User's Manual of `linkalman`

Su, Danyang

September 30, 2019

To my family

Contents

1 Introduction

`linkalman` is a python package that solves linear structural time series models with Gaussian noises. Compared with some other popular Kalman filter packages written in python, `linkalman` has a combination of several advantages:

1 Account for partially and fully incomplete measurements

2 Flexible and convenient model structure

3 Robust and efficient implementation

4 Proper implementation for unknown priors

5 Built-in numerical and EM algorithm

6 Open-source with a comprehensive user manual

7 Modular design with intuitive model specification

Kalman filtering is a technique that provides an elegant solution to a wide range of time series problems. When I started learning Kalman filtering, I found most many existing Kalman filter packages are based on standard textbook models that are over-simplified for pedagogical purpose. In practice, a dynamic system may assume complex functional forms and may have incomplete measurements. In addition, solving a Kalman filter requires knowledge of initial conditions, which is rarely satisfied in real world problems. Finally, numerical implementation of Kalman filter algorithms are vulnerable to failures from rounding errors. `linkalman` package provides a solid solution to all these challenges.

It is noteworthy that `linkalman` is designed for linear state space problems with Gaussian errors and predetermined system dynamics. Therefore, it is not suitable for generalized non-linear state space problem, nor does it directly solve problems with unpredictable shifting in system dynamics (e.g. vehicle maneuver). The suitable solutions are particle filters and adaptive Kalman filters, respectively. In addition, `linkalman` is not optimized for any particular type of problems that are more easily solved with specialized algorithms. As a result, `linkalman` is most suitable for problems with moderate-sized data but complex structures for which it is difficult to calculate derivatives.

This user manual is a product of my learning on Kalman filters[1] and serves as the theoretical foundation of `linkalman`. Section 2 lays out the general structure of a Bayesian Structural Time Series (BSTS) model. Section 3 presents some common time series examples. Section 4 discusses the package design of `linkalman`. The rest of the document provides the technical details of algorithms related to `linkalman`. Section 5 and 6 provide detailed discussions on Kalman filters and Kalman smoothers, respectively. Section 7 concludes with a discussion of numerical methods and EM algorithms to estimate underlying parameters of a BSTS model.

2 Model Setup

Consider the following Linear Dynamic System:

$$\xi_{t+1} = F_t\xi_t + B_t x_t + v_t \tag{2.1}$$
$$y_t = H_t\xi_t + D_t x_t + w_t \tag{2.2}$$

Equation (2.1) governs a Markov state transition process. ξ_t $(m \times 1)$ is the latent state random vector at time t, x_t $(k \times 1)$ is the deterministic input signal, v_t $(m \times 1)$ is the exogenous process noise. We assume

[1] The primary reference is the wonderful textbook by (Durbin and S.J. Koopman 2012). I also refer to many other papers for technical details omitted in the textbook.

that $v_t \sim \mathcal{N}(0, Q_t)$ is white noise[2]. F_t $(m \times m)$, B_t $(m \times k)$, and v_{t+1} specify the transition dynamics between t and $t+1$.

Equation (2.2) is the measurement specification. y_t $(n \times 1)$ is the measurement vector at time t. w_t $(n \times 1)$ is the exogenous measurement noise. In addition to assuming $w_t \sim \mathcal{N}(0, R_t)$ is white noise, I also assume that $w_t \perp v_s \ \forall \ t, s \in \{0, 1, ..., T\}$. H_t $(n \times m)$ and D_t $(n \times k)$ dictate interaction among ξ_t, y_t and x_t.

Equations (2.1) and (2.2) characterize a BSTS model, with system matrices $\{F_t, B_t, H_t, D_t, Q_t, R_t\}$ parameterized by θ. System matrices may vary over time. For example cyclical pattern can be modeled by using Trigonometric Cycles (A. C. Harvey 1985).

The subscript t allows flexible model specification. For example, regression effects in time series models are placed in $B_t x_t$; an ARMA process can be modeled by $F_t \xi_t$ and v_t; additive outliers fit into $B_t x_t$.

3 Examples

In this section, I discuss some popular models in time series analysis. It is noteworthy that while a model may be defined in many ways, estimation of a model converges faster when the model is identifiable (i.e. there is no flat regions in likelihood functions near true parameters). In addition, one can also boost the model performance by imposing the correct specification. For example, if we know a process is AR(1), we may impose the transition parameter to take values between 0 and 1. Another technique to reduce the number of parameters is to use concentrated likelihood functions, but that generally involves calculating first order conditions. Alternatively, we may use sample statistics to directly infer system dynamics. For ARMA(p, q) process, we may use observed sample average to estimate population mean (D) directly, without having to reply on optimizers to estimate the value. Finally, it is always a good idea to add more data to the model so that the objective function is smoother; on-gradient optimizers are very slow if there are many parameters with little data. To achieve better performance, one should carefully design their model and refer to literatures when necessary.

3.1 AR(p)

An AR(p) process is defined as:

$$y_t - \mu = \phi_1(y_{t-1} - \mu) + \phi_2(y_{t-2} - \mu) + ... + \phi_p(y_{t-p} - \mu) + \varepsilon_t$$

Writing it as a state space model, we have:

$$F = \begin{pmatrix} \phi_1 & \phi_2 & \cdots & \phi_{p-1} & \phi_p \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}, Q = \begin{pmatrix} \sigma_\varepsilon & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}, D = \mu, \ B = 0, \ R = 0$$

If initial state values are not provided, `linkalman` is able to calculate the ergodic mean and covariance. In practice, y_t may not accurately measure ξ_t (e.g. sample averages of sales to measure expected sales). We can also add measurement noise to the process to allow for better signal extraction (i.e. $R > 0$).

[2]For process noises that are not white noise, we can re-write equation (2.1) to maintain independence across time. See Section (3) for examples

3.2 ARMA(p, q)

ARMA(p, q) generalizes AR(p) and MA(q). Define $r \equiv max(p, q+1)$, then ARMA(p, q) is of the form:

$$\begin{aligned} y_t - \mu =& \phi_1(y_{t-1} - \mu) + \phi_2(y_{t-2} - \mu) + ... + \phi_p(y_{t-r} - \mu) \\ &+ \varepsilon_{t-1} + \theta_1\varepsilon_{t-2} + ... + \theta_{r-1}\varepsilon_{t-r} \end{aligned}$$

Let $\phi_j = 0$ for $j > p$ and $\theta_i = 0$ for $i > q$, then we have a state space representation of ARMA(p, q) processes. There are multiple ways of writing an ARMA(p, q) process in to state space forms. I follow the form proposed by (Hamilton 1994) and rewrite ARMS(p, q) as:

$$F = \begin{pmatrix} \phi_1 & \phi_2 & \cdots & \phi_{r-1} & \phi_r \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}, Q = \begin{pmatrix} \sigma_\varepsilon & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & \theta_1 & \theta_2 & \cdots & \theta_{r-1} \end{pmatrix}, D = \mu,\ B = 0,\ R = 0$$

Effectively, F and Q define the following process:

$$(1 - \phi_1 L - \phi_2 L^2 - ... - \phi_r L^r)\xi_{t,1} = \varepsilon_{t-1}$$

where $\xi_{t,1}$ is the first element of ξ_t. H defines the following process:

$$y_t - \mu = (1 + \theta_1 L + \theta_2 L^2 + ... + \theta_{r-1} L^{r-1})\xi_t$$

Combining the two together, we have the conventional representation of ARMA(p, q):

$$\begin{aligned} &(1 - \phi_1 L - \phi_2 L^2 - ... - \phi_r L^r)(y_t - \mu) \\ =&(1 + \theta_1 L + \theta_2 L^2 + ... + \theta_{r-1} L^{r-1})\varepsilon_{t-1} \end{aligned}$$

3.3 Stochastic Linear Trend Model

A random walk process can be written as:

$$\alpha_t = \alpha_{t-1} + \beta_{t-1} + \varepsilon_{t-1}\beta_t = \beta_{t-1} + \delta_{t-1}y_t = \alpha_t + \omega_t$$

where β_t is regarded as the stochastic increment between $alpha_{t-1}$ and α_t. β_t follows a random walk, and without δ_t, movement of α does not allows turning-points.

In state space form, a stochastic linear trend process may be written as:

$$\begin{aligned} F = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, Q = \begin{pmatrix} \sigma_\varepsilon & 0 \\ 0 & \sigma_\delta \end{pmatrix},\ \xi_t = \begin{pmatrix} \alpha_t \\ \beta_t \end{pmatrix} \\ H = \begin{pmatrix} 1 & 0 \end{pmatrix}, R = \sigma_\omega \end{aligned}$$

A stochastic linear trend process is non-stationary, and therefore is difficult to forecast long-term outcomes.

