Training Data Soft Selection via Joint Density Ratio Estimation

Ryuta Matsuno RYUTA-MATSUNO@NEC.COM
NEC Corporation.

Editors: Hung-yi Lee and Tongliang Liu

Abstract

This paper studies the training data selection problem, focusing on the selection of effective samples to improve model training using data affected by distributional shifts (i.e., data drifts). Existing drift-detection-based methods struggle with local drifts, while recent drift-localization-based methods lack theoretical support for the problem and are often ineffective. To tackle these issues, this paper proposes TSJD, a training data soft selection method based on joint density ratio estimation. TSJD assigns training weights (i.e., soft selects) to samples based on the estimated joint density ratio to align the selected data with the recent data distribution. By evaluating each sample independently of time, TSJD effectively addresses local data drifts. We also provide theoretical guarantees by deriving an upper bound on the generalization error for models trained with data selected by TSJD. In numerical experiments with four real-world datasets, TSJD shows great versatility, achieving the best or comparable results over baseline methods in all of the experiments.

Keywords: Training data selection; data drift; joint density ratio estimation;

1. Introduction

Supervised learning aims at training a prediction model to minimize test error, assuming that the data distribution is consistent between the training and test data. However, real-world applications often violate this assumption and the data distribution changes over time, known as data drift. These drifts make it ineffective to directly use the given training data (Awasthi et al., 2024; Shimodaira, 2000; Quionero-Candela et al., 2009). As a result, a problem of training data selection arises, i.e., selecting effective samples from drifting data to improve the prediction performance of models trained with (Hinder et al., 2022; Liu et al., 2017).

A naive approach to the problem just uses recent samples, assuming that the data distribution is approximately consistent for a short time span (Wang et al., 2003; Woźniak, 2013; Brzezinski and Stefanowski, 2014). However, this approach discards all the older samples, which is potentially effective for model training. Drift detection methods (Bifet and Gavaldà, 2007; Page, 1954) improve this approach by determining when to separate the recent and old samples, performing concept drift detection from the present to the past, and utilizing the samples up to the time when a drift is detected. However, they still select samples only based on time, failing to adapt to local concept drifts that occurred in a small part of the input space, as well as recurring concept drifts (Hinder et al., 2022).

Recent studies propose drift localization methods (Hinder et al., 2022; Liu et al., 2017), which detect local data drift in the sample space based on recent data, and we can select samples based on the input and output of each sample, aligned with the recent data distribution for model training. Although these approaches are capable of flexibly selecting samples from old samples, the theoretical properties of models trained with the selected samples remain unknown, limiting their validity to the problem. Indeed, the empirical performance of these methods is inferior to naive baselines, as shown in our experiments.

To address this, we propose TSJD, a Training data soft Selection method based on Joint Density ratio estimation. TSJD first trains a joint density ratio estimator between recent and old data distributions. It then assigns training weights to each sample (i.e., soft-selects) based on the estimated ratio, effectively addressing local data drifts by utilizing both inputs and outputs. In addition, we provide a theoretical upper bound on the generalization error of models trained with our method, ensuring the validity of our method for the training data selection problem. Experiments on four real-world datasets show TSJD consistently achieves the best or comparable results across all 30 settings, demonstrating its effectiveness and versatility.

Our contributions are summarized as follows;

- We propose TSJD, a training data soft selection method using a joint density ratio estimator to effectively handle local data drifts (Section 3).
- We offer theoretical analysis and establish a generalization error upper bound to support the validity of TSJD (Section 4).
- We conduct extensive numerical experiments and provide empirical evidences which highlight the superiority of TSJD over various baseline methods (Section 5).

Due to the space limitation, all proofs of theorems and lemmas are presented in the supplementary material. We also report comprehensive experiments on seven real-world datasets across 126 settings in our supplementary material.

2. Preliminary

In this section, we explain the problem formulation as well as related works briefly.

2.1. Problem Formulation

We consider a supervised classification problem. The input space is $\mathcal{X} \subseteq \mathbb{R}^d$ and the output space is $\mathcal{Y} = [K]$, where $[K]$ denotes the set of integers from 1 to K, i.e., $[K] := \{1, ..., K\}$ and the integer $K \in \mathbb{Z}_{\geq 2}$ is the number of classes. Let $p_t(\mathbf{x}, y)$ be a joint distribution over $\mathcal{X} \times \mathcal{Y}$ at time $t \in \mathbb{Z}_{\geq 1}$. A sample $(\mathbf{x}_t, y_t)$ is sampled from $p_t(\mathbf{x}, y)$ at every time step $t \in [T]$, where $T \in \mathbb{Z}_{\geq 1}$ is the current time. All samples available at the training phase form a datasets $D := \{(\mathbf{x}_t, y_t)\}_{t=1}^{T}$. A standard approach for classification tasks is to train a model by minimizing the cross-entropy loss, i.e.,

$$\ell_{\mathrm{CE}}(h(\mathbf{x}), y) := -\log(h(y|\mathbf{x})), \tag{1}$$

where $h : \mathcal{X} \to \Delta^{K-1}$ is a probabilistic classification model, $\Delta^{K-1} := \{\mathbf{p} \in \mathbb{R}^{K}_{\geq 0} \,|\, \|\mathbf{p}\|_1 = 1\} \subset \mathbb{R}^K$ is the $(K-1)$ dimensional probability simplex, and $h(y|\mathbf{x}) = (h(\mathbf{x}))_y$ computes the probability that an input $\mathbf{x} \in \mathcal{X}$ belongs to a class $y \in \mathcal{Y}$. The model h predicts a class of $\mathbf{x} \in \mathcal{X}$ by $\arg\max_{y \in \mathcal{Y}} h(y|\mathbf{x})$.

Let $\mathcal{H}$ be the hypothesis space of a classification model h. We aim to find $h^* \in \mathcal{H}$ that maximizes accuracy over the next M steps, i.e., $t = T+1, \ldots, T+M$. To achieve this, we seek to minimize the expected zero-one loss over the distributions $p_{T+1}(\mathbf{x}, y), \ldots, p_{T+M}(\mathbf{x}, y)$ (a.k.a. the zero-one risk), denoted by R_{01};

$$R_{01}(h) := \frac{1}{M} \sum_{t=T+1}^{T+M} \mathop{\mathbb{E}}_{p_t(\mathbf{x},y)} [\ell_{01}(h(\mathbf{x}), y)], \tag{2}$$

where $\ell_{01}(h(\mathbf{x}), y) = \mathbb{I}[\arg\max_k h(k|\mathbf{x}) \neq y]$ and $\mathbb{I}[P]$ is the Iverson bracket, which is 1 if the proposition P is true and 0 otherwise. The notation $\mathbb{E}_{p(\mathbf{x},y)}[f(\mathbf{x}, y)] := \int_{\mathcal{X}\times\mathcal{Y}} f(\mathbf{x}, y)p(\mathbf{x}, y)d\mathbf{x}dy$ is the expectation of a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ over the joint distribution $p(\mathbf{x}, y)$.

