Theory and Algorithm for Batch Distribution Drift Problems

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study a problem of gradual *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. Additionally, we report the results of extensive experiments demonstrating the benefits of our drifting algorithm, including comparisons with natural baselines.

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, 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, fraud detection, network intrusion detection, sales prediction, and many others.

In such tasks, the distribution gradually changes over time. 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 here study 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 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 both the online and offline setting, see Appendix A.

This paper studies a frequent batch scenario of distribution drift where distribution time segments are known to the learner and one can 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. If the segments are not known apriori, we provide in Appendix E an algorithm for detecting the segments.

Our analysis and algorithm make use of the discrepancy, as in (Mohri and Muñoz, 2012). However, our discrepancy-based generalization bounds are novel and distinct. Also, that study 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. In contrast, our algorithm DRIFT simultaneously learns both the weights and the hypothesis. Our analysis and algorithm also hold for general hypothesis sets

Submitted to the Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

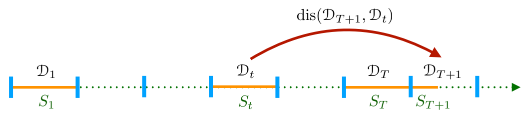

Figure 1: Illustration of the learning scenario: distributions $\mathcal{D}_t$, samples $S_t \sim \mathcal{D}_t^{m_t}$, and discrepancies $\mathrm{dis}(\mathcal{D}_{T+1}, \mathcal{D}_t)$, where $|S_t| = m_t$ and $\sum_{s=1}^{T+1} m_s = m$.

and are expressed in terms of a weighted Rademacher complexity of the hypothesis set used. In the following, we present our new bounds, our DRIFT algorithm and extensive experimental results.

2 Learning scenario

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

We study the following *distribution drift* problem. Let $\mathcal{D}_1, \ldots, \mathcal{D}_{T+1}$ be $(T+1)$ distributions over $\mathcal{X} \times \mathcal{Y}$. 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 $\mathcal{D}_t$, $t \in [T+1]$, with $n_t = \sum_{s=1}^{t-1} m_s$, see Figure 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, with $m_{T+1} \ll \sum_{t=1}^{T} m_t$. For any t, will denote by $\widehat{\mathcal{D}}_t$ the empirical distribution defined by the sample S_t and will denote by $\mathcal{D}_{t,X}$ the margin distribution of $\mathcal{D}_t$ on $\mathcal{X}$. The goal is to use these samples to learn a hypothesis h for the target distribution $\mathcal{D}_{T+1}$ with small expected loss $\mathcal{L}(\mathcal{D}_{T+1}, h)$. Of course, one could use just the sample S_{T+1} available from the target to train a predictor. However, when the distributions $\mathcal{D}_t$, $t \in [T]$, are somewhat similar to the target distribution, using the samples S_t, $t \in [T]$, may help select a more accurate predictor.

An appropriate measure of the distance between distributions is necessary to tackle the distribution drifting problem. Mohri and Muñoz (2012) argued that a suitable measure is that of *discrepancy*, previously used in the context of adaptation (Kifer et al., 2004; Ben-David et al., 2006; Mansour et al., 2009; Cortes and Mohri, 2014; Cortes et al., 2019b), as it takes into account both the loss function and the hypothesis set. It can also be estimated from a finite sample and upper bounded by other divergence measures such as the relative entropy and total variation (Mansour et al., 2021).

We call $\mathrm{dis}(\mathcal{D}_i, \mathcal{D}_j)$ the *labeled discrepancy* between $\mathcal{D}_i$ and $\mathcal{D}_j$:

$$\mathrm{dis}(\mathcal{D}_i, \mathcal{D}_j) = \sup_{h \in \mathcal{H}} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_i}[\ell(h(x), y)] - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_j}[\ell(h(x), y)]. \tag{1}$$

In all the definitions above, we also allow $\mathcal{D}_i$ and $\mathcal{D}_j$ to be finite signed measures over $\mathcal{X} \times \mathcal{Y}$, thus the weights may not sum to one. In addition, we (abusively) allow distributions over sample indices: given a sample S and a distribution q over its $[m]$ indices, we define the discrepancy $\mathrm{dis}(\widehat{\mathcal{D}}, \mathsf{q})$

