We use `coxed()` to provide an answer to the key question, "how much longer will negotiations take for an ideologically polarized coalition as compared to an ideologically homogeneous one?" Specifically, we call `coxed()` and specify two new datasets, one in which `rgovm = 0` indicating that all political parties in the governing coalition have the same ideological position (i.e., a coalition of one party), and one in which `rgovm = 1.24`, indicating that the parties have very different ideological positions.[9] We use `mutate()` from the **dplyr** package (Wickham et al., 2018) to quickly create new data frames in which `rgovm` equals 0 or 1.24 for all cases, and set these two data frames as `newdata` and `newdata2` inside `coxed()`.

```
me <- coxed(mv.cox, method = "gam", bootstrap = TRUE, B = 30,
            newdata = mutate(martinvanberg, rgovm = 0),
            newdata2 = mutate(martinvanberg, rgovm = 1.24))
```

`coxed()` calculates expected durations for all cases under each new data frame and subtracts the durations for each case. To obtain point estimates we can request the mean or median difference.

```
> summary(me, stat = "mean")
             mean bootstrap.se     lb     ub
newdata2   28.927        3.285 22.489 35.365
newdata    25.321        2.632 20.163 30.480
difference  3.605        2.417 -1.133  8.343
> summary(me, stat = "median")
           median bootstrap.se     lb     ub
newdata2   22.392        3.234 16.053 28.730
newdata    19.692        3.449 12.932 26.451
difference  2.928        1.931 -0.857  6.714
```

These results demonstrate that a coalition in which the parties have average ideological differences will take 3.6 more days on average (with a median of 2.9 days) to conclude negotiations than a coalition in which all parties have the same position (i.e., a single-party government).

The NPSF method can be used to compute estimates of these same quantities simply by specifying `method = "npsf"` in the `coxed()` function. Additionally, the package includes a function called `sim.survdata()` designed for simple simulations of duration data that do not assume a distributional form for the baseline hazard. This method, which is fully described in Harden and Kropko (2019), can be useful in several applied and computational settings that involve the Cox model.

Conclusions

The Cox model is popular among applied researchers in a wide range of disciplines due to its inherent flexibility. However, this flexibility makes conveying the substantive meaning of results challenging. By using only the rank ordering of the observed duration times, the Cox model limits researchers to interpreting results in the language of hazard and changes in risk. This yields two key problems. First, it is substantively vague because hazard does not have a meaningful scale. This hinders researchers' capacity to determine whether an estimated effect is substantively "large" or "small." Furthermore, hazard-based interpretations require specialized knowledge to understand. This makes the research less accessible to general audiences, who may be able to learn from the work but cannot due to the means by which results are communicated.

The COX ED methods provide a solution to these problems by allowing researchers to compute duration-based quantities from the Cox model. Communicating results in the language of time allows for more substantive precision and is intuitive to a broad audience of readers. We demonstrate above that COX ED is straightforward to implement in R. The **coxed** package contains functions that allow researchers to use the methods even with minimal knowledge of R. Additionally, the functions are flexible; users can make several changes to many of their features to suit the problem at hand. Finally, the output from the functions provide point estimates, standard errors, and confidence intervals, so researchers can report their results with appropriate measures of uncertainty.

In sum, the **coxed** package provides a useful alternative for researchers to communicate results from the Cox model. It gives them the benefits of the intuitive quantities available in parametric models while retaining the desirable estimation properties of the Cox model. Thus, the analysis can be guided by appropriate modeling choices, but reported in an intuitive, accessible manner.

[9] Martin and Vanberg (2003) select these values in making hazard rate comparisons. The value `rgovm = 1.24` reflects the average ideological range of coalition governments in the sample.

Bibliography

A. C. Cameron, J. B. Gelbach, and D. L. Miller. Bootstrap based improvements for inference with clustered errors. *Review of Economics and Statistics*, 90(3):414–427, 2008. URL https://doi.org/10.1162/rest.90.3.414. [p]

D. R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34(2):187–220, 1972. URL https://doi.org/10.1111/j.2517-6161.1972.tb00899.x. [p]

D. R. Cox and D. Oakes. *Analysis of Survival Data*. Monographs on Statistics & Applied Probability. Chapman & Hall/CRC, Boca Raton, FL, 1984. [p]

T. J. DiCiccio and B. Efron. Bootstrap confidence intervals. *Statistical Science*, 11(3):189–228, 1996. URL https://doi.org/10.1214/ss/1032280214. [p]

J. J. Harden and J. Kropko. Simulating duration data for the Cox model. *Political Science Research and Methods*, 7(4):921–928, 2019. URL https://doi.org/10.1017/psrm.2018.19. [p]

F. E. Harrell. *rms: Harrell Miscellaneous*, 2018. R package version 5.1–2. http://biostat.mc.vanderbilt.edu/wiki/Main/Rrms. [p]

T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman & Hall/CRC, Boca Raton, FL, 1990. [p]

J. Kropko and J. J. Harden. Beyond the hazard ratio: Generating expected durations from the Cox proportional hazards model. *British Journal of Political Science*, 50(1):303–320, 2020. URL https://doi.org/10.1017/S000712341700045X. [p]

L. W. Martin and G. Vanberg. Wasting time? The impact of ideology and size on delay in coalition formation. *British Journal of Political Science*, 33(2):323–344, 2003. URL https://doi.org/10.1017/S0007123403000140. [p]

T. H. Scheike and M.-J. Zhang. Analyzing competing risk data using the R timereg package. *Journal of Statistical Software*, 38(2):1–15, 2011. URL http://dx.doi.org/10.18637/jss.v038.i02. [p]

T. Therneau. *survival: A Package for Survival Analysis in S*, 2015. R package version 2.38. [p]

D. Tingley, T. Yamamoto, K. Hirose, L. Keele, and K. Imai. mediation: R package for causal mediation analysis. *Journal of Statistical Software*, 59(5):1–38, 2014. URL http://dx.doi.org/10.18637/jss.v059.i05. [p]

H. Wickham, R. François, L. Henry, and K. Müller. *dplyr: A Grammar of Data Manipulation*, 2018. URL https://CRAN.R-project.org/package=dplyr. R package version 0.7.6. [p]

S. N. Wood. *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC, Boca Raton, FL, 2006. [p]

S. N. Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semi-parametric generalized linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 73(1):3–36, 2011. URL https://doi.org/10.1111/j.1467-9868.2010.00749.x. [p]

Jonathan Kropko
University of Virginia
School of Data Science
Dell 1 Building
Charlottesville, VA 22904
jkropko@virginia.edu

Jeffrey J. Harden
University of Notre Dame
Department of Political Science
2055 Jenkins Nanovic Halls
Notre Dame, IN 46556
jeff.harden@nd.edu

Modeling regimes with extremes: the bayesdfa package for identifying and forecasting common trends and anomalies in multivariate time-series data

by Eric J. Ward, Sean C. Anderson, Luis A. Damiano, Mary E. Hunsicker, Michael A. Litzow

Abstract The **bayesdfa** package provides a flexible Bayesian modeling framework for applying dynamic factor analysis (DFA) to multivariate time-series data as a dimension reduction tool. The core estimation is done with the Stan probabilistic programming language. In addition to being one of the few Bayesian implementations of DFA, novel features of this model include (1) optionally modeling latent process deviations as drawn from a Student-t distribution to better model extremes, and (2) optionally including autoregressive and moving-average components in the latent trends. Besides estimation, we provide a series of plotting functions to visualize trends, loadings, and model predicted values. A secondary analysis for some applications is to identify regimes in latent trends. We provide a flexible Bayesian implementation of a Hidden Markov Model — also written with Stan — to characterize regime shifts in latent processes. We provide simulation testing and details on parameter sensitivities in supplementary information.

Overview

A goal of many multivariate statistical techniques is to reduce dimensionality in observed data to identify shared or latent processes. Factor analysis models represent a general class of models used to relate multiple observations to a lower dimension (factors), while also considering different covariance structures of the observed data. Factors are not directly observed, but represent a hidden, shared process among variables. Though goals of factor analysis are sometimes similar to techniques such as principal component analysis (PCA), factor analysis models explicitly estimate residual error terms, whereas PCA does not (Anderson and Rubin, 1956; Jolliffe, 1986). These factor models are written as $y_i = u_i + \mathbf{Z}\, f_i + \varepsilon_i$, where observed data y_i is a linear combination of an intercept u_i and the product of latent factors f_i and loadings $\mathbf{Z}$ (loadings are sometimes referred to in the literature as $\mathbf{L}$).

In a time-series setting, factor models may be extended to dynamic factor analysis (DFA) models. DFA models aim to reduce the dimensionality of a collection of time series by estimating a set of shared trends and factors, representing the linear effects of each trend on the observed data (Molenaar, 1985; Zuur et al., 2003; Stock and Watson, 2005). The number of trends m is chosen to be less or equal than the number of time series n. The general form of the DFA model can be formulated as a state-space model (Petris, 2010). The latent processes (also referred to as 'trends') are generally modeled as random walks, so that trend i is modeled as $x_{i,t+1} = x_{i,t} + w_{i,t}$ where $x_{i,t}$ is the value of the i-th latent trend at time t, and the deviations $w_{i,t}$ are modeled as white noise. Across trends, these deviations are modeled as $\mathbf{w}_t \sim \text{MVN}(0, \mathbf{Q})$. The latent trends $x_{i,t}$ are linked to data via a loadings matrix $\mathbf{Z}$ whose values do not evolve through time, $y_t = \mathbf{Z}x_t + \mathbf{a} + \mathbf{B}d_t + \mathbf{e}_t$. The loadings matrix $\mathbf{Z}$ is dimensioned $n \times m$ so that $Z_{j,i}$ represents the effect of trend i on time series j. The parameters $\mathbf{a}$ and $\mathbf{B}$ are optional parameters, representing time-series-specific intercepts and effects of covariates, d_t. Finally, the residual errors are assumed to be $\mathbf{e}_t \sim \text{MVN}(0, \mathbf{R})$, where $\mathbf{R}$ is an estimated covariance matrix.

Estimation of DFA models is typically done in a maximum likelihood framework, using the expectation-maximization (EM) algorithm or other optimization tools. Implementation of these methods is available in multiple R packages including **dlm** (Petris, 2010), **KFAS** (Helske, 2017), **MARSS** (Holmes et al., 2012b), and **tsfa** (Gilbert and Meijer, 2005). Challenges in parameter estimation and interpretation for DFA models have been well studied. Without constraints, parameters in the DFA model are not identifiable (Harvey, 1990; Zuur et al., 2003). To ensure identifiability of variance parameters, for example, the covariance matrix $\mathbf{Q}$ is generally fixed as an identity matrix (Harvey, 1990). To avoid confounding the latent trends and loadings matrix $\mathbf{Z}$, elements of $\mathbf{Z}$ must also be constrained. A common choice of constraints is for the elements in the first $m - 1$ rows of $\mathbf{Z}$ to be set to zero if the column index is greater than the row index, $j > i$ (Harvey, 1990), though other constraints have been proposed (Bai and Wang, 2015). For a 3-trend DFA model for instance, these constraints

would mean that the **Z** matrix parameters would be configured as

$$\begin{bmatrix} Z_{1,1} & 0 & 0 \\ Z_{2,1} & Z_{2,2} & 0 \\ Z_{3,1} & Z_{3,2} & Z_{3,3} \\ \dots & \dots & \dots \end{bmatrix}.$$

Several previous approaches to DFA estimation in a maximum likelihood framework also center (subtract the sample means) or standardize (subtract the sample means and divide by the sample standard deviations) data prior to fitting DFA models and set the intercepts a equal to zero to avoid potential confounding of level parameters (Holmes et al., 2012a). We adopt a similar approach, allowing users to either center or standardize data before estimation, and not including the intercepts as estimated parameters.

Label switching

We developed our DFA model in a Bayesian framework, using Stan and the package **rstan** (Stan Development Team, 2016), which implements Markov chain Monte Carlo (MCMC) using the No-U Turn Sampling (NUTS) algorithm (Hoffman and Gelman, 2014; Carpenter et al., 2017). Although estimation of the DFA model in a Bayesian setting is not new (Aguilar and West, 2000; Koop and Korobilis, 2010; Stock and Watson, 2011), it presents several interesting challenges over the EM algorithm. In addition to the constraints on **Q** and **Z**, Bayesian estimation suffers from a problem of label switching. In particular, elements of **F** or **Z** may flip sign within an MCMC chain, or multiple chains may converge on parameters that are identical in magnitude but with different signs.

To minimize issues with label switching, previous work on Bayesian factor analysis has proposed additional constraints on the loadings matrix, including setting the elements of **Z** to be constrained (-1, 1), or adding a positive constraint to the diagonal, $Z_{ii} > 0$ (Aguilar and West, 2000; Geweke and Zhou, 1996). Though these constraints generally help, there may be situations where MCMC chains still do not converge. To address this issue, we adopt the parameter-expanded priors for the loadings and trends proposed by Ghosh and Dunson (2009). To ensure that the sign of the estimated quantities is the same across MCMC chains, we created the function `flip_trends()` to flip the posterior samples of MCMC chains relative to the first chain as needed.

The Bayesian dynamic factor model with extremes

There are several approaches for modeling extreme deviations in time series models. Techniques include modeling deviations as a two-component mixture (Ward et al., 2007; Evin et al., 2011), or modeling deviations with non-Gaussian distributions including the Student-t distribution (Praetz, 1972; Anderson et al., 2017; Anderson and Ward, 2018). There are several existing packages to include Student-t distributions; these include **heavy** for applications to regression and mixed effects models (Osorio and F., 2018), **bsts** for univariate time series models (Scott, 2018), and **stochvol** for stochastic volatility models (Kastner, 2016). Because switching from a Gaussian to Student-t distribution only introduces a single parameter, ν, the degrees of freedom, we extend the latter approach to a multivariate setting to model extreme events in the latent trends, so that deviations in the trends are modeled as $w_t \sim \text{MVT}(\nu, 0, \mathbf{Q})$. As before, **Q** is fixed as an identity matrix **I**. Our parameterization constrains DFA models to have the same degrees of freedom ν in the residuals of the multiple trends, which may be fixed *a priori* or treated as a free parameter with a gamma(shape = 2, rate = 0.1)[2,∞] prior (Juárez and Steel, 2010).

Including autoregressive and moving average components

The trends of the dynamic factor model are most commonly modeled as non-stationary random walks, $x_{i,t+1} = x_{i,t} + w_{i,t}$, where the $w_{i,t} \sim N(0,1)$ are Gaussian white noise. Like with other vector autoregressive time series models, this framework can be easily extended to include optional autoregressive (AR) or moving average (MA) components (Chow et al., 2011). We allow for AR(1) and MA(1) processes to be specified with boolean arguments to the `fit_dfa()` function. For both the AR(1) and MA(1) components, we assume separate parameters for each trend. Including the AR(1) component ϕ_i makes the trend process become $x_{i,t+1} = \phi_i x_{i,t} + w_{i,t}$, where values of ϕ_i close to 1 make the trend behave as a random walk, and small values of ϕ_i close to 0 make the trend behave as white noise. Similarly, we model the MA(1) component as an AR(1) process on the error terms $w_{i,t}$. Instead of being independent at each time step, θ_i controls the degree of autocorrelation among deviations, $w_{i,t} \sim \text{N}\left(\theta_i w_{i,t-1}, 1\right)$. For stationarity and invertability, we constrain $|\phi_i| < 1$ and $|\theta_i| < 1$.

Rotation of trends and loadings

Like factor analysis models, there are many solutions from a DFA model capable of producing the same fit to the data. Following previous authors, we use a varimax rotation of the loadings matrix $\mathbf{Z}$ to transform the posterior loadings and trends (Kaiser, 1958; Harvey, 1990; Holmes et al., 2012a). If $\widehat{\mathbf{Z}}$ is the posterior mean of the loadings matrix from a DFA model of 4 time series and 2 trends for example, the rotation matrix $\mathbf{W}^* = \text{varimax}(\widehat{\mathbf{Z}})$ is dimensioned 2×2. The rotated loadings matrix can then be calculated as $\widehat{\mathbf{Z}}^* = \widehat{\mathbf{Z}}\,\mathbf{W}^*$ and rotated trends calculated as $\widehat{\mathbf{x}}^* = \mathbf{W}^{*-1}\widehat{\mathbf{x}}$, where $\widehat{\mathbf{x}}$ is the posterior mean of the trends.

Identifying data support for the number of trends

Since the number of trends in a DFA model is not a parameter, comparing data support across models is often necessary. Using model selection tools to identify data support is available via Akaike's Information Criterion (AIC) in packages implementing maximum likelihood for estimation of state-space models (Petris, 2010; Holmes et al., 2012b). In addition to comparing the relative support of different number of trends, model selection for Bayesian dynamic factor models may be useful for evaluating the error structure for the residual error covariance matrix $\mathbf{R}$, whether covariates should be included, whether latent trends are better modeled with a distribution allowing for extremes (MVT versus MVN), and whether the latent trends support estimation of AR or MA components. For our Bayesian DFA models, we extend the **loo** package (Vehtari et al., 2016a,b) to generate estimates of LOOIC (Leave-One-Out Information Criterion) for fitted models. To ease the selection process, **bayesdfa** includes the function `find_dfa_trends()` to run multiple models specified by the user. It returns a table of LOOIC values (denoting which of those failed convergence criteria) and the model with the lowest LOOIC value.

Anomalies or black-swan events

As a diagnostic tool, we include the function `find_swans()` to fitted DFA models. We adopt the same approach and terminology for 'black-swan events' as in Anderson et al. (2017), where black-swan events are rare and unexpected extremes. Our `find_swans()` function first-differences the posterior mean estimates of each DFA trend and evaluates the probability of observing a difference that is more extreme than expected under a normal distribution with the same scale parameter. Events beyond a user-defined threshold (e.g. 1 in 100, or 1 in 10,000) are then classified as outliers and plotted.

Simulation tests

To evaluate the ability of the Bayesian DFA model to identify anomalies in latent processes, we created simulated data using our `sim_dfa()` function. We generated simulated multivariate time series ($n = 4$ time series with $T = 20$ time steps each) with $m = 2$ underlying latent trends. Extremes were included as a step-change in the midpoint of the first trend in each simulated dataset. We varied the value of the step from -4 to -8, which represent unlikely events under the assumption that temporal deviations in the latent trends are distributed according to $N(0, 1)$. Because increased observation error may corrupt inference about anomalies in the trends, we considered three levels of observation error ($\sigma = 0.25, 0.75, 1.25$). We generated 200 simulated samples for each permutation of parameters, resulting in a total of 3000 datasets.

We fit the Bayesian DFA model with Student-t errors to each simulated dataset. As expected, the posterior estimates from these simulations illustrate that the ability to estimate low degrees of freedom is related to the magnitude of extremes (Figure 1). Similarly, higher observation error corrupts the ability to estimate extreme events, even when they are large in magnitude (Figure 1).

Using HMMs to classify regimes in latent DFA trends

An alternative approach to DFA for dimension reduction of multivariate time series data are Hidden Markov Models (HMMs). Like DFA models, they model a latent process for a time series (or collection of multivariate time series). Instead of the latent process being modeled continuously (e.g. as a random walk in DFA), HMMs conceive the latent process as a series of discrete-time, discrete-state first-order Markov chains $s_t \in \{1, \ldots, G\}$ with the number of possible states G specified *a priori*. State transition is characterized by the $G \times G$ transition matrix with simplex rows $\mathbf{A} = \left\{a_{ig}\right\}$ where $a_{ig} = p(s_t = g | s_{t-1} = i)$ represents the probability of transitioning from state i to g. Useful quantities

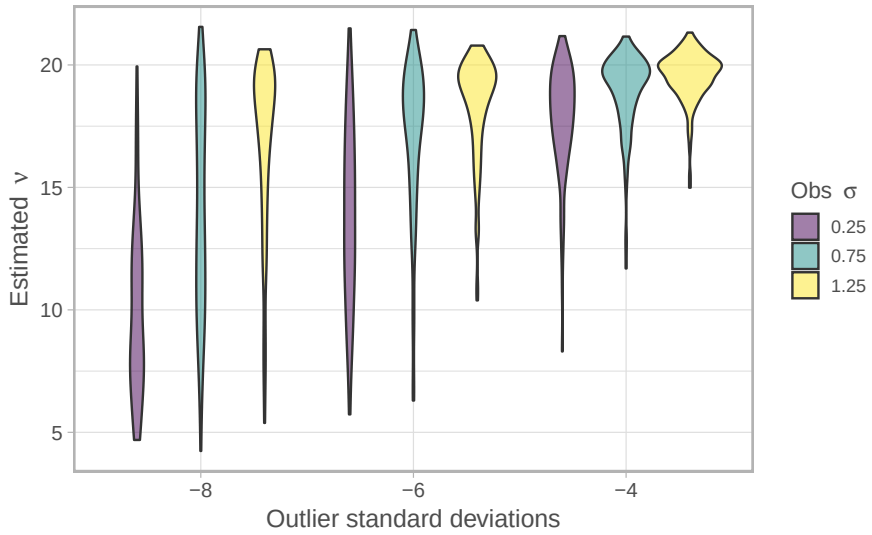

Figure 1: Results for simulated data illustrating support for the Student-t distribution (low values of nu), varying the magnitude of extremes (standard deviations from the mean) and magnitude of observation error.

from HMMs include the transition probabilities between latent states, and the probability of being in a given lantent state at each point in time (Zucchini et al., 2017).

HMMs can be applied to raw multivariate data to identify latent states; however, they may also be linked with DFA to identify regimes and transitions in the latent DFA trends. Similar to DFA, applications of HMMs are widely available in R, including via the packages **depmixS4** (Visser and Speekenbrink, 2010), **HMM** (Himmelmann, 2010), and **msm** (Jackson, 2011). Consistent with our implementation of the Bayesian DFA model, we include fully Bayesian inference in Stan based on Damiano et al. (2018). We apply independent HMM models to each DFA trend to identify alternate states or regimes. Like with the estimation of DFA models, we use the LOOIC metric to evaluate the relative support for HMMs with different numbers of underlying states, selecting the converged model with the lowest LOOIC. By default, we assume the observation model of the input time series to be normally distributed with the scale parameter equal to the estimated residual variance. However, for some applications, such as datasets with changing sampling frequencies over time, uncertainty in DFA trends may also vary through time. To propogate this uncertainty forward, we also allow the residual variance to be entered as a known quantity for every data point in our `find_regimes()` function.

Example application: identifying common patterns in sea surface temperatures in the Northeast Pacific Ocean

To illustrate an example application of the **bayesdfa** package to real data, we use monthly anomalies of sea surface temperature (SST, measured in $C°$). SST is observed from satellite and buoy data at fixed locations, and model-based interpolations are used to generate estimates at additional gridded locations[1]. We used estimates generated at the locations of 4 observing stations used by the Pacific Fisheries Environmental Laboratory[2] from the west coast of North America (USA). The four stations have some degree of correlation with one another, and are separated by approximately 6 degrees of latitude from one another. In summary, we work with $n = 4$ monthly time series with $T = 167$ observations each (from 2003–01 to 2016–05) and no missing values.

Initially, we fit a DFA model with 2 hidden trends, and will assume the 4 time series to have the same error variances **R**. We will fit the DFA model with possible extremes, modeling process error with a Student-t distribution by using the argument `estimate_nu()`. To evaluate whether these data support an extreme DFA with trends modeled as a t-distribution, we will fit two competing forms: one modeling the random walks with a Gaussian distribution, and the other using a Student-t distribution. Generating posterior samples for each model takes approximately 7 minutes per chain, when MCMC chains aren't run in parallel.

[1] https://coastwatch.pfeg.noaa.gov/erddap/info/osuSstAnom/index.html
[2] https://www.pfeg.noaa.gov/

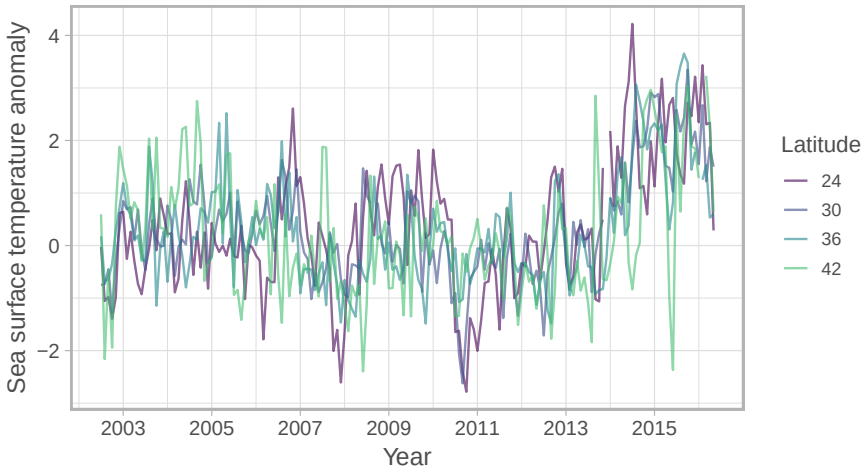

Figure 2: Sea surface temperature anomalies, at four stations on the west coast of the USA ordered by increasing latitude. The station coordinates are (113W, 24N), (119W, 30N), (122W, 36N), (125W, 42N).

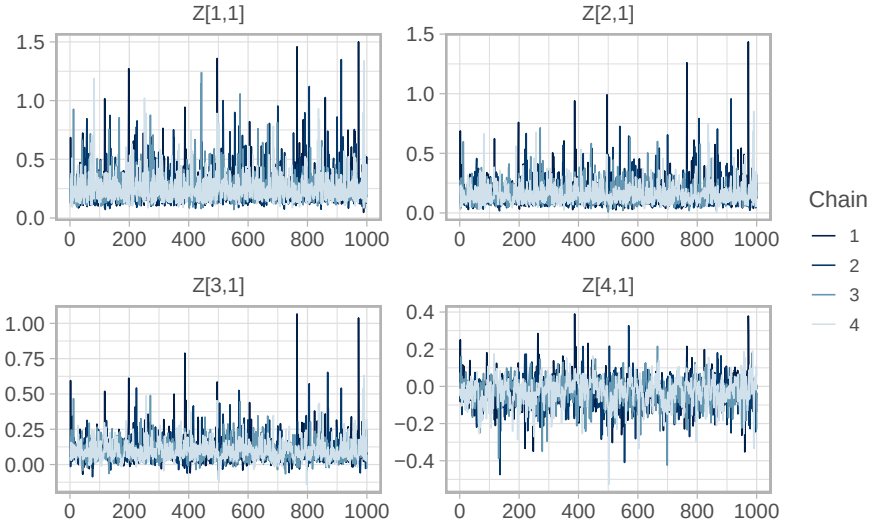

Figure 3: MCMC trace plots of loading parameters (Z) in the DFA model with Student-t errors.

After fitting the models, we confirm whether the MCMC chains are consistent with convergence using a threshold value of $\hat{R} = 1.05$ (Gelman et al., 2014) using our `is_converged()` function. We also visually inspect chain traceplots (e.g. Figure 3) and check the minimum effective sample size across parameters: NaN.

As a consistency diagnostic, we also retrieve the estimated degrees of freedom from the Student-t model ν. By visual inspection, Figure 4 shows that the posterior distribution on ν is lower than the prior distribution.

Visualizing the trends and loadings

We will focus the remaining portion of our analysis on the results from the DFA model with Student-t deviations. In Figure 5, we observe that Trend 1 and Trend 2 both support SST anomalies increasing over the latter half of the time series. Both trends appear to have reversed direction (reverting to the mean in the last 2–3 years) and this pattern is more evident in Trend 1. Because we do not model seasonality explicitly, for example by including a covariate effect for the month, each of the estimated trends also includes the within-year variability that describes seasonal patterns in observed sea surface temperature.

In the violin plot of Figure 6, we note that more southern stations (24 and 30N) contribute largely to Trend 1, while the more northern stations appear to load more heavily on Trend 2.

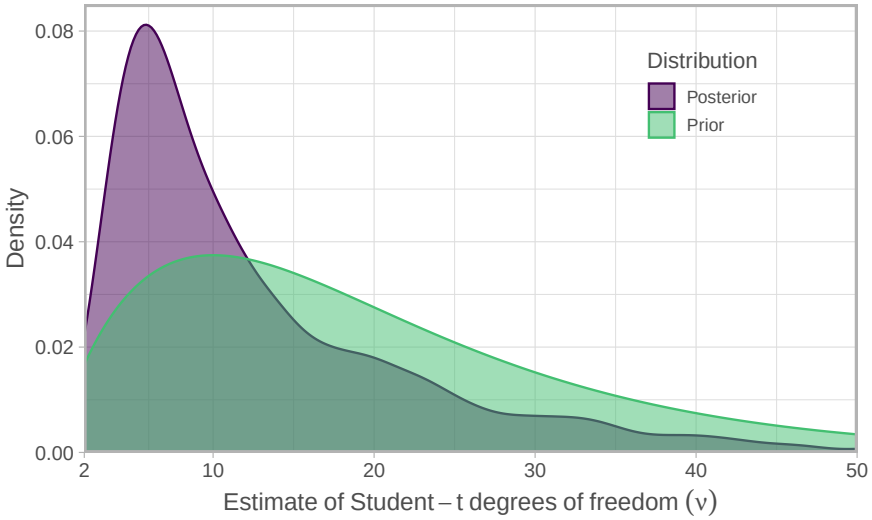

Figure 4: Posterior and prior degrees of freedom in the DFA model with Student-t errors.

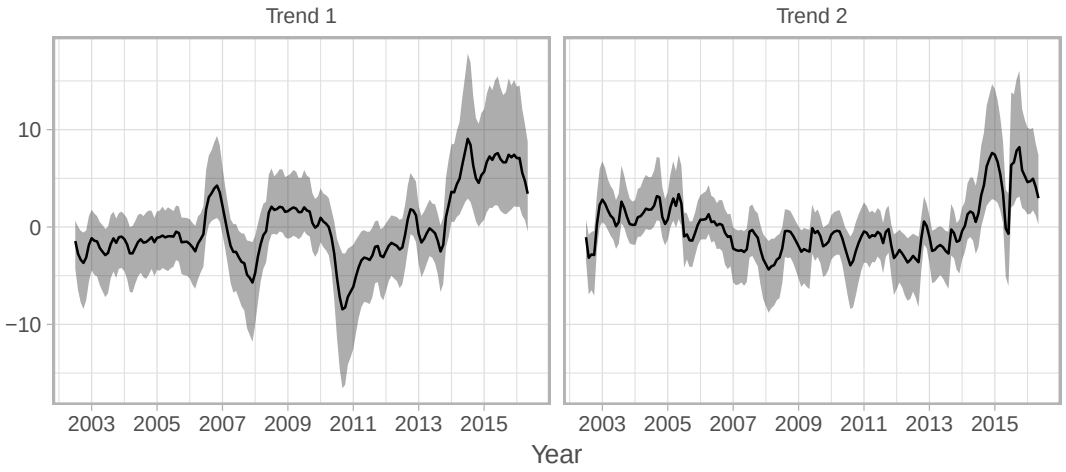

Figure 5: Latent trends from the DFA model with Student-t process deviations. Trends are rotated using the `stats::varimax()` rotation.

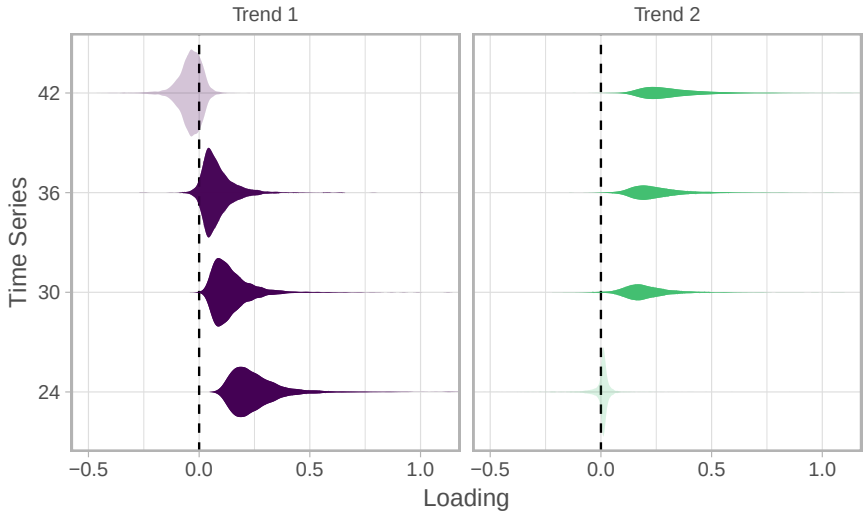

Figure 6: Loadings from the DFA model with Student-t process deviations. Loadings are rotated using the `stats::varimax()` rotation.

Regimes	LOOIC Trend 1	LOOIC Trend 2
1	855.5	756.7
2	31.0	30.3
3	69.1	99.9
4	139.7	164.6

Table 1: LOOIC estimates across different numbers of regimes for each latent DFA trend. LOOIC is calculated using the `loo::loo()` function.

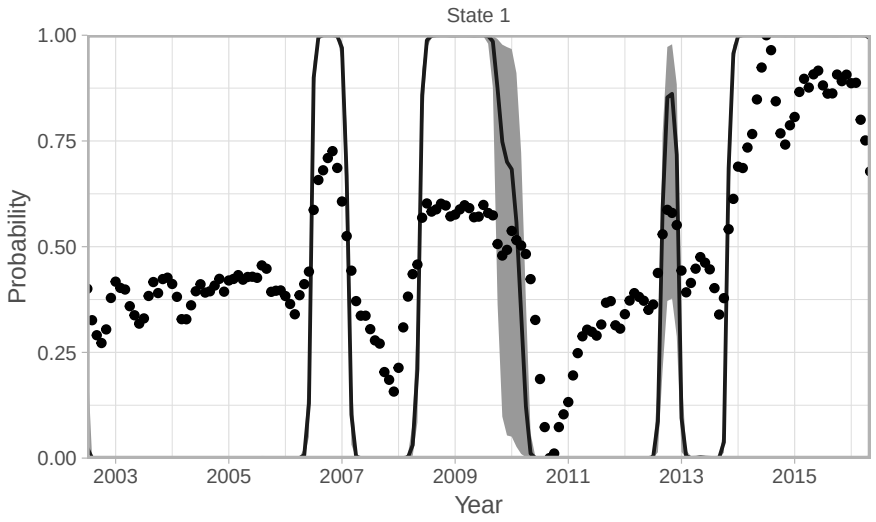

Figure 7: Estimated regimes from the 2-regime HMM in Trend 1 of the DFA model fit to the sea surface temperature anomaly data. The visualization summarizes the assignment probabilities $p(s_t = 1|x_T)$ of Trend 1 being in State 1 (for the sea surface temperature case study, State 1 is associated with warm periods). Dots represent the latent DFA trend scaled to an interval [0, 1]. The black line represents the median and the shaded area uncertainty (90% posterior interval).

Identifying regimes in the latent DFA trends with Hidden Markov Models

For each trend, we apply independent HMMs to examine the support for differing numbers of underlying regimes. Both the posterior mean and standard deviation (optional argument) will be the inputs to the HMM.

Using LOOIC as a metric of support for the number of regimes, the estimates reported in Table 1 support the inclusion of 2 regimes for both Trends 1 and 2.

Our `fit_regimes()` function computes the probability of each time point being in one of the regime states, which may also be visualized using `plot_regime_model()`. For example, the output of the 2-regime model for Trend 1 in Figure 7 suggests a change in the middle of the time series, then changing back again to State 1. Similarly, by the end of the series, the HMM assigns Trend 1 to being in State 1.

Extensions

There are a number of extensions to our implementation of the Bayesian DFA model with extremes that could make the model more applicable to a wider range of problems. Examples for the process model include adopting a skew-t distribution for asymmetric extremes. For models estimating multiple trends, multiple parameters may be treated hierarchically (e.g. covariate effects, variance parameters). For the observation or data model, our implementation of the Bayesian DFA model only includes data arising from a Gaussian or Student-t distribution, though this could be extended to include discrete or other continuous densities. Finally, spatial dynamic factor models (sDFA) have emerged as a useful tool for complicated multivariate spatial datasets (Lopes et al., 2011; Thorson et al., 2015), and could be similarly implemented in Stan.

Conclusion

This paper presents the **bayesdfa** package for applying Bayesian DFA to multivariate time series as a dimension reduction tool, particularly if extreme events may be present in observed data. In addition to allowing for the inclusion of covariates, we also extend the conventional dynamic factor model to include optionial moving average and autoregressive components in the latent trends. Applying this package to a dataset of sea surface temperature from the Northeast Pacific Ocean, we fit DFA models with Gaussian and Student-t errors. Though the model with Student-t errors has slightly lower LOOIC, the results from the two models are similar. Output from these 2-trend DFA models of sea surface temperature are useful in demonstrating a north-to-south gradient in temperature anomalies (Figure 6). Standardized temperature data from southern stations experience more interannual variability and temperatures that are greater in magnitude compared to northern stations (Figure 5). We also illustrate how latent trends from DFA models can be analyzed in a HMM framework to identify regimes and transitions; applied to the sea surface temperature data, both Trend 1 and Trend 2 support 2-regime models (roughly interpreted as 'warm' and 'cool' regimes; Figure 7).

Acknowledgements

This work was funded by NOAA's Fisheries and the Environment (FATE) Program. Development of this package benefitted from discussions with other members of our working group (Jin Gao, Chris Harvey, Sam McClatchie, Stepahni Zador) and scientists at the Northwest Fisheries Science Center (including Mark Scheuerell, James Thorson, Eli Holmes, and Kelly Andrews). 2 anonymous reviewers helped improve the clarity and plots of this paper.

Bibliography

O. Aguilar and M. West. Bayesian Dynamic Factor Models and Portfolio Allocation. *Journal of Business & Economic Statistics*, 18(3):338–357, July 2000. doi: 10.1080/07350015.2000.10524875. [p]

S. C. Anderson and E. J. Ward. Black swans in space: Modelling spatiotemporal processes with extremes. *Ecology*, In press, 2018. doi: 10.1002/ecy.2403. [p]

S. C. Anderson, T. A. Branch, A. B. Cooper, and N. K. Dulvy. Black-swan events in animal populations. *Proceedings of the National Academy of Sciences*, 114(12):3252–3257, 2017. doi: 10.1073/pnas.1611525114. [p]

T. W. Anderson and H. Rubin. Statistical inference in factor analysis. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 5: Contributions to Econometrics, Industrial Research, and Psychometry*, pages 111–150, Berkeley, California, 1956. University of California Press. [p]

J. Bai and P. Wang. Identification and Bayesian Estimation of Dynamic Factor Models. *Journal of Business & Economic Statistics*, 33(2):221–240, Apr. 2015. doi: 10.1080/07350015.2014.941467. [p]

B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software, Articles*, 76(1):1–32, 2017. ISSN 1548-7660. doi: 10.18637/jss.v076.i01. URL https://www.jstatsoft.org/v076/i01. [p]

S.-M. Chow, N. Tang, Y. Yuan, X. Song, and H. Zhu. Bayesian estimation of semiparametric nonlinear dynamic factor analysis models using the Dirichlet process prior. *The British journal of mathematical and statistical psychology*, 64(Pt 1):69–106, Feb. 2011. doi: 10.1348/000711010X497262. [p]

L. Damiano, B. Peterson, and M. Weylandt. A tutorial on hidden Markov models using Stan. 2018. doi: 10.5281/zenodo.1284341. URL https://doi.org/10.5281/zenodo.1284341. [p]

G. Evin, J. Merleau, and L. Perreault. Two-component mixtures of normal, gamma, and Gumbel distributions for hydrological applications. *Water Resources Research*, 47(8), Aug. 2011. doi: 10.1029/2010WR010266. [p]

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, Boca Raton, FL, third edition, 2014. [p]

J. Geweke and G. Zhou. Measuring the pricing error of the arbitrage pricing theory. *The Review of Financial Studies*, 9(2):557–587, 1996. doi: 10.1093/rfs/9.2.557. URL http://dx.doi.org/10.1093/rfs/9.2.557. [p]