Theory and Algorithm for Batch Distribution Drift Problems

Pranjal Awasthi
Google Research
New York, NY 10011
pranjalawasthi@google.com

Corinna Cortes
Google Research
New York, NY 10011
corinna@google.com

Christopher Mohri
Cornell University
Ithaca, NY 14853
ccm244@cornell.edu

Abstract

We study a problem of *batch distribution drift* motivated by several applications, which consists of determining an accurate predictor for a target time segment, for which a moderate amount of labeled samples are at one's disposal, while leveraging past segments for which substantially more labeled samples are available. We give new algorithms for this problem guided by a new theoretical analysis and generalization bounds derived for this scenario. We further extend our results to the case where few or no labeled data is available for the period of interest. Finally, we report the results of extensive experiments demonstrating the benefits of our drifting algorithm, including comparisons with natural baselines. A by-product of our study is a principled solution to the problem of multiple-source adaptation with labeled source data and a moderate amount of target labeled data, which we briefly discuss and compare with.

1 Introduction

The standard assumption in learning theory and algorithm design is that training and test distributions coincide and that the distributions are fixed over time. However, this assumption does not always hold in practice. In many applications, the learning environment is non-stationary and subject to a continuous drift over time. These include tasks such as political sentiment analysis, news stories, spam detection, financial market prediction under mildly fluctuating economic conditions, fraud detection, network intrusion detection, sales prediction, and many others.

In such tasks, the distribution changes over time gradually. For example, sales or fraud patterns are relatively stable within a time segment, which may be a month or two long, but they may change at the subsequent period. We are interested here in the study of prediction in such gradual distribution drift scenarios, which are distinct from and more favorable than the most general scenarios of time series prediction where more drastic changes of the distributions may occur (Engle, 1982; Bollerslev, 1986; Brockwell and Davis, 1986; Box and Jenkins, 1990; Hamilton, 1994; Meir, 2000; Kuznetsov and Mohri, 2015).

The problem of predicting in a distribution drift setting, also known as the *concept drift* problem, is more challenging, however, than learning in the standard i.i.d. environments. In general, concept drift is defined by a change in the joint distribution over the input points and their labels (Gama et al., 2014). The problem has been studied both in the on-line and batch learning settings. This paper deals with the batch setting. For a discussion of related work in the online setting, see Appendix A.

For offline or batch learning, Helmbold and Long (1994) provided learning bounds in the case where only the target was allowed to drift. Bartlett (1992) presented an analysis for a drifting of the joint distribution based on the total variation as the distance between distributions, and Barve and Long (1997) gave a tight bound for this scenario. Under a persistent or even rapid rate of change assumption, Freund and Mansour (1997) improved these theoretical learning results. However, such studies for the batch learning make a rather strong assumption about the rate of drift, which implies that training only on the most recent examples is sufficient for a certain period of time. This approach therefore does not benefit from all *older* examples that are at the learner's disposal. The results just discussed are also all based on the ℓ_1-distance as a measure of divergence between two consecutive distributions. As argued by Mohri and Muñoz Medina (2012), tighter learning bounds can be achieved using a notion of *discrepancy*, which can be viewed as a more suitable divergence measure since it takes into account both the loss function and the hypothesis set. Concept drift has also been studied in both the online and offline setting for clustering, where labels are not available (Moulton et al., 2018).

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

This paper deals with the particular batch scenario of distribution drift that often appears in applications. In that scenario, distribution time segments are known to the learner and one can thus expect to receive i.i.d. data from the same distribution within each period. The task consists of making use of the data from the previous time segments to make accurate predictions for a new segment for which there can be a moderate amount of labeled data. This could for example correspond to the first few days of a month-long time segment. We will also consider an alternative weakly supervised drifting problem where few or no labeled data is available from the new period. This notion of a predefined structural component of a time series is common in statistical analysis like Kalman filters for time series decomposition (Harvey, 1990; Durbin and Koopman, 2002; Campagnoli et al., 2009) and the BSTS technique (Scott and Varian, 2014). Note, however, that in our formulation we do not require the segments to be of a fixed length.

Much of the recent literature on drifting has been related to *drift detection* and subsequent *model adaptation*, as also detailed in Appendix A. This paper deals with the model adaptation problem in drifting. Thus, our assumption about the distribution time segments being available holds either when the task admits some clearly defined segments, or when a prior drift detection technique has been used to determine the segments. In Section 4, we provide an algorithm based on discrepancy for automatically detecting time segment boundaries, should they not have been provided a priori.

Unlike some of the past literature on drifting, our analysis makes no direct assumption about the δ-closeness of consecutive distributions. Instead, we make use of estimates of the discrepancy between the distribution at each time segment and that of the target segment. Many algorithms dealing with this scenario consist of a fixed reweighting of the labeled examples in the past time segments. We give a new and general analysis of generalization for reweighting in the drifting setting. Next, we use that theory to guide the design of a new algorithm, DRIFT, as well as a simpler version, SDRIFT.

Our theoretical analysis and algorithm are distinct from those of Mohri and Muñoz Medina (2012), which also make use of the discrepancy. Our discrepancy-based generalization bounds for reweighted samples in the drifting scenario are novel and very general: they hold for arbitrary hypothesis sets and are expressed in terms of their weighted Rademacher complexity. In contrast, the guarantees provided by that prior work are specific to an online-to-batch solution. Our algorithm crucially learns *simultaneously* the weights and the hypothesis, while theirs relies on an online learning algorithm to generate hypotheses in a first stage and then determines weights in the second stage to form an average of the hypotheses. The adversarial online algorithm used can be too conservative and return predictors in the first stage that are too weak to combine into a strong solution in the second stage. This is what we observe in our empirical analysis, where we also find that our algorithm outperforms theirs in all the tasks considered (Section 5).

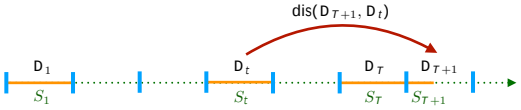

Figure 1: Illustration of the Learning Scenario: distributions D_t, samples $S_t \sim \mathsf{D}_t^{m_t}$, and discrepancies $\mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)$, where $|S_t| = m_t$ and $\sum_{s=1}^{T+1} m_s = m$.

In the next section (Section 2), we give a formal description of the batch distribution drift scenario we consider, along with several relevant definitions of *discrepancy*. In Section 3, we prove generalization bounds for learning with weighted labeled samples in this drifting scenario, including guarantees that hold uniformly over the choice of the weights. We discuss the insights gained from these learning bounds and use them to directly design new algorithms, DRIFT and SDRIFT, in Section 4. In Appendix D, we further extend our theoretical and algorithmic results to the scenario of weakly supervised and unsupervised drifting where few or no labeled data is available from the target period.

A by-product of our study is a principled solution to the problem of multiple-source adaptation with labeled source data and a moderate amount of target labeled data, which can be viewed as a special instance of our drifting problem where the data is not sequential. In Section 4.6, we briefly discuss that scenario and compare it with those studied in previous work. Finally, we report the results of extensive experiments with our algorithm on synthetic as well as real-world data and comparisons with several baselines in Section 5.

2 Learning Scenario

Let X denote the input space, Y the output space, and H a hypothesis set of functions mapping from X to Y. We will consider a loss function $\ell\colon \mathsf{Y}\times\mathsf{Y}\to\mathsf{R}$ assumed to take values in $[0,1]$. For any distribution P over $\mathsf{X}\times\mathsf{Y}$, we denote by $\mathcal{L}(\mathsf{P},h)$ the expected loss of $h\in\mathsf{H}$ for the distribution P: $\mathcal{L}(\mathsf{P},h) = \mathsf{E}_{(x,y)\sim\mathsf{P}}[\ell(h(x),y)]$.

We study the following *distribution drift* problem. Let $\mathsf{D}_1,\ldots,\mathsf{D}_{T+1}$ be $(T+1)$ distributions over $\mathsf{X}\times\mathsf{Y}$. The learner receives a labeled i.i.d. sample $S_t = ((x_{n_t+1}, y_{n_t+1}),\ldots,(x_{n_t+m_t}, y_{n_t+m_t}))$ of size m_t from each distribution D_t, $t\in[T+1]$, with $n_t = \sum_{s=1}^{t-1} m_s$, see Figure 1. The notation $t\in[T+1]$ refers to $t\in\{1,2,\ldots,T+1\}$. We will also use the shorthand $m = n_{T+2} = \sum_{t=1}^{T+1} m_t$ for the total sample size. We will be particularly interested in cases where m_{T+1} is significantly smaller than the total sample encountered in the first T segments, that is $m_{T+1} \ll \sum_{t=1}^{T} m_t$. For any t, will denote by $\widehat{\mathsf{D}}_t$ the empirical distribution defined by the sample S_t and will denote by $\mathsf{D}_{t;X}$ the marginal

distribution of D_t on X. The goal is to use these samples to learn a hypothesis h for the target distribution D_{T+1} with small expected loss $\mathcal{L}(\mathsf{D}_{T+1}, h)$.

A key challenge in this problem is that, in general, the source distributions D_t, $t \in [T]$ do not coincide with the target distribution D_{T+1}. Of course, one could use just the sample S_{T+1} available from the target to train a predictor. However, when the distributions D_t, $t \in [T]$, are somewhat similar to the target distribution, using the samples S_t, $t \in [T]$, may help select a more accurate predictor.

An appropriate divergence measure between distributions is needed to analyze the distribution drifting problem. Mohri and Muñoz Medina (2012) argued that a suitable measure in the context of drifting is that of Y-*discrepancy*, which we will refer to as *labeled discrepancy*. The unlabeled counterpart of the notion of discrepancy was introduced by Mansour, Mohri, and Rostamizadeh (2009a) and shown to be tailored to the analysis of domain adaptation (Kifer et al., 2004; Ben-David et al., 2006; Mansour et al., 2009a; Cortes and Mohri, 2014; Cortes et al., 2019b). Discrepancy takes into account both the loss function and the hypothesis set. Moreover, it can be estimated from a finite sample (Mansour et al., 2021). Unlabeled discrepancy also coincides with the so-called d_A-*distance* coined by Kifer et al. (2004), in the particular case where the zero-one loss is used.

The *labeled discrepancy* between D_i and D_j, $\mathsf{dis}(\mathsf{D}_i, \mathsf{D}_j)$ (Mohri and Muñoz Medina, 2012; Cortes et al., 2019b), is defined as follows:

$$\mathsf{dis}(\mathsf{D}_i, \mathsf{D}_j) = \sup_{h \in \mathsf{H}} \mathop{\mathsf{E}}_{(x,y)\sim \mathsf{D}_i}[\ell(h(x), y)] - \mathop{\mathsf{E}}_{(x,y)\sim \mathsf{D}_j}[\ell(h(x), y)].$$

Observe that we do not take the absolute value of the difference between the expectations as in the initial definitions of discrepancy. This simpler definition is adequate for much of our results. We define the discrepancy with absolute values as: $\mathsf{Dis}(\mathsf{D}_i, \mathsf{D}_j) = \max\{\mathsf{dis}(\mathsf{D}_i, \mathsf{D}_j), \mathsf{dis}(\mathsf{D}_j, \mathsf{D}_i)\}$.

Labeled discrepancy can be upper bounded by the ℓ_1-distance when the loss function admits an upper bound B (see also (Mansour et al., 2009a)):

$$\begin{aligned}\mathsf{dis}(\mathsf{D}_i, \mathsf{D}_j) &= \sup_{h \in \mathsf{H}} \sum_{(x,y)} [\mathsf{D}_i(x,y) - \mathsf{D}_j(x,y)]\, \ell(h(x), y) \\ &\le \sup_{h \in \mathsf{H}} \sum_{(x,y)} |\mathsf{D}_i(x,y) - \mathsf{D}_j(x,y)|\, |\ell(h(x), y)| \\ &\le B \sum_{(x,y)} |\mathsf{D}_i(x,y) - \mathsf{D}_j(x,y)| \\ &= B\, \ell_1(\mathsf{D}_i, \mathsf{D}_j).\end{aligned}$$

By Pinsker's inequality, it can thus also be upper bounded in terms of the relative entropy. However, these divergence measures do not take into account the hypothesis set and the loss function and in general cannot be accurately estimated from finite samples. Since it takes into account the loss function, labeled discrepancy is also a finer measure, more relevant to the task at hand than the $\mathcal{H}$-divergence adopted for time series modeling in (Ganin and Lempitsky, 2015; Sicilia et al., 2021; Lu et al., 2022).

In all the definitions above, we also allow D_i and D_j to be finite signed measures over $\mathsf{X} \times \mathsf{Y}$, thus the weights may not sum to one. In addition, we (abusively) allow distributions over sample indices. For example, given a sample S and a distribution q over its $[m]$ indices, we write

$$\mathsf{dis}(\widehat{\mathsf{D}}, \mathsf{q}) = \sup_{h \in \mathsf{H}} \frac{1}{m} \sum_{i=1}^{m} \ell(h(x_i), y_i) - \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i).$$

3 Generalization Bounds for Batch Drifting Scenarios

In this section, we give new generalization bounds for the distribution drift problem, using the notion of discrepancy. We will denote by S the full sample $S = (S_1, \ldots, S_{T+1})$ of size m. For a non-negative vector q in $[0,1]^m$, we denote by $\overline{\mathsf{q}}_t$ the total *weight* on the points in sample S_t, $t \in [T{+}1]$: $\overline{\mathsf{q}} = \sum_{i=1}^{m_t} \mathsf{q}_{n_t+i}$ and by $\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H})$ the q-weighted Rademacher complexity, an extension of Rademacher complexity taking into account the weights q:

$$\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) = \mathop{\mathsf{E}}_{S,\sigma}\left[\sup_{h \in \mathsf{H}} \sum_{i=1}^{m} \sigma_i \mathsf{q}_i \ell(h(x_i), y_i)\right], \tag{1}$$

where σ_is are independent and uniform random variables taking values in $\{-1, +1\}$.

We first present a learning guarantee for batch drifting for fixed values of the weights q, expressed in terms of the discrepancy between D_{T+1} and a weighted sum of all segment distributions D_t.

Theorem 1. *Fix a vector q in $[0,1]^m$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S drawn from $\mathsf{D}_1^{m_1} \otimes \cdots \otimes \mathsf{D}_{T+1}^{m_{T+1}}$, for all $h \in \mathsf{H}$:*

$$\mathcal{L}(\mathsf{D}_{T+1}, h) \le \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\right) + 2\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) + \|\mathsf{q}\|_2 \sqrt{\frac{\log \frac{1}{\delta}}{2}}.$$

Furthermore, when q is a distribution, $\|\mathsf{q}\|_1 = 1$, the discrepancy term can be replaced by $\sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)$.

The simplification of the second term when q is a distribution stems from $\mathsf{dis}\big((1 - \overline{\mathsf{q}}_{T+1})\mathsf{D}_{T+1}, \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathsf{D}_t\big) = \mathsf{dis}\big(\sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathsf{D}_{T+1}, \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathsf{D}_t\big) = \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)$. The full proof is given in Appendix B, where we also prove that the result is tight in terms of the discrepancy term (Theorem 3). The following theorem further extends this result to a bound that can be used to choose both $h \in \mathsf{H}$ and q. For this result, we consider a reference distribution p^0, which

can be thought of as a reasonable first estimate for q. A natural choice is the uniform distribution over just the target points. We then derive a bound that holds uniformly for all q in $\{\mathsf{q}: 0 < \|\mathsf{q} - \mathsf{p}^0\|_1 < 1\}$.