3.4 Cyclical Model

A cyclical process exhibits recurring patterns. There are two ways of modeling the process: fixed valued cycles and trigonometric cycles. Following (A. C. Harvey 1985), I write a cyclical process as:

$$\begin{pmatrix} \phi_t \\ \phi_t^* \end{pmatrix} = \rho \begin{pmatrix} cos\lambda & sin\lambda \\ -sin\lambda & cos\lambda \end{pmatrix} \begin{pmatrix} \phi_{t-1} \\ \phi_{t-1}^* \end{pmatrix} + \begin{pmatrix} \omega_{t-1} \\ \omega_{t-1}^* \end{pmatrix}$$
$$y_t = \phi_t + \varepsilon_t$$

where $\lambda \in (0, \pi)$ determines the frequency of the cycle, and $\rho \in (0, 1)$ is the dampening parameter. If $\rho = 1$, the process is non-stationary. Let $\xi_t = (\phi_t, \phi_t^*)'$, and we can easily derive the state space representation, which I omit here. This approach fits well the cyclical behavior with the restriction of trigonometric formulation.

Another approach further relax the restriction at the cost of increasing number of states to be estimated. Instead of using sine and cosine functions to approximate a cycle, it directly assigns a state to each point in a cycle[3]. For example, weekly effects may be represented as:

$$\delta_t = -\sum_{j=1}^{6} \delta_{t-j} + \varepsilon_{t-1}$$

The formulation means the sum of weekly effect bounces around 0. If it is not 0, then we have a linear trend as well, which should be part of the linear trend model.

Note that a cyclical process is different from fixed effects models, where the cyclical pattern is accounted for by regression factors. For example, one may use dummy variables for weekdays and extract weekly fixed effect from D_t. The distinction is that the fixed effects approach try to estimate the average cyclical pattern of the time series, whereas the cyclical model is more time sensitive and relies more on the recent data to form the cyclical pattern for prediction.

3.5 Counterfactuals

Getting counterfactuals for treatments are of great interest in many areas[4]. Kalman filters/smoothers provides an elegant solution from a time series perspective. Treatments effectively alter system dynamics. If we treat the measurement during treatment periods as missing. We may find counterfactuals using filtered predictions. In addition, if the treatment effect is transient, we can also use the smoothed predictions for better performance.

4 `linkalman` Design

The first principle of `linkalman` is flexibility. I design `linkalman` such that it may be used for a wide variety of BSTS models. With minimal I/O restriction, users may write their customized system dynamics function, as well as plug-in solver function for additional flexibility. `linkalman` also features flexibility in data inputs. It is capable of handling both completely and partially missing measurements, which are common for real world data. In addition, whereas many other packages use approximation approach to handle unknown initial values, `linkalman` provides the rigorous diffuse filter/smoother techniques, but also gives users the option to feed in customized initial state values.

The second principle of `linkalman` is modularity. At the fundamental level, this user's manual documents all the technical details that other people may refer to in building their own customized systems, for example, in `C++` for better performance. The structure of the package also makes it easier to invoke

[3] See (Andrew C Harvey, Trimbur, and Van Dijk 2007) for details

[4] Please refer to (Brodersen and al. 2015), (A. C. Harvey 1985), and (Andrew C Harvey, Trimbur, and Van Dijk 2007) and papers they cite for more examples.

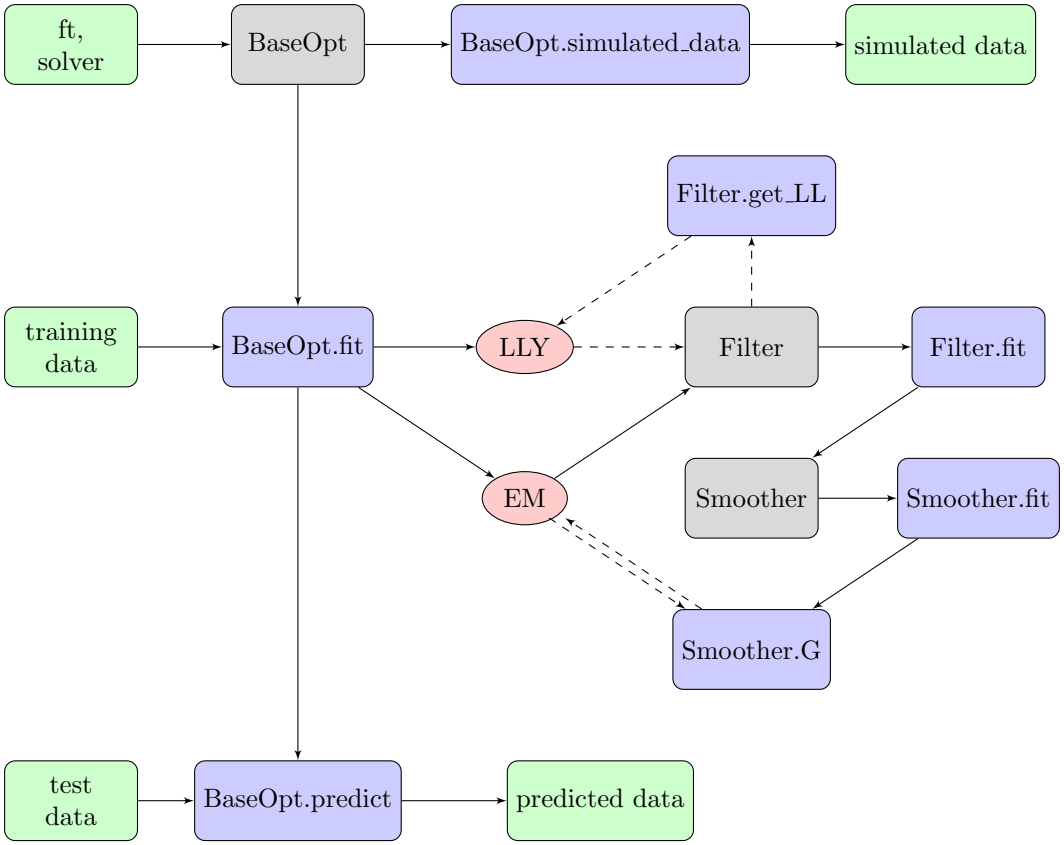

Figure 1: High level design of `linkalman`. Green blocks are I/O components, gray blocks are modules, purple blocks are class methods, and red circles are optimizer options. Solid lines indicate the order of model execution. Dotted lines indicate the modules/functions directly involved in numerical optimization.

part of the modules without relying on others. The building blocks are `Filter`, which produces states estimates of a time series given system matrices. It is used by the direct likelihood methods (LLY) for parameter estimation. The `Smoother` module takes a fitted `Filter` object as an argument and produces smoothed estimates and the objective function used by EM algorithm (EM). For `linkalman` to handle flexible system dynamics and missing data, I limit the solver choices to non-gradient based algorithms, which inevitably slow down the optimization in cases where gradients can be easily computed (e.g. AR(1) processes). Therefore, users may only use the `Filter` or `Smoother` modules without invoking the entire BaseOpt module, the primary model solver of `linkalman`. For example, if we have fully observed data and models with only unknown variances in noises, we are able to derive analytical score matrices from smoother outputs to greatly boost performance of EM algorithm as well as direct likelihood methods. Figure (1) shows the architecture of `linkalman`, which shows separability between the `BaseOpt` module and `Filter`/`Smoother` modules it calls.

The third principle of `linkalman` is numerical performance. Although `linkalman` is not designed to be the fastest solver for any specific type of problems, it nevertheless strives to achieve numerical robustness and computational efficiency[5]. First of all, I use both Joseph Representation and nearest positive semi-definite matrix technique to ensure positive semi-definiteness of covariance matrices. Secondly, I use sequential filtering and smoothing to further reduce the computational burden from matrix inverse.

[5]The effort in boosting performance is mostly algorithmic, so there is still ample room for improvement from the programming perspective. For example, the computationally heavy filter updating methods can be written in `Cython`.

5 Kalman Filter

5.1 Filtering with Known Initial Conditions

Given a set of measurements over time, and suppose θ is known. To predict y_{T+1}, we need to perform forward filtering. We start by making the following notations:

$$\begin{aligned} Y_t &\equiv \{y_1, y_2, ..., y_t\} \\ X_t &\equiv \{x_1, x_2, ..., x_t\} \\ \Xi_t &\equiv \{\xi_1, \xi_2, ..., \xi_t\} \\ \hat{\xi}_{t|s} &\equiv E(\xi_t | X_T, Y_s; \theta) \\ P_{t|s} &\equiv E[(\xi_t - \hat{\xi}_{t|s})(\xi_t - \hat{\xi}_{t|s})'] \end{aligned}$$

With the assumption that w_t and v_t are Gaussian white noises, conditional distribution of y_t and ξ_t are fully characterized by Gaussian process with $\hat{\xi}_{t|s}$ and $P_{t|s}$. Now consider at time t, we know Y_t and X_t, $\hat{\xi}_{t|t-1}$, and $P_{t|t-1}$. By equation (2.1), we have[6]:

$$\hat{\xi}_{t|t} = \hat{\xi}_{t|t-1} + K_t(y_t - H_t\hat{\xi}_{t|t-1} - D_t x_t) \tag{5.1}$$

$$K_t = P_{t|t-1}H_t^{'}\Upsilon_t^{-1} \tag{5.2}$$

$$\Upsilon_t \equiv H_t P_{t|t-1}H_t^{'} + R_t \tag{5.3}$$

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1} \tag{5.4}$$

$$\hat{\xi}_{t+1,t} = F_t\hat{\xi}_{t|t} + B_t x_t \tag{5.5}$$

$$P_{t+1,t} = F_t P_{t|t} F_t^{'} + Q_t \tag{5.6}$$

K_t is the famous Kalman gain matrix. In this subsection, we assume that initial conditions $\hat{\xi}_{1|0}$ and $P_{1|0}$ are known. The Kalman filter proceeds as follows:

1 Given θ, begin with initial value $\hat{\xi}_{1|0}$ and $P_{1|0}$.

2 Use equation (5.1) through (5.6) to calculate $\hat{\xi}_{2|1}$, and $P_{2|1}$.

3 Repeat step 2 for $t \in \{3, 4, ..., T\}$.

5.2 Joseph Form and Numerical Robustness

We define $P_{t|t}$ as a covariance matrix, which is a symmetric positive semi-definite (PSD) matrix, but in practice, we may get $P_{t|t}$ that is neither symmetric nor PSD, due to rounding errors. Look at equation (5.4) again. The subtraction makes calculating $P_{t|t}$ susceptible to rounding errors, sometimes even resulting in negative semi-definite matrix. I use the Joseph form to numerically enforce symmetric PSD.

