# Continual Density Ratio Estimation

**Yu Chen**
University of Bristol
yc14600@bristol.ac.uk

**Song Liu**
University of Bristol
song.liu@bristol.ac.uk

**Tom Diethe**
Amazon Research
tdiethe@amazon.com

**Peter Flach**
University of Bristol
Peter.Flach@bristol.ac.uk

## Abstract

In online applications with streaming data, awareness of how far the empirical training or test data has shifted away from its original data distribution can be crucial to the performance of the model. However, historical samples in the data stream may not be kept either due to space requirements or for regulatory reasons. To cope with such situations, we propose Continual Density Ratio Estimation (CDRE), for estimating density ratios between the initial and latest distributions ($p/q_t$) of a data stream without the need of storing past samples, where $q_t$ shifted away from $p$ after a time period $t$. In particular, CDRE is more accurate than standard Density Ratio Estimation (DRE) when the two distributions are less similar, despite not requiring samples from the reference distribution. CDRE can be applied in scenarios of online or continual learning, such as importance weighted covariate shift, measuring dataset changes for better decision making.

## 1 Introduction

Online applications are ubiquitous in practice since large amounts of data are generated and processed in a streaming manner. There are two types of machine learning scenarios commonly deployed for such streaming data: 1) train a model online on the streaming data (*e.g.* online learning [8] and continual learning [5]) – in this case the training set may be shifting over time; 2) train a model offline and deploy it online – in this case the test set may be shifting over time. In both cases, the main problem is data distribution shift, i.e. the data distribution changes gradually over time. Awareness of how far the training or test set has shifted can be crucial to the performance of the model [6]. For example, when the training set is shifting, the latest model may become less accurate on test samples from earlier data distributions. In the other case, the performance of a pre-trained model may gradually degrade when the test set shifts away from the training set over time. It will be beneficial to trace the distribution difference caused by such shift so that we can decide when to update the model for preventing degradation in performance.

Density Ratio Estimation (DRE) [10] is a methodology for estimating the ratio between two probability distributions that can reflect the difference between the two distributions. In particular, it can be applied to settings in which only samples of the two distributions are available, which is usually the case in practice. However, under certain restrictive conditions in online applications – e.g., unavailability of historical samples in an online data stream – existing DRE methods are no longer applicable. More importantly, DRE exhibits difficulties for accurate estimations when there exists significant differences between the two distributions [10, 4, 7].

Workshop on Distribution Shifts, 35th Conference on Neural Information Processing Systems (NeurIPS 2021).

In this paper, we propose a new framework of density ratio estimation called Continual Density Ratio Estimation (CDRE) which is capable of coping with the online scenarios and gives better estimation than standard DRE when the two distributions are less similar.

## 2 Continual Density Ratio Estimation

We first give the problem setting of CDRE. Suppose we want to estimating density ratios between two distributions $r_t(x) = p(x)/q_t(x), t \geq 1$, where $t$ is the index of time steps. We refer to $p(x)$ as the *reference distribution* and $q_t(x)$ as the *dynamic distribution* of $p(x)$. The dynamic distribution is assumed to be shifting away from its reference distribution gradually over time. We assume when $t > 1$ the samples of $p(x)$ are not necessarily available, instead, samples of $q_{t-1}$ and $q_t$ are available.

### 2.1 The basic form of CDRE

Denote the true density ratio $r_t^*(x) \triangleq p(x)/q_t(x)$, it can be decomposed as follows:

$$r_t^*(x) = \frac{q_{t-1}(x)}{q_t(x)} \frac{p(x)}{q_{t-1}(x)} = r_{s_t}^*(x) r_{t-1}^*(x), \;\; r_{s_t}^*(x) \triangleq \frac{q_{t-1}(x)}{q_t(x)} = \frac{r_t^*(x)}{r_{t-1}^*(x)}, \;\; t > 1, \tag{1}$$

where $r_{s_t}^*(x)$ represents the true density ratio between the two latest dynamic distributions. Using this decomposition we can estimate $p(x)/q_t(x)$ in an iterative manner without the need of storing samples from $p(x)$ when $t$ increases. The key point is that we can estimate $r_t^*(x)$ by estimating $r_{s_t}^*(x)$ when the estimation of $r_{t-1}^*(x)$ is known. In particular, this introduces a constraint: $\int r_{s_t}^*(x) q_t(x) dx = \int (r_t^*(x)/r_{t-1}^*(x)) q_t(x) dx = 1$. Existing methods of DRE can be applied to estimating the initial ratio $r_1^*(x)$ and the latest ratio $r_{s_t}^*(x), \forall t > 1$, as the basic ratio estimator of CDRE. Let $r_t(x)$ be the estimation of $r_t^*(x)$, where $r_{t-1}$ is already obtained, then the objective of CDRE can be expressed as:

$$J_{CDRE}(r_t) = J_{DRE}\left(\frac{r_t}{r_{t-1}}\right), \;\; s.t. \;\; \frac{1}{N}\sum_{n=1}^{N} \frac{r_t(x_n)}{r_{t-1}(x_n)} = 1, \;\; x_n \sim q_t(x). \tag{2}$$

where $J_{DRE}$ can be the objective of any method used for standard DRE.

### 2.2 An effective solution for CDRE: Continual KLIEP

Kullback-Leibler Importance Estimation Procedure (KLIEP) is a basic method for density ratio estimation introduced in [11]. We introduce the essential idea of KLIEP in the following. Let $r^*(x) = p(x)/q(x)$ be the (unknown) true density ratio, then $p(x)$ can be estimated by $\tilde{p}(x) = r(x)q(x)$, where $r(x)$ is an estimation of $r^*(x)$. Hence, we can optimize $r(x)$ by minimizing the KL-divergence between $p(x)$ and $\tilde{p}(x)$. The empirical objective of optimizing $r(x)$ is as follows:

$$J_r = \max_r \frac{1}{N}\sum_{i=1}^{N} \log r(x_i), \;\; x_i \sim p(x), \quad s.t. \;\; \frac{1}{M}\sum_{j=1}^{M} r(x_j) = 1, \;\; x_j \sim q(x), \;\; r(x) \geq 0. \tag{3}$$

One convenient way of parameterizing $r(x)$ is by using a log-Neural-Network (log-NN) model with normalization, which then automatically satisfies the constraints in Eq. (3):

$$r(x;\beta) = \frac{\exp(\psi_\beta(x))}{\frac{1}{M}\sum_{j=1}^{M} \exp(\psi_\beta(x_j))}, \quad x_j \sim q(x), \quad \psi_\beta : \mathbb{R}^D \to \mathbb{R}, \tag{4}$$

where $\psi_\beta$ can be any deterministic function: we use a neural network as $\psi_\beta$ in our implementations, $\beta$ then representing parameters of the neural network.

