b}^\top\boldsymbol{\alpha})\boldsymbol{b}/(\boldsymbol{b}^\top\boldsymbol{b})$;
    $\boldsymbol{\alpha} \longleftarrow \max(\boldsymbol{0}, \boldsymbol{\alpha})$;
    $\boldsymbol{\alpha} \longleftarrow \boldsymbol{\alpha}/(\boldsymbol{b}^\top\boldsymbol{\alpha})$;
end
$\widehat{w}(\boldsymbol{x}) \longleftarrow \sum_{\ell=1}^{b} \alpha_\ell \varphi_\ell(\boldsymbol{x})$;

Figure 1: Pseudo code of the optimization procedure for KLIEP. $\boldsymbol{0}$ and $\boldsymbol{1}$ denote the vectors with all zeros and ones, respectively. './' indicates the element-wise division and $^\top$ denotes the transpose. Inequalities and the 'max' operation for vectors are applied in the element-wise manner.

Input: $\mathcal{M} = \{m = \{\varphi_\ell^m(\boldsymbol{x})\}_{\ell=1}^{b_m}\}$, $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$, $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$
Output: $\widehat{w}(\boldsymbol{x})$

Split $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ into R disjoint subsets $\{\mathcal{X}_r\}_{r=1}^{R}$;
for each model $m \in \mathcal{M}$
    for each split $r = 1, \ldots, R$
        $\widehat{w}_r(\boldsymbol{x}) \longleftarrow \mathrm{KLIEP}(m, \{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}, \{\mathcal{X}_j\}_{j \neq r})$;
        $\widehat{J}_r(m) \longleftarrow \frac{1}{|\mathcal{X}_r|} \sum_{\boldsymbol{x} \in \mathcal{X}_r} \log \widehat{w}_r(\boldsymbol{x})$;
    end
    $\widehat{J}(m) \longleftarrow \frac{1}{R} \sum_{r=1}^{R} \widehat{J}_r(m)$;
end
$\widehat{m} \longleftarrow \mathrm{argmax}_{m \in \mathcal{M}} \widehat{J}(m)$;
$\widehat{w}(\boldsymbol{x}) \longleftarrow \mathrm{KLIEP}(\widehat{m}, \{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}, \{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}})$;

Figure 2: Pseudo code of model selection for KLIEP by LCV.

2009*a*; Kanamori, Hido and Sugiyama, 2009*b*) uses the squared loss for density-ratio function fitting. The density ratio $w(\boldsymbol{x})$ is again modeled by the linear model (3).

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

$$\begin{aligned} J_0(\boldsymbol{\alpha}) &= \frac{1}{2}\int \left(\widehat{w}(\boldsymbol{x}) - w(\boldsymbol{x})\right)^2 p_{\mathrm{te}}(\boldsymbol{x})d\boldsymbol{x} \\ &= \frac{1}{2}\int \widehat{w}(\boldsymbol{x})^2 p_{\mathrm{te}}(\boldsymbol{x})d\boldsymbol{x} - \int \widehat{w}(\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})d\boldsymbol{x} + \frac{1}{2}\int w(\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})d\boldsymbol{x}, \end{aligned} \tag{6}$$

where the last term is a constant and therefore can be safely ignored. Let us denote the first two terms by J:

$$J(\boldsymbol{\alpha}) = \frac{1}{2}\int \widehat{w}(\boldsymbol{x})^2 p_{\mathrm{te}}(\boldsymbol{x})d\boldsymbol{x} - \int \widehat{w}(\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})d\boldsymbol{x}.$$

Approximating the expectations in J by empirical averages, we obtain

$$\begin{aligned} \widehat{J}(\boldsymbol{\alpha}) &= \frac{1}{2n_{\mathrm{te}}}\sum_{i=1}^{n_{\mathrm{te}}} \widehat{w}(\boldsymbol{x}_i^{\mathrm{te}})^2 - \frac{1}{n_{\mathrm{tr}}}\sum_{j=1}^{n_{\mathrm{tr}}} \widehat{w}(\boldsymbol{x}_j^{\mathrm{tr}}) \\ &= \frac{1}{2}\sum_{\ell,\ell'=1}^{b} \alpha_\ell \alpha_{\ell'} \widehat{H}_{\ell,\ell'} - \sum_{\ell=1}^{b} \alpha_\ell \widehat{h}_\ell, \end{aligned} \tag{7}$$

where

$$\widehat{H}_{\ell,\ell'} = \frac{1}{n_{\mathrm{te}}}\sum_{i=1}^{n_{\mathrm{te}}} \varphi_\ell(\boldsymbol{x}_i^{\mathrm{te}})\varphi_{\ell'}(\boldsymbol{x}_i^{\mathrm{te}}), \tag{8}$$

$$\widehat{h}_\ell = \frac{1}{n_{\mathrm{tr}}}\sum_{j=1}^{n_{\mathrm{tr}}} \varphi_\ell(\boldsymbol{x}_j^{\mathrm{tr}}). \tag{9}$$

Taking into account the non-negativity of the density-ratio function $w(\boldsymbol{x})$, the optimization problem is formulated as follows.

$$\begin{aligned} &\min_{\{\alpha_\ell\}_{\ell=1}^b} \left[\frac{1}{2}\sum_{\ell,\ell'=1}^{b} \alpha_\ell \alpha_{\ell'} \widehat{H}_{\ell,\ell'} - \sum_{\ell=1}^{b} \alpha_\ell \widehat{h}_\ell + \lambda \sum_{\ell=1}^{b} \alpha_\ell\right] \\ &\text{s.t. } \alpha_1, \ldots, \alpha_b \geq 0, \end{aligned} \tag{10}$$

where a penalty term $\lambda \sum_{\ell=1}^b \alpha_\ell$ is included for regularization purposes with λ (≥ 0) being a regularization parameter. Eq.(10) is a convex quadratic programming problem and therefore the unique global optimal solution can be computed efficiently by a standard optimization package.

Model selection of the Gaussian width σ and the regularization parameter λ is possible by a variant of CV: First, $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$ and $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ are divided into R disjoint subsets

Input: $\widehat{\boldsymbol{H}}$ and $\widehat{\boldsymbol{h}}$ % see Eqs.(8) and (9) for the definition
Output: entire regularization path $\widehat{\boldsymbol{\alpha}}(\lambda)$ for $\lambda \geq 0$

$\tau \longleftarrow 0$; $k \longleftarrow \mathrm{argmax}_i\{\widehat{h}_i \mid i = 1, \ldots, b\}$;
$\lambda_\tau \longleftarrow \widehat{h}_k$; $\widehat{\mathcal{A}} \longleftarrow \{1, \ldots, b\} \backslash \{k\}$;
$\widehat{\boldsymbol{\alpha}}(\lambda_\tau) \longleftarrow \mathbf{0}_b$; % the vector with all zeros
While $\lambda_\tau > 0$
  $\widehat{\boldsymbol{E}} \longleftarrow \boldsymbol{O}_{|\widehat{\mathcal{A}}| \times b}$; % the matrix with all zeros
  For $i = 1, \ldots, |\widehat{\mathcal{A}}|$
    $\widehat{E}_{i,j_i} \longleftarrow 1$; % $\widehat{\mathcal{A}} = \{j_1, \ldots, j_{|\widehat{\mathcal{A}}|} \mid j_1 < \cdots < j_{|\widehat{\mathcal{A}}|}\}$
  end
  $\widehat{\boldsymbol{G}} \longleftarrow \begin{pmatrix} \widehat{\boldsymbol{H}} & -\widehat{\boldsymbol{E}}^\top \\ -\widehat{\boldsymbol{E}} & \boldsymbol{O}_{|\widehat{\mathcal{A}}| \times |\widehat{\mathcal{A}}|} \end{pmatrix}$;
  $\boldsymbol{u} \longleftarrow \widehat{\boldsymbol{G}}^{-1} \begin{pmatrix} \widehat{\boldsymbol{h}} \\ \mathbf{0}_{|\widehat{\mathcal{A}}|} \end{pmatrix}$; $\boldsymbol{v} \longleftarrow \widehat{\boldsymbol{G}}^{-1} \begin{pmatrix} \mathbf{1}_b \\ \mathbf{0}_{|\widehat{\mathcal{A}}|} \end{pmatrix}$;
  If $\boldsymbol{v} \leq \mathbf{0}_{b+|\widehat{\mathcal{A}}|}$ % the final interval
    $\lambda_{\tau+1} \longleftarrow 0$; $\widehat{\boldsymbol{\alpha}}(\lambda_{\tau+1}) \longleftarrow (u_1, \ldots, u_b)^\top$;
  else % an intermediate interval
    $k \longleftarrow \mathrm{argmax}_i\{u_i/v_i \mid v_i > 0,\ i = 1, \ldots, b + |\widehat{\mathcal{A}}|\}$;
    $\lambda_{\tau+1} \longleftarrow \max\{0, u_k/v_k\}$;
    $\widehat{\boldsymbol{\alpha}}(\lambda_{\tau+1}) \longleftarrow (u_1, \ldots, u_b)^\top - \lambda_{\tau+1}(v_1, \ldots, v_b)^\top$;
    If $1 \leq k \leq b$
      $\widehat{\mathcal{A}} \longleftarrow \widehat{\mathcal{A}} \cup \{k\}$;
    else
      $\widehat{\mathcal{A}} \longleftarrow \widehat{\mathcal{A}} \backslash \{j_{k-b}\}$;
    end
  end
  $\tau \longleftarrow \tau + 1$;
end
$$\widehat{\boldsymbol{\alpha}}(\lambda) \longleftarrow \begin{cases} \mathbf{0}_b & \text{if } \lambda \geq \lambda_0 \\ \frac{\lambda_{\tau+1} - \lambda}{\lambda_{\tau+1} - \lambda_\tau} \widehat{\boldsymbol{\alpha}}(\lambda_\tau) + \frac{\lambda - \lambda_\tau}{\lambda_{\tau+1} - \lambda_\tau} \widehat{\boldsymbol{\alpha}}(\lambda_{\tau+1}) & \text{if } \lambda_{\tau+1} \leq \lambda \leq \lambda_\tau \end{cases}$$

