A Self-supervised Approach to Hierarchical Forecasting with Applications to Groupwise Synthetic Controls

Konstantin Mishchenko [1 2] **Mallory Montgomery** [2] **Federico Vaggi** [2]

Abstract

When forecasting time series with a hierarchical structure, the existing state of the art is to forecast each time series independently, and, in a post-treatment step, to reconcile the time series in a way that respects the hierarchy (Hyndman et al., 2011; Wickramasuriya et al., 2018). We propose a new loss function that can be incorporated into any maximum likelihood objective with hierarchical data, resulting in reconciled estimates with confidence intervals that correctly account for additional uncertainty due to imperfect reconciliation. We evaluate our method using a non-linear model and synthetic data on a counterfactual forecasting problem, where we have access to the ground truth and contemporaneous covariates, and show that we largely improve over the existing state-of-the-art method. All experiments in this paper are done using a Python library, which will be made available upon publication.

1. Introduction

Forecasting problems frequently have natural hierarchies. For example, looking at employment levels in the United States, state-level figures must sum to national-level, and industry-level must sum to overall employment. In Figure 1, y_4 through y_7 could be counties that sum to the states y_2 and y_3, which sum to the national employment y_1.

Forecasting each time series independently leads to several undesirable outcomes. It throws away information on the nested structure of the problem, which can result in less accurate forecasts and unreliably estimated confidence intervals. Researchers estimating many time series simultaneously test multiple hypotheses about the nested forecasts, requiring them to adjust confidence intervals, diluting the statistical power of models. Further, discrepancies make decision-making more difficult for policymakers, who may not know how to translate a case where the child-level estimates add up to more than the parent-level forecast into good policy.

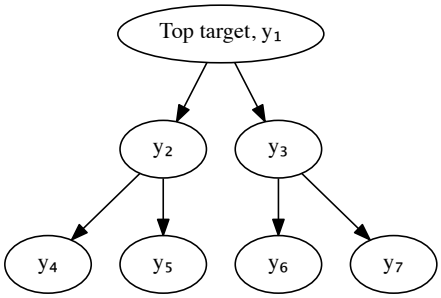

Figure 1: An example of a two-level hierarchy, in which $y_1 = y_2 + y_3$, $y_2 = y_4 + y_5$, $y_3 = y_6 + y_7$.

When forecasting time series with a known hierarchical structure, the current standard for achieving coherence in a given hierarchy is by (Hyndman et al., 2011; Wickramasuriya et al., 2018), who propose an approach to combine independent forecasts for each time series to obtain a weighted sum that minimizes the reconciliation error. Given a set of forecasts $\hat{y}_i$ and a hierarchy S, Hyndman's "optimal reconciliation" (often referred to as HTS for "hierarchical time series") estimates a matrix P such that $\tilde{y}_i = SP\hat{y}_i$, where $\tilde{y}_i$ are the reconciled forecasts. They use the forecasts and estimated variances of each time series to create simple prediction intervals that ignore the hierarchical structure. (Wickramasuriya et al., 2018) derive an estimator for P based on the trace of the covariance matrix of the forecast errors.

We propose a different approach, leveraging techniques from the machine learning literature in self-supervised learning (SSL), which utilizes the structure of the problem to improve predictions when limited labeled data is available. In our case, the forecaster has access to labeled data from the training period alongside unlabeled data from the forecast period by relying on structural properties of the data. This approach has been used successfully in fields such as computer vision. In (Rasmus et al., 2015), the

[1]King Abdullah University of Science and Technology [2]Amazon. Correspondence to: Konstantin Mishchenko <konstmish.github.io>.

Proceedings of the Time Series Workshop at the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

authors propose that the classifications of perturbed (e.g., rotated) versions of the same image should be *similar* to one another, even if the original is not labeled and thus we do not know whether they are correct.

