Forecasting Hierarchical Time Series with a Regularized Embedding Space

Jeffrey L. Gleason
KUNGFU.AI
Austin, Texas
jeffrey.gleason.3@gmail.com

ABSTRACT

Collections of time series with hierarchical and/or grouped structure are pervasive in real-world forecasting applications. Furthermore, one is often required to forecast across multiple levels of the hierarchy simultaneously, and to have all forecasts reconcile. This paper proposes a new approach to reconciliation, whereby a regularization term penalizes deviation from the known structure of the collection in a learned embedding space. Experiments on real-world Australian travel data demonstrate that the proposed regularization outperforms state-of-the-art MinT reconciliation [24] in three different forecasting settings. These settings include two challenging forecasting settings: short training sequences and a long forecast horizon. We also show that our proposed regularization term is robust to the relative size of the learned embedding space.

KEYWORDS

hierarchical time series, grouped time series, time series forecasting, embedding space, neural network

ACM Reference Format:
Jeffrey L. Gleason. 2020. Forecasting Hierarchical Time Series with a Regularized Embedding Space. In *MileTS '20: 6th KDD Workshop on Mining and Learning from Time Series, August 24th, 2020, San Diego, California, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Many high-dimensional, real world collections of time series exhibit complex hierarchical and grouped structure. For example, the international sales of a multinational company can be disaggregated into sales at the country, state/province, and county levels. Additionally, coarse groupings of different products can be separated into finer and finer groupings until each product is represented by an individual series. In forecasting applications, one is often required to make forecasts for multiple different levels of the hierarchy simultaneously. It follows that leveraging the inherent structure of the collection can produce better forecasts.

Two classic approaches to utilizing hierarchical structure are "bottom-up" forecasting [6, 12, 20] and "top-down" forecasting [7, 13]. As its name suggests, the "bottom-up" method makes individual predictions at the most diffuse level and then sums these predictions according to the hierarchical structure. Similarly, the "top-down" approach makes an initial forecast for the most aggregate level and then estimates the disaggregation proportions.

Recent approaches propose post-hoc reconciliation methods that can be applied after forecasting all of the series individually [8, 22, 24]. Specifically, van Erven and Cugliari propose a minimax optimization problem where the optimal solution makes the minimum possible adjustments to reconcile a set of base forecasts [22]. Meanwhile, Wickramasuriya et al. learn a transformation that minimizes the trace of the covariance matrix of the reconciled forecast errors, under the assumption that the base forecasts are unbiased [24]. They call this approach minimum trace (MinT) reconciliation.

More recently, Mishchenko et al. propose a self-supervised reconciliation term that can be added to any maximum likelihood objective and that only constrains the forecasts for future time steps [11]. This approach is attractive because it permits concurrent optimization of both the forecasting and reconciliation objectives. However, this optimization is only efficient if the function being optimized defines unique parameters for each individual time series. This allows for the usage of a randomized coordinate descent algorithm.

This deficiency is also non-trivial, as global forecasting models, which share parameters across the entire collection of time series, have demonstrated preeminence across multiple model families in recent years. Recurrent models include DeepAR [17], the deep, recurrent quantile forecaster of Wen et al. [23], and deep state space models [15], among others [1, 2, 9, 16]. Temporal convolutional models include the WaveNet-inspired architecture of Borovykh et al. [3, 21] and the DeepTCN model of Chen et al. [4]. Finally, matrix factorization approaches include the temporally regularized matrix factorization of Yu et al. [25] and the DeepGLO model of Sen et al. [19].

This paper builds off of the learned categorical embedding space from the DeepAR modeling approach [17]. In DeepAR, a unique embedding is learned for each series in the collection. These embeddings are used, in conjunction with other features, as input to the recurrent component of the model. In this work, we add a regularization term to the embedding space that penalizes deviation from the known structure of the hierarchy. Specifically, we penalize the sum of the distances between each parent series and all of its descendants at the bottom level of the hierarchy. We use descendants at the bottom level to express constraints uniquely, following the examples of [5, 24]. We also normalize the penalty by the total number of constraints. This regularization encourages the embeddings of descendants to be similar to the embeddings of

MileTS '20, August 24th, 2020, San Diego, California, USA

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

their parents. This also expresses the motivation behind our regularization: children are more similar to their parents than to other series in the hierarchy. Finally, our approach also permits efficient stochastic optimization of models with global parameters because it applies the regularization to the embedding space instead of the output space.

We demonstrate the efficacy of our proposed approach on the data set of Australian travel flows presented in [8, 24]. This data set contains 555 total time series and 251 hierarchical constraints. It is also the largest public collection of time series with both hierarchical and grouped structure of which we are aware. On this data set, our embedding reconciliation outperforms state-of-the-art MinT reconciliation in three different forecasting settings. These settings include two challenging forecasting settings: short training sequences and a long forecast horizon. Furthermore, we demonstrate that our approach is robust to the relative size of the learned embedding dimension.

2 BACKGROUND

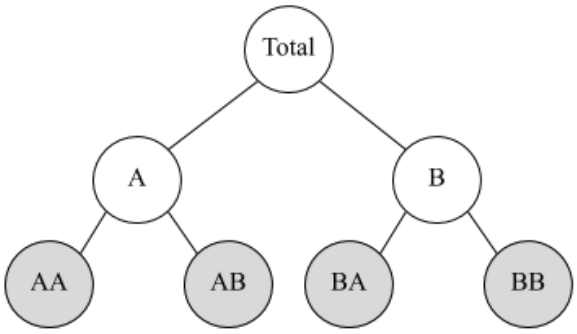

Figure 1: Simple, three-level hierarchical time series

Our notation follows the example of Corani et al. [5]. The hierarchy contains n total time series, m of which are at the most diffuse level (the shaded nodes in the example in Figure 1). The vector of observations for all series at time t is $\mathbf{y}_t = (y_t^1, \ldots, y_t^n) \in \mathbb{R}^n$. We can partition $\mathbf{y}_t$ into $[\mathbf{u}_t, \mathbf{b}_t]$, where $\mathbf{b}_t \in \mathbb{R}^m$ are the observations of the series at the most diffuse level (AA, AB, BA and BB in Figure 1) and $\mathbf{u}_t \in \mathbb{R}^{n-m}$ are the observations of aggregate series across all other levels of the hierarchy (A, B and $Total$ in Figure 1).

The hierarchical structure of the collection can be represented with a summing matrix, $\mathbf{S} \in \mathbb{R}^{n \times m}$, such that $\mathbf{y}_t = \mathbf{S}\mathbf{b}_t$. $\mathbf{S}$ contains one row for each series in the collection, but only contains columns for those series in the most diffuse level. The aggregation constraints determine each individual entry of $\mathbf{S}$:

$$\mathbf{S}_{ij} = \begin{cases} 1, & \text{if series } j \text{ is included in the aggregation to series } i \\ 0, & \text{otherwise} \end{cases}.$$

Thus, $\mathbf{S}$ represents each series in the hierarchy as a sum of series at the most diffuse level. This means that $\mathbf{S}_{n-m+1:n,1:m} \in \mathbb{R}^{m \times m}$, the sub-matrix of $\mathbf{S}$ including only those series from the most diffuse level, is $\mathbb{I} \in \mathbb{R}^{m \times m}$.

Finally, the task is to make forecasts for all time series in the collection for multiple time steps in the future given training observations $t = 1, \ldots, t_0$. We let $\hat{\mathbf{y}}_{t+h} = (\hat{y}_{t+h}^1, \ldots, \hat{y}_{t+h}^n) \in \mathbb{R}^n$ represent the vector of forecasts for all series at time $t + h$, where $t = t_0 + 1, \ldots, T$.

2.1 MinT Reconciliation

MinT reconciliation [24] learns a matrix $\mathbf{P}$, such that

$$\tilde{\mathbf{y}}_{t+h} = \mathbf{S}\mathbf{P}\hat{\mathbf{y}}_{t+h},$$

where $\tilde{\mathbf{y}}_{t+h}$ are the reconciled forecasts at time $t + h$. Specifically, Wickramasuriya et al. derive $\mathbf{P}$ such that it minimizes the trace of the covariance matrix of the reconciled forecast errors,

$$\mathrm{Tr}(\mathrm{Var}([\mathbf{y}_{t+h} - \tilde{\mathbf{y}}_{t+h} | \mathbf{y}_1, \ldots, \mathbf{y}_{t_0}])),$$

under the constraint $\mathbf{SPS} = \mathbf{S}$. This constraint enforces that the reconciled forecasts are unbiased as long as the original forecasts are also unbiased. The expression for $\mathbf{P}$ depends on the covariance matrix of the original forecast errors:

$$\mathbf{W} = \mathbb{E}[(\mathbf{y}_{t+h} - \hat{\mathbf{y}}_{t+h})(\mathbf{y}_{t+h} - \hat{\mathbf{y}}_{t+h})^\intercal | \mathbf{y}_1, \ldots, \mathbf{y}_{t_0}].$$

Different strategies exist for estimating $\mathbf{W}$ and in our experiments we use the shrinkage estimator from [18]. This estimator shrinks the off-diagonal elements of $\mathbf{W}$ toward zero, and it performs the best in the experiments in [24].

2.2 Self-Supervised Reconciliation

In [11], Mishchenko et al. propose a reconciliation term of the form:

$$\lambda \sum_{t=t_0+1}^{T} \sum_{c=0}^{C} \left\| \hat{y}_t^{c0} - \sum_{j=1}^{J} \hat{y}_t^{cj} \right\|^2.$$

This term sums over all of the constraints C in the hierarchy, where $\hat{y}_t^{c0}$ and the $\hat{y}_t^{cj}$ are the forecasts for the root node and leaf nodes, respectively, in constraint c at time t.

Notice that the regularization term only applies to time steps $t = t_0 + 1, \ldots, T$. Therefore, it only enforces reconciliation between future predictions and not between in-sample predictions, which is the reason for the "self-supervised" appellation. It is not necessary to constrain in-sample predictions because the forecasting loss also implicitly minimizes the reconciliation loss over the training time steps. Specifically, if we assume accurate training data, $\mathbf{y}_t$ must satisfy the hierarchy when $t = 1, \ldots, t_0$, but it is not guaranteed to do so for $t = t_0 + 1, \ldots, T$.

Finally, Mishchenko et al. use a randomized block coordinate descent algorithm to optimize their objective function. However, the use of a coordinate descent algorithm assumes that a separate parametric function $\hat{\mathbf{y}}_{t+h}^i = f_{\theta_i}(\mathbf{X}_{t+h}^i)$ exists for each time series, where θ_i are the parameters for series i. These separate parametric functions allow the forecasts that are not being updated, but are needed to compute the reconciliation term, to remain fixed. On the other hand, a global model, which learns a shared parametric function $\hat{\mathbf{y}}_{t+h}^i = f_\theta(\mathbf{X}_{t+h}^i)$, where θ is shared across all series, requires that all forecasts are updated at each iteration. This is necessary in

order to compute the reconciliation term.[1] This prevents stochastic optimization routines that sample mini-batches over the entire collection and thus does not scale as the number of series in the collection increases.

3 METHOD

We propose a reconciliation term that is similar in form to that of Mishchenko et al., but which penalizes distance in a learned embedding space instead of distance in the output space. The first version of our regularization uses the squared L_2 norm as the distance metric:

$$\lambda \frac{\sum_{c=0}^{C} \sum_{j=1}^{J} \|E_\phi(\mathbf{v}_{c0}) - E_\phi(\mathbf{v}_{cj})\|^2}{C}. \tag{1}$$

Here, E_ϕ is an embedding matrix with learned parameters ϕ, which maps the set of one-hot categorical vectors, representing individual time series $\{\mathbf{v} \in \{0,1\}^n : \sum_{i=1}^{n} v_i = 1\}$, to d-dimensional real-valued representations, $E_\phi : \mathbf{v} \rightarrow \mathbb{R}^d$. Our constraint is similar to [11] in that we sum over all constraints C, where $\mathbf{v}_{c0}$ and $\mathbf{v}_{cj}$ are the one-hot encodings of the root node and leaf nodes, respectively, in constraint c. However, instead of penalizing the distance between each parent series and the sum of its children, we penalize the the sum of the distances between each parent series and all of its children. Furthermore, we normalize by the total number of constraints. Finally, notice that this formulation obviates the need to sum over time steps, as $E_\phi(\mathbf{v})$ is time-invariant.

We also experiment with using the cosine distance instead of the squared L_2 norm as the reconciliation distance metric, which leads to a second version of our regularization term:

$$\lambda \frac{\sum_{c=0}^{C} \sum_{j=1}^{J} 1 - \frac{E_\phi(\mathbf{v}_{c0}) \cdot E_\phi(\mathbf{v}_{cj})}{\|E_\phi(\mathbf{v}_{c0})\| \|E_\phi(\mathbf{v}_{cj})\|}}{C}. \tag{2}$$

This second version is motivated by the observed correlation between cosine similarity and word similarity in learned word vector representation spaces [10, 14]. We will compare the empirical performance of the two distance constraints in Section 4.

In addition to improved empirical performance, our regularization presents two distinct advantages over previous approaches. First, like [11], our regularization term permits concurrent optimization of both the forecasting and reconciliation objectives without requiring a separate, post-hoc reconciliation step. Although it is true that neither our method nor [11] guarantee perfect reconciliation, the hyperparameter λ allows both methods to move on a continuum of reconciliation, from lightly encouraged to all but mandated. This permits flexibility in choosing λ across different applications. Furthermore, it is also possible to apply MinT reconciliation as an additional post-processing step to the forecasts that our method generates. This provably produces even better predictions [22, 24].

Second, our approach is more efficient than the method of Mishchenko and colleagues. Our regularization only requires evaluating the function $E_\phi(\mathbf{v}_i)$, instead of evaluating the function $f_\theta(\mathbf{X}_t^i)$, for $i = 1, \ldots, n$ and $t = t_0 + 1, \ldots, T$. Specifically, in each iteration, our approach only requires a single matrix multiplication, while the approach of Mishchenko and colleagues requires a forward pass through the entire model. Additionally, recall that $E_\phi(\mathbf{v}_i)$ is time-independent and thus our reconciliation term does not require summation over all future time steps. Table 2 presents an empirical comparison of the two regularization approaches. To summarize, our regularization term permits batched, stochastic optimization of a forecasting model with global parameters, while the approach in [11] necessitates un-batched, deterministic optimization.

[1] One could design a batching scheme that samples over constraints. However, this seems to present a significant challenge for most data sets because neither the number of series included in each constraint nor the number of constraints containing each series are not uniformly distributed.

4 APPLICATION

4.1 Data

We performed our empirical experiments on the Australian travel flow data presented in [8] and [24].[2] Observations were recorded with monthly frequency from 1998 to 2016 and satisfy both a geographic hierarchy and a travel type grouping. The geographic hierarchy comprises four levels: country, state, zone and region. Furthermore, each series at each level can be sub-divided into four different types of travel: holiday, visiting friends and relatives, business, and other [24]. Thus, the data set contains 555 total time series and 251 total hierarchical and grouping constraints.

4.2 Evaluation

We followed the example of Wickramasuriya et al. [24] and evaluated forecasts using the root mean squared error (RMSE), averaged over all folds and over all series at separate levels of aggregation. Specifically, we present the mean and standard deviation over f rolling folds (f depends on the specific experiment, but remains constant for all methods in that experiment) of the RMSE, averaged over all series at specific level of aggregations. This means that the standard deviation in the presented tables only accounts for variation in the average RMSE across folds. It does not account for variation in the individual RMSEs within folds. From here on, we refer to the RMSE averaged over all series at a specific level of aggregation as the aRMSE (average RMSE). We present results at three separate levels of aggregation: all series across all levels, the single series at the top level, and all series at the most disaggregated level. In each experiment, the baseline forecast was produced by a DeepAR model that included a learned embedding space that was not regularized. The forecasting objective was to estimate the parameters of a time-varying negative binomial distribution, as the travel flow data are positive. Although the travel flow data are real-valued and not count data, the negative binomial distribution produced better forecasts than a Student's t-distribution, which is continuous, but has support over the reals. The settings of other model and optimization hyperparameters can be found in Appendix A.

4.3 Forecasting Comparison

In our initial experiment, we compared six different reconciliation approaches: self-supervised reconciliation [11], post-hoc MinT reconciliation [24], squared L_2 embedding reconciliation, cosine

[2] Publicly available here: https://robjhyndman.com/publications/hierarchical-tourism/.

Table 1: Mean ± standard deviation aRMSE (averaged over all series at three separate levels of aggregation) across 10 folds

Method	All Levels (555 series)	Top Level (1 series)	Bottom Level (304 series)
DeepAR Baseline	181.7 ± 42.1	2426 ± 1008	59.6 ± 4.0
+Self-Supervised	1411.9 ± 90.4	24520 ± 1792	174.7 ± 8.9
+MinT	177.7 ± 28.4	2237 ± 653	58.5 ± 4.3
+Squared L_2	163.1 ± 32.2	2037 ± 736	58.6 ± 4.8
+Cosine Distance	162.7 ± 30.0	2074 ± 663	58.8 ± 4.2
+Squared L_2 and MinT	163.5 ± 27.1	1955 ± 573	**57.4 ± 5.0**
+Cosine Distance and MinT	**160.8 ± 29.3**	**1870 ± 601**	57.5 ± 4.7

Table 2: Wall time for 1 epoch of training (50 iterations) on 1 NVIDIA K80 GPU

Method	Wall Time, 1 Epoch
DeepAR Baseline	6s
+Self-Supervised	20s
+Squared L_2	17s
+Cosine Distance	17s

distance embedding reconciliation, squared L_2 embedding reconciliation with post-hoc MinT reconciliation, and cosine distance embedding reconciliation with post-hoc MinT reconciliation. In this experiment, we set the forecast horizon to 12 months and split the 228 total observations into rolling folds with training lengths of 108 observations. This produced 10 total folds.

The addition of either embedding reconciliation term substantially reduced the mean aRMSE, compared to either the baseline or MinT reconciliation forecasts (Table 1). Both embedding reconciliation terms also caused the model to converge to a slightly higher training loss (Figure 2, Appendix B). This is evidence that the regularization term helped prevent overfitting. Furthermore, we observed that the combination of an embedding reconciliation term and MinT reconciliation reduced the forecast error even further, in all but one, exceptional case.[3]

We also observed that the relative reduction in average forecast error was substantially greater for the top level series than for the bottom level series. Specifically, the four methods that included embedding reconciliation terms decreased the mean aRMSE in the top level series by between 14.5% and 22.9%. However, they only decreased the mean aRMSE in the bottom level series by between 1.4% and 3.8%. This indicates that the embedding reconciliation term is especially proficient at reducing the forecast error of the top level series.

Furthermore, we observed a substantial increase in forecast error when we added the self-supervised reconciliation term of Mishchenko et al. This occurred despite setting $\lambda = 10e{-}7$, substantially lower than $\lambda = 1$ and $\lambda = 10$ from [11]. We chose $\lambda = 10e{-}7$ because it was proportional to the ratio between the forecasting loss and the reconciliation loss at the beginning of training in our experiments. The reason for the significant increase in forecast error is that the form of the self-supervised reconciliation term exerts a strong bias toward low-valued forecasts. Specifically, all else being equal, forecasting low values for all series incurs a smaller reconciliation penalty than forecasting high values for all series.

Finally, we compared the time it took for each of the methods to complete one epoch of training (Table 2). We observed that one epoch of training took approximately three times as long when a regularization term was included in the model compared to when one was not. However, we also observed that training with our proposed embedding term was approximately 15% more efficient than training with the self-supervised reconciliation term.

4.4 Short Training Sequences

Next, we compared the same set of reconciliation methods (excepting self-supervised reconciliation) on artificially shortened training sequences. Specifically, we limited the training length in each fold to 54 observations and maintained a forecast horizon of 12 months. This produced 14 total folds.

In this setting, the addition of the squared L_2 embedding reconciliation reduced the mean aRMSE by approximately the same amount as post-hoc MinT reconciliation (Table 3). On the other hand, the addition of the cosine distance embedding reconciliation reduced the mean aRMSE significantly more than post-hoc MinT reconciliation. We also made a number of observations that were similar to those from the initial experiment. First, both embedding reconciliation terms caused the model to converge to a higher training loss (Figure 3, Appendix B). This presents additional evidence that the regularization term helps prevent overfitting, especially in the context of short training sequences. Second, the relative reduction in average forecast error was again substantially greater for the top level series than for the bottom level series. And finally, the combination of the cosine distance regularization and MinT reconciliation again led to the smallest forecast error, this time across all three levels of aggregation.

4.5 Long Forecast Horizon

In our next experiment we compared the performance of the reconciliation methods in the context of a longer forecast horizon. Specifically, we set the forecast horizon to 24 months and kept the training sequence length fixed at 108 observations. This produced four total folds.

Once again, in this setting, adding either embedding reconciliation term reduced the mean aRMSE, compared either to the baseline

[3]This exception is likely due to a poor estimate of $\mathbf{W}$, the covariance matrix of the original forecast errors.

Table 3: Mean ± standard deviation aRMSE (averaged over all series at three separate levels of aggregation) across 14 folds with training sequences of length 54

Method	All Levels (555 series)	Top Level (1 series)	Bottom Level (304 series)
DeepAR Baseline	190.5 ± 49.6	2494 ± 992	60.3 ± 4.3
+MinT	178.9 ± 27.0	2223 ± 591	59.0 ± 4.1
+Squared L_2	180.2 ± 35.2	2195 ± 743	61.5 ± 5.6
+Cosine Distance	169.3 ± 24.3	2101 ± 556	60.2 ± 4.3
+Squared L_2 and MinT	182.5 ± 36.4	2283 ± 760	59.5 ± 4.1
+Cosine Distance and MinT	**165.1 ± 25.5**	**1892 ± 546**	**58.9 ± 3.8**

Table 4: Mean ± standard deviation aRMSE (averaged over all series at three separate levels of aggregation) across 4 folds with forecast horizon of length 24

Method	All Levels (555 series)	Top Level (1 series)	Bottom Level (304 series)
DeepAR Baseline	210.5 ± 90.9	2888 ± 1692	62.9 ± 8.9
+MinT	207.3 ± 82.0	2811 ± 1548	60.2 ± 8.5
+Squared L_2	197.9 ± 33.6	2838 ± 872	60.0 ± 4.6
+Cosine Distance	202.0 ± 32.7	2983 ± 477	60.1 ± 6.0
+Squared L_2 and MinT	**191.1 ± 25.2**	**2696 ± 580**	**57.9 ± 4.6**
+Cosine Distance and MinT	193.0 ± 42.2	2704 ± 883	58.1 ± 5.4

Table 5: Mean ± standard deviation aRMSE (averaged over all series at all levels) across 10 folds with different ratios of embedding dimension to recurrent hidden dimension

Dim. Ratio	DeepAR	+ Squared L_2	+ Cosine Distance
0.5	180.9 ± 34.5	179.7 ± 84.2	**162.6 ± 23.1**
1	181.7 ± 42.1	163.1 ± 32.2	**162.7 ± 30.0**
2	184.8 ± 63.5	172.1 ± 42.0	**168.7 ± 31.8**
4	170.7 ± 23.1	171.2 ± 29.7	**167.3 ± 23.3**

or MinT reconciliation forecasts (Table 4). The cosine distance reconciliation model also converged to a higher training loss than the baseline model (Figure 4, Appendix B). However, this time, the squared L_2 reconciliation model converged to a slightly lower training loss. We also observed that the combination of an embedding reconciliation term with MinT reconciliation further reduced the mean aRMSE. One interesting observation was that the relative reduction in average forecast error was not substantially greater for the top level series than for the bottom level series in this setting.

4.6 Sensitivity Analysis

Finally, we compared the performances of the proposed reconciliations to that of the baseline model as we varied the dimension of the embedding space. Specifically, we repeated the baseline experiment, varying the embedding dimension from half as large as the hidden dimension of the recurrent component to four times as large.

Both reconciliation terms led to better forecast performance when the embedding dimension was half as large, the same size, and twice as large as the recurrent hidden dimension (Table 5). The single exception occurred, and only for the squared L_2 term, when the embedding dimension was four times as large. Additionally, the cosine distance term outperformed the squared L_2 term across all four settings of the dimension ratio. This is strong evidence that the effectiveness of the regularization is robust to the relative size of the embedding dimension.

5 CONCLUSION

In conclusion, this paper proposes two versions of an embedding reconciliation term that penalizes deviation from the known structure of a hierarchical and/or grouped collection of time series. The reconciliation term can be learned contemporaneously with other model parameters and optimized efficiently. We demonstrated the effectiveness of both the cosine distance and squared L_2 norm versions of the reconciliation term across three different forecasting settings. These settings included two challenging forecasting settings: short training sequences and a long forecast horizon. Additionally, we showed that both formulations of the regularization are robust to the relative size of the embedding dimension. Thus, the embedding reconciliation term is an efficient and effective method for regularizing global forecasting models learned on hierarchical collections of time series.

ACKNOWLEDGMENTS

This was supported by the Defense Advanced Research Projects Agency (DARPA) under D3M (FA8750-17-C0094). Views, opinions, and findings contained in this report are those of the authors. We would like to thank Alexandra Mably, Sanjeev Namjoshi, Steve Kramer, and Benjamin Johnson for their helpful comments.

REFERENCES

[1] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. 2019. GluonTS: Probabilistic Time Series Models in Python. (2019). arXiv:cs.LG/1906.05264v2

[2] Kasun Bandara, Christoph Bergmeir, and Slawek Smyl. 2020. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications* 140 (2020), 112896.

[3] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. 2017. Conditional Time Series Forecasting with Convolutional Neural Networks. (2017). arXiv:stat.ML/1703.04691v5

[4] Yitian Chen, Yanfei Kang, Yixiong Chen, and Zizhuo Wang. 2020. Probabilistic Forecasting with Temporal Convolutional Neural Network. (2020). arXiv:stat.ML/1906.04397v3

[5] Giorgio Corani, Dario Azzimonti, and Marco Zaffalon. 2019. Reconciling Hierarchical Forecasts via Bayes' Rule. (2019). arXiv:stat.AP/1906.03105v4

[6] D. M. Dunn, W. H. Williams, and T. L. Dechaine. 1976. Aggregate versus Subaggregate Models in Local Area Forecasting. *J. Amer. Statist. Assoc.* 71, 353 (1976), 68–71.

[7] Charles W. Gross and Jeffrey E. Sohl. 1990. Disaggregation methods to expedite product line forecasting. *Journal of Forecasting* 9, 3 (1990), 233–254.

[8] Rob J. Hyndman, Roman A. Ahmed, George Athanasopoulos, and Han Lin Shang. 2011. Optimal combination forecasts for hierarchical time series. *Comput. Stat. Data Anal.* 55, 9 (2011), 2579–2589.

[9] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. 2017. Time-series extreme event forecasting with neural networks at uber. *ICML Time Series Workshop* (2017).

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 3111–3119.

[11] Konstantin Mishchenko, Mallory Montgomery, and Federico Vaggi. 2019. A Self-supervised Approach to Hierarchical Forecasting with Applications to Groupwise Synthetic Controls. *ICML Time Series Workshop 2019* (2019).

[12] Guy H. Orcutt, Harold W. Watts, and John B. Edwards. 1968. Data Aggregation and Information Loss. *The American Economic Review* 58, 4 (1968), 773–787.

[13] Mijung Park and Marcel Nassar. 2014. Variational Bayesian inference for forecasting hierarchical time series. *Divergence Methods in Probabilistic Inference Workshop at ICML* (2014).

[14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543.

[15] Syama Sundar Rangapuram, Matthias W. Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. 2018. Deep State Space Models for Time Series Forecasting. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 7796–7805.

[16] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. 2019. High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 6824–6834.

[17] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2019. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* (2019).

[18] Juliane Schäfer and Korbinian Strimmer. 2005. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in genetics and molecular biology* 4, 1 (2005).

[19] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. 2019. Think Globally, Act Locally: A Deep Neural Network Approach to High-Dimensional Time Series Forecasting. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 4837–4846.

[20] E. Shlifer and R. W. Wolff. 1979. Aggregation and Proration in Forecasting. *Management Science* 25, 6 (1979), 594–603.

[21] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. *CoRR* (2016). http://arxiv.org/abs/1609.03499

[22] Tim van Erven and Jairo Cugliari. 2015. Game-Theoretically Optimal Reconciliation of Contemporaneous Hierarchical Time Series Forecasts. In *Modeling and Stochastic Learning for Forecasting in High Dimensions*, Brossat X. Antoniadis A., Poggi JM. (Ed.). Vol. 217. Springer, Cham.

[23] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. 2017. A Multi-Horizon Quantile Recurrent Forecaster. (2017). arXiv:stat.ML/1711.11053v2

[24] Shanika L. Wickramasuriya, George Athanasopoulos, and Rob J. Hyndman. 2019. Optimal Forecast Reconciliation for Hierarchical and Grouped Time Series Through Trace Minimization. *J. Amer. Statist. Assoc.* 114, 526 (2019), 804–819.

[25] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S. Dhillon. 2016. Temporal Regularized Matrix Factorization for High-dimensional Time Series Prediction. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 847–855.

A MODEL AND OPTIMIZATION HYPERPARAMETERS

We set all model and training hyperparameters to the default values in the **GluonTS** implementation [1] of DeepAR, except for the number of epochs, the dimension of the embedding space, and the instance sampler used for training. We set the number of epochs to 50 instead of 100. Additionally, in all experiments in Sections 4.3, 4.4, 4.5 we made the dimension of the embedding space equal to the recurrent hidden dimension. The default setting in **GluonTS** is to set the embedding dimension to $[(|\mathbf{v}| + 1)/2]$, where $|\mathbf{v}|$ is the cardinality of the entire collection. We used an instance sampler for training that created a histogram of the scales of the time series. An individual time series was then sampled with probability equal to the inverse of the number of series in its bucket. Thus, time series with larger scales, which for this application includes the country-level series and the state-level series, were more likely to be sampled in each batch. The default training sampler in the **GluonTS** DeepAR implementation samples each series once, in expectation, in each epoch. Random initializations were not fixed across folds or across models and neither were the random seeds used to generate samples from the learned probability distributions.

B TRAINING LOSS CURVES

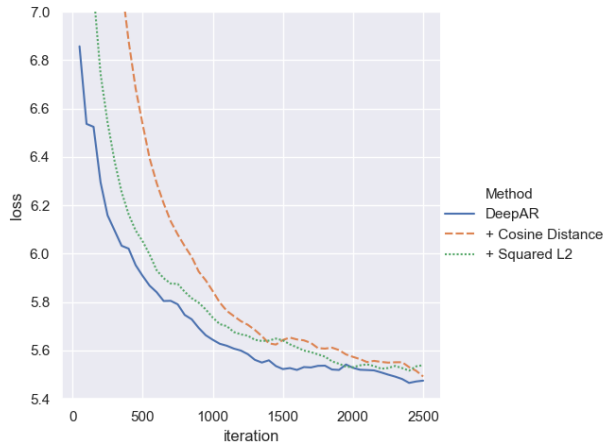

Figure 2: Training loss curves from initial experiment, exponentially smoothed over 1250 iterations and averaged across all 10 folds

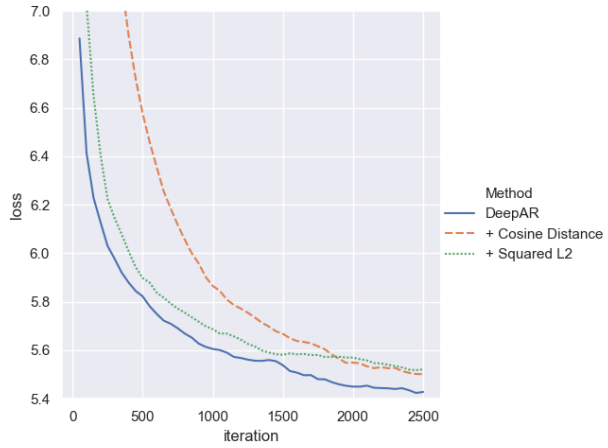

Figure 3: Training loss curves from short training sequences experiment, exponentially smoothed over 1250 iterations and averaged across all 14 folds

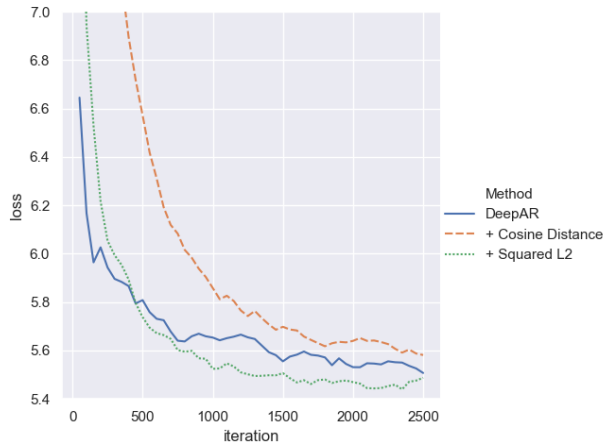

Figure 4: Training loss curves from long forecast horizon experiment, exponentially smoothed over 1250 iterations and averaged across all 4 folds