
Time Series Prediction under Distribution Shift using Differentiable Forgetting

Stefanos Bennett^{1,2} Jase Clarkson^{1,2}

Abstract

Time series prediction is often complicated by distribution shift which demands adaptive models to accommodate time-varying distributions. We frame time series prediction under distribution shift as a weighted empirical risk minimisation problem. The weighting of previous observations in the empirical risk is determined by a forgetting mechanism which controls the trade-off between the relevancy and effective sample size that is used for the estimation of the predictive model. In contrast to previous work, we propose a gradient-based learning method for the parameters of the forgetting mechanism. This speeds up optimisation and therefore allows more expressive forgetting mechanisms.

1. Introduction

We introduce a new framework for the problem of predicting time series that exhibit distribution shift, also known as *concept drift* in the domain of engineering (Lu et al., 2019) or *forecasting under unstable environments* in economics (Rossi, 2013). Time series exhibiting distribution shift appear in a wide range of domains (Zliobaite et al., 2016) from industrial management to biomedical applications. Financial time series distribution shift has been studied in the concept drift literature (Harries & Horn, 1996), as well as more recently in the statistics (McCarthy & Jensen, 2016) and learning theory (Kuznetsov & Mohri, 2020) literatures. We develop the forgetting mechanism approach to predicting time series under distribution shift. The key novel contribution of our work is a gradient-based learning framework for directly optimising the validation performance of differentiable forgetting mechanisms. The main advantages of our framework are that it is easy-to-use with any machine learning method based on empirical risk minimisation, provides quick estimation of forgetting mechanism hyperparameters –

thereby facilitating expressive forgetting mechanisms – and maintains competitive performance to other common approaches for handling time series distribution shift. While our framework can be applied to any general time series distribution shift problem, we show how it may in particular improve financial risk predictive modelling. Our code is available [here](#)¹.

2. Problem Definition

Consider a sequential supervised learning problem where for each $t = 1, \dots, T$, we wish to learn a mapping $f(\cdot; \theta_t) : X_t \mapsto Y_t$ between features X_t and labels Y_t where $\theta_t \in \Theta$, for some parameter space Θ . The labels, conditional on the features, follow a time-varying distribution $Y_t|X \sim \pi_t(\cdot|X)$. At each time point $t - 1$, we are interested in minimising the one-step-ahead path-dependent risk $R_t(\theta) = \mathbb{E}_{Y_t \sim \pi_t(\cdot|X_t)} [L(f(X_t; \theta), Y_t) | \{(X_\tau, Y_\tau)\}_{\tau=1}^{t-1}]$, over $\theta \in \Theta$, for a given loss function L . We follow an Empirical Risk Minimisation (ERM) (Kuznetsov & Mohri, 2020) approach in which we estimate the optimal model parameters θ_t by minimising the risk estimator $\hat{R}_t(\theta)$.

3. Model Description

3.1. Model Adaptation to Distribution Shift via Weighted Empirical Risk Minimisation

Our approach to accommodate distribution shift in the ERM formulation is through the choice of the one-step-ahead risk estimator

$$\hat{R}_t(\theta) = \sum_{\tau=1}^{t-1} \alpha_{|t-1-\tau|} L(f(X_\tau; \theta), Y_\tau)$$

where the weights $\{\alpha_i\}_{i=0}^{T-1}$ are the outputs of a forgetting mechanism $\alpha(i; \eta) = \alpha_i$, $i = 0, \dots, T-1$. The purpose of re-weighting empirical loss samples is to find a linear combination of training set sample losses that yields a model with improved generalisation. We can encode inductive biases by choosing the functional form of the forgetting mechanism. For instance, a monotonically decreasing forgetting mechanism encodes the hypothesis that more recent samples are more relevant for training. We introduce examples of

¹Department of Statistics, University of Oxford, UK ²Alan Turing Institute, London, UK. Correspondence to: Stefanos Bennett <stefanos.bennett@stats.ox.ac.uk>, Jase Clarkson <jason.clarkson@stats.ox.ac.uk>.