Compared to HTS, our method results in more accurate forecasts (as measured by out-of-sample mean squared errors in simulations) that are closely reconciled. Because all forecasts are estimated simultaneously, our estimated confidence intervals account for uncertainty arising from reconciliation error, and avoid ex-post adjustments for multiple hypothesis testing.

When the child time series are heterogeneous, and the researcher has access to high-quality covariates that predict the parent and child time series well, disaggregating a single parent time series into multiple child time series and forecasting the entire hierarchy at once can lead to significant improvements in test accuracy, as shown in section 3.1. Randomly or arbitrarily splitting the parent into smaller time series will not improve test errors, and may lead to randomly divergent forecasts. For example, splitting a sample of companies alphabetically will be less useful than splitting them by size, location, or industry, for which we can find suitable theory-driven covariates.

This method is applicable to pure forecasting (with access only to past observations of the target time series and covariates) as well as to counterfactual forecasting (with access to contemporaneous covariate time series). In fact, this method can be used in any case with a maximum likelihood objective. Here we focus on counterfactual forecasting using a synthetic control method, initially developed by (Abadie et al., 2010). They construct a counterfactual forecast for a US state with a tobacco tax increase by constructing an artificial control group using unaffected states as covariates in a pre-treatment window and creating a weighted sum. In the original paper, the synthetic control was limited to a nonnegative weighted sum of other covariates (also called donors) and sparsity between the donors was enforced by choosing coefficients that were on the simplex (i.e., they sum to one). These restrictions were lifted in subsequent work (Doudchenko & Imbens, 2016), allowing for more flexible relationships between target and donors. Once an artificial control group has been built, the effect of the intervention is defined as the difference between the control group and the treated group after the intervention. In the absence of an intervention, the synthetic control forecast should closely match the observed values.

(Brodersen et al., 2015) proposed an extension to the synthetic control framework by using Bayesian structural time series models (Scott & Varian, 2013) to form the synthetic control group. By leveraging the flexibility of structural time series models, they can incorporate components into their counterfactual such as seasonality, covariates, and trend terms. There are many varieties of synthetic control methods, including generalized synthetic controls(Xu, 2017), any of which can be estimated and reconciled using our proposed loss function.

2. Theory

Notation: We denote target time series as $\mathbf{y}_i = (y_i^1, \dots, y_i^T)^\top \in \mathbb{R}^T$ for $i = 1, \dots, n$. For all i, we write the forecast for $\mathbf{y}_i$ as $\hat{\mathbf{y}}_i = (\hat{y}_i^1, \dots, \hat{y}_i^T)^\top$.

We now describe the theoretical basis for our estimations, beginning with the objective function we are interested in minimizing for the simple case of a hierarchy of n time series, with $n-1$ children that sum to a single parent. (Additional levels of the hierarchy are estimated and reconciled in the same way. Note that we do not require the hierarchy to be structured as a tree – we can optimize the model as long as the hierarchy can be described using a DAG.[1]) The objective function is:

$$\min_{\hat{\mathbf{y}}_i} \overbrace{\sum_{i=1}^{n}\sum_{t=1}^{t_0} l\left(\hat{y}_i^t, y_i^t\right)}^{\text{forecasting loss}} + \overbrace{\lambda \sum_{t=t_0+1}^{T} \left\| \hat{y}_1^t - \sum_{i=2}^{n} \hat{y}_i^t \right\|^2}^{\text{reconciliation loss}}, \quad (1)$$

where $l(\cdot,\cdot)$ is a loss function, $\{y_i^t\}_{t=1}^{t_0}$ are the observations in the training period, $\{\hat{y}_i^t\}_{t=1}^{T}$ are forecasts in the test period of the i-th target and λ is the reconciliation penalty size. Without loss of generality, we will assume that $l(\hat{y}, y) = (\hat{y} - y)^2$ and that $\hat{y}_i^t(\theta)$ is a parametric forecast.