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

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. 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 $\mathfrak{R}_{\mathsf{q}}(\ell \circ \mathcal{H})$ the q-weighted Rademacher complexity, an extension of Rademacher complexity taking into account the weights q:

$$\mathfrak{R}_{\mathsf{q}}(\ell \circ \mathcal{H}) = \mathop{\mathbb{E}}_{S, \boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \sum_{i=1}^{m} \sigma_i \mathsf{q}_i \ell(h(x_i), y_i) \right], \tag{2}$$

where σ_is are independent and uniform random variables taking values in $\{-1, +1\}$. 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 $\{\mathsf{q}\colon 0 < \|\mathsf{q}-\mathsf{p}^0\|_1 < 1\}$. The proof is given in Appendix B.

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

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

Analysis of bounds. Theorem 1 gives a guarantee on the expected loss based on a q-weighted sample, the labeled discrepancy, the q-weighted Rademacher complexity, and $\|\mathsf{q}\|_2$. When q is a distribution a term to minimize is $\sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathrm{dis}(\mathcal{D}_{T+1})$. The bound thus recommends less allocation of weight (indicated by $\overline{\mathsf{q}}_t$) to samples that have a large discrepancy with the target – they do not contain as useful training points. Another way to see this is by looking at the loss from an arbitrary sample, $\sum_{i=n_t}^{n_t+m_t} \mathsf{q}_i[\ell(h(x_i), y_i) + \mathrm{dis}(\mathcal{D}_{T+1}, \mathcal{D}_t)]$, which shows the loss from each point being enlarged by the appropriate discrepancy. There is also a natural balance between the q-weighted empirical loss and $\|\mathsf{q}\|_2$ terms: we are interested in minimizing the former, but not at the expense of giving most points a weight of zero and thus increasing $\|\mathsf{q}\|_2$ too much. The last term also lends itself to an interpretation of an *effective sample size* gleaned from q, as we can compare $\|\mathsf{q}\|_2$ to the inverse of square-root of the sample size from other bounds. Theorem 1 additionally contain $\|\mathsf{q}-\mathsf{p}^0\|_1$ and $\mathrm{dis}(\mathsf{q}, \mathsf{p}^0)$ terms, which both suggest that the q should not be too far from the reference p^0. The global insight suggested by this bound is that a balance of all these terms is important for generalization to be successful in drifting. We next describe our DRIFT algorithm based on these observations.

4 DRIFT Algorithm

Theorem 1 suggests minimizing the right-hand side of the inequality with an ideal choice of $h \in \mathcal{H}$ and $\mathsf{q} \in [0,1]^m$. If we assume that $\mathcal{H}$ 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:

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

where the weighted Rademacher complexity is upper-bounded as by Lemma 1, Appendix B. For p^0 we make the natural choice of 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, used for all experiments, where the $\mathrm{dis}(\mathsf{q}, \mathsf{p}^0)$ term is upper-bounded by $\|\mathsf{q}-\mathsf{p}^0\|_1$, allowing it to be absorbed into λ_1. In Appendix F we also introduce a Naive-DRIFT algorithm where segments $S_1, \ldots, S_T$ are combined in one.

Note that $\mathrm{dis}(\mathsf{q}, \mathsf{p}^0)$ is a convex function of q since it is a supremum of convex functions of q: $\mathrm{dis}(\mathsf{q}, \mathsf{p}^0) = \sup_{h \in \mathcal{H}} \{\sum_{i=1}^{m} (\mathsf{q}_i - \mathsf{p}_i^0) \ell(h(x_i), y_i)\}$. 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 alternating minimization or DC-programming. Here, alternating minimization alternates 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. In Appendix D and Appendix E we also discuss how to estimate discrepancies and automatically detect segments.

5 Experimental evaluation

We compare SDRIFT to several baseline algorithms in real-world regression and classification settings. In Appendix G we further provide experimental results on sythetic data illustrating a number of favorable qualities of SDRIFT such as automatically honing in on segments of low discrepancy.

Table 1: Performance of the SDRIFT algorithm against baselines. For regression (top 5 rows) we report relative errors normalized so that training on target has an MSE of 1.0. For classification (bottom 4 rows) we report relative accuracies normalized so training on just target has an accuracy of 1.0. Best results in boldface, ties in italics.