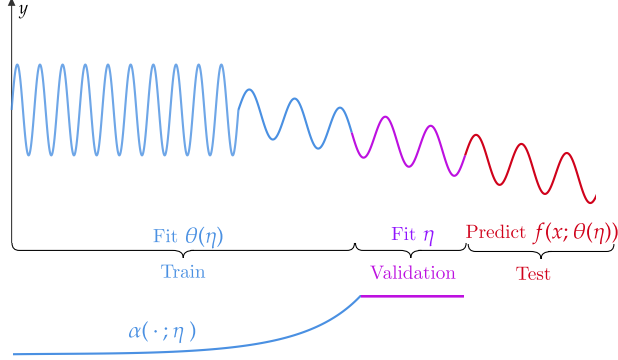


Figure 1. Method visualisation: time series training samples (top) are weighted by the forgetting mechanism $\alpha(\cdot; \eta)$ (bottom). The forgetting mechanism, which is parameterised by η and optimises predictive performance on the validation set data, assigns weight on the most recent, representative training samples.

specific functional forms in Section 5.3. The learning of forgetting mechanism parameters η controls the trade-off between the adaptivity of the forecaster and the statistical efficiency of the learning procedure: forgetting mechanisms that assign high weight to a few points are able to capture more relevant data at the cost of reducing the effective sample size that is used to fit the underlying model.

3.2. Differentiable Bi-Level Optimisation

We jointly learn the predictive model and forgetting mechanism parameters (θ, η) by solving the bi-level optimisation problem

$$\min_{\eta} g^U(\eta, \hat{\theta}) \text{ such that } \hat{\theta} \in \operatorname{argmin}_{\theta} g^L(\eta, \theta),$$

where g^U and g^L represent the upper and lower level objective functions

$$g^U(\eta, \theta) := \sum_{(X_t, Y_t) \in D_{\text{valid}}} L(f(X_t; \theta), Y_t)$$

$$g^L(\eta, \theta) := \sum_{(X_\tau, Y_\tau) \in D_{\text{train}}} \alpha(t^* - \tau; \eta) L(f(X_\tau; \theta), Y_\tau).$$

Here, D_{train} and D_{valid} denote training and validation sets, where the validation set is selected to be the most recent $T - t^*$ data samples, for some choice of t^* . The assumption underlying this choice of validation set is that the distribution $\pi_t(\cdot|X)$ does not, on average, substantially change in any given small time span (Kuznetsov & Mohri, 2020).

Applying the implicit function theorem to bi-level optimisation, we are able to perform gradient-based learning of the forgetting mechanism parameters η . Specifically, Gould et al. (2016) provide the following Lemma for differentiating through the argmin in the lower level problem:

Lemma 3.1. *Let $g^L : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}$ be an element of $C^2(\mathbb{R}^m, \mathbb{R}^n)$, the set of twice continuously differentiable*

functions in each argument, and let $\hat{\theta}(\eta) = \operatorname{argmin}_{\theta} g^L(\eta, \theta)$.

Then the derivative of $\hat{\theta}$ with respect to η_i , $i = 1, \dots, m$ is given by

$$\frac{\partial \hat{\theta}(\eta)}{\partial \eta_i} = -(\nabla_{\hat{\theta}}^2 g^L(\eta, \theta)|_{\hat{\theta}})^{-1} \frac{\partial}{\partial \eta_i} \nabla_{\theta} g^L(\eta, \theta)|_{\hat{\theta}}. \quad (1)$$

Therefore, by the chain rule, the total derivative of $g^U(\eta, \hat{\theta}(\eta))$ with respect to η_i is given by:

$$\frac{d}{d\eta_i} g^U(\eta, \hat{\theta}(\eta)) = \frac{\partial g^U}{\partial \eta_i} + (\nabla_{\theta} g^U)^T \frac{\partial \hat{\theta}(\eta)}{\partial \eta_i} \bigg|_{\hat{\theta}} \quad (2)$$

where we use the gradient expression in equation (1). Equation (2) gives us the required expression to apply gradient descent in η to the upper-level objective function:

$$\eta \leftarrow \eta - \lambda \frac{d}{d\eta} g^U(\eta, \hat{\theta}(\eta)).$$

where $\lambda \in \mathbb{R}$ is the step size.