Theorem 1. *Assume that problem* (1) *has solutions for all* $\lambda \geq 0$. *Fix* $\lambda > 0$ *and let* θ_λ^* *be an approximate solution of* (1) *with functional suboptimality not greater than* ε. *Then, we have for the corresponding outputs* $\hat{\boldsymbol{y}}_1, \dots, \hat{\boldsymbol{y}}_n$

$$\sum_{t=t_0+1}^{T} \left\| \hat{y}_1^t(\theta) - \sum_{i=2}^{n} \hat{y}_i^t(\theta) \right\|^2 = O\Big(\frac{1}{\lambda}\Big) + \frac{\varepsilon}{\lambda} \quad (2)$$

***Proof**: See Appendix Section A.*

If $\lambda = 0$, forecasts for each target y_i become independent and are simply equal to the minimizers of the forecasting loss function. For any $\lambda > 0$, each forecast will be adjusted to ensure $\|\hat{y}_1^t - \sum_{i=2}^{n} \hat{y}_i^t\|^2 = O(\frac{1}{\lambda})$. Note that the reconciliation penalty only applies for $t > t_0$, because, for $t \leq t_0$, minimizing the forecasting loss already implicitly minimizes the reconciliation loss, as the ground truth data y_i^t satisfies the hierarchy by construction.

[1] For example, using overlapping product families that must simultaneously reconcile – 'popular categories for kids' and 'popular categories for teens' may both contain apparel and school supplies. Our method forecasts and reconciles all categories and groupings simultaneously.

This approach is applicable to any method of predicting y_i^t. If y_i^t is a differentiable parametric function $f(\theta_i)$, we can optimize objective (1) using gradient based methods. In Appendix B we provide an efficient algorithm to solve (1).

We can view the reconciliation penalty as penalizing the forecaster if the time series start deviating from their natural hierarchy, and we can also justify it rigorously. Ideally, we would like to find parameters that minimize test error $(\hat{y}_i^t - y_i^t)^2$ for $i = 1, \dots, n$, but we do not have access to y_i^t for $t > t_0$. However, the reconciliation penalty serves as a lower bound for the sum of test errors. Namely, since $y_1^t = \sum_{i=2}^n y_i^t$, the Cauchy-Schwarz inequality implies

$$\begin{aligned}\Big(\hat{y}_1^t - \sum_{i=2}^n \hat{y}_i^t\Big)^2 &= \Big(\hat{y}_1^t - y_1^t - \sum_{i=2}^n (\hat{y}_i^t - y_i^t)\Big)^2 \\ &\le \sum_{i=1}^n 1^2 \sum_{i=1}^n (\hat{y}_i^t - y_i^t)^2 \\ &= n \sum_{i=1}^n (\hat{y}_i^t - y_i^t)^2.\end{aligned}$$

Intuitively, having perfect reconciliation is a necessary but not sufficient condition for a perfect forecast, therefore, if a hierarchical forecast does not perfectly reconcile, we can use that information to quantify the minimum amount by which the forecast is off.

2.1. Bayesian modeling

We now show that we can cast reconciliation in a Bayesian formulation, and use it to obtain confidence intervals that automatically incorporate uncertainty from the reconciliation step. We focus our attention on the case of counterfactual forecasting, where we have highly predictive covariates $\mathbf{X}_i$ available, and thus use a simple linear model for the likelihood. For a review of Bayesian statistics, including in time series, see (Geweke, 2005).

We forecast each time series $\mathbf{y}_i$ using a potentially disjoint set of covariates $\mathbf{X}_i$, and estimate a joint likelihood for all observations:

$$\begin{aligned}(y_1^t, \dots, y_n^t) \mid X_1^t, \dots, X_n^t, \theta_1, \dots, \theta_n, \sigma &\sim \mathcal{N}(\mu^t, \Sigma) \\ \text{with } \mu^t &\stackrel{\text{def}}{=} (X_1^t\theta_1, \dots, X_n^t\theta_n)^\top \\ \text{and LKJ prior on } \Sigma &\sim \text{LKJcorr}(\omega_\Sigma, \eta),\end{aligned} \tag{3}$$

where LKJ prior is standard choice of matrix distribution for covariance matrix from (Lewandowski et al., 2009), η is a parameter that controls how uniform the non-diagonal terms are, and ω_Σ is a standard half-Cauchy prior on the diagonal elements of the covariance.