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	*0.945 ± 0.01*	*0.94 ± 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	$\mathbf{0.96 \pm 0.006}$
`STAGGER`	0.69 ± 0.006	0.73 ± 0.05	0.74 ± 0.01	1.02 ± 0.03	0.98 ± 0.02	$\mathbf{1.05 \pm 0.03}$
`Electricity`	0.95 ± 0.01	0.93 ± 0.02	0.84 ± 0.02	1.09 ± 0.02	1.02 ± 0.07	$\mathbf{1.13 \pm 0.02}$
`Room Occupancy`	0.62 ± 0.02	0.63 ± 0.01	0.72 ± 0.03	1.02 ± 0.04	$\mathbf{1.07 \pm 0.01}$	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*	*1.01 ± 0.004*

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, 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: 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.

Regression and classification tasks

We compare the SDRIFT algorithm to that of the baselines on a number of regression and classification tasks. For pointers to the dataset and details on the experimental procedure, see Appendix G. For regression we report performance in terms of MSE and normalize so training only on the target gives an MSE of 1. Thus, well-performing algorithms have an MSE < 1. For classification, we use accuracy and well-performing algorithms have an accuracy > 1. Table 1 reports our results. The KMM and DM algorithms admit no principled mechanism for down-weighting 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 Figures 6 -7 in Appendix G, the SDRIFT algorithm effectively discards many segments and assigns them little or no q-mass. 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 poorer performance. The EXP algorithm is competitive and ties in some instances with SDRIFT, for example when past segments receive very little from SDRIFT. Finally, we compare to the BSTS algorithm. For dataset with a clear time component: `wind` (month), `news` (weekday), `airline` (hour), `traffic` (hour) `Room` (hour) it provides a strong baseline, but proves sub-optimal for general drifting problems. In preliminary results we also outperform the MDAN soft-max algorithm (Zhao et al., 2018).

6 Conclusion

We presented a detailed study of a distribution drift problem that arises in many applications, and we derived an algorithm based on a detailed theoretical analysis. Our experimental results suggest that this algorithm is of practical use with significant benefits in several tasks, although it requires careful tuning of three hyperparameters. Our analysis and theory are likely to be useful in the study of other drifting problems and adaptation tasks.

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.

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/.

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.

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. 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.

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. 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.

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.

J. Haslett and A. E. Raftery. Space-time modeling with long-memory dependence: assessing ireland's wind-power resource. technical report. 1987.

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.

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.

E. Ikonomovska. Airline dataset. URL http://kt.ijs.si/elena_ikonomovska/data.html.

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.

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.

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*, 2009.

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. M. Muñoz. 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.

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.

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.

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, *Proceedinds 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.

H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31:8559–8570, 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

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 Offline setting

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 (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). Finally, Zhao et al. (2018) provide generalization bounds and algorithms for domain adaptation with multiple source domains, but in an unsupervised setting that lacks a time component.

A.3 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

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

Theorem 2. *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 $\mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_{T+1}^{m_{T+1}}$, the following holds for all $h \in \mathcal{H}$:*

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

Furthermore, when q is a distribution, $\|\mathsf{q}\|_1 = 1$, the inequality can be replaced with

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

The simplification of the second term when q is a distribution stems from the following steps: $\mathrm{dis}\big((1-\overline{\mathsf{q}}_{T+1})\mathcal{D}_{T+1}, \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathcal{D}_t\big) = \mathrm{dis}\big(\sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathcal{D}_{T+1}, \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathcal{D}_t\big) = \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathrm{dis}(\mathcal{D}_{T+1}, \mathcal{D}_t)$.

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 $\mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_{T+1}^{m_{T+1}}$, we define $\Phi(S)$ as follows:

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

Changing point x_i to some other point x_i' affects $\Phi(S)$ at most by q_i, as we consider loss functions $\ell \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ 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 $h \in \mathcal{H}$:

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

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

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

Thus, the expectation term can be expressed as follows:

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

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

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

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

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

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* Δ_m*. Then, for any* $\epsilon > 0$*, there exists* $h \in \mathcal{H}$ *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* $\mathcal{D}_1^{m_1}\otimes\cdots\otimes\mathcal{D}_{T+1}^{m_{T+1}}$*:*

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

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

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

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 $h \in \mathcal{H}$ such that $\mathcal{L}(\mathcal{D}_{T+1}, h) - \mathcal{L}\big(\sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathcal{D}_t, h\big) \geq \operatorname{dis}\big(\mathcal{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathcal{D}_t\big) - \epsilon$. For that h, we have

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

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

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

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:*

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

Proof. The result follows immediately Talagrand's contraction lemma, by the $\|\mathsf{q}\|_\infty$-Lipschitness 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 $\mathfrak{R}_{\mathsf{q}}(\ell \circ \mathcal{H}) = \mathfrak{R}_m(\ell \circ \mathcal{H})$.