3.3. Gradient Optimisation Routine

Since the upper level problem is generally non-convex with multiple local minima, we propose running gradient descent with multiple restarts to explore different local solutions.

Once we have estimated the parameters η using bi-level optimisation on the train-validation split, we refit our predictive model on the combined train and validation sets. The weighting scheme for the training data points is given by the forgetting mechanism using the estimated η while a uniform weighting scheme is applied to the validation data points: each of these is given the same weight as the last training set point, $\alpha(0, \hat{\eta})$.

There are further considerations when applying the bi-level optimisation procedure using a neural network as the underlying predictive model. Optimisation techniques such as flatness-aware learning (Izmailov et al., 2018) and Hessian approximation (Lorraine et al., 2020) are relevant for numerical stability and large Hessian computations; see Appendix C for further discussion.

4. Related Work

Recursive least squares (Haykin, 1986; Rossi, 2013; Harvey, 1990) provides an early example of the use of a simple forgetting mechanism for model adaptation. Since then, the concept drift literature has studied the use of forgetting mechanisms for time series distribution shift on an algorithm-by-algorithm basis (Koychev, 2000; Klinkenberg, 2004). The field of non-stationary time series forecasting (McCarthy & Jensen, 2016; Kuznetsov & Mohri, 2020) has also considered the use of forgetting mechanisms for dealing

with distribution shift. McCarthy & Jensen (2016) propose an alternative Bayesian formulation to our ERM-based approach. Kuznetsov & Mohri (2020) propose an estimator for the one-step-ahead risk that is based on a weighted empirical risk. They derive a learning guarantee on the one-step-ahead generalisation error that holds in the non-stationary time series setting. They minimise this learning guarantee over the weights in their empirical risk and the parameters of their model. The first key difference between their work and ours is that we model the weights using a parametric forgetting mechanism. Secondly, they optimise an upper bound on the generalisation error to derive $\{\alpha_i\}$ estimates whereas we directly minimise the validation-set prediction error to derive $\{\alpha_i\}$.

Our work distinguishes itself from the existing concept drift and non-stationary time series forecasting literature as it provides gradient-based learning for a generic forgetting mechanism and ERM-based learning model. A key novelty of our work is gradient-based minimisation of the validation error over the parameters of the forgetting mechanism. Gradient-based approaches are known to be much faster (Hutter et al., 2018) than grid search, which has typically been used to estimate the forgetting mechanism parameters in the concept drift (Koychev, 2000; Klinkenberg, 2004) and non-stationary time series literature (McCarthy & Jensen, 2016). Faster learning allows for more complex, richly-parameterised forgetting mechanisms that would be too computationally burdensome to fit using grid search. Our method connects time series prediction under distribution shift to the growing field of automated machine learning (Hutter et al., 2018), allowing us to leverage powerful modern automatic differentiation libraries for efficient implementation.

Beyond time series prediction, the idea of re-weighting empirical loss to address distribution shift has been considered in the context of importance weighting for transfer learning (Fang et al., 2020) and robust deep learning (Shu et al., 2019). For instance, in order to address the problem of data noise and class imbalance, Shu et al. (2019) propose to train a neural network that outputs the ERM importance weight of a training sample given that sample’s loss. The work of Shu et al. (2019) and Jenni & Favaro (2018) illustrates the interest in applying differentiable bi-level optimisation methods to the problem of generalisation in non-temporal data settings.