We now demonstrate how to instantiate CDRE by KLIEP, which we call Continual KLIEP (CKLIEP). Define $r_t(x), r_{t-1}(x)$ by the log-NN form as in Eq. (4), let $N_t = N_{t-1} = N$ as the sample size of each distribution, $x_{t,i} \sim q_t(x), \;\; x_{t-1,j} \sim q_{t-1}(x)$, then $r_{s_t}$ is as follows:

$$r_{s_t}(x) = \frac{r_t(x)}{r_{t-1}(x)} = \exp\{\psi_{\beta_t}(x) - \psi_{\beta_{t-1}}(x)\} \times \frac{\frac{1}{N}\sum_{j=1}^{N} \exp\{\psi_{\beta_{t-1}}(x_{t-1,j})\}}{\frac{1}{N}\sum_{i=1}^{N} \exp\{\psi_{\beta_t}(x_{t,i})\}}, \tag{5}$$

where $\beta_t, \beta_{t-1}$ represent parameters of $r_t(x), r_{t-1}(x)$, respectively. We then have the following equality by substituting Eq. (5) into the constraint in Eq. (2):

$$\frac{\sum_{i=1}^{N} \exp\{\psi_{\beta_t}(x_{t,i})\}}{\sum_{j=1}^{N} \exp\{\psi_{\beta_{t-1}}(x_{t-1,j})\}} = \frac{1}{N}\sum_{i=1}^{N} \exp\{\psi_{\beta_t}(x_{t,i}) - \psi_{\beta_{t-1}}(x_{t,i})\} \tag{6}$$

$r_{s_t}$ can then be rewritten in the same form of Eq. (4) by combining Eq. (6) and Eq. (5):

$$r_{s_t}(x) = \frac{\exp\{\phi_{\beta_t}(x)\}}{\frac{1}{N}\sum_{i=1}^{N} \exp\{\phi_{\beta_t}(x_i)\}}, \quad \phi_{\beta_t}(x) \triangleq \psi_{\beta_t}(x) - \psi_{\beta_{t-1}}(x). \tag{7}$$

Now we can instantiate $J_{CDRE}$ in Eq. (2) by the objective of KLIEP (Eq. (3)) and adding the equality constraint (Eq. (6)) into the objective with a hyperparameter $\lambda_c$, which gives the objective as follows:

$$\begin{aligned} &\mathscr{L}_t^*(\beta_t) = \max_{\beta_t} \frac{1}{N}\sum_{j=1}^{N} \log r_{s_t}(x_{t-1,j}) + \lambda_c \left( \frac{\Psi_t(x_t)}{\Phi_t(x_t)\Psi_{t-1}(x_{t-1})} - 1 \right)^2, \\ &\Phi_t(x_t) \triangleq \frac{1}{N}\sum_{i=1}^{N} \exp\{\phi_{\beta_t}(x_{t,i})\}, \;\; \Psi_t(x_t) \triangleq \frac{1}{N}\sum_{i=1}^{N} \exp\{\psi_{\beta_t}(x_{t,i})\} \\ &\text{where } \; t > 1, \;\; x_t \sim q_t(x), \;\; x_{t-1} \sim q_{t-1}(x), \end{aligned} \tag{8}$$

Here $\beta_{t-1}$ is the estimated parameter of $r_{t-1}(x)$ and hence a constant in the objective. We provide theoretical analysis of asymptotic normality of this objective in Appx. A.1.

## 3 Related Work

There are existing methods for detecting changing points online by direct DRE [2, 3, 1], which estimate density ratios between distributions of two consecutive time intervals. These prior work were proposed for detecting abrupt changes in an online data stream. In contrast, CDRE estimates density ratios between distributions of the initial and latest time intervals without storing historical samples. Hence, CDRE is more suitable for cases in which the difference between two consecutive intervals is subtle but the accumulated difference is notable.

A concurrent work [7] has developed Telescoping Density Ratio Estimation (TRE) by a consecutive decomposition that is similar with our method (Eq. (1)) but with the following main differences:

*1*) TRE simultaneously optimizes $m$ ratio estimators as below:

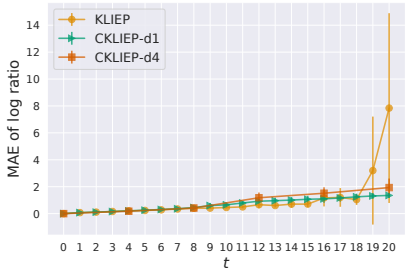
$$\mathscr{L}_{TRE}(r) = \frac{1}{m}\sum_{k=1}^{m} \mathscr{L}(r_k), \;\; r = \frac{p_0}{p_m} = \frac{p_0}{p_1}\frac{p_1}{p_2}\cdots\frac{p_{m-1}}{p_m}, \;\; r_k = \frac{p_{k-1}}{p_k}, \;\; k \in \{1,\ldots,m\}$$

where $m$ is the number of decomposed ratios of the target ratio ($p_0/p_m$). The intermediate distributions ($\{p_1,\ldots,p_{m-1}\}$) are designed by gradually changing from $p_0$ to $p_m$. $\mathscr{L}(r_k)$ is the loss function of estimating the $k$-th ratio $r_k = p_k/p_{k+1}$. Estimating the intermediate ratios would be easier than directly estimating $p_0/p_m$ because the two adjacent distributions are more similar to each other. TRE attempts to estimate the $m$ ratios at the same time. In comparison, CDRE estimates $p_0/p_m$ in an iterative fashion and at each time step ($\forall m \geq 1$) CDRE only optimizes the latest ratio estimator $r_m = p_{m-1}/p_m$ ;

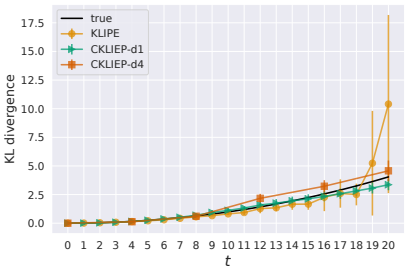
*2*) According to the optimization objective, TRE requires samples of all intermediate distributions as well as the reference distribution. Applying TRE to online applications would need to retain $m$ estimators and $mN$ samples due to the nature of its algorithm, where $m$ increases over time. In contrast, CDRE only requires samples of the two latest distributions, which leads to a constant memory cost at every time step.

*3*) TRE applies the logistic loss that is commonly used in binary classifiers for estimating the ratios, which does not include the constraint of exact density ratio estimation: $\mathbb{E}_{q(x)}[r(x)] = 1$.