The Joseph form (Bucy and Joseph 1968) of $P_{t|t}$ is[7]:

$$P_{t|t} = [I - K_t H_t]P_{t|t-1}[I - K_t H_t]' + K_t R_t K_t^{'}$$

If we guarantee $P_{t|t-1}$ and R_t to be PSD, then $P_{t|t}$ is PSD by construction. Define M_t, L_t, and d_t as:

$$\begin{aligned} M_t &\equiv F_t K_t \\ L_t &\equiv F_t - M_t H_t \\ d_t &\equiv y_t - H_t\hat{\xi}_{t|t-1} - D_t x_t \end{aligned}$$

[6] The derivation closely follows Chapter 13 in (Hamilton 1994). In particular, derivation of equation (5.4) is given in Appendix A.1.

[7] For a detailed proof, see Appendix A.2

We are able to obtain a more concise updating rule:

$$\hat{\xi}_{t+1,t} = F_t \hat{\xi}_{t|t-1} + M_t d_t \tag{5.7}$$

$$P_{t+1,t} = L_t P_{t|t-1} L_t^{'} + M_t R_t M_t^{'} + Q_t \tag{5.8}$$

Using equation (5.7) through (5.8) recursively, we may implement Kalman filtering in the Joseph form.

5.3 Initialization with Diffuse Filtering

So far we have assumed that we know values of $\hat{\xi}_{1|0}$ and $P_{1|0}$. In practice, such information are rarely available. If instead, we know that the time series is stationary, we can use the unconditional mean and variance for $\hat{\xi}_{1|0}$ and $P_{1|0}$. If the time series is at least partially non-stationary, then we should assume no prior information and set $\hat{\xi}_{1|0} = 0$ and $P_{1|0} = \infty$ for the non-stationary part of ξ_t. A commonly used solution is to set $P_{1|0}$ to some large value (e.g. $\kappa = 10^7$) to approximate infinity, but this technique is difficult to implement when the initial condition structure is complex[8]. Better algorithms exist for exact initialization, and I will discuss one such algorithm that is implemented by `linkalman`.

Consider $\hat{\xi}_{1|0}$ again. Following (Siem Jan Koopman 1997), we define $\hat{\xi}_{1|0}$ as:

$$\hat{\xi}_{1|0} = a + A\eta + \Pi\varepsilon \tag{5.9}$$

A and Π are selection matrices, $\eta \sim \mathcal{N}(0, \kappa I)$, and $\varepsilon \sim \mathcal{N}(0, Q_*)$. Set 0 for elements in a corresponding to A, and stationary means for elements corresponding to Π. Q_* is the ergodic covariance of the stationary component of the model (or process with known initial values). Essentially, equation (5.9) groups $\hat{\xi}_{1|0}$ into two categories. If $\hat{\xi}^i_{1|0}$, the i-th element of $\hat{\xi}_{1|0}$, is non-stationary and not known, it has a distribution $\mathcal{N}(0, \kappa)$. With $\kappa \to \infty$, $\mathcal{N}(0, \kappa)$ captures the fact that we know nothing about the initial value of a non-stationary process[9]. If on the other hand, the initial state of $\hat{\xi}^i_{1|0}$ is either known or stationary, It has a proper distribution with unconditional mean and covariances[10].

From equation (5.9), we have:

$$\hat{\xi}_{1|0} = a \tag{5.10}$$

$$P_{1|0} = \kappa P_\infty + P_* \tag{5.11}$$

where $P_\infty = AA^{'}$ and $P_* = \Pi Q_* \Pi^{'}$. By linearity of the filtering process, we can write $P_{t|t-1}$ as:

$$P_{t|t-1} = \kappa P_{\infty,t} + P_{*,t} + \mathcal{O}(\kappa^{-1}) \tag{5.12}$$

where $\mathcal{O}(\kappa^{-1})$ is a function $f(\kappa) < \infty$ as $\kappa \to \infty$. We can write Υ_t as:

$$\Upsilon_t = \kappa \Upsilon_{\infty,t} + \Upsilon_{*,t} + \mathcal{O}(\kappa^{-1})$$

In what follows, I describe the updating rule for exact initialization, for technical details, please refer to Appendix A.3.

Given $\hat{\xi}_{t|t-1}$ and $P_{t|t-1}$, if $\Upsilon_{\infty,t} \neq 0$:

1. Calculate K_t using equation (A.5) to (A.7)
2. Calculate $\hat{\xi}_{t|t}$ and $P_{t|t}$ from equation (A.8) and (A.9)

If $\Upsilon_{\infty,t} = 0$:

[8] See Section 3 of (Doan 2010) for further discussions.

[9] The specification of equation (5.9), despite its seemingly arbitrariness, is valid, because of the invariance property of diffuse prior. Interested readers can refer to Appendix B of (Doan 2010) for details.

[10] If x_t is involved in the state transition process then a process is not stationary unless its transition is not affected by $\{x_t\}_{t\in(-\infty,\infty)}$.

1 Calculate K_t using equation (A.10)

2 Calculate $\hat{\xi}_{t|t}$ and $P_{t|t}$ from equation (A.11) and (A.12)

We can update the diffuse Kalman filter with the new expressions for K_t, $\hat{\xi}_{t+1,t}$, and $P_{t+1,t}$.

5.4 Transition to the Usual Kalman Filter

Diffused initial state is to represent the fact the we have no prior knowledge of initial the state of non-stationary processes. As a result, that state estimate will be dominated by initial data. In other words, the few initial observations are used for form the informative initial conditions, and the rest of the data are used to perform usual Kalman filtering. (De Jong 1991) and (Durbin and S.J. Koopman 2003) show that a diffuse Kalman filter will degenerate to a usual Kalman filter after a few time periods t_q, which in general is the number of diffuse initial states.

A diffuse Kalman filter degenerates to a usual one if $P_{\infty,t} = 0$. In practice, we may check if $P_{\infty,t} = 0$, but it is subject to numerical errors. (Helske 2016) implemented a more robust algorithms in his R package `KFAS`. I provide the procedure below, and interested readers may refer to Appendix A.4 for a simple proof[11].

Here I focus on the univariate measurement case, because it is less complex and convertible from multivariate Kalman filters. Define $q_1 \equiv rank(P_{\infty,1})$, then by construction of $P_{1,0}$, q_1 has rank q, where q is the number of diffuse state variables. Let $\varepsilon_t > 0$ be some small threshold value[12] such that if $\Upsilon_{\infty,t} < \varepsilon_t$, we treat $\Upsilon_{\infty,t} = 0$. Define If $\Upsilon_{\infty,t} \geq \varepsilon_t$, then:

$$q_{t+1} = min[q_t - 1, rank(P_{\infty,t+1})]$$

When $q_t = 0$ for $t = t_q$, the diffuse Kalman filter degenerates into a usual one. This formulation guarantees degeneration to regular Kalman filters.

5.5 Missing Measurements and Sequential Filtering

So far we have assumed that we observe complete measurement y_t for each t. If we have incomplete measurements, we can instead update the Kalman Filter sequentially (see (Durbin and S.J. Koopman 2012) Section 6.4 and Durbin and S.J. Koopman 2000 for details), based only on observed measurements. Sequential filtering also boosts speed dramatically in the presence of large measurement dimensions, reducing cost from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$. In addition, univariate treatment of Kalman filters allows clean formulation with diffuse priors. The techniques developed in this subsection is implemented in `linkalman` and will be discussed in detail.

Denote $y_{t:(i)}$ as the i-th observed measurement at time t, then $y_{t:(1)}$ is the first non-missing univariate measurement at time t. Let (n_t) as the index for the last non-missing univariate measurement at time t. Define $\hat{\xi}_{t:(i)}$ and $P_{t:(i)}$ as the state estimates after updating univariate measurement (i) at time t:

$$\hat{\xi}_{t:(i)} \equiv E(\xi_t | Y_{t-1}, y_{t:(1)}, ..., y_{t:(i-1)}, X_T; \theta)$$
$$P_{t:(i)} \equiv Var(\xi_t | Y_{t-1}, y_{t:(1)}, ..., y_{t:(i-1)}, X_T; \theta)$$

[11] For a complete treatment, refer to (Siem Jan Koopman 1997) for details.