5. Experiments

5.1. Synthetic Data

Our synthetic data experiments are based on those from Kuznetsov & Mohri (2020) in order to facilitate comparison with their method. Four distribution shift settings are consid-

ered: regularly occurring change points, gradual distribution drift, irregularly occurring change points and no distribution shift. Data are generated by the following equations for $t = 1, \dots, 3000$ and $\epsilon_t \stackrel{\text{iid}}{\sim} N(0, 0.05^2)$:

FixedRegime: $Y_t = \theta_t Y_{t-1} + \epsilon_t$, where $\theta_t = -0.9$ if $t \in [1000, 2000]$ and 0.9 otherwise.

RandomWalk: $Y_t = \theta_t Y_{t-1} + \epsilon_t$, where $\theta_t = 1 - (t/1500)$.

RandomRegime: $Y_t = \theta_{i(t)} Y_{t-1} + \epsilon_t$, where $\theta_1 = -0.5$, $\theta_2 = 0.9$ and i_t is the stochastic process on $\{0, 1\}$ such that

$\mathbb{P}(i_{s+t} = i | i_{s+t-1:s} = i, i_{s-1} \neq i) = (0.9999825)^t$.

Stat: $Y_t = -0.5 Y_{t-1} + \epsilon_t$.

The task in each setting is to forecast the time series value Y_t using a no-intercept linear model with a feature vector that consists of the last three values of the time series $X_t = (Y_{t-1}, Y_{t-2}, Y_{t-3})$. The train-validation-test split for the synthetic data experiments is $D_{train} = \{(X_t, Y_t)\}_{t=1}^{2875}$, $D_{val} = \{(X_t, Y_t)\}_{t=2876}^{2975}$ and $D_{test} = \{(X_t, Y_t)\}_{t=2976}^{3000}$. We use 192 Monte Carlo runs for each data setting.

5.2. Real Data

We consider a financial dataset consisting of 19 years of daily log-returns for 50 NYSE equities; see Appendix A for data processing details.

Our first real data experiment investigates distribution shift in the problem of risk factor modelling (Fama & French, 1993). Specifically, we linearly decompose each equity return $Y_t^{(i)}$, $i = 1, \dots, 50$ in excess of the risk-free rate RF_t using a three factor model parameterised by $\theta_t^{(i)} \in \mathbb{R}^4$:

$$Y_t^{(i)} - RF_t = \theta_t^{(i)T} X_t \text{ where } X_t := (1, MR_t, SB_t, HL_t)^T$$

The three Fama-French factors correspond to market risk (MR), the risk factor related to size (SB) and the risk factor related to book-to-market equity (HL) (Fama & French, 1993).

Our second real data experiment investigates distribution shift in the problem of forecasting the absolute values of financial returns (Taylor, 1986). For this experiment, we restrict our dataset to 15 Exchange Traded Fund (ETF) instruments tracking industry sectors, commodity and bond prices². We forecast the next day absolute value of the return for each ETF, $Y_t^{(i)}$, $i = 1, \dots, 15$, using the five previous absolute return values for that ETF with a linear model parameterised by $\theta_t^{(i)} \in \mathbb{R}^5$:

$$|Y_t^{(i)}| = \theta_t^{(i)T} X_t \text{ where } X_t := (|Y_{t-1}^{(i)}|, \dots, |Y_{t-5}^{(i)}|)^T.$$

²The ETFs used are listed in the Appendix A.

Table 1. This table reports the Mean Squared Error (MSE) for each model averaged across Monte Carlo repetitions or across time series (for the real data). For each data set, the MSE of the best performing model is shown in bold. A star is placed by a model’s MSE whenever its MSE is significantly larger, at the 5% level, than the best model’s MSE using a paired Wilcoxon signed-rank test³ (Wilcoxon, 1945).