The following theorem further extends this result to a bound that can be used to choose both $h \in \mathcal{H}$ 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 $\{\mathsf{q} \colon 0 < \|\mathsf{q} - \mathsf{p}^0\|_1 < 1\}$.

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

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

Proof. Consider two sequences $(\epsilon_k)_{k \geq 0}$ and $(\mathsf{q}^k)_{k \geq 0}$. By Theorem 2, for any fixed $k \geq 0$, we have:

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

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

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

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

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

Furthermore, for that k, the following inequalities hold:

$$\begin{aligned}
\sum_{i=1}^{m} \mathsf{q}_i^k \ell(h(x_i), y_i) &\le \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \mathrm{dis}(\mathsf{q}^k, \mathsf{q}) \\
&\le \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \mathrm{dis}(\mathsf{q}^k, \mathsf{p}^0) + \mathrm{dis}(\mathsf{p}^0, \mathsf{q}) \\
&\le \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \|\mathsf{q}^k - \mathsf{p}^0\|_1 + \mathrm{dis}(\mathsf{q}, \mathsf{p}^0) \\
&\le \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \|\mathsf{q} - \mathsf{p}^0\|_1 + \mathrm{dis}(\mathsf{q}, \mathsf{p}^0), \\
\mathrm{dis}\Bigg(\mathcal{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t^k \mathcal{D}_t\Bigg) &\le \mathrm{dis}\Bigg(\mathcal{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathcal{D}_t\Bigg) + \|\mathsf{q}_t^k - \mathsf{q}_t\|_1 \\
&\le \mathrm{dis}\Bigg(\mathcal{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathcal{D}_t\Bigg) + \|\mathsf{q}^k - \mathsf{p}^0\|_1 + \|\mathsf{p}^0 - \mathsf{q}\|_1 \\
&\le \mathrm{dis}\Bigg(\mathcal{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathcal{D}_t\Bigg) + 2\|\mathsf{p}^0 - \mathsf{q}\|_1, \\
\mathfrak{R}_{\mathsf{q}^k}(\ell \circ \mathcal{H}) &\le \mathfrak{R}_{\mathsf{q}}(\ell \circ \mathcal{H}) + \|\mathsf{q}^k - \mathsf{q}\|_1 \le \mathfrak{R}_{\mathsf{q}}(\ell \circ \mathcal{H}) + 2\|\mathsf{q} - \mathsf{p}^0\|_1, \\
\text{and} \qquad \|\mathsf{q}^k\|_2 &\le \|\mathsf{q}\|_2 + \|\mathsf{q}^k - \mathsf{q}\|_2 \le \|\mathsf{q}\|_2 + \|\mathsf{q}^k - \mathsf{q}\|_1 \le \|\mathsf{q}\|_2 + 2\|\mathsf{q} - \mathsf{p}^0\|_1.
\end{aligned}$$

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 $\mathcal{D}_1^{m_1} \otimes \cdots \otimes \mathcal{D}_{T+1}^{m_{T+1}}$, the following holds for all $h \in \mathcal{H}$ and $\mathsf{q} \in \{\mathsf{q}\colon 0 \le \|\mathsf{q} - \mathsf{p}^0\|_1 < 1\}$:*

$$\begin{aligned}
\mathcal{L}(\mathcal{D}_{T+1}, h) \le \sum_{i=1}^{m} \mathsf{q}_i \ell(h(x_i), y_i) + \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathrm{dis}(\mathcal{D}_{T+1}, \mathcal{D}_t) + \mathrm{dis}(\mathsf{q}, \mathsf{p}^0) + 2\mathfrak{R}_{\mathsf{q}}(\ell \circ \mathcal{H}) + 6\|\mathsf{q} - \mathsf{p}^0\|_1 \\
+ \big[\|\mathsf{q}\|_2 + 2\|\mathsf{q} - \mathsf{p}^0\|_1\big]\Bigg[\sqrt{\log\log_2 \tfrac{2}{1 - \|\mathsf{q} - \mathsf{p}^0\|_1}} + \sqrt{\tfrac{\log \frac{2}{\delta}}{2}}\Bigg].
\end{aligned}$$