[12] In practice, I follow (Helske 2016) and set $\varepsilon_t = \varepsilon \times min(|H_t|; |H_t| > \varepsilon)^2$, where ε =1e-8 is some base threshold, and $min(|H_t|; |H_t| > \varepsilon)$ is the minimum of the absolute values in H_t that is not 0.

In addition, at the beginning of time t before any Kalman updating, we have:

$$\begin{aligned}\hat{\xi}_{t:(1)} &\equiv E(\xi_t|Y_{t-1};\theta) \\ &= F_{t-1}\hat{\xi}_{t-1:(n_{t-1}+1)|t-1:(n_{t-1})} + B_{t-1}x_{t-1} \qquad (5.13)\\ P_{t:(1)} &\equiv Var(\xi_t|Y_{t-1};\theta) \\ &= F_{t-1}P_{t-1:(n_{t-1}+1)|t-1:(n_{t-1})}F_{t-1} + Q_{t-1} \qquad (5.14)\end{aligned}$$

For sequential filtering to work, we need R_t to be diagonal. In the case where R_t is not diagonal, we can use LDL Decomposition[13] to transform the original BSTS model into one with independent measurement noise. Given that R_t is PSD, we have $R_t = l_t\Lambda_t l_t^{'}$. Pre-multiply equation (2.2) by l_t^{-1}, and we have[14]:

$$\tilde{y}_t = \tilde{H}_t\xi_t + \tilde{D}_t x_t + \tilde{w}_t \qquad (5.15)$$

where $\tilde{(\cdot)}_t = l_t^{-1}(\cdot)_t$. In the following sections, I will omit the $(\sim)$ sign, and assume R_t is always diagonal.

For a BSTS model with diagonal R_t, equation (2.2) then becomes:

$$y_t = \begin{pmatrix} y_{t:(1)} \\ \vdots \\ y_{t:(n_t)} \end{pmatrix} = \begin{pmatrix} H_{t:(1)}\xi_t + D_{t:(1)}x_t + w_{t:(1)} \\ \vdots \\ H_{t:(n_t)}\xi_t + D_{t:n_t}x_t + w_{t:(n_t)} \end{pmatrix}$$

where $H_{t:(i)}$ and $D_{t:(i)}$ are the (i)-th row of H_t and D_t.

We initialize the measurement update process with $\hat{\xi}_{1:(1)}$ and $P_{1:(1)}$ from equation (5.10) and (5.11). Define $\Upsilon_{t:(i)}$ as:

$$\Upsilon_{t:(i)} = H_{t:(i)}P_{t:(i)}H_{t:(i)}^{'} + R_{t:(i)}$$

where $R_{t:(i)}$ is the (i)-th diagonal value of R_t. For $\Upsilon_{t:(i)}$, we have:

$$\begin{aligned}\Upsilon_{\infty,t:(i)} &= H_{t:(i)}P_{\infty,t:(i)}H_{t:(i)}^{'} \\ \Upsilon_{*,t:(i)} &= H_{t:(i)}P_{*,t:(i)}H_{t:(i)}^{'} + R_{t:(i)}\end{aligned}$$

For each successive measurement (i), if $\Upsilon_{\infty,t:(i)} \neq 0$, we have:

$$\hat{\xi}_{t:(i+1)} = \hat{\xi}_{t:(i)} + K_{t:(i)}^{(0)}d_{t:(i)} + \mathcal{O}(\kappa^{-1}) \qquad (5.16)$$

$$\begin{aligned}P_{t:(i+1)} =& \kappa(I - K_{t:(i)}^{(0)}H_{t:(i)})P_{\infty,t:(i)}(I - K_{t:(i)}^{(0)}H_{t:(i)})^{'} \\ &+ (I - K_{t:(i)}^{(0)}H_{t:(i)})P_{*,t:(i)}(I - K_{t:(i)}^{(0)}H_{t:(i)})^{'} \\ &+ K_{t:(i)}^{(0)}R_{t:(i)}K_{t:(i)}^{(0)'} + \mathcal{O}(\kappa^{-1})\end{aligned} \qquad (5.17)$$

where

$$\begin{aligned}d_{t:(i)} &= y_{t:(i)} - H_{t:(i)}\hat{\xi}_{t:(i)} - D_{t:(i)}x_t \\ K_{t:(i)}^{(0)} &= \frac{P_{\infty,t:(i)}H_{t:(i)}^{'}}{\Upsilon_{\infty,t:(i)}} = \frac{P_{\infty,t:(i)}H_{t:(i)}^{'}}{H_{t:(i)}P_{\infty,t:(i)}H_{t:(i)}^{'}}\end{aligned}$$

If $\Upsilon_{\infty,t:(i)} = 0$ and $P_{\infty,t:(i-1)} \neq 0$, we have:

$$\hat{\xi}_{t:(i+1)} = \hat{\xi}_{t:(i)} + K_{t:(i)}^{(*)}d_{t:(i)} + \mathcal{O}(\kappa^{-1}) \qquad (5.18)$$

$$\begin{aligned}P_{t:(i+1)} =& \kappa P_{\infty,t:(i)} + (I - K_{t:(i)}^{(*)}H_{t:(i)})P_{*,t:(i)}(I - K_{t:(i)}^{(*)}H_{t:(i)})^{'} \\ &+ K_{t:(i)}^{(*)}R_{t:(i)}K_{t:(i)}^{(*)'} + \mathcal{O}(\kappa^{-1})\end{aligned} \qquad (5.19)$$

[13] In practice, I perform LDL Decomposition with `scipy.linalg.ldl`.

[14] Due to special properties of triangular matrices, in practice, I perform matrix inverse on triangular matrix with `scipy.linalg.lapack.dtrtri` subroutine.

where

$$K_{t:(i)}^{(*)} = \frac{P_{*,t:(i)} H_{t:(i)}^{'}}{\Upsilon_{*,t:(i)}} = \frac{P_{*,t:(i)} H_{t:(i)}^{'}}{H_{t:(i)} P_{*,t:(i)} H_{t:(i)}^{'} + R_{t:(i)}}$$

If $\Upsilon_{\infty,t:(i)} = 0$ and $P_{\infty,t:(i-1)} = 0$, the diffuse Kalman filter degenerates into a usual one, and instead of using equation (5.19), we use:

$$P_{t:(i+1)} = (I - K_{t:(i)}^{(*)} H_{t:(i)}) P_{*,t:(i)} (I - K_{t:(i)}^{(*)} H_{t:(i)})^{'} + K_{t:(i)}^{(*)} R_{t:(i)} K_{t:(i)}^{(*)'} \tag{5.20}$$

Now we may proceed with sequential filtering as follows:

1. Initialize state conditions using equation (5.10) and (5.11)
2. For period t, calculate $\hat{\xi}_{t:(0)}$ and $P_{t:(0)}$ using equation (5.13) and (5.14)
3. Use LDL transformation to obtain diagonalized measurement equation (5.15)
4. If $\Upsilon_{\infty,t:(i)} \neq 0$, use equation (5.16) and (5.17) to update $\hat{\xi}_{t:(i+1)}$ and $P_{t:(i+1)}$
5. If $\Upsilon_{\infty,t:(i)} = 0$ and $\hat{q}_{t:(i)} > 0$, use equation (5.18) and (5.19) to update $\hat{\xi}_{t:(i+1)}$ and $P_{t:(i+1)}$
6. If $\hat{q}_{t:(i)} = 0$, use equation (5.18) and (5.20) to update $\hat{\xi}_{t:(i+1)}$ and $P_{t:(i+1)}$
7. Update $\hat{q}_{t:(i+1)} = min[\hat{q}_{t:(i)} - 1, q_{t:(i+1)}]$. If $i = n_t$, $\hat{q}_{t+1:(1)} = \hat{q}_{t:(n_t+1)}$
8. Repeat step (2) through (7) for $t \in \{1, 2, ..., T\}$

An alternative view of the procedure is to treat y_t as $(y_{t:(1)}, ..., y_{t:(n_t)})$, with $F_{t:(i)} = I$, $Q_{t:(i)} = 0$, and $B_{t:(i)} = 0$ for $i \in 1, ..., n_t$. The transition from $[t : (n_t)]$ to $[t+1 : (1)]$ with transition matrices F_t, Q_t, and B_t is broken down into two steps: $[t : (n_t)]$ to $[t : (n_t + 1)]$ with $F_{t:(n_t)} = I$, $Q_{t:(n_t)} = 0$, and $B_{t:(n_t)} = 0$, then $[t : (n_t + 1)]$ to $[t + 1 : (1)]$ with $F_{t:(n_t+1)} = F_t$, $Q_{t:(n_t+1)} = Q_t$, and $B_{t:(n_t+1)} = B_t$. Note that the second step does not involve measurement update.

6 Kalman Smoother

6.1 State Smoother

In Section 5, we use Kalman Filter to find $\{\hat{\xi}_{t|t}, K_t, P_{t|t}, \hat{\xi}_{t|t-1}, P_{t|t-1}\}$ for each t. If a dataset is given, we have the entire measurement sequence Y_T. Kalman Smoother is a technique of integrating information up to T to infer ξ_t at time t, $\hat{\xi}_{t|T}$ and $P_{t|T}$.