MODEL	FIXEDREGIME ($\times 10^{-3}$)	RANDOMWALK ($\times 10^{-3}$)	RANDOMREGIME ($\times 10^{-3}$)	STAT ($\times 10^{-3}$)	FACTOR ($\times 10^{-4}$)	VOL ($\times 10^{-5}$)
STATIONARY	4.00*	17.2*	4.20	2.54	2.60*	8.77
WINDOW	2.62	3.10*	4.63*	2.57*	2.62*	8.96
STATESPACE	2.73*	3.30*	5.25*	2.60*	3.81*	9.75*
ARIMA	2.65*	13.4*	4.41*	2.57*	—	9.24*
DBF	4.44*	23.3*	4.55*	2.64*	5.11*	10.7*
GRIDSEARCHEXP	2.63	3.00	4.31*	2.58*	2.59*	8.94
GRADEXP	3.96*	17.2*	4.20	2.55	2.59*	8.77
GRADMIXEDDECAY	2.60	2.80	4.39*	2.57*	2.49	8.76

For each real data time series, we use expanding time series cross validation to evaluate the methods. We start training for each time series using 6 years of data, the validation set is 150 days and we use the next 150 days as out-of-sample data. The train-validation-test split is moved forward by 150 days for the next model update.

5.3. Methods

We consider two simple instances of forgetting mechanisms in our experiments

$$\alpha(\tau; \eta) = \exp(-\eta_1 \tau), \quad (3)$$

$$\alpha(\tau; \eta) = \exp(-\eta_1 \tau - \eta_2 \tau^2 - \eta_3 \log(\tau + 1)). \quad (4)$$

Forgetting mechanism (3) corresponds to exponential decay while (4) corresponds to a mixture of various functional forms of decay. We fit each of these forgetting mechanisms using differentiable bi-level optimisation and call the resulting methods GradExp for mechanism (3) and GradMixedDecay for mechanism (4). Our gradient optimisation routines are implemented using 5 restarts from random initialisation, with each run performing 50 epochs of stochastic gradient updates. As a comparison to gradient-based optimisation, we fit forgetting mechanism (3) using grid search over η_1 : the corresponding method is known as GridSearchExp.

We compare against five baselines: Stationary, Window, StateSpace, ARIMA and DBF. The Stationary method assumes no distribution shift and fits the underlying models using unweighted historical samples. The commonly used Window method applies uniform weights to a fixed length interval of the most recent samples. In the StateSpace model, the state is the vector of model parameters which is modelled using random-walk update state transition equations (Brockwell & Davis, 2009). ARIMA is an autoregressive model (not applicable to the

factor experiment) often used in econometrics (Brockwell & Davis, 2009). DBF is an implementation of the two-step, discrepancy-based forecasting algorithm from Kuznetsov & Mohri (2020). Details of model implementation and hyperparameters are reported in the Appendix B.

5.4. Results

We see from Table 1 that the GradMixedDecay performs best in 4 out of 6 datasets. It has near-best performance in the RandomRegime and Stat data settings. These results demonstrate the competitive predictive performance under distribution shift that can be achieved with the use of a forgetting mechanism which is trained using gradient-based bi-level optimisation. The outperformance of GradMixedDecay over GradExp illustrates the advantage of using more expressive differentiable forgetting mechanisms, which would be too computationally burdensome to fit using grid search. We see that GradExp has comparable performance to GridSearchExp on most datasets. This shows that gradient-based optimisation is capable of achieving comparable performance to grid search in the low-dimensional η cases when grid search is feasible.

6. Conclusion

In summary, we propose learning forgetting mechanisms for time series prediction under distribution shift using gradient-based optimisation. Our approach is flexible, fast and can achieve competitive performance to baseline methods for time series prediction under distribution shift. In future work, we will investigate more complex forgetting mechanisms and sophisticated predictive models. Another interesting research extension would be to adapt our method to the online learning setting so that the predictive model can be updated in response to streaming data. More broadly, our paper motivates further research between the topics of distribution shift, time series prediction and hyper-parameter learning.