For the reconciliation term, note that, by definition, the difference between jointly Gaussian random variables is itself Gaussian, and therefore, for a simple hierarchy with a single parent y_1 we have:

$$\Big\{\{y_1^t\}_{t=t_0+1}^T - \sum_{i=2}^n \{y_i^t\}_{t=t_0+1}^T\Big\} \sim \mathcal{N}(0, \sigma_{rec}), \tag{4}$$

with prior:

$$\sigma_{rec} \sim \text{half-Cauchy}\Big(\frac{1}{\lambda_{rec}}\Big).$$

In this formulation, the prior on the reconciliation variance λ_{rec} plays the same role as the λ hyperparameter in (1). If we choose a large value for λ_{rec}, we are expressing our prior belief that forecasts should reconcile closely.

An alternative approach to estimating σ_{rec} is to observe that every y_i is a random variable, and therefore, the linear combination $y_1^t - \sum_{i=2}^n y_i^t$ is random as well. For $t > t_0$, its variance by definition is equal to

$$\begin{aligned}&\text{Var}\Big(y_1^t - \sum_{i=2}^n y_i^t\Big) \\ &= \sum_{i=1}^n \text{Var}(y_i^t) - 2\sum_{i=2}^n \text{cov}(y_1^t, y_i^t) + \sum_{i,j=2}^n \text{cov}(y_i^t, y_j^t) \\ &= \sum_{i=1}^n \Sigma_{i,i} - 2\sum_{i=2}^n \Sigma_{1,i} + \sum_{\substack{i,j=2 \\ i\neq j}}^n \Sigma_{i,j}.\end{aligned} \tag{5}$$

Therefore, when we have the full covariance matrix of the forecast errors Σ, we can derive σ_{rec} analytically. One property of this approach is that by specifying an additional likelihood for each reconciliation step, we avoid unrealistically narrow confidence intervals: if the model has learned a very accurate forecast for every target variable, the variance in the likelihood of the reconciliation term will necessarily be small as well, as it is a linear function of the forecast variances. This means that the model cannot overfit to the training set and predict unreasonably tight confidence intervals, as those predictions will give a large penalty for non-reconciliation. By jointly estimating the reconciliation and forecast likelihoods, we force the model to come to terms with what it does not know, which translates into decreased confidence in the forecast.

3. Applications

3.1. Synthetic data

Synthetic control models are typically used to estimate the causal impact of a treatment, but there need not be an intervention. Here, we use synthetic data with no intervention, but that includes predictive contemporaneous covariates.