Obtaining h^* is challenging because the future distribution p_t for $t > T$ is unknown. Additionally, if there are concept drifts (changes in the conditional distribution of y given $\mathbf{x}$, $p(y|x)$, a.k.a. conditional shift) with in D, the naive use of all of D (a.k.a. *ERM: empirical risk minimization*) is unsuitable for finding h^*. A common practical approach is to use the most recent N samples from D, i.e., $\{(\mathbf{x}_t, y_t)\}_{t=T-N+1}^{T}$, as training data (Wang et al., 2003; Woźniak, 2013; Brzezinski and Stefanowski, 2014). Here, $N \in [T]$ is determined based on domain knowledge or set as a hyperparameter. This approach is based on an implicit assumption as follows.

Assumption 1 (Temporal consistency of the joint distribution) *There exists an integer $N \in [T]$ and small constants $\tau_X, \tau_{Y|X} \geq 0$ such that for any $t \in \{T-N+1, ..., T\}$ and $t' \in \{T+1, ..., T+M\}$ each of the followings holds;*

$$d_X(p_t, p_{t'}) \leq \tau_X \tag{3}$$

$$\forall \mathbf{x} \in \mathcal{X}, \quad d_{Y|X}(p_t(\cdot|\mathbf{x}), p_{t'}(\cdot|\mathbf{x})) \leq \tau_{Y|X}, \tag{4}$$

where d_X and $d_{Y|X}$ compute the distances of two marginal ($p_t(\mathbf{x})$ and $p_{t'}(\mathbf{x})$) and conditional ($p_t(y|\mathbf{x})$ and $p_{t'}(y|\mathbf{x})$) distributions, respectively.

We define $p_t(\cdot|\mathbf{x})$ as $p_t(\cdot|\mathbf{x}) := [p_t(y = 1|\mathbf{x}), \ldots, p_t(y = K|\mathbf{x})]^\top \in \Delta^{K-1}$. Specifically, in our analysis in Section 4, we use the Wasserstein 1-distance (Edwards, 2011) (a.k.a. earth mover's distance) W_1 for d_X and the L^2-norm for $d_{Y|X}$. Various choices of τ_X and $\tau_{Y|X}$ have been explored in examples such as the following;

Example 1 *The traditional ERM (Hastie et al., 2001) assumes $\tau_X = \tau_{Y|X} = 0$.*

Example 2 *Covariate shift (Shimodaira, 2000) assumes $\tau_X > 0$ and $\tau_{Y|X} = 0$.*

The naive approach under Assumption 1 selects the recent samples $\{(\mathbf{x}_t, y_t)\}_{t=T-N+1}^{T}$ and discards older samples from $t = 1$ to $t = T - N$. Although the older samples might worsen the training of h due to data drifts, selecting effective ones can enhance its performance. To tackle this, we employ a soft selection method by assigning positive weights to each sample in D. In summary, this paper formulates the problem as follows.

Definition 1 (Training Data Soft Selection Problem) *Given a dataset* $D = \{(\mathbf{x}_t, y_t)\}_{t=1}^{T}$*, where* $(\mathbf{x}_t, y_t)$ *is independently sampled from* p_t *and assume Assumption 1 holds, the task is to find sample weights* $W^* = \{w_t\}_{t=1}^{T} \in \mathbb{R}_{\geq 0}^{T}$ *which minimizes the zero-one risk of the trained model with, i.e.,*

$$W^* = \mathop{\arg\min}_{W \in \mathbb{R}_{\geq 0}^{T}} R_{01}\left(\mathop{\arg\min}_{h \in \mathcal{H}} \sum_{t=1}^{T} w_t \ell_{\mathrm{CE}}(h(\mathbf{x}_t), y_t)\right), \tag{5}$$

where $\mathcal{H}$ *is an arbitrary hypothesis space.*

Remark 2 *In our problem formulation,* $\mathcal{H}$ *is given after choosing* W*. If* $\mathcal{H}$ *were given before deciding* W*, we could optimize both* W *and* h *simultaneously, likely improving* R_{01} *risk (Zhang et al., 2020; Bassily et al., 2024; Mohri and Muñoz Medina, 2012). In practice, however, AutoML tools, such as auto-sklearn*[1] *and PyCaret,*[2] *often handle the training of* h*, limiting customization of the training. In addition,* $\mathcal{H}$ *is often composed of different models with different behaviors, including decision trees, linear models, gradient boosting models, and neural networks, making the optimization of* W *along with* h *unstable. Furthermore, in MLOps frameworks (Kreuzberger et al., 2023; Ruf et al., 2021), data preparation and model training are separate steps. These conditions make joint optimization impractical, while our formulation remains usable.*

2.2. Related Works

We review drift detection methods, drift localization methods, and density ratio estimation methods as related works as follows;

Drift Detection Methods. Drift detection methods identify change points in data distribution and have been studied for over a half century (Page, 1954; Bifet and Gavaldà, 2007; Gama et al., 2004; Mayaki and Riveill, 2022). Concept drift can be detected by applying these drift detection methods to the stream of the prediction losses (Mehmood et al., 2021; Gonçalves et al., 2014) and this can be applied to our problem by detecting concept drift from the present $t = T$ backward to the past $t = 1$ and selecting samples until a drift is detected. However, as noted by Hinder et al. (2022), "*... if a drift only occurs in a small region of the entire feature space, the other non-drifted regions may also be suspended, thereby reducing the learning efficiency of models.*", these time-based methods often fail to flexibly select samples, which can decrease the efficiency of learning models.

Drift Localization Methods. Unlike traditional drift detection methods that determine *when* drift occurs, recent drift localization techniques identify *where* drift happens. Liu et al. (2017) introduce LDD-DIS, which detects local drift by comparing the number of recent and old samples in the k-nearest neighbors among the data. Building on this, LDD-DSDA is developed to select samples for the problem. Hinder et al. (2022) propose a theoretical framework called LCD, which reframes drift localization as a supervised classification problem, offering improved detection performance over LDD-DIS. However, both methods lack theoretical analysis for model training and often struggle to select samples effectively.