Theorem 2. *For any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S drawn from $\mathsf{D}_1^{m_1} \otimes \cdots \otimes \mathsf{D}_{T+1}^{m_{T+1}}$, the following holds for all $h \in \mathsf{H}$ and $\mathsf{q} \in \{\mathsf{q}: 0 \leq \|\mathsf{q} - \mathsf{p}^0\|_1 < 1\}$:*

$$\mathcal{L}(\mathsf{D}_{T+1}, h) \leq \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\right) + \mathsf{dis}(\mathsf{q}, \mathsf{p}^0) + 2\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) + 5\|\mathsf{q} - \mathsf{p}^0\|_1 + \left[\|\mathsf{q}\|_2 + 2\|\mathsf{q} - \mathsf{p}^0\|_1\right]\left[\sqrt{\log \log_2 \frac{2}{1 - \|\mathsf{q} - \mathsf{p}^0\|_1}} + \sqrt{\frac{\log \frac{2}{\delta}}{2}}\right].$$

This proof is also given in Appendix B.

Analysis of the bounds. Theorem 1 provides a guarantee for the expected loss of any predictor h in H based on its q-weighted sample loss, the labeled discrepancy, the q-weighted Rademacher complexity, and $\|\mathsf{q}\|_2$. When q is a distribution, the discrepancy term reduces to $\sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)$ and the bound suggests allocating less total weight, $\overline{\mathsf{q}}_t$, to a segment t with a larger target discrepancy, $\mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)$. Equivalently, the first two terms of the bound can then be combined as $\sum_{t=1}^{T+1} \sum_{i=n_t}^{n_t+m_t} \mathsf{q}_i[\ell(h(x_i), y_i) + \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)]$, which can be interpreted as the loss on each sample point being augmented with the discrepancy term corresponding to its segment. There is also a natural balance between the q-weighted empirical loss and $\|\mathsf{q}\|_2$ terms: the learning guarantee prescribes minimizing the former, but not at the expense of assigning most points a weight of zero which results in a large value for $\|\mathsf{q}\|_2$. Note that the last term suggests an interpretation of $1/\|\mathsf{q}\|_2^2$ as the *effective sample size*, since in standard learning bounds, ignoring constants, the term appearing in lieu of $\|\mathsf{q}\|_2$ is $1/\sqrt{|\text{sample}|}$. The bound of Theorem 2 additionally includes the terms $\|\mathsf{q} - \mathsf{p}^0\|_1$ and $\mathsf{dis}(\mathsf{q}, \mathsf{p}^0)$, which both recommend choosing q not too far from the reference p^0. The global insight suggested by these learning bounds is that a balance of all these terms is important for generalization to be successful in drifting. In the next section we describe an algorithm based on these observations.

4 Drifting Distributions Algorithms

We present new learning algorithms that leverage the theoretical foundations above.

4.1 DRIFT Algorithm

Theorem 2 suggests minimizing the right-hand side of the inequality with an ideal choice of $h \in \mathsf{H}$ and $\mathsf{q} \in [0, 1]^m$. If we assume that H is a subset of a normed vector space and that the Rademacher complexity term can be upper-bounded on the norm squared $\|h\|^2$, the optimization problem with λ_1, λ_2 and λ_∞ as non-negative hyperparameters is as follows:

$$\min_{h \in \mathsf{H}; \mathsf{q} \in [0;1]^m} \sum_{i=1}^{m} \mathsf{q}_i[\ell(h(x_i), y_i)] + \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t) + \mathsf{dis}(\mathsf{q}, \mathsf{p}^0) + \lambda_\infty \|\mathsf{q}\|_\infty \|h\|^2 + \lambda_1 \|\mathsf{q} - \mathsf{p}^0\|_1 + \lambda_2 \|\mathsf{q}\|_2^2,$$

where we used a tight upper bound on the weighted Rademacher complexity given by Lemma 1, see Appendix B. We must still choose the reference p^0. A natural choice is the uniform distribution over just S_{T+1}, the empirical distribution without any points from previous distributions. We call DRIFT the algorithm seeking to solve this optimization problem. We also introduce a simpler algorithm SDRIFT, where we upper-bound the $\mathsf{dis}(\mathsf{q}, \mathsf{p}^0)$ term by $\|\mathsf{q} - \mathsf{p}^0\|_1$, allowing it to be absorbed into λ_1. We use SDRIFT for all experimental evaluation.

Both of our algorithms, DRIFT and SDRIFT, directly benefit from the theoretical guarantees of Theorem 2, since they seek to minimize the right-hand side of that bound, or a natural upper bound derived by replacing $\mathsf{dis}(\mathsf{q}, \mathsf{p}^0)$ term by $\|\mathsf{q} - \mathsf{p}^0\|_1$.

Note that $\mathsf{dis}(\mathsf{q}, \mathsf{p}^0)$ is a convex function of q since it is a supremum of convex functions of q: $\mathsf{dis}(\mathsf{q}, \mathsf{p}^0) = \sup_{h \in \mathsf{H}} \{\sum_{i=1}^{m} (\mathsf{q}_i - \mathsf{p}_i^0) \ell(h(x_i), y_i)\}$. Thus, when the loss function ℓ is convex with respect to its first argument, the objective function is convex in q and convex in h. In general, however, it is not jointly convex. To minimize the objective, we use alternate minimization or DC-programming. Here, alternate minimization switches between optimizing with respect to h or with respect to q, each time solving a convex optimization problem. The method admits convergence guarantees under certain assumptions (Grippo and Sciandrone, 2000; Li et al., 2019; Beck, 2015). The description and guarantees for DC-programming are discussed in Appendix C.

4.2 Discrepancy Estimation

The optimization problem for our DRIFT algorithm requires discrepancy values $d_t = \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)$, which we can estimate from labeled samples. Here, we analyze this estimation problem in detail.

An empirical estimate $\widehat{d}_t$ of the discrepancy d_t can be obtained as the solution of the problem:

$$\max_{h \in \mathsf{H}} \left\{ \frac{1}{m_{T+1}} \sum_{i=n_{T+1}+1}^{n_{T+1}+m_{T+1}} \ell(h(x_i), y_i) - \frac{1}{m_t} \sum_{i=n_t+1}^{n_t+m_t} \ell(h(x_i), y_i) \right\}.$$

When the loss function ℓ is convex, the objective function is a difference of two convex functions. Thus, the problem can be cast as an instance of DC-programming, which can

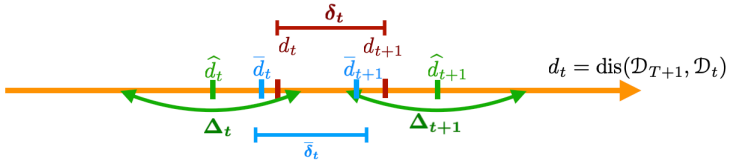

Figure 2: Enhanced Discrepancy Estimation: $\widehat{d}_t$s are original discrepancy estimates; $\overline{d}_t$s are corrected estimates leveraging the higher quality estimates $\overline{\delta}_t$s and the sequentiality of the drifting distribution.

be tackled using the DCA algorithm (Tao and An, 1998), see also Appendix C. In the special case of the squared loss, the problem is an instance of the *trust-region problem* and a method based on the DCA algorithm is guaranteed to converge to the global optimum (Tao and An, 1998). More generally, the global optimum can be found by combining the DCA algorithm with a branch-and-bound or cutting plane method (Tuy, 1964; Horst and Thoai, 1999; Tao and An, 1997). Reformulating the maximization problem as a minimization, the DCA solution consists of solving the following sequence of convex optimizations with h_{k+1} the solution of kth problem, $k \in [K]$, and h_1 chosen at random:

$$h_{k+1} \in \operatorname*{argmin}_{h \in \mathsf{H}} \left\{ \frac{1}{m_t} \sum_{i=n_t+1}^{n_t+m_t} \ell(h(x_i), y_i) - \frac{1}{m_{T+1}} \sum_{i=n_{T+1}+1}^{n_{T+1}+m_{T+1}} \nabla \ell(h_k(x_i), y_i) \cdot (h - h_k) \right\},$$

where the second term of the objective is obtained by linearization of the loss, with $\nabla \ell$ a sub-gradient of the loss. By McDiarmid's inequality, with high probability, $|\mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t) - \widehat{d}_t|$ can be upper-bounded by $O\Big(\sqrt{1/m_t + 1/m_{T+1}}\Big)$.

Finer guarantees can be given when the discrepancy is relatively small, using relative deviation bounds or Bernstein-type bounds (Cortes et al., 2019a). When the sample $S_{T=1}$ is large enough, we can reduce the hypothesis space H and have a more precise local discrepancy where the maximum is now taken over this smaller set. We reduce H by training a relatively accurate classifier $h_{\mathsf{D}_{T+1}}$ on a fraction n of points from $S_{T=1}$ so we can restrict H to a ball $\mathsf{B}(h_{\mathsf{D}_{T+1}}, r)$ of radius $r \sim 1/\sqrt{n}$.

We could use directly the discrepancy estimates $\widehat{d}_t$ in the optimization problem of our DRIFT algorithm. However, we can leverage the sequential aspect of our distribution drift problem to derive better estimates. Note that the width $\quad t$ of the confidence interval guaranteed by our learning bounds is in $O\Big(\sqrt{1/m_t + 1/m_{T+1}}\Big)$ and while we expect m_t to be typically large, m_{T+1} could be only moderately large and affect the accuracy of our estimation. First, note that, by the triangle inequality, for any $t \in [T-1]$, the following holds: $\mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_{t+1}) - \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t) \leq \mathsf{dis}(\mathsf{D}_t, \mathsf{D}_{t+1})$. Thus, we have $|d_{t+1} - d_t| \leq \mathsf{Dis}(\mathsf{D}_t, \mathsf{D}_{t+1})$. In many prior analyses of the drifting distribution problem, consecutive distributions are assumed to be δ-close (Helmbold and Long, 1994; Long, 1999; Mohri and Muñoz Medina, 2012) for the ℓ_1-distance or the two-sided discrepancy. Thus, we could adopt the assumption $\mathsf{Dis}(\mathsf{D}_t, \mathsf{D}_{t+1}) \leq \delta$ here. However, we can instead estimate accurately $\mathsf{Dis}(\mathsf{D}_t, \mathsf{D}_{t+1})$ modulo an error in $O\Big(\sqrt{1/m_t + 1/m_{t+1}}\Big)$ which would be small, since both m_t and m_{t+1} are typically large. Let $\widehat{\delta}_t$ denote that estimate, then this leads to searching our discrepancy estimated $\overline{d}_t$ as the solution of the following problem:

$$\min_{\overline{d}_1, \ldots, \overline{d}_T} \sum_{t=1}^{T} \left|\overline{d}_t - \widehat{d}_t\right|^2 \quad \text{s.t.} \quad |\overline{d}_{t+1} - \overline{d}_t| \leq \overline{\delta}_t = \widehat{\delta}_t + \sqrt{\tfrac{1}{m_t} + \tfrac{1}{m_{t+1}}}, \tag{2}$$

which helps us derive better estimates, as illustrated in Figure 2. Note that, with high probability, the true discrepancies d_t satisfy the constraints and are thus feasible solutions.

Let us also add that some authors have (naively) suggested to simply remove the supremum in the definition of discrepancy, especially in a scenario where labeled samples are available from the target domain. But, without the supremum, the analysis and the optimization would simply boil down to training on the (small) target labeled sample S_{T+1}, since the sum of the empirical loss and empirical estimate of the difference without a supremum is then $\sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \left[\sum_{i=n_{T+1}+1}^{n_{T+1}+m_{T+1}} \mathsf{q}_i \ell(h(x_i), y_i) - \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i)\right] = \sum_{i=n_{T+1}+1}^{n_{T+1}+m_{T+1}} \mathsf{q}_i \ell(h(x_i), y_i)$, thereby losing the benefit of the labeled data from the T segments. Furthermore, we cannot derive generalization bounds such as those of Theorems 1 and 2 in the absence of the supremum.

Computational complexity: the computational complexity of our algorithm corresponds to that of estimating the discrepancies and subsequently that of solving the main optimization problem via alternating minimization. The main cost for discrepancy estimation is that of solving the DC-program in 4.2. Each DC-program is solved by solving a sequence of K convex optimization problems. Thus, the total complexity for this part is $O(TKC)$, where C is the cost of solving each convex program, which depends on the properties of the loss and hypothesis set. In our experiments, K is never more than 20. The cost of our alternating minimization is that of solving K' times two convex optimizations C_q and C_h, thus in total $O(K'(C_q + C_h))$. C_h is the cost of a standard weighted-ERM for the hypothesis set considered. C_q runs in $O(m)$ and K' is never more than 50

in our experiments. Our experiments show that, in practice, our algorithm is very efficient and practical.

4.3 Automatic Determination of Distributions D_t

The DRIFT algorithm hinges on the knowledge of the segments supporting the distributions D_t, which are used to estimate discrepancy and improve predictions on the target segment D_{T+1}. Often, the distributions D_t admit an inherent time segmentation such as days, weeks, or months, but, for some other distributions, there may not be such a natural pattern, and one can ask how to determine the splits automatically from data. There is a wide literature on drift detection tackling this problem (see Appendix A). Standard algorithms for change-point detection in time series analysis also provide useful tools relevant to this task. Here, we briefly describe a natural method related to discrepancy.

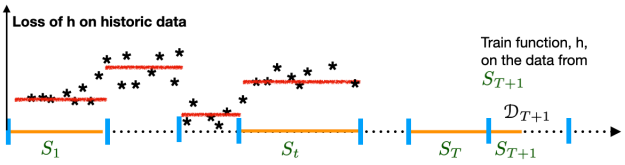

Figure 3: Illustration of how to automatically determine the distributions D_t with homogeneous discrepancies $\mathsf{dis}(T+1, t)$. A classifier h is determined by minimizing its loss on the data S_{T+1}. Its loss on the historic data is determined, and a step function fitted to the losses.

The distributions D_t of the DRIFT algorithm are characterized by their discrepancy $\mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)$. In the absence of the segmentation information, we cannot estimate these quantities. But, we can use a classifier trained on the target sample to identify the segments, using its losses on historical data. The difference of the expected loss of this classifier on the target and on any past segment provides a lower bound on the corresponding discrepancy. Thus, let h be a classifier trained on the target sample S_{T+1}. We apply h to the historical data and record its pointwise losses, see Figure 3. One may then fit a piecewise constant function number of points per region to ensure estimation accuracy. The knots determined in this way specify the split between the distributions. A discrepancy lower bound for the region can be found from the differences in losses of h on the regions.

4.4 Weakly Supervised Drifting

By unsupervised or weakly supervised drifting, we refer to the scenario where we have very few or no labeled data points from the target D_{T+1}. We study this scenario in detail in Appendix D: we extend our bounds to that scenario, bound labeled discrepancy terms in terms of unlabeled discrepancy ones, and then derive an algorithm WDRIFT based on unlabeled discrepancy terms.