³Note that the independence assumption of the Wilcoxon test does not hold in the real data setting due to correlations between time series.

Acknowledgements

The authors would like to thank Mihai Cucuringu, Gesine Reinert and Jake Topping for their feedback. Both authors are supported by the EPSRC CDT in Modern Statistics and Statistical Machine Learning (EP/S023151/1) and the Alan Turing Institute (EP/N510129/1).

References

- Brockwell, P. and Davis, R. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer New York, 2009. ISBN 9781441903204.
- Fama, E. F. and French, K. R. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993. ISSN 0046-9777. doi: 10.2469/dig.v36.n3.4225.
- Fang, T., Lu, N., Niu, G., and Sugiyama, M. Rethinking importance weighting for deep learning under distribution shift. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11996–12007, 2020.
- Gould, S., Fernando, B., Cherian, A., Anderson, P., Cruz, R. S., and Guo, E. On Differentiating Parameterized Argmin and Argmax Problems with Application to Bi-level Optimization. *arXiv:1607.05447*, 2016.
- Harries, M. and Horn, K. Detecting concept drift in financial time series prediction using symbolic machine learning. *AI’95 Proceedings*, 1996.
- Harvey, A. C. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1990.
- Haykin, S. *Adaptive Filter Theory*. Prentice Hall, 1986.
- Hutter, F., Kotthoff, L., and Vanschoren, J. *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2018.
- Izmailov, P., Podoprikin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv:1803.05407*, 2018.
- Jenni, S. and Favaro, P. Deep bilevel learning. *arXiv:1809.01465*, 1809.01465, 2018.
- Klinkenberg, R. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3):281–300, 2004.
- Koychev, I. Gradual Forgetting for adaptation to concept drift. *Proceedings of ECAI 2000 Workshop on Current Issues in Spatio-Temporal Reasoning*, 2000.
- Kuznetsov, V. and Mohri, M. Discrepancy-Based Theory and Algorithms for Forecasting Non-Stationary Time Series. *Annals of Mathematics and Artificial Intelligence*, 88(4):367–399, 2020.
- Lorraine, J., Vicol, P., and Duvenaud, D. Optimizing millions of hyperparameters by implicit differentiation. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1540–1552. PMLR, 26–28 Aug 2020. URL <https://proceedings.mlr.press/v108/lorraine20a.html>.
- Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12): 2346–2363, 2019.
- McCarthy, D. and Jensen, S. T. Power-weighted densities for time series data. *The Annals of Applied Statistics*, 10(1):305–334, 2016.
- Rossi, B. Advances in forecasting under instability. In Elliott, G. and Timmermann, A. (eds.), *Chapter 21*, volume 2 of *Handbook of Economic Forecasting*, pp. 1203–1324. Elsevier, 2013.
- Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-Weight-Net: Learning an Explicit Mapping for Sample Weighting. *Advances in Neural Information Processing Systems*, 32:1–23, 2019.
- Taylor, S. *Modelling Financial Time Series*. Wiley, Chichester, 1986. ISBN 9780471909934.
- Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. ISSN 00994987.
- WRDS. Wharton Research Data Services (CRSP). <https://wrds-www.wharton.upenn.edu/>. Accessed: 2020.
- Zliobaite, I., Pechenizkiy, M., and Gama, J. *An Overview of Concept Drift Applications*, pp. 91–114. Springer International Publishing, Cham, 2016.

A. Data

We consider the universe of NYSE equities spanning from 04-01-2000 to 31-12-2019 from Wharton’s CRSP database (WRDS). The data consists of daily closing prices from which we compute daily log-returns. We also compute the average daily dollar volume that is traded for each equity. We subset to the 50 equities that have the largest average daily dollar volume. Any missing prices are forward-filled prior to the calculation of log-returns.