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

$$\begin{aligned}
\mathrm{dis}\Bigg(\mathcal{D}_{T+1}, \sum_{t=1}^{T+1} \overline{\mathsf{q}}_t \mathcal{D}_t\Bigg) &= \mathrm{dis}\Bigg(\Bigg[(1 - \mathsf{q}_{T+1}) + \sum_{t=1}^{T} \overline{\mathsf{q}}_t\Bigg]\mathcal{D}_{T+1}, \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathcal{D}_t\Bigg) \\
&\le \Bigg(\sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathcal{D}_{T+1}, \sum_{t=1}^{T} \overline{\mathsf{q}}_t \mathcal{D}_t\Bigg) + |1 - \|\mathsf{q}\|_1| \\
&= \sum_{t=1}^{T} \overline{\mathsf{q}}_t (\mathcal{D}_{T+1}, \mathcal{D}_t) + |\|\mathsf{p}\|_1 - \|\mathsf{q}\|_1| \\
&= \sum_{t=1}^{T} \overline{\mathsf{q}}_t (\mathcal{D}_{T+1}, \mathcal{D}_t) + |\|\mathsf{p} - \mathsf{q}\|_1|.
\end{aligned}$$

Combining this inequality with the bound of Theorem 1 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 $(x, x') \in \mathcal{X}^2$ and $\alpha \in [0, 1]$, by the convexity of f and the monotonicity of $x \mapsto x^2$ on $\mathbb{R}_+$, we can write

$$f^2(\alpha x + (1 - \alpha)x') \le \big[\alpha f(x) + (1 - \alpha) f(x')\big]^2 \le \alpha f^2(x) + (1 - \alpha) f^2(x'),$$

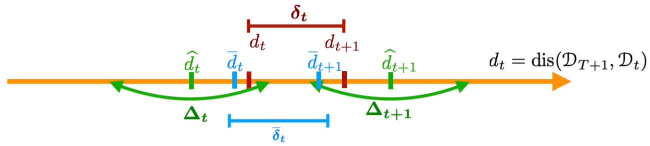

Figure 2: Enhanced discrepancy estimation: $\widehat{d}_t$s are original discrepancy estimates; $\overline{d}_t$s are corrected estimates leveraging the higher quality estimates $\overline{\delta}_t$s and the sequentiality of the drifting distribution.

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 Discrepancy estimation

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

We define the discrepancy with absolute values as: $\mathrm{Dis}(\mathcal{D}_i, \mathcal{D}_j) = \max\{\mathrm{dis}(\mathcal{D}_i, \mathcal{D}_j), \mathrm{dis}(\mathcal{D}_j, \mathcal{D}_i)\}$.

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

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

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 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:

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

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, $|\mathrm{dis}(\mathcal{D}_{T+1}, \mathcal{D}_t) - \widehat{d}_t|$ 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 $\mathcal{H}$ and have a more precise local discrepancy where the maximum is now taken over this smaller set. We reduce $\mathcal{H}$ by training a relatively accurate classifier $h_{\mathcal{D}_{T+1}}$ on a fraction n of points from $S_{T=1}$ so we can restrict $\mathcal{H}$ to a ball $\mathsf{B}(h_{\mathcal{D}_{T+1}}, r)$ 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 Δ_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

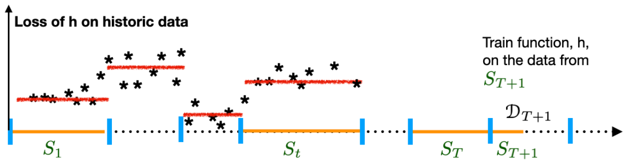

Figure 3: Illustration of how to automatically determine the distributions $\mathcal{D}_t$ with homogeneous discrepancies $\mathrm{dis}(T+1, t)$. A classifier h is determined by minimizing its loss on the data $\mathcal{S}_{T+1}$. Its loss on the historic data is determined, and a step function fitted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

large and affect the accuracy of our estimation. First, note that, by the triangle inequality, for any $t \in [T-1]$, the following holds: $\mathrm{dis}(\mathcal{D}_{T+1}, \mathcal{D}_{t+1}) - \mathrm{dis}(\mathcal{D}_{T+1}, \mathcal{D}_t) \leq \mathrm{dis}(\mathcal{D}_t, \mathcal{D}_{t+1})$. Thus, we have $|d_{t+1} - d_t| \leq \mathrm{Dis}(\mathcal{D}_t, \mathcal{D}_{t+1})$. 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, 2012) for the ℓ_1-distance or the two-sided discrepancy. Thus, we could adopt the assumption $\mathrm{Dis}(\mathcal{D}_t, \mathcal{D}_{t+1}) \leq \delta$ here. However, we can instead estimate accurately $\mathrm{Dis}(\mathcal{D}_t, \mathcal{D}_{t+1})$ 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 optimization problem:

$$\min_{\overline{d}_1, \ldots, \overline{d}_T} \sum_{t=1}^{T} \left|\overline{d}_t - \widehat{d}_t\right|^2 \quad \text{s.t. } |\overline{d}_{t+1} - \overline{d}_t| \leq \overline{\delta}_t = \widehat{\delta}_t + \sqrt{\frac{1}{m_t} + \frac{1}{m_{t+1}}}. \tag{5}$$

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

E Automatic determination of distributions $\mathcal{D}_t$

The DRIFT algorithm hinges on the knowledge of the segments supporting the distributions $\mathcal{D}_t$, which are used to estimate discrepancy and improve predictions on the target segment $\mathcal{D}_{T+1}$. Often, the distributions $\mathcal{D}_t$ 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). Here, we briefly describe a natural method related to discrepancy.

The distributions $\mathcal{D}_t$ of the DRIFT algorithm are characterized by their discrepancy $\mathrm{dis}(\mathcal{D}_{T+1}, \mathcal{D}_t)$. 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 losses, see Figure 3. One may then fit a piecewise constant function specifying a minimum 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.

E.1 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.

F Comparison of DRIFT and a naive-DRIFT solution

A naive baseline to compare the DRIFT algorithm to is that of simply combining $\mathcal{D}_1$ to $\mathcal{D}_T$ to form a single distribution $\mathcal{D}_1$, and then applying the DRIFT algorithm with the same target $\mathcal{D}_{T+1}$. 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

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

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

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

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

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

In our case, we have:

$$\mathrm{dis}(\mathcal{D}_1,\frac{1}{2}(\mathcal{D}_1+\mathcal{D}_2)) = \max_{h\in\mathcal{H}} \sum_{(x,y)} [\mathcal{D}_1(x,y) - \frac{1}{2}(\mathcal{D}_1(x,y)+\mathcal{D}_2(x,y))]\ell(h(x),y)$$
$$= \frac{1}{2}\max_{h\in\mathcal{H}} \sum_{(x,y)} [\mathcal{D}_1(x,y)-\mathcal{D}_2(x,y)]\ell(h(x),y) = \frac{1}{2}\mathrm{dis}(\mathcal{D}_1,\mathcal{D}_2) = \frac{1}{2}.$$

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

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

For the naive approach, these terms simplify to

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

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 $\mathrm{dis}(\mathcal{D}_1,\mathcal{D}_{T+1})=0$ and $\sum_{i=1}^m \mathsf{q}_i = 1$, the algorithm can allocate significantly more weight to the samples coming from $\mathcal{D}_1$, which should show an improvement over the naive approach.

F.1 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.

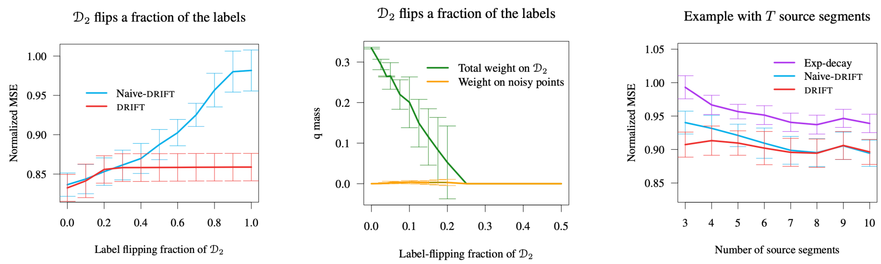

Figure 4: Synthetic data: (Left) and (Middle) with label-flipping and three segments $\mathcal{D}_1 = \mathcal{D}_3 \neq \mathcal{D}_2$. Left: MSE as a function of increasing discrepancy; Middle: the amount of q-mass assigned to $\mathcal{D}_2$ by SDRIFT, in particularly the points with flipped labels.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

G 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.

G.1 Synthetic data

Our synthetic data experiments demonstrate how the SDRIFT 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, $x \in \mathbb{R}^n, n = 20$, is sampled from a normal distribution, $\mathcal{N}(0, I_{n\times n})$. The labels y are based on a randomly drawn weight vector $w \in \mathbb{R}^n$ of unit length, and $y = w \cdot x$.

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 SDRIFT, S_1 and S_3 are drawn from the same distribution, while we artificially control the discrepancy d_2 by flipping the sign of a fraction of its labels.

We estimate the empirical discrepancy, $\widehat{d_2}$ as outlined in Appendix D, 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 S_2 is illustrated in Figure 4. In the figure we compare the performance to that of Naive-DRIFT, see Appendix F, where the samples S_1 and S_2 are assumed to belong to just one 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 $\mathcal{D}_1$ and $\mathcal{D}_2$ 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 $\mathcal{D}_1 \neq \mathcal{D}_2$. 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 $\mathcal{D}_2$. 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 results in samples with smaller discrepancy to $\mathcal{D}_{T+1}$ and the overall performance improves. For this setting a natural baseline is exponential decay of the

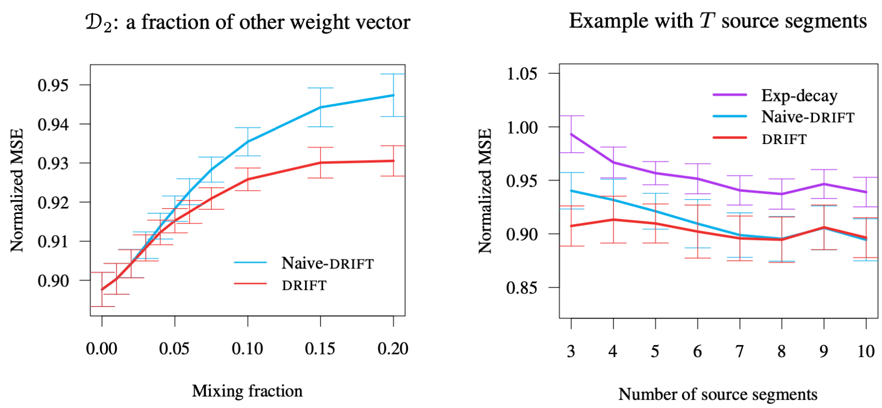

Figure 5: Left: Performance in the weight-mixing example of synthetic data with three distributions $\mathcal{D}_1 = \mathcal{D}_3 \neq \mathcal{D}_2$ as a function of increasing discrepancy. Right: Performance in the example with k source distributions.

Table 2: MSE of the SDRIFT algorithm against baselines. We report relative errors normalized so that training on target has an MSE of 1.0. Best results in boldface, ties in italics.

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	*0.945 ± 0.01*	*0.94 ± 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	$\mathbf{0.96 \pm 0.006}$

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 G.

Figure 5 (left) illustrates the normalized MSE for a weight mixing example. We use the same experimental setup as for the example with three distributions, but here the labels of $\mathcal{D}_2$ 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 $y_{\mathcal{D}_2} = (\alpha w_2 + (1-\alpha)\, w) \cdot x$. Again, we observe how the SDRIFT algorithm can effectively make use of the data from $\mathcal{D}_2$ 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 $\mathcal{D}_1$, 100% of the labels are flipped. As can be seen in Figure 5(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 $\{1e-3, 1e-2, 1e-1\}$.

G.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.