[1] https://automl.github.io/auto-sklearn/master/

[2] https://pycaret.org/

Density Ratio Estimation. The density ratio, which compares two probability distributions, has been a research focus for over two decades (Shimodaira, 2000). Kernel based methods, such as KLEIP (Sugiyama et al., 2007a), uLSIF (Kanamori et al., 2009), RuLSIF (Yamada et al., 2013), KMM (Schölkopf et al., 2007), and other methods (Sugiyama et al., 2012; Kato and Teshima, 2021; Zhang et al., 2020) have been proposed and utilized not only for covariate shift adaptation (Shimodaira, 2000; Sugiyama et al., 2007a), but also for generative models (Goodfellow et al., 2014), mutual information approximation (Suzuki et al., 2009), and change point detection (Liu et al., 2013). Various extensions exist, such as joint-to-marginal (Matsushita et al., 2022), conditional distribution given input (Sugiyama et al., 2010) and output (Sugiyama, 2010), and continuous covariate shift (Zhang et al., 2023). However, joint density ratio estimation, crucial for addressing our problem, remains insufficiently explored.

3. Proposed Method

This section introduces our method, TSJD, a training data soft selection method based on joint density ratio estimation. Section 3.1 explains the notation and assumptions while Section 3.2 present the algorithm of TSJD. Section 3.3 describes training of the joint density estimator, and Section 3.4 offers details on modeling and hyperparameter tuning.

3.1. Notation and Assumption

We define the marginalization of the N-recent data distribution as

$$\bar{p}_T(\mathbf{x}, y) := \frac{1}{N} \sum_{t=T-N+1}^{T} p_t(\mathbf{x}, y), \tag{6}$$

where we use the subscript T to denote *Target.* We consider the data $D_T := \{(\mathbf{x}_t, y_t)\}_{t=T-N+1}^{T}$ to be approximately i.i.d. samples from the distribution $\bar{p}_T(\mathbf{x}, y)$. Similarly, we define

$$\bar{p}_S(\mathbf{x}, y) := \frac{1}{T-N} \sum_{t=1}^{T-N} p_t(\mathbf{x}, y), \tag{7}$$

as the old data distribution, and $D_S := \{(\mathbf{x}_t, y_t)\}_{t=1}^{T-N}$ is considered as samples of size $(T-N)$ from the distribution $\bar{p}_S(\mathbf{x}, y)$. Here, the subscript S is short for *Source.* Moreover, we define the *test* distribution p_{te} as

$$p_{\mathrm{te}}(\mathbf{x}, y) := \frac{1}{M} \sum_{T+1}^{T+M} p_t(\mathbf{x}, y). \tag{8}$$

and assume that $R_{01}(h) = \mathbb{E}_{p_{\mathrm{te}}(\mathbf{x},y)}[\ell_{01}(h(\mathbf{x}), y)]$.

These formulation and assumption allow us to view the problem of Definition 1 as one that to relate the three distributions $\bar{p}_S, \bar{p}_T$, and p_{te}. With this understanding, we present our method in the next section.

3.2. Algorithm of TSJD

To derive our proposed method, we establish a key theorem that links the zero-one risk R_{01} with the squared L^2-norm of the difference between $h(\mathbf{x})$ and $\bar{p}_T(\cdot|\mathbf{x})$ over $\bar{p}_T$, as follows.

Theorem 3 *For any $h \in \mathcal{H}$, the following holds.*

$$R_{01}(h) - B_{01} = \mathcal{O}\left(\mathop{\mathbb{E}}_{\bar{p}_T(\mathbf{x})} \left[\|\bar{p}_T(\cdot|\mathbf{x}) - h(\mathbf{x})\|_2^2 \right] + Z(h)\tau_X + \tau_{Y|X}^2 \right), \tag{9}$$

where we define $Z(h) := \sup_{\mathbf{x}\in\mathcal{X}} \left\| \nabla \|\bar{p}_T(\cdot|\mathbf{x}) - h(\mathbf{x})\|_2^2 \right\|_2$ and $B_{01} := \min_f R_{01}(f)$ as the Bayes error, i.e., B_{01} is the lowest value of R_{01} among any possible classification model $f : \mathcal{X} \to \Delta^{K-1}$.

Theorem 3 indicates that fitting $h(\mathbf{x})$ to $\bar{p}_T(\cdot|\mathbf{x})$ is sufficient for the problem, i.e., selecting samples to make h learn $\bar{p}_T$ solves the problem of Definition 1.

Remark 4 *Although we have another h-related term $Z(h)\tau_X$ beside the first term $\mathbb{E}_{\bar{p}_T(\mathbf{x})}\left[\|\bar{p}_T(\cdot|\mathbf{x}) - h(\mathbf{x})\|_2^2\right]$ in Eq. (9), it can be considered negligible due to the following reasons;*

- *Small τ_X Assumption: The value of τ_X is assumed to be a small constant, e.g., $\tau_X \ll 1$, inherently reducing the impact of the term $Z(h)\tau_X$.*
- *Convergence of $Z(h)$: Even if τ_X is not particularly small, $Z(h) \to 0$ holds with the first term in Eq. (9) converges to zero, i.e., $\mathbb{E}_{\bar{p}_T(\mathbf{x})}\left[\|\bar{p}_T(\cdot|\mathbf{x}) - h(\mathbf{x})\|_2^2\right] \to 0$, further diminishing the significance of the term $Z(h)\tau_X$.*

This understanding makes us to use the following weighting strategies;

Weights for the recent samples. We set the weight $w_t \propto 1$ for all $t \in \{T-N+1, ..., T\}$, as the naive approach does. Since D_T is assumed to be sampled from $\bar{p}_T$, this weights enable us to empirically approximate the the expected cross-entropy loss of h over $\bar{p}_T$, denoted by $L_{\mathrm{CE}}(h)$ as

$$L_{\mathrm{CE}}(h) := \mathop{\mathbb{E}}_{\bar{p}_T(\mathbf{x},y)}[\ell_{\mathrm{CE}}(h(\mathbf{x}), y)] \approx \frac{1}{|D_T|} \sum_{(\mathbf{x},y)\in D_T} \ell_{\mathrm{CE}}(h(\mathbf{x}), y). \tag{10}$$

By the strict-properness of the the cross-entropy loss (Gneiting and Raftery, 2007) and Theorem 3, the minimization of Eq. (10) leads to minimize $R_{01}(h)$.

Weights for the old samples. For the old samples with $t \in [T-N]$, we apply *the importance weighting* technique (Shimodaira, 2000; Sugiyama et al., 2007a), initially proposed for covariate shift adaptation; Let $r(\mathbf{x}, y) := \frac{\bar{p}_T(\mathbf{x},y)}{\bar{p}_S(\mathbf{x},y)}$ be the joint density ratio of $\bar{p}_T$ over $\bar{p}_S$. Then, we have

$$\mathop{\mathbb{E}}_{\bar{p}_T(\mathbf{x},y)}[f(\mathbf{x}, y)] = \int f(\mathbf{x}, y)\frac{\bar{p}_T(\mathbf{x}, y)}{\bar{p}_S(\mathbf{x}, y)}\bar{p}_S(\mathbf{x}, y)d\mathbf{x}dy = \mathop{\mathbb{E}}_{\bar{p}_S(\mathbf{x},y)}[r(\mathbf{x}, y)f(\mathbf{x}, y)]. \tag{11}$$

Hence, setting a sample weight w_t to be $w_t \propto r(\mathbf{x}_t, y_t)$ converts the expectation over $\bar{p}_S$ into that over $\bar{p}_T$. With the same logic for the recent samples, the minimization of $\ell_{\mathrm{CE}}(h(\mathbf{x}_t), y_t)$

Algorithm 1 Main algorithm of TSJD

```
Input: Data $D = ((\mathbf{x}_1, y_1), ..., (\mathbf{x}_T, y_T)) \in (\mathcal{X} \times \mathcal{Y})^T$, Number of recent samples $N \in [T-1]$
    w.r.t. Assumption 1
    // Step 1
    $\forall t \in \{T - N + 1, ..., T\}, w_t \leftarrow \frac{1}{2N}$
    // Step 2
    Train a joint density ratio estimator $\widehat{g} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ based on $D_T$ and $D_S$
    $\forall t \in [T - N], w_t \leftarrow \frac{1}{2(T-N)} \widehat{g}(\mathbf{x}_t, y_t)$
Output: Sample weights $W = [w_1, ..., w_T]^T \in \mathbb{R}^T_{\geq 0}$
```

with the weight $w_t = r(\mathbf{x}_t, y_t)$ over $\bar{p}_S$ leads to minimize $R_{01}(h)$. Since $r(\mathbf{x}, y)$ is not explicitly available, we train a joint density ratio estimator $\widehat{g} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ and use $w_t \propto \widehat{g}(\mathbf{x}_t, y_t)$ for the weight for all $t \in [T - N]$.

Algorithm 1 summarizes our method for the training data selection problem, where we normalize the weights using N and T. By our Algorithm 1, the model h will be trained to minimize $\widehat{L}_{\mathrm{CE}}(h; D_S, D_T)$ defined as

$$\widehat{L}_{\mathrm{CE}}(h; D_S, D_T) := \frac{1}{2}\left(\frac{1}{|D_T|}\sum_{(\mathbf{x},y)\in D_T} \ell_{\mathrm{CE}}(h(\mathbf{x}), y) + \frac{1}{|D_S|}\sum_{(\mathbf{x},y)\in D_S} \widehat{g}(\mathbf{x}, y)\ell_{\mathrm{CE}}(h(\mathbf{x}), y)\right), \tag{12}$$

with our joint density ratio estimator $\widehat{g}$. Next, we specify how to train $\widehat{g}$ based on D_S and D_T.

3.3. Training the Density Ratio Estimator $\widehat{g}$

We train a density ratio estimator $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ which approximates the true density ratio $r(\mathbf{x}, y) = \frac{\bar{p}_T(\mathbf{x},y)}{\bar{p}_S(\mathbf{x},y)}$ by minimizing the expected squared error $J(g)$ over $\bar{p}_S$ (Kanamori et al., 2009; Zhang et al., 2020);

$$J(g) := \mathop{\mathbb{E}}_{\bar{p}_S(\mathbf{x},y)}\left[\left(g(\mathbf{x}, y) - r(\mathbf{x}, y)\right)^2\right], \tag{13}$$

whose empirical version $\widehat{J}(g; D_S, D_T)$ is defined as

$$\widehat{J}(g; D_S, D_T) := \frac{1}{|D_S|}\sum_{(\mathbf{x},y)\in D_S} g(\mathbf{x}, y)^2 - \frac{2}{|D_T|}\sum_{(\mathbf{x},y)\in D_T} g(\mathbf{x}, y) + C_r, \tag{14}$$

where $C_r := \mathbb{E}_{\bar{p}_S(\mathbf{x},y)}\left[r(\mathbf{x}, y)^2\right] = \mathbb{E}_{\bar{p}_T(\mathbf{x},y)}[r(\mathbf{x}, y)]$ is an independent constant and can be ignored to train g. In addition, g needs to satisfy

$$1 = \mathop{\mathbb{E}}_{\bar{p}_S(\mathbf{x},y)}[g(\mathbf{x}, y)] \approx \frac{1}{|D_S|}\sum_{(\mathbf{x},y)\in D_S} g(\mathbf{x}, y) \tag{15}$$

to be a proper density ratio due to the fact $\mathbb{E}_{\bar{p}_S(\mathbf{x},y)}[r(\mathbf{x},y)] = \int \bar{p}_T(\mathbf{x},y)d\mathbf{x}dy = 1$. Hence, we add an empirical constraint with a hyperparameter $\beta > 0$. The final loss function to train g is defined as

$$\begin{aligned} L(g; D_S, D_T) :=& \frac{1}{|D_S|} \sum_{(\mathbf{x},y)\in D_S} g(\mathbf{x},y)^2 - \frac{2}{|D_T|} \sum_{(\mathbf{x},y)\in D_T} g(\mathbf{x},y) \\ &+ \beta \left(\frac{1}{|D_S|} \sum_{(\mathbf{x},y)\in D_S} g(\mathbf{x},y) - 1 \right)^2, \end{aligned} \tag{16}$$

and we denote the minimizer of $L(g; D_S, D_T)$ by $\widehat{g}$.

Remark 5 *The constraint Eq. (15) is often overlooked in existing methods (Kanamori et al., 2009; Yamada et al., 2013; Zhang et al., 2020), as claimed by Sugiyama et al. (2007b) "... the normalization constraint (Eq. (15)) is not generally satisfied exactly ... this may not be critical in practice since the scale of the importance is often irrelevant in subsequent learning algorithms.". However, since we use both D_S and D_T, the correct scale is vital to control and balance the effects of the sample weights. Additionally, in our analysis, we assume $\bar{p}_S(\mathbf{x},y)g(\mathbf{x},y)$ is a probability density. Therefore, contrary to the claim, the constraint term is crucial in our problem.*

3.4. Implementation and Hyperparameter Tuning of $\widehat{g}$

We employ a linear-in-parameter model (Zhang et al., 2020) (a.k.a. linear basis expansion) with the softplus activation; $\mathrm{softplus}(x) := \log(1 + \exp(x))$ for g as

$$g(\mathbf{x},y) = \mathrm{softplus}\left(\sum_{i=1}^{N_M} a_i \phi_i(\mathbf{x},y) \right), \tag{17}$$

where $a_i \in \mathbb{R}$ is the learning parameter, $\phi_i : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is the i-the feature mapping (a.k.a. basis function), and $N_M = 200$ is the number of the feature mappings. The feature mapping ϕ_i is modeled using the Gaussian RBF as $\phi_i(\mathbf{x},y) := \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|_2^2}{\sigma_x}\right) \max(\sigma_y, \mathbb{I}[y = y_i])$, where $(\mathbf{x}_i, y_i)$ is the kernel center, sampled from D_T uniformly at random, $\sigma_x > 0$ and $\sigma_y \geq 0$ are the hyperparameters. The parameters $\{a_i\}_{i=1}^{N_M}$ are optimized using gradient descent.

The hyperparameters of TSJD, i.e., σ_x, σ_y, and β, are tuned by a grid search and ones that minimize $\widehat{J}(\widehat{g}; D_S, D_T)$, satisfying the following constraint is selected;

$$\left| \frac{1}{|D_S|} \sum_{(\mathbf{x},y)\in D_S} \widehat{g}(\mathbf{x},y) - 1 \right| \leq G \sqrt{\frac{\log \frac{2}{\delta}}{2|D_S|}}, \tag{18}$$

where $\widehat{g}$ is obtained by minimizing Eq. (16) with each set of the hyperparameters, and we set $G = 10$ and $\delta = 0.05$. Note this constraint is different from the term inside Eq. (16); This is based on the following lemma, that states that even the training is perfect, i.e., $\widehat{g} = r$, the constraint Eq. (15) can only be satisfied with a margin $G\sqrt{\frac{\log \frac{2}{\delta}}{2|D_S|}}$ in probability. The proof is omitted since it is trivial by Hoeffding's inequality.

Lemma 6 *Assume $r(\mathbf{x}, y) \le G$ for any $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$. Then, for any $\delta \in (0, 1)$, the following holds with probability at least $1 - \delta$.*

$$\left| \frac{1}{|D_S|} \sum_{(\mathbf{x},y)\in D_S} r(\mathbf{x}, y) - 1 \right| \le G \sqrt{\frac{\log \frac{2}{\delta}}{2|D_S|}} \tag{19}$$

4. Theoretical Analysis

In this section, we provide our theoretical analysis, bounding the generalization error of models trained with our method. Before presenting our analysis, we clarity the notation and assumption used in our analysis as follows;

- Let $\mathcal{G}_+$ be the hypothesis space for the joint density ratio predictor $g : \mathcal{X} \times \mathcal{Y} \to [0, G]$ with a constant $G \ge 1$ and assume that $\forall g \in \mathcal{G}_+$, $\mathbb{E}_{\bar{p}_S(\mathbf{x},y)}[g(\mathbf{x}, y)] = 1$ holds.
- Let $\mathcal{G}$ be defined as $\mathcal{G} := \mathcal{G}_+ \cup \{g' : (\mathbf{x}, y) \mapsto -g(\mathbf{x}, y) | g \in \mathcal{G}_+\}$.
- Assume that $\forall(\mathbf{x}, y, h) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{H} : \ell_{\mathrm{CE}}(h(\mathbf{x}), y) \le U$ holds with a constant $U \ge 0$.

4.1. Main Result

Our analysis yields an upper bound on the zero-one risk of $\widehat{h}$, trained with weights computed by our method. The main theorem detailing the generalization error bound and its order is presented in Theorem 7 and Corollary 8, respectively. Notably, $C_4(\delta) = \mathcal{O}(\mathfrak{R}_N(\mathcal{H}) + \mathfrak{R}_{T-N}(\mathcal{H}))$ and $C_3(\delta) = \mathcal{O}(\mathfrak{R}_N(\mathcal{G}) + \mathfrak{R}_{T-N}(\mathcal{G}))$ are defined using the Rademacher complexity $\mathfrak{R}$ (Koltchinskii, 2001; Cortes et al., 2016; Maurer, 2016; Ledoux and Talagrand, 2013; Mohri et al., 2018). The exact definitions and notation will be provided in the subsequent sections.

Theorem 7 (Generalization error bound) *For any $\delta \in (0, 1)$, the following inequality holds with probability at least $1 - \delta$;*

$$\begin{aligned} &R_{01}(\widehat{h}) - B_{01} \\ &\le 4K\eta_{min}^{-2}\left(T_{KL}(h^*) + U\sqrt{J(g^*)} + C_4(\delta/5) + U\sqrt{C_3(\delta/5)} + \frac{1}{K}Z(\widehat{h})\tau_X + \frac{1}{K}\tau_{Y|X}^2\right), \end{aligned} \tag{20}$$

where $T_{KL}(h)$ is the expected Kullback-Leibler divergence between $\bar{p}_T(\cdot|\mathbf{x})$ and $h(\mathbf{x})$ over $\bar{p}_T(\mathbf{x})$, i.e.,

$$T_{KL}(h) := \mathop{\mathbb{E}}_{\bar{p}_T(\mathbf{x})}[D_{KL}(\bar{p}_T(\cdot|\mathbf{x})||h(\mathbf{x}))]. \tag{21}$$

Corollary 8 *Assume that $\mathfrak{R}_n(\mathcal{G}) = \mathcal{O}(n^{-1/2})$ and $\mathfrak{R}_n(\mathcal{H}) = \mathcal{O}(n^{-1/2})$, then following order holds;*

$$R_{01}(\widehat{h}) - B_{01} = \mathcal{O}\Big(T_{KL}(h^*) + \sqrt{J(g^*)} + Z(\widehat{h})\tau_X + \tau_{Y|X}^2\Big) + \mathcal{O}_p\Big(N^{-\frac{1}{4}} + (T - N)^{-\frac{1}{4}}\Big), \tag{22}$$

where $\mathcal{O}_p$ denotes the order in probability.

Remark 9 *Theorem 7 and Corollary 8 show that if $p(y|\mathbf{x}) = h^*(y|\mathbf{x})$ and $r(\mathbf{x}, y) = g^*(\mathbf{x}, y)$ hold for any $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, then as N and T increase, the difference between $R_{01}(\widehat{h})$ and the Bayes error B_{01} approaches $\mathcal{O}\Big(Z(\widehat{h})\tau_X + \tau_{Y|X}^2\Big)$, which is inevitable due to data drift. Therefore, the generalization error of $\widehat{h}$ can be considered optimal, theoretically validating our method of Algorithm 1.*

In the following sections, we introduce the key lemmas and theorems for deriving Theorem 7.

4.2. Generalization Error Bound of $\widehat{g}$

We first establish the generalization error bound of our joint density ratio estimator $\widehat{g} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ in terms of the expected squared error J.

Theorem 10 *Let $\widehat{g}$ and g^* be the minimizers of $\widehat{J}(g; D_S, D_T)$ and $J(g)$ among $\mathcal{G}_+$, respectively. Then, for any $\delta \in (0, 1)$, the following inequality holds with probability at least $1 - \delta$;*

$$J(\widehat{g}) \leq J(g^*) + C_3(\delta), \tag{23}$$

where $C_3(\delta) := 4G\mathfrak{R}_{|D_S|}(\mathcal{G}) + 4\mathfrak{R}_{|D_T|}(\mathcal{G}) + 4G^2\sqrt{\frac{\log\frac{3}{\delta}}{2}}\left(\frac{1}{\sqrt{|D_S|}} + \frac{1}{\sqrt{|D_T|}}\right)$ and $\mathfrak{R}_n(\mathcal{G})$ is the Rademacher complexity (Koltchinskii, 2001) of $\mathcal{G}$ with sampling size n.

The following corollary is obvious from Theorem 10.

Corollary 11 *Assume that $\mathfrak{R}_n(\mathcal{G}) = \mathcal{O}(n^{-1/2})$, then following order holds;*

$$J(\widehat{g}) = J(g^*) + \mathcal{O}_p\Big(N^{-1/2} + (T - N)^{-1/2}\Big), \tag{24}$$

Remark 12 *Corollary 11 indicates that if $\mathcal{G}_+$ is properly chosen and $r = g^* \in \mathcal{G}_+$, the right hand of Eq. (24) decreases to 0 at the rate of $(T - N)^{-1/2} + N^{-1/2}$ in probability. This ensures that $\widehat{g}$ converges to r as N and T approach infinity, confirming the theoretical soundness of our method for training g.*

4.3. Generalization Error Bound of h Trained with Our Method

Next, we analyze the generalization error of $\widehat{h}$, a clasification model trained with our selected training sample using $\widehat{g} \in \mathcal{G}_+$. The following Lemma 13 and Lemma 14 provide the relation between $L_{\mathrm{CE}}(h)$ and empirical error of a classification model h.

Lemma 13 *For any $\delta \in (0, 1)$ and $h \in \mathcal{H}$, over the draw of i.i.d. samples S_T from $\bar{p}_T$, the following inequality holds with probability at least $1 - \delta$;*

$$L_{\mathrm{CE}}(h) \leq \frac{1}{|S_T|}\sum_{(\mathbf{x},y)\in S_T} \ell_{\mathrm{CE}}(h(\mathbf{x}), y) + C_1(\delta) \tag{25}$$

where $C_1(\delta) := 2\sqrt{2}\exp(U)\mathfrak{R}_{|S_T|}(\mathcal{H}) + U\sqrt{\frac{\log\frac{1}{\delta}}{2|S_T|}}$ and $\mathfrak{R}_n(\mathcal{H})$ is the vector-valued Rademacher complexity (Maurer, 2016; Cortes et al., 2016) of $\mathcal{H}$ with sampling size n.

Table 1: Dataset statistics.

Data set	Samples	Features	Classes
Weather	18159	8	2
Smartmeter	22950	96	10
Powersupply	29928	2	24
Forest	581012	54	2

Lemma 14 *For any $\delta \in (0,1)$ and any $h \in \mathcal{H}$, over the draw of i.i.d. samples S from $\bar{p}_S$, the following inequality holds with probability at least $1-\delta$:*

$$L_{\mathrm{CE}}(h) \quad \leq \frac{1}{|S|}\sum_{(\mathbf{x},y)\in S} g(\mathbf{x},y)\ell_{\mathrm{CE}}(h(\mathbf{x}),y) + C_2(\delta) + U\sqrt{\mathop{\mathbb{E}}_{\bar{p}_S(\mathbf{x},y)}\left[(r(\mathbf{x},y)-g(\mathbf{x},y))^2\right]} \tag{26}$$

where $C_2(\delta) := 2(2U+G)\exp(U)\mathfrak{R}_{|S|}(\mathcal{H}) + 2(U+2G)\mathfrak{R}_{|S|}(\mathcal{G}) + MG\sqrt{\frac{\log\frac{1}{\delta}}{2|S|}}$.

Based on Lemma 13 and Lemma 14, we obtain the generalization error bound w.r.t. L_{CE}.

Theorem 15 *Let $\widehat{h}$ and h^* be the minimizers of $\widehat{L}_{\mathrm{CE}}(h)$ and $L_{\mathrm{CE}}(h)$ among $\mathcal{H}$, respectively. Then, for any $\delta \in (0,1)$, the following inequality holds with probability at least $1-\delta$;*

$$L_{CE}(\widehat{h}) - L_{CE}(h^*) \leq U\sqrt{J(g^*)} + C_4(\delta/5) + U\sqrt{C_3(\delta/5)} \tag{27}$$

where we define

$$\begin{aligned} C_4(\delta) := {} & \sqrt{2}\exp(U)\mathfrak{R}_{|D_T|}(\mathcal{H}) + U\sqrt{\frac{\log\frac{1}{\delta}}{2|D_T|}} + (2U+G)\exp(U)\mathfrak{R}_{|D_S|}(\mathcal{H}) \\ & + (U+2G)\mathfrak{R}_{|D_S|}(\mathcal{G}) + GU\sqrt{\frac{\log\frac{1}{\delta}}{2|D_S|}}. \end{aligned} \tag{28}$$

Based on Theorem 15 and Theorem 3, we derive the generalization upper bound for the risk $R_{01}(\widehat{h})$, which is our final target to minimize. The bound is presented in Section 4.1, and we have already discussed its implication, showing theoretical validity of our method.

5. Numerical Experiments

We conducted experiments to test the empirical effectiveness of our method on real-world datasets.

Dataset. We use four real-world multi-class classification datasets obtained from USP DS Repository (Souza et al., 2020)[3]. We select two severely drifting (Powersupply and Forest) and two relatively stationary datasets (Weather and Smartmeter), as shown in Table 1.

[3] https://sites.google.com/view/uspdsrepository, Accessed: 2025-06-24

Table 2: Average zero-one loss (↓) over 30 random trials. **Boldfaces with star*** highlight the lowest errors and basic **boldfaces** show comparable results based on the Wilcoxon signed-rank test (Wilcoxon, 1945) with the significance level of 1%.

Data	Model	N	T	Naive Baseline		Time-based		Cov.shift	Drift Localization		(Ours)
				D_T	D	PHT	ADWIN	uLSIF	LDD-DSDA	LCD	TSJD
Weather	LGBM	200	2000	29.57	**21.97***	**22.00**	**22.30**	28.33	24.60	**22.53**	**22.90**
		500	5000	19.90	**17.73**	**17.73**	**17.60***	19.63	**18.70**	**18.33**	**18.43**
		1000	10000	23.70	**20.57**	**21.13**	**21.30**	22.93	**21.03**	**20.43***	**21.37**
	NN	200	2000	27.50	**20.63**	**21.13**	**20.47***	26.43	**22.73**	**20.63**	**22.03**
		500	5000	19.90	**17.57**	**17.63**	**17.53***	**17.83**	**18.47**	**17.80**	**18.00**
		1000	10000	**20.70**	**18.93***	**19.10**	**19.37**	**21.03**	**19.93**	**19.60**	**19.23**
Smartmeter	LGBM	200	2000	26.67	19.90	20.07	**20.10**	27.40	21.77	20.13	**17.67***
		500	5000	22.97	**13.50**	17.33	19.90	24.80	16.73	**14.33**	**12.83***
		1000	10000	21.13	**13.13**	17.40	19.57	21.90	15.40	**12.70**	**12.07***
		2000	20000	22.23	15.20	17.43	17.80	23.23	16.53	15.67	**13.37***
	NN	200	2000	36.30	36.80	36.73	**33.23**	40.00	37.77	36.43	**31.00***
		500	5000	35.27	33.33	35.27	36.43	37.43	34.27	33.27	**29.20***
		1000	10000	36.10	**31.53**	39.13	37.83	37.13	34.83	**31.80**	**30.13***
		2000	20000	37.23	**32.17**	33.80	36.10	37.97	34.33	**31.60**	**30.07***
Powersupply	LGBM	200	2000	**80.43***	85.20	85.33	85.60	84.70	85.10	85.53	**83.43**
		500	5000	**79.60***	83.60	83.70	84.10	83.77	83.43	84.73	**81.80**
		1000	10000	**82.57***	**84.20**	86.17	**85.57**	85.00	**84.10**	**85.37**	**83.30**
		2000	20000	**81.63**	**80.53***	82.93	**82.00**	**81.87**	**80.57**	**81.53**	**82.07**
	NN	200	2000	**85.30**	**83.27**	**83.27**	**83.60**	**84.70**	**86.20**	**83.90**	**82.23***
		500	5000	**81.33**	82.57	82.77	82.93	**80.63**	**81.87**	83.90	**78.83***
		1000	10000	**80.67**	83.77	85.07	83.80	**81.17**	**82.47**	**84.03**	**79.77***
		2000	20000	**78.40**	**78.77**	**79.13**	**78.37**	**77.60***	**78.60**	**79.67**	**78.23**
Forest	LGBM	200	2000	34.00	**3.13**	**4.37**	12.40	31.83	9.93	**2.77***	**2.93**
		500	5000	**14.77**	**4.40**	**4.00***	**4.33**	15.47	**5.40**	**4.20**	**4.80**
		1000	10000	**3.07**	**3.43**	**3.63**	**3.00**	**3.43**	**3.53**	3.60	**2.77***
		2000	20000	**5.07**	6.67	6.50	**5.00**	**6.20**	6.40	7.03	**4.43***
	NN	200	2000	35.90	**3.80**	**6.17**	16.10	32.70	9.40	**3.43***	**4.07**
		500	5000	14.53	**4.73**	**5.03**	**3.93***	15.40	5.20	**4.43**	**3.97**
		1000	10000	**4.00**	5.07	5.20	**4.20**	**4.27**	**4.87**	**4.70**	**3.60***
		2000	20000	**5.83**	8.60	8.57	**6.27**	8.17	8.77	8.87	**5.63***
Average Rank				5.27	3.40	4.93	4.43	6.00	5.03	4.37	**2.40***
#Best				3	3	1	4	1	0	3	**15***
#Best or Comparable				14	19	13	18	10	14	20	**30***

Setting. We vary N among $200, 500, 1000$, and 2000, setting $T = 10N$. The number of test data, M, is consistently set to 100 across all settings. In each dataset, we select continuous T samples starting from a randomly chosen index and use them for D. The subsequent M samples form the test data, D^{te}. Using each baseline and our method, we select the training data from D and then train a classifier $\widehat{h}$. The classifier is either modeled by LightGBM (Ke et al., 2017) or a three-layer neural network with 100 hidden units as two representative classification models. The evaluation is based on the average zero-one loss on D^{te}, i.e., $\widehat{R}_{01}(\widehat{h}; D^{\mathrm{te}}) = \frac{1}{|D^{\mathrm{te}}|} \sum_{(\mathbf{x},y)\in D^{\mathrm{te}}} \ell_{01}\left(\widehat{h}(\mathbf{x}), y\right)$. We repeat each setting for 30 times with different random seeds, and report the average.

Comparison methods. We compare our method with seven various baselines, including naive baselines, drift detection, covariate shift adaptation, and drift localization methods as follows;

- D_T, D: Naive baselines. Naively use each of the recent data D_T and whole data D.
- PHT (page-hinkley test) (Page, 1954), ADWIN (Bifet and Gavaldà, 2007): Representative time-based drift detection methods. First train a LightGBM classification model on D_S and then apply drift detection to the prediction loss from the present to the past. We select samples until a drift is detected.
- uLSIF (Kanamori et al., 2009): Covariate shift adaptation method; a variant of our approach, not with *joint* density ratio, but with covariate density ratio. Efficient hyperparameter tuning proposed by the authors is conducted for each experiment.
- LDD-DSDA (Liu et al., 2017): An existing method for training data selection, which selects samples based on drift localization method, LDD-DIS. Default parameters provided by the authors are used.
- LCD (Hinder et al., 2022): A drift localization method; we use all D_T and *no-drifting* samples (p-value of drift ≥ 0.05) in D_S. Parameters provided by the authors are used.
- TSJD: Our method detailed in Section 3. Hyperparameters are pre-tuned for each dataset and N pair using the entire dataset based on Section 3.4.

Results. The results are presented in Table 2. Our method achieves the best average rank of 2.40 and consistently shows the best or comparable results across all datasets and settings. Among the baselines, LCD achieves the best or comparable results 20 times. However, its average rank is 4.37, which is worse than the naive baseline using D. This highlights the weakness of LCD and underscores the superiority of TSJD. Overall, these findings empirically demonstrate the effectiveness and versatility of our method for the problem of training data selection.

6. Conclusion

This paper studied the training data selection problem, focusing on the selection of effective samples to improve model training from drifting data. We proposed TSJD, which assigns training weights for each sample based on joint density ratio estimation. We provide a theoretical analysis that bounds the generalization error of our method. Extensive experiments on real-world datasets demonstrate the superiority of TSJD over baseline methods.

References

Pranjal Awasthi, Corinna Cortes, and Mehryar Mohri. Best-effort adaptation. *Annals of Mathematics and Artificial Intelligence*, 92(2), 2024. doi: 10.1007/s10472-023-09917-3.

Raef Bassily, Corinna Cortes, Anqi Mao, and Mehryar Mohri. Differentially private domain adaptation with theoretical guarantees. In *International Conference on Machine Learning*, 2024.

Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *SDM*, 2007.

Dariusz Brzezinski and Jerzy Stefanowski. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 25(1), 2014.

Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohrii, and Scott Yang. Structured prediction theory based on factor graph complexity. In *International Conference on Neural Information Processing Systems*, 2016.

David Albert Edwards. On the kantorovich–rubinstein theorem. *Expositiones Mathematicae*, 29(4), 2011. doi: 10.1016/j.exmath.2011.06.005.

João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *Advances in Artificial Intelligence*, 2004. doi: 10.1007/978-3-540-28645-5_29.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477), 2007. doi: 10.1198/016214506000001437.

Paulo M. Gonçalves, Silas G.T. de Carvalho Santos, Roberto S.M. Barros, and Davi C.L. Vieira. A comparative study on concept drift detectors. *Expert Systems with Applications*, 41(18), 2014. doi: 10.1016/j.eswa.2014.07.019.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *International Conference on Neural Information Processing Systems*, 2014.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*, volume 1. Springer Series in Statistics, 2001.

Fabian Hinder, Valerie Vaquet, Johannes Brinkrolf, André Artelt, and Barbara Hammer. Localization of concept drift: Identifying the drifting datapoints. In *International Joint Conference on Neural Networks*, 2022. doi: 10.1109/IJCNN55064.2022.9892374.

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *J. Mach. Learn. Res.*, 10, 2009.

Masahiro Kato and Takeshi Teshima. Non-negative bregman divergence minimization for deep direct density ratio estimation. In *International Conference on Machine Learning*, volume 139, 2021.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5), 2001.

Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access*, 11, 2023.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Classics in Mathematics. Springer Berlin Heidelberg, 2013.

Anjin Liu, Yiliao Song, Guangquan Zhang, and Jie Lu. Regional concept drift detection and density synchronized drift adaptation. In *International Joint Conference on Artificial Intelligence*, 2017. doi: 10.24963/ijcai.2017/317.

Song Liu, Makoto Yamada, Nigel Collier, and Masashi Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43, 2013. doi: 10.1016/j.neunet.2013.01.012.

Yukitoshi Matsushita, Taisuke Otsu, and Keisuke Takahata. Estimating density ratio of marginals to joint: Applications to causal inference. *Journal of Business and Economic Statistics*, 2(41), 2022.

Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *Algorithmic Learning Theory*, 2016.

Mansour Zoubeirou A. Mayaki and Michel Riveill. Autoregressive based drift detection method. *International Joint Conference on Neural Networks*, 2022.

Hassan Mehmood, Panos Kostakos, Marta Cortes, Theodoros Anagnostopoulos, Susanna Pirttikangas, and Ekaterina Gilman. Concept drift adaptation techniques in distributed environment for real-world data streams. *Smart Cities*, 4(1), 2021. doi: 10.3390/smartcities4010021.

Mehryar Mohri and Andres Muñoz Medina. New analysis and algorithm for learning with drifting distributions. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, 2012. doi: 10.1007/978-3-642-34106-9_13.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, second edition, 2018.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41, 1954.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

Philipp Ruf, Manav Madan, Christoph Reich, and Djaffar Ould-Abdeslam. Demystifying mlops and presenting a recipe for the selection of open-source tools. *Applied Sciences*, 2021.

Bernhard Schölkopf, John Platt, and Thomas Hofmann. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, 2007.

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

Vinicius M. A. Souza, Denis M. dos Reis, André Gustavo Maletzke, and Gustavo E. A. P. A. Batista. Challenges in benchmarking stream learning algorithms with real-world data. *Data Mining and Knowledge Discovery*, 34, 2020.

Masashi Sugiyama. Superfast-trainable multi-class probabilistic classifier by least-squares posterior fitting. *IEICE TRANSACTIONS on Information*, E93-D(10), 2010. doi: 10.1587/transinf.E93.D.2690.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Neural Information Processing Systems*, 2007a.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Vonbunau, and Motoaki Kawanabe. Kullback-leibler importance estimation procedure for covariate shift adaptation. *JSAI Technical Report, Type 2 SIG*, (DMSM-A702), 2007b. doi: 10.11517/jsaisigtwo.2007.DMSM-A702_03.

Masashi Sugiyama, Ichiro Takeuchi, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Daisuke Okanohara. Conditional density estimation via least-squares density ratio estimation. In *International Conference on Artificial Intelligence and Statistics*, volume 9, 2010.

Masashi Sugiyama, Teruyuki Suzuki, and Takafumi Kanamori. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64, 2012.

Taiji Suzuki, Masashi Sugiyama, and Toshiyuki Tanaka. Mutual information approximation via maximum likelihood estimation of density ratio. In *IEEE International Symposium on Information Theory*, 2009. doi: 10.1109/ISIT.2009.5205712.

Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 1945.

Michał Woźniak. Application of combined classifiers to data stream classification. In *Computer Information Systems and Industrial Management*, 2013.

Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama. Relative density-ratio estimation for robust distribution comparison. *Neural Computation*, 25(5), 2013. doi: 10.1162/NECO_a_00442.

Tianyi Zhang, Ikko Yamane, Nan Lu, and Masashi Sugiyama. A one-step approach to covariate shift adaptation. In *Asian Conference on Machine Learning*, volume 129, 2020.

Yu-Jie Zhang, Zhen-Yu Zhang, Peng Zhao, and Masashi Sugiyama. Adapting to continuous covariate shift via online density ratio estimation. In *International Conference on Neural Information Processing Systems*, 2023.