4.5 Extension to Other Algorithms

There are several algorithms used in the context of drifting that consist of assigning weights, often fixed ones such as exponentially decaying ones, to the samples losses. Other reweighting algorithms originally designed for domain adaptation are also sometimes used in this context, including KMM (Huang et al., 2006), KLIEP (Sugiyama et al., 2007), importance weighting (Cortes et al., 2010), discrepancy minimization (Cortes and Mohri, 2014) and many others. Our learning bounds for weighted samples are general and can be applied to the analysis of these algorithms. Our analysis suggests however that an algorithm such as DRIFT, which seeks to minimize the bounds, benefits from a more favorable theoretical guarantee.

Note that we consider a batch scenario with infrequent updates or retraining, for example large recommendation systems with monthly, bimonthly or other season-long updates. Retraining is further natural since each new segment comes with a fair amount of new labeled data. In scenarios with more frequent segments, we can learn an ensemble of the hypotheses h_T learned for previous target segments T, using the small labeled sample from the new target, or simply fine-tune the previous model, and only retrain occasionally.

4.6 Relationship with Multiple-Source Adaptation

While the main motivation for this work is the batch distribution drifting problem, our theory, analysis, and algorithms also provide a new and principled solution to the problem of multiple-source adaptation (MSA) with a specific target distribution and labeled data. In this scenario, the learner receives labeled data from multiple source distributions, as well as a moderate amount of labeled data from the target distribution. The goal is to come up with an accurate predictor for that target distribution. This problem is distinct from the drifting one we consider by the absence of sequentiality of the data received by the learner, which we specifically leverage in our case to derive more accurate estimates of the discrepancies. If we ignore the sequential aspect, critical in drifting, our theoretical analysis holds for this MSA scenario and, if we use directly the discrepancy estimates $\widehat{d}_t$ defined in Section 4.2, the algorithm readily applies and provides a principled solution for this problem.

This MSA scenario is distinct from the one first introduced and analyzed by Mansour et al. (2009a,b), later extensively studied and extended by Hoffman et al. (2018, 2021, 2022) and Cortes et al. (2021b). In that scenario, the learner has only access to unlabeled data from the source distributions and a trained predictor for each source domain, with no access to source labeled data or target labeled or unlabeled data. The goal in that scenario is to combine the existing pre-trained source models to come up with a predictor that is accurate for any distribution that is a mixture of the source distributions. This approach has been further used

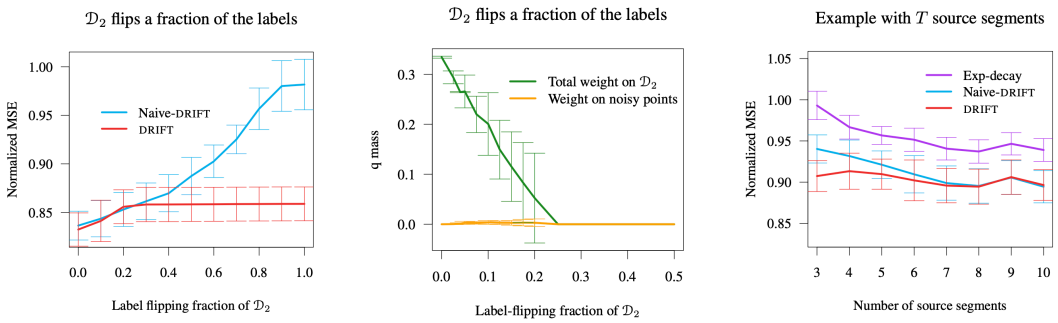

Figure 4: Synthetic Data: (Left) and (Middle) with label-flipping and three segments $\mathsf{D}_1 = \mathsf{D}_3 \neq \mathsf{D}_2$. Left: MSE as a function of increasing discrepancy; Middle: the amount of q-mass assigned to D_2 by SDRIFT, in particularly the points with flipped labels. Right: MSE performance for k sources.

successfully in many applications such as object recognition (Hoffman et al., 2012; Gong et al., 2013a,b). Since the target distribution is not specified beforehand, this scenario can be viewed as an instance of the *domain generalization* problem, where no labeled input data is available but only relatively accurate predictors for each source domain.

Zhao et al. (2018) and Wen et al. (2020) studied an MSA scenario where there are multiple source domains with labeled instances and one target domain with unlabeled instances. This scenario is also distinct from ours since no labeled data is available from the target. Our learning bounds can, however, be used to analyze the scenario considered by these authors and lead to more general guarantees with uniform convergence bounds over the weights and finer quantities such as the weighted Rademacher complexity and labeled discrepancies. In Section 5, we are presenting an empirical comparison with MDAN (Zhao et al., 2018) and DARN (Wen et al., 2020). Cortes et al. (2021a) study a similar but distinct scenario in a boosting setting, where the target distribution is unspecified but assumed to be a mixture of the source distributions and where no target data is available. The domain generalization problem of Blanchard et al. (2021) is a similar scenario but one where at prediction time unlabeled data is available from the target distribution.

Finally, an MSA scenario with labeled data from the source domains and only limited labeled data from the target domain was studied in (Konstantinov and Lampert, 2019; Mansour et al., 2021; Shui et al., 2021). Mansour et al. (2021) presented a theoretical analysis of this scenario using the notion of discrepancy but their analysis and algorithms also assume that the target distribution is a mixture of the source distributions, as in (Mansour et al., 2009a,b), or is close to being a mixture. They argued that the approach adopted by Konstantinov and Lampert (2019) could be sub-optimal in general. The analyses of Konstantinov and Lampert (2019) and Mansour et al. (2021) both use the discrepancy, which is a finer divergence measure than the Wasserstein distance used by Shui et al. (2021). The scenario studied in these publications is distinct from ours since we do not require an assumption about the target distribution and since we assume a moderate amount of labeled data from the target. The latter assumption can be relaxed in our case, however, since we can extend our theory to the weakly supervised setting, as discussed in Section 4.4. Our algorithms provide in fact a novel and principled solution to this problem based on a reweighting of all sample points. This is in contrast with the solutions considered in these publications, which only assign a global weight to each domain.

5 Experimental Evaluation

Here, we study properties of our new DRIFT algorithm and report a series of comparison results with several baselines.

5.1 Synthetic Data

Our synthetic data experiments demonstrate how the DRIFT algorithm effectively and automatically hones in on low-discrepancy source segments to boost its performance. We predetermine the distributions to control the discrepancy between the distributions. All experiments are for the regression setting and use a linear hypothesis set and a squared error loss. For all examples, $x \in \mathsf{R}^n, n = 20$, is sampled from a normal distribution, $\mathcal{N}(0, I_{n\times n})$. The labels y are based on a randomly drawn weight vector $w \in \mathsf{R}^n$ of unit length, and $y = w \cdot x$. We use the SDRIFT algorithm, see Section 4.1, for all experimental evaluation.

The first scenario is with just two source segments with samples S_1 and S_2, and a target sample S_3. To illustrate the benefit of our new approach, S_1 and S_3 are drawn from the same distribution, while S_2 differs slightly. We artificially control its discrepancy d_2 to the S_3 by flipping the sign of a fraction of its labels.

We estimate the empirical discrepancy, $\widehat{d}_2$ as outlined in Section 4, and then run algorithm SDRIFT by carrying out a grid search over the three hyperparameters, λ_∞, λ_1, and λ_2. The best performance is determined by evaluation on an independent validation set of size $10|S_i|$, with $|S_i| = 120$, and we report mean and standard deviations over 10 runs as measured on a test set of size $100|S_i|$. Performance in terms of MSE and amount of q-weight assigned to the sample

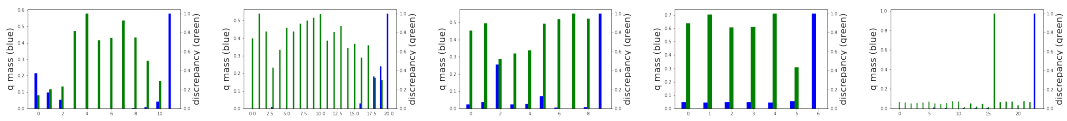

Figure 5: A plot of the total average probability mass assigned (in blue) to each segment by the SDRIFT algorithm along side the corresponding (normalized) discrepancy values (in green).

Table 1: MSE of the SDRIFT Algorithm Against Baselines for Regression Tasks. We report relative errors normalized so that training on target has an MSE of 1.0. Best results in boldface.

Dataset	KMM	DM	MM	EXP	BSTS	SDRIFT
Wind	1.19 ± 0.07	1.12 ± 0.06	$1.19 \pm .07$	0.98 ± 0.04	0.98 ± 0.01	$\mathbf{0.95 \pm 0.02}$
Airline	2.45 ± 0.17	1.78 ± 0.11	1.41 ± 0.28	0.98 ± 0.03	$\mathbf{0.945 \pm 0.01}$	$\mathbf{0.94 \pm 0.03}$
Gas	0.45 ± 0.02	0.42 ± 0.02	0.47 ± 0.04	0.94 ± 0.03	1.02 ± 0.2	$\mathbf{0.4 \pm 0.01}$
News	1.1 ± 0.02	1.13 ± 0.01	1.1 ± 0.03	0.98 ± 0.02	1.00 ± 0.02	$\mathbf{0.97 \pm 0.004}$
Traffic	2.3 ± 0.12	2.2 ± 0.11	0.99 ± 0.12	0.996 ± 0.008	0.98 ± 0.03	$\mathbf{0.96 \pm 0.006}$

S_2 is illustrated in Figure 4. In the figure we compare the performance to that of Naive-DRIFT, see Appendix E, where the samples S_1 and S_2 are assumed to be drawn from the same distribution.

In all regression experiments, we normalize the MSE by the one obtained from training on S_3 only. Figure 4-Left illustrates how the samples from D_1 and D_2 aide learning. For low noise level, and hence low discrepancy, the algorithm obtains significantly better performance, MSE < 1. As the discrepancy $\widehat{d}_2$ increases, the MSE increases. However, even when all the signs of the labels of S_2 are flipped, the algorithm is able to make use of the good samples of S_1 and performs better than training just on S_3. This left plot also demonstrates the performance gains over Naive-DRIFT, which cannot take advantage of the difference in distributions $\mathsf{D}_1 \neq \mathsf{D}_2$. The middle plot shows the amount of q-weight allocated by the SDRIFT algorithm to the points in S_2, and also the points with noisy flipped labels. As the discrepancy increases, less total q-mass is allocated to the points in D_2. Even as the label-flipping fraction becomes very small, SDRIFT detects the few noisy points and gives them almost no weight.

Figure 4-Right also illustrates the performance of SDRIFT for a synthetic setting with T sources diverging away from S_{T+1}. Higher values of T result in samples with smaller discrepancy to D_{T+1} and the overall performance improves. For this setting a natural baseline is exponential decay of the weights q, keeping them constant within a segment. However as the figure illustrates, SDRIFT also outperforms this baseline. For details and more experiments using synthetic data, see Appendix F.

5.2 Real-World Data

We compare SDRIFT to several baseline algorithms in real-world regression and classification settings.

Baseline Algorithms. We compare with the following baseline algorithms, modified to incorporate the labeled sample S_{T+1}:

KMM (Huang et al., 2006): The algorithm assigns weights to the sample points in $S_1, S_2, \ldots, S_T$ so that the kernelized mean feature vector of each segment matches that of S_{T+1} in terms of mean squared error. We run linear KMM for each segment to derive the q_i-weights. We then minimize a squared error loss using these weights, adding in the target points with uniform weights.

DM (Cortes and Mohri, 2014): This method also performs a two-stage optimization, but uses the unlabeled discrepancy to determine weights per segment. These weights and uniform $1/(m_{T+1})$ weights for the target points are then used for training a squared error loss.

MM (Mohri and Muñoz Medina, 2012): In an online learning phase this algorithm first generates multiple hypotheses. In a second phase it determines weights to form a weighted average of the hypotheses.

EXP (Ross et al., 2012): This method often used in drifting and time-series modeling exponentially down-weights past samples. For our comparisons, we keep the weights fixed within each past segment.

BSTS (Scott and Varian, 2014): A state-of-the-art time-series modeling technique that incorporates drift as well as segment indicators.

MDAN (Zhao et al., 2018) and **DARN** (Wen et al., 2020): two state-of-the-art multiple-source domain adaptation algorithms for the scenario of labeled source data plus unlabeled target data (only).

Regression Tasks. We compare the SDRIFT algorithm to that of the baselines on a number of regression tasks. For pointers to the dataset and details on the experimental procedure, see Appendix F. For each dataset, we form T source segments and define a target distribution. We estimate the discrepancy $\widehat{d}_i$, $i \in [T]$, as outlined in Section 4, determine the best hyper-parameters via cross-validation on an

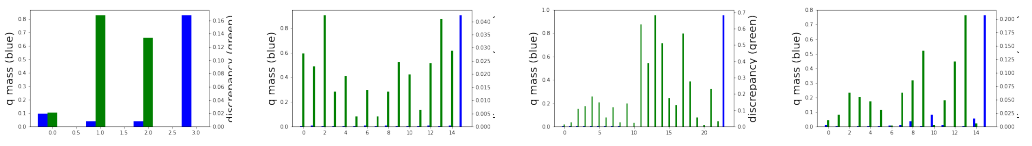

Figure 6: Average probability mass assigned (in blue) to each segment by the SDRIFT algorithm along side the corresponding (normalized) discrepancy values (in green).

Table 2: Accuracy of the SDRIFT Against Baselines for Classification Tasks. We report relative accuracies normalized so training on just target has an accuracy of 1.0. Best results are in boldface.

Dataset	KMM	DM	MM	EXP	BSTS	MDAN	DARN	SDRIFT
STAGGER	0.69 ± 0.006	0.73 ± 0.05	0.74 ± 0.01	1.02 ± 0.03	0.98 ± 0.02	0.71 ± 0.02	0.88 ± 0.04	**1.05 ± 0.03**
Electricity	0.95 ± 0.01	0.93 ± 0.02	0.84 ± 0.02	1.09 ± 0.02	1.02 ± 0.07	**1.12 ± 0.02**	1.08 ± 0.02	**1.13 ± 0.02**
Room Occupancy	0.62 ± 0.02	0.63 ± 0.01	0.72 ± 0.03	1.02 ± 0.04	**1.07 ± 0.01**	0.60 ± 0.02	0.60 ± 0.02	1.02 ± 0.02
Adult Income	0.97 ± 0.007	0.98 ± 0.01	0.99 ± 0.005	**1.00 ± 0.01**	**1.00 ± 0.02**	0.97 ± 0.01	0.98 ± 0.01	**1.01 ± 0.004**

independent validation set and measure the test error on a different and independent test set. Reported results are mean and standard deviations over ten different splits of the data. For the objective, we use the squared loss and the hypothesis set is that of linear functions.

Table 1 provides results for 5 regression tasks in terms of MSE, normalized so that training only on the data from the target segment gives an error of MSE = 1. Hence, we are seeking algorithms achieving a better performance, that is MSE<1. The KMM and DM algorithms admits no principled mechanism for down-weighing segments that are too far from the target, thus all segments are assigned the same total mass in the loss function. In contrast, as can be seen from Figure 5, the SDRIFT algorithm effectively discards many segments and assigns them little or no q-mass, indicated by small blue segment bars. In addition, KMM and DM do not make use of any labels to match distributions.