Suppose in addition to state estimates from Kalman Filter, we also want to know $\hat{\xi}_{t|T}$ and $P_{t|T}$. The technique for computing $\hat{\xi}_{t|T}$ and $P_{t|T}$ is called backwards smoothing[15]. Here I present the iterative formula for Kalman smoothing. Following derivations in Appendix B.1, I define two auxiliary variables: r_t and N_t, where $r_T = 0$, $N_T = 0$, and:

$$r_{t-1} = H_t^{'} \Upsilon_t^{-1} d_t + L_t^{'} r_t \tag{6.1}$$

$$N_{t-1} = H_t^{'} \Upsilon_t^{-1} H_t + L_t^{'} N_t L_t \tag{6.2}$$

Using $\{r_t\}_{1,...,T}$ and $\{N_t\}_{1,...,T}$, we have the iterative formulation for $\hat{\xi}_{t|T}$ and $P_{t|T}$:

$$\hat{\xi}_{t|T} = \hat{\xi}_{t|t-1} + P_{t|t-1} r_{t-1} \tag{6.3}$$

$$P_{t|T} = P_{t|t-1} - P_{t|t-1} N_{t-1} P_{t|t-1} \tag{6.4}$$

[15] See (De Jong 1989) for details. I also provided a proof in Appendix B.1 that is consistent with notations in this document and has more details.

As we will see in Section 7.3 and Appendix C.3, we also need to calculate covariance matrices of smoothed estimators. Define $P_{t,s|n}$ as:

$$P_{t,s|n} \equiv Cov(\xi_t, \xi_s | Y_n, X_T; \theta)$$

Following (Siem Jan Koopman and Shephard 1992a), (De Jong and Mackinnon 1988), and (De Jong 1989)[16], we have for $1 \leq t < j \leq T$:

$$P_{t,j|T} = P_{t|t-1} \prod_{i=t}^{j-1} L_i^{'} (I - N_{j-1} P_{j|j-1})$$

In particular, we are interested in:

$$P_{t,t+1|T} = P_{t+1,t|T}^{'} = P_{t|t-1} L_t^{'} (I - N_t P_{t+1|t}) \tag{6.5}$$

To perform backwards smoothing, we can simply start at time T with $r_T = 0$ and $N_T = 0$, then use equation (6.1) through (6.5) for $t \in \{T-1, T-2, ..., 1\}$ to obtain $\{\hat{\xi}_{t|T}\}_{1,2,...,T}$, $\{P_{t|T}\}_{1,2,...,T}$ and $\{P_{t,t+1|T}\}_{1,2,...,T}$.

6.2 Rounding Errors and Nearest PSD

Note that for Kalman Smoother, we don't have Joseph Form formulation, and therefore we may still get non-PSD covariance matrix and cause the smoother to fail. By construction, I guarantee symmetry of covariance matrices. Therefore, we may check PSD using Cholesky Decomposition, which is efficient under symmetry conditions.

If a covariance matrix C is not PSD, we can use the nearest PSD matrix with the techniques developed by (Higham 1988). Omitting technical details, it amounts to replacing C with $\bar{C}$, where $\bar{A}$ is be calculated as:

$$\begin{aligned} C &= USV^{'} \\ \Sigma &= VSV^{'} \\ \bar{C} &= \frac{\Sigma + \Sigma^{'} + C + C^{'}}{4} \end{aligned}$$

where the first equality is the SVD decomposition of C. By construction, N_{t-1} is PSD, then we only need to correct $P_{t|T}$ if necessary.

6.3 Diffuse Smoother

When initial conditions are unknown, we can use diffuse smoothers. Similar to diffuse filters, we expand $P_{t|T}$, r_{t-1}, and N_{t-1}. Appendix B.3 provide detailed derivation of recursive formula, so here I will just give the result. First of all, I give the expression for $\hat{\xi}_{t|T}$, $P_{t|T}$, and $P_{t,t+1|T}$ as:

$$\hat{\xi}_{t|T} = \hat{\xi}_{t|t-1} + P_{*,t} r_{t-1}^{(0)} + P_{\infty,t} r_{t-1}^{(1)} + \mathcal{O}(\kappa^{-1}) \tag{6.6}$$

$$P_{t|T} = P_{*,t} - \langle P_{\infty,t} N_{t-1}^{(1)} P_{*,t} \rangle - P_{*,t} N_{t-1}^{(0)} P_{*,t} - P_{\infty,t} N_{t-1}^{(2)} P_{\infty,t} + \mathcal{O}(\kappa^{-1}) \tag{6.7}$$

$$\begin{aligned} P_{t,t+1|T} =& (P_{*,t} L_t^{(0)'} + P_{\infty,t} L_t^{(1)'})(I - N_t^{(1)} P_{\infty,t+1} - N_t^{(0)} P_{*,t+1}) \\ & - P_{\infty,t} L_t^{(0)'} (N_t^{(2)} P_{\infty,t+1} + N_t^{(1)} P_{*,t+1}) + \mathcal{O}(\kappa^{-1}) \end{aligned} \tag{6.8}$$

Now we need to compute r_{t-1}, and N_{t-1}. If $\Upsilon_{\infty,t} \neq 0$, we use:

[16] See Appendix B.2 for details.

1 Begin with $r^{(0)}_{t_q} = r_{t_q}$, $r^{(1)}_{t_q} = 0$, $N^{(0)}_{t_q} = N_{t_q}$, and $N^{(1)}_{t_q} = N^{(2)}_{t_q} = 0$

2 Use equation (B.1) through (B.2) to compute expansions of r_{t-1}

3 Use equation (B.3) through (B.5) to compute expansions of N_{t-1}

If $\Upsilon_{\infty,t} = 0$, we instead use[17]:

1 Use equation (B.6) through (B.7) to compute expansions of r_{t-1}

2 Use equation (B.8) through (B.10) to compute expansions of N_{t-1}

We can then calculate $\hat{\xi}_{t|T}$, $P_{t|T}$, and $P_{t,t+1|T}$ through equation (6.6) through (6.8).

6.4 Sequential Smoother

As with the case of Kalman filtering, calculating Υ_t^{-1} is expensive and subject to failure due to rounding errors. (Durbin and S.J. Koopman 2000) proposes sequential smoothing to greatly improved the efficiency and robustness of Kalman Smoothers. The univariate approach in Kalman filter essentially treats observations flowing in one at a time, so we can readily adapt our smoother.

Define $y_{t:(i)}$ as the i-th non-missing measurement of y_t. To derive the univariate formula for diffuse smoothers, we can treat smoothing between two observations within the same period as if the transition matrix is F_{t-1} between $t-1:(n_t)$ and $t:(1)$, and is an identity matrix if otherwise. In addition, for notational purpose, we can add $r_{t:(0)}$ and $N_{t:(0)}$. It is easy to see $r_{t:(0)} = r_t$, and $N_{t:(0)} = N_t$. For the case $(i) \leq (n_t)$, if $\Upsilon_{\infty,t:(i)} \neq 0$, the recursive formula for r_{t-1} and N_{t-1} are:

$$r^{(0)}_{t:(i-1)} = L^{(0)'}_{t:(i)} r^{(0)}_{t:(i)} \tag{6.9}$$

$$\begin{aligned} r^{(1)}_{t:(i-1)} =& H'_{t:(i)} \left(\frac{d_{t:(i)}}{\Upsilon_{\infty,t:(i)}} - K^{(1)'}_{t:(i)} r^{(0)}_{t:(i)} \right) + L^{(0)'}_{t:(i)} r^{(1)}_{t:(i)} \\ =& \frac{H'_{t:(i)} d_{t:(i)}}{\Upsilon_{\infty,t:(i)}} + L^{(1)'}_{t:(i)} r^{(0)}_{t:(i)} + L^{(0)'}_{t:(i)} r^{(1)}_{t:(i)} \end{aligned} \tag{6.10}$$

$$N^{(0)}_{t:(i-1)} = L^{(0)'}_{t:(i)} N^{(0)}_{t:(i)} L^{(0)}_{t:(i)} \tag{6.11}$$

$$N^{(1)}_{t:(i-1)} = \frac{H'_{t:(i)} H_{t:(i)}}{\Upsilon_{\infty,t:(i)}} + \langle L^{(1)'}_{t:(i)} N^{(0)}_{t:(i)} L^{(0)}_{t:(i)} \rangle + L^{(0)'}_{t:(i)} N^{(1)}_{t:(i)} L^{(0)}_{t:(i)} \tag{6.12}$$

$$\begin{aligned} N^{(2)}_{t:(i-1)} =& - \frac{H'_{t:(i)} H_{t:(i)} \Upsilon_{*,t:(i)}}{\Upsilon^2_{\infty,t:(i)}} + \langle L^{(1)'}_{t:(i)} N^{(1)}_{t:(i)} L^{(0)}_{t:(i)} \rangle \\ & + L^{(0)'}_{t:(i)} N^{(2)}_{t:(i)} L^{(0)}_{t:(i)} + L^{(1)'}_{t:(i)} N^{(0)}_{t:(i)} L^{(1)}_{t:(i)} \end{aligned} \tag{6.13}$$

$$L^{(0)}_{t:(i)} = I - K^{(0)}_{t:(i)} H_{t:(i)} = I - \frac{P_{\infty,t:(i)} H'_{t:(i)} H_{t:(i)}}{H_{t:(i)} P_{\infty,t:(i)} H'_{t:(i)}} \tag{6.14}$$

$$L^{(1)}_{t:(i)} = - K^{(1)}_{t:(i)} H_{t:(i)} = - \frac{(P_{*,t:(i)} H'_{t:(i)} - K^{(0)}_{t:(i)} \Upsilon_{*,t:(i)}) H_{t:(t)}}{H_{t:(i)} P_{\infty,t:(i)} H'_{t:(i)}} \tag{6.15}$$

[17] Note that we do not have to provide initial values in this case, as diffuse filters always end with $\Upsilon_{\infty,t} \neq 0$.