Figure 3: Pseudo code for computing the entire regularization path of LSIF. The computation of $\widehat{\boldsymbol{G}}^{-1}$ is sometimes unstable. For stabilization purposes, small positive diagonals may be added to $\widehat{\boldsymbol{H}}$.

$\{\mathcal{X}_k^{\mathrm{te}}\}_{k=1}^R$ and $\{\mathcal{X}_k^{\mathrm{tr}}\}_{k=1}^R$, respectively. Then a density-ratio estimate $\widehat{w}_r(\boldsymbol{x})$ is obtained using $\{\mathcal{X}_k^{\mathrm{te}}\}_{k\neq r}$ and $\{\mathcal{X}_k^{\mathrm{tr}}\}_{k\neq r}$ (i.e., without $\mathcal{X}_r^{\mathrm{te}}$ and $\mathcal{X}_r^{\mathrm{tr}}$), and the cost J is approximated using the hold-out samples $\mathcal{X}_r^{\mathrm{te}}$ and $\mathcal{X}_r^{\mathrm{tr}}$ as

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

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

$$\widehat{J} = \frac{1}{R}\sum_{r=1}^{k} \widehat{J}_r.$$

The LSIF solution $\widehat{\boldsymbol{\alpha}}$ is shown to be piecewise linear with respect to the regularization parameter λ (Kanamori, Hido and Sugiyama, 2009*b*). Therefore, the *regularization path* (i.e., solutions for all λ) can be computed efficiently based on the *parametric optimization technique* (Best, 1982; Efron, Hastie, Johnstone and Tibshirani, 2002; Hastie, Rosset, Tibshirani and Zhu, 2004; Stein, Branke and Schmeck, 2008). A pseudo code of the regularization path tracking algorithm for LSIF is described in Figure 3. This implies that a quadratic programming solver is no longer needed for obtaining the LSIF solution—just computing matrix inverses is enough. This highly contributes to saving the computation time. Furthermore, the regularization path algorithm is computationally very efficient when the solution is sparse, i.e., most of the elements are zero since the number of change points tends to be small for sparse solutions.

An R implementation of the entire LSIF algorithm is available from the following web page:

`http://www.math.cm.is.nagoya-u.ac.jp/~kanamori/software/LSIF/`

3.5 Unconstrained Least-Squares Importance Fitting (uLSIF)

LSIF combined with regularization path tracking is computationally very efficient. However, it sometimes suffers from a numerical problem and therefore is not practically reliable. To cope with this problem, an approximation method called *unconstrained LSIF* (uLSIF) has been introduced (Kanamori, Hido and Sugiyama, 2009*a*; Kanamori, Hido and Sugiyama, 2009*b*).

The original objective function of uLSIF is also the squared error between the true and estimated importance function shown in Eq.(6). Thus the optimization problem for uLSIF is also derived as Eq.(7). The approximation idea is very simple: the non-negativity constraint in the optimization problem (10) is dropped. This results in the following unconstrained optimization problem.

$$\min_{\{\alpha_\ell\}_{\ell=1}^b} \left[\frac{1}{2}\sum_{\ell,\ell'=1}^{b} \alpha_\ell\alpha_{\ell'}\widehat{H}_{\ell,\ell'} - \sum_{\ell=1}^{b}\alpha_\ell\widehat{h}_\ell + \frac{\lambda}{2}\sum_{\ell=1}^{b}\alpha_\ell^2\right]. \tag{11}$$

In the above, a quadratic regularization term $\lambda \sum_{\ell=1}^{b} \alpha_\ell^2/2$ is used instead of the linear one since the linear penalty term does not work as a regularizer without the non-negativity constraint. Eq.(11) is an unconstrained convex quadratic programming problem, so the solution can be analytically computed as

$$\widetilde{\boldsymbol{\alpha}} = (\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_b)^\top = (\widehat{\boldsymbol{H}} + \lambda \boldsymbol{I}_b)^{-1} \widehat{\boldsymbol{h}},$$

where $\boldsymbol{I}_b$ is the b-dimensional identity matrix. Since the non-negativity constraint $\alpha_\ell \geq 0$ is dropped, some of the learned parameters could be negative. To compensate for this approximation error, the solution is modified as

$$\widehat{\alpha}_\ell = \max(0, \widetilde{\alpha}_\ell) \quad \text{for } \ell = 1, \ldots, b. \tag{12}$$

See Kanamori, Hido and Sugiyama (2009*b*) for theoretical error analysis. An advantage of the above unconstrained formulation is that the solution can be computed just by solving a system of linear equations. Therefore, the computation is fast and stable. See Kanamori, Suzuki and Sugiyama (2009) for theoretical analysis of the algorithmic stability.

Another, and more significant advantage of uLSIF is that the score of leave-one-out cross-validation (LOOCV) can be computed analytically—thanks to this property, the computational complexity for performing LOOCV is the same order as just computing a single solution, which is explained below. In the current setting, two sets of samples $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$ and $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ are given, which generally have different sample size. For explaining the idea in a simple manner, we assume that $n_{\mathrm{te}} < n_{\mathrm{tr}}$ and $\boldsymbol{x}_i^{\mathrm{te}}$ and $\boldsymbol{x}_i^{\mathrm{tr}}$ $(i = 1, \ldots, n_{\mathrm{te}})$ are held out at the same time; $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=n_{\mathrm{te}}+1}^{n_{\mathrm{tr}}}$ are always used for density-ratio estimation.

Let $\widehat{w}^{(i)}(\boldsymbol{x})$ be an estimate of the density ratio obtained without $\boldsymbol{x}_i^{\mathrm{te}}$ and $\boldsymbol{x}_i^{\mathrm{tr}}$. Then the LOOCV score is expressed as

$$\mathrm{LOOCV} = \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \left[\frac{1}{2} (\widehat{w}^{(i)}(\boldsymbol{x}_i))^2 - \widehat{w}^{(i)}(\boldsymbol{x}_i') \right]. \tag{13}$$

A key trick to efficiently calculate the LOOCV score is to use the *Sherman-Woodbury-Morrison* formula (Golub and Loan, 1996) for computing matrix inverses. A pseudo code of uLSIF with LOOCV-based model selection is summarized in Figure 4. MATLAB® and R implementations of the entire uLSIF algorithm are available from the following web pages:

`http://sugiyama-www.cs.titech.ac.jp/~sugi/software/uLSIF/`
`http://www.math.cm.is.nagoya-u.ac.jp/~kanamori/software/LSIF/`

4 Outlier Detection by uLSIF

In this section, we discuss the characteristics of importance estimation methods reviewed in the previous section and propose a practical outlier detection procedure based on uLSIF.

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

$b \longleftarrow \min(100, n_{\mathrm{tr}})$; $\quad \overline{n} = \min(n_{\mathrm{te}}, n_{\mathrm{tr}})$;
Randomly choose b centers $\{\boldsymbol{c}_\ell\}_{\ell=1}^{b}$ from $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$;
For each candidate of Gaussian width σ

$$\widehat{H}_{\ell,\ell'} = \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \exp\left(-\frac{\|\boldsymbol{x}_i^{\mathrm{te}} - \boldsymbol{c}_\ell\|^2 + \|\boldsymbol{x}_i^{\mathrm{te}} - \boldsymbol{c}_{\ell'}\|^2}{2\sigma^2}\right) \text{ for } \ell, \ell' = 1, \ldots, b;$$

$$\widehat{h}_\ell = \frac{1}{n_{\mathrm{tr}}} \sum_{j=1}^{n_{\mathrm{tr}}} \exp\left(-\frac{\|\boldsymbol{x}_j^{\mathrm{tr}} - \boldsymbol{c}_\ell\|^2}{2\sigma^2}\right) \text{ for } \ell = 1, \ldots, b;$$

$$X_{\ell,i} \longleftarrow \exp\left(-\frac{||\boldsymbol{x}_i^{\mathrm{te}} - \boldsymbol{c}_\ell||^2}{2\sigma^2}\right) \text{ for } i = 1, \ldots, \overline{n} \text{ and } \ell = 1, \ldots, b;$$

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

For each candidate of regularization parameter λ

$$\widehat{\boldsymbol{B}} \longleftarrow \widehat{\boldsymbol{H}} + \frac{\lambda(n_{\mathrm{te}} - 1)}{n_{\mathrm{te}}} \boldsymbol{I}_b;$$

$$\boldsymbol{B}_0 \longleftarrow \widehat{\boldsymbol{B}}^{-1}\widehat{\boldsymbol{h}}\mathbf{1}_{\overline{n}}^\top + \widehat{\boldsymbol{B}}^{-1}\boldsymbol{X}\,\mathrm{diag}\left(\frac{\widehat{\boldsymbol{h}}^\top \widehat{\boldsymbol{B}}^{-1}\boldsymbol{X}}{n_{\mathrm{te}}\mathbf{1}_{\overline{n}}^\top - \mathbf{1}_b^\top(\boldsymbol{X} * \widehat{\boldsymbol{B}}^{-1}\boldsymbol{X})}\right);$$

$$\boldsymbol{B}_1 \longleftarrow \widehat{\boldsymbol{B}}^{-1}\boldsymbol{X}' + \widehat{\boldsymbol{B}}^{-1}\boldsymbol{X}\,\mathrm{diag}\left(\frac{\mathbf{1}_b^\top(\boldsymbol{X}' * \widehat{\boldsymbol{B}}^{-1}\boldsymbol{X})}{n_{\mathrm{te}}\mathbf{1}_{\overline{n}}^\top - \mathbf{1}_b^\top(\boldsymbol{X} * \widehat{\boldsymbol{B}}^{-1}\boldsymbol{X})}\right);$$

$$\boldsymbol{B}_2 \longleftarrow \max\left(\boldsymbol{O}_{b\times\overline{n}},\ \frac{n_{\mathrm{te}} - 1}{n_{\mathrm{te}}(n_{\mathrm{tr}} - 1)}(n_{\mathrm{tr}}\boldsymbol{B}_0 - \boldsymbol{B}_1)\right);$$

$$\boldsymbol{r} \longleftarrow (\mathbf{1}_b^\top(\boldsymbol{X} * \boldsymbol{B}_2))^\top; \quad \boldsymbol{r}' \longleftarrow (\mathbf{1}_b^\top(\boldsymbol{X}' * \boldsymbol{B}_2))^\top;$$

$$\mathrm{LOOCV}(\sigma, \lambda) \longleftarrow \frac{\boldsymbol{r}^\top \boldsymbol{r}}{2\overline{n}} - \frac{\mathbf{1}_{\overline{n}}^\top \boldsymbol{r}_{\mathrm{tr}}}{\overline{n}};$$

end
end

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

$$\widetilde{H}_{\ell,\ell'} = \frac{1}{n_{\mathrm{te}}} \sum_{i=1}^{n_{\mathrm{te}}} \exp\left(-\frac{\|\boldsymbol{x}_i^{\mathrm{te}} - \boldsymbol{c}_\ell\|^2 + \|\boldsymbol{x}_i^{\mathrm{te}} - \boldsymbol{c}_{\ell'}\|^2}{2\widehat{\sigma}^2}\right) \text{ for } \ell, \ell' = 1, \ldots, b;$$

$$\widetilde{h}_\ell = \frac{1}{n_{\mathrm{tr}}} \sum_{j=1}^{n_{\mathrm{tr}}} \exp\left(-\frac{\|\boldsymbol{x}_j^{\mathrm{tr}} - \boldsymbol{c}_\ell\|^2}{2\widehat{\sigma}^2}\right) \text{ for } \ell = 1, \ldots, b;$$

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

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

Figure 4: Pseudo code of uLSIF with LOOCV. $\boldsymbol{B} * \boldsymbol{B}'$ denotes the element-wise multiplication of matrices $\boldsymbol{B}$ and $\boldsymbol{B}'$ of the same size. For n-dimensional vectors $\boldsymbol{b}$ and $\boldsymbol{b}'$, $\mathrm{diag}\left(\frac{\boldsymbol{b}}{\boldsymbol{b}'}\right)$ denotes the $n \times n$ diagonal matrix with the i-th diagonal element b_i/b'_i.

Table 1: Relation between direct density ratio estimation methods.

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

4.1 Discussions

For KMM, there is no objective model selection method. Therefore, model parameters such as the Gaussian width need to be determined by hand, which is highly subjective in outlier detection. On the other hand, LogReg and KLIEP give an estimate of the entire importance function. Therefore, the importance values at unseen points can be estimated and CV becomes available for model selection. However, LogReg and KLIEP are computationally rather expensive since non-linear optimization problems have to be solved. LSIF has qualitatively similar properties to LogReg and KLIEP, but it is advantageous over LogReg and KLIEP in that it is equipped with a regularization path tracking algorithm. Thanks to this, model selection of LSIF is computationally much more efficient than LogReg and KLIEP. However, the regularization path tracking algorithm tends to be numerically unstable.

Table 1 summarizes the characteristics of the direct density ratio estimation methods. uLSIF inherits the preferable properties of LogReg, KLIEP, and LSIF, i.e., it can avoid density estimation, model selection is possible, and non-linear optimization is involved. Furthermore, the solution of uLSIF can be computed analytically through matrix inversion and therefore uLSIF is computationally very efficient. Thanks to the availability of the closed-form solution, the LOOCV score can also be analytically computed without repeating the hold-out loop, which highly contributes to reducing the computation time in the model selection phase.

Based on the above discussion, we decided to use uLSIF in our outlier detection procedure.

4.2 Heuristic of Basis Function Choice

In uLSIF, a good model may be chosen by LOOCV, given that a set of promising model candidates is prepared. Here we propose to use a Gaussian kernel model centered at the training points $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ as model candidates, i.e.,

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

where $K_\sigma(\boldsymbol{x}, \boldsymbol{x}')$ is the Gaussian kernel (1) with kernel width σ.

The reason why the training points $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ are chosen as the Gaussian centers, not the test points $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$, is as follows. By definition, the importance $w(\boldsymbol{x})$ tends to take large values if the training density $p_{\mathrm{tr}}(\boldsymbol{x})$ is large and the test density $p_{\mathrm{te}}(\boldsymbol{x})$ is small; conversely, $w(\boldsymbol{x})$ tends to be small (i.e., close to zero) if $p_{\mathrm{tr}}(\boldsymbol{x})$ is small and $p_{\mathrm{te}}(\boldsymbol{x})$ is large. When a function is approximated by a Gaussian kernel model, many kernels may be needed in the region where the output of the target function is large; on the other hand, only a small number of kernels would be enough in the region where the output of the target function is close to zero. Following this heuristic, we decided to allocate many kernels at high training density regions, which can be achieved by setting the Gaussian centers at the training points $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$.

Alternatively, we may locate $(n_{\mathrm{tr}}+n_{\mathrm{te}})$ Gaussian kernels at both $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ and $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$. However, in our preliminary experiments, this did not further improve the performance, but just slightly increased the computational cost. Since n_{tr} is typically very large, just using all the training points $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ as Gaussian centers is already computationally rather demanding. To ease this problem, we practically propose to use a subset of $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ as Gaussian centers for computational efficiency, i.e., for some b such that $1 \leq b \leq n_{\mathrm{tr}}$,

$$\widehat{w}(\boldsymbol{x}) = \sum_{\ell=1}^{b} \alpha_\ell K_\sigma(\boldsymbol{x}, \boldsymbol{c}_\ell),$$

where $\boldsymbol{c}_\ell$ is a template point randomly chosen from $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$.

We use the above basis functions in LogReg, KLIEP, and uLSIF in the experiments.

4.3 Illustrative Examples

Here, we illustrate how uLSIF behaves in inlier-based outlier detection.

4.3.1 Toy Dataset

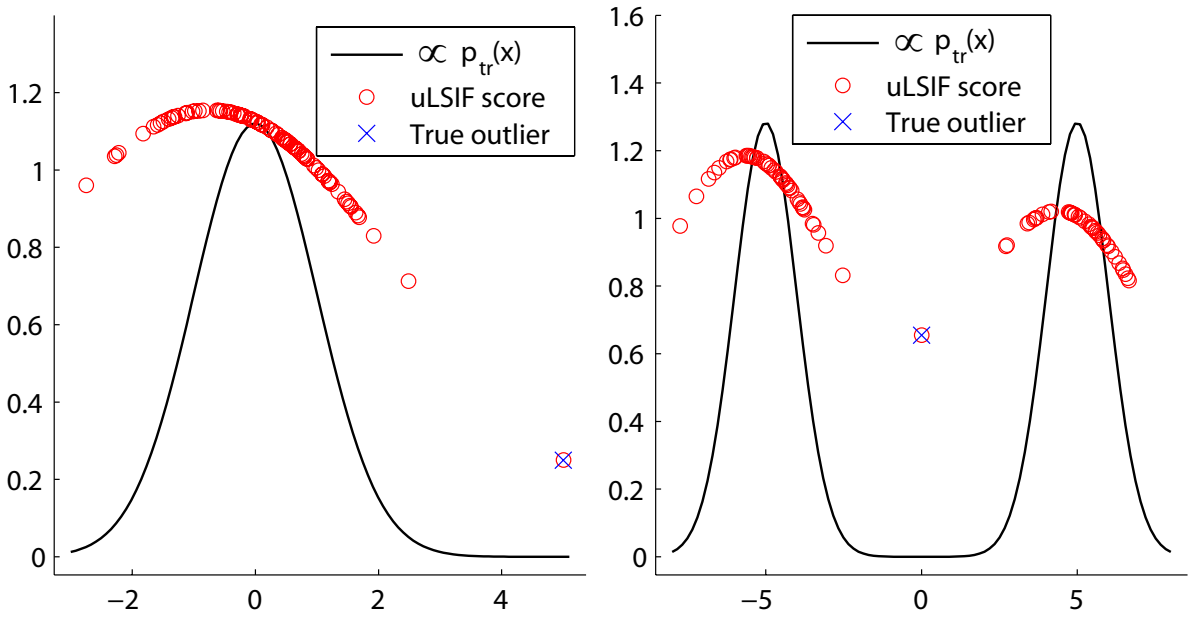
Let the dimension of the data domain be $d = 1$, and let the training density be

(a) $p_{\mathrm{tr}}(x) = \mathcal{N}(x; 0, 1)$,

(b) $p_{\mathrm{tr}}(x) = 0.5\mathcal{N}(x; -5, 1) + 0.5\mathcal{N}(x; 5, 1)$,

where $\mathcal{N}(x; \mu, \sigma^2)$ denotes the Gaussian density with mean μ and variance σ^2. We draw $n_{\mathrm{tr}} = 300$ training samples and 99 test samples from $p_{\mathrm{tr}}(x)$, and we add an outlier sample at $x = 5$ for the case (a) and at $x = 0$ for the case (b) to the test set; thus the total number of test samples is $n_{\mathrm{te}} = 100$. The number of basis functions in uLSIF is fixed to $b = 100$, and the Gaussian width σ and the regularization parameter λ are chosen from a wide range of values based on LOOCV.

The data densities as well as the importance values (i.e., the inlier scores) obtained by uLSIF are depicted in Figure 5. The graphs show that the outlier sample has the smallest inlier score among all samples and therefore the outlier can be successfully detected. Since the solution of uLSIF tends to be sparse, it may be natural to have a Gaussian-like profile as the inlier score (see Figure 5 again).



(a) $\mathcal{N}(x; 0, 1)$ (b) $\frac{1}{2}\mathcal{N}(x; -5, 1) + \frac{1}{2}\mathcal{N}(x; 5, 1)$

Figure 5: Illustration of uLSIF-based outlier detection.

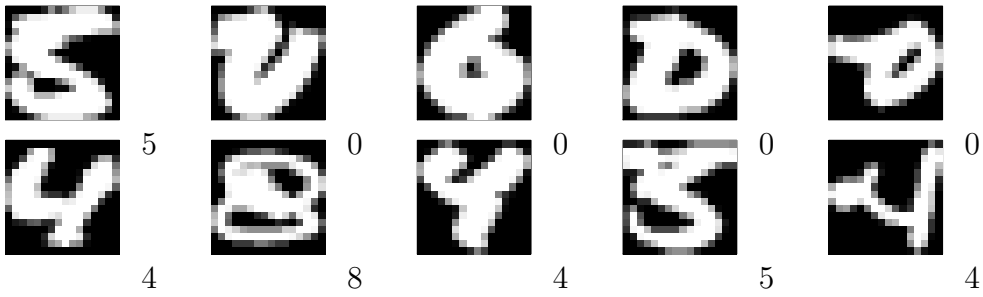


Figure 6: Outliers in the USPS test set.

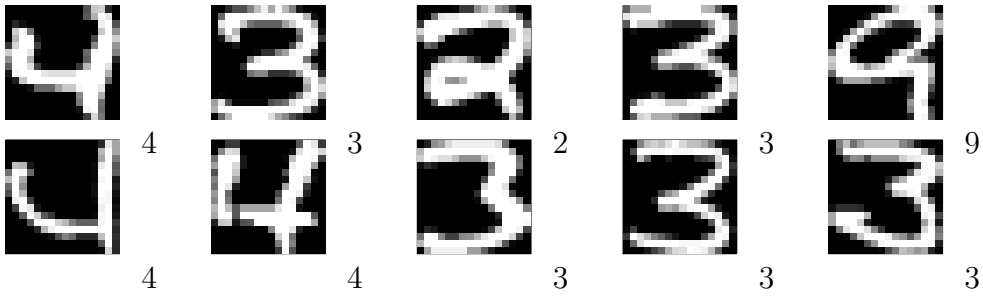


Figure 7: 'Outliers' in the USPS training set.

4.3.2 USPS Dataset

The USPS dataset contains images of hand-written digits provided by U.S. Postal Service. Each digit image consists of 256 ($= 16 \times 16$) pixels, each of which takes a value between -1 to $+1$ representing its color in gray-scale. The class labels attached to the images are integers between 0 and 9 denoting the digits the images represent. Here, we try to find irregular samples in the USPS dataset by uLSIF.

To the 256-dimensional image vectors, we append 10 additional dimensions indicating the true class to identify mislabeled images. In uLSIF, we set $b = 100$ and σ and λ are chosen from a wide range of values based on LOOCV. Figure 6 shows the top 10 outlier samples in the USPS test set (of size 2007) found by uLSIF (from left-top to right-bottom, the outlier rank goes from 1 to 10); the original labels are attached next to the images. This result clearly shows that the proposed method successfully detects outlier samples which are very hard to recognize even by humans.

Let us also consider an inverse scenario: we switch the training and test sets and examine the USPS training set (of size 7291). Figure 7 depicts the top 10 outliers found by uLSIF, showing that they are relatively 'good' samples. This implies that the USPS training set consists only of high-quality samples.

5 Relation to Existing Outlier Detection Methods

In this section, we discuss the relation between the proposed density-ratio based outlier detection approach and existing outlier detection methods.

The outlier detection problem we are addressing in this paper is to find outliers in the test set $\{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}$ based on the training set $\{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}}$ consisting only of inliers. On the other hand, the outlier detection problem that the existing methods reviewed here are solving is to find outliers in the test set without the training set. Thus the setting is slightly different. However, the existing methods can also be employed in our setting by simply using the union of training and test samples as a test set:

$$\{\boldsymbol{x}_k\}_{k=1}^{n} = \{\boldsymbol{x}_j^{\mathrm{tr}}\}_{j=1}^{n_{\mathrm{tr}}} \cup \{\boldsymbol{x}_i^{\mathrm{te}}\}_{i=1}^{n_{\mathrm{te}}}.$$

5.1 Kernel Density Estimator (KDE)

KDE is a non-parametric technique to estimate a density $p(\boldsymbol{x})$ from samples $\{\boldsymbol{x}_k\}_{k=1}^{n}$. KDE with the Gaussian kernel is expressed as

$$\widehat{p}(\boldsymbol{x}) = \frac{1}{n(2\pi\sigma^2)^{d/2}} \sum_{k=1}^{n} K_\sigma(\boldsymbol{x}, \boldsymbol{x}_k),$$

where $K_\sigma(\boldsymbol{x}, \boldsymbol{x}')$ is the Gaussian kernel (1).

The performance of KDE depends on the choice of the kernel width σ, but its value can be objectively determined based on LCV (Härdle et al., 2004): a subset of $\{\boldsymbol{x}_k\}_{k=1}^{n}$

is used for density estimation and the rest is used for estimating the likelihood of the hold-out samples. Note that this LCV procedure corresponds to choosing σ such that the Kullback-Leibler divergence from $p(\boldsymbol{x})$ to $\widehat{p}(\boldsymbol{x})$ is minimized. The estimated density values could be directly used as an inlier score. A variation of the KDE approach has been studied in Latecki, Lazarevic and Pokrajac (2007), where local outliers are detected from multi-modal datasets.

However, kernel density estimation is known to suffer from the *curse of dimensionality* (Vapnik, 1998), and therefore the KDE-based outlier detection method may not be reliable in practice.

The density ratio can also be estimated by KDE, i.e., first estimating the training and test densities separately and then taking the ratio of the estimated densities. However, the estimation error tends to be accumulated in this two-step procedure and our preliminary experiments showed that this is not useful.

5.2 One-class Support Vector Machine (OSVM)

SVM is one of the most successful classification algorithms in machine learning. The core idea of SVM is to separate samples in different classes by the maximum margin hyperplane in a kernel-induced feature space.

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

$$\begin{aligned}&\min_{\{w_k\}_{k=1}^n} \frac{1}{2}\sum_{k,k'=1}^n w_k w_{k'} K_\sigma(\boldsymbol{x}_k, \boldsymbol{x}_{k'})\\ &\text{s.t.} \sum_{k=1}^n w_k \;=\; 1 \;\;\text{and}\;\; 0 \le w_1, \ldots, w_n \le \frac{1}{\nu n},\end{aligned}$$

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

OSVM inherits the concept of SVM, so it is expected to work well. However, the OSVM solution is dependent on the outlier ratio ν and the Gaussian kernel width σ; choosing these tuning parameter values could be highly subjective in unsupervised outlier detection. This is a critical limitation in practice. Furthermore, inlier scores cannot be directly obtained by OSVM; the distance from the separating hyperplane may be used as an inlier score (we do so in the experiments in Section 6), but its statistical meaning is rather unclear.

A similar algorithm named *Support Vector Data Description* (SVDD) (Tax and Duin, 2004) is known to be equivalent to OSVM if the Gaussian kernel is used.

5.3 Local Outlier Factor (LOF)

LOF is an outlier score suitable for detecting local outliers apart from dense regions (Breunig, Kriegel, Ng and Sander, 2000). The LOF value of a sample $\boldsymbol{x}$ is defined using the ratio of the average distance from the nearest neighbors as

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

where $\mathrm{nearest}_i(\boldsymbol{x})$ represents the i-th nearest neighbor of $\boldsymbol{x}$ and $\mathrm{lrd}_k(\boldsymbol{x})$ denotes the inverse of the average distance from the k nearest neighbors of $\boldsymbol{x}$. If $\boldsymbol{x}$ lies around a high density region and its nearest neighbor samples are close to each other in the high density region, $\mathrm{lrd}_k(\boldsymbol{x})$ tends to become much smaller than $\mathrm{lrd}_k(\mathrm{nearest}_i(\boldsymbol{x}))$ for every i. In such cases, $\mathrm{LOF}_k(\boldsymbol{x})$ has a large value and $\boldsymbol{x}$ is regarded as a local outlier.

Although the LOF values seem to be a suitable outlier measure, the performance strongly depends on the choice of the parameter k. To the best of our knowledge, there is no systematic method to select an appropriate value for k. In addition, the computational cost of the LOF scores is expensive since it involves a number of nearest neighbor search procedures.

5.4 Learning from Positive and Unlabeled Data

A formulation called *learning from positive and unlabeled data* has been introduced in Liu, Dai, Li, Lee and Yu (2003): given positive and unlabeled datasets, the goal is to detect positive samples contained in the unlabeled dataset. The assumption behind this formulation is that most of the unlabeled samples are negative (outlier) samples, which is different from the current outlier detection setup. In Li, Liu and Ng (2007), a modified formulation has been addressed in the context of text data analysis—the unlabeled dataset contains only a small number of negative documents. The key idea is to construct a single representative document of the negative (outlier) class based on the difference between the distributions of positive and unlabeled documents. Although the problem setup is similar to ours, the method is specialized in text data, i.e., the *bag-of-words* expression.

Since the above methods of learning from positive and unlabeled data do not fit general inlier-based outlier detection scenarios, we will not include them in the experiments in Section 6.

5.5 Discussions

In summary, the proposed density-ratio based approach with direct density-ratio estimation would be more advantageous than KDE since it allows us to avoid density estimation which is known to be a hard task. Compared with OSVM and LOF, the density-ratio based approach with uLSIF (and also LogReg, KLIEP, and LSIF) would be more useful since it is equipped with a model selection procedure. Furthermore, uLSIF is computationally more efficient than OSVM and LOF thanks to the analytic-form solution.

6 Experiments

In this section, we experimentally compare the performance of the proposed and existing outlier detection algorithms. For all experiments, we use the statistical language environment *R* (R Development Core Team, 2008). We implemented uLSIF, KLIEP, LogReg, KDE, and KMM by ourselves. uLSIF and KLIEP are implemented following Kanamori, Hido and Sugiyama (2009*b*) and Sugiyama, Suzuki, Nakajima, Kashima, von Bünau and Kawanabe (2008), respectively. A package of the *L-BFGS-B* method called *optim* is used in our LogReg implementation, and a quadratic program solver called *ipop* contained in the *kernlab* package (Karatzoglou, Smola, Hornik and Zeileis, 2004) is used in our KMM implementation. We use the *ksvm* function contained in the *kernlab* package for OSVM and the *lofactor* function included in the *dprep* package (Fernandez, 2005) for LOF.

6.1 Benchmark Datasets

We use 12 datasets available from Rätsch's Benchmark Repository (Rätsch, Onoda and Müller, 2001). Note that they are originally binary classification datasets—here we regard the positive samples as inliers and the negative samples as outliers. All the negative samples are removed from the training set, i.e., the training set only contains inlier samples. In contrast, a fraction ρ of randomly chosen negative samples are retained in the test set, i.e., the test set includes all inlier samples and some outliers.

When evaluating the performance of outlier detection algorithms, it is important to take into account both the *detection rate* (the amount of true outliers an outlier detection algorithm can find) and the *detection accuracy* (the amount of true inliers that an outlier detection algorithm misjudges as outliers). Since there is a trade-off between the detection rate and detection accuracy, we decided to adopt the *Area Under the ROC Curve* (AUC) (Bradley, 1997) as our error metric here.

We compare the AUC values of the density-ratio based methods (KMM, LogReg, KLIEP, and uLSIF) and other methods (KDE, OSVM, and LOF). All the tuning parameters included in LogReg, KLIEP, uLSIF, and KDE are chosen based on CV from a wide range of values. CV is not available to KMM, OSVM, and LOF; the Gaussian kernel width in KMM and OSVM is set as the median distance between samples, which has been shown to be a useful heuristic[1] (Schölkopf and Smola, 2002). For KMM, we fix the other tuning parameters at $B = 1000$ and $\epsilon = (\sqrt{n_{\mathrm{te}}} - 1)/\sqrt{n_{\mathrm{te}}}$ following Huang et al. (2007). For OSVM, we fix the tuning parameter at $\nu = 0.1$. The number of basis functions in uLSIF is fixed to $b = 100$. Note that b can also be optimized by CV, but our preliminary experimental results showed that the performance is not so sensitive to the choice of b and $b = 100$ seems to be a reasonable choice. For LOF, we test 3 different values for the number k of nearest neighbors.

The mean AUC values over 20 trials as well as the computation time are summarized in Table 2, where the value is normalized so that the computation time of uLSIF is one. Since the type of outliers may be diverse depending on the datasets, no single

[1] We experimentally confirmed that this heuristic works reasonably well in the current experiments.

Table 2: Mean AUC values over 20 trials for the benchmark datasets.

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

best may consistently outperform the others for all the datasets. To evaluate the overall performance, we included the averaged AUC values over all datasets at the bottom of Table 2.

The results show that uLSIF works fairly well on the whole. KLIEP tends to perform similarly to uLSIF since the same linear model is used for importance estimation. However, uLSIF is computationally much more efficient than KLIEP. LogReg overall works reasonably well, but it performs poorly for some datasets such as splice, twonorm, and waveform, and the average AUC performance is not as good as uLSIF or KLIEP.

KMM and OSVM are not comparable to uLSIF in AUC and they are computationally inefficient. Note that we also tested KMM and OSVM with several different Gaussian widths and experimentally found that the heuristic of using the median sample distance as the Gaussian kernel width works reasonably well in this experiment. Thus the AUC values of KMM and OSVM are expected to be close to their optimal values. LOF with large k is shown to work well, although it is not clear whether the heuristic of simply using large k is always appropriate. In fact, the average AUC values of LOF is slightly higher for $k = 30$ than $k = 50$ and there is no systematic way to choose the optimal value for k. LOF is computationally demanding since nearest neighbor search is expensive. KDE sometimes works reasonably well, but the performance fluctuates depending on the dataset. Therefore, its averaged AUC value is not as good as uLSIF and KLIEP.

Overall, the proposed uLSIF-based method could be regarded a reliable and computationally efficient alternative to existing outlier detection methods.

6.2 SMART Datasets

Next, let us consider a real-world failure prediction problem in hard-disk drives equipped with the *Self-Monitoring and Reporting Technology* (SMART). The SMART system monitors individual drives and stores some attributes (e.g., the number of read errors) as time-series data. We use the SMART dataset provided by a manufacturer (Murray, Hughes and Kreutz-Delgado, 2005). The dataset consists of 369 drives, where 178 drives are labeled as 'good' and 191 drives are labeled as 'failed'. Each drive stores up to the last 300 records which are logged almost every 2 hours. Although each record originally includes 59 attributes, we use only 25 variables chosen based on the feature selection test following Murray et al. (2005). The sequence of records are converted into data samples in a sliding-window manner with window size ℓ.

In practice, undetected defects may exist in the training set. In order to simulate such realistic situations, we add a small fraction τ of '*before-fail*' samples to the training set in addition to the records of the 178 good drives; the before-fail samples are taken from the 191 failed drives more than 300 hours prior to failure. The test set is made of the records of the good drives and the records of the 191 failed drives less than 100 hours prior to failure; the samples corresponding to the failed drives are regarded as outliers in this experiment.

First, we perform experiments for the window size $\ell = 5, 10$ and evaluate the dependence of the feature dimension on the outlier detection performance. The fraction τ of

Table 3: SMART dataset: mean AUC values when changing the window size ℓ and the outlier ratio ρ

Dataset		uLSIF	KLIEP	LogReg	KMM	OSVM	LOF			KDE
ℓ	ρ	(CV)	(CV)	(CV)	(med)	(med)	$k=5$	$k=30$	$k=50$	(CV)
5	0.01	0.894	0.842	0.851	0.822	0.919	0.854	0.937	0.933	0.918
	0.02	0.870	0.810	0.862	0.813	0.896	0.850	0.934	0.928	0.892
	0.05	0.885	0.858	0.888	0.849	0.864	0.789	0.911	0.923	0.883
10	0.01	0.868	0.805	0.827	0.889	0.812	0.880	0.925	0.920	0.557
	0.02	0.879	0.845	0.852	0.894	0.785	0.860	0.919	0.917	0.546
	0.05	0.889	0.857	0.856	0.898	0.783	0.849	0.915	0.916	0.619
Average		0.881	0.836	0.856	0.861	0.843	0.847	0.924	0.923	0.736
Comp. time		1.00	1.07	3.11	4.36	26.98	65.31			2.19

Table 4: SMART dataset: mean AUC values when changing heterogeneousness τ ($\rho = 0.05$ and $\ell = 10$)

Dataset	uLSIF	KLIEP	LogReg	KMM	OSVM	LOF			KDE
τ	(CV)	(CV)	(CV)	(med)	(med)	$k=5$	$k=30$	$k=50$	(CV)
0.05	0.889	0.857	0.856	0.898	0.783	0.849	0.915	0.916	0.619
0.10	0.885	0.856	0.846	0.890	0.785	0.846	0.841	0.914	0.618
0.15	0.868	0.814	0.785	0.886	0.784	0.831	0.835	0.899	0.536
0.20	0.870	0.815	0.778	0.872	0.749	0.847	0.866	0.838	0.540
Average	0.878	0.836	0.816	0.887	0.775	0.843	0.864	0.892	0.578
Comp. time	1.00	1.19	3.78	5.68	30.83	74.30			2.76

before-fail samples in the training set is fixed to 0.05. Other settings including the fraction ρ of outliers and the number b of basis functions are the same as the previous experiments. The results are summarized in Table 3. It shows that the density ratio based methods work well overall; among them, uLSIF has the highest accuracy with the lowest computational cost. Their performance tends to be increased as the outlier fraction ρ increases. On the other hand, the performance of OSVM, LOF, and KDE tends to be degraded as ρ increases. Furthermore, they (especially OSVM and KDE) perform poorly when the feature dimension ℓ increases. This indicates that the density-ratio based methods are more robust to the high dimensionality of the dataset. LOF also works very well if the number of nearest neighbors k is chosen appropriately. However, a good choice of k may be problem-dependent and the computation time of LOF is very slow due to extensive nearest neighbor search.

Next, we change the fraction of before-fail samples in the training set as $\tau = 0.05, 0.10, 0.15, 0.20$ and evaluate the effect of heterogeneousness of the training set on the outlier detection performance. The fraction ρ of outliers in the test set is fixed to 0.05 and the window size ℓ is fixed to 10. Table 4 summarizes the results and shows that the density-ratio based methods still work well. Compared to them, OSVM and KDE perform poorly. Though LOF with $k = 50$ shows the best average value, its computation is slow and the performance is unstable when the fraction τ is changed. Indeed, the performance of LOF with $k = 30$ and $k = 50$ tends to be degraded if the fraction τ of before-fail samples in the training set is increased. This implies that noisy samples in the training set degrade the performance of LOF. On the other hand, uLSIF and KMM are stable even when τ increases. The performance of KMM is slightly better than that of uLSIF, though uLSIF is much faster.

6.3 In-house Financial Datasets

Finally, we use an in-house real-world dataset (named '*RealF*') which we acquired from loan business. A sample in the RealF dataset corresponds to transaction data of a customer for 7 months, which consists of 11-dimensional features. Each customer is labeled according to his/her risk, 'low' or 'high', determined after 6 months of transactions.

Similarly to the previous experiments, we first separate the samples into the positive (low risk) and negative (high risk) ones. Then 200 training samples are randomly taken from the positive dataset, and the test set consisting of randomly chosen 1000 positive samples and a fraction ρ of negative samples are formed. Since the true outlier ratio (the ratio of high-risk customers in the population) is around 5% in real-world loan business, we test $\rho = 0.03, 0.05, 0.07$ in the experiments. Our task is to detect high-risk customers in the test set given the training set including only low-risk customers. This is a highly important problem in practice since loan companies can take precautions against risks by, e.g., limiting the maximum amount of debt of suspicious customers based on their outlier scores.

We perform experiments for 7-month data and 4-month data—the experiment using 4-month data corresponds to early detection of high-risk customers, which is more im-

Table 5: RealF dataset: mean AUC values when changing the data period and the outlier ratio ρ

Dataset		uLSIF	KLIEP	LogReg	KMM	OSVM	LOF			KDE
#Month	ρ	(CV)	(CV)	(CV)	(med)	(med)	$k=5$	$k=30$	$k=50$	(CV)
7	0.03	0.671	0.643	0.390	0.565	0.483	0.666	0.659	0.667	0.636
	0.05	0.672	0.681	0.397	0.492	0.503	0.643	0.663	0.673	0.667
	0.07	0.677	0.664	0.391	0.514	0.505	0.639	0.669	0.685	0.669
4	0.03	0.628	0.625	0.394	0.457	0.495	0.640	0.611	0.622	0.630
	0.05	0.634	0.646	0.394	0.444	0.496	0.608	0.616	0.627	0.635
	0.07	0.640	0.648	0.382	0.473	0.509	0.608	0.622	0.633	0.648
Average		0.654	0.651	0.391	0.491	0.498	0.634	0.640	0.651	0.647
Comp. time		1.00	1.62	2.68	419.66	32.66	36.73			0.99

portant and challenging in practice. The mean AUC values and the computation time are summarized in Table 5. The results show that uLSIF performs excellently both in accuracy and computation time. KLIEP has comparable accuracy to uLSIF with a slight increase in computation time. On the other hand, LogReg, KMM, and OSVM perform poorly for the RealF dataset for all choices of #Month and ρ, indicating that these algorithms tend to fail in recognizing the distribution of the high-risk customers in the dataset. LOF works well, but its performance again depends on the choice of the parameter k and it is computationally expensive. KDE works stable and quite well for this dataset with slightly lower accuracy than uLSIF. As the outlier fraction ρ increases, the AUC values of uLSIF, KLIEP, and KDE tend to be increased. Their performance tends to be better for the 7-month data than for the 4-month data since the detection model can benefit from the additional information included in the 7-month data. Thus the accuracy will be further improved if customers' record data longer than 7 months is used.

These results indicate that our algorithm using the density ratio is accurate and computationally efficient in real-world failure prediction tasks—in particular, the use of uLSIF seems promising both in accuracy and computational efficiency.

7 Concluding Remarks

We have cast the inlier-based outlier detection problem as a problem of estimating the ratio of probability densities (i.e., the *importance*). The basic assumption behind our framework is that a data sample lying in the region where the test input density significantly exceeds the training input density is plausible to be an outlier. Our framework requires estimating the density ratio, but accurate estimation of probability density functions is difficult especially when the data has neither low dimensionality nor a simple distribution (e.g., the Gaussian distribution). To avoid density estimation, we proposed a practical outlier detection algorithm based on direct density ratio estimation methods including unconstrained least-squares importance fitting (uLSIF).

In uLSIF, the density ratio is modeled by a linear model and the squared loss is used for density-ratio function fitting (Kanamori, Hido and Sugiyama, 2009*a*; Kanamori, Hido and Sugiyama, 2009*b*). The solution of the optimization problem of uLSIF can be computed analytically through matrix inversion and therefore uLSIF is computationally very efficient. uLSIF is equipped with a variant of cross-validation (CV), so the values of tuning parameters such as the regularization parameter can be objectively determined without subjective trial and error. Therefore, we can obtain a purely objective solution to the outlier detection problem. This is highly important in unsupervised settings where no prior knowledge is usually available. Furthermore, the uLSIF-based outlier detection method allows us to compute the outlier score just by solving a system of linear equations—the leave-one-out cross-validation (LOOCV) error can also be computed analytically. Thus, the uLSIF-based method is computationally very efficient and therefore is scalable to massive datasets.

Through extensive simulations with benchmark and real-world datasets, the usefulness of the proposed approach was demonstrated. The experimental results for the UCI datasets showed that uLSIF and KLIEP work very well in terms of accuracy. Although other methods also performed well for some datasets, they also exhibited poor performance in other cases. On the other hand, the performance of uLSIF and KLIEP was shown to be relatively stable over various datasets. In addition, from the viewpoint of computation time, uLSIF was shown to be much faster than KLIEP and other methods. In the experiment on the SMART disk-failure datasets, uLSIF was shown to be competitive to the best method LOF in accuracy, but is computationally much more efficient than LOF. For the in-house financial datasets, uLSIF was shown to be the most accurate and the fastest among the methods we have tested. Based on the experimental results, we conclude that the proposed uLSIF-based method should be regarded as a reliable and computationally efficient alternative to existing outlier detection methods.

Independently of our work, a similar method of outlier detection based on the density ratio has been proposed recently (Smola, Song and Teo, 2009) and shown to work well. Also, an earlier report showed that our method is useful in visual inspection of real-world precision instruments (Takimoto, Matsugu and Sugiyama, 2009). Thus the density-ratio method would be a promising approach to outlier detection.

MATLAB® implementations of the uLSIF- and KLIEP-based outlier detection methods (which are referred to as *least-squares outlier detection* and *maximum likelihood outlier detection*) are available from the following web pages:

```
http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LSOD/
http://sugiyama-www.cs.titech.ac.jp/~sugi/software/MLOD/
```

As shown in this paper, the density ratio plays a crucial role in outlier detection. A similar technique may be used for online detection of change points in time series (Kawahara and Sugiyama, 2009). These methods may be regarded as a new approach to the traditional likelihood ratio test. Thus we expect that two sample problems of testing whether two sets of samples are drawn from the same distributions or not could also be successfully approached based on density ratio estimation methods.

Now we can further generalize this line of research—looking at various data processing tasks from the viewpoint of density ratios (Sugiyama, Kanamori, Suzuki, Hido, Sese, Takeuchi and Wang, 2009). *Importance sampling* would be a natural application of the density ratio (Fishman, 1996), where samples taken from one distribution are used for computing the expectation over another distribution. Following this line, *non-stationarity adaptation* based on density ratios has been extensively studied these days (Shimodaira, 2000; Zadrozny, 2004; Sugiyama and Müller, 2005; Sugiyama, Krauledat and Müller, 2007; Quiñonero-Candela, Sugiyama, Schwaighofer and Lawrence, 2009; Sugiyama, von Bünau, Kawanabe and Müller, 2010), and it has been successfully applied to various real-world problems such as brain-computer interface (Sugiyama et al., 2007; Li, Koike and Sugiyama, 2009), robot control (Hachiya, Akiyama, Sugiyama and Peters, 2009; Hachiya, Peters and Sugiyama, 2009), spam filtering (Bickel and Scheffer, 2007), speaker identification (Yamada, Sugiyama and Matsui, 2010), and natural language processing (Tsuboi, Kashima, Hido, Bickel and Sugiyama, 2009). Active learning is also a crucial application of density ratios (Wiens, 2000; Kanamori and Shimodaira, 2003; Sugiyama, 2006; Kanamori, 2007), with successful real-world applications in semi-conductor wafer alignment (Sugiyama and Nakajima, 2009) and robot control (Akiyama, Hachiya and Sugiyama, 2010).

Furthermore, *mutual information*, which plays an important role in information theory (Cover and Thomas, 1991), can be approximated by using density ratio estimation methods (Suzuki, Sugiyama, Sese and Kanamori, 2008; Suzuki, Sugiyama and Tanaka, 2009). Since mutual information allows one to identify statistical independence among random variables, it can be used for various purposes such as *independent component analysis* (Suzuki and Sugiyama, 2009*a*), *feature selection* (Suzuki, Sugiyama, Kanamori and Sese, 2009), and *dimensionality reduction* (Suzuki and Sugiyama, 2009*b*). Density ratio estimation may also be used for *conditional density estimation* since a conditional density can be expressed by the ratio of the joint density and the marginal density (Sugiyama, Takeuchi, Suzuki, Kanamori, Hachiya and Okanohara, 2010).

Thus our important future work is to further improve the accuracy of density ratio estimation, which will highly contribute to enhancing the performance of various algorithms listed above. For example, a density ratio estimation method combined with dimensionality reduction has been proposed in Sugiyama, Kawanabe and Chui (2010), where a supervised dimensionality reduction technique called local Fisher discriminant analysis (Sugiyama, 2007; Sugiyama, Idé, Nakajima and Sese, 2010) is used for identifying a subspace in which two distributions are significantly different. Density ratio estimation beyond linear/kernel models has also been studied, e.g., for log-linear models (Tsuboi et al., 2009) and Gaussian mixture models (Yamada and Sugiyama, 2009). Furthermore, theoretically investigating advantages of direct density ratio estimation beyond Vapnik's principle of avoiding density estimation (Vapnik, 1998) is necessary. Thus research on density ratio estimation would be an emerging and challenging paradigm in data mining and machine learning.

Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments. MS was supported by AOARD, SCAT, and the JST PRESTO program.

References

Akiyama, T., Hachiya, H. and Sugiyama, M. (2010), 'Efficient exploration through active learning for value function approximation in reinforcement learning', *Neural Networks* . to appear.

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

Bickel, S., Brückner, M. and Scheffer, T. (2007), Discriminative learning for differing training and test distributions, *in* 'Proceedings of the 24th International Conference on Machine Learning', pp. 81–88.

Bickel, S. and Scheffer, T. (2007), Dirichlet-enhanced spam filtering based on biased samples, *in* 'Advances in Neural Information Processing Systems 19', MIT Press, Cambridge, MA, pp. 161–168.

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

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

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000), LOF: Identifying density-based local outliers, *in* 'Proceedings of the ACM SIGMOD International Conference on Management of Data', pp. 93–104.

Chan, J., Bailey, J. and Leckie, C. (2008), 'Discovering correlated spatio-temporal changes in evolving graphs', *Knowledge and Information Systems* **16**(1), 53–96.

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

Cover, T. M. and Thomas, J. A. (1991), *Elements of Information Theory*, John Wiley & Sons, Inc., New York, NY, USA.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2002), 'Least angle regression', *Annals of Statistics* **32**, 407–499.

Fan, H., Zaïane, O. R., Foss, A. and Wu, J. (2009), 'Resolution-based outlier factor: detecting the top-n most outlying data points in engineering data', *Knowledge and Information Systems* **19**(1), 31–51.

Fernandez, E. A. (2005), The dprep package, Technical report, University of Puerto Rico. **URL:** *http://math.uprm.edu/~edgar/dprep.pdf*

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

Fujimaki, R., Yairi, T. and Machida, K. (2005), An approach to spacecraft anomaly detection problem using kernel feature space, *in* 'Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 401–410.

Gao, J., Cheng, H. and Tan, P.-N. (2006*a*), A novel framework for incorporating labeled examples into anomaly detection, *in* 'Proceedings of the 2006 SIAM International Conference on Data Mining', pp. 593–597.

Gao, J., Cheng, H. and Tan, P.-N. (2006*b*), Semi-supervised outlier detection, *in* 'Proceedings of the 2006 ACM symposium on Applied Computing', pp. 635–636.

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

Hachiya, H., Akiyama, T., Sugiyama, M. and Peters, J. (2009), 'Adaptive importance sampling for value function approximation in off-policy reinforcement learning', *Neural Networks* **22**(10), 1399–1410.

Hachiya, H., Peters, J. and Sugiyama, M. (2009), Efficient sample reuse in EM-based policy search, *in* W. Buntine, M. Grobelnik, D. Mladenic and J. Shawe-Taylor, eds, 'Machine Learning and Knowledge Discovery in Databases', Vol. 5781 of *Lecture Notes in Computer Science*, Springer, Berlin, pp. 469–484.

Härdle, W., Müller, M., Sperlich, S. and Werwatz, A. (2004), 'Nonparametric and semiparametric models', *Springer Series in Statistics* .

Hastie, T., Rosset, S., Tibshirani, R. and Zhu, J. (2004), 'The entire regularization path for the support vector machine', *Journal of Machine Learning Research* **5**, 1391–1415.

Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M. and Kanamori, T. (2008), Inlier-based outlier detection via direct density ratio estimation, *in* 'Proceedings of the 8th IEEE International Conference on Data Mining', pp. 223–232.

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

Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. and Schölkopf, B. (2007), Correcting sample selection bias by unlabeled data, *in* 'Advances in Neural Information Processing Systems', Vol. 19.

Idé, T. and Kashima, H. (2004), Eigenspace-based anomaly detection in computer systems, *in* 'Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 440–449.

Jiang, X. and Zhu, X. (2009), 'veye: behavioral footprinting for self-propagating worm detection and profiling', *Knowledge and Information Systems* **18**(2), 231–262.

Kanamori, T. (2007), 'Pool-based active learning with optimal sampling distribution and its information geometrical interpretation', *Neurocomputing* **71**(1-3), 353–362.

Kanamori, T., Hido, S. and Sugiyama, M. (2009*a*), Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection, *in* D. Koller, D. Schuurmans, Y. Bengio and L. Bottou, eds, 'Advances in Neural Information Processing Systems 21', MIT Press, pp. 809–816.

Kanamori, T., Hido, S. and Sugiyama, M. (2009*b*), 'A least-squares approach to direct importance estimation', *Journal of Machine Learning Research* **10**, 1391–1445.

Kanamori, T. and Shimodaira, H. (2003), 'Active learning algorithm using the maximum weighted log-likelihood estimator', *Journal of Statistical Planning and Inference* **116**(1), 149–162.

Kanamori, T., Suzuki, T. and Sugiyama, M. (2009), Condition number analysis of kernel-based density ratio estimation, Technical report, arXiv.
URL: *http://www.citebase.org/abstract?id=oai:arXiv.org:0912.2800*

Karatzoglou, A., Smola, A., Hornik, K. and Zeileis, A. (2004), 'kernlab—an S4 package for kernel methods in R', *Journal of Statistical Software* **11**(9), 1–20.

Kawahara, Y. and Sugiyama, M. (2009), Change-point detection in time-series data by direct density-ratio estimation, *in* H. Park, S. Parthasarathy, H. Liu and Z. Obradovic, eds, 'Proceedings of 2009 SIAM International Conference on Data Mining (SDM2009)', Sparks, Nevada, USA, pp. 389–400.

Latecki, L. J., Lazarevic, A. and Pokrajac, D. (2007), Outlier detection with kernel density functions, *in* 'Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition', pp. 61–75.

Li, X., Liu, B. and Ng, S.-K. (2007), Learning to identify unexpected instances in the test set, *in* 'Proceedings of the 20th International Joint Conference on Artificial Intelligence', pp. 2802–2807.

Li, Y., Koike, Y. and Sugiyama, M. (2009), A framework of adaptive brain computer interfaces, *in* 'Proceedings of the 2nd International Conference on BioMedical Engineering and Informatics (BMEI09)', Tianjin, China, pp. 473–4–77.

Liu, B., Dai, Y., Li, X., Lee, W. S. and Yu, P. S. (2003), Building text classifiers using positive and unlabeled examples, *in* 'Proceedings of the 3rd IEEE International Conference on Data Mining', pp. 179–186.

Manevitz, L. M. and Yousef, M. (2002), 'One-class SVMs for document classification', *Journal of Machine Learning Research* **2**, 139–154.

Minka, T. P. (2007), A comparison of numerical optimizers for logistic regression, Technical report, Microsoft Research.

Murray, J. F., Hughes, G. F. and Kreutz-Delgado, K. (2005), 'Machine learning methods for predicting failures in hard drives: A multiple-instance application', *Journal of Machine Learning Research* **6**, 783–816.

Nguyen, X., Wainwright, M. J. and Jordan, M. I. (2008), Estimating divergence functions and the likelihood ratio by penalized convex risk minimization, *in* 'Advances in Neural Information Processing Systems 20', pp. 1089–1096.

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

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A. and Lawrence, N., eds (2009), *Dataset Shift in Machine Learning*, MIT Press, Cambridge, MA.

R Development Core Team (2008), *The R Manuals*. http://www.r-project.org.

Rätsch, G., Onoda, T. and Müller, K. R. (2001), 'Soft margins for AdaBoost', *Machine Learning* **42**(3), 287–320.

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

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

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

Smola, A., Song, L. and Teo, C. H. (2009), Relative novelty detection, *in* 'Proceedings of the 14th International Workshop on Artificial Intelligence and Statistics', Vol. 5, pp. 536–543.

Stein, M., Branke, J. and Schmeck, H. (2008), 'Efficient implementation of an active set algorithm for large-scale portfolio selection', *Computers & Operations Research* **35**(12), 3945–3961.

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

Sugiyama, M. (2006), 'Active learning in approximately linear regression based on conditional expectation of generalization error', *Journal of Machine Learning Research* **7**, 141–166.

Sugiyama, M. (2007), 'Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis', *Journal of Machine Learning Research* **8**, 1027–1061.

Sugiyama, M., Idé, T., Nakajima, S. and Sese, J. (2010), 'Semi-supervised local Fisher discriminant analysis for dimensionality reduction', *Machine Learning* **78**(1–2), 35–61.

Sugiyama, M., Kanamori, T., Suzuki, T., Hido, S., Sese, J., Takeuchi, I. and Wang, L. (2009), 'A density-ratio framework for statistical data processing', *IPSJ Transactions on Computer Vision and Applications* **1**, 183–208.

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

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

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

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

Sugiyama, M., Nakajima, S., Kashima, H., von Bünau, P. and Kawanabe, M. (2008), Direct importance estimation with model selection and its application to covariate shift adaptation, *in* 'Advances in Neural Information Processing Systems 20', pp. 1433–1440.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P. and Kawanabe, M. (2008), 'Direct importance estimation for covariate shift adaptation', *Annals of the Institute of Statistical Mathematics* **60**(4).

Sugiyama, M., Takeuchi, I., Suzuki, T., Kanamori, T., Hachiya, H. and Okanohara, D. (2010), 'Least-squares conditional density estimation', *IEICE Transactions on Information and Systems* **E93-D**(3). to appear.

Sugiyama, M., von Bünau, P., Kawanabe, M. and Müller, K.-R. (2010), *Covariate Shift Adaptation: Towards Machine Learning in Non-Stationary Environment*, MIT Press, Cambridge, MA. to appear.

Suzuki, T. and Sugiyama, M. (2009*a*), Estimating squared-loss mutual information for independent component analysis., *in* T. Adali, C. Jutten, J. M. T. Romano and A. K. Barros, eds, 'Independent Component Analysis and Signal Separation', Vol. 5441 of *Lecture Notes in Computer Science*, Springer, Berlin, pp. 130–137.

Suzuki, T. and Sugiyama, M. (2009*b*), Sufficient dimension reduction via squared-loss mutual information estimation, Technical Report TR09-0005, Department of Computer Science, Tokyo Institute of Technology.
URL: *http://www.cs.titech.ac.jp/*

Suzuki, T., Sugiyama, M., Kanamori, T. and Sese, J. (2009), 'Mutual information estimation reveals global associations between stimuli and biological processes', *BMC Bioinformatics* **10**(1), S52.

Suzuki, T., Sugiyama, M., Sese, J. and Kanamori, T. (2008), Approximating mutual information by maximum likelihood density ratio estimation, *in* Y. Saeys, H. Liu, I. Inza, L. Wehenkel and Y. V. de Peer, eds, 'JMLR Workshop and Conference Proceedings', Vol. 4 of *New Challenges for Feature Selection in Data Mining and Knowledge Discovery*, pp. 5–20.

Suzuki, T., Sugiyama, M. and Tanaka, T. (2009), Mutual information approximation via maximum likelihood estimation of density ratio, *in* 'Proceedings of 2009 IEEE International Symposium on Information Theory (ISIT2009)', Seoul, Korea, pp. 463–467.

Takimoto, M., Matsugu, M. and Sugiyama, M. (2009), Visual inspection of precision instruments by least-squares outlier detection, *in* 'Proceedings of The Fourth International Workshop on Data-Mining and Statistical Science (DMSS2009)', Kyoto, Japan, pp. 22–26.

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

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

Vapnik, V. N. (1998), *Statistical Learning Theory*, Wiley.

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

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

Yamada, M., Sugiyama, M. and Matsui, T. (2010), 'Semi-supervised speaker identification under covariate shift', *Signal Processing* . to appear.

Yamanishi, K., Takeuchi, J.-I., Williams, G. and Milne, P. (2004), 'On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms', *Data Mining and Knowledge Discovery* **8**(3), 275–300.

Yankov, D., Keogh, E. and Rebbapragada, U. (2008), 'Disk aware discord discovery: finding unusual time series in terabyte sized datasets', *Knowledge and Information Systems* **17**(2), 241–262.

Zadrozny, B. (2004), Learning and evaluating classifiers under sample selection bias, *in* 'Proceedings of the Twenty-First International Conference on Machine Learning', ACM Press, New York, NY, pp. 903–910.