Abstract

In many real world data, time series are often hierarchically organized. Based on features such as products or geography, time series can be aggregated and disaggregated at several different levels. The so-called ‘hierarchical time series’ are often forecast using simple top-town or bottom-up approaches. In this paper, we build a probabilistic model that involves dynamically evolving latent variables to capture the proportion changes in time series at each hierarchy. We derive the variational Bayesian expectation maximisation (VBEM) algorithm under the new model. In our algorithm, we implement the posterior inference in a sequential manner that significantly decreases computational overhead common in large hierarchical time series data. Furthermore, unlike the standard EM algorithm that provides point estimates of model parameters, our algorithm yields the distribution over the model parameters, which give us an insight to which subset of features yields the proportion changes of the time series. Simulation results show that our method significantly outperforms other methods in prediction.

1. Introduction

Time series in business and economics are often organized in a hierarchical structure based on dimensions such as products or regions. Forecasting the hierarchical time series is an important task in a number of industrial sectors (Sbrana & Silvestrini, 2013; Fischer et al., 2013; Kalchschmidt et al., 2006; Zotteri & Kalchschmidt, 2007; Fliedner, 1999). For instance, companies that offer a broad range of items or services to their customers need to plan their future supply process in order to minimize potential costs (e.g., inventory costs) (Kerkkaenen et al., 2009). A similar problem occurs in governmental budgeting that needs to be optimized throughout hierarchically organized departments. As an example, military budgeting in the context of hierarchical time series forecasting can be found in (Moon et al., 2013; 2012).

1.1. Motivation

The main challenge in forecasting hierarchical time series is that various components at different levels of a hierarchy can interact in a complex manner: small changes at a given level of hierarchy can largely affect the time series at other levels. Furthermore, forecasting each times series individually can be time consuming and computationally intensive for large datasets. Consequently, there is a dire need for rapid, efficient, and automated forecasting methods that exploit the hierarchy in the time series to obtain better prediction performance.

1.2. Prior Work

Different approaches to forecasting hierarchical time series data are summarized in (Athanasopoulos et al., 2009; Hyndman et al., 2011) and can be categorized into three main categories: (1) top-down methods; (2) bottom-up methods; and (3) alternative statistical methods. In the following, we review each of these categories in detail.

1) Top-down (TD) approaches: distribute the top-level forecasts down the hierarchy using the historical proportions of the data (Gross & Sohl, 1990; Fliedner, 1999). As a result, only the top level forecast of the time series is needed. Examples of proportions are: (1) average historical proportions that are the average of the historical proportions of the bottom level series relative to the top level series over a certain period; and (2) proportions of the his-
2. Hierarchical Bayesian dynamic proportions model

2.1. Observation model

Suppose an observation of the top level at time \( t \) denoted by \( n_t \) is disaggregated into \( k \) different categories. We model the observations at the subsequent level as multinomial random variables, \( y_t = [y^1_t, \ldots, y^k_t]^T \) such that \( n_t = \sum_{j=1}^{k} y^j_t \). Conditioned on latent states \( z_t \in \mathbb{R}^k \), the likelihood of the observed data is given by

\[
p(y_t|z_t, n_t) = \text{Mu}(\pi(z_t)|n_t), \quad (1)
\]

\[
= \frac{n_t!}{y^1_t! \cdots y^k_t!} \prod_{j=1}^{k} [\pi^j(z_t)]^{y^j_t}, \quad (2)
\]

where each proportion is given by the softmax function as follows:

\[
\pi^j(z_t) = \frac{\exp(z^j_t)}{\sum_{j=1}^{k} \exp(z^j_t)}. \quad (3)
\]

2.2. Latent states

The latent states in the HB-DP model evolve linearly with time:

\[
p(z_t|z_{t-1}) = \mathcal{N}(Az_{t-1}, \Sigma), \quad (4)
\]

where \( A \in \mathbb{R}^{k \times k} \) is the dynamics matrix. The evolution noise covariance is denoted by \( \Sigma \in \mathbb{R}^{k \times k} \). Therefore the parameters in our model are \( \theta = \{A, \Sigma\} \).
2.3. Priors on parameters

Typically, it is expected that future proportion of a given category would depend on the current proportions of few related categories. To capture this, we impose the ARD prior on each row \( a_j \) of the dynamic matrix \( A \):

\[
p(a_j|\alpha) = \mathcal{N}(a_j|0, \text{diag}(\alpha_1, \cdots, \alpha_k)^{-1}). \tag{5}
\]

This prior induces many zeros (i.e., sparse) in the estimate of \( A \), which tells us which elements in \( A \) contribute dynamical proportion changes. If we share the hyperparameters \( \alpha \) across rows the resulting prior on the dynamic matrix \( A \) is given by

\[
p(A|\alpha) = \prod_{j=1}^{k} \mathcal{N}(a_j|0, \text{diag}(\alpha)^{-1}). \tag{6}
\]

This prior on \( A \) could be less flexible than an independent ARD prior on each row of \( A \) using different precisions for each row, i.e., \( a_j \sim \mathcal{N}(0, \text{diag}(\alpha_j)^{-1}) \). Our choice for the shared precisions across rows of \( A \) is based on that: (1) having too many hyperparameters can be harmful due to over-fitting; (2) our interest is not finding maximally sparse \( A \), but finding which dimension in the latent variables contributes proportion changes in the time series.

We impose the Gaussian prior on the initial latent states:

\[
p(z_0) = \mathcal{N}(z_0|\mu_0, \Sigma_0). \tag{7}
\]

In total, the hyperparameters are \( \phi = \{\alpha, \mu_0, \Sigma_0\} \).

2.4. Approximate posterior

We assume the approximate posterior over the parameter \( \theta \) and latent variables, \( q(\theta, z_{0:T}) \), is factorized in the following way:

\[
p(\theta, z_{0:T}|y_{1:T}) \approx q(\theta, z_{0:T}) = q_0(\theta)q_x(z_{0:T}). \tag{8}
\]

where the approximate posterior over the parameters is further factorized as

\[
q_0(\theta) = q_{A|\Sigma}(A|\Sigma)q_{\Sigma}(\Sigma), \quad q_x(z_{0:T}) = q_{A|\Sigma}(A|\Sigma)\delta(\Sigma - \Sigma_{ML}). \tag{9}\tag{10}
\]

We approximate the posterior where \( \Sigma \) coincides the ML estimate of \( \Sigma \) for simplicity.

2.5. Variational lower bound

Using the approximate posterior, we can lower bound the marginal likelihood of the observations by the KL divergence between the approximate posterior \( q(\theta, z_{0:T}) \) and the true posterior \( p(\theta, z_{0:T}, y_{1:T}) \):

\[
\log p(y_{1:T}) \geq \int d\theta \ dz_{0:T} q(\theta, z_{0:T}) \log \frac{p(\theta, z_{0:T}, y_{1:T})}{q(\theta, z_{0:T})}.
\]

We maximize the lower bound by iterating the variational Bayesian expectation maximization (VBEM) algorithm (Beal, 2003), which consists of: (1) variational Bayesian expectation (VBE) step for computing \( q_x(z_{0:T}) \):

\[
q_x(z_{0:T}) \propto \exp \left[ \int d\theta q_0(\theta) \log p(z_{0:T}, y_{1:T}|\theta) \right], \tag{11}
\]

and (2) variational Bayesian maximization (VBM) step for computing \( q_0(\theta) \):

\[
q_0(\theta) \propto p(\theta) \exp \left[ \int d z_{0:T} q_x(z_{0:T}) \log p(z_{0:T}, y_{1:T}|\theta) \right]. \tag{12}
\]

In each iteration, we also update the hyperparameters by computing the derivatives of the lower bound given \( q(\theta, z_{0:T}) \) with respect to each hyperparameter.

3. Variational Bayesian EM

3.1. VBE step

In VBE step, we compute

\[
\log q_x(z_{0:T}) = \mathbb{E}_{q_0(\theta)} \log p(z_{0:T}, y_{1:T}|\theta) + \text{const}, \tag{13}
\]

where the integrand in eq. 13, the so-called complete-data log likelihood, is written by

\[
\sum_{t=1}^{T} \{ \log p(y_t|z_t) + \log p(z_t|z_{t-1}, \theta) \}, \tag{14}
\]

which tells us that the log posterior over latent variables is quadratic in each \( z_t \). This enables us to use the sequential forward/backward message passing algorithm (see Fig. 1) to compute the posterior over latent variables in the following.

Forward message (filtering)

The forward message at each time is given by

\[
\alpha(z_t) \triangleq p(z_t|y_{1:t}),
\]

\[
= \mu_{f_{t-1}>z_t}(z_t),
\]

\[
= \int f_t(z_{t-1}, z_t) \alpha(z_{t-1}) dz_{t-1},
\]

\[
\times p(y_t|z_t) \times \int \exp \left( \mathbb{E}_{q_0(\theta)} \log p(z_t|z_{t-1}) \right) \alpha(z_{t-1}) dz_{t-1},
\]

where we approximate \( \alpha(z_{t-1}) \) to a Gaussian:

\[
\alpha(z_{t-1}) \approx \mathcal{N}(\mu_{t-1}, \Sigma_{t-1}), \tag{15}
\]

and assume \( q_0(z_0) = \mathcal{N}(\mu_0, \Sigma_0) \). After some algebraic manipulation, the forward message is given by

\[
\alpha(z_t) \approx p(y_t|z_t)\mathcal{N}(z_t|\mu_t, \Sigma_t), \tag{16}
\]
Due to the multinomial likelihood term, eq. 16 is not Gaussian, so we approximate eq. 16 as a Gaussian in \([z_t, z_{t+1}]^T\), and extract those parts that correspond to \(z_t\) to approximately compute the integral.

The first derivative of the logarithm of the integrand denoted by \(\Phi(z_t, z_{t+1})\) is given by

\[
\frac{\partial \Phi(z_t, z_{t+1})}{\partial z_t z_{t+1}} = \begin{bmatrix}
-\langle A \rangle^\top \Sigma^{-1} \langle A \rangle & -\langle A \rangle^\top \Sigma^{-1} \\
\Sigma^{-1} \langle A \rangle & -\Sigma^{-1} + \Gamma_{t+1}^{-1}
\end{bmatrix}
\]

where we denote \(W(z_{t+1})\) by \(W_{t+1}\). Using Schur complement, we obtain the covariance \(\Gamma_t\) by

\[
\Gamma_t^{-1} = \langle A \rangle^\top \Sigma^{-1} \langle A \rangle - \langle A \rangle^\top \Sigma^{-1} \Gamma_{t+1}^{-1} \Sigma^{-1} \langle A \rangle.
\]

where \(\Gamma_{t+1}^{-1} = \Sigma^{-1} + \Gamma_t^{-1} + W_{t+1}\).

**Computing posterior marginals**

Using the forward and backward messages, we can compute the posterior marginals for the latent variables. First, we define

\[
\gamma(z_t) \triangleq p(z_t|y_{1:T}), \quad \propto p(z_t|y_{1:T})p(y_{t+1:T}|z_t) = \alpha(z_t)\beta(z_t),
\]

where \(\alpha(z_t) \approx N(\mu_t, V_t)\).

Second, we also define the joint posterior between neighboring (in time) latent variables

\[
\xi(z_{t-1}, z_t) \triangleq p(z_{t-1}, z_t|y_{1:T}), \quad \propto \alpha(z_{t-1})p(y_t|z_t)
\]

\[
\exp \left( \mathbb{E}_{q_{\theta}}(\log p(z_t|z_{t-1})) \right) \beta(z_t).
\]

We approximate the joint distribution over \(z_{t-1}\) and \(z_t\) to a Gaussian. The second derivative of the logarithm of \(\xi(z_{t-1}, z_t)\) is given by

\[
-\left[ \langle A \rangle^\top \Sigma^{-1} \langle A \rangle + V_{t-1}^{-1} - \langle A \rangle^\top \Sigma^{-1} \Gamma_{t+1}^{-1} \right],
\]

where we denote \(W(\mu_t)\) by \(W_t\). Using the Schur complement, we can obtain the cross covariance of \((z_{t-1}, z_t)\).
3.2. VBM step

In VBM step, we compute \( q_{0}(\theta) \) by extracting all the terms in \( \log p(z_{0:T}, y_{1:T}|\theta) \) that depend on \( \theta \) and then taking the expectation over \( z_{0:T} \):

\[
\log q_{0}(\theta) = \mathbb{E}_{q_{0}(z_{0:T})} [\log p(z_{0:T}|\theta)] + \log p(\theta) + \text{const.}
\]

where the first term on RHS is given by

\[
-\frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} \sigma_{ij}^{2} (a_{i}^{T} W_{A} a_{j} - 2 a_{i}^{T} S_{A} e_{j}) \tag{27}
\]

where \( \sigma_{ij} \) is the \( i, j \)th element of ML estimate\(^1\) of the latent noise covariance \( \Sigma_{,} \) and \( e_{j} \) is the unit vector where the \( j \)th element is 1. The sufficient statistics of latent variables are denoted by \( W_{A} \) and \( S_{A} \):

\[
W_{A} = \sum_{t=1}^{T} < z_{t-1} z_{t-1}^{T} >, \quad S_{A} = \sum_{t=1}^{T} < z_{t-1} z_{t}^{T} > .
\]

Clearly, the joint posterior over the rows of \( A \) does not factorize. However, to avoid computational burden finding the joint posterior for large data, we approximate the posterior over \( A \) as\(^2\)

\[
q_{0}(\theta) = \prod_{j=1}^{T} \mathcal{N}(a_{j}|\mu_{a_{j}}, \Sigma_{A}),
\]

where the covariance and mean are given by

\[
\Sigma_{A}^{-1} = W_{A} + \text{diag}(\alpha), \tag{28}
\]

\[
\mu_{a_{j}} = \Sigma_{A} S_{A} e_{j}. \tag{29}
\]

3.3. Hyperparameter estimation

We update hyperparameters so as to maximize the variational lower bound on the marginal likelihood (eq. 11). The lower bound can be simplified as\(^3\)

\[
\log p(y_{1:T}|\phi) \geq \log Z' - KL(q(\theta)||p(\theta)), \tag{30}
\]

where

\[
Z' = \int dz_{0:T} \exp \left( \mathbb{E}_{q_{0}(\theta)} \log p(z_{0:T}, y_{1:T}|\theta) \right). \tag{31}
\]

The KL divergence between \( q(\theta) \) and \( p(\theta) \) is given by

\[
\sum_{j=1}^{k} \int da_{j} \mathcal{N}(a_{j}|\mu_{a_{j}}, \Sigma_{A}) \log \mathcal{N}(a_{j}|\mu_{a_{j}}, \Sigma_{A}) \mathcal{N}(a_{j}|0, \text{diag}(\alpha^{-1}))
\]

\[
= \sum_{j=1}^{k} \left( -\frac{1}{2} \log |\text{diag}(\alpha)| \Sigma_{A}^{2} + \frac{1}{2} \text{Tr} \left[ \text{diag}(\alpha) (\Sigma_{A}^{2} - \text{diag}(\alpha)^{-1} + \mu_{a_{j}}^{2}) \right] \right).
\]

The first derivative expression of \( \alpha \) gives us the following update:

\[
\alpha^{-1} = \frac{1}{k} \text{diag} \left[ \sum_{j=1}^{k} (\Sigma_{A} + \mu_{a_{j}} \mu_{a_{j}}^{T}) \right]. \tag{32}
\]

Similarly, we update the hyperparameters for initial states by

\[
\mu_{0} = \omega_{0}, \tag{33}
\]

\[
\Sigma_{0} = Y_{0,0}. \tag{34}
\]

The summary of the entire algorithm is given below:

**Algorithm 1** VBEM for dynamic proportion models

Given data \( D \) and initial \( q(\theta) \), iterate the following:

1. VBE step: Given \( q(\theta) \), compute forward (\( \alpha \)), backward (\( \beta \)) and marginal (\( \gamma \)) messages and the crosscovariation of messages at each time.

2. VBM Step: Given \( q(z_{0:T}) \), update \( q(\theta) \).

3. Update hyperparameters.

Until convergence.

4. Prediction

Given \( p(z_{T}|y_{1:T}) \approx \mathcal{N}(\hat{\mu}_{T}, \hat{V}_{T}) \), we want to make a prediction on the time series in each level of hierarchy by

\[
p(y_{T+1}|y_{1:T}) = \int p(y_{T+1}|z_{T+1}) p(z_{T+1}|y_{1:T}) dz_{T+1}, \tag{35}
\]

where the second part of the integrand is

\[
p(z_{T+1}|y_{1:T}) = \int \exp \left( \mathbb{E}_{q_{0}(\theta)} \log p(z_{T+1}|y_{1:T}) \right)
\]

\[
\mathcal{N}(z_{T+1}|\hat{\mu}_{T+1}, \hat{V}_{T+1}) dz_{T+1}, \tag{36}
\]

where the mean and covariance are given by

\[
\hat{V}_{T+1}^{-1} = \Sigma + \langle A \rangle \hat{V}_{T}(A)^{T},
\]

\[
\hat{\mu}_{T+1} = \langle A \rangle \hat{\mu}_{T}.
\]

The integral in eq. 35 is not analytically tractable due to the non-Gaussian likelihood term. One can do is to draw samples of \( z_{T+1}^{T_{+}} \) from eq. 36, and approximate

\[
p(y_{T+1}|y_{1:T}) \approx \sum_{i} p(y_{T+1}|y_{1:T}, z_{T+1}^{i}). \tag{38}
\]

It is straightforward to extend this to multiple steps ahead prediction.

\(^1\)The formula for ML estimate of the noise covariance is given in (Bishop, 2006)

\(^2\)This corresponds to assuming the latent noise covariance to be diagonal.

\(^3\)See Ch.5 in (Beal, 2003) for derivation in detail
5. Experiments

We apply our method to forecasting time series using data generated from a toy two-level hierarchical network given in Fig. 2. This network represents the hierarchy present in Australian domestic tourism data. This data is an indicator of tourism activity: the number of visitor nights per quarter consists of time series. There are two levels of hierarchy in the data as shown in Fig. 2. The aggregated domestic tourism demand for the entire Australia consists of the top level time series. At (sub) level 1, the top level time series is disaggregated by four different purpose of travel: Holiday, Visiting friends and relatives, Business, and Others. At (sub) level 2, the level 1 time series is disaggregated by seven different states and territories they visited: New South Wales, Queensland, Victoria, Western Australia, Tasmania, and the Northern Territory. Therefore, there are 4 time series at level 1, 24 time series at level 2.

We generated each time series from the AR-1 processes with random parameters. To make sure the sum of sub-level time series matches the upper-level time series, we divide the sub-level time series by the sum of sub-level time series. This gives us a proportion at the sub-level, which we multiply by the upper-level time series to obtain the revised sub-level time series.

For forecasting individual time series independently, we used an ARMA model using the automatic algorithm developed in (Hyndman & Khandakar, 2008). We used the first 120 observations (1980:Q1-2010:Q4) as a training set and predicted tourism demands up to 4-steps ahead (i.e., Q1, Q2, Q3, and Q4 in 2011). We computed the MAPE (mean absolute percentage error) values from these results and computed the average in Table 1. For the top-down method (TD), we used the forecasted proportion (FP) method. For the bottom-up method (BU), we used the simple summation of the individual forecasts at the bottom level. As shown in Table 1, our method outperforms other methods.

6. Discussion

In this paper, we modeled the proportion changes of hierarchically structured time series using linear dynamical systems with multinomial observations. We developed the variational Bayesian expectation maximization algorithm for posterior inference and parameter estimation. The sequential forward/backward type algorithm allows us to parallelize the posterior inference, which would be beneficial when dealing with large datasets. Simulation results on a toy dataset show the effectiveness of our approach.

A potential criticism of our approach would be that our method highly relies on top level prediction. However, top level time series are often periodic and less abruptly changing over time (since they are sum of many sub-level time series) compared to individual time series at the bottom level. The prediction performance at the top level typically outperforms sub-level prediction (Table 1 shows the same trend). It would be interesting to test our method to large hierarchical time series datasets in future work.

Table 1. Forecasting performance (MAPE) Our method (DPM) outperforms other methods. The DPM achieved the lowest average MAPE across the three levels, 22.80. The second best method (TD) achieved average MAPE 28.78.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOP LEVEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEP</td>
<td>9.32</td>
<td>34.01</td>
<td>26.58</td>
<td>20.17</td>
<td>22.52</td>
</tr>
<tr>
<td>TD</td>
<td>13.36</td>
<td>38.65</td>
<td>31.46</td>
<td>28.05</td>
<td>27.88</td>
</tr>
<tr>
<td>BU</td>
<td>10.86</td>
<td>49.19</td>
<td>37.51</td>
<td>32.27</td>
<td>32.46</td>
</tr>
<tr>
<td>OPTIMAL</td>
<td>13.55</td>
<td>41.56</td>
<td>33.56</td>
<td>29.72</td>
<td>29.60</td>
</tr>
<tr>
<td>DPM</td>
<td>9.36</td>
<td>30.43</td>
<td>23.78</td>
<td>20.86</td>
<td>21.11</td>
</tr>
<tr>
<td>LEVEL-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEP</td>
<td>14.07</td>
<td>50.76</td>
<td>38.99</td>
<td>33.50</td>
<td>33.58</td>
</tr>
<tr>
<td>TD</td>
<td>13.36</td>
<td>38.65</td>
<td>31.46</td>
<td>28.05</td>
<td>27.88</td>
</tr>
<tr>
<td>BU</td>
<td>10.86</td>
<td>49.19</td>
<td>37.51</td>
<td>32.27</td>
<td>32.46</td>
</tr>
<tr>
<td>OPTIMAL</td>
<td>13.55</td>
<td>41.56</td>
<td>33.56</td>
<td>29.72</td>
<td>29.60</td>
</tr>
<tr>
<td>DPM</td>
<td>9.36</td>
<td>30.43</td>
<td>23.78</td>
<td>20.86</td>
<td>21.11</td>
</tr>
<tr>
<td>LEVEL-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEP</td>
<td>26.11</td>
<td>58.91</td>
<td>46.52</td>
<td>40.51</td>
<td>43.01</td>
</tr>
<tr>
<td>TD</td>
<td>24.62</td>
<td>47.16</td>
<td>38.15</td>
<td>33.87</td>
<td>35.95</td>
</tr>
<tr>
<td>BU</td>
<td>26.11</td>
<td>58.91</td>
<td>46.52</td>
<td>40.51</td>
<td>43.01</td>
</tr>
<tr>
<td>OPTIMAL</td>
<td>27.21</td>
<td>52.00</td>
<td>41.90</td>
<td>37.00</td>
<td>39.53</td>
</tr>
<tr>
<td>DPM</td>
<td>10.23</td>
<td>29.96</td>
<td>27.16</td>
<td>31.74</td>
<td>24.77</td>
</tr>
</tbody>
</table>
References