Similarly, if $\Upsilon_{\infty,t:(i)} = 0$ and $t < t_q$, the recursive formula for $r_{t:(i)}$ and $N_{t:(i)}$ are:

$$\begin{aligned}
r^{(0)}_{t:(i-1)} &= \frac{H^{'}_{t:(i)} d_{t:(i)}}{H_{t:(i)} P_{*,t:(i)} H^{'}_{t:(i)} + R_{t:(i)}} + (I - K^{(*)}_{t:(i)} H_{t:(i)})^{'} r^{(0)}_{t:(i)} \\
&= \frac{H^{'}_{t:(i)} d_{t:(i)}}{\Upsilon_{*,t:(i)}} + L^{(*)'}_{t:(i)} r^{(0)}_{t:(i)} && (6.16) \\
r^{(1)}_{t:(i-1)} &= r^{(1)}_{t:(i)} && (6.17) \\
N^{(0)}_{t:(i-1)} &= \frac{H^{'}_{t:(i)} H_{t:(i)}}{H_{t:(i)} P_{*,t:(i)} H^{'}_{t:(i)} + R_{t:(i)}} + (I - K^{(*)}_{t:(i)} H_{t:(i)})^{'} N^{(0)}_{t:(i)} (I - K^{(*)}_{t:(i)} H_{t:(i)}) \\
&= \frac{H^{'}_{t:(i)} H_{t:(i)}}{\Upsilon_{*,t:(i)}} + L^{(*)'}_{t:(i)} N^{(0)}_{t:(i)} L^{(*)}_{t:(i)} && (6.18) \\
N^{(1)}_{t:(i-1)} &= (I - K^{(*)}_{t:(i)} H_{t:(i)})^{'} N^{(1)}_{t:(i)} (I - K^{(*)}_{t:(i)} H_{t:(i)}) \\
&= L^{(*)'}_{t:(i)} N^{(1)}_{t:(i)} L^{(*)}_{t:(i)} && (6.19) \\
N^{(2)}_{t:(i-1)} &= N^{(2)}_{t:(i)} && (6.20) \\
K^{(*)}_{t:(i)} &= \frac{P_{*,t:(i)} H^{'}_{t:(i)}}{H_{t:(i)} P_{*,t:(i)} H^{'}_{t:(i)} + R_{t:(i)}} && (6.21) \\
L^{(*)}_{t:(i)} &= I - K^{(*)}_{t:(i)} H_{t:(i)} && (6.22)
\end{aligned}$$

For the regular Kalman smoother where $t \geq t_q$, the formula are:

$$\begin{aligned}
r^{(0)}_{t:(i-1)} &= \frac{H^{'}_{t:(i)} d_{t:(i)}}{\Upsilon_{*,t:(i)}} + L^{(*)'}_{t:(i)} r^{(0)}_{t:(i)} && (6.23) \\
N^{(0)}_{t:(i-1)} &= \frac{H^{'}_{t:(i)} H_{t:(i)}}{\Upsilon_{*,t:(i)}} + L^{(*)'}_{t:(i)} N^{(0)}_{t:(i)} L^{(*)}_{t:(i)} && (6.24)
\end{aligned}$$

Essentially, we only update $r^{(0)}_{t:(i-1)}$ and $N^{(0)}_{t:(i-1)}$.

For the case $(i) = (0)$, then the recursive formula for $r_{t-1:(n_{t-1})}$ and $N_{t-1:(n_{t-1})}$ become:

$$\begin{aligned}
r^{(0)}_{t-1:(n_{t-1})} &= F^{'}_{t-1} r^{(0)}_{t:(0)} && (6.25) \\
r^{(1)}_{t-1:(n_{t-1})} &= F^{'}_{t-1} r^{(1)}_{t:(0)} && (6.26) \\
N^{(0)}_{t-1:(n_{t-1})} &= F^{'}_{t-1} N^{(0)}_{t:(0)} F_{t-1} && (6.27) \\
N^{(1)}_{t-1:(n_{t-1})} &= F^{'}_{t-1} N^{(1)}_{t:(0)} F_{t-1} && (6.28) \\
N^{(2)}_{t-1:(n_{t-1})} &= F^{'}_{t-1} N^{(2)}_{t:(0)} F_{t-1} && (6.29)
\end{aligned}$$

Now we can find the univariate versions of $\hat{\xi}_{t|T}$, $P_{t|T}$, and $P_{t,t+1|T}$ by applying equation (6.9) through (6.29) to equation (6.6) through (6.8). In particular, in the case of univariate smoother, $\hat{\xi}_{t:(i)} = \hat{\xi}_{t:(1)}$ and $P_{t:(i)} = P_{t:(1)}$, because there is no change in t[18]. In addition, EM algorithm requires computing $P_{t,t+1|T}$ even if t or $t+1$ has fully missing measurements. Unlike Kalman filters, we only need to store one set

[18]It is easy to show the equivalence by expanding the expression of $\hat{\xi}^{(i+1)}_{t|T}$ and comparing it against $\hat{\xi}^{(i)}_{t|T}$.

of state values for each t. Define $P_{t,t+1|T} \equiv P_{t:(n_t+1),t+1:(1)|T}$. If $t < t_q$, we have:

$$\hat{\xi}_{t:(1)|T} = \hat{\xi}_{t:(1)} + P_{*,t:(1)} r^{(0)}_{t:(0)} + P_{\infty,t:(1)} r^{(1)}_{t:(0)} \tag{6.30}$$

$$\begin{aligned} P_{t:(1)|T} =& P_{*,t:(1)} - \langle P_{\infty,t:(1)} N^{(1)}_{t:(0)} P_{*,t:(1)} \rangle - P_{*,t:(1)} N^{(0)}_{t:(0)} P_{*,t:(1)} \\ & - P_{\infty,t:(1)} N^{(2)}_{t:(0)} P_{\infty,t:(1)} \end{aligned} \tag{6.31}$$

$$\begin{aligned} P_{t,t+1|T} =& P_{*,t:(n_t+1)} F_t^{'} (I - N^{(1)}_{t+1:(0)} P_{\infty,t+1:(1)} - N^{(0)}_{t+1:(0)} P_{*,t+1:(1)}) \\ & - P_{\infty,t:(n_t+1)} F_t^{'} (N^{(2)}_{t+1:(0)} P_{\infty,t+1:(1)} + N^{(1)}_{t+1:(0)} P_{*,t+1:(1)}) \end{aligned} \tag{6.32}$$

If $t \geq t_q$, we have:

$$\hat{\xi}_{t:(1)|T} = \hat{\xi}_{t:(1)} + P_{t:(1)} r_{t:(0)} \tag{6.33}$$

$$P_{t:(1)|T} = P_{t:(1)} - P_{t:(1)} N_{t:(0)} P_{t:(1)} \tag{6.34}$$

$$P_{t,t+1|T} = P_{t:(n_t+1)} F_t^{'} (I - N_{t+1:(0)} P_{t+1:(1)}) \tag{6.35}$$

Recall that the state estimates are the same between $t:(i)$ and $t:(i+1)$ for any given time, so we may choose any measurement index (i). Equation (6.30) and (6.35) are preferred here because their indexing are consistent with those in Kalman filters and smoothers, and are capable of handling fully missing measurements.

6.5 Interpolating Missing Measurements

Quite often we would like to know the smoothed value of missing measurements (e.g. counterfactuals for studying treatment effects), this subsection provides formula for smoothed means and variances of y_t (interested readers may refer to Appendix B.4 for details):

$$E(y^{(1)}_t | Y^{(1)}_t) = y^{(1)}_t \tag{6.36}$$

$$Var(y^{(1)}_t | Y^{(1)}_t) = 0 \tag{6.37}$$

$$E(y^{(2)}_t | Y^{(1)}_t) = H^{(2)}_t \hat{\xi}_{t|T} + D^{(2)}_t x_t + \mathcal{B}_t \epsilon_t \tag{6.38}$$

$$Var(y^{(2)}_t | Y^{(1)}_t) = R^{(2,2)}_t - R^{(2,1)}_t \left(R^{(1,1)}_t\right)^{-1} R^{(1,2)}_t + \left(H^{(2)}_t - \mathcal{B}_t H^{(1)}_t\right) P_{t|T} \left(H^{(2)}_t - \mathcal{B}_t H^{(1)}_t\right)^{'} \tag{6.39}$$

Using equation (6.36) through (6.39). We have the smoothed distribution of missing measurements. Note that in practice, I restore the index of smoothed estimates of y_t to their original order. It is also worth noting that when all measurements at time t are missing, equation (6.38) and (6.39) becomes:

$$\begin{aligned} E(y^{(2)}_t | Y^{(1)}_t) &= H_t \hat{\xi}_{t|T} + D_t x_t \\ Var(y^{(2)}_t | Y^{(1)}_t) &= R_t + H_t P_{t|T} H_t^{'} \end{aligned}$$

7 Parameter Estimation

Up to this point, we have assumed that system matrices of a BSTS model is known. In practice, such conditions are rarely met. In this section, I will present two methods: numerical maximization and EM algorithm. Direct numerical maximization has the advantage of faster convergence close to true values, and it does not requires Kalman smoothers, which requires vast computational costs. On the other hand, the EM algorithm converges faster at the beginning, and has a cleaner derivation. It is important to note, however, that the two approaches are consistent, but not unbiased. Therefore, users should practice with caution when using likelihood based models for parameter estimation. In what follows, I provide more details about the two algorithms in the case of diffuse priors.