The `wind` dataset (Haslett and Raftery, 1987) 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

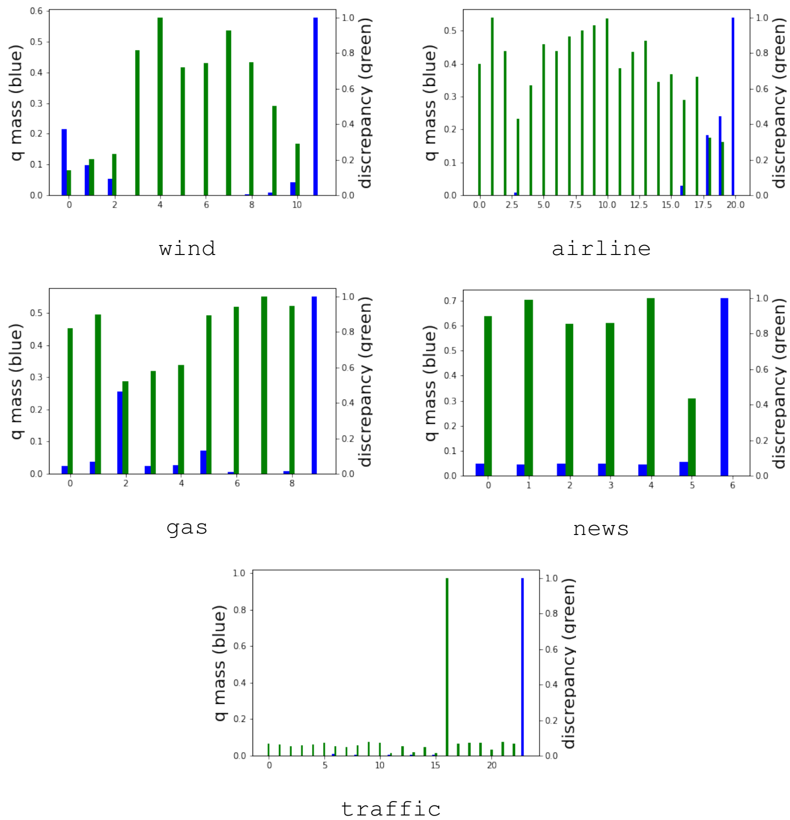

Figure 6: A plot of the total average probability mass assigned (in blue) to each segment by the SDRIFT algorithm along side the corresponding (normalized) discrepancy values (in green).

the 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 Ikonomovska 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; 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.

Table 2 (same as Table 1(top) in the main paper) 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 6, 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.

In Figure 6, 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.

G.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.

G.4 Experimental details for real-world data

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 Appendix D, determine the best hyper-parameters via cross-

Table 3: 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	SDRIFT
STAGGER	0.69 ± 0.006	0.73 ± 0.05	0.74 ± 0.01	1.02 ± 0.03	0.98 ± 0.02	$\mathbf{1.05 \pm 0.03}$
Electricity	0.95 ± 0.01	0.93 ± 0.02	0.84 ± 0.02	1.09 ± 0.02	1.02 ± 0.07	$\mathbf{1.13 \pm 0.02}$
Room Occupancy	0.62 ± 0.02	0.63 ± 0.01	0.72 ± 0.03	1.02 ± 0.04	$\mathbf{1.07 \pm 0.01}$	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*	*1.01 ± 0.004*

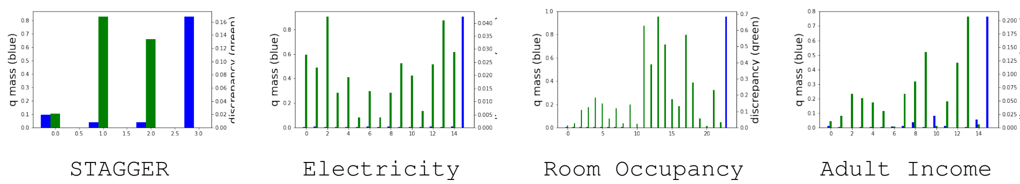

Figure 7: Average probability mass assigned (in blue) to each segment by the SDRIFT algorithm along side the corresponding (normalized) discrepancy values (in green).

validation on an 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.

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 (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.

G.5 Pseudocode for the alternate minimization procedure

In Figure 8 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 8: Alternate minimization procedure for weights and hypothesis estimation.