**Limitation of CDRE**: In the ideal case that we obtain the true ratio function $r_{s_t}^*$ at a time step $t$, we have $r_t(x) = r_{s_t}^*(x) r_{t-1}(x)$, then $\log r_t^*(x) - \log r_t(x) = \log r_{t-1}^*(x) - \log r_{t-1}(x)$, which means the error inherited from $r_{t-1}(x)$ will be the intrinsic error for estimating $r_t^*(x)$. This intrinsic error from




(a) Comparing MAE of log ratios estimated by CKLIEP and KLIEP




(b) Comparing KL-divergence estimated by CKLIEP and KLIEP

Figure 1: Comparing the performance of CKLIEP and KLIEP by synthetic data. (a) & (b) compare the Mean Absolute Error (MAE) of log ratios and estimated KL-divergences. CKLIEP-d1 estimates $p(x)/q_t(x)$ at each time step; CKLIEP-d4 estimates $p(x)/q_t(x)$ at every 4 steps. The true values of KL-divergences are computed by true ratios. The error bar is the standard deviation of 10 runs.

$r_{t-1}(x)$ represents the accumulated estimation error from time step 1 to $t-1$ because the objective of CDRE is iterative. TRE has the same error accumulation problem since it uses the same consecutive decomposition. Due to this reason, we would prefer smaller difference between each intermediate $q_\tau$ and $q_{\tau-1}$ ($\forall 1 < \tau \leq t$) because it leads to a smaller estimation error at each step. We demonstrate this in Figs. 1a and 1b by CKLIEP-d1 and CKLIEP-d4. Moreover, Fig. 3 shows the variance of the estimation may grow rapidly when the difference between the two distributions exceeds a certain value. CDRE could prevent such an issue by ensuring the difference between any intermediate pairs of distributions is relatively small, e.g., when the data distribution changes fast we could set smaller time intervals for collecting samples to give smaller changes at each step. Regarding applying CDRE in static scenarios that are not online, i.e., the goal is estimating the final ratio between $p(x)$ and $q_T(x)$ where $T$ is the index of the last step, it would require more computational time than TRE or standard DRE since the procedure cannot be paralleled. However, the memory cost and computational time would be the same as DRE in the online setting because the ratio between $p(x)$ and $q_t(x)$ at each step $t$ needs to be estimated anyway.

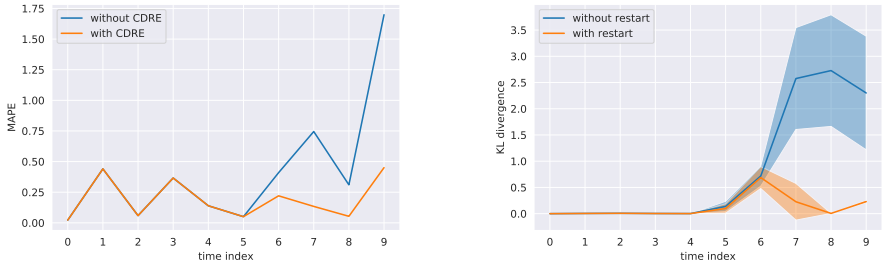
## 4 Experimental Results

In this section we demonstrate the effectiveness of CDRE by several experiments. We apply CKLIEP to instantiate CDRE in our implementation. Due to the limitation of space, we put details of experimental setting and experiments of backward covariate shift in Appx. A.3.

### 4.1 Measuring distribution shifts via KL-divergence

We first demonstrate that CDRE can provide reliable estimations by comparing the results with true ratios using synthetic data. We can approximate KL-divergence between two distributions by estimating density ratios: $\mathscr{D}_{KL}(p||q) = \mathbb{E}_q\left[-\log\left(\frac{p(x)}{q(x)}\right)\right] \approx \frac{1}{N}\sum_{i=1}^{N} -\log\left(r(x_i)\right)$, where $x_i \sim q(x)$. Thus, we can measure the distribution shifts by approximating the KL-divergence between $p(x)$ and $q_t(x)$. We compare the performance of CKLIEP with KLIEP and true values using synthetic Gaussian data, where KLIEP has access to samples of reference distributions at all time. The sample size of each distribution in these experiments is 50,000.

We first simulate the data distribution by a 64-D Gaussian distribution $p(x) = \mathcal{N}(\mu_0, \sigma_0^2 I)$, where $\mu_0 = 0, \sigma_0 = 1$. At each step, we shift the distribution by a constant change on its mean and variance: $q_t(x) = \mathcal{N}(\mu_t, \sigma_t^2 I), \mu_t = \mu_0 + \Delta\mu * k * t, \sigma_t = \sigma_0 - \Delta\sigma * k * t, \Delta\mu = \Delta\sigma = 0.02$, $k$ is the number of steps within one estimation time interval. We set the total number of steps to 20. We estimate $p(x)/q_t(x)$ by applying CKLIEP with two different time intervals: (1). CKLIEP-d1 estimates $p(x)/q_t(x)$ at each step, i.e. $k = 1$; (2). CKLIEP-d4 estimates $p(x)/q_t(x)$ at every four steps, i.e. $k = 4$. We compare the Mean Absolute Error (MAE) of log ratios ($\mathscr{L}_{\text{MAE}} = \frac{1}{N}\sum_{n=1}^{N}|\log r^*(x_n) - \log \hat{r}(x_n)|$) estimated by CKLIEP and KLIEP in Fig. 1a. We also compare the estimated KL-divergence with the true value in Fig. 1b. According to Theorem 2 in Appx. A.1, the difference between $q_{t-1}(x)$ and




(a) MAPE of the regression model

(b) Estimated KL divergence

Figure 2: Monitoring stock data by CDRE. (a) shows the MAPE given by the regression model without and with re-training triggered by CDRE. (b) shows KL divergence between the training set of the regression model and samples from the latest time window. The blue line is without restart during the progress of CDRE (the regression model has not been updated), the orange line is with restart (the regression model has been retrained by latest samples when the KL divergence shows a significant increase at time index 6). The shaded area is plotted by the standard deviation of 5 runs.

$q_t(x)$ plays an important role in the estimation convergence, which explains why CKLIEP-d4 gets worse performance than CKLIEP-d1 in later steps since $q_{t-1}(x)$ and $q_t(x)$ are much less similar in the case of CKLIEP-d4. KLIEP can be viewed as a special case of CKLIEP when $q_{t-1}(x) = p(x)$. We can see that in Figs. 1a and 1b KLIEP has become much worse at the last two steps due to the two distributions are too far away from each other and thus causes serious difficulties in its convergence with a fixed sample size.