7.1 Initialization with Lyapunov Equation and Directed Graphs

If we are dealing with a BSTS model with constant system matrices, and $P_{1|0}$ or $\hat{\xi}_{1|0}$ is not provided, it is possible to compute them[19]. Effectively we want to find the ergodic mean and variance (or their diffuse counterparts). First consider $P_{1|0}$. To find ergodic variance is equivalent to solving a Lyapunov stability equation:

$$FP_{1|0}F^{'} + Q = P_{1|0} \tag{7.1}$$

where F is F_t without the time subscript t. I use `scipy.linalg.solve_discrete_lyapunov` function to find $P_{1|0}$. If there is at least one unit root for F, the resulting $P_{1|0}$ will have very large values along the diagonal axis at indices that we may treat as diffuse states. Q is the assumed to be common error covariance matrix for state transition errors up to$t = 0$[20]. It is important to keep Q a positive semi-definite[21], or we have pathological systems.

In addition, we may run into BSTS models with explosive roots (i.e. eigenvalues of F are greater than one), equation (7.1) still generate a finite (but unstable) $P_{1|0}$ and fails to detect the diffuse state. In such cases, I use techniques in graph theory and first partition F into several strongly connected components[22]. I then calculate the largest eigenvalue in each component, and label these states as diffuse by setting a very large value on the corresponding diagonal values of F. Now the explosive states are properly marked, so solving a Lyapunov Equation will produce the correction partition between the diffuse component and the ergodic component of $P_{1|0}$. Note, however, that the assumption of marginal likelihood correction may be violated[23]. It is advised that users provide sensible estimates for the initial state.

The solution for $\hat{\xi}_{1|0}$ is easier to obtain. Consider the following:

$$\begin{aligned}\hat{\xi}_{1|0} &= F\hat{\xi}_{1|0} + Bx \\ &= (I - F)^{-1}Bx\end{aligned}$$

where B and x are B_t and x_t without t, respectively. In practice, it is easier to set to 0 the rows of B, and F corresponding to the diffuse states. Alternatively, users may provide precomputed or customized $P_{1|0}$ and $\hat{\xi}_{1|0}$. It is also very important to conduct sanity checks on system matrices to avoid pathological systems (e.g. $\xi_t = \xi_t + v_t$).

7.2 Numerical Maximization

This section follows closely (Durbin and S.J. Koopman 2012) with marginal likelihood corrections from (Marc K Francke, Siem Jan Koopman, and De Vos 2010) and (Harville 1974). First we parameterize system matrices. To ensure PSD of Q_t and R_t while allowing unconstrained parameterization, I use Cholesky decomposition described in (Pinheiro and Bates 1996). Next, I provide the expression for marginal likelihood[24] $l_m(Y_T)$ as:

$$l_m(Y_T) = Const - \frac{1}{2}\sum_{t=1}^{t_q-1}\sum_{i=1}^{n_t}\Psi_{t:(i)} - \frac{1}{2}\sum_{t=t_q}^{T}\sum_{i=1}^{n_t}\left(log|\Upsilon_{t:(i)}| + \frac{d^{'}_{t:(i)}d_{t:(i)}}{\Upsilon_{t:(i)}}\right) + \frac{1}{2}log\left|\sum_{t=1}^{T}(Z_t^{'}Z_t)\right| \tag{7.2}$$

$$\Psi_{t:(i)} = \begin{cases} log|\Upsilon_{\infty,t:(i)}| & \Upsilon_{\infty,t:(i)} > 0 \\ log|\Upsilon_{*,t:(i)}| + \frac{d^{'}_{t:(i)}d_{t:(i)}}{\Upsilon_{*,t:(i)}} & \Upsilon_{\infty,t:(i)} = 0 \end{cases}$$

[19] For BSTS models with time varying system matrices, it is advised to supply user-defined $\hat{\xi}_{1|0}$ and $P_{1|0}$.

[20] If Q_t or any system matrix related to state transitioning is not constant or unknown, we should use diffuse initialization for these states.

[21] There are several techniques to ensure both uniqueness and unboundedness in parametrizing Q. `linkalman` uses the Cholesky decomposition technique described in (Pinheiro and Bates 1996), which also outlines several alternative approaches.

[22] I use the `networkx` package to find such components.

[23] See (M. Francke, Koopman, A. d. Vos, et al. 2010) for details.

[24] Interested readers may refer to Appendix C.1.

The last term in l_m is the marginal likelihood correction term, expressed as:

$$\begin{aligned} Z_t &= H_t \phi_t \\ \phi_{t+1} &= F_t \phi_t \\ \phi_1 &= A \end{aligned} \tag{7.3}$$

Note that equation (7.2) only requires Kalman filters, so it avoids costly Kalman smoothers and is faster to evaluate. In the univariate case, H_t is $\tilde{H}_t$, and if the i-th measurement is missing, the i-th row of H_t is 0. In practice, we can use popular numerical optimization packages such as `scipy` and `nlopt` to solve it numerically.

7.3 EM Algorithm

One disadvantage of numerical maximization is the speed of the convergence is slow at the beginning. In addition, we need to run the Kalman filter for every parameter evaluation. As an alternative, EM algorithm proposed by (R. H. Shumway and D. S. Stoffer 1982) has a better early-stage performance[25]. The EM algorithm converges to a local optimum iteratively, and in particular, within each iteration, the Kalman smoother is evaluated only once. Its rate of convergence is substantially slower than numerical optimization in the neighborhood of the true parameters. One may use EM algorithm to cold start the optimization, then switch to numerical optimization to fine-tune the estimates. Note that equations used by EM algorithms are closely related to score functions, which can be used by gradient-based algorithms for fast optimization. In what follows, I will provide details of implementing the EM algorithm[26].

The log likelihood function for a BSTS model is:

$$L(Y_T|X_T;\theta) = log[\mathbb{P}(Y_T|X_T;\theta)]$$

Following equation (C.2), we maximize marginal $L(Y_T|X_T;\theta)$ by maximizing:

$$G(Y_T, X_T, \theta) = \int log[\mathbb{P}(Y_T, \Xi_T|X_T, \theta)]\mathbb{P}(\Xi_T|Y_T, X_T, \theta)d\Xi_T + \frac{1}{2}log\left|\sum_{t=1}^{T}(Z_t^{'}Z_t)\right|$$

Denote $G(\theta, \theta_i)$ as:

$$G(\theta, \theta_i) \equiv \int log[\mathbb{P}(Y_T, \Xi_T|X_T, \theta)]\mathbb{P}(\Xi_T|Y_T, X_T, \theta_i)d\Xi_T + \frac{1}{2}log\left|\sum_{t=1}^{T}(Z_t^{'}Z_t)\right|$$

Note that here Z_t is not the same as $\tilde{Z}_t$ in Section 7.2. As explained in Appendix C.3, we need pre-orthogonalized y_t for calculating $G(\theta, \theta_i)$, and as a result H_t is pre-orthogonalized as well.

EM algorithm proceeds as follows:

1. Start with initial parameter value θ_0
2. For iteration i, compute the conditional distribution of $(\Xi_T|Y_T, X_T, \theta_{i-1})$
3. Find θ_i that maximizes $G(\theta, \theta_{i-1})$ as specified in Appendix C.3
4. Repeat step 2 and 3 until $\{G(Y_T, X_T, \theta_i)\}_i$ converges to a local optimal[27]

[25] Given that the intended purpose of `linkalman` is solving BSTS models with sophisticated model structures, I use non-gradient-based methods. As a result, the primary advantage of EM algorithm (i.e. ease to compute derivatives) is not utilized, and the performance of EM algorithm is inferior to numerical methods.

[26] For technical details, refer to Appendix C.2.

[27] In practice, if one is using numerical optimization within each iteration, it is important to fine-tune the optimizer's tolerance parameters so that function evaluation is always improving with each iteration.

References

Brodersen, K. H. and F. Gallusser et al. (2015). "Inferring Causal Impact Using Bayesian Structural Time-series Models". In: *The Annals of Applied Statistics* 9.1, pp. 247–274.

Bucy, R.S. and P.D. Joseph (1968). *Filtering for Stochastic Processes with Application to Guidance*. Interscience.

