Reconciling forecasts for hierarchical and grouped time series

Rob J Hyndman, George Athanasopoulos

Saturday, 24 May 2014

Time series can often be naturally disaggregated in a hierarchical structure using attributes such as product type. For example, the total number of bicycles sold by a cycling warehouse can be disaggregated into a hierarchy of bicycle types. Such a warehouse will sell road bikes, mountain bikes, children bikes or hybrids. Each of these can be disaggregated into finer categories. Children's bikes can be divided into balance bikes for children under 4 years old, single-geared bikes for children between 4 and 6 and multi-geared bikes for children over the age of 6. Hybrid bikes can be divided into city, commuting, comfort, and trekking bikes; and so on. Such disaggregation imposes a hierarchical structure. We refer to these as hierarchical time series.

To take another example, a manufacturing company may track sales by country, and within each country by region, and within each region by sales outlet. Again, this is a hierarchical structure, and the complete collection of sales data can be thought of as hierarchical time series.

We get a similar structure when the two attributes above are combined. For example a bicycle manufacturer may disaggregate sales by product and also by geographical location. In this case we still have a hierarchical structure however the structure does not naturally disaggregate in a unique way. We can disaggregate by product type and then geographical location but also vice versa. We usually refer to these structures as "grouped time series". For the ease of the exposition in this paper we will concentrate on hierarchical time series however when necessary we will comment on grouped time series.

The optimal reconciliation method we discuss here can handle both hierarchical and grouped time series in contrast to traditional forecasting approaches which completely ignore the non-unique structure of grouped time series.

Figure 1: *A two level hierarchical tree diagram.*

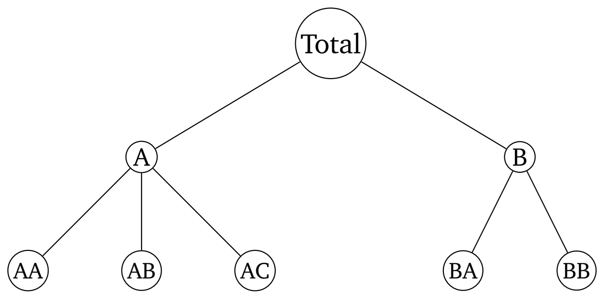

Figure 1 shows a very simple hierarchy. At the top of the hierarchy is the "Total", the most aggregate level of the data. In this example, the total is split into two categories, which are in turn split into three and two sub-categories respectively. If y_X denotes the value of the time series at node X, then $y_{AA} + y_{AB} + y_{AC} = y_A$, $y_{BA} + y_{BB} = y_B$ and $y_A + y_B = y_{\text{Total}}$.

The data within the hierarchical structure will add up appropriately, but the forecasts generally will not. For example, consider only the first level of the hierarchy in Figure 1. If we forecast the time series for each node independently, the forecasts for y_A and y_B will not necessarily add up to the forecasts for y_{Total}. This can cause confusion when making decisions based on the forecasts (e.g. how much stock of different types to ship, or how much money to allocate to different divisions of a company). Consequently, it is standard to require hierarchical forecasts to add up appropriately (a feature we call "aggregate consistency") — the sum of the forecasts must equal the forecast of the sum.

Traditional forecasting approaches

The simplest method for hierarchical forecasting is the **bottom-up approach**. In this approach, we first generate independent forecasts for each series at the bottom level of the hierarchy and then aggregate these to produce forecasts for the upper levels of the hierarchy.

For example, for the hierarchy of Figure 1 we first generate forecasts for the bottom level series: $\hat{y}_{\text{AA}}$, $\hat{y}_{\text{AB}}$, $\hat{y}_{\text{AC}}$, $\hat{y}_{\text{BA}}$ and $\hat{y}_{\text{BB}}$. Aggregating these up the hierarchy, we get forecasts for the rest of the series: $\hat{y}_{\text{A}} = \hat{y}_{\text{AA}} + \hat{y}_{\text{AB}} + \hat{y}_{\text{AC}}$, $\hat{y}_{\text{B}} = \hat{y}_{\text{BA}} + \hat{y}_{\text{BB}}$ and $\hat{y}_{\text{Total}} = \hat{y}_{\text{A}} + \hat{y}_{\text{B}}$.

An alternative is to start with forecasting the most aggregated series (the Total), and then disaggregate the forecasts down the hierarchy. This is known as a **top-down approach**. There are several possible disaggregation approaches, the most commonly used being to disaggregate by the average proportions computed on the historical data. For example, the forecast for node A would be $\hat{y}_A = p_A \hat{y}_{\text{Total}}$ where p_A is the average of the proportion of A making up the Total in the historical data.

In general, top-down approaches based on historical proportions tend to produce less accurate forecasts at lower levels of the hierarchy than bottom-up approaches because the historical proportions used for disaggregation do not take account of how those proportions might change over time. To address this issue, proportions based on forecasted rather than historical data can be used, as described by Athanasopoulos et al. (2009). Using forecasted proportions, p_A would be the proportion of the forecast of A relative to the forecast of the sum of A and B.

A third approach that is widely used in practice is known as "middle-out" forecasting. In this case, a middle level of the hierarchy is selected and forecasts are computed for each node at that level. Levels above that are computed by aggregation, and levels below that by disaggregation. For example, in the hierarchy of Figure 1, forecasts may be calculated for A and B (the middle level). Then the Total forecast is obtained by adding these two forecasts, and the forecasts for the bottom level would be obtained by disaggregating the A and B forecasts, as in a top-down approach.

Until recently, these three approaches (bottom-up, top-down, and middle-out) were the only possible ways of ensuring that forecasts across the hierarchy added up appropriately.

Optimal reconciliation

Hyndman et al. (2011) proposed a new approach to handling hierarchical time series. They suggested that forecasts for all nodes should be calculated, ignoring the fact that they will not add up. Then a reconciliation step is used to adjust the independent forecasts, giving revised forecasts which add up in a way that is consistent with the hierarchical structure. The revised forecast at each node is a weighted average of the forecasts from all nodes.

Further, they derived an optimal combination of the independent forecasts that gives the best possible reconciled forecasts taking account of all the forecasts at all levels. Unlike any other existing method, this approach uses all the information available within a hierarchy. It allows for correlations and interactions between series at each level of the hierarchy, it accounts for ad hoc adjustments of forecasts at any level, and, provided the base forecasts are unbiased, it produces unbiased revised forecasts.

In empirical comparisons, the optimal reconciliation approach tends to give more accurate forecasts than any of the bottom-up, top-down or middle-out methods, because it uses more of the available information than the other approaches.

The one difficulty with using the optimal reconciliation approach is that the reconciliation weights can be potentially difficult to compute. To overcome this problem, Hyndman et al. (2011) suggested ignoring the scale of the time series, and the correlations between time series, so that the weights depend only on the hierarchical structure and not on the actual observed data. That way, the weights can be computed once for the hierarchy, and reapplied whenever forecasts need to be reconciled.

We write the independent forecast for the time series at node X as $\hat{y}_X$, and the revised forecast for the time series at node X as $\tilde{y}_X$. Then for the hierarchy in Figure 1, the weights are given in the following table.

	$\hat{y}_{\text{Total}}$	$\hat{y}_A$	$\hat{y}_B$	$\hat{y}_{AA}$	$\hat{y}_{AB}$	$\hat{y}_{AC}$	$\hat{y}_{BA}$	$\hat{y}_{BB}$
$\tilde{y}_{\text{Total}}$	0.586	0.310	0.276	0.103	0.103	0.103	0.138	0.138
$\tilde{y}_A$	0.310	0.517	-0.207	0.172	0.172	0.172	-0.103	-0.103
$\tilde{y}_B$	0.276	-0.207	0.483	-0.069	-0.069	-0.069	0.241	0.241
$\tilde{y}_{AA}$	0.103	0.172	-0.069	0.724	-0.276	-0.276	-0.034	-0.034
$\tilde{y}_{AB}$	0.103	0.172	-0.069	-0.276	0.724	-0.276	-0.034	-0.034
$\tilde{y}_{AC}$	0.103	0.172	-0.069	-0.276	-0.276	0.724	-0.034	-0.034
$\tilde{y}_{BA}$	0.138	-0.103	0.241	-0.034	-0.034	-0.034	0.621	-0.379
$\tilde{y}_{BB}$	0.138	-0.103	0.241	-0.034	-0.034	-0.034	-0.379	0.621

For example, the revised forecast for node A is equal to a combination of all the original forecasts with weights taken from the second row of this table:

$$\tilde{y}_A = 0.310\hat{y}_{\text{Total}} + 0.517\hat{y}_A - 0.207\hat{y}_B + 0.172\hat{y}_{AA} + 0.172\hat{y}_{AB} + 0.172\hat{y}_{AC} - 0.103\hat{y}_{BA} - 0.103\hat{y}_{BB}.$$

Notice that the weights can be negative. This is to effectively remove unwanted components. For example, in the above equation, the value of $\hat{y}_{\text{Total}}$ contains information from node B as well as node A. Consequently, terms associated with node B have negative coefficients ($-0.207\hat{y}_B$, $-0.103\hat{y}_{BA}$ and $-0.103\hat{y}_{BB}$) to offset the inclusion of the Total.

The optimal reconciliation weights are designed so that the revised forecasts will always add up appropriately. That is $\tilde{y}_{\text{A}} = \tilde{y}_{\text{AA}} + \tilde{y}_{\text{AB}} + \tilde{y}_{\text{AC}}$, $\tilde{y}_{\text{B}} = \tilde{y}_{\text{BA}} + \tilde{y}_{\text{BB}}$ and $\tilde{y}_{\text{Total}} = \tilde{y}_{\text{A}} + \tilde{y}_{\text{B}}$. This can easily be checked in the above table.

The weights are obtained by solving a large linear regression problem, where all the independent forecasts from all nodes are regressed against a set of dummy variables indicating which of the bottom level series contribute to each node. They are "optimal" in the sense that the mean squared reconciliation error, computed using the differences between the reconciled and independent forecasts, is as small as possible. Thus, the weights enable us to revise the independent forecasts by the smallest possible amount until they add up.

Implementation in R

To implement the method, it is necessary to be able to automatically generate the independent forecasts (using any time series forecasting algorithm), and to be able to generate the optimal weights by solving the large linear regression problem. Many software packages allow for automatic forecasting. Generating the weights is tricky as standard regression software will not handle the size of the problem when there are many thousands of time series involved. We have developed some fast and efficient procedures for handling this in R.

The traditional approaches and the optimal reconciliation approach are all implemented in the hts package for R (Hyndman et al., 2014). The package also allows for automatic forecasting of time series (using either exponential smoothing or ARIMA modeling) (Kolassa and Hyndman, 2010).

Suppose we have a collection of time series with the hierarchical structure given in Figure 1. Then the following R code will compute bottom-up forecasts for all series in the hierarchy, using exponential smoothing for the bottom-level series, and aggregation for all other series.

R code

```
# bts is a time series matrix containing the bottom level series
# The first three series belong to one group, and the last two series
# belong to a different group
y <- hts(bts, nodes=list(2, c(3,2)))
fc <- forecast(y, method="bu", fmethod="ets", h=12)
```

The `nodes` argument specifies the hierarchical structure, the `method` argument specifies we want "bottom-up" forecasting, the `fmethod` argument shows we want to use exponential smoothing, and `h=12` indicates we want to forecast 12 steps ahead.

Alternatively, if we wanted to use the optimal reconciliation approach with ARIMA models, we would replace the last line with the following command.

R code

```
fc <- forecast(y, method="comb", fmethod="arima", h=12)
```

(To see all the available options, use `help(forecast.gts)`.)

That's it! Everything else is handled automatically. The reconciliation is very fast, even for tens of thousands of time series. However, the forecasts for each node can be time-consuming as each individual series must be modelled and forecast independently.

Example: Australian tourism hierarchy

We demonstrate the reconciliation of independent forecasts using optimal reconciliation weights with a simple example. We forecast Australian domestic tourism demand, measured by the visitor nights spent away from home by Australians. At the first level of the hierarchy we consider visitor nights for the three largest states of Australia, namely New South Wales, Victoria and Queensland. The remaining states are classified as "Other". At the second level we consider the capital cities of each state, namely, Sydney, Melbourne, Brisbane and Gold Coast (a coastal city and a major tourism attraction near Brisbane), and the other five capital cities of the remaining states and territories of Australia.

We estimate ETS models (models that generate exponential smoothing forecasts) using the automatic algorithm in the forecast package (Hyndman, 2014) with data to the end of 2011, and use these models to forecast the four quarters of 2012 for each series. We reconcile the forecasts using the optimal reconciliation weights. The plots below demonstrate the difference between the initial independent forecasts and the reconciled forecasts. The MAPE for the optimal reconciliation forecasts (over all the series and over the four quarters of the 2012) is 5.35% compared to 5.58% for the bottom-up forecasts and 5.7% for the independent unreconciled forecasts. The example demonstrates the basic idea of the optimal reconciliation approach which is to forecast each series in the hierarchy as accurately as possible, taking advantage of the characteristics of each, and then revise/alter these forecasts by the smallest possible amount so that they become reconciled.

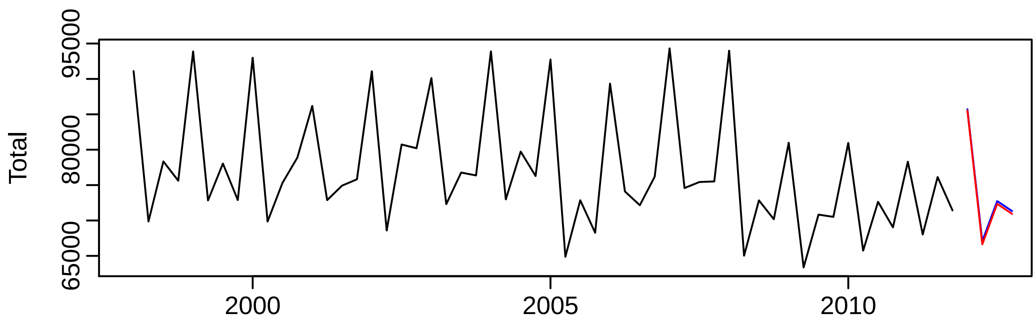

Figure 2: *Total Australian visitor nights over the period 1998–2012. The blue line shows the independent forecasts for the four quarters of 2012, and the red line shows the corresponding reconciled forecasts.*

Pros and cons and future directions

The most commonly applied method of hierarchical forecasting is the bottom-up approach due to its simplicity. Its greatest advantage is that we are modelling and forecasting the data at the most disaggregate level and therefore we do not lose any information due to aggregation. However, we should keep in mind that the more we thin our data by disaggregation the noisier it becomes.

The simplicity of the application of the top-down approach using historical proportions is its greatest attribute. Here we only need to build one model for the most aggregate level and we then distribute the forecasts generated from this model to lower levels. In general top-down approaches seem to produce quite reliable forecasts for the aggregate levels, and they are very useful with low count data. A disadvantage of this approach is the loss of information due to aggregation. Using historical proportions we are unable to capture and take advantage of individual series characteristics such as

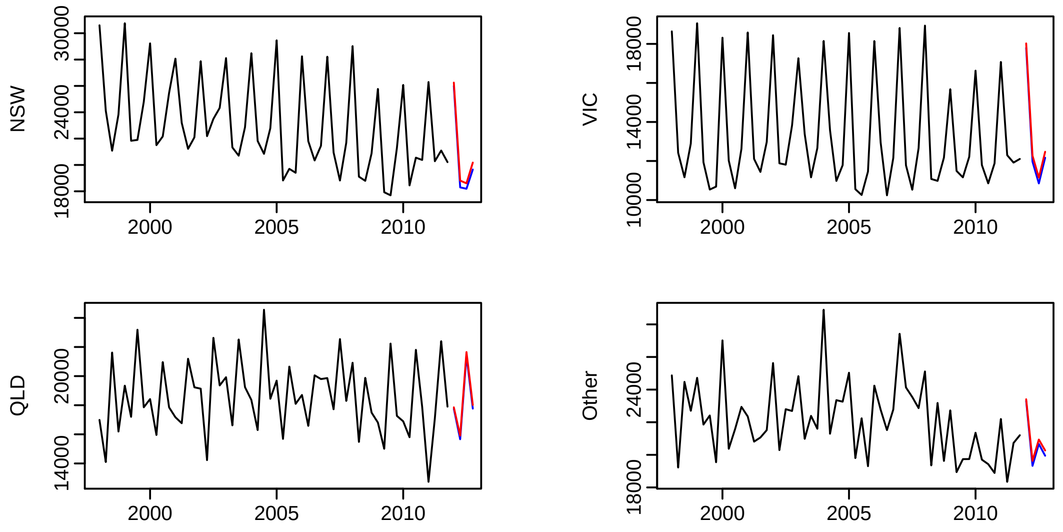

Figure 3: *Visitor nights by State over the period 1998–2012. The blue line shows the independent forecasts for the four quarters of 2012, and the red line shows the corresponding reconciled forecasts.*

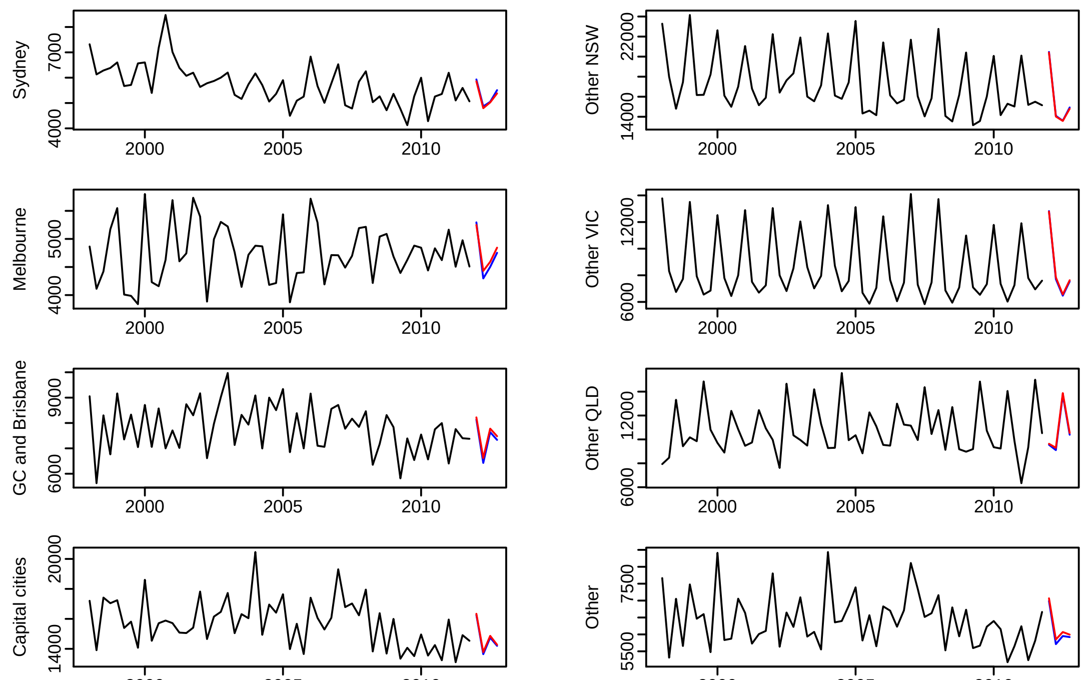

Figure 4: *Visitor nights by Capital cities over the period 1998–2012. The blue line shows the independent forecasts for the four quarters of 2012, and the red line shows the corresponding reconciled forecasts.*

time dynamics, special events, etc. For example, the tourism data above are highly seasonal. The seasonal pattern of visitor nights may vary across series depending on the tourism destination, e.g., a ski resort versus a beach resort. Using forecasted proportions instead of historical proportions will improve this aspect when using a top-down approach.

However, Hyndman et al. (2011) derive an important theoretical result. They show that no matter what proportions are used, disaggregation of forecasts inevitably introduces bias. Therefore, even if our independent forecasts are unbiased, disaggregating them to get forecasts at lower levels (as is required with any top-down or middle-out approach) will generate biased forecasts for the lower levels. We consider this to be the greatest disadvantage of all top-down and middle out approaches.

In contrast to all previously existing methods for forecasting hierarchical time series, the optimal reconciliation approach allows us to take advantage of all the available data, and model the dynamics of each individual series. One feature that should be highlighted here is that with grouped data, where alternative disaggregation paths exist, the optimal reconciliation approach allows us to forecast every time series from every disaggregation path. Reconciled forecasts are then calculated for the whole hierarchy. Unlike all top-down and middle-out approaches, the optimal reconciliation weights do not introduce any bias to the reconciled forecasts.

One limitation of the existing optimal reconciliation weights is that they are calculated based only on the structure of the hierarchy, and do not depend on the data within the hierarchy. We are currently developing better procedures for determining these weights, so that the scale of the original data, and the correlations between series, can be also accounted for.

We are also currently looking at some possible approaches for computing prediction intervals for the reconciled forecasts. It is currently not possible to produce proper prediction intervals for bottom-up, top-down, or middle-out forecasts either, and we hope to develop methods to allow the calculation of prediction intervals for these approaches also.

References

Athanasopoulos, G., R. A. Ahmed and R. J. Hyndman (2009) Hierarchical forecasts for australian domestic tourism, *Int. J. Forecast.*, **25**(1), 146–166.

Hyndman, R. J. (2014) *forecast: Forecasting functions for time series*.
URL: *cran.r-project.org/package=forecast*.

Hyndman, R. J., R. A. Ahmed, G. Athanasopoulos and H. L. Shang (2011) Optimal combination forecasts for hierarchical time series, *Comput. Stat. Data Anal.*, **55**(9), 2579–2589.

Hyndman, R. J., A. J. Lee and E. Wang (2014) *hts: Hierarchical time series*.
URL: *cran.r-project.org/package=hts*.

Kolassa, S. and R. J. Hyndman (2010) Free open-source forecasting using R, *Foresight: the International Journal of Applied Forecasting*, **17**, 19–24.