			no_hierarchy	*no_hierarchy + hts*	*full_hierarchy_lam_1*	*full_hierarchy_lam_10*
test	**mse**	y_1	5.78 ± 12.7	5.78 ± 12.8	1.62 ± 2.56	**1.61 ± 2.72**
		y_2	3.02 ± 9.36	3.02 ± 9.44	0.894 ± 1.59	**0.89 ± 1.39**
		y_3	2.79 ± 5.02	2.78 ± 5.01	1.03 ± 2.13	**0.956 ± 1.85**
		y_4	1.29 ± 4.1	1.3 ± 4.0	0.496 ± 0.874	**0.462 ± 0.692**
		y_5	1.06 ± 2.1	1.07 ± 2.09	**0.42 ± 0.673**	0.425 ± 0.703
		y_6	1.1 ± 1.92	1.1 ± 1.94	0.501 ± 1.17	**0.452 ± 0.826**
		y_7	1.09 ± 1.8	1.1 ± 1.78	0.445 ± 0.638	**0.422 ± 0.609**
	rec	total	0.0392 ± 0.0668	**4.62e-32 ± 1e-31**	3.28e-05 ± 1.66e-05	1.5e-05 ± 6.81e-06
train	**mse**	y_1	**0.00575 ± 0.00518**	*	0.0212 ± 0.0205	0.0265 ± 0.0253
		y_2	**0.00414 ± 0.00408**		0.0108 ± 0.0104	0.0134 ± 0.0122
		y_3	**0.00414 ± 0.00397**		0.0109 ± 0.0105	0.0135 ± 0.0128
		y_4	**0.00311 ± 0.00318**		0.00562 ± 0.00543	0.00684 ± 0.00632
		y_5	**0.00311 ± 0.00305**		0.00563 ± 0.00543	0.0068 ± 0.00627
		y_6	**0.00311 ± 0.00307**		0.00562 ± 0.0054	0.00685 ± 0.0063
		y_7	**0.0031 ± 0.00304**		0.00567 ± 0.00551	0.00696 ± 0.00657
	rec	total	**0.000195 ± 0.000118**		0.000311 ± 0.000141	0.000817 ± 0.000346

Table 1: The first column, *no_hierarchy* shows results from an independently estimated, unreconciled model, while the next column, *no_hierarchy + hts* uses the same forecasts, reconciled using Hyndman's HTS. Both *full_hierarchy_lam_1* and *full_hierarchy_lam_10* use our method with $\lambda = 1$ and $\lambda = 10$ respectively. * *We omit train MSE numbers in the second column because they are identical to the unreconciled figures.*

To assess the extent to which our reconciliation approach works for non-linear models, we consider an artificial dataset with ground truth available, with m covariates and n target variables, as in the hierarchy from Figure 1. In this setting, we show that reconciliation serves as an effective regularizer, improving forecasts even compared to HTS.

Each covariate time series $\mathbf{X}_i$ is drawn from a Gaussian process prior using the Celerite library (Foreman-Mackey et al., 2017) and each leaf target variable y_i^t is defined as a linear transformation of both $(X_1^t, \ldots, X_m^t)$ and $(X_1^t \cdot t, \ldots, X_m^t t)$ plus random noise. For non-leaf target variables, we sum up the children using the hierarchy shown in Figure 1. Formally, we take a kernel $k(\cdot, \cdot)$ from Celerite and sample

$$X_j^t \sim GP(0, k(X_j^t, X_j^{t'})), \quad j = 1, \ldots, m$$
$$y_i^t \stackrel{\text{def}}{=} (X_1^t\theta_1 + (X_1^t \cdot t)\phi_1, \ldots, X_m^t\theta_m + (X_m^t \cdot t)\phi_m + \epsilon)^\top.$$

We then use a multi-layer perceptron (one hidden layer with 100 units and ReLU activations) to forecast Y using X as an input. The neural network gets the first 1000 time steps as a training set, and we then report the MSE (averaged over 1000 independent experiments) on train, test, and reconciliation in table 3.1.

We trained the neural networks without a reconciliation penalty (*no_hierarchy*), and with two different values of λ, 1 and 10 (*full_hierarchy_lam_1* and *full_hierarchy_lam_10*). We also used HTS (conjugate gradient method) to reconcile the forecasts from the *no_hierarchy*) baseline and report those values in the *no_hierarchy + hts* column.

Reconciliation serves as a regularizer: by adding the reconciliation penalty (*full_hierarchy_lam_1* and *full_hierarchy_lam_10* columns), we decrease train set performance (4x increase in train MSE), but improve test set performance, reducing MSE by up to 3.5x compared to the *no_hierarchy* baseline in the parent node $\mathbf{y}_1$. Although the reconciliation penalty decreases test set error for all nodes, the reduction is greater on nodes higher in the hierarchy ($\mathbf{y}_1$, $\mathbf{y}_2$, and $\mathbf{y}_3$). The decrease in test error is far greater than the decrease in reconciliation error; we believe this is because, by jointly training with a reconciliation loss, we give the model an inductive bias toward simpler solutions with lower test error. In contrast, HTS adjusts the forecasts as little as possible after training is complete in order to achieve reconciliation ex-post, so it can at most reduce the forecast error by the amount of reconciliation error, which is much smaller. This is why HTS reduces the reconciliation error to approximately zero, but with little improvement in test forecast accuracy.

4. Conclusion

This paper focuses on counterfactual forecasting, however, the reconciliation loss can be applied to any maximum likelihood objective with hierarchical data, including multi-target regression and pure forecasting. When estimating hierarchical times series using e.g. ARIMA, we can add a reconciliation term to the maximum likelihood objective and automatically obtain forecasts that are nearly reconciled and have more sensible uncertainty estimates.

Although we only prove lower bounds on the error term, we see that applying our reconciliation loss to synthetic data, where we have access to real ground truth, forecasts with the reconciliation loss term have higher accuracy compared to a strong HTS baseline. In future work, we are planning a more systematic investigation into how reconciliation affects the error as well as the inductive bias of the model.

References

Abadie, A., Diamond, A., and Hainmueller, J. Synthetic control methods for comparative case studies: Estimating the effect of californias tobacco control program. *Journal of the American statistical Association*, 105 (490):493–505, 2010.

Brodersen, K. H., Gallusser, F., Koehler, J., Remy, N., and Scott, S. L. Inferring causal impact using bayesian structural time-series models. *Annals of Applied Statistics*, 9: 247–274, 2015.

Doudchenko, N. and Imbens, G. W. Balancing, regression, difference-in-differences and synthetic control methods: A synthesis. Technical report, National Bureau of Economic Research, 2016.

Foreman-Mackey, D., Agol, E., Ambikasaran, S., and Angus, R. Fast and scalable gaussian process modeling with applications to astronomical time series. *The Astronomical Journal*, 154(6):220, 2017.

Geweke, J. *Contemporary Bayesian Econometrics and Statistics*. Wiley, 2005.

Hyndman, R. J., Ahmed, R. A., Athanasopoulos, G., and Shang, H. L. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, 55(9):2579–2589, 2011.

Lewandowski, D., Kurowicka, D., and Joe, H. Generating random correlation matrices based on vines and extended onion method. *Journal of multivariate analysis*, 100(9):1989–2001, 2009.

Mishchenko, K. and Richtárik, P. A stochastic penalty model for convex and nonconvex optimization with big constraints. *arXiv preprint arXiv:1810.13387*, 2018.

Rasmus, A., Berglund, M., Honkala, M., Valpola, H., and Raiko, T. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pp. 3546–3554, 2015.

Scott, S. L. and Varian, H. R. Predicting the present with bayesian structural time series. *Available at SSRN 2304426*, 2013.

Wickramasuriya, S. L., Athanasopoulos, G., and Hyndman, R. J. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 0(0):1–16, 2018. doi: 10.1080/01621459.2018.1448825. URL `https://doi.org/10.1080/01621459.2018.1448825`.

Xu, Y. Generalized synthetic control method: Causal inference with interactive fixed effects models. *Political Analysis*, 25:57–76, 2017. doi: 10.1017/pan.2016.2.

A. Proof of theorem 1

The analysis below is motivated by that of constrained optimization with penalties (Mishchenko & Richtárik, 2018).

Proof. The objective (1) can be also seen as a penalty reformulation of the harder problem:

$$\begin{aligned} \min_{\hat{\mathbf{y}}_i} \quad & \sum_{i=1}^{n}\sum_{t=1}^{t_0} l\left(\hat{y}_i^t, y_i^t\right) \\ \text{subject to} \quad & \hat{y}_1^t = \sum_{i=2}^{n} \hat{y}_i^t, \;\; t = t_0+1, \ldots, T. \end{aligned} \tag{6}$$

Let θ^* be a solution of (6), and denote by $P_\lambda(\cdot)$ the objective function in (1). We plug θ^* into (1) and by the definition of θ_λ^* we have

$$P_\lambda(\theta_\lambda^*) \leq \min_\theta P_\lambda(\theta) + \varepsilon \leq P_\lambda(\theta^*) + \varepsilon.$$

Moreover, for $\lambda = 0$ we get by definition of θ_0^* that $P_0(\theta_0^*) \leq P_0(\theta_\lambda^*)$. Noting that

$$P_\lambda(\theta) = P_0(\theta) + \lambda \sum_{t=t_0+1}^{T} \left\| \hat{y}_1^t(\theta) - \sum_{i=2}^{n} \hat{y}_i^t(\theta) \right\|^2,$$

we deduce $P_0(\theta_\lambda^*) \leq P_\lambda(\theta_\lambda^*)$. Since θ^* is a feasible solution of (6), we additionally get $P_\lambda(\theta^*) = P_0(\theta^*)$. Therefore,

$$\lambda \sum_{t=t_0+1}^{T} \left\| \hat{y}_1^t(\theta^*) - \sum_{i=2}^{n} \hat{y}_i^t(\theta^*) \right\|^2 \leq P_\lambda(\theta^*) - P_\lambda(\theta_\lambda^*) + \varepsilon \leq P_0(\theta^*) - P_0(\theta_0^*) + \varepsilon.$$

Dividing both sides by λ finishes the proof. □

B. Optimization algorithm

Algorithm 1 Randomized block coordinate descent for (1)

1: **Input:** initial parameter vectors $\theta_1^0, \ldots, \theta_n^0$, stepsizes $\gamma_1, \ldots, \gamma_n$, penalty size $\lambda > 0$, forecast model $\hat{y}_i^t = f(X^t; \theta_i)$, number of iterations K
2: **for** $k = 0, \ldots, K$ **do**
3: Sample i uniformly from $\{1, \ldots, n\}$
4: Compute partial gradients $g_i^k = \sum_{t=1}^{t_0} \frac{\partial}{\partial \theta_i} l(f(\mathbf{X}_i^t; \theta_i), y_i^t) + 2\lambda \sum_{t=t_0+1}^{T} \frac{\partial}{\partial \theta_i} f(\mathbf{X}_i^t; \theta_i) \left\| \hat{y}_1^t - \sum_{j=2}^{n} \hat{y}_j^t \right\|$
5: Update parameters $\theta_i^{k+1} = \theta_i^k - \gamma_i g_i^k$ and $\theta_j^{k+1} = \theta_j^k$ for $j \neq i$
6: Update forecasts $\hat{y}_i^t = f(\mathbf{X}_i^t; \theta_i)$ for $t = 1, \ldots, T$
7: **end for**

Problems (1) and (6) can be solved efficiently using gradient methods. If the model used to produce $\hat{y}_i^t$ is simple, we can use projected gradient methods for (6). Its relaxed version (1), however, is easier since we can compute the derivatives of the penalty term. Furthermore, it allows for using more efficient coordinate descent optimizers such as Algorithm 1.

The main feature of randomized coordinate descent applied to (1) is that it automatically randomizes data sampling as well. Indeed, if we are adjusting the parameters used to produce $\hat{y}_i^t$, $t = 1, \ldots, T$, all terms that use y_j^t for any t and $j \neq i$ are not updated and, thus, we do not have to look up the related data. This means that the update is n times more efficient in how many coordinates it updates and, additionally, up to n times more efficient in how much data it processes. This makes overall training much faster.

However, we would like to note that with Bayesian model (3) we do not need to choose λ if the adaptive rule (5) is used. Since it requires estimating the covariance matrix, the rule from (5) is much harder numerically, but would produce better estimates.