De Jong, Piet (1989). "Smoothing and interpolation with the state-space model". In: *Journal of the American Statistical Association* 84.408, pp. 1085–1088.

— (1991). "The diffuse Kalman filter". In: *The Annals of Statistics* 19.2, pp. 1073–1083.

De Jong, Piet and Murray J Mackinnon (1988). "Covariances for smoothed estimates in state space models". In: *Biometrika* 75.3, pp. 601–602.

Doan, Thomas A (2010). "Practical issues with state-space models with mixed stationary and non-stationary dynamics". In: *Technical paper* 2010-1.

Durbin, J. and S.J. Koopman (2000). "Fast Filtering and Smoothing for Multivariate State Space Models". In: *Journal of Time Series Analysis* 21.3, pp. 281–296.

— (2003). "Filtering and Smoothing of State Vector for Diffuse State-space Models". In: *Journal of Time Series Analysis* 24.1, pp. 85–98.

— (2012). *Time Series Analysis by State Space Methods*. Oxford University Press.

Francke, Marc K, Siem Jan Koopman, and Aart F De Vos (2010). "Likelihood functions for state space models with diffuse initial conditions". In: *Journal of Time Series Analysis* 31.6, pp. 407–414.

Francke, Marc K and Aart F de Vos (2007). "Marginal likelihood and unit roots". In: *Journal of Econometrics* 137.2, pp. 708–728.

Francke, MK, SJ Koopman, AF de Vos, et al. (2010). "Likelihood functions for state space models with diffuse initial conditions". In: *Journal of Time Series Analysis* 31.

Hamilton, James D. (1994). *Time Series analysis*. Princeton University Press.

Harvey, A. C. (1985). "Trends and Cycles in Macroeconomic Time Series". In: *Journal of Business and Economic Statistics* 3, pp. 216–227.

Harvey, Andrew C, Thomas M Trimbur, and Herman K Van Dijk (2007). "Trends and cycles in economic time series: A Bayesian approach". In: *Journal of Econometrics* 140.2, pp. 618–649.

Harville, David A (1974). "Bayesian inference for variance components using only error contrasts". In: *Biometrika* 61.2, pp. 383–385.

Helske, Jouni (2016). "KFAS: Exponential family state space models in R". In: *arXiv:1612.01907*. URL: http://CRAN.R-project.org/package=KFAS.

Higham, Nicholas J. (1988). "Computing a Nearest Symmetric Positive Semidefinite Matrix". In: *Linear Algebra and Its Applications* 103, pp. 103–118.

Koopman, Siem Jan (1997). "Exact Initial Kalman Filtering and Smoothing for Nonstationary Time Series Models". In: *Journal of the American Statistical Association* 92, pp. 1630–1638.

Koopman, Siem Jan and Neil Shephard (1992a). "Exact score for time series models in state space form". In: *BIOMETRIKA-CAMBRIDGE-* 79, pp. 823–823.

— (1992b). "Exact score for time series models in state space form". In: *BIOMETRIKA-CAMBRIDGE-* 79, pp. 823–823.

Petersen, Kaare Brandt, Michael Syskind Pedersen, et al. (2008). "The matrix cookbook". In: *Technical University of Denmark* 7.15, p. 510.

Pinheiro, José C and Douglas M Bates (1996). "Unconstrained parametrizations for variance-covariance matrices". In: *Statistics and computing* 6.3, pp. 289–296.

Shumway, R. H. (2000). "Dynamic mixed models for irregularly observed time series". In: *Resenhas-Reviews of the Institute of Mathematics and Statistics* 4, pp. 433–456.

Shumway, R. H. and D. S. Stoffer (1982). "An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm". In: *Journal of Time Series Analysis* 3.4, pp. 253–264.

Shumway, Robert H and David S Stoffer (2017). *Time series analysis and its applications: with R examples*. Springer.

A Kalman Filter

A.1 Derivation of $\hat{\xi}_{t|t}$ and $P_{t|t}$

Lemma 1 (Law of Iterated Projections) *Let $\mathcal{P}(Y_3|Y_2, Y_1)$ be the projection of Y_3 on (Y_2, Y_1). Denote Ω_{ij} as $\Omega_{ij} = E(Y_i Y_j^{'})$, then we have projections:*

$$\mathcal{P}(Y_3|Y_2, Y_1) = \mathcal{P}(Y_3|Y_1) + H_{32}H_{22}^{-1}[Y_2 - \mathcal{P}(Y_2|Y_1)]$$

and variance matrix:

$$Var(Y_3|Y_2, Y_1) = H_{33} - H_{32}H_{22}^{-1}H_{32}^{'}$$

where

$$\begin{aligned}
H_{22} &= E\{[Y_2 - \mathcal{P}(Y_2|Y_1)][Y_2 - \mathcal{P}(Y_2|Y_1)]^{'}\} \\
H_{23} &= H_{32}^{'} = E\{[Y_2 - \mathcal{P}(Y_2|Y_1)][Y_3 - \mathcal{P}(Y_3|Y_1)]^{'}\} \\
H_{33} &= E\{[Y_3 - \mathcal{P}(Y_3|Y_1)][Y_3 - \mathcal{P}(Y_3|Y_1)]^{'}\}
\end{aligned}$$

Let ξ_t as Y_3, y_t as Y_2, and (x_t, Y_{t-1}) as Y_1, we obtain formula for $\hat{\xi}_{t|t}$ and $P_{t|t}$.

A.2 Derivation of Joseph Form of $P_{t|t}$

$$\begin{aligned}
P_{t|t} &= [I - K_t H_t]P_{t|t-1}[I - K_t H_t]' + K_t R_t K_t^{'} \\
&= P_{t|t-1} - K_t H_t P_{t|t-1} - P_{t|t-1}H_t^{'}K_t^{'} + K_t H_t P_{t|t-1}H_t^{'}K_t^{'} + K_t R_t K_t^{'} \\
&= P_{t|t-1} - K_t H_t P_{t|t-1} - P_{t|t-1}H_t^{'}K_t^{'} + K_t(H_t P_{t|t-1}H_t^{'} + R_t)K_t^{'} \\
&= P_{t|t-1} - K_t H_t P_{t|t-1} - P_{t|t-1}H_t^{'}K_t^{'} + P_{t|t-1}H_t^{'}K_t^{'} \\
&= P_{t|t-1} - K_t H_t P_{t|t-1}
\end{aligned}$$

The fourth equality follows from equation (5.2).

A.3 Proof of Diffuse Kalman Filter

This section derives from Section 5 of (Durbin and S.J. Koopman 2012), but provides an alternative formulation using the Joseph form. For the original proof without the Joseph formulation, please refer to (Durbin and S.J. Koopman 2012). The key intuition is to give an exact result for the initial state distribution with arbitrarily large but finite variances, then find out the limit as the variances approach infinity. Given definition of $P_{t|t-1}$ in equation (5.12), and linearity of Kalman filtering, we have:

$$\begin{aligned}
\Upsilon_t &= \kappa\Upsilon_{\infty,t} + \Upsilon_{*,t} + \mathcal{O}(\kappa^{-1}) \\
\Upsilon_{\infty,t} &= H_t P_{\infty,t} H_t^{'} \\
\Upsilon_{*,t} &= H_t P_{*,t} H_t^{'} + R_t
\end{aligned}$$

We are interested in finding Υ_t^{-1}. Expanding Υ_t^{-1} as a power series in κ^{-1}, we have:

$$\Upsilon_t^{-1} = \Upsilon_t^{(0)} + \kappa^{-1}\Upsilon_t^{(1)} + \kappa^{-2}\Upsilon_t^{(2)} + \mathcal{O}(\kappa^{-3})$$

where $\Upsilon_t^{(i)}$ is the i-th term associated with the power series expansion. We collapse other terms of the expansion to $\mathcal{O}(\kappa^{-3})$ as they will converge to 0 as $\kappa \to \infty$. We find $\Upsilon_t^{(i)}$ by using the fact that:

$$\begin{aligned}
I =& \Upsilon_t \Upsilon_t^{-1} \\
=& (\kappa\Upsilon_{\infty,t} + \Upsilon_{*,t} + \kappa^{-1}\Upsilon_{a,t} + \kappa^{-2}\Upsilon_{b,t}) \\
&\times (\Upsilon_t^{(0)} + \kappa^{-1}\Upsilon_t^{(1)} + \kappa^{-2}\Upsilon_t^{(2)} + \mathcal{O}(\kappa^{-3}))
\end{aligned} \tag{A.1}$$

Solving equation (A.1), we have:

$$\Upsilon_{\infty,t}\Upsilon_t^{(0)} = 0 \tag{A.2}$$

$$\Upsilon_{*,t}\Upsilon_t^{(0)} + \Upsilon_{\infty,t}\Upsilon_t^{(1)} = I \tag{A.3}$$

$$\Upsilon_{a,t}\Upsilon_t^{(0)} + \Upsilon_{*,t}\Upsilon_t^{(1)} + \Upsilon_{\infty,t}\Upsilon_t^{(2)} = 0 \tag{A.4}$$

For a full treatment when Υ_t is not 1-by-1 matrix, see (Siem Jan Koopman 1997). If we are combine initialization with sequential filtering/smoothing, the solