Constructing hierarchies

Abstract

Keywords: Forecasting, Hierarchical time series,

1. Introduction

2. Methodology

2.1. Framework

1. construct hierarchies based on historic observations 2. forecast and reconciliation 3. evaluation

2.2. Cluster hierarchies

Aghabozorgi et al. (2015) summarised the four components of time series clustering, including time series representations, distance measures, clustering algorithms and evaluation. In our framework, the first three components are involved in the constructing hierarchies procedure. The last component includes evaluation of reconciled forecasts of the original hierarchy.

Time series representations

Time-series representations represent raw time series in another space through techniques like feature extraction or dimensionality reduction. By utilising diverse time series representations, we are able to obtain distinct clusters which offer varied perspectives on the same dataset. Despite the absence of the theoretical evidence regarding how groups in hierarchical time series affect forecast reconciliation performance, we seek to explore multiple time series representations and discern potential correlations between hierarchical structure and the efficacy of forecast reconciliation.

Considering the impracticality of exploring every conceivable time series representation, we focus on four key ones: raw time series, in-sample error, features of time series, and features of in-sample error. The inclusion of raw time series is motivated by its simplicity and broad applicability. A crucial element contributing to the success of the minimum trace method proposed by Wickramasuriya et al. (2019) lies in estimating the covariance matrix of base forecast errors based on in-sample fit error. Hence, we consider in-sample error as a representation to examine how the structure within the error series influences reconciliation performance. Notably, to the best of our knowledge, we are the first to utilise in-sample error as a time series representation in the context of forecast reconciliation literature. The time series and in-sample error representations are standardised on a per-series basis to eliminate the impact of scale variations. Features, widely employed in capturing time series characteristics

across literature, play a pivotal role in various time series applications, including clustering (Tiano et al., 2021) and forecasting (Wang et al., 2022; Li et al., 2023). In our exploration, we incorporate features of both time series and in-sample fit error as representations. These representations allow us to glean insights into the diverse aspects of hierarchical time series data and enhance our understanding of how different structures contribute to the forecast reconciliation performance.

Distance measures

Distance measures serve as crucial tools for assessing the similarity between two series, forming the foundation for clustering algorithms to detect clusters within datasets. In the realm of time series clustering literature, we consider two widely applied distance measures: Euclidean distance and dynamic time warping (DTW) (Warren Liao, 2005).

Given the limited number of time series in comparison to the variable dimensions (i.e., the length of time series and the number of time series features) within hierarchical time series in our applications, dimension reduction on the aforementioned representations becomes imperative when employing Euclidean distance. Failure to undertake dimension reduction may result in undesirable clustering outcomes due to the challenges posed by the curse of dimensionality. To address this, we perform principal component analysis, extracting the first few principal components that collectively explain at least 80% of the variance within the data.

In contrast, DTW (Sakoe and Chiba, 1978), demonstrates reduced sensitivity to the curse of dimensionality. Unlike the one-to-one point comparison inherent in Euclidean distance, DTW accommodates time series of varying lengths through many-to-one comparisons. This adaptive approach allows for the recognition of time series with similar shapes, even in the presence of signal transformations such as shifting and/or scaling.

Clustering algorithms

Regarding clustering algorithms, we focus on two prevalent approaches, k-Medoids and agglomerative hierarchical clustering. The k-Medoids algorithm, a classical partitioning clustering method, aims to minimise the total distance between all samples within a cluster and their respective cluster centres. Unlike k-Means, which employs the mean vector of samples as the cluster centre, k-Medoids selects one sample within the cluster as the centre. Specif-

ically, we adopt the k-Medoids variant, partitioning around medoids (PAM, Kaufman and Rousseeuw, 1990). Following the recommendation of Kaufman and Rousseeuw (1990), we determine the optimal number of cluster using the average silhouette width (ASW). ASW, a popular cluster validation index, assesses the quality of clustering results by measuring the proximity of samples within a cluster compared to neighbouring clusters. Given a clustering $\mathcal{C}$, the silhouette width for the ith sample is calculated as

$$SW(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

where $a(i)$ represents the average distance of the ith sample to others in the same cluster, and $b(i)$ is the average distance to samples in the nearest cluster it is not assigned to. A higher silhouette width indicates greater proximity to samples within the same cluster than to those in neighbouring clusters. The ASW is the average of silhouette widths across all samples. We select the clustering result $\mathcal{C}$ that maximises the ASW by iterating over all possible numbers of clusters. However, an inherent limitation of ASW is its inexistence when there is only one cluster, making it challenging to discern the presence of well-separated clusters. Gap statistics (Tibshirani et al., 2002) can be used to address this limitation, but it is incompatible with DTW.

Agglomerative hierarchical clustering begins by considering each sample as a cluster, and then gradually merges clusters until all samples forms a single cluster. This process results in a binary hierarchical tree with $2n - 1$ nodes. We employ Ward's linkage (Murtagh and Legendre, 2014) to merge clusters, minimising the increase in within-cluster variances at each step. Subsequently, we construct the middle-level series from the clustering tree.

Figure 1 illustrates two example hierarchies generated by these clustering algorithms, showcasing their distinct behaviours. The primary discrepancy between the two algorithms lies in two aspects. Firstly, k-Medoids constructs a hierarchy with a single middle level, while hierarchical clustering generates multiple middle levels with a nested structure. Secondly, k-Medoids constructs a hierarchy with the same number of series as the optimal number of clusters, whereas hierarchical clustering produces multiple middle levels with $n - 2$ series. These distinctions become particularly relevant when the number of bottom-level series is substantial. We aim to investigate how these structural differences influence the performance of forecast

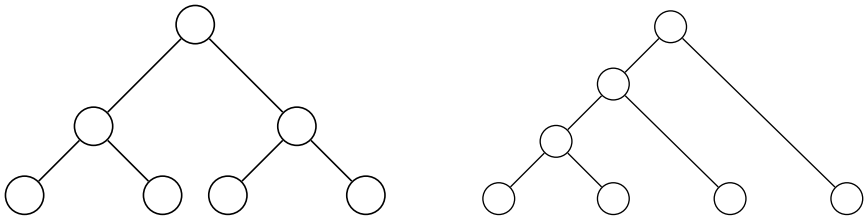

Figure 1: Example clustering results of two clustering algorithms. Left panel displays example for k-Medoids algorithm, and right panel displays example for agglomerative hierarchical algorithm.

reconciliation.

2.3. Random hierarchies

A fundamental question arises concerning the factor that plays the most crucial role if the constructed hierarchies resulting from clustering significantly enhance forecast performance. One plausible argument posits that the improved forecasting performance should attribute to the clustering procedure, which organises series with similar features into groups. This organisation creates middle-level series that are hypothesised to be easier to forecast, leading to improved reconciled forecasts. However, this logical chain is not readily traceable for two main reasons. Firstly, groups of similar time series do not inherently translate to superior middle-level forecasts. Secondly, existing literature lacks systematic theory regarding how the pattern of the variance matrix of base forecast errors influences the performance of reconciled forecasts. The complexities introduced by uncertainties in clustering, base forecasting model estimation, and variance matrix estimation compound the challenge of drawing definitive conclusions (Pritularga et al., 2021).

An alternative explanation for this question posits that the enhancement is primarily a result of an increased number of middle-level series. The perspective of forecast reconciliation as a form of forecast combination (Hollyman et al., 2021) supports this notion. As new hierarchies are constructed, the reconciled forecasts are combinations of a greater number of base forecasts, leading to improved reconciled forecasts.

To systematically explore these two explanations, we propose two randomised approaches to hierarchy construction. Constructing hierarchies randomly allows us to eliminate the influence of clustering and focus specifically on investigating the second explanation. The first approach generates hierarchies akin to those produced by k-Medoids clustering algorithms. Given m bottom-level series and a number k, we create $\lceil \frac{m}{k} \rceil$ groups. The first $\lfloor \frac{m}{k} \rfloor$ groups

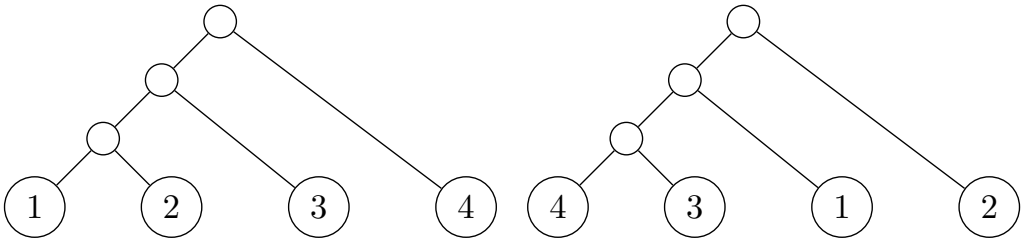

Figure 2: Examples of randomisation of a given hierarchy. Right hierarchy is obtained by shuffling the leaf nodes of left hierarchy.

each contain k series, and the last group accommodates $n - k\lfloor \frac{n}{k} \rfloor$ series. The bottom-level series are then randomly assigned to these groups. This approach offers a straightforward way to construct random hierarchies, which is proposed to investigate the effect of new middle-level series and verify the second explanation.

The second approach involves generating random hierarchies based on a given hierarchy, such as one constructed from clustering algorithms. Given a hierarchy tree, it creates new hierarchies by randomly permuting the leaf nodes. In other words, it shuffles the columns of A matrix of the given hierarchy. This method yields a new random hierarchy with an identical structure, enabling to verify both hypothesises simultaneously. If the randomised version of a hierarchy significantly outperforms itself, it suggests that the second explanation holds greater importance that the first, and vice versa. Figure 2 shows an example of this approach. The left hierarchy is the hierarchy shown in Figure 1 obtained from agglomerative hierarchical algorithm. The right hierarchy is an example of randomisation of the left hierarchy.

2.4. Combination hierarchies

Randomness technique is usually combined with forecast combination approach in the literature. For example, Petropoulos et al. (2018) illustrates the effectiveness of bootstrapping aggregation for time series forecasting which combines forecast obtained from bootstrapped versions of the time series. Therefore, we propose the "combination hierarchies" which utilises multiple hierarchies to obtain the reconciled forecasts. The hierarchies in the pool can be random hierarchies or cluster hierarchies. There are two straightforward approaches to combine multiple hierarchies. The first one is to combine their structure.

Table 1: Parameter setting for all clusters in the simulation experiments.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
Trend	Increase	Increase	None	None	Decrease	Decrease
Seasonality	Odd	Even	Odd	Even	Odd	Even

3. Simulation

In this section, we demonstrate that hierarchies constructed by clustering may not outperform hierarchies constructed randomly even if there are true clusters in the bottom level.

3.1. Simulation design

Time series generation

We assume the bottom-level time series follow an additive time series pattern with a data generating process described as follows:

$$\begin{aligned} y_t &= l_t + s_t + e_t \\ s_t &= s_{t \mod m} \\ l_t &= at + \varepsilon_t, \end{aligned} \tag{1}$$

where l_t represents the trend term which increases or decreases over time at a slope of a. The seasonal component, denoted by s_t, remains constant. Both e_t and ε_t are white noises.

We create 6 clusters at the bottom level by adjusting the directions of the trend and patterns of the seasonal components. The pattern settings for all clusters can be found in Table 1. For an increasing trend we set the slope a to 0.001, and for a decreasing trend, we set it to -0.002. The variances of the corresponding white noise ε_t are set to 2.5×10^{-5} and 4.9×10^{-5}. The terms "Even" and "Odd" indicate the position of the seasonal peak. For "Even" seasonality, the peaks are located at positions $2, 4, \ldots, m$, and vice versa. The values for these seasonal peaks and troughs are uniformly drawn from $[2, 3]$ and $[0, 1]$, respectively. The variance of e_t is set to 0.25. We generate monthly time series data with 20 time series per cluster. Figure 3 displays example time series from each cluster while Figure 4 visualises these generated time series based on the first two principal components extracted from principal component analysis of the series.

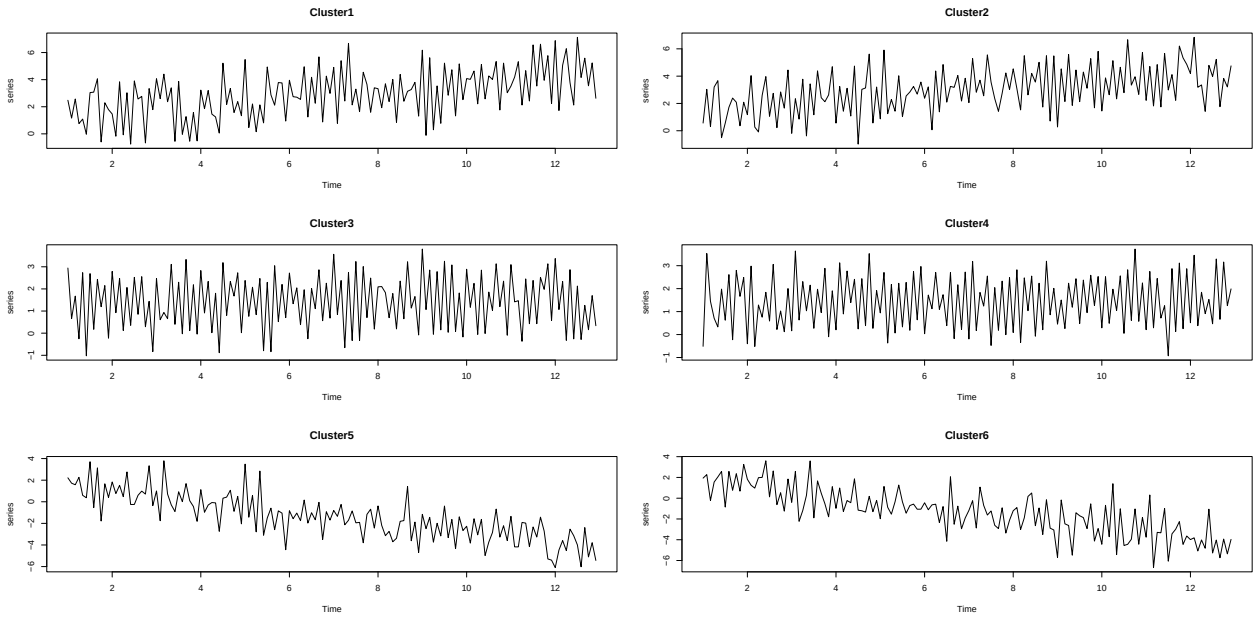

Figure 3: Example time series for each cluster in the simulation experiments.

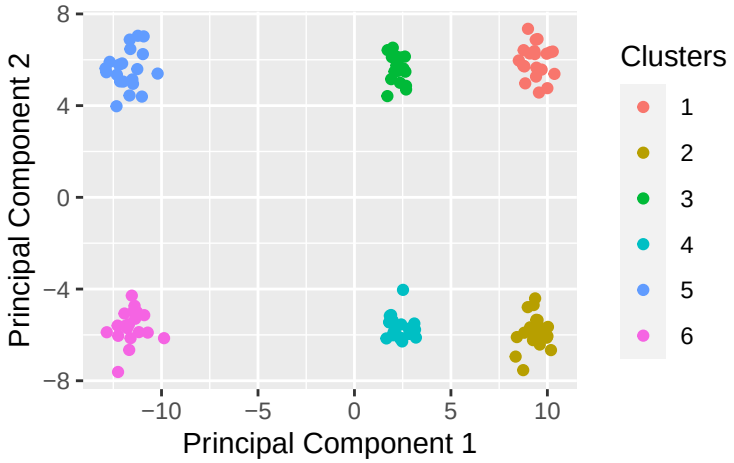

Figure 4: Visualisation of the generated time series in the simulation experiments.

Hierarchies construction

There are a total of 120 bottom series and 1 total series in the hierarchy, referred to as "C0". Additionally, we consider 4 clustering approaches and 2 forecast combination approaches. "C1" creates 6 middle-level series according to the designed correct clusters. We merge clusters with the same trend pattern, resulting in 3 clusters in the middle level, denoted by "C2". Similarly, we construct "C3" and "C4" based on the presence of the trend term and seasonal peak positions, resulting in 2 and 3 middle-level series, respectively. "A1" combines reconciled forecasts obtained from all four clustering approaches ("C1" to "C4") using equal weights. By shuffling the bottom-level series within the hierarchy "C1", we can randomly generate different hierarchies with identical structure. "A2" combines reconciled forecasts from 10 randomly constructed hierarchies using equal weights.

Forecasting

We construct two scenarios by considering different base forecasting models. In the first scenario, exponential smoothing (ETS) is used to generate base forecasts for all time series in hierarchies. This simulates a situation where the time series themselves are inputs to the clustering algorithms. In the second scenario, we use historic mean to generate base forecasts for bottom-level series and ETS for other time series in hierarchies, which simulates the case where in-sample errors are inputs to the clustering algorithms. We refer to these scenarios as "clustering by time series" and "clustering by error", respectively. To reconcile the base forecasts, we employ the minimum trace method with the shrinkage estimator (Wickramasuriya et al., 2019). The shrinkage estimator is effective at capturing the dependence structure within the forecast errors and has demonstrated superior perofrmance in various applications. For each series, we generate 144 observations, and the last 12 observations are reserved for evaluation purpose.

3.2. Evaluation

The purpose of constructing middle-level series is to improve the forecast performance of the total-level and bottom-level series by leveraging the strength of these new series. Therefore, we only evaluate the reconciled forecasts at total level and bottom level. To assess the accuracy of single time series, we use root mean squared error (RMSE) as our metric. The simulation is repeated 500 times, resulting in a total of 500×121 RMSEs for each approach. Multiple

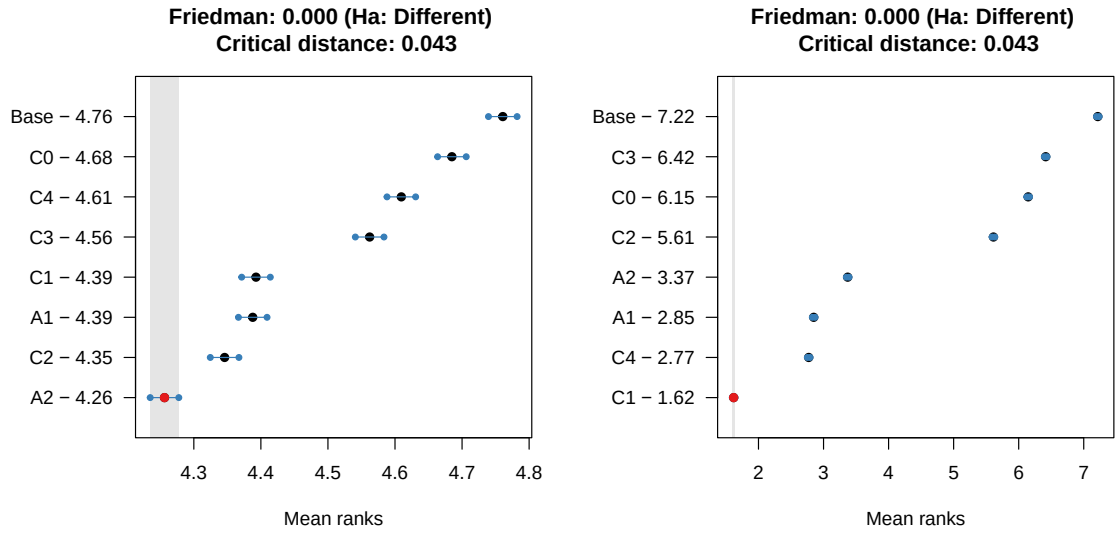

Figure 5: Average ranks and 95% confidence intervals based on the MCB Test for the 8 approaches in the two scenarios of the simulation experiments. Left panel displays test results for the "clustering by time series" scenario and right panel displays test results for the "clustering by error" scenario.

comparisons with the best (MCB) test is then applied to compute the average ranks of the 7 approaches along with base forecasts and to determine whether the performance differences are statistically different (Koning et al., 2005).

3.3. Results

Figure 5 displays the MCB test results for the two scenarios. In both scenarios, most approaches perform better than the base forecasts and the original two-level hierarchy "C0". This suggests that constructing middle-level series generally improves forecast reconciliation performance. In the "clustering by time series" scenario, the correct cluster "C1" ranks 4th, indicates that optimal clusters do not guarantee optimal forecast reconciliation performance. On the other hand, "C1" ranks 1st in the "clustering by error" scenarios. This highlights the importance of carefully selecting time series representations when constructing new middle levels using clustering algorithms. The simple average of 10 random hierarchies ranks 1st in the first scenario and the simple average of 4 clustering hierarchies ranks 3rd in both scenarios, indicating the importance of forecast combination.

4. Empirical studies

4.1. Datasets

We analyse two datasets in our empirical studies. The first dataset is the monthly Australian domestic tourism dataset, which provides visitor nights numbers from 1998 to 2016. The tourism demand of Australia is geographically disaggregated into 7 states and territories, further divided into 27 zones and 76 regions. Additionally, each geographical series is divided by four travel purposes (Wickramasuriya et al., 2019). This dataset totally contains 555 time series with 304 at the bottom level.

The second dataset focuses on causes of death in the U.S., using the ICD 10 coding system. We obtain monthly cause-specific death counts from the Center for Disease Control and Prevention (CDC) for the period between 1999 and 2019. The coding system forms an unbalanced hierarchy with 137 time series, out of which 113 time series are in the bottom level. To consolidate data with suppressed values, we combine causes that share a parent cause and calculate their death counts by subtracting death counts of sibling causes from death counts of their parent cause. The final dataset includes 120 time series with 98 in the bottom level.

4.2. Experiment design

We focus on the performance of total-level series and bottom-level series, disregarding the multiple middle levels in the original hierarchies known as "natural hierarchies". The natural hierarchy is considered one way to construct middle-level series, similar to hierarchies created through clustering. However, we demonstrate that for the purpose of forecast performance, the natural hierarchy may not be the most effective hierarchical structure.

To construct middle levels using clustering, we combine four time series representations with two distance measures and two clustering algorithms. This results in twelve different construction approaches listed in Table 2.

Additionally, we consider three forecast combination approaches. In the first approach, we create one middle level with 15 time series by randomly assigning bottom-level series as their children. We ensure that all the middle-level series have approximately an equal number of children. The choice of having 15 middle-level series is arbitrary with the goal of creating a moderate number of groups, each containing a moderate number of series. We repeat this process to create 50 such hierarchies and combine their reconciled forecasts using equal weights.

Table 2: Hierarchy construction approaches used in empirical studies.

Approaches	Representation	Dimension reduction	Distance measure	Clustering algorithms
TS-MED	Time series	Yes	Euclidean	k-Medoids
ER-MED	In-sample error	Yes	Euclidean	k-Medoids
TSF-ME	Time series features	Yes	Euclidean	k-Medoids
ERF-ME	In-sample error features	Yes	Euclidean	k-Medoids
TS-HC	Time series	Yes	Euclidean	hierarchical clustering
ER-HC	In-sample error	Yes	Euclidean	hierarchical clustering
TSF-HC	Time series features	Yes	Euclidean	hierarchical clustering
ERF-HC	In-sample error features	Yes	Euclidean	hierarchical clustering
TS-MED-DTW	Time series	No	DTW	k-Medoids
TS-HC-DTW	In-sample error	No	DTW	hierarchical clustering
ER-MED-DTW	Time series	No	DTW	k-Medoids
ER-HC-DTW	In-sample error	No	DTW	hierarchical clustering

This approach is referred to as "FC-R". In the second approach, we create 10 new hierarchies by randomly shuffling the positions of bottom-level series in the natural hierarchy and combine their reconciled forecasts using equal weights. This approach is denoted by "FC-N". The third approach, labelled by "FC-C", involves equally-weighted combining the reconciled forecasts obtained from the 12 hierarchies shown in Table 2.

The time series features used in this experiment are calculated using the `tsfeatures` package (Hyndman et al., 2022) for R. After filtering out the features that are constant across all series, 56 features are reserved. The complete list of these feature can be found in Appendix. The k-Medoids and hierarchical clustering algorithms are implemented using the `cluster`(Maechler et al., 2022) package for R. The base forecasts are generated using the automatic ETS models, which are then reconciled using the minimum trace method with shrinkage estimator.

We employ the rolling window strategy to evaluate the performance of different approaches for both datasets. We start with 96 observations and fit the base forecasting models using the available data. Then we calculate time series representations and construct new hierarchies by clustering or randomness, and then forecast 12 steps ahead. After that, the training set is increased by one observation and new forecasts are obtained. The procedure is repeated until the last 12 observations are used for evaluation. Finally, we can obtain 121 12-step-ahead forecasts for the tourism dataset and 144 12-step-ahead forecasts for the mortality dataset. We compare the forecast performance of the 15 approaches as well as the natural hierarchy, two-level hierarchy and base forecasts for both total-level and bottom-level series.

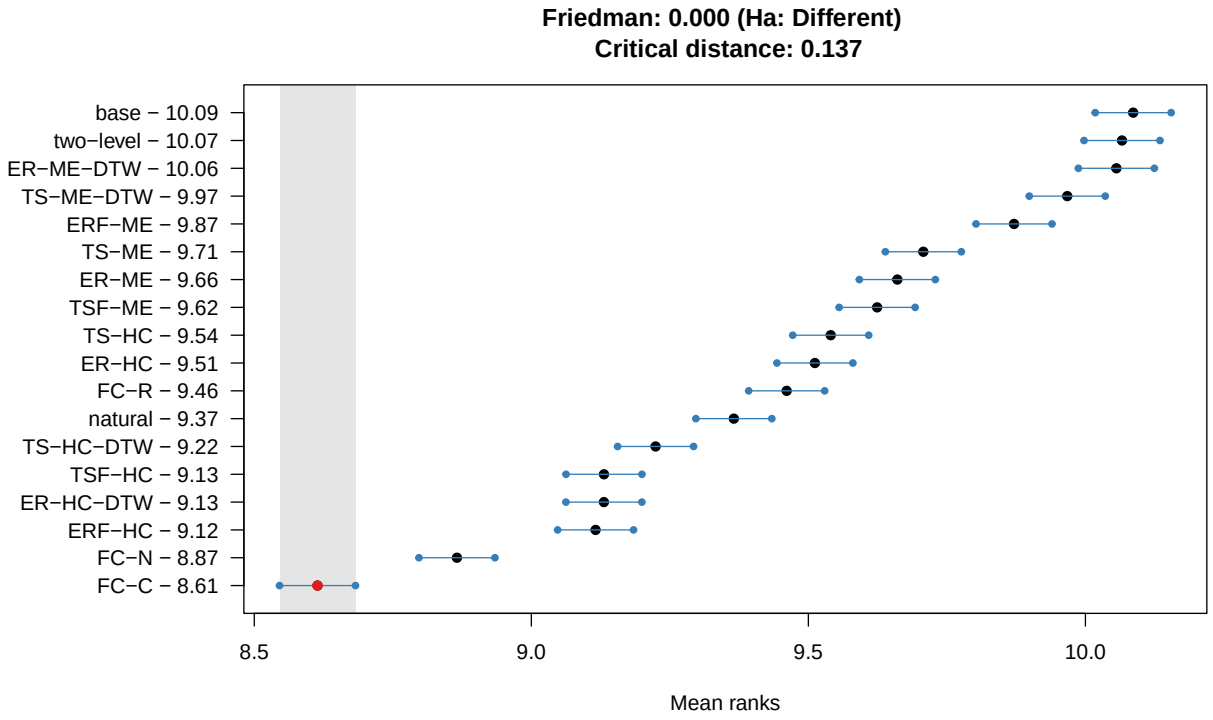

Figure 6: Average ranks and 95% confidence intervals based on the MCB Test for the 18 approaches on the tourism dataset.

4.3. Results

Same as the evaluation measure described in Section 3.2, we combine the forecasts of total-level and bottom-level series across all evaluation windows and conduct the MCB test. The test results for the tourism dataset and mortality dataset are shown in Figure 6 and Figure 7, respectively.

Most hierarchy construction approaches outperform both base forecasts and the two-level hierarchy, except for three approaches on the mortality dataset. For both datasets, the simple average of randomised natural hierarchies ("FC-N") performs better than both natural hierarchy and all clustering-based hierarchies. Additionally, the simple average of 12 clustering-based hierarchies ("FC-C") significantly outperforms all other approaches. These results indicate that using a clustering-based hierarchy can improve forecast performance. However, forecast combination is even more beneficial, as evidenced by the high ranking of "FC-N" and "FC-R".

The approach based on hierarchical clustering outperforms the approach based on k-Medoids on the tourism dataset when using the same representation and distance metric, e.g., "TSF-HC" significantly ourperforms "TSF-ME". However, this is not consistently ob-

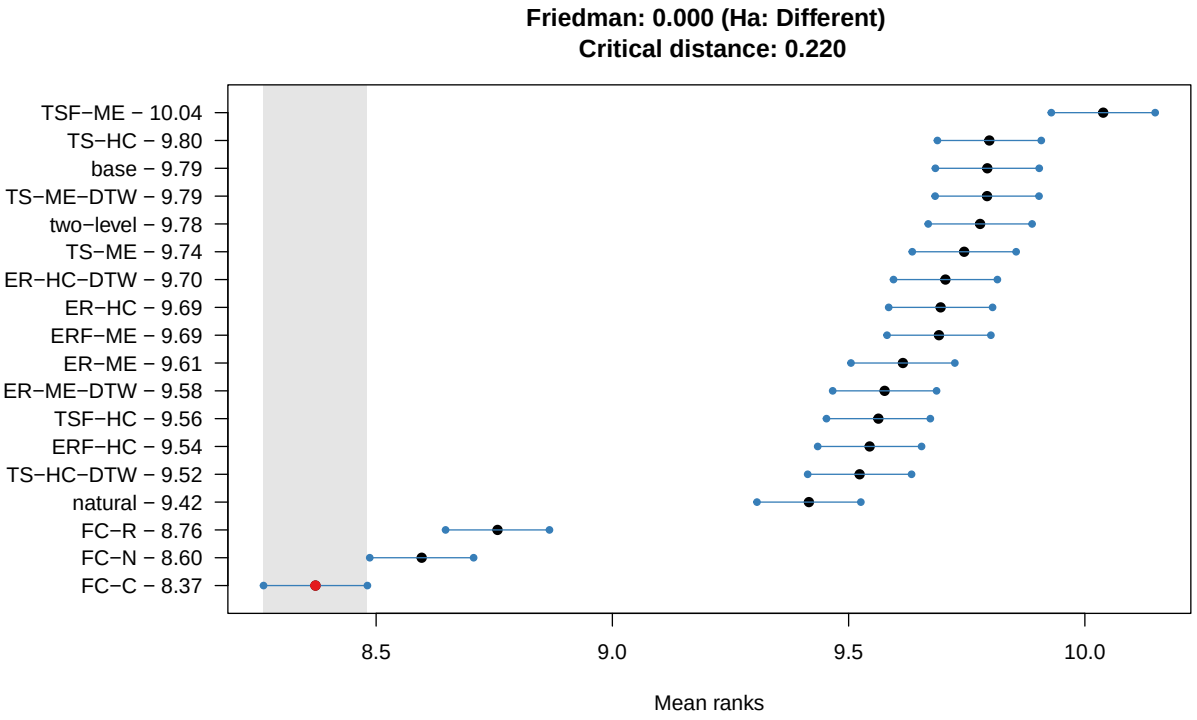

Figure 7: Average ranks and 95% confidence intervals based on the MCB Test for the 18 approaches on the mortality dataset.

served for every combination of representation and distance metric on the mortality dataset. We believe this discrepancy is due to different time series patterns at the bottom level of these two datasets. In the case of the mortality dataset, we combine rare causes of death that have suppressed values and record death counts at a national level. As a result, most series at the bottom level exhibit strong seasonality and trend. On the other hand, in the tourism dataset, the tourism demand is disaggregated for both travel purposes and geographical regions. This leads to numerous bottom-level series with high volatility and many zeros, making them more challenging to forecast. Therefore, having a hierarchy with more middle-level series can be more advantageous for reconciliation in the tourism dataset compared to the mortality dataset. Additionally, we suspect that the inferior performance of "FC-R" on the tourism dataset may be partly attributed to an insufficient number of middle-level series.

In most cases, the in-sample error representation outperforms the time series representation when using the same distance metric and clustering algorithms. However, this performance difference is not as significant as what was observed in Section 3.3. Intuitively, hierarchies constructed based on clustering in-sample error should be similar to random hierarchies when

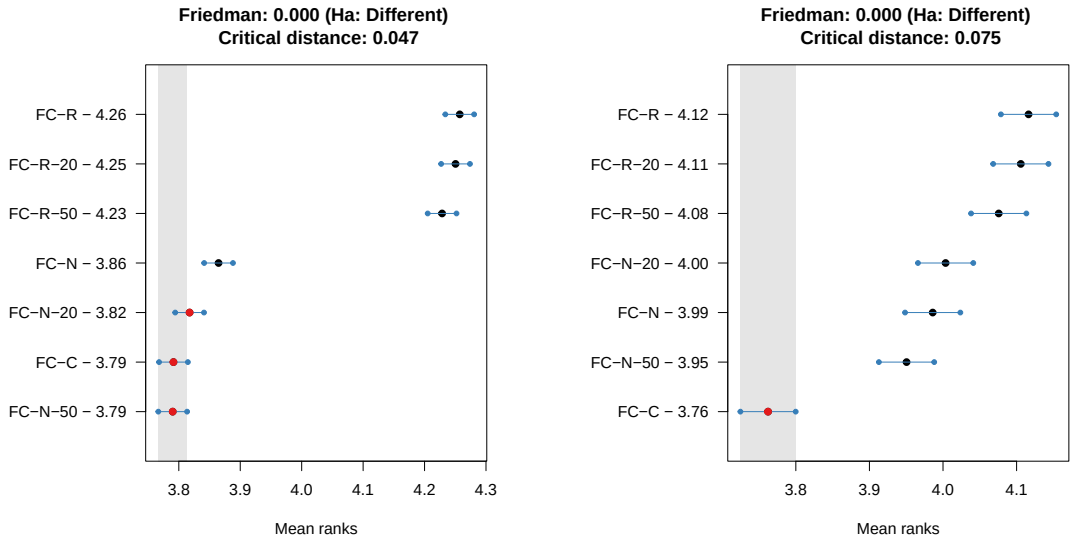

Figure 8: Average ranks and 95% confidence intervals based on the MCB Test of different number of random hierarchies on two datasets. Left and right panels display test results on tourism dataset and mortality dataset, respectively.

all base models are correctly specified. However, due to ubiquitous model misspecification in practice, vague patterns emerge in the in-sample error, which leads to contradictory results.

Effect of number of random hierarchies

In our previous experiments, we set the number of random hierarchies in "FC-R" and "FC-N" to 10 for fair comparisons with "FC-C". However, we believe that increasing the number of random hierarchies could further enhance performance. In this subsection, we explore the effects of using 20 and 50 random hierarchies, referred to as "FC-N-20", "FC-N-50", "FC-R-20", and "FC-R-50". We compare these variations with "FC-R", "FC-C", and "FC-N" using the same evaluation procedure described earlier. The results from the MCB test are presented in Figure 8, where the left panel shows results on the tourism dataset and the right panel displays results on the mortality dataset.

Generally, we observe that forecast performance improves as the number of random hierarchies increases, except for "FC-N-20" and "FC-N" on the mortality dataset. Surprisingly, on the tourism dataset, "FC-N-50" performs slightly better than "FC-C". It is important to note that employing more random hierarchies comes at a cost of increased computational resources. Therefore, one must carefully consider trade-offs between efficiency and performance when making decisions.

5. Conclusion

Acknowledgements

References

Aghabozorgi, S., Seyed Shirkhorshidi, A. and Ying Wah, T. (2015), 'Time-series clustering – A decade review', *Information Systems* **53**, 16–38.

Hollyman, R., Petropoulos, F. and Tipping, M. E. (2021), 'Understanding forecast reconciliation', *European Journal of Operational Research* **294**(1), 149–160.
URL: *https://www.sciencedirect.com/science/article/pii/S0377221721000199*

Hyndman, R. J., Kang, Y., Montero-Manso, P., Talagala, T., Wang, E., Yang, Y. and O'Hara-Wild, M. (2022), *tsfeatures: Time Series Feature Extraction*. R package version 1.1.0.9000.
URL: *https://pkg.robjhyndman.com/tsfeatures/*

Kaufman, L. and Rousseeuw, P. J. (1990), Partitioning Around Medoids (Program PAM), *in* 'Finding Groups in Data', John Wiley & Sons, Ltd, chapter 2, pp. 68–125.

Koning, A. J., Franses, P. H., Hibon, M. and Stekler, H. O. (2005), 'The M3 competition: Statistical tests of the results', *International Journal of Forecasting* **21**(3), 397–409.

Li, L., Kang, Y., Petropoulos, F. and Li, F. (2023), 'Feature-based intermittent demand forecast combinations: accuracy and inventory implications', *International Journal of Production Research* **61**(22), 7557–7572.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M. and Hornik, K. (2022), *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.4.
URL: *https://cran.r-project.org/web/packages/cluster/index.html*

Murtagh, F. and Legendre, P. (2014), 'Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?', *Journal of Classification* **31**(3), 274–295.
URL: *https://doi.org/10.1007/s00357-014-9161-z*

Petropoulos, F., Hyndman, R. J. and Bergmeir, C. (2018), 'Exploring the sources of uncertainty: Why does bagging for time series forecasting work?', *European Journal of Operational Research* **268**(2), 545–554.
URL: *https://www.sciencedirect.com/science/article/pii/S037722171830081X*

Pritularga, K. F., Svetunkov, I. and Kourentzes, N. (2021), 'Stochastic coherency in forecast reconciliation', *International Journal of Production Economics* **240**, 108221.
URL: *https://www.sciencedirect.com/science/article/pii/S0925527321001973*

Sakoe, H. and Chiba, S. (1978), 'Dynamic Programming Algorithm Optimization for Spoken Word Recognition', *IEEE Transactions on Acoustics, Speech, and Signal Processing* **26**(1), 43–49.

Tiano, D., Bonifati, A. and Ng, R. (2021), FeatTS: Feature-based Time Series Clustering, *in* 'Proceedings of the 2021 International Conference on Management of Data', SIGMOD '21, Association for Computing Machinery, New York, NY, USA, pp. 2784–2788.

Tibshirani, R., Walther, G. and Hastie, T. (2002), 'Estimating the number of clusters in a data set via the gap statistic', *Journal of the Royal Statistical Society Series B: Statistical Methodology* **63**(2), 411–423.

Wang, X., Kang, Y., Petropoulos, F. and Li, F. (2022), 'The uncertainty estimation of feature-based forecast combinations', *Journal of the Operational Research Society* **73**(5), 979–993. **URL:** *https://doi.org/10.1080/01605682.2021.1880297*

Warren Liao, T. (2005), 'Clustering of time series data—a survey', *Pattern Recognition* **38**(11), 1857–1874.

Wickramasuriya, S. L., Athanasopoulos, G. and Hyndman, R. J. (2019), 'Optimal Forecast Reconciliation for Hierarchical and Grouped Time Series Through Trace Minimization', *Journal of the American Statistical Association* **114**(526), 804–819.