### 4.2 Monitoring real stock data for a regression model

We demonstrate the effectiveness of CDRE in practice by real stock data. The dataset consists of one-day transactions of the Microsoft stock. It includes the transaction time, price, volume and direction (initiated by selling or buying). We augment each transaction by concatenating it with its five previous transactions (excluding timestamps) and then treat all of them as i.i.d. samples. We draw samples from a two-hour time window of the data and slide the window with a 30-minute step size. We first train a Gaussian process regression model by samples from the initial two-hour time window to predict the price of a future transaction , then we apply CKLIEP to detect the data shift every 30 minutes. When the KL divergence between the training set ($p(x)$) and the data from the latest two-hour window ($q_t(x)$) shows a significant increase (i.e. an abrupt spike appears) and the KL-divergence larger than 0.5 (according to previous experience in 64D-Gaussian), we retrain the regression model by samples from the latest two hours. We also restart the progress of CKLIEP by replacing the reference distribution $p(x)$ by $q_t(x)$ when we retrain the regression model at time $t$. This procedure can also be done by applying other DRE methods rather than KLIEP into the framework of CDRE, such as a discriminator used in GANs. A more sophisticated strategy for deciding the restart step would be required in a more complicated setting.

We use the last 500 transactions of each two-hour window as the test set of each time step, which are excluded from the training set. And the training sample size is 6000 for each distribution. We evaluate the performance of the regression model by Mean Absolute Percentage Error (MAPE) ($\mathscr{L}_{MAPE} = \frac{1}{N}\sum_{n=1}^{N} 100 \times |y_n - \hat{y}_n|/y_n$) and provide the experiment results in Fig. 2. Fig. 2a shows that using CDRE to monitor the training data shift can effectively prevent large degradation of the performance. The restart strategy of CDRE also helps with reducing the estimation variance in latter steps as shown in Fig. 2b because the new reference distribution is much closer to the dynamic distribution after the restart.

## 5 Further Discussions

Although we just provide the analysis of asymptotic normality for CKLIEP, it seems common properties are shared across different DRE methods when applying them in CDRE. It could be possible to have a general theoretical analysis of CDRE and even TRE. We will leave this for future work.

# A Appendix

## A.1 Asymptotic normality of CKLIEP

Define $\hat{\beta}_t$ as the estimated parameter that satisfies:

$$\mathscr{L}_t'(\hat{\beta}_t) \triangleq \nabla_{\beta_t} \mathscr{L}_t(\beta_t)|_{\beta_t=\hat{\beta}_t} = 0 \tag{9}$$

***Assumptions***: Assume $\phi_{\beta_t}(x)$ (Eq. (7)) includes the correct function that there exists $\beta_t^*$ recovers the true ratio over the population:

$$r_{s_t}^*(x) = \frac{q_{t-1}(x)}{q_t(x)} = \frac{\exp\{\phi_{\beta_t^*}(x)\}}{\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t^*}(x)\}]}, \text{ where } \phi_{\beta_t^*}(x) = \psi_{\beta_t^*}(x) - \psi_{\beta_{t-1}}(x), \tag{10}$$

Assume $q_t(x)$ and $q_{t-1}(x)$ are independent, $n_t = n_{t-1} = n$, where $n_t$ is the sample size of $q_t(x)$. Let $S_t$ be the support of $q_t$, we assume $S_{t-1} \subseteq S_t$ in all cases.

***Notations***: $\rightsquigarrow$ and $\xrightarrow{P}$ mean convergence in distribution and convergence in probability, respectively. $o_P(1)$ means convergence to zero in propability.

**Lemma 1.** *Let* $\ell_r'(\beta_t^*) \triangleq \frac{1}{n}\sum_{j=1}^n \nabla_{\beta_t} \log r_{s_t}(x_{t-1,j})|_{\beta_t=\beta_t^*}$, *we have* $\sqrt{n}\ell_r'(\beta_t^*) \rightsquigarrow \mathscr{N}(0,\sigma^2)$, *where*

$$\sigma^2 = Cov_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t^*}(x)] + \frac{Cov_{q_t}[\nabla_{\beta_t}\exp\{\phi_{\beta_t^*}(x)\}]}{\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t^*}(x)\}]^2}$$

*Proof.* Because

$$\nabla_{\beta_t}\log r_{s_t}(x) = \nabla_{\beta_t}\phi_{\beta_t}(x) - \frac{\sum_i^n \nabla_{\beta_t}\exp\{\phi_{\beta_t}(x_{t,i})\}}{\sum_i^n \exp\{\phi_{\beta_t}(x_{t,i})\}} \tag{11}$$

then

$$\sqrt{n}\ell_r'(\beta_t) = \frac{\sqrt{n}}{n}\sum_{j=1}^n \nabla_{\beta_t}\phi_{\beta_t}(x_{t-1,j}) - \sqrt{n}\frac{\frac{1}{n}\sum_i^n \nabla_{\beta_t}\exp\{\phi_{\beta_t}(x_{t,i})\}}{\frac{1}{n}\sum_i^n \exp\{\phi_{\beta_t}(x_{t,i})\}} \tag{12}$$

By the central limit theorem we have:

$$\begin{aligned}
&\frac{1}{n}\sum_{j=1}^n \nabla_{\beta_t}\phi_{\beta_t}(x_{t-1,j}) \rightsquigarrow \mathscr{N}\left(\mathbb{E}_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t}(x)], \frac{Cov_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t}(x)]}{n}\right), \\
&\frac{1}{n}\sum_i^n \nabla_{\beta_t}\exp\{\phi_{\beta_t}(x_{t,i})\} \rightsquigarrow \mathscr{N}\left(\mathbb{E}_{q_t}[\nabla_{\beta_t}\exp\{\phi_{\beta_t}(x)\}], \frac{Cov_{q_t}[\nabla_{\beta_t}\exp\{\phi_{\beta_t}(x)\}]}{n}]\right),
\end{aligned} \tag{13}$$

and by the weak law of large numbers:

$$\frac{1}{n}\sum_i^n \exp\{\phi_{\beta_t}(x_{t,i})\} \xrightarrow{P} \mathbb{E}_{q_t}[\exp\{\phi_{\beta_t}(x)\}] \tag{14}$$

Because $q_t(x)$ and $q_{t-1}(x)$ are assumed independent, combine the above results we get:

$$\begin{aligned}
&\sqrt{n}\ell_r'(\beta_t^*) \rightsquigarrow \mathscr{N}(\mu,\sigma^2), \\
&\mu = \sqrt{n}\left(\mathbb{E}_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t^*}(x)] - \frac{\mathbb{E}_{q_t}[\nabla_{\beta_t}\exp\{\phi_{\beta_t^*}(x)\}]}{\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t^*}(x)\}]}\right), \\
&\sigma^2 = Cov_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t^*}(x)] + \frac{Cov_{q_t}(\nabla_{\beta_t}\exp\{\phi_{\beta_t^*}(x)\})}{\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t^*}(x)\}]^2}
\end{aligned} \tag{15}$$

Taking derivatives from both sides of $1 = \int r_{s_t}^*(x)q_t(x)dx$:

$$\begin{aligned}
0 = \nabla_{\beta_t}\mathbb{E}_{q_t}[r_{s_t}^*(x)] &= \int \nabla_{\beta_t} r_{s_t}^*(x)q_t(x)dx = \int \frac{\nabla_{\beta_t} r_{s_t}^*(x)}{r_{s_t}^*(x)} r_{s_t}^*(x)q_t(x)dx \\
&= \int \nabla_{\beta_t}\log r_{s_t}^*(x)q_{t-1}(x)dx = \mathbb{E}_{q_{t-1}}[\nabla_{\beta_t}\log r_{s_t}^*(x)] \\
&= \mathbb{E}_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t^*}(x)] - \frac{\mathbb{E}_{q_t}[\nabla_{\beta_t}\exp\{\phi_{\beta_t^*}(x)\}]}{\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t^*}(x)\}]}
\end{aligned} \tag{16}$$

which gives $\mu = 0$. This completes the proof. □

**Lemma 2.** *Let* $\ell_r''(\beta_t^*) \triangleq \frac{1}{n}\sum_{j=1}^n \nabla_{\beta_t}^2 \log r_{s_t}(x_{t-1,j})|_{\beta_t=\beta_t^*}$, *then* $\ell_r''(\beta_t^*) \xrightarrow{P} -I_{\beta_t^*}$, *where* $I_{\beta_t^*} \triangleq Cov_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t^*}(x)]$.

*Proof.* According to Eq. (11)

$$(\nabla_{\beta_t}\log r_{s_t}(x))^2 = \left(\frac{\sum_i^n \nabla_{\beta_t}\exp\{\phi_{\beta_t}(x_{t,i})\}}{\sum_i^n \exp\{\phi_{\beta_t}(x_{t,i})\}}\right)^2 - \quad 2\nabla_{\beta_t}\phi_{\beta_t}(x)\frac{\sum_i^n \nabla_{\beta_t}\exp\{\phi_{\beta_t}(x_{t,i})\}}{\sum_i^n \exp\{\phi_{\beta_t}(x_{t,i})\}} + (\nabla_{\beta_t}\phi_{\beta_t}(x))^2 \tag{17}$$

By the law of large numbers,

$$\begin{aligned}\frac{1}{n}\sum_{j=1}^n (\nabla_{\beta_t}\log r_{s_t}(x_{t-1,j}))^2 \xrightarrow{P} &\left(\frac{\mathbb{E}_{q_t}[\nabla_{\beta_t}\exp\{\phi_{\beta_t}(x_{t,i})\}]}{\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t}(x_{t,i})\}]}\right)^2 \\ &- 2\mathbb{E}_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t}(x)]\frac{\mathbb{E}_{q_t}[\nabla_{\beta_t}\exp\{\phi_{\beta_t}(x_{t,i})\}]}{\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t}(x_{t,i})\}]} + \mathbb{E}_{q_{t-1}}[(\nabla_{\beta_t}\phi_{\beta_t}(x))^2]\end{aligned} \tag{18}$$

Substituting Eq. (16) to the right side of the above equation, we can get:

$$\begin{aligned}\frac{1}{n}\sum_{j=1}^n (\nabla_{\beta_t}\log r_{s_t}(x_{t-1,j}))^2|_{\beta_t=\beta_t^*} &\xrightarrow{P} \mathbb{E}_{q_{t-1}}[(\nabla_{\beta_t}\phi_{\beta_t^*}(x))^2] - \mathbb{E}_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t^*}(x)]^2 \\ &= Cov_{q_{t-1}}[\nabla_{\beta_t}\phi_{\beta_t^*}(x)] = I_{\beta_t^*}\end{aligned} \tag{19}$$

Because

$$\nabla_{\beta_t}^2 \log r_{s_t}(x) = \frac{\nabla_{\beta_t}^2 r_{s_t}(x)}{r_{s_t}(x)} - (\nabla_{\beta_t}\log r_{s_t}(x))^2, \tag{20}$$

then according to Eq. (19)

$$\ell_r''(\beta_t^*) \xrightarrow{P} \mathbb{E}_{q_{t-1}}\left[\frac{\nabla_{\beta_t}^2 r_{s_t}^*(x)}{r_{s_t}^*(x)}\right] - I_{\beta_t^*} = \int \nabla_{\beta_t}^2 r_{s_t}^*(x) q_t(x)dx - I_{\beta_t^*} \tag{21}$$

Under mild assumptions we can interchange the integral and derivative operators:

$$\int \nabla_{\beta_t}^2 r_{s_t}^*(x) q_t(x)dx = \nabla_{\beta_t}^2 \int r_{s_t}^*(x) q_t(x)dx = \nabla_{\beta_t}^2 \int \frac{q_{t-1}(x)}{q_t(x)} q_t(x)dx = 0 \tag{22}$$

which completes the proof. □

**Lemma 3.** *3 Let* $\ell_c(\beta_t) \triangleq \lambda_c\left(\frac{\Psi_t(x_t)}{\Phi_t(x_t)\Psi_{t-1}(x_{t-1})} - 1\right)^2$, *and* $\ell_c'(\beta_t^*) \triangleq \nabla_{\beta_t}\ell_c(\beta_t)|_{\beta_t=\beta_t^*}$, *if we set* $\lambda_c = \frac{A}{\sqrt{n}}$, *where A is a positive constant, then* $\sqrt{n}\ell_c'(\beta_t^*) \xrightarrow{P} 0$.

*Proof.*

$$\sqrt{n_{t-1}}\ell_c'(\beta_t) = 2A\left(\frac{\Psi_t(x_t)}{\Phi_t(x_t)\Psi_{t-1}(x_{t-1})} - 1\right) \times \left(\nabla_{\beta_t}\frac{\Psi_t(x_t)}{\Phi_t(x_t)\Psi_{t-1}(x_{t-1})}\right)$$

By the law of large numbers,

$$\left.\frac{\Psi_t(x_t)}{\Phi_t(x_t)\Psi_{t-1}(x_{t-1})}\right|_{\beta_t=\beta_t^*} \xrightarrow{P} \frac{\mathbb{E}_{q_t}[\exp\{\psi_{\beta_t^*}(x)\}]}{\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t^*}(x)\}]\mathbb{E}_{q_{t-1}}[\exp\{\psi_{\beta_{t-1}}(x)\}]}$$

Define $\tilde{r}_{t-1}(x) \triangleq \frac{\exp\{\psi_{\beta_{t-1}}(x)\}}{\mathbb{E}_{q_{t-1}}[\exp\{\psi_{\beta_{t-1}}(x)\}]}$, by the definition of $r_{s_t}^*(x)$ (Eq. (10)):

$$\int \tilde{r}_{t-1}(x) r_{s_t}^*(x) q_t(x)dx = \int \tilde{r}_{t-1}(x)\frac{q_{t-1}(x)}{q_t(x)} q_t(x)dx = \int \tilde{r}_{t-1}(x) q_{t-1}(x)dx = 1$$

Substituting the right side of Eq. (10) into the left side of the above equation, we can get:

$$\frac{\mathbb{E}_{q_t}[\exp\{\psi_{\beta_t^*}(x)\}]}{\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t^*}(x)\}]\mathbb{E}_{q_{t-1}}[\exp\{\psi_{\beta_{t-1}}(x)\}]} = 1 \tag{23}$$

which completes the proof. □

**Theorem 1.** *Suppose $\lambda_c = \frac{A}{\sqrt{n}}$, where A is a positive constant, assume $\hat{\beta}_t - \beta_t^* = o_p(1)$, let $\ell_{t,x}(\beta_t)$ denote the loss at a point x, assume it satisfies the Lipshitz condition: $\|\ell'_{t,x}(\beta_t^{(1)}) - \ell'_{t,x}(\beta_t^{(2)})\| \leq \|\ell''_{t,x}(\beta_t)\|\|\beta_t^{(1)} - \beta_t^{(2)}\|$, where $\beta_t^{(1)}, \beta_t^{(2)}$ are in a neighborhood of $\beta_t^*$, $\mathbb{E}[\|\ell''_{t,x}(\beta_t)\|^2] < \infty$, then we have:*

$$\begin{aligned} \sqrt{n}(\hat{\beta}_t - \beta_t^*) &\rightsquigarrow \mathcal{N}(0, \nu^2), \\ \nu^2 = I_{\beta_t^*}^{-1} + \mathbb{E}_{q_t}[\exp\{\phi_{\beta_t^*}(x)\}]^{-2} &\times I_{\beta_t^*}^{-1} Cov_{q_t}[\nabla_{\beta_t} \exp\{\phi_{\beta_t^*}(x)\}] I_{\beta_t^*}^{-1} \end{aligned} \tag{24}$$

*Proof.* By combining the results of Lemmas 1 to 3 we can get $\sqrt{n}\mathscr{L}_t'(\beta_t^*) = \sqrt{n}(\ell'_r(\beta_t^*) + \ell'_c(\beta_t^*)) \rightsquigarrow \mathcal{N}(0, \sigma^2)$, $\mathscr{L}_t''(\beta_t^*) = (\ell''_r(\beta_t^*) + \ell''_c(\beta_t^*)) \xrightarrow{P} -I_{\beta_t^*}$. According to Theorem 5.21 in [12] and the results of Lemmas 1 to 3:

$$\begin{aligned} \sqrt{n}(\hat{\beta}_t - \beta_t^*) &\rightsquigarrow \mathcal{N}(0, \nu^2), \\ \nu^2 &= I_{\beta_t^*}^{-1} \sigma^2 I_{\beta_t^*}^{-1}, \\ &= I_{\beta_t^*}^{-1} + \mathbb{E}_{q_t}[\exp\{\phi_{\beta_t^*}(x)\}]^{-2} \times I_{\beta_t^*}^{-1} Cov_{q_t}[\nabla_{\beta_t} \exp\{\phi_{\beta_t^*}(x)\}] I_{\beta_t^*}^{-1}, \end{aligned} \tag{25}$$

□

We make the assumption $\hat{\beta}_t - \beta_t^* = o_p(1)$ here which relies on the parameterization of $r_{s_t}$. We define $r_{s_t}$ by neural networks because they have the flexibility to simulate arbitrary functions. The experimental results show that a simple two-layer MLP can work well for Gaussian distributions. For more complex distributions, a more sophisticated network architecture might be preferred.

**Theorem 2.** *Suppose $p(x)$ and $q_t(x)$ $(\forall t)$ are from the exponential family, define $r_{s_t}^*(x) = \exp\{\phi_{\beta_t^*}(x)\}$, $\phi_{\beta_t^*}(x) = \beta_t^* T(x) + C$, $T(x)$ is a sufficient statistic of x, C is a constant, then $\sqrt{n}(\hat{\beta}_t - \beta_t^*) \rightsquigarrow \mathcal{N}(0, \nu_e^2)$, where $T(x)$ is a column vector, $T(x)^2 = T(x)T(x)^T$, $I_{\beta_t^*} = Cov_{q_{t-1}}[T(x)]$:*

$$\nu_e^2 = I_{\beta_t^*}^{-1} + I_{\beta_t^*}^{-1}(\mathbb{E}_{q_{t-1}}[r_{s_t}^*(x)T(x)^2] - \mathbb{E}_{q_{t-1}}[T(x)]^2) I_{\beta_t^*}^{-1} \tag{26}$$

*Proof.* Because $\phi_t^*(x) \triangleq \psi_{\beta_t^*}(x) - \psi_{\beta_{t-1}^*}(x)$, then $\nabla_{\beta_t} \phi_{\beta_t}^*(x) = T(x)$, we have

$$I_{\beta_t^*} = Cov_{q_{t-1}}[\nabla_{\beta_t} \phi_{\beta_t}^*(x)] = Cov_{q_{t-1}}[T(x)], \tag{27}$$

Because $r_{s_t}^*(x) = \exp\{\phi_t^*(x)\}$,

$$\mathbb{E}_{q_t}[\exp\{\phi_{\beta_t}^*(x)\}] = \mathbb{E}_{q_t}[r_{s_t}^*(x)] = 1,$$

In addition,

$$Cov_{q_t}[\nabla_{\beta_t} \exp\{\phi_{\beta_t}^*(x)\}] = Cov_{q_t}[r_{s_t}^*(x)T(x)] = \mathbb{E}_{q_t}[(r_{s_t}^*(x)T(x))^2] - \mathbb{E}_{q_t}[r_{s_t}^*(x)T(x)]^2 \tag{28}$$

where

$$\begin{aligned} \mathbb{E}_{q_t}[(r_{s_t}^*(x)T(x))^2] &= \int q_t(x)(r_{s_t}^*(x)T(x))^2 dx = \int q_{t-1}(x) r_{s_t}^*(x) T(x)^2 dx = \mathbb{E}_{q_{t-1}}[r_{s_t}^*(x)T(x)^2], \\ \mathbb{E}_{q_t}[r_{s_t}^*(x)T(x)] &= \int q_t(x) r_{s_t}^*(x) T(x) dx = \int q_{t-1}(x) T(x) dx = \mathbb{E}_{q_{t-1}}[T(x)] \end{aligned} \tag{29}$$

Substitute above results into 1, this proves the theorem. □

Theorem 1 shows how the covariance matrix $\nu^2$ depends on the latest density ratio ($r_{s_t}^*$) and Theorem 2 is the specific case when the two distributions are from a same exponential family. In Eq. (26) all terms are solely decided by $q_{t-1}(x)$ except $\mathbb{E}_{q_{t-1}}[r_{s_t}^*(x)T(x)^2]$. Since a smaller variance is better for convergence, we would prefer $r_{s_t}^*(x) = q_{t-1}(x)/q_t(x)$ is not large, which means when $q_{t-1}(x)$ is large $q_t(x)$ should be also large. In this case, $r_{s_t}^*$ is less likely to explode and thus the variance of the estimated parameter would be likely confined. We demonstrate this by experiments with 1-D Gaussian distributions. We fix $q_{t-1}(x) = \mathcal{N}(0,1)$, testing different $q_t(x) = \mathcal{N}(\mu_t, 1)$, where $\mu_t = \delta k, \delta = 0.1, k \in \{0, 1, \ldots, 20\}$. In this case, $T(x) = \{x, x^2\}$, $\beta_t = \{\beta_{t,1}, \beta_{t,2}\}$. When $\mu_t$ is larger, $q_t$ is farther to $q_{t-1}$. We display the diagonal of $\nu_e^2$ (variance of $\beta_{t,1}, \beta_{t,2}$) in Fig. 3. It is clear that the variance of $\beta_t$ is getting larger when $q_t$ is farther to $q_{t-1}$. It indicates that smaller difference between each intermediate $q_\tau$ and $q_{\tau-1}$ $(\forall 1 < \tau \leq t)$ leads to a better estimation. We demonstrate in Sec. 4.1 that it is often the case in practice even when $\psi_{\beta_t^*}(x)$ is approximated by a non-linear model.

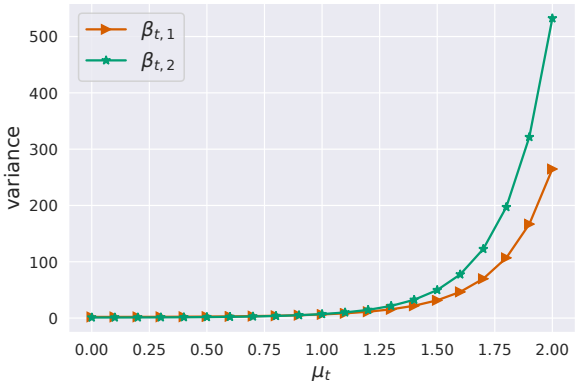



Figure 3: Demonstration of the variance of estimated parameters in Theorem 2 by 1-D Gaussian distributions: fix $q_{t-1}(x) = \mathcal{N}(0,1)$ and vary $q_t(x) = \mathcal{N}(\mu_t, 1)$ by setting $\mu_t = \delta k, \delta = 0.1, k \in \{0,1,\ldots,20\}$. When $\mu_t$ is larger the two distributions are less similar and the variance is larger, which aligns with Theorem 2.

### A.2 Multiple reference distributions in CDRE

We can trace multiple reference distributions by CDRE as well. It avoids building separated estimators for tracing different reference distributions in an application (e.g. seasonal data). It also matches the setting of training generative models in continual learning as the model needs to learn generating samples from a new data distribution at each task whilst it can still generate samples from all seen distributions. In this case a new pair of original and dynamic distributions will be added into the training process of the estimator at some time point. Here we refer to an reference distribution as $p_\tau(x)$, where $\tau$ is the time index of starting tracing the reference distribution. And samples of $p_\tau(x)$ are not available when $t > \tau$. Similarly, $q_{\tau,t}(x)$ denotes the dynamic distribution that corresponding to $p_\tau(x)$ at time $t$, thus $r^*_{\tau,t}(x) = r^*_{s_{\tau,t}}(x) r^*_{\tau,t-1}(x)$, where $r^*_{s_{\tau,t}}(x) = q_{\tau,t-1}(x)/q_{\tau,t}(x)$. In this case, we optimize the estimator at time $t$ by an averaged objective:

$$\max_{\beta_t} \bar{\mathscr{L}}_t(\beta_t) = \max_{\beta_t} \frac{1}{|\mathbb{T}|} \sum_{\tau \in \mathbb{T}} \mathscr{L}_t(\beta_t; \tau) \tag{30}$$

where $\mathbb{T}$ is the set of time indices of adding reference distributions, $|\mathbb{T}|$ is the size of $\mathbb{T}$. $\mathscr{L}_t(\beta_t;\tau)$ is as the same as the loss function of a single reference distribution ( Eq. (8)) for a given $\tau$. Further, $r_{s_{\tau,t}}(x)$ is also defined by the same form of Eq. (7), the difference is that $\psi_{\beta_t}(x)$ becomes $\psi_{\beta_t}(x;\tau)$:

$$r_{s_{\tau,t}} = \frac{\exp\{\phi_{\beta_t}(x;\tau)\}}{\frac{1}{N}\sum_{i=1}^N \exp\{\phi_{\beta_t}(x_i;\tau)\}}, \quad \text{where} \quad \phi_{\beta_t}(x;\tau) \triangleq \psi_{\beta_t}(x;\tau) - \psi_{\beta_{t-1}}(x;\tau). \tag{31}$$

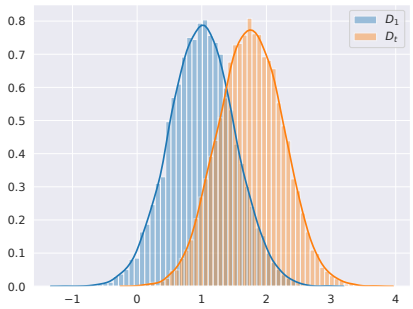
In our implementation, we concatenate the time index $\tau$ to each data sample as the input of the ratio estimator. In addition, we set the output of $\psi_{\beta_t}(\cdot)$ as a $|\mathbb{T}|$-dimensional vector $\{o_1,\ldots,o_i,\ldots,o_{|\mathbb{T}|}\}$ where $o_i$ corresponds to the output of $\psi_{\beta_t}(x;\tau = \mathbb{T}_i)$. Thus, we can avoid learning separate ratio estimators for multiple reference distributions. Note that with CDRE we have the flexibility to extend the model architecture since the latest estimator function $\phi_{\beta_t}$ only needs the output of the previous estimator function $\psi_{\beta_{t-1}}$. This can be beneficial when the model capacity becomes a bottleneck of the performance.

### A.3 Experiments

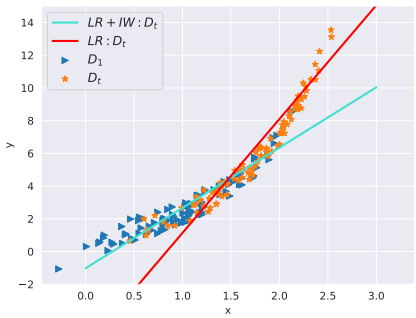
We provide details of our experimental settings and more experimental results in this section.

#### A.3.1 Experimental Settings

In all of our experiments, $\psi(\cdot)$ is a neural network with two and three dense layers for a single and multiple reference distributions respectively, each layer having 256 hidden units and ReLU activations. $\lambda_c$ (the hyperparameter used for controlling the strength of the constraint in Eq. (8)) is set to 10 for




(a) the data distribution at $\tau = 1$ ($D_1$) and $\tau = t$ ($D_t$)




(b) linear regression trained by $D_t$ for predicting test samples from $D_1$

Figure 4: Demo experiment of backward covariate shift. (a) shows the data distribution of training set at $\tau = 1$ and $\tau = t$. (b) displays the regression lines learnt by the model at $\tau = t$, the cyan and red lines are fitted by $D_t$ with and without importance weights, respectively.

experiments with a single reference distribution, and set to $100k$ for experiments with the multiple reference distributions, where $k$ is the number of joined reference distributions at each time step.

The real stock data used in our experiments can be downloaded from `https://lobsterdata.com/info/DataSamples.php`, which is sample data for free. We only used the one level data.

### A.3.2 Backward covariate shift

We demonstrate that CDRE can be applied in *backward covariate shift*, where the training set shifts and the test set is from a previous distribution. It just swaps the situation of training and test set in the scenario of common covariate shift [11, 9, 6]. We assume a linear regression model defined as $\hat{y} = wx + b + \varepsilon_0$, where the noise $\varepsilon_0 \sim \mathcal{N}(0, 0.01)$. At time $\tau$, the training data $x \in D_\tau$ and $D_\tau$ shifts away from $D_1$ gradually, where $\tau \in \{1, 2, \ldots, t\}$ and $t = 10$ is the latest time index. Fig. 4a displays the data distribution at time $\tau = 1$ and $\tau = t$ in which we can see there exists notable difference between the two distributions. When the model is trained by $D_t$, it will not be able to accurately predict on test samples from $D_1$ unless we adjust the loss function by importance weights (i.e. density ratios) as in handling covariate shift:

$$\mathscr{L} = \mathbb{E}_{x \sim q_t(x)} \left[ \frac{q_1(x)}{q_t(x)} (y - \hat{y})^2 \right]$$

Fig. 4b shows the regression lines learned by the model at $\tau = t$ with and without the importance weights, where the weights $q_1(x)/q_t(x)$ are estimated by CKLIEP. We can see that the line learned with importance weights fits $D_1$ more accurately than the one without the weights. This enables the model to make reasonable predictions on test samples from $D_1$ when the training set drifts to $D_t$ and $D_1$ is not available. We demonstrate that CDRE can be applied in *backward covariate shift*, where the training set shifts and the test set is from a previous distribution. It just swaps the situation of training and test set in the scenario of common covariate shift [11, 9, 6]. We assume a linear regression model defined as $\hat{y} = wx + b + \varepsilon_0$, where the noise $\varepsilon_0 \sim \mathcal{N}(0, 0.01)$. At time $\tau$, the training data $x \in D_\tau$ and $D_\tau$ shifts away from $D_1$ gradually, where $\tau \in \{1, 2, \ldots, t\}$ and $t = 10$ is the latest time index. Fig. 4a displays the data distribution at time $\tau = 1$ and $\tau = t$ in which we can see there exists notable difference between the two distributions. When the model is trained by $D_t$, it will not be able to accurately predict on test samples from $D_1$ unless we adjust the loss function by importance weights (i.e. density ratios) as in handling covariate shift:

$$\mathscr{L} = \mathbb{E}_{x \sim q_t(x)} \left[ \frac{q_1(x)}{q_t(x)} (y - \hat{y})^2 \right]$$

Fig. 4b shows the regression lines learned by the model at $\tau = t$ with and without the importance weights, where the weights $q_1(x)/q_t(x)$ are estimated by CKLIEP. We can see that the line learned with importance weights fits $D_1$ more accurately than the one without the weights. This enables the model to make reasonable predictions on test samples from $D_1$ when the training set drifts to $D_t$ and $D_1$ is not available.

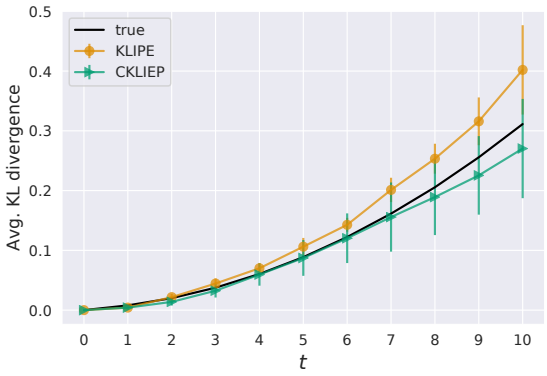



Figure 5: Comparing average KL-divergence estimated by CKLIEP and KLIEP in the scenario of multiple reference distributions. The true values of KL-divergences are computed by true ratios. The error bar is the standard deviation of 10 runs.

### A.3.3 Tracing multiple reference distributions

Here we provide the experimental results of multiple reference distributions. We also simulate the scenario of multiple reference distributions by 64-D Gaussian data: $p_\tau(x) = \mathcal{N}(\mu_\tau, \sigma_\tau^2 I)$, $\tau \in \{1,2,\ldots,t\}, \mu_\tau = 2\tau, \sigma_\tau = 1$, in which cases we add a new reference distribution $(p_t(x))$ at each time step. We shift each joined reference distribution $(p_\tau(x), \forall \tau < t)$ by a constant change as the same as the single pair scenario and set $k = 1, \Delta\mu = \Delta\sigma = 0.01$. In Appx. A.3.3, we compare the averaged KL-divergences $(\bar{\mathscr{D}} = \frac{1}{t}\sum_{\tau=1}^{t} \mathscr{D}_{KL}(q_{\tau,t}||p_\tau))$ estimated by CKLIEP and KLIEP with the true value. CKLIEP outperforms KLIEP when the dynamic distributions getting farther away from the reference distributions, which aligns with the scenario of a single reference distribution.

## References


[1] I. Bouchikhi, A. Ferrari, C. Richard, A. Bourrier, and M. Bernot. Non-parametric online change-point detection with kernel lms by relative density ratio estimation. In *2018 IEEE Statistical Signal Processing Workshop (SSP)*, pages 538–542. IEEE, 2018.

[2] Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 389–400. SIAM, 2009.

[3] S. Liu, M. Yamada, N. Collier, and M. Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 43:72–83, 2013.

[4] D. McAllester and K. Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884, 2020.

[5] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

[6] J. Quiñonero-Candela, M. Sugiyama, N. D. Lawrence, and A. Schwaighofer. *Dataset shift in machine learning*. Mit Press, 2009.

[7] B. Rhodes, K. Xu, and M. U. Gutmann. Telescoping density-ratio estimation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4905–4916. Curran Associates, Inc., 2020.

[8] S. Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[9] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[10] M. Sugiyama, T. Suzuki, and T. Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[11] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[12] A. W. Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.