The MM algorithm does incorporate the performance of the hypotheses found in the online training phase, and hence in its final training it puts most weight on the hypotheses from the target segment. However, the simple online hypotheses are weaker than the result from batch training on the target and as a result, this method also obtains an MSE>1. Finally, we compare to the BSTS algorithm. For dataset with a clear time component: `wind` (month), `news` (weekday), `airline` (hour), `traffic` (hour) it provides a strong baseline, but proves sub-optimal for general drifting problems. BSTS falls short similarly for classification, see below.

Figure 5 provides further insight into how the SDRIFT algorithm achieves an improved performance and how it effectively allocates q-mass to the source segments with lower discrepancy. The green bars indicates the estimated discrepancy between the sources and the target segments, the blue bars illustrate the q-mass the algorithm assigns. As expected, segments with higher discrepancy are assigned lower q-mass. The figure and a further discussion can be found in Appendix F.

Classification Tasks. We report results on 4 classification tasks for which we report performance in terms of accuracy and normalize so training only on the target gives an accuracy of one (see Appendix F for more details). Thus, well-performing algorithms have an accuracy superior to one. Table 2 reports our results. KMM, DM, and MM again under-perform, see discussion under regression tasks. The EXP algorithm is competitive and ties in some instances with SDRIFT, for example when past segments receive very little weight from SDRIFT (`Room`) or when the number of past segments is small (`STAGGER`). BSTS outperforms on `Room` that has a strong time component. Figure 6 shows the discrepancies and q-mass allocated by the SDRIFT algorithm to each segment. Again, we see how the algorithm effectively leverages segments with lower target discrepancy.

6 Conclusion

We presented a comprehensive study of a distribution drift problem that arises in many applications. We presented a detailed theoretical analysis of this problem based on the notion of labeled discrepancy, including learning bounds that hold uniformly over the sample weights. We also gave a principled algorithm for this problem that directly benefits from our theoretical analysis. We showed how the sequential nature of the data can be exploited when estimating the discrepancy. We further extended both our theory and algorithms to a weakly supervised scenario where few or no labeled is at hand from the target domain.

Our analysis and theory are likely to be useful in the study of other drifting problems and adaptation tasks. In fact, a direct by-product of our study is a principled solution to the problem of multiple-source adaptation with labeled source data and a moderate amount of target labeled data, which we briefly described. Our experimental results suggest that our algorithm is of practical use with significant benefits in several tasks, including the scenario of multiple-source adaptation just outlined.

References

D. Adamskiy, W. M. Koolen, A. Chernov, and V. Vovk. A closer look at adaptive regret. In *ALT*, pages 290–304, 2012.

S. Bach and M. Maloof. A Bayesian approach to concept drift. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

P. L. Bartlett. Learning with a slowly changing distribution. In *Proceedings of COLT*, pages 243–252, New York, NY, USA, 1992. ACM.

P. L. Bartlett, S. Ben-David, and S. Kulkarni. Learning changing concepts by exploiting the structure of change. *Machine Learning*, 41:153–174, 2000.

R. D. Barve and P. M. Long. On the complexity of learning from drifting distributions. *Information and Computation*, 138(2):101–123, 1997.

A. Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM J. Optim.*, 25(1):185–209, 2015.

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Proceedings of NIPS*, pages 137–144. MIT Press, 2006.

A. Bifet and E. Ikonomovska. Airline dataset, 2009. URL https://www.openml.org/d/1169.

G. Blanchard, A. A. Deshmukh, Ü. Dogan, G. Lee, and C. Scott. Domain generalization by marginal transfer learning. *J. Mach. Learn. Res.*, 22:2:1–2:55, 2021.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *J Econometrics*, 1986.

G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer-Verlag, New York, 1986.

K. H. Brodersen, F. Gallusser, J. Koehler, N. Remy, and S. L. Scott. Inferring causal impact using Bayesian structural time-series models. *Annals of Applied Statistics*, 9:247–274, 2014. URL https://research.google/pubs/pub41854/.

P. Campagnoli, S. Petrone, and G. Petris. *Dynamic Linear Models with R*. Springer New York, 2009. doi: 10.1007/b135794.

L. M. Candanedo and V. Feldheim. Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models. *Energy and Buildings*, 112:28–39, 2016. ISSN 0378-7788. doi: https://doi.org/10.1016/j.enbuild.2015.11.071. URL https://www.sciencedirect.com/science/article/pii/S0378778815304357.

G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Tracking the best hyperplane with a simple budget perceptron. *Machine Learning*, 69(2/3):143–167, 2007.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

N. Cesa-Bianchi, P. Gaillard, G. Lugosi, and G. Stoltz. Mirror descent meets fixed share (and feels no regret). In *NIPS*, pages 980–988, 2012.

C. Cortes and M. Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theor. Comput. Sci.*, 519:103–126, 2014.

C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. In *Proceedings of NIPS*, pages 442–450. Curran Associates, Inc., 2010.

C. Cortes, S. Greenberg, and M. Mohri. Relative deviation learning bounds and generalization with unbounded loss functions. *Ann. Math. Artif. Intell.*, 85(1):45–70, 2019a.

C. Cortes, M. Mohri, and A. Muñoz Medina. Adaptation based on generalized discrepancy. *J. Mach. Learn. Res.*, 20:1:1–1:30, 2019b.

C. Cortes, M. Mohri, D. Storcheus, and A. T. Suresh. Boosting with multiple sources. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Proceedings of NeurIPS*, pages 17373–17387, 2021a.

C. Cortes, M. Mohri, A. T. Suresh, and N. Zhang. A discriminative technique for multiple-source adaptation. In M. Meila and T. Zhang, editors, *Proceedings of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 2132–2143. PMLR, 2021b.

K. Crammer, E. Even-Dar, Y. Mansour, and J. W. Vaughan. Regret minimization with concept drift. In *COLT*, pages 168–180, 2010.

A. Daniely, A. Gonen, and S. Shalev-Shwartz. Strongly adaptive online learning. In *Proceedings of ICML*, pages 1405–1411, 2015.

M. DOT, 2019. URL https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

J. Durbin and S. J. Koopman. A simple and efficient simulation smoother for state space time series analysis. *Biometrika*, 89(3):603–615, 2002. ISSN 00063444.

R. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

K. Fernandes. A proactive intelligent decision support system for predicting the popularity of online news. In *Springer Science and Business Media*, 08 2015. doi: 10.1007/978-3-319-23485-4_53.

Y. Freund and Y. Mansour. Learning under persistent drift. In *EuroColt*, pages 109–118, 1997.

J. Gama, P. Medas, G. Castillo, and P. Rodrigues. Learning with drift detection. In *Advances in Artificial Intelligence – SBIA 2004*, pages 286–295, 2004.

J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and H. Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46, 04 2014. doi: 10.1145/2523813.

Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*, volume 28, pages 222–230, 2013a.

B. Gong, K. Grauman, and F. Sha. Reshaping visual datasets for domain adaptation. In *NIPS*, pages 1286–1294, 2013b.

L. Grippo and M. Sciandrone. On the convergence of the block nonlinear gauss-seidel method under convex constraints. *Oper. Res. Lett.*, 26(3):127–136, 2000.

A. Gyorgy, T. Linder, and G. Lugosi. Efficient tracking of large classes of experts. *IEEE Transactions on Information Theory*, 58(11):6709–6725, 2012.

H. Gâlmeanu and R. Andonie. Concept drift adaptation with incremental–decremental svm. *Applied Sciences*, 11(20), 2021.

J. D. Hamilton. *Time series analysis*. Princeton, 1994.

M. Harries and N. S. Wales. Splice-2 comparative evaluation: Electricity pricing, 1999.

A. C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1990. doi: 10.1017/CBO9781107049994.

J. Haslett and A. E. Raftery. Space-time modeling with long-memory dependence: assessing ireland's wind-power resource. *Journal of the Royal Statistical Society*, 38(1), 1989.

E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of ICML*, pages 393–400. ACM, 2009.

D. P. Helmbold and P. M. Long. Tracking drifting concepts by minimizing disagreements. *Machine Learning*, 14(1): 27–46, 1994.

M. Herbster and M. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1: 281–309, 2001.

J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, volume 7573, pages 702–715, 2012.

J. Hoffman, M. Mohri, and N. Zhang. Algorithms and theory for multiple-source adaptation. In *Proceedings of NeurIPS*, pages 8256–8266, 2018.

J. Hoffman, M. Mohri, and N. Zhang. Multiple-source adaptation theory and algorithms. *Ann. Math. Artif. Intell.*, 89(3-4):237–270, 2021.

J. Hoffman, M. Mohri, and N. Zhang. Multiple-source adaptation theory and algorithms – addendum. *Ann. Math. Artif. Intell.*, 90(6):569–572, 2022.

R. Horst and N. V. Thoai. DC programming: overview. *Journal of Optimization Theory and Applications*, 103(1): 1–43, 1999.

J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS 2006*, volume 19, pages 601–608, 2006.

D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proceedings of VLDB*, pages 180–191. Morgan Kaufmann, 2004.

R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intell. Data Anal.*, 8:281–300, 08 2004.

N. Konstantinov and C. Lampert. Robust learning from untrusted sources. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 3488–3498. PMLR, 2019.

W. M. Koolen and S. de Rooij. Universal codes from switching strategies. *IEEE Transactions on Information Theory*, 59(11):7168–7185, 2013.

V. Kuznetsov and M. Mohri. Learning theory and algorithms for forecasting non-stationary time series. In *Proceedings of NIPS*, volume 28. Curran Associates, Inc., 2015.

Q. Li, Z. Zhu, and G. Tang. Alternating minimizations converge to second-order optimal solutions. In *Proceedings of ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 3935–3943. PMLR, 2019.

P. M. Long. The complexity of learning according to two models of a drifting environment. *Machine Learning*, 37: 337–354, 1999.

J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *CoRR*, abs/2004.05785, 2020.

W. Lu, J. Wang, Y. Chen, and X. Sun. DIVERSIFY to generalize: Learning generalized representations for time series classification, 2022. URL https://openreview.net/forum?id=NX0nX7TE4lc.

J. López Lobo. Synthetic datasets for concept drift detection purposes, 2020. URL https://doi.org/10.7910/DVN/5OWRGB.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009a.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *NIPS*, pages 1041–1048, 2009b.

Y. Mansour, M. Mohri, J. Ro, A. T. Suresh, and K. Wu. A theory of multiple-source adaptation with limited target labeled data. In *Proceedings of AISTATS*, volume 130, pages 2332–2340. PMLR, 2021.

R. Meir. Nonparametric time series prediction through adaptive model selection. *Machine Learning*, pages 5–34, 2000.

M. Mohri and A. Muñoz Medina. New analysis and algorithm for learning with drifting distributions. In *Proceedings of ALT*, volume 7568 of *Lecture Notes in Computer Science*, pages 124–138. Springer, 2012.

M. Mohri and S. Yang. Competing with automata-based expert sequences. In *Proceedings of AISTATS*, volume 84, pages 1732–1740. PMLR, 09–11 Apr 2018.

C. Monteleoni and T. S. Jaakkola. Online learning of non-stationary sequences. In *NIPS*, page None, 2003.

R. H. Moulton, H. L. Viktor, N. Japkowicz, and J. Gama. Clustering in the presence of concept drift. In *ECML/PKDD*, 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12 (Oct):2825–2830, 2011.

I. Rodriguez-Lujan, J. Fonollosa, A. Vergara, M. Homer, and R. Huerta. On the calibration of sensor arrays for pattern recognition using the minimal number of experiments. *Chemometrics and Intelligent Laboratory Systems*, 130:123–134, 2014. ISSN 0169-7439. doi: https://doi.org/10.1016/j.chemolab.2013.10.012. URL https://www.sciencedirect.com/science/article/pii/S0169743913001937.

G. J. Ross, N. M. Adams, D. K. Tasoulis, and D. J. Hand. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, 33 (2):191–198, jan 2012. doi: 10.1016/j.patrec.2011.08.019. URL https://doi.org/10.1016%2Fj.patrec.2011.08.019.

S. L. Scott and H. R. Varian. Predicting the present with bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5 (1-2):4–23, 2014.

C. Shui, Z. Li, J. Li, C. Gagné, C. X. Ling, and B. Wang. Aggregating from multiple target-shifted sources. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9638–9648. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/shui21a.html.

A. Sicilia, X. Zhao, and S. J. Hwang. Domain adversarial neural networks for domain generalization: When it works and how to improve, 2021.

A. Sicilia, K. Atwell, M. Alikhani, and S. J. Hwang. Pac-bayesian domain adaptation bounds for multiclass learners. In J. Cussens and K. Zhang, editors, *Proceedings of UAI*, volume 180 of *Proceedings of Machine Learning Research*, pages 1824–1834. PMLR, 2022.

B. Silva, N. Marques, and G. Panosso. Applying neural networks for concept drift detection in financial markets. *CEUR Workshop Proceedings*, 960:43–47, 01 2012.

B. K. Sriperumbudur, D. A. Torres, and G. R. G. Lanckriet. Sparse eigen methods by D.C. programming. In *ICML*, pages 831–838, 2007.

M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Proceedings of NIPS*, pages 1433–1440. Curran Associates, Inc., 2007.

A. Tahmasbi, E. Jothimurugesan, S. Tirthapura, and P. B. Gibbons. Driftsurf: Stable-state / reactive-state learning under concept drift. In M. Meila and T. Zhang, editors, *Proceedings of ICML*, volume 139, pages 10054–10064, 18–24 Jul 2021.

P. D. Tao and L. T. H. An. Convex analysis approach to DC programming: theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.

P. D. Tao and L. T. H. An. A DC optimization algorithm for solving the trust-region subproblem. *SIAM Journal on Optimization*, 8(2):476–505, 1998.

A. Tsymbal. The problem of concept drift: Definitions and related work. Technical report, Department of Computer Science, Trinity College Dublin, Ireland, 05 2004. TCD-CS-2004-15.

H. Tuy. Concave programming under linear constraints. *Translated Soviet Mathematics*, 5:1437–1440, 1964.

A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166-167:320–329, 2012. ISSN 0925-4005. doi: https://doi.org/10.1016/j.snb.2012.01.074. URL https://www.sciencedirect.com/science/article/pii/S0925400512002018.

V. Vovk. Derandomizing stochastic prediction strategies. *Machine Learning*, 35(3):247–282, 1999.

J. Wen, R. Greiner, and D. Schuurmans. Domain aggregation networks for multi-source domain adaptation. In *International Conference on Machine Learning*, pages 10214–10224. PMLR, 2020.

G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23, 1996.

L. Yang. Active learning with a drifting distribution. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Proceedings of NIPS*, volume 24. Curran Associates, Inc., 2011.

A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

Y. Zhang, T. Liu, M. Long, and M. Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR, 2019.

H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J. Gordon. Adversarial multiple source domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Proceedings of NIPS*, volume 31, 2018.

P. Zhao, L.-W. Cai, and Z.-H. Zhou. Handling concept drift via model reuse. *Machine Learning*, 109, 2020.

Contents of Appendix

A Related Work **15**

A.1 Online Setting 15

A.2 Drift Detection 15

B Main Theorems **15**

C DC-Programming **19**

D Unsupervised or Weakly Supervised Drifting **19**

D.1 Unsupervised or Weakly Supervised Drifting Generalization Bounds 19

D.2 WDRIFT Algorithm 20

D.3 Theorems and Proofs 20

D.4 Labeled Discrepancy Upper Bounds 21

E Comparison of DRIFT and a Naive-DRIFT Solution **22**

F Experimental Results **23**

F.1 Synthetic Data 23

F.2 Regression Datasets 23

F.3 Classification Datasets 25

F.4 Hyperparameters for Real-World Data 25

F.5 Pseudocode for the Alternate Minimization Procedure 26

A Related Work

Here, we further discuss some related work. Let us emphasize that we do not anticipate any negative societal impact of our work in the near future.

A.1 Online Setting

In on-line learning, the benchmark typically adopted is that of external regret, which measures the cumulative loss of the algorithm against that of the best *static* expert in hindsight (Cesa-Bianchi and Lugosi, 2006). This framework was extended by Herbster and Warmuth (2001), who studied the scenario where the best expert could *shift* over time at most a finite number of times. The analysis was later improved to account for broader expert classes (Gyorgy et al., 2012) and to deal with unknown parameters (Monteleoni and Jaakkola, 2003). It was further generalized (Vovk, 1999; Cesa-Bianchi et al., 2012; Koolen and de Rooij, 2013) and used to extend the perceptron algorithm (Cavallanti et al., 2007). A more general theoretical and algorithmic analysis of online learning with dynamic sequences of experts based on weighted automata was given by Mohri and Yang (2018), which comprehensively covers past competitor classes considered in the literature. An alternative study of dynamic environments based on the notion of *adaptive regret* was also suggested by Hazan and Seshadhri (2009), which was later strengthened and generalized (Adamskiy et al., 2012; Daniely et al., 2015). Bartlett et al. (2000) considered other settings allowing arbitrary but infrequent changes, such as sequences corresponding to slow walks. Crammer et al. (2010) analyzed an intermediate model of drift based on a *near* function, where consecutive distributions could change arbitrarily, provided that the region of disagreement between nearby functions were assigned limited distribution mass at any time. Ensemble learning was suggested as a solution technique for drifting in Tsymbal (2004). In a somewhat related work, Zhao et al. (2020) introduced an algorithm based on model reuse and weight updating. Finally, a study of active learning in the online setting with drifting distributions was presented by Yang (2011).

A.2 Drift Detection

Much of the recent literature on drifting has been related to drift detection and subsequent model adaptation. The detection of a drift significant enough to warrant updating the model is critical, as retraining is computationally expensive. The theoretical results suggest the use of only a most recent set of training examples. Hence, it is important to identify a (changing) window of examples to train on. FLORA (Widmer and Kubat, 1996) was one of the original algorithms to train with a fixed window. Later versions of this algorithm study an adaptive window (using methods such as a Hoeffding statistical test in Gâlmeanu and Andonie (2021) which does not require subsequent entire model retraining) as well as gradual forgetting of data points (Gama et al., 2014; Klinkenberg, 2004). An error-based method of drift detection is now one of the most popular approaches to drift detection, originating from the Drift Detection Method of Gama et al. (2004), which identifies an acceptable level of error for the most recent window of online examples. Other methods include distribution-based drift detection and more recently the use of multiple (parallel or hierarchical) hypothesis tests to detect drift (Lu et al., 2020). A Bayesian approach has also been studied (Bach and Maloof, 2010). In an application to financial markets and more specifically the Dow Jones, neural networks have been used to detect concept drift (Silva et al., 2012). Analysis has also been extended to the active learning setting, where Tahmasbi et al. (2021) claim to outperform standalone drift detection.

B Main Theorems

Theorem 1. *Fix a vector* q *in* $[0,1]^m$*. Then, for any* $\delta > 0$*, with probability at least* $1-\delta$ *over the choice of a sample* S *drawn from* $\mathsf{D}_1^{m_1} \otimes \cdots \otimes \mathsf{D}_{T+1}^{m_{T+1}}$*, for all* $h \in \mathsf{H}$*:*

$$\mathcal{L}(\mathsf{D}_{T+1}, h) \leq \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\right) + 2\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) + \|\mathsf{q}\|_2 \sqrt{\frac{\log \frac{1}{\delta}}{2}}.$$

Furthermore, when q *is a distribution,* $\|\mathsf{q}\|_1 = 1$*, the discrepancy term can be replaced by* $\sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)$.

Proof. Let $\mathcal{L}_S(\mathsf{q}, h)$ denote the q-weighted empirical loss: $\mathcal{L}_S(\mathsf{q}, h) = \sum_{i=1}^{m} \mathsf{q}_t \ell(h(x_i), y_i)$. For any sample S drawn from $\mathsf{D}_1^{m_1} \otimes \cdots \otimes \mathsf{D}_{T+1}^{m_{T+1}}$, we define (S) as follows:

$$(S) = \sup_{h \in \mathsf{H}} \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathcal{L}(\mathsf{D}_t, h) - \mathcal{L}_S(\mathsf{q}, h).$$

Changing point x_i to some other point x_i' affects $\quad(S)$ at most by q_i, as we consider loss functions $\ell\colon \mathsf{Y}\times\mathsf{Y}\to\mathsf{R}$ assumed to take values in $[0,1]$. Thus, by McDiarmid's inequality, which only requires independent random variables and not the same distribution, for any $\delta>0$, with probability at least $1-\delta$, the following holds for all $h\in\mathsf{H}$:

$$\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t\mathcal{L}(\mathsf{D}_t,h)\le\mathcal{L}_S(\mathsf{q},h)+\mathsf{E}[\quad(S)]+\|\mathsf{q}\|_2\sqrt{\frac{\log\frac{1}{\delta}}{2}}. \tag{3}$$

We now analyze the expectation term. Observe that for any sample S, we can write:

$$\mathop{\mathsf{E}}_{S}[\mathcal{L}_S(\mathsf{q},h)]=\sum_{i=1}^{m}\mathsf{q}_i\,\mathsf{E}[\ell(h(x_i),y_i)]=\sum_{t=1}^{T+1}\sum_{i=1}^{m_t}\mathsf{q}_{n_t+i}\,\mathsf{E}[\ell(h(x_{n_t+i}),y_{n_t+i})]=\sum_{t=1}^{T+1}\sum_{i=1}^{m_t}\mathsf{q}_{n_t+i}\mathcal{L}(\mathsf{D}_t,h)=\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t\mathcal{L}(\mathsf{D}_t,h).$$

Thus, the expectation term can be expressed as follows:

$$\begin{aligned}
\mathsf{E}[\quad(S)] &= \mathop{\mathsf{E}}_{S}\left[\sup_{h\in\mathsf{H}}\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t\mathcal{L}(\mathsf{D}_t,h)-\mathcal{L}_S(\mathsf{q},h)\right]\\
&= \mathop{\mathsf{E}}_{S}\left[\sup_{h\in\mathsf{H}}\mathop{\mathsf{E}}_{S'}[\mathcal{L}_{S'}(\mathsf{q},h)-\mathcal{L}_S(\mathsf{q},h)]\right]\\
&\le \mathop{\mathsf{E}}_{S;S'}\left[\sup_{h\in\mathsf{H}}\mathcal{L}_{S'}(\mathsf{q},h)-\mathcal{L}_S(\mathsf{q},h)\right] && \text{(by the sub-additivity of the supremum operator)}\\
&= \mathop{\mathsf{E}}_{S;S'}\left[\sup_{h\in\mathsf{H}}\sum_{i=1}^{m}\mathsf{q}_i\ell(h(x_i'),y_i')-\mathsf{q}_i\ell(h(x_i),y_i)\right]\\
&= \mathop{\mathsf{E}}_{S;S';}\left[\sup_{h\in\mathsf{H}}\sum_{i=1}^{m}\sigma_i\Big(\mathsf{q}_i\ell(h(x_i'),y_i')-\mathsf{q}_i\ell(h(x_i),y_i)\Big)\right] && \text{(introducing Rademacher variables)}\\
&\le \mathop{\mathsf{E}}_{S';}\left[\sup_{h\in\mathsf{H}}\sum_{i=1}^{m}\sigma_i\mathsf{q}_i\ell(h(x_i'),y_i')\right]+\mathop{\mathsf{E}}_{S;}\left[\sup_{h\in\mathsf{H}}\sum_{i=1}^{m}\sigma_i\mathsf{q}_i\ell(h(x_i),y_i)\right]\\
& && \text{(by the sub-additivity of the supremum operator)}\\
&= 2\mathop{\mathsf{E}}_{S;}\left[\sup_{h\in\mathsf{H}}\sum_{i=1}^{m}\sigma_i\mathsf{q}_i\ell(h(x_i),y_i)\right]=2\mathsf{R}_{\mathsf{q}}(\ell\circ\mathsf{H}).
\end{aligned}$$

Now, for any $h\in\mathsf{H}$, we have

$$\mathcal{L}(\mathsf{D}_{T+1},h)-\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t\mathcal{L}(\mathsf{D}_t,h)=\mathcal{L}(\mathsf{D}_{T+1},h)-\mathcal{L}\left(\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t\mathsf{D}_t,h\right)\le\mathsf{dis}\left(\mathsf{D}_{T+1},\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t\mathsf{D}_t\right).$$

When q is a distribution, we have $\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t=1$ and

$$\begin{aligned}
\mathsf{dis}\left(\mathsf{D}_{T+1},\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t\mathsf{D}_t\right) &= \max_{h\in\mathsf{H}}\left\{\mathcal{L}(\mathsf{D}_{T+1},h)-\mathcal{L}\left(\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t\mathsf{D}_t,h\right)\right\}\\
&= \max_{h\in\mathsf{H}}\left\{\mathcal{L}(\mathsf{D}_{T+1},h)-\sum_{t=1}^{T+1}\overline{\mathsf{q}}_t\mathcal{L}(\mathsf{D}_t,h)\right\}\\
&= \max_{h\in\mathsf{H}}\left\{\sum_{t=1}^{T}\overline{\mathsf{q}}_t[\mathcal{L}(\mathsf{D}_{T+1},h)-\mathcal{L}(\mathsf{D}_t,h)]\right\}\\
&\le \sum_{t=1}^{T}\overline{\mathsf{q}}_t\max_{h\in\mathsf{H}}\{[\mathcal{L}(\mathsf{D}_{T+1},h)-\mathcal{L}(\mathsf{D}_t,h)]\}\\
&= \sum_{t=1}^{T}\overline{\mathsf{q}}_t\mathsf{dis}(\mathsf{D}_{T+1},\mathsf{D}_t).
\end{aligned}$$

This completes the proof. □

The following result shows that the bound is tight as a function of the weighted-discrepancy term.

Theorem 3. *Fix a distribution* q *in* $\quad_m$*. Then, for any* $\epsilon > 0$*, there exists* $h \in \mathsf{H}$ *such that, for any* $\delta > 0$*, the following lower bound holds with probability at least* $1 - \delta$ *over the choice of a sample* S *drawn from* $\mathsf{D}_1^{m_1} \otimes \cdots \otimes \mathsf{D}_{T+1}^{m_{T+1}}$*:*

$$\mathcal{L}(\mathsf{D}_{T+1}, h) \geq \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\right) - 2\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) - \|\mathsf{q}\|_2 \sqrt{\frac{\log \frac{1}{\delta}}{2}} - \epsilon.$$

In particular, for $\|\mathsf{q}\|_2, \mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) \in O(\frac{1}{\sqrt{m}})$*, we have:*

$$\mathcal{L}(\mathsf{D}_{T+1}, h) \geq \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\right) - \quad \left(\frac{1}{\sqrt{m}}\right).$$

Proof. Let $\mathcal{L}(\mathsf{q}, h)$ denote $\sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i)$. By definition of discrepancy as a supremum, for any $\epsilon > 0$, there exists $h \in \mathsf{H}$ such that $\mathcal{L}(\mathsf{D}_{T+1}, h) - \mathcal{L}\big(\sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t, h\big) \geq \mathsf{dis}\big(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\big) - \epsilon$. For that h, we have

$$\mathcal{L}(\mathsf{D}_{T+1}, h) - \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\right) - \mathcal{L}(\mathsf{q}, h) \geq \mathcal{L}\left(\sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t, h\right) - \mathcal{L}(\mathsf{q}, h) - \epsilon = \mathop{\mathsf{E}}_{S}[\mathcal{L}_S(\mathsf{q}, h)] - \mathcal{L}(\mathsf{q}, h) - \epsilon.$$

By McDiarmid's inequality, with probability at least $1 - \delta$, we have $\mathsf{E}[\mathcal{L}(\mathsf{q}, h)] - \mathcal{L}(\mathsf{q}, h) \geq -2\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) - \|\mathsf{q}\|_2 \sqrt{\frac{\log \frac{1}{\delta}}{2}}$. Thus, we have:

$$\mathcal{L}(\mathsf{D}_{T+1}, h) - \mathcal{L}(\mathsf{q}, h) - \overline{\mathsf{q}}\mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{Q}) \geq -2\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) - \|\mathsf{q}\|_2 \sqrt{\frac{\log \frac{1}{\delta}}{2}} - \epsilon.$$

The last inequality follows directly by using the assumptions and Lemma 1, see below. □

Lemma 1. *Fix a distribution* q *over* $[m]$*. Then, the following holds for the* q*-weighted Rademacher complexity:*

$$\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) \leq \|\mathsf{q}\|_\infty m \, \mathsf{R}_m(\ell \circ \mathsf{H}).$$

Proof. The result follows immediately Talagrand's contraction lemma, by the $\|\mathsf{q}\|_\infty$-Lipschitzness of each function $x \mapsto \mathsf{q}_i x$. □

Note that the bound is tight since for q uniform, we have $\|\mathsf{q}\|_\infty = \frac{1}{m}$ and $\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) = \mathsf{R}_m(\ell \circ \mathsf{H})$.

Theorem 2. *For any* $\delta > 0$*, with probability at least* $1 - \delta$ *over the choice of a sample* S *drawn from* $\mathsf{D}_1^{m_1} \otimes \cdots \otimes \mathsf{D}_{T+1}^{m_{T+1}}$*, the following holds for all* $h \in \mathsf{H}$ *and* $\mathsf{q} \in \big\{\mathsf{q} : 0 \leq \|\mathsf{q} - \mathsf{p}^0\|_1 < 1\big\}$*:*

$$\begin{aligned}\mathcal{L}(\mathsf{D}_{T+1}, h) \leq \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\right) + \mathsf{dis}(\mathsf{q}, \mathsf{p}^0) + 2\mathsf{R}_{\mathsf{q}}(\ell \circ \mathsf{H}) \\ + 5\|\mathsf{q} - \mathsf{p}^0\|_1 + \big[\|\mathsf{q}\|_2 + 2\|\mathsf{q} - \mathsf{p}^0\|_1\big]\left[\sqrt{\log \log_2 \tfrac{2}{1 - \|\mathsf{q} - \mathsf{p}^0\|_1}} + \sqrt{\tfrac{\log \frac{2}{\delta}}{2}}\right].\end{aligned}$$

Proof. Consider two sequences $(\epsilon_k)_{k \geq 0}$ and $(\mathsf{q}^k)_{k \geq 0}$. By Theorem 1, for any fixed $k \geq 0$, we have:

$$\mathsf{P}\left[\mathcal{L}(\mathsf{D}_{T+1}, h) > \sum_{i=1}^{m} \mathsf{q}_i^k \ell(h(x_i), y_i) + \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t^k \mathsf{D}_t\right) + 2\mathsf{R}_{\mathsf{q}^k}(\ell \circ \mathsf{H}) + \frac{\|\mathsf{q}^k\|_2}{\sqrt{2}} \epsilon_k\right] \leq e^{-\frac{2}{k}}.$$

Choose $\epsilon_k = \epsilon + \sqrt{2 \log(k+1)}$. Then, by the union bound, we can write:

$$\begin{aligned}\mathsf{P}\Bigg[\exists k \geq 1 : \mathcal{L}(\mathsf{D}_{T+1}, h) > \sum_{i=1}^{m} \mathsf{q}_i^k \ell(h(x_i), y_i) + \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t^k \mathsf{D}_t\right) + 2\mathsf{R}_{\mathsf{q}^k}(\ell \circ \mathsf{H}) + \frac{\|\mathsf{q}^k\|_2}{\sqrt{2}} \epsilon_k\Bigg] \\ \leq \sum_{k=0}^{+\infty} e^{-\frac{2}{k}} \leq \sum_{k=0}^{+\infty} e^{-\frac{2}{2} - \log((k+1)^2)} = e^{-\frac{2}{2}} \sum_{k=1}^{+\infty} \frac{1}{k^2} = \frac{\pi^2}{6} e^{-\frac{2}{2}} \leq 2e^{-\frac{2}{2}}. \quad (4)\end{aligned}$$

We can choose q^k such that $\|q^k - p^0\|_1 = 1 - \frac{1}{2^k}$. Then, for any $q \in \{q : 0 \leq \|q - p^0\|_1 < 1\}$, there exists $k \geq 0$ such that $\|q^k - p^0\|_1 \leq \|q - p^0\|_1 < \|q^{k+1} - p^0\|_1$ and thus such that

$$\begin{aligned}\sqrt{2\log(k+1)} = \sqrt{2\log\log_2 \frac{1}{1 - \|q^{k+1} - p^0\|_1}} &= \sqrt{2\log\log_2 \frac{2}{1 - \|q^k - p^0\|_1}} \\ &\leq \sqrt{2\log\log_2 \frac{2}{1 - \|q - p^0\|_1}}.\end{aligned}$$

Furthermore, for that k, the following inequalities hold:

$$\begin{aligned}\sum_{i=1}^m q_i^k \ell(h(x_i), y_i) &\leq \sum_{i=1}^m q_i \ell(h(x_i), y_i) + \mathrm{dis}(q^k, q) \\ &\leq \sum_{i=1}^m q_i \ell(h(x_i), y_i) + \mathrm{dis}(q^k, p^0) + \mathrm{dis}(p^0, q) \\ &\leq \sum_{i=1}^m q_i \ell(h(x_i), y_i) + \|q^k - p^0\|_1 + \mathrm{dis}(q, p^0) \\ &\leq \sum_{i=1}^m q_i \ell(h(x_i), y_i) + \|q - p^0\|_1 + \mathrm{dis}(q, p^0), \\ \mathrm{dis}\left(D_{T+1}, \sum_{t=1}^{T+1} \overline{q}_t^k D_t\right) &\leq \mathrm{dis}\left(D_{T+1}, \sum_{t=1}^{T+1} \overline{q}_t D_t\right) + \|q_t^k - q_t\|_1 \\ &\leq \mathrm{dis}\left(D_{T+1}, \sum_{t=1}^{T+1} \overline{q}_t D_t\right) + \|q^k - p^0\|_1 + \|p^0 - q\|_1 \\ &\leq \mathrm{dis}\left(D_{T+1}, \sum_{t=1}^{T+1} \overline{q}_t D_t\right) + 2\|p^0 - q\|_1, \\ R_{q^k}(\ell \circ H) &\leq R_q(\ell \circ H) + \|q^k - q\|_1 \leq R_q(\ell \circ H) + 2\|q - p^0\|_1, \\ \text{and} \qquad \|q^k\|_2 &\leq \|q\|_2 + \|q^k - q\|_2 \leq \|q\|_2 + \|q^k - q\|_1 \leq \|q\|_2 + 2\|q - p^0\|_1.\end{aligned}$$

Plugging in these inequalities in (4) concludes the proof. □

Corollary 1. *For any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S drawn from $D_1^{m_1} \otimes \cdots \otimes D_{T+1}^{m_{T+1}}$, the following holds for all $h \in H$ and $q \in \{q : 0 \leq \|q - p^0\|_1 < 1\}$:*

$$\begin{aligned}\mathcal{L}(D_{T+1}, h) \leq \sum_{i=1}^m q_i \ell(h(x_i), y_i) + \sum_{t=1}^T \overline{q}_t \mathrm{dis}(D_{T+1}, D_t) + \mathrm{dis}(q, p^0) + 2R_q(\ell \circ H) + 6\|q - p^0\|_1 \\ + \left[\|q\|_2 + 2\|q - p^0\|_1\right]\left[\sqrt{\log\log_2 \tfrac{2}{1 - \|q - p^0\|_1}} + \sqrt{\tfrac{\log \frac{2}{\delta}}{2}}\right].\end{aligned}$$

Proof. By definition of the discrepancy, we can write:

$$\begin{aligned}\mathrm{dis}\left(D_{T+1}, \sum_{t=1}^{T+1} \overline{q}_t D_t\right) &= \mathrm{dis}\left(\left[(1 - q_{T+1}) + \sum_{t=1}^T \overline{q}_t\right] D_{T+1}, \sum_{t=1}^T \overline{q}_t D_t\right) \\ &\leq \left(\sum_{t=1}^T \overline{q}_t D_{T+1}, \sum_{t=1}^T \overline{q}_t D_t\right) + |1 - \|q\|_1| \\ &= \sum_{t=1}^T \overline{q}_t (D_{T+1}, D_t) + |\|p\|_1 - \|q\|_1| \\ &= \sum_{t=1}^T \overline{q}_t (D_{T+1}, D_t) + |\|p - q\|_1|.\end{aligned}$$

Combining this inequality with the bound of Theorem 2 completes the proof. □

C DC-Programming

We can reduce the optimization problem of DRIFT to an instance of DC-programming (difference of convex) by writing the objective as a difference. Note that for any non-negative and convex function f, f^2 is convex: for all $(x, x') \in \mathsf{X}^2$ and $\alpha \in [0,1]$, by the convexity of f and the monotonicity of $x \mapsto x^2$ on R_+, we can write

$$f^2(\alpha x + (1-\alpha)x') \leq \left[\alpha f(x) + (1-\alpha) f(x')\right]^2 \leq \alpha f^2(x) + (1-\alpha) f^2(x'),$$

where the last inequality holds by the convexity of $x \mapsto x^2$. Thus, we can rewrite the non-jointly convex terms of the objective as the following DC-decompositions:

$$\mathsf{q}_i \ell(h(x_i), y_i) = \frac{1}{2}\Big[\big[\mathsf{q}_i + u\big]^2 - \big[\mathsf{q}_i^2 + u^2\big]\Big] \qquad \|\mathsf{q}\|_\infty \|h\|^2 = \frac{1}{2}\Big[\big[\|\mathsf{q}\|_\infty + \|h\|^2\big]^2 - \big[\|\mathsf{q}\|_\infty^2 + \|h\|^2\big]\Big],$$

where $u = \ell(h(x_i), y_i)$. We can then apply the DCA algorithm of Tao and An (1998), (see also Tao and An (1997)), which in our differentiable case coincides with the CCCP algorithm of Yuille and Rangarajan (2003) further analyzed by Sriperumbudur et al. (2007). The DCA algorithm does indeed guarantee convergence.

D Unsupervised or Weakly Supervised Drifting

The analysis of Section 3 can also be used to derive finer guarantees for weakly supervised drifting, the scenario where the learner has access to few or no labeled points from the target segment. In this section, we analyze the case where points from the target segment D_{T+1} are all unlabeled. The extension of our analysis to the case where a small fraction of the points from that segment are labeled is straightforward. The new guarantees lead to the design of better algorithms for weakly supervised drifting.

D.1 Unsupervised or Weakly Supervised Drifting Generalization Bounds

For convenience, we will use an alternative notation here for the weights on the first T source samples and the sample S_{T+1} from the target sample: we will denote by $\mathsf{q} \in [0,1]^{n_{T+1}}$ the weight vector for the source samples and by $\mathsf{q}' \in [0,1]^{m_{T+1}}$ the weight vector for the target (unlabeled) samples. Since the labels are not available for points in S_{T+1}, we upper-bound the reweighted empirical loss in terms of a p-weighted empirical loss and a weighted discrepancy term, for any weight vector $\mathsf{p} \in [0,1]^{n_{T+1}}$:

$$\sum_{i=1}^{n_{T+1}} \mathsf{q}_i \ell(h(x_i), y_i) + \sum_{i=n_{T+1}+1}^{m} \mathsf{q}'_i \ell(h(x_i), y_i) \leq \sum_{i=1}^{n_{T+1}} (\mathsf{q}_i + \mathsf{p}_i) \ell(h(x_i), y_i) + \mathsf{dis}(\mathsf{q}', \mathsf{p}). \tag{5}$$

This yields immediately the following theorem, using Theorem 1.

Theorem 4. *Fix the vectors $\mathsf{q} \in [0,1]^{n_{T+1}}$ and $\mathsf{q}' \in [0,1]^{m_{T+1}}$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S drawn from $\mathsf{D}_1^{m_1} \otimes \cdots \otimes \mathsf{D}_{T+1}^{m_{T+1}}$, the following holds for all $\mathsf{p} \in [0,1]^{n_{T+1}}$ and $h \in \mathsf{H}$:*

$$\mathcal{L}(\mathsf{D}_{T+1}, h) \leq \sum_{i=1}^{n_{T+1}} (\mathsf{q}_i + \mathsf{p}_i) \ell(h(x_i), y_i) + \mathsf{dis}(\mathsf{q}', \mathsf{p}) + \mathsf{dis}\Big(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\Big) + 2\mathsf{R}_{(\mathsf{q};\mathsf{q}')}(\ell \circ \mathsf{H}) + \sqrt{\frac{(\|\mathsf{q}\|_2^2 + \|\mathsf{q}'\|_2^2) \log \frac{1}{\delta}}{2}}.$$

Here we denote by $(\mathsf{q}, \mathsf{q}')$ the vector in $[0,1]^m$ obtained by appending q' to q. This learning bound can be extended to hold uniformly over $\big\{(\mathsf{q}, \mathsf{q}') \in [0,1]^m : 0 < \|(\mathsf{q}, \mathsf{q}') - \mathsf{p}^0\|_1 < 1\big\}$ and all p in $[0,1]^{n_{T+1}}$, where p^0 is a reference (or ideal) reweighting choice over the m points (see Theorem 5 and Corollary 2 in Appendix D).

Here, both p and q' can be chosen to make the weighted-discrepancy term $\mathsf{dis}(\mathsf{q}', \mathsf{p})$ smaller. Several of the comments on Theorem 1 similarly apply here. When we do not have labels from S_{T+1}, the discrepancy terms must be upper-bounded with unlabeled discrepancies, using only unlabeled data from D_{T+1}. A detailed analysis is presented in Appendix D.4

D.2 WDRIFT Algorithm

The analysis of the previous section suggests seeking $h \in \mathsf{H}$ and q and p in $[0,1]^{n_{T+1}}$ and q' in $[0,1]^{m_{T+1}}$ to minimize the bound of Theorem 5 or that of Corollary 2. As in Section 4, assume that H is a subset of a normed vector space and that the Rademacher complexity term can be bounded in terms of an upper bound on the norm squared $\|h\|^2$. Then, the optimization problem corresponding to Corollary 2 can be written as follows:

$$\begin{aligned}\min_{\substack{h\in\mathsf{H};\mathsf{q};\mathsf{p}\in[0;1]^{n_{T+1}}\\ \mathsf{q}'\in[0;1]^{m_{T+1}}}} & \sum_{i=1}^{n_{T+1}} (\mathsf{q}_i + \mathsf{p}_i)\,\ell(h(x_i), y_i) \\ & + \sum_{t=1}^{T} \overline{\mathsf{q}}_t \overline{\mathsf{dis}}(\mathsf{D}_{T+1}, \mathsf{D}_t) + \overline{\mathsf{dis}}(\mathsf{q}', \mathsf{p}) + \overline{\mathsf{dis}}((\mathsf{q}, \mathsf{q}'), \mathsf{p}^0) \\ & + \lambda_\infty \|(\mathsf{q}, \mathsf{q}')\|_\infty \|h\|^2 + \lambda_1 \|(\mathsf{q}, \mathsf{q}') - \mathsf{p}^0\|_1 + \lambda_2 (\|\mathsf{q}\|_2^2 + \|\mathsf{q}'\|_2^2),\end{aligned} \tag{6}$$

where λ_1, λ_2 and λ_∞ are non-negative hyperparameters and where we used the shorthand. We will refer by WDRIFT to the algorithm seeking to minimize this objective. We are omitting subscripts to simplify the presentation but, as discussed in the previous section, the unlabeled discrepancies in the optimization problem may be local unlabeled discrepancies, which are finer quantities. Here too, a natural choice for p^0 is the uniform distribution over the input points of S'. In practice, there may be better choices motivated by specific applications.

Our comments and analysis of the DRIFT optimization (Section 4) apply similarly here. In particular, the problem can be similarly cast as an alternate minimization or a DC-programming problem. The unlabeled discrepancy terms can be accurately estimated using the samples available.

D.3 Theorems and Proofs

Let $(\mathsf{q}, \mathsf{q}')$ denote the vector in $[0,1]^m$ formed by appending q' to q. The learning bound of Theorem 4 can be extended to hold uniformly over all p in $[0,1]^{n_{T+1}}$ and $(\mathsf{q}, \mathsf{q}')$ in $\left\{(\mathsf{q}, \mathsf{q}') \in [0,1]^m : 0 < \|(\mathsf{q}, \mathsf{q}') - \mathsf{p}^0\|_1 < 1\right\}$, where p^0 is a reference (or ideal) reweighting choice over the m points.

Theorem 5. *For any $\delta > 0$, with probability at least $1-\delta$ over the draw of a sample S from $\mathsf{D}_1^{m_1} \otimes \cdots \otimes \mathsf{D}_{T+1}^{m_{T+1}}$, the following holds for all $h \in \mathsf{H}$, $\mathsf{q} \in \left\{\mathsf{q} : 0 \le \|(\mathsf{q}, \mathsf{q}') - \mathsf{p}^0\|_1 < 1\right\}$ and all $\mathsf{p} \in [0,1]^{n_{T+1}}$:*

$$\begin{aligned}\mathcal{L}(\mathsf{D}_{T+1}, h) \le & \sum_{i=1}^{n_{T+1}} (\mathsf{q}_i + \mathsf{p}_i)\ell(h(x_i), y_i) + \mathsf{dis}(\mathsf{q}', \mathsf{p}) + \mathsf{dis}\left(\mathsf{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathsf{D}_t\right) \\ & + \mathsf{dis}((\mathsf{q}, \mathsf{q}'), \mathsf{p}^0) + 2\mathsf{R}_{(\mathsf{q};\mathsf{q}')}(\ell \circ \mathsf{H}) + 5\|(\mathsf{q}, \mathsf{q}') - \mathsf{p}^0\|_1 \\ & + \left[\|\mathsf{q}\|_2 + 2\|(\mathsf{q}, \mathsf{q}') - \mathsf{p}^0\|_1\right]\left[\sqrt{\log \log_2 \frac{2}{1 - \|(\mathsf{q};\mathsf{q}') - \mathsf{p}^0\|_1}} + \sqrt{\frac{\log \frac{2}{\delta}}{2}}\right].\end{aligned}$$

Proof. The proof follows immediately by applying inequality (5), which holds for all $\mathsf{p} \in [0,1]^{n_{T+1}}$, to the bound of Theorem 2. □

Corollary 2. *For any $\delta > 0$, with probability at least $1-\delta$ over the draw of a sample S from $\mathsf{D}_1^{m_1} \otimes \cdots \otimes \mathsf{D}_{T+1}^{m_{T+1}}$, the following holds for all $h \in \mathsf{H}$, $\mathsf{q} \in \left\{\mathsf{q} : 0 \le \|(\mathsf{q}, \mathsf{q}') - \mathsf{p}^0\|_1 < 1\right\}$ and all $\mathsf{p} \in [0,1]^{n_{T+1}}$:*

$$\begin{aligned}\mathcal{L}(\mathsf{D}_{T+1}, h) \le & \sum_{i=1}^{m} (\mathsf{q}_i + \mathsf{p}_i)\ell(h(x_i), y_i) + \mathsf{dis}(\mathsf{q}', \mathsf{p}) + \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t) \\ & + \mathsf{dis}((\mathsf{q}, \mathsf{q}'), \mathsf{p}^0) + 2\mathsf{R}_{(\mathsf{q};\mathsf{q}')}(\ell \circ \mathsf{H}) + 6\|(\mathsf{q}, \mathsf{q}') - \mathsf{p}^0\|_1 \\ & + \left[\|\mathsf{q}\|_2 + 2\|(\mathsf{q}, \mathsf{q}') - \mathsf{p}^0\|_1\right]\left[\sqrt{\log \log_2 \frac{2}{1 - \|(\mathsf{q};\mathsf{q}') - \mathsf{p}^0\|_1}} + \sqrt{\frac{\log \frac{2}{\delta}}{2}}\right].\end{aligned}$$

Proof. The result follows Theorem 5 and the application of the upper bound used in the proof of Corollary 1. □

D.4 Labeled Discrepancy Upper Bounds

The definition of labeled discrepancy naturally requires labels from D_i and D_j. When we do not have access to these labels, the *unlabeled discrepancy* $\overline{\mathsf{dis}}(\mathsf{D}_i, \mathsf{D}_j)$ is appropriate, defined as:

$$\overline{\mathsf{dis}}(\mathsf{D}_i, \mathsf{D}_j) = \sup_{h, h' \in \mathsf{H}} \mathop{\mathsf{E}}_{x \sim \mathsf{D}_{iX}} [\ell(h(x), h'(x))] - \mathop{\mathsf{E}}_{x \sim \mathsf{D}_{jX}} [\ell(h(x), h'(x))]. \tag{7}$$

We define $\overline{\mathsf{Dis}}(\mathsf{D}_i, \mathsf{D}_j)$ as the version of unlabeled discrepancy with absolute values. When H has a favorable Rademacher complexity such as a finite VC-dimension, the unlabeled discrepancy can be accurately estimated with finite (unlabeled) samples from the marginal distributions D_{iX} and D_{jX} (Mansour et al., 2009a). The finer notion of *local labeled discrepancy* for some suitably chosen subsets H_1 and H_2 of H is defined by:

$$\overline{\mathsf{dis}}_{\mathsf{H}_1 \times \mathsf{H}_2}(\mathsf{D}_i, \mathsf{D}_j) = \sup_{(h, h') \in \mathsf{H}_1 \times \mathsf{H}_2} \mathop{\mathsf{E}}_{x \sim \mathsf{D}_{iX}} [\ell(h(x), h'(x))] - \mathop{\mathsf{E}}_{x \sim \mathsf{D}_{jX}} [\ell(h(x), h'(x))]. \tag{8}$$

Local discrepancy (Cortes et al., 2019b) is a more favorable quantity because it is defined by a supremum over smaller sets. As an example, when a small but sufficient amount of labeled data is available from the target, we can use, instead of the hypothesis space H, a ball of an appropriate radius around a classifier h_{T+1} obtained by training on that data. Other instances of local discrepancy are adopted in (Zhang et al., 2019) and (Sicilia et al., 2022), with notions based on a single supremum. However, the loss function in then ignored or reduced to a binary loss.

In the absence of labeled points from the target segment or when only few labeled points are available, we need to resort to upper bounds in terms of this unlabeled discrepancy, which can be estimated using only unlabeled data. Such upper bounds can be derived straightforwardly using previous work and analysis based on discrepancy (Cortes and Mohri, 2014; Cortes et al., 2019b). We will briefly discuss here such upper bounds.

For the squared loss, for any hypothesis $h_0 \in \mathsf{H}$, the following upper bound based on a local discrepancy can be derived:

$$\mathsf{dis}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) \leq \overline{\mathsf{dis}}_{\mathsf{H} \times \{h_0\}}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) + 2\delta_{\mathsf{H}; h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t),$$

where $\widehat{\mathsf{D}}_{T+1}$ and $\widehat{\mathsf{D}}_t$ denote the empirical distributions associated to D_{T+1} and D_t and where δ is defined by (Cortes and Mohri, 2014):

$$\delta_{\mathsf{H}; h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) = \sup_{h \in \mathsf{H}} \left| \mathop{\mathsf{E}}_{(x,y) \sim \widehat{\mathsf{D}}_{T+1}} [h(x)(y - h_0(x))] - \mathop{\mathsf{E}}_{(x,y) \sim \widehat{\mathsf{D}}_t} [h(x)(y - h_0(x))] \right|.$$

For a suitable choice of $h_0 \in \mathsf{H}$, the term $\delta_{\mathsf{H}; h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t)$ captures the closeness of the empirical output labels on $\widehat{\mathsf{D}}_{T+1}$ and $\widehat{\mathsf{D}}_t$. The unlabeled discrepancy term can be accurately estimated from unlabeled samples (Mansour et al., 2009a). When a relatively small labeled sample S' drawn i.i.d. from D_{T+1} is available, we can use it to select h_0 via

$$h_0 = \mathop{\mathsf{argmin}}_{h_0 \in \mathsf{H}} \delta_{\mathsf{H}; h_0}(\widehat{\mathsf{D}}_{S'; T+1}, \widehat{\mathsf{D}}_t),$$

where $\widehat{\mathsf{D}}_{S'; T+1}$ denotes the empirical distribution associated to S'. When no labeled data from the target segment is at our disposal, we cannot choose h_0 by leveraging any existing information. We can then assume that $\mathsf{min}_{h_0 \in \mathsf{H}} \delta_{\mathsf{H}; h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) \ll 1$, that is that the source labels are relatively close to the target ones based on these measures and use the standard unlabeled discrepancy:

$$\mathsf{dis}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) \leq \overline{\mathsf{dis}}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) + 2 \min_{h_0 \in \mathsf{H}} \delta_{\mathsf{H}; h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t).$$

When the covariate-shift assumption holds and the problem is separable, h_0 can be chosen so that $\delta_{\mathsf{H}; h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) = 0$. More generally, when h_0 can be chosen so that $|y - h_0(x)|$ is relatively small on both samples corresponding to $\widehat{\mathsf{D}}_{T+1}$ and $\widehat{\mathsf{D}}_t$ and the hypotheses $h \in \mathsf{H}$ are bounded by some $M > 0$, then $\delta_{\mathsf{H}; h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t)$ is relatively small.

For a μ-Lipschitz loss, similarly, the following upper bound on the labeled discrepancy can be used:

$$\mathsf{dis}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) \leq \overline{\mathsf{dis}}_{\mathsf{H} \times \{h_0\}}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) + \mu \eta_{\mathsf{H}; h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t).$$

where, for any $h_0 \in \mathsf{H}$, $\eta_{\mathsf{H};h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t)$ is defined by (Cortes et al., 2019b):

$$\eta_{\mathsf{H};h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t) = \mathop{\mathsf{E}}_{(x;y)\sim\widehat{\mathsf{D}}_{T+1}}[|y - h_o(x)|] + \mathop{\mathsf{E}}_{(x;y)\sim\widehat{\mathsf{D}}_t}[|y - h_o(x)|].$$

The Lipschitz loss labeled discrepancy $\eta_{\mathsf{H};h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t)$ is a coarser quantity than $\delta_{\mathsf{H};h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t)$. In particular, even when $\widehat{\mathsf{D}}_{T+1} = \widehat{\mathsf{D}}_t$, $\eta_{\mathsf{H};h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t)$ is not zero, as pointed out by Cortes and Mohri (2014). However, as with $\delta_{\mathsf{H};h_0}(\widehat{\mathsf{D}}_{T+1}, \widehat{\mathsf{D}}_t)$, it captures the closeness of the output labels on $\widehat{\mathsf{D}}_{T+1}$ and $\widehat{\mathsf{D}}_t$. The rest of the discussion in the case of a Lipschitz loss is similar to the squared loss case. In particular, when a relatively small labeled sample S' is available from the target segment, then we can choose h_0 as follows: $h_0 = \mathsf{argmin}_{h_0\in\mathsf{H}}\, \eta_{\mathsf{H};h_0}(\widehat{\mathsf{D}}_{S';T+1}, \widehat{\mathsf{D}}_t)$.

E Comparison of DRIFT and a Naive-DRIFT Solution

A naive baseline to compare the DRIFT algorithm to is that of simply combining D_1 to D_T to form a single distribution D_1, and then applying the DRIFT algorithm with the same target D_{T+1}. We will refer to this method by naive-DRIFT, since ignores the differences between the first T distributions. Here, we present a simple case to illustrate how DRIFT can outperform this baseline.

The DRIFT algorithm introduced in Section 4 optimizes the following objective

$$\min_{h\in\mathsf{H};\mathsf{q}\in[0;1]^m} \sum_{i=1}^m \mathsf{q}_i[\ell(h(x_i), y_i)] + \sum_{t=1}^T \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t) + \mathsf{dis}(\mathsf{q}, \mathsf{p}^0) + \lambda_\infty \|\mathsf{q}\|_\infty \|h\|^2 + \lambda_1 \|\mathsf{q} - \mathsf{p}^0\|_1 + \lambda_2 \|\mathsf{q}\|_2^2,$$

Let there be two distributions D_1 and D_2, which are alternating up until and including D_{T+1}. Thus, we have the sequence $\mathsf{D}_1, \mathsf{D}_2, \mathsf{D}_1, \mathsf{D}_2, \ldots, \mathsf{D}_2, \mathsf{D}_1$ with $\mathsf{D}_{T+1} = \mathsf{D}_1$ and $\mathsf{dis}(\mathsf{D}_1, \mathsf{D}_2) = 1$. The only difference between the two approaches is then the term $\sum_{t=1}^T \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)$ from the optimization problem. In the naive approach of combining the T distributions, we have:

$$\sum_{t=1}^T \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t) = \overline{\mathsf{q}} \mathsf{dis}\left(\mathsf{D}_{T+1}, \frac{1}{T}\sum_{t=1}^T \mathsf{D}_t\right) = \overline{\mathsf{q}} \mathsf{dis}(\mathsf{D}_1, \frac{1}{2}(\mathsf{D}_1 + \mathsf{D}_2)) = \frac{\overline{\mathsf{q}}}{2}.$$

The last step comes from applying the following analysis. In general, we have:

$$\mathsf{dis}(\mathsf{D}_i, \mathsf{D}_j) = \max_{h\in\mathsf{H}} \mathop{\mathsf{E}}_{\substack{(x;y)\sim\\ \mathsf{D}_i}}[\ell(h(x), y)] - \mathop{\mathsf{E}}_{\substack{(x;y)\sim\\ \mathsf{D}_j}}[\ell(h(x), y)] = \max_{h\in\mathsf{H}} \sum_{(x;y)} [\mathsf{D}_i(x, y) - \mathsf{D}_j(x, y)]\ell(h(x), y).$$

In our case, we have:

$$\begin{aligned}\mathsf{dis}(\mathsf{D}_1, \frac{1}{2}(\mathsf{D}_1 + \mathsf{D}_2)) &= \max_{h\in\mathsf{H}} \sum_{(x;y)} [\mathsf{D}_1(x, y) - \frac{1}{2}(\mathsf{D}_1(x, y) + \mathsf{D}_2(x, y))]\ell(h(x), y) \\ &= \frac{1}{2}\max_{h\in\mathsf{H}} \sum_{(x;y)} [\mathsf{D}_1(x, y) - \mathsf{D}_2(x, y)]\ell(h(x), y) = \frac{1}{2}\mathsf{dis}(\mathsf{D}_1, \mathsf{D}_2) = \frac{1}{2}.\end{aligned}$$

The first two terms of the objective of the DRIFT optimization can alternatively be written as

$$\begin{aligned}&\sum_{i=1}^m \mathsf{q}_i[\ell(h(x_i), y_i)] + \sum_{t=1}^T \overline{\mathsf{q}}_t \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t) \\ &= \sum_{t=1}^T \sum_{i=n_t+1}^{n_t+m_t} \mathsf{q}_i[\ell(h(x_i), y_i) + \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)] + \sum_{i=n_{T+1}+1}^m \mathsf{q}_i[\ell(h(x_i), y_i)].\end{aligned}$$

For the naive approach, these terms simplify to

$$\begin{aligned}&\sum_{t=1}^T \sum_{i=n_t+1}^{n_t+m_t} \mathsf{q}_i[\ell(h(x_i), y_i) + \mathsf{dis}(\mathsf{D}_{T+1}, \mathsf{D}_t)] + \sum_{i=n_{T+1}+1}^m \mathsf{q}_i[\ell(h(x_i), y_i)] \\ &= \sum_{i=1}^{m-m_t} \mathsf{q}_i\left[\ell(h(x_i), y_i) + \frac{1}{2}\right] + \sum_{i=n_{T+1}+1}^m \mathsf{q}_i[\ell(h(x_i), y_i)].\end{aligned}$$

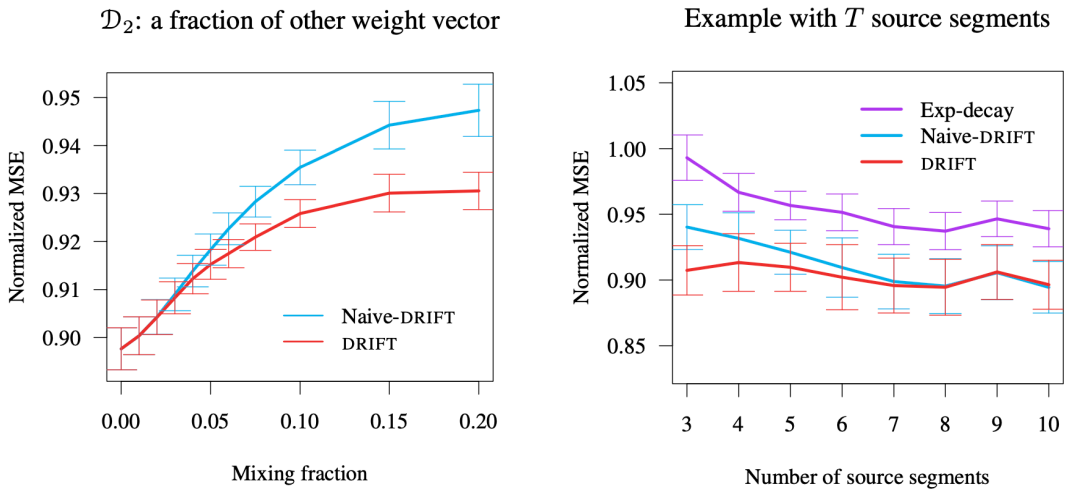

Figure 7: Left: Performance in the weight-mixing example of synthetic data with three distributions $\mathsf{D}_1 = \mathsf{D}_3 \neq \mathsf{D}_2$ as a function of increasing discrepancy. Right: Performance in the example with k source distributions.

The extra loss of $1/2$ in the objective for any example from the first T distributions forces in the naive approach q to be quite small, allocating little weight to these points. As such, the naive approach does not allow us to benefit much from the training points from the samples from D_1, while they are drawn from the same distribution as the target. In the more nuanced approach, since $\mathsf{dis}(\mathsf{D}_1, \mathsf{D}_{T+1}) = 0$ and $\sum_{i=1}^{m} \mathsf{q}_i = 1$, the algorithm can allocate significantly more weight to the samples coming from D_1, which should show an improvement over the naive approach.

F Experimental Results

We here provide more experimental data and detail of the results reported in the main paper, Section 5. Our proposed SDRIFT algorithm requires computing the discrepancy values between the source segments and the target segment. Since for the squared loss and the logistic loss over linear models, the discrepancy equals the difference of two convex terms, we approximate the discrepancy value via DC programming (Tao and An, 1997, 1998). We use a fixed learning rate of 0.01 for regression tasks and a learning rate of 0.001 for classification tasks.

F.1 Synthetic Data

Figure 7 (left) illustrates the normalized MSE for a weight mixing example. We use the same experimental setup as for the example with three distributions detailed in the main paper, but here the labels of D_2 are modified by mixing in an increasing fraction, α, of a different weight vector w_2, also randomly drawn and with unit length, such that $y_{\mathsf{D}_2} = (\alpha w_2 + (1-\alpha)\, w) \cdot x$. Again, we observe how the DRIFT algorithm can effectively make use of the data from D_2 and obtains a normalized MSE < 1 for a much larger range of label corruption than that of Naive-DRIFT.

We also compare the performance of our proposed algorithm for varying number, T, of source segments. For each $T \in \{3, 4, \ldots, 10\}$, the labels are generated as $y = w \cdot x + \mathcal{N}(0, \sigma^2)$, with $\sigma = 0.1$. Each source segment is generated in the same manner and we artificially inject a varying amount of noise within each of them. For a source segment $i \in \{1, 2, \ldots, T\}$, an $\alpha = ((T - 1 + i)/T$ fraction of the predictions are flipped. That is, for D_1, 100% of the labels are flipped. As can be seen in Figure 7(right), our proposed algorithm outperforms the baselines and its performance is unaffected across different values of T. For both Naive-DRIFT and SDRIFT the hyperparameters $\lambda_\infty, \lambda_1, \lambda_2$ were chosen via cross validation in the range $\{1e-3, 1e-2, 1e-1\} \cup \{0, 1, 2, \ldots, 10\} \cup \{0, 1000, 2000, 10000, 50000, 100000\}$. The h optimization step of alternate minimization was performed using sklearn's linear regression method (Pedregosa et al., 2011). For the q optimization we used projected gradient descent and the step size was chosen via cross validation in the range $\{1e-3, 1e-2, 1e-1\}$.

F.2 Regression Datasets

Here, we provide details on the datasets used for regression. In the final version of the paper we will provide GitHub links to all datasets.

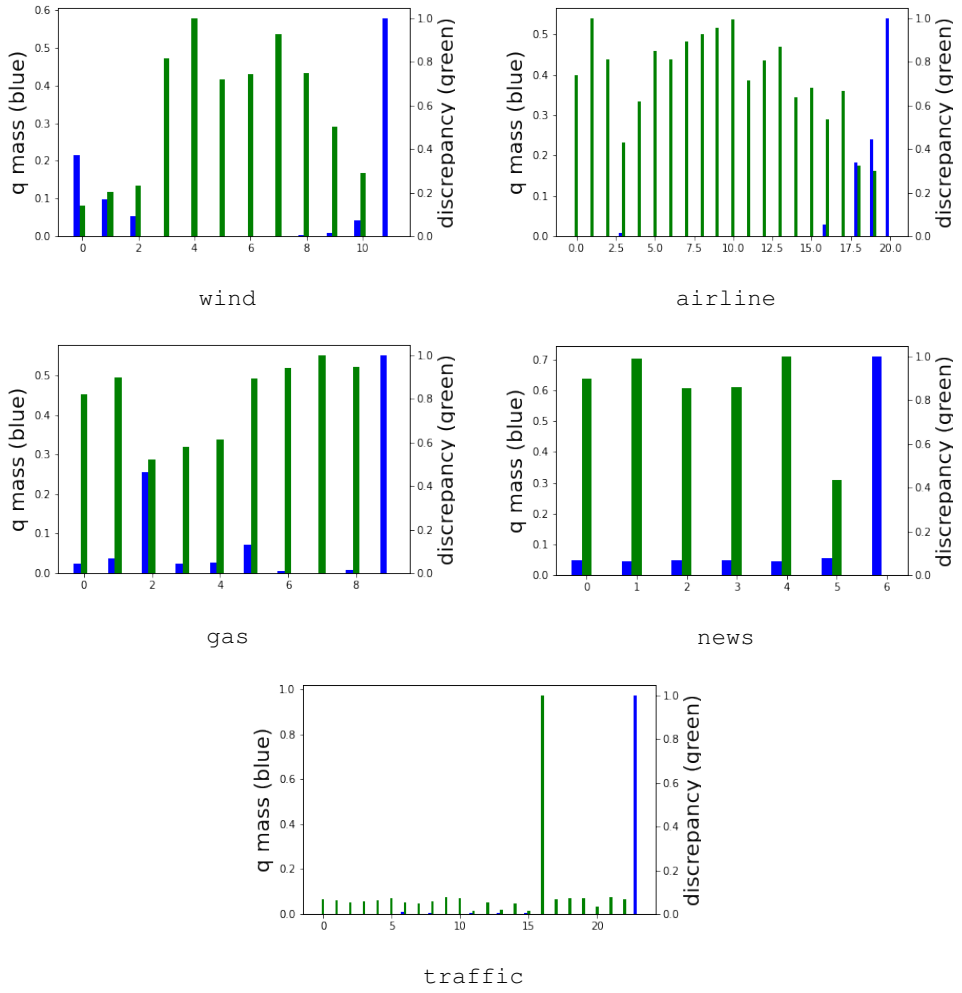

Figure 8: (Same as Figure 5 in the main paper.) A plot of the total average probability mass assigned (in blue) to each segment by the SDRIFT algorithm along side the corresponding (normalized) discrepancy values (in green).

The `wind` dataset (Haslett and Raftery, 1989) is related to wind speeds (in knots) in Ireland from 1961 to 1987. Measurements were collected from 12 meterological stations, and we chose to predict the wind speed at the "Malin Head" station using the values as the 11 other stations as features. Our 11 source segments consist of data from the first 11 months of the year, and our target is data from the month of December. Each of the source segments is of size ~500, and for the target we use a split of ~150/~200/~200 for training/validation/test.

The `airline` dataset was derived from Bifet and Ikonomovska (2009) and contains information regarding flights into Chicago O'Haire International Airport (ORD) in 2008. We use as features the arrival time, distance, whether or not the flight was diverted, and the day of the week for predicting the amount of time the flight was delayed. Our source segments are comprised from the hours of the day, and our target segment is one of the busier hours. Each of the source segments is of size 800, and for the target we have sizes 200 train/300 validation/300 test.

The `gas` dataset (Rodriguez-Lujan et al., 2014; Vergara et al., 2012; Dua and Graff, 2017) is a commonly used drift dataset with measurements from 16 chemical sensors at varying concentrations of 6 gases. The dataset has predetermined batches, and we reserved the seventh one as our target. The source batches vary in size from ~150 to ~3500, and for the target batch we have sizes ~600 train/~1000 validation/~2000 test.

The `news` dataset (Fernandes, 2015; Dua and Graff, 2017) consists of data gleaned from articles on www.mashable.com, with the goal of predicting their popularity in terms of the number of shares. Our 6 source segments consist of the 6 days of the week from Monday to Saturday and our target is data from Sunday. The weekday source segments are of size ~6000 and weekend of size ~2500, and for the target we have sizes 737 train/1000 validation/1000 test.

The `traffic` dataset from the Minnesota Department of Transportation (DOT, 2019; Dua and Graff, 2017) contains information about the weather and traffic volume on the Westbound Interstate 94, which is located between Minneapolis and St Paul. We split the data into segments by hour, and chose our target segment to be the one starting at 9am. The source segments are of size 100, and for the target we have sizes 200 train/400 validation/400 test.

To obtain standard deviations for the errors, we randomly sampled data from the target into train/validation/test 10 times.

In Figure 8, (same as Figure 5 in the main paper), we show in blue the average probability mass assigned by SDRIFT to each segment in the regression tasks. The green bars indicate the normalized discrepancy to the target segment. It is noticeable how the SDRIFT algorithm assigns more probability mass to segments of lower discrepancy.

F.3 Classification Datasets

Here, we provide details on the datasets used for classification tasks. In the final version of the paper we will provide GitHub links to all dataset.

The `STAGGER` dataset (López Lobo, 2020) is a common synthetic dataset used for concept drift detection. It contains 4 concepts, and the drifts are abrupt. The data exhibits 3 numeric features for a binary classification setting. We artificially added noise to the target (last) training sample by flipping the class for 20% of the points. The source segments are of size 10,000, and for the target we have sizes 2000 train/4000 validation/4000 test.

The `Electricity` dataset (Harries and Wales, 1999; Gama et al., 2004) is a popular dataset used for predicting the price movement (up or down compared to a 24 hour moving average) for the price of electricity in the Australian New South Wales Electricity Market. The data comes from May 1996 to December 1998, and we split it into segments of roughly two months each, with the target being the most recent one. Each of the source segments is of size ~3000, and for the target we have sizes ~400 train/~600 validation/~600 test.

The `Room` dataset (Candanedo and Feldheim, 2016; Dua and Graff, 2017) presents a binary classification problem (occupied or not) of an office room given features such as the light, temperature, humidity and CO2 measurements. Our segments consisted of one for each of the 24 hours of the day, and our target was the data from the 8am hour, which is occupied about 10% of the time (not the busiest, but nevertheless sometimes occupied unlike hours in the night-time). Each of the source segments is of size ~100, and for the target we have sizes ~100 train/~100 validation/~100 test.

The `Adult Income` dataset (Dua and Graff, 2017) is a popular dataset for predicting whether or not the income of an adult is greater than $50,000 from features such as their education and sex. Our source segments came from 15 of the 16 specified education levels, and our target was that of adults who had only completed 10th grade of high school. The source batches vary in size from ~100 to ~8000, and for the target batch we have sizes ~200 train/~400 validation/~400 test.

Similar to the regression datasets, to obtain standard deviations for the accuracies, we randomly sampled data from the target into train/validation/test 10 times.

F.4 Hyperparameters for Real-World Data

SDRIFT. The hyperparameters for SDRIFT were chosen via cross validation in the same range as the one used for synthetic data. For the h minimization step of the SDRIFT algorithm we used sklearn's logistic regression method (Pedregosa et al., 2011).

Baselines. For the exponential weighting heuristic the base value was chosen via cross validation in the range $\{1, 2, \ldots, 10\}$. For both discrepancy minimization (DM) (Cortes and Mohri, 2014) and Kernel Mean Matching (KMM) (Huang et al., 2006) a linear kernel was used. The DM algorithm was implemented via projected gradient descent and the learning rate was chosen via cross validation in the range $\{1e-3, 1e-2, 1e-1\}$. For the algorithm of Mohri and Muñoz Medina (2012) we used online gradient descent for regression tasks and the perceptron algorithm for the classification settings. The learning rates for online gradient descent and the second stage weight optimization were chosen via cross validation in the range $\{1e-3, 1e-2, 1e-1\}$. To run the BSTS algorithm (Scott and Varian, 2014) we used the CausalImpact python library (Brodersen et al., 2014) and the algorithm was run with the default parameters. For computational tractability, we sample 100 random points from each segment to form the time series data that was fed to the algorithm. For the MDAN algorithm, we use the code provided by the authors, and the μ hyperparameter was chosen in the range $\{1e-5, 1e-4, \ldots 1e2\}$. We report the best result from running the soft-max and hard-max version. For the DARN algorithm, we use the code provided by the authors, and we perform a grid search over $\{1e-3, 1e-1, 1e1\}$ for μ and $\{1, 10\}$ for γ.

F.5 Pseudocode for the Alternate Minimization Procedure

In Figure 9 we provide the algorithm description of our alternate minimization procedure for solving the batch distribution drift problem.

Input: Samples $\{(x_1, y_1), \ldots (x_m, y_m)\}$, tolerance τ, distribution p_0, max iterations N, hyperparameters $\lambda_\infty, \lambda_1, \lambda_2$, discrepancy estimates $\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_T$.

1. Initialize q_0 to be the uniform distribution over $[m]$.
2. Let $\mathcal{OPT}(q, h) = \sum_{i=1}^{m} \mathsf{q}_i[\ell(h(x_i), y_i)] + \sum_{t=1}^{T} \overline{\mathsf{q}}_t \hat{d}_t + \lambda_\infty \|\mathsf{q}\|_\infty \|h\|^2 + \lambda_1 \|\mathsf{q} - \mathsf{p}^0\|_1 + \lambda_2 \|\mathsf{q}\|_2^2$
3. Initialize $h_0 = \operatorname{argmin}_{h \in H} \mathcal{OPT}(q_0, h)$.
4. For $j = 1, \ldots N$,
   - Set curr_obj_val = $\mathcal{OPT}(q_{j-1}, h_{j-1})$.
   - Compute $q_j = \operatorname{argmin}_{q \in \Delta_m} \mathcal{OPT}(q, h_{j-1})$.
   - Compute $h_j = \operatorname{argmin}_{h \in H} \mathcal{OPT}(q_j, h)$.
   - Set new_obj_val = $\mathcal{OPT}(q_j, h_j)$.
   - If |curr_obj_val – new_obj_val| $\leq \tau$, return q_j, h_j
5. Print: *AM did not converge in T iterations.* Return q_N, h_N.

Figure 9: Alternate minimization procedure for weights and hypothesis estimation.