Factor returns were downloaded from Kenneth French’s data repository https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

The ETF ticker names used for volatility experiments are {‘SPY’, ‘IWM’, ‘EEM’, ‘TLT’, ‘USO’, ‘GLD’, ‘XLF’, ‘XLB’, ‘XLK’, ‘XLV’, ‘XLI’, ‘XLU’, ‘XLY’, ‘XLP’, ‘XLE’}.

B. Model Details

Each model is grid searched over its relevant hyper-parameters to maximise predictive performance on the validation set for each train-validation split of the data. In this Appendix, we provide details of the range of hyperparameter values searched over for each model.

Linear models in the *Stationary*, *Window*, *DBF*, *GridSearchExp*, *GradExp* and *GradMixedDecay* cases were fit using ridge regression with a squared error loss function, where the range of ridge penalty values searched over is $\lambda \in \{1 \times 10^{-i} : 3 \leq i \leq 6\} \cup \{0\}$.

- **GradExp, GradMixedDecay:** For optimising η on the validation set we use standard mini-batch stochastic gradient descent with momentum for all experiments, keeping the learning rate and momentum constant at 0.1 and 0.9 respectively. We perform 5 restarts⁴ of gradient descent from a random initialisation per run. We used a batch size of 32 examples, and train for 50 epochs per restart. We then select the value of η with the lowest overall validation loss for each method.
- **Window:** For a time series of length T , we grid search the length of the window over 25 values linearly spaced in $[5, T]$.
- **GridSearchExp:** We grid search η_1 in Equation (3) over 25 values. In order to ensure that the grid of values considered is of plausible a priori range, the η_1 grid is computed based on the window length grid which was searched over in the *Window* method. In particular, for each window length value considered, we select a value of η_1 that results in the *GridSearchExp* forgetting mechanism assigning almost all of its weight to the same span length.
- **ARIMA:** The hyperparameters of ARIMA are the autoregressive lag p , the differencing order d and the number of moving average components q . Following Kuznetsov & Mohri (2020), we grid search over $p, d, q \in \{0, 1, 2\}$ ³. The model is fit using maximum likelihood under a Gaussian emission model (Brockwell & Davis, 2009)
- **StateSpace:** The state-space model’s observation equation is a linear model with Gaussian noise where the states correspond to the predictive model’s regression coefficients for the given problem. In the state equation, the states are updated with a Gaussian increment. The model is fit using maximum likelihood.
- **DBF:** The DBF algorithm is an iterative two stage procedure that iteratively fits the predictive model and the weights in the ERM problem (see Kuznetsov & Mohri (2020), Section 7, Equation 14). Their procedure is run for 5 iterations. The lookback parameter, s , corresponding to the window in which the *discrepancy* is thought to be bounded (Kuznetsov & Mohri, 2020) is set to the value used in the original paper $s = 20$. The DBF algorithm has two penalty parameters λ_1 and λ_2 , which correspond to regularisation on the weighting scheme and the model parameters. We grid search both of these parameters over the same grid as the ridge regression penalty.

C. Bi-Level Optimisation for Deep Learning

In ongoing work not presented in this paper, we found that further considerations are required when the lower level problem has multiple local minima (as is the case when using a neural network as the predictive model). We have empirically found

⁴However, we have found that similar results can be obtained using a single run of gradient descent for most forgetting mechanisms and datasets.

that flatness-aware learning ([Izmailov et al., 2018](#)) of the lower level problem is important for the stability and generalisation performance of the bi-level optimisation solution. We conjecture that this is because a sharp minima leads to a large Hessian condition number and therefore unstable inversion in Equation (1).

Secondly, we have observed that identity approximation to the Hessian in Equation (1) yields competitive results while reducing the computational cost of running bi-level optimisation using a highly-parameterised predictive model such as a neural network. A number of other Hessian approximation schemes have been proposed in the gradient hyper-parameter optimisation literature, see [Lorraine et al. \(2020\)](#) for a comparison of these approaches in the context of hyper-parameter optimisation.