# Online Direct Density-Ratio Estimation Applied to Inlier-based Outlier Detection

Marthinus Christoffel du Plessis
The University of Tokyo
christo@ms.k.u-tokyo.ac.jp

Hiroaki Shiino
Tokyo Institute of Technology

Masashi Sugiyama
The University of Tokyo
sugi@k.u-tokyo.ac.jp
http://www.ms.k.u-tokyo.ac.jp

**Abstract**

Many machine learning problems, such as non-stationarity adaptation, outlier detection, dimensionality reduction, and conditional density estimation, can be effectively solved by using the ratio of probability densities. Since the naive two step procedure of first estimating the probability densities and then taking their ratio performs poorly, methods to directly estimate the density ratio from two sets of samples without density estimation have been extensively studied recently. However, these methods are *batch* algorithms that use the whole datasets to estimate the density ratio, and they are inefficient in the *online* setup where training samples are provided sequentially and solutions are updated incrementally without storing previous samples. In this paper, we propose two online density ratio estimators based on the *adaptive regularization of weight vectors*. Through experiments on inlier-based outlier detection, we demonstrate the usefulness of the proposed methods.

**Keywords**

Online learning, density ratio estimation, adaptive regularization of weight vectors, outlier detection

# 1 Introduction

Almost all machine learning problems can be solved through density estimation, because knowing the probability density is equivalent to knowing everything about the data. Thus, density estimation is the most versatile approach to machine learning, and various methods have been proposed so far. However, without strong parametric assumptions, density estimation is hard to perform accurately in high-dimensional problems. Thus, it is desirable to solve target machine learning tasks directly without performing density estimation (Vapnik, 1998).

Following this idea, the machine learning approach based on the ratio of probability densities has attracted attention recently (Sugiyama et al., 2012b). The rationale behind density ratio estimation is that many machine learning problems — such as transfer learning, outlier detection, change detection, and dimension reduction — can be solved in a unified manner using just the density ratio. By directly estimating this density ratio, the difficult task of density estimation can be avoided, leading to better empirical performance.

Due to the the practical utility, several methods for density ratio estimation have been proposed (Kanamori et al., 2009; Sugiyama et al., 2012a; Izbicki et al., 2014). In Sugiyama et al. (2012a) it was shown that density-ratio estimation may be performed by matching the density ratio to a model of the density ratio under a Bregman divergence. This view of density-ratio estimation is of great interest, since it relates to several existing methods, and the resulting estimators can be interpreted in terms of the Bregman divergence. The simplest Bregman divergence corresponds to the squared loss between the density ratio and its model (Kanamori et al., 2009). The main advantage of the least-squares based density ratio estimator is that the solution can be analytically obtained. Another choice for the Bregman divergence is the Kullback-Leibler loss. This Kullback-Leibler based estimator also appears in the variational estimation of the Kullback-Leibler divergence (Nguyen et al., 2010). The main disadvantage of the Kullback-Leibler based estimator is that it does not have a closed-form solution, and optimization is usually performed via gradient or quasi-Newton methods. Although these estimators have been demonstrated to work well on many different problems (Sugiyama et al., 2012b; Sugiyama and Kawanabe, 2012), they work only in a *batch* mode and thus are not efficient in *online* problems where training samples are provided sequentially and solutions are updated incrementally without storing previous samples.

In this paper, we propose online algorithms of the least-squares and Kullback-Leibler based density ratio estimators in the framework of *adaptive regularization of weight vectors* (Crammer et al., 2009), which was originally proposed for regression and classification. We experimentally demonstrate that, for a fixed computational budget, our proposed online algorithms achieve greater performance than both the batch solutions and online solutions via a naive stochastic gradient descent in inlier-based outlier detection.

# 2 Batch Density-Ratio Estimation

In this section, we formulate the batch density-ratio estimation problem, and review density ratio estimation using Bregman divergences.

## 2.1 Problem Formulation

Suppose that we are given a set of independent and identically distributed (i.i.d.) samples $\{\boldsymbol{x}_i\}_{i=1}^n$ from a probability distribution with density $p(\boldsymbol{x})$ and another set of i.i.d. samples $\{\boldsymbol{x}'_j\}_{j=1}^{n'}$ from a probability distribution with density $p'(\boldsymbol{x})$ in the same domain. Under the assumption that $p(\boldsymbol{x}) > 0$ for all $\boldsymbol{x}$, our goal is to estimate the density ratio function,

$$r(\boldsymbol{x}) := \frac{p'(\boldsymbol{x})}{p(\boldsymbol{x})},$$

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

A naive approach is to estimate $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ from $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{x}'_j\}_{j=1}^{n'}$ separately and to take the ratio of the estimated densities. However, such a two-step plug-in approach is not reliable because the first step of density estimation is performed without regards to the second step of taking the ratio (Sugiyama et al., 2012b). Below, we review a direct density-ratio estimation method that does not involve density estimation.

## 2.2 Batch Algorithm

Here, let us review batch density ratio estimation algorithms using Bregman divergences.

### 2.2.1 General Framework with Bregman Divergences

The density ratio can be estimated by minimizing the Bregman divergence between the true density ratio $r^*(\boldsymbol{x})$ and a parameterized model of the density ratio, $r_{\boldsymbol{\theta}}(\boldsymbol{x})$ (Sugiyama et al., 2012a). The Bregman divergence from $t^*$ to $t$ is defined as follows (Bregman, 1967):

$$\mathrm{BR}_f(t^* \| t) = f(t^*) - f(t) - \partial f(t)(t^* - t), \tag{1}$$

where $f(t)$ is a strictly convex function. The above is minimized with respect to $t$ when $t = t^*$. The above distance is useful, since $t^*$ occurs linearly in all terms that include both $t^*$ and $t$. Minimizing the Bregman divergence between the true density ratio $r^*(\boldsymbol{x})$ and a model of the density ratio $r_{\boldsymbol{\theta}}(\boldsymbol{x})$, weighted by $p(\boldsymbol{x})$, gives

$$\begin{aligned}
\mathrm{BR}_f(r^*, r) &= \int \mathrm{BR}_f\Big(r^*(\boldsymbol{x}), r_{\boldsymbol{\theta}}(\boldsymbol{x})\Big) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int p(\boldsymbol{x}) \left( \partial f\Big(r_{\boldsymbol{\theta}}(\boldsymbol{x})\Big) r_{\boldsymbol{\theta}}(\boldsymbol{x}) - f\Big(r_{\boldsymbol{\theta}}(\boldsymbol{x})\Big) \right) \mathrm{d}\boldsymbol{x} \\
&\quad - \int p'(\boldsymbol{x}) \partial f\Big(r_{\boldsymbol{\theta}}(\boldsymbol{x})\Big) + C,
\end{aligned}$$

where $C = \int f\Big(r^*(\boldsymbol{x})\Big)p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$ is constant w.r.t. $\boldsymbol{\theta}$. The empirical version for the above problem is therefore

$$\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} \partial f\Big(r_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\Big) r_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - f\Big(r(\boldsymbol{x}_i)\Big) + \frac{1}{n'}\sum_{j=1}^{n'} \partial f\Big(r_{\boldsymbol{\theta}}(\boldsymbol{x}'_j)\Big) + \frac{1}{2}\lambda\boldsymbol{\theta}^\top\boldsymbol{\theta},$$

where we included a regularization term with regularization parameter $\lambda > 0$.

Let us consider the following linear-in-parameter model for density ratios:

$$r_{\boldsymbol{\theta}}(\boldsymbol{x}) := \sum_{\ell=1}^{b} \theta_\ell \phi_\ell(\boldsymbol{x}) = \boldsymbol{\theta}^\top\boldsymbol{\phi}(\boldsymbol{x}),$$

where $b$ is the number of basis functions,

$$\boldsymbol{\phi}(\boldsymbol{x}) := (\phi_1(\boldsymbol{x}), \ldots, \phi_b(\boldsymbol{x}))^\top$$

is the vector of basis functions,

$$\boldsymbol{\theta} := (\theta_1, \ldots, \theta_b)^\top$$

is the vector of parameters, and $^\top$ denotes the transpose.

A final consideration is choosing a suitable Bregman divergence by defining the function $f(t)$. Below we discuss two such choices for $f(t)$.

### 2.2.2 Kullback-Leibler Approach

Choosing the Bregman divergence as $f(t) = t\log t - t$ results in the Kullback-Leibler (KL) divergence:

$$\mathrm{KL}(t^*\|t) = t - t^*\log t + t^*\log t^* - t^*.$$

The empirical and regularized optimization problem is given by

$$\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\theta}^\top\boldsymbol{\phi}(\boldsymbol{x}_i) - \frac{1}{n'}\sum_{j=1}^{n'} \log \boldsymbol{\theta}^\top\boldsymbol{\phi}(\boldsymbol{x}'_j) + \frac{\lambda}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta},$$

where $\lambda > 0$ is the regularization parameter.

Since the objective function is smooth and convex, we may find the globally optimal solution by a standard optimization technique such as gradient descent or quasi-Newton methods. The gradient of the above objective function with respect to $\boldsymbol{\theta}$ is given by

$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}_i) - \frac{1}{n'}\sum_{j=1}^{n'} \frac{1}{\boldsymbol{\theta}^\top\boldsymbol{\phi}(\boldsymbol{x}'_j)}\boldsymbol{\phi}(\boldsymbol{x}'_j) + \lambda\boldsymbol{\theta}.$$

The regularization parameter $\lambda$ and hyper-parameters included in the basis function $\boldsymbol{\phi}(\boldsymbol{x})$ can be objectively selected via cross-validation with respect to the objective function.

### 2.2.3 Least-Squares Approach

Choosing the Bregman divergence as $f(t) = \frac{1}{2}t^2$ gives the squared loss:

$$\mathrm{LS}\left(t^* \| t\right) = \frac{1}{2}t^* - \frac{1}{2}t^2 - tt^* + t^2 = \frac{1}{2}\left(t^* - t\right)^2.$$

This corresponds to a least-squares fitting of the density ratio model to the true density ratio (Kanamori et al., 2009). The objective function can then be expressed as

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2n}\sum_{i=1}^{n} r_{\boldsymbol{\theta}}(\boldsymbol{x}_i)^2 - \frac{1}{n'}\sum_{j=1}^{n'} r(\boldsymbol{x}'_j) + \frac{\lambda}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta}.$$

By substituting the linear model for $g(\boldsymbol{x})$, this is simplified as a quadratic problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\boldsymbol{\theta}^\top\widehat{\boldsymbol{H}}\boldsymbol{\theta} - \boldsymbol{\theta}^\top\widehat{\boldsymbol{h}} + \frac{\lambda}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta},$$

where $\widehat{\boldsymbol{H}}$ is a $b \times b$ matrix and $\widehat{\boldsymbol{h}}$ is a $b \times 1$ vector given as

$$\widehat{H}_{\ell,\ell'} = \frac{1}{n}\sum_{i=1}^{n}\phi_\ell(\boldsymbol{x}_i)\phi_{\ell'}(\boldsymbol{x}_i) \quad \text{and} \quad \widehat{h}_\ell = \frac{1}{n'}\sum_{j=1}^{n'}\phi_\ell(\boldsymbol{x}'_j).$$

The solution can then be analytically calculated as

$$\boldsymbol{\theta} = \left(\widehat{\boldsymbol{H}} + \lambda\boldsymbol{I}\right)^{-1}\widehat{\boldsymbol{h}},$$

where $\boldsymbol{I}$ is the $b \times b$ identity matrix. All hyper-parameters in the model can be objectively set via cross-validation.

# 3 Online Density-Ratio Estimation

In this section, we consider an online learning setup where samples $\boldsymbol{x}_t$ and $\boldsymbol{x}'_t$ following $p(\boldsymbol{x})$ and $p'(\boldsymbol{x})$ are given sequentially at time step $t$. We first propose an online KL-based density ratio estimator and then an LS-based online density-ratio estimator.

## 3.1 Online KL Density Ratio Estimation

Given the current parameter $\boldsymbol{\theta}_t$ that has been estimated using $\{\boldsymbol{x}_i\}_{i=1}^{t-1} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x})$ and $\{\boldsymbol{x}'_j\}_{j=1}^{t-1} \overset{\text{i.i.d.}}{\sim} p'(\boldsymbol{x})$, the basic idea for the online method is to update the parameter to minimize the error for the next samples $\boldsymbol{x}_t$ and $\boldsymbol{x}'_t$:

$$\boldsymbol{\theta}^\top\boldsymbol{\phi}(\boldsymbol{x}_t) - \log\boldsymbol{\theta}^\top\boldsymbol{\phi}(\boldsymbol{x}'_t).$$

We employ the idea of *adaptive regularization of weight vectors* (AROW) (Crammer et al., 2009): for parameter vector $\boldsymbol{\theta}$, the normal distribution $N(\boldsymbol{\theta}, \boldsymbol{\Sigma})$ with the mean vector $\boldsymbol{\theta}$ and the covariance matrix $\boldsymbol{\Sigma}$ is considered, and the update of parameters is penalized by the KL divergence. More specifically, the AROW KL-based training criterion to be minimized is

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\Sigma}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}_t) - \log \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}'_t) \\ + \frac{\gamma}{2}\left( \mathrm{tr}\left(\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\Sigma}\right) + (\boldsymbol{\theta}_t - \boldsymbol{\theta})^\top \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{\theta}_t - \boldsymbol{\theta}) - b - \log \frac{\det(\boldsymbol{\Sigma})}{\det(\boldsymbol{\Sigma}_t)} \right) \\ + \frac{1}{2}\Big( \phi(\boldsymbol{x}_t)^\top \boldsymbol{\Sigma} \phi(\boldsymbol{x}_t) + \phi(\boldsymbol{x}'_t)^\top \boldsymbol{\Sigma} \phi(\boldsymbol{x}'_t) \Big), \end{aligned} \tag{2}$$

where $\gamma > 0$ is the passiveness parameter. The first two terms in the right-hand side of Eq.(2) correspond to the KL error for the next samples $\boldsymbol{x}_t$ and $\boldsymbol{x}'_t$, the third term is the KL penalty for parameter updates, and the fourth term is the regularizer for covariance matrix $\boldsymbol{\Sigma}$.

The optimality condition w.r.t. the mean $\boldsymbol{\theta}$ is

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = \boldsymbol{\phi}(\boldsymbol{x}_t) - \frac{\boldsymbol{\phi}(\boldsymbol{x}'_t)}{\boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}'_t)} - \gamma \boldsymbol{\Sigma}_t^{-1}(\boldsymbol{\theta}_t - \boldsymbol{\theta}) = \mathbf{0}.$$

Note that, since the domain of $\log(t)$ is $t > 0$, in the above we should take into account the constraint $\boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}'_t) > 0$. We can simplify the above by making the substitution

$$\eta := \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}'_t),$$

and multiplying the entire equation with $\eta$:

$$\eta \boldsymbol{\phi}(\boldsymbol{x}_t) - \boldsymbol{\phi}(\boldsymbol{x}'_t) - \gamma\eta \boldsymbol{\Sigma}_t^{-1}\boldsymbol{\theta}_t + \gamma\eta \boldsymbol{\Sigma}_t^{-1}\boldsymbol{\theta} = \mathbf{0}. \tag{3}$$

Multiplying this with $\boldsymbol{\Sigma}_t$ from the left of Eq.(3), we have

$$\eta \boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}_t) - \boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}'_t) - \gamma\eta \boldsymbol{\theta}_t + \gamma\eta \boldsymbol{\theta} = \mathbf{0}. \tag{4}$$

Again multiplying with $\boldsymbol{\phi}(\boldsymbol{x}'_t)^\top$ from the left gives

$$\eta \boldsymbol{\phi}(\boldsymbol{x}'_t)^\top \boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}_t) - \boldsymbol{\phi}(\boldsymbol{x}'_t)^\top \boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}'_t) - \eta\gamma \boldsymbol{\phi}(\boldsymbol{x}'_t)^\top \boldsymbol{\theta}_t + \gamma\eta^2 = 0,$$

and collecting the terms gives:

$$\gamma\eta^2 + \eta \left[ \boldsymbol{\phi}(\boldsymbol{x}'_t)^\top \boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}_t) - \gamma \boldsymbol{\phi}(\boldsymbol{x}'_t)^\top \boldsymbol{\theta}_t \right] - \boldsymbol{\phi}(\boldsymbol{x}'_t)^\top \boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}'_t) = 0.$$

By defining

$$\begin{aligned} \beta &= \boldsymbol{\phi}(\boldsymbol{x}'_t)^\top \boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}_t) - \gamma \boldsymbol{\phi}(\boldsymbol{x}'_t)^\top \boldsymbol{\theta}_t, \\ c &= -\boldsymbol{\phi}(\boldsymbol{x}'_t)^\top \boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}'_t), \end{aligned}$$

we can solve it for $\eta$ as

$$\eta = \frac{-\beta \pm \sqrt{\beta^2 - 4\gamma c}}{2\gamma}.$$

Note that this quadratic has two solutions. However, since $\eta = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x}'_t) > 0$ because the domain of the logarithm is positive, the solution $\eta_+$ is given by

$$\eta_+ = \frac{-\beta + \sqrt{\beta^2 - 4\gamma c}}{2\gamma}.$$

This is then substituted in Eq.(4) to obtain the update rule:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{1}{\gamma}\boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}_t) + \frac{1}{\gamma\eta_+}\boldsymbol{\Sigma}_t \boldsymbol{\phi}(\boldsymbol{x}'_t). \tag{5}$$

Next, we equate the derivative of $L(\boldsymbol{\theta}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\Sigma}$ to zero:

$$\frac{\partial L}{\partial \boldsymbol{\Sigma}} = \frac{1}{2}\Big(\gamma\boldsymbol{\Sigma}_t^{-1} - \gamma\boldsymbol{\Sigma}^{-1} + \boldsymbol{\phi}(\boldsymbol{x}_t)\boldsymbol{\phi}(\boldsymbol{x}_t)^\top + \boldsymbol{\phi}(\boldsymbol{x}'_t)\boldsymbol{\phi}(\boldsymbol{x}'_t)^\top\Big) = \boldsymbol{O}.$$

Using the *Sherman-Morrison formula*[1] gives

$$\begin{aligned}\boldsymbol{\Sigma}_{t+1} &= \left(\boldsymbol{\Sigma}_t^{-1} + \frac{\boldsymbol{\phi}(\boldsymbol{x}_t)\boldsymbol{\phi}(\boldsymbol{x}_t)^\top}{\gamma} + \frac{\boldsymbol{\phi}(\boldsymbol{x}'_t)\boldsymbol{\phi}(\boldsymbol{x}'_t)^\top}{\gamma}\right)^{-1} \\ &= \boldsymbol{\Sigma}'_t - \frac{\boldsymbol{\Sigma}'_t\boldsymbol{\phi}(\boldsymbol{x}'_t)\boldsymbol{\phi}(\boldsymbol{x}'_t)^\top\boldsymbol{\Sigma}'_t}{\gamma + \boldsymbol{\phi}(\boldsymbol{x}'_t)^\top\boldsymbol{\Sigma}'_t\boldsymbol{\phi}(\boldsymbol{x}'_t)},\end{aligned} \tag{6}$$

where we put

$$\boldsymbol{\Sigma}'_t := \boldsymbol{\Sigma}_t - \frac{\boldsymbol{\Sigma}_t\boldsymbol{\phi}(\boldsymbol{x}_t)\boldsymbol{\phi}(\boldsymbol{x}_t)^\top\boldsymbol{\Sigma}_t}{\gamma + \boldsymbol{\phi}(\boldsymbol{x}_t)^\top\boldsymbol{\Sigma}_t\boldsymbol{\phi}(\boldsymbol{x}_t)}.$$

## 3.2 Online LS Density Ratio Estimation

Given the current parameter $\boldsymbol{\theta}_t$ that has been estimated using $\{\boldsymbol{x}_i\}_{i=1}^{t-1} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x})$ and $\{\boldsymbol{x}'_j\}_{j=1}^{t-1} \overset{\text{i.i.d.}}{\sim} p'(\boldsymbol{x})$, the basic idea for the online method is to update the parameter to minimize the error for the next samples $\boldsymbol{x}_t$ and $\boldsymbol{x}'_t$:

$$\frac{1}{2}r_{\boldsymbol{\theta}}(\boldsymbol{x}_t)^2 - r_{\boldsymbol{\theta}}(\boldsymbol{x}'_t).$$

[1] For matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$, it holds that

$$(\boldsymbol{A} + \boldsymbol{b}\boldsymbol{b}^\top)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{b}\boldsymbol{b}^\top\boldsymbol{A}^{-1}}{1 + \boldsymbol{b}^\top\boldsymbol{A}^{-1}\boldsymbol{b}}.$$

Then, the AROW-type update rule, derived just as in Section 3.1, is

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \left(\boldsymbol{\phi}(\boldsymbol{x}_t)\boldsymbol{\phi}(\boldsymbol{x}_t)^\top + \gamma\boldsymbol{\Sigma}_t^{-1}\right)^{-1}(\boldsymbol{\phi}(\boldsymbol{x}'_t) + \gamma\boldsymbol{\Sigma}_t^{-1}\boldsymbol{\theta}_t)\\ &= \boldsymbol{\theta}'_t - \frac{\boldsymbol{\phi}(\boldsymbol{x}_t)^\top\boldsymbol{\theta}'_t}{\boldsymbol{\phi}(\boldsymbol{x}_t)^\top\boldsymbol{\Sigma}_t\boldsymbol{\phi}(\boldsymbol{x}_t)+\gamma}\boldsymbol{\Sigma}_t\boldsymbol{\phi}(\boldsymbol{x}_t),\end{aligned}$$

where the second line follows from the application of the *Sherman-Morrison formula* and $\boldsymbol{\theta}'_t$ is defined as

$$\boldsymbol{\theta}'_t := \boldsymbol{\theta}_t + \frac{\boldsymbol{\Sigma}_t\boldsymbol{\phi}(\boldsymbol{x}'_t)}{\gamma}.$$

The update rule for $\boldsymbol{\Sigma}_{t+1}$ is exactly the same as that for the KL method (see (6)).

Note that the above online LS density ratio estimator can be regarded as an application of classical *recursive least-squares* (Haykin, 2002) to density ratio estimation.

# 4 Experiments

In this section, we experimentally investigate the performance of the proposed online density-ratio estimators on the problem of online *inlier-based outlier detection* (Hido et al., 2011).

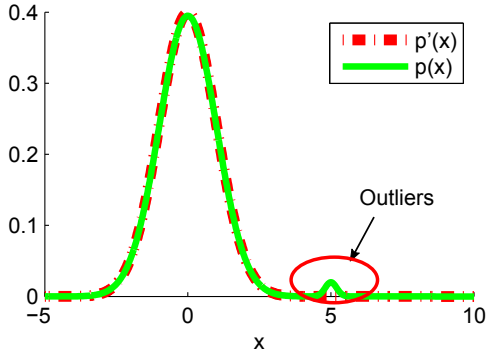
## 4.1 Setup

Following Hido et al. (2011), we formulate the inlier-based outlier detection problem as the problem of estimating the density ratio $r(\boldsymbol{x}) = p'(\boldsymbol{x})/p(\boldsymbol{x})$, where $p'(\boldsymbol{x})$ is the density of inliers and $p(\boldsymbol{x})$ is the density of unlabeled samples (i.e., a mixture of inliers and outliers). Samples for which the density ratio $r(\boldsymbol{x})$ is low, tend to be outliers (see Fig. 1). This is due to the fact that the probability that a sample is an inlier is proportional to the density ratio. To see this, assume that inliers are labeled as $y = 1$ an outliers are labeled as $y = -1$. Then the probability that sample $\boldsymbol{x}$ is an inlier is

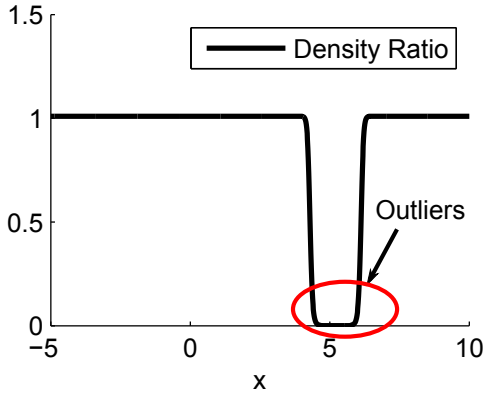
$$p(y=1|\boldsymbol{x}) \propto \frac{p(\boldsymbol{x}|y=1)}{p(y=1)p(\boldsymbol{x}|y=1)+p(y=-1)p(\boldsymbol{x}|y=-1)} = \frac{p'(\boldsymbol{x})}{p(\boldsymbol{x})}.$$

To identify outliers, we are therefore interested in areas where the density ratio is low. Note that traditional outlier detection methods assume that outliers occur in areas where the inlier density is low. Inlier-based outlier detection is however not constrained by such an assumption and can identify outliers that occur in high-density areas (cf. Fig. 1 and Fig. 2).

In the experiments below, we first prepare training and test sets which both contain inliers and outliers. Then we form the inlier set $\{\boldsymbol{x}'_j\}_{j=1}^{n'}$ from the inliers contained in the training set and form the unlabeled set $\{\boldsymbol{x}_i\}_{i=1}^{n}$ from both the inliers and outliers in the training set.

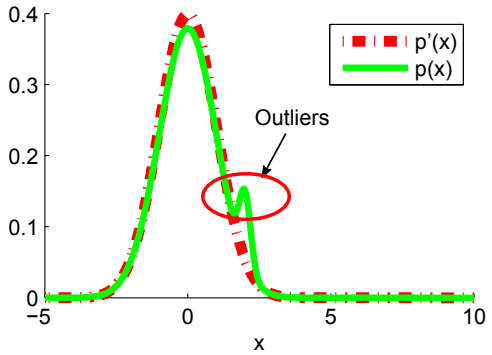



(a) The density of inliers $p'(\boldsymbol{x})$ and unlabeled samples $p(\boldsymbol{x})$.

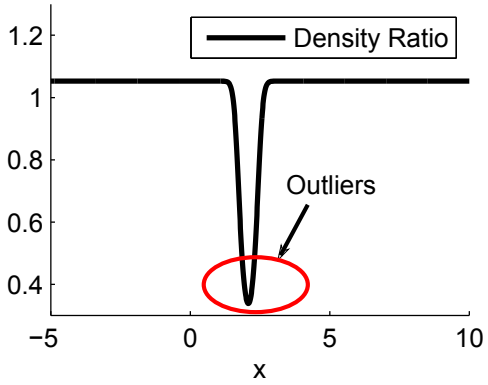



(b) The density ratio $r(\boldsymbol{x}) = \frac{p'(\boldsymbol{x})}{p(\boldsymbol{x})}$.

Figure 1: An example of density ratio based outlier detection. Densities for a dataset consisting of inliers $p'(\boldsymbol{x})$ and a corrupted dataset consisting of both inliers and outliers $p(\boldsymbol{x})$ are given in Fig 1(a). We see in Fig. 1(b) that the density ratio $r(\boldsymbol{x}) = p'(\boldsymbol{x})/p(\boldsymbol{x})$ takes a small value in the region where $p(\boldsymbol{x})$ significantly differs from $p'(\boldsymbol{x})$. It is in this region where outliers occur.




(a) The density of inliers $p'(\boldsymbol{x})$ and unlabeled samples $p(\boldsymbol{x})$.




(b) The density ratio $r(\boldsymbol{x}) = \frac{p'(\boldsymbol{x})}{p(\boldsymbol{x})}$.

Figure 2: An example of density ratio-based outlier detection when the inlier and outlier distributions overlap. We see that even when outliers occur in high-density areas (Fig. 2(a)), they can be identified via the density ratio (Fig. 2(b)).

We initially give to outlier detectors 150 inlier samples and 150 unlabeled samples randomly chosen from the inlier and unlabeled sets, respectively. Then, a pair of randomly chosen inlier and unlabeled samples is given to outlier detectors in an online manner over iterations. In practice, the class prior of the mixture dataset is unknown. Therefore, the score calculated by $r(\boldsymbol{x})$ can not be appropriately thresholded[2]. Therefore, instead of the classification accuracy, the performance of outlier detectors is evaluated by the *area under the receiver operating characteristic curve* (AUC) for the test set. The AUC is used since it is independent of the particular class prior of the unlabeled dataset or threshold.

For density ratio estimation, we use 150 Gaussian kernels as basis functions $\{\phi_\ell(\boldsymbol{x})\}_{\ell=1}^{150}$:

$$\phi_\ell(\boldsymbol{x}) := \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_\ell\|^2}{2\sigma^2}\right),$$

where $\|\cdot\|$ denotes the Euclidean norm, $\sigma$ is the Gaussian bandwidth, and $\{\boldsymbol{c}_\ell\}_{\ell=1}^{150}$ are the Gaussian centers chosen from the initial numerator (inlier) samples. The density ratio was estimated using five methods:

- The batch KL method (Batch-KL) described in Section 2
- The online KL method (AROW-KL) proposed in Section 3.
- The online KL method with naive stochastic gradient descent (SGD-KL).
- The batch LS method (Batch-LS) described in Section 2.
- The online LS method (AROW-LS), described in Section 3.
- The online LS method with naive stochastic gradient descent (SGD-LS).

Density ratio estimators contain unknown hyper-parameters. Cross-validation is the most standard way to select these hyper-parameter values. However, previous samples are not stored in the online methods, and therefore cross-validation cannot be directly employed. Here, we maintain all models with different hyper-parameter values throughout the online learning process, and use newer samples for validation to choose the best model at each time step. More specifically, at time $t+400$, estimation is carried out using samples only up to time $t$, and the latest 400 samples are used for computing the validation error with respect to the objective function. For fair comparison, we also use the same hyper-parameter selection scheme for the batch method.s

The Gaussian kernel model for all density ratio estimators contains a hyper-parameter for the kernel width, which was selected from the following candidates:

$$\sigma \in \{0.3, 0.3+\frac{4-0.3}{10}, \ldots, 4\} \times \mathrm{median}(\{\|\boldsymbol{c}_\ell - \boldsymbol{c}_{\ell'}\|\}_{\ell,\ell'=1}^{150}).$$

[2]In practice, a common strategy is to rank the unlabeled samples according to the score then remove a percentage of the samples with the lowest score. This percentage is specified by the practitioner based on domain knowledge.

In addition to the kernel width, each method contains one additional hyper-parameter. The regularization parameter $\lambda$ for the batch density ratio estimators are selected from

$$\lambda \in \{10^{-3}, 10^{-\frac{5}{2}}, \ldots, 10^{1}\},$$

the passiveness parameter $\gamma$ for the AROW based density ratio estimators are selected from

$$\gamma \in \{10^{-1}, 10^{-\frac{1}{2}}, \ldots, 10^{\frac{5}{2}}, 10^{3}\}.$$

For the stochastic gradient descent methods, the step size $\eta$ was treated as a hyper-parameter and selected from

$$\eta \in \left\{10^{-6}, 10^{-5.5}, \ldots, 10^{-2}\right\}.$$

## 4.2 Spambase Dataset

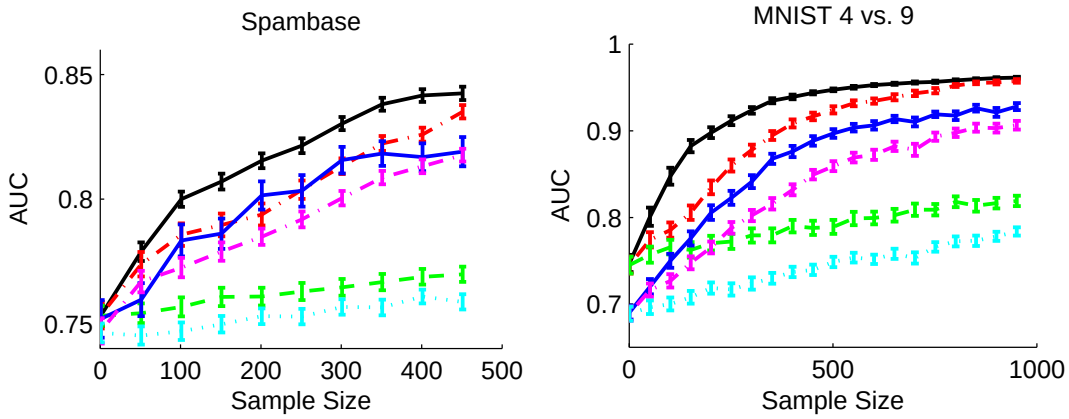
First, we perform experiments using the *Spambase* dataset[3], which contains 4061 e-mail samples with 57 attributes. There are 1813 spam samples in the dataset. Among 57 attributes, we only use 48 attributes (the percentage of words in an e-mail) for outlier detection. 25% of the dataset is used for evaluation and the remaining 75% is training data. It is assumed that all the positive samples are inliers. The probability that an outlier occurs is set to 20%.

Fig. 3(a) (left) depicts the AUC values as a function of the sample size. This figure shows that the batch methods are generally more accurate than their online counterparts. This is expected since the batch methods have access to all samples. The proposed online methods are not much worse than their batch counterparts and much better than naive stochastic gradient descent based online methods.

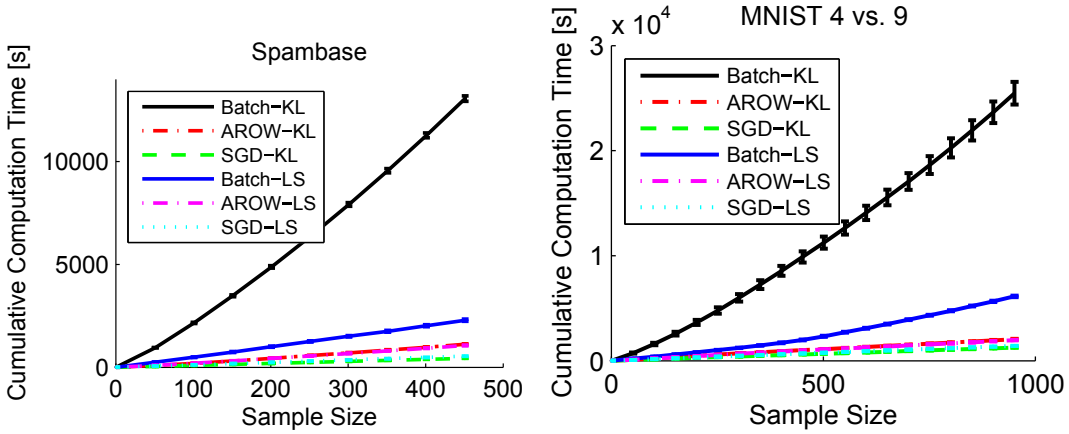
Another interesting thing to note is that the batch KL method is generally better than the batch LS method. Fig. 1 shows that outliers occur in areas where the density ratio is small. In Fig. 4 the KL divergence and squared loss is plotted. From these plots, we can confirm that the KL divergence penalizes error more severely when the density ratio is small. This may contribute to the greater accuracy in identifying outliers.

Fig. 3(b) (left) depicts the cumulative computation time as a function of the sample size, showing that the computation time of the online methods is significantly lower than the batch methods. Stochastic gradient descent based online methods are faster than the proposed AROW based online methods, perhaps since the proposed methods contain $b \times b$ matrix $\Sigma$ that is updated at each time step. This may be mitigated by approximating $\Sigma$ by a diagonal matrix, as in the original AROW paper (Crammer et al., 2009). Due to the fact that the LS method has an analytic solution, the batch LS method is much faster than the batch KL method. This aspect of the LS method is a major motivation why it is often preferred over the KL method. However, we see that in the online setup, both the KL and LS methods are about the same speed.

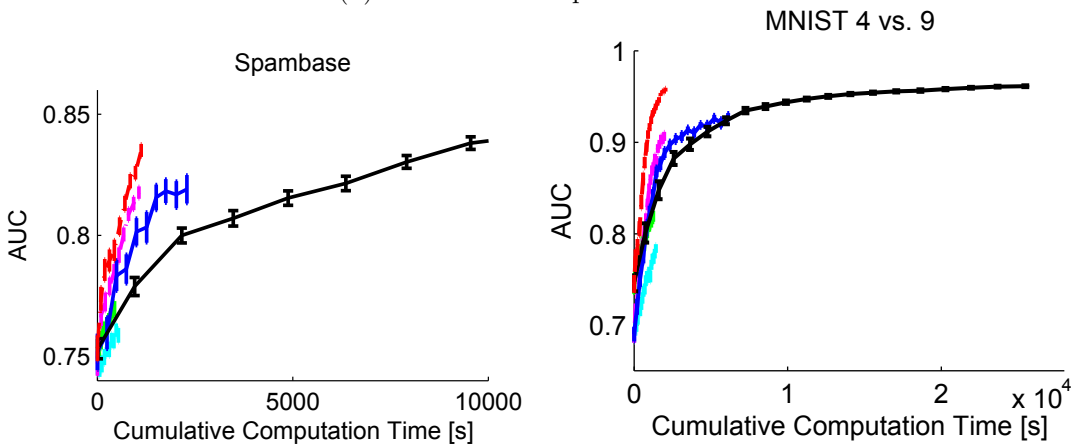
[3]The dataset was obtained from `http://archive.ics.uci.edu/ml/`.




(a) AUC vs. sample size

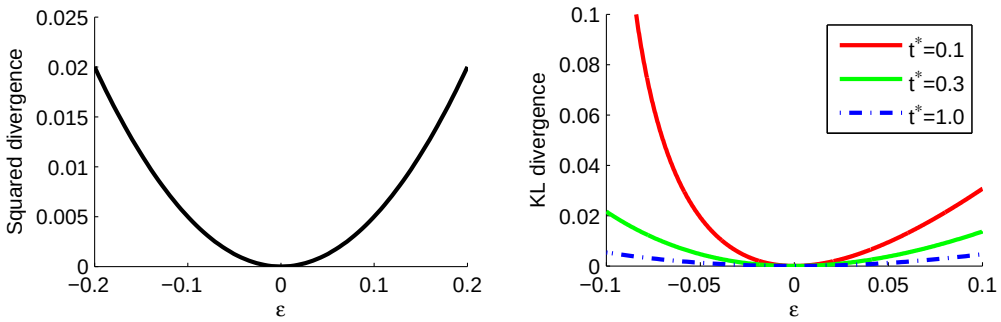



(b) Cumulative computation time




(c) AUC vs. computation time

Figure 3: Experimental results for the Spambase (left column) and MNIST (right column) datasets. The standard error is given as error bars. "Batch-KL" and "Batch-LS" use all the samples in a batch setup, "AROW-KL" and "AROW-LS" are the proposed AROW-based online methods. "SGD-KL" and "SGD-LS" are the naive stochastic gradient descent based online methods.




(a) LS $(t^* \| t^* + \epsilon)$ (b) KL $(t^* \| t^* + \epsilon)$

Figure 4: The squared loss LS $(t^* \| t^* + \epsilon)$ and the KL loss KL $(t^* \| t^* + \epsilon)$ for different values of $t^*$. The KL loss penalizes errors when $t^*$ is small much more strongly.

In Fig. 3(c) (left), we plot the cumulative computation time against the AUC values. From this, we see that, with a limited computational budget, the proposed online methods significantly outperform both their batch and the stochastic gradient descent counterparts.

## 4.3 MNIST Dataset

Next, we use the *MNIST* dataset[4], which contains images of hand-written digits of size $28 \times 28$ pixels. Each image is represented by a 784-dimensional feature vector which is of much higher dimensional than the previous Spambase dataset. Each pixel in the images are normalized to $[0, 1]$ representing its gray-scale intensity level.

For the first experiment, we use the images of "4"or "9" in the datasets, and regard "4" as inliers and "9" as outliers. 25% of the dataset is used for evaluation and the remaining 75% is training data. The probability that an outlier is drawn in the unlabeled dataset was set to 20%.

The experimental results are given in the right-hand column of Fig. 3. From the graphs, we see that the results show similar tendencies to the previous results. That is, the proposed online methods are significantly faster than their batch versions. Furthermore, due to the better estimate of the density ratio, the outlier detection accuracy is much higher than the online stochastic gradient methods. With a limited computational budget, the proposed methods also give higher classification accuracy than all other methods.

For the second experiment, we choose one digit as an inlier class, and regard all other digits as outliers. The initial inlier datarset contained 150 samples and the unlabeled dataset 150 samples. The probabilty that an outlier occurs in the unlabeled dataset was set to 20%. This experiment was performed by selecting "1"–"9" in turn as inliers (i.e., 9 separate experiments). The AUC versus cumulative computation time is plotted in Fig. 5. From the graphs, we again see that for a limited computational budget, the proposed online KL divergence gives more accurate results.

[4]The dataset was obtained from `http://yann.lecun.com/exdb/mnist/`.

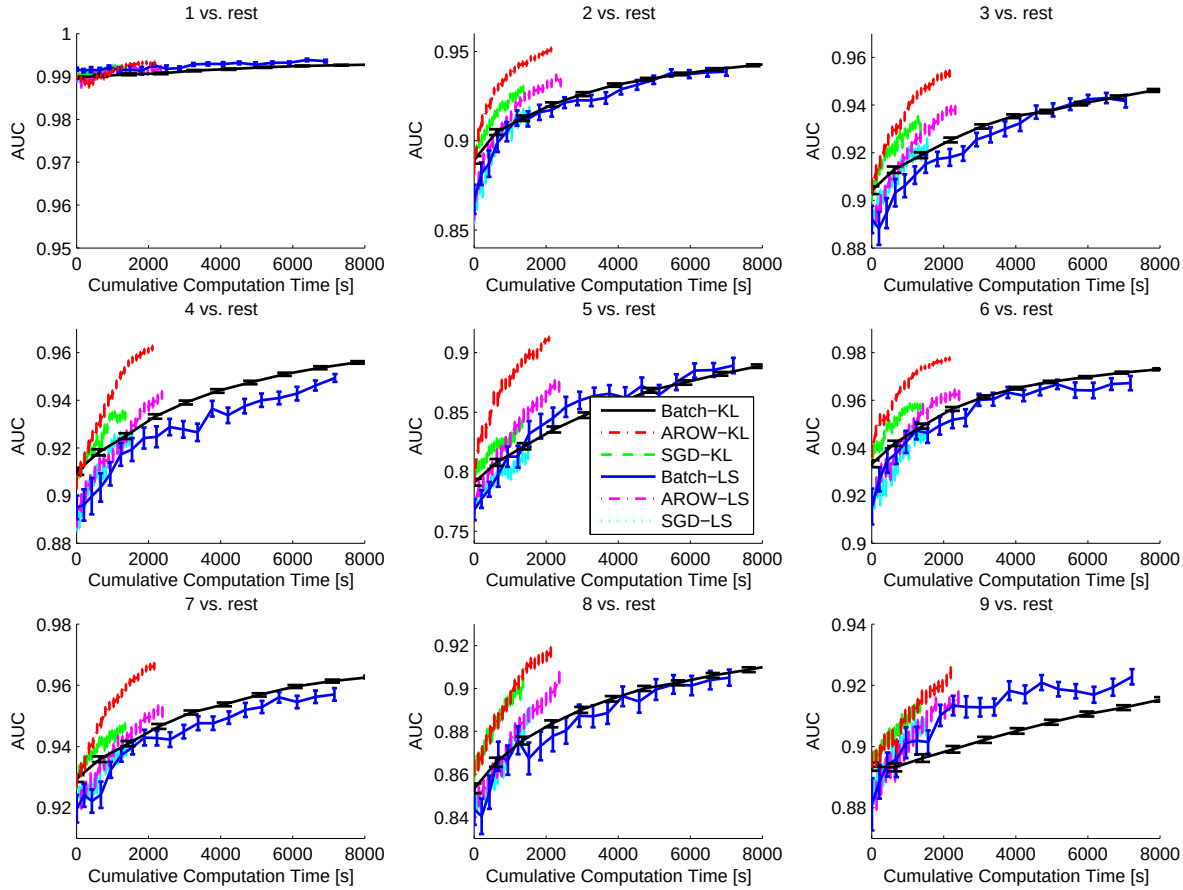



Figure 5: Computational time vs. accuracy when a single digit is considered an inlier and the rest are considered outliers.

# 5 Conclusion

Various machine learning problems can be solved via density ratio estimation, which can be performed by matching the density ratio to a model under a Bregman divergence. Two popular approaches is to use the Kullback-Leibler loss and the squared loss. In this paper, we extended the original batch density ratio estimators to an online learning scenario based on the idea of *adaptive regularization of weight vectors*, which has been successfully used in regression and classification (Crammer et al., 2009). Through experiments on inlier-based outlier detection (Hido et al., 2011), we demonstrated the usefulness of the proposed methods. We showed that, for a given computational budget, online AROW based methods outperform both online stochastic gradient descent and batch methods. We also showed that the KL divergence based loss may be more suited to the outlier detection problem.

## Acknowledgment

The authors thank Mr. Tomoya Sakai for his valuable comments. MCdP was supported by the JST CREST project, and MS was supported by KAKENHI 25700022.

## References

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 414–422, 2009.

S. Haykin. *Adaptive Filter Theory*. Prentice Hall, Upper Saddle River, NJ, USA, 2002.

S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Statistical outlier detection using direct density ratio estimation. *Knowledge and Information Systems*, 26(2):309–336, 2011.

R. Izbicki, A.B. Lee, and C.M. Schafer. High-Dimensional Density Ratio Estimation with Extensions to Approximate Likelihood Computation. In G. Lebanon and S. V. N. Vishwanathan (Eds.), *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS2014)*, JMLR Workshop and Conference Proceedings, vol. 33, pages 420–429, 2014.

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

X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

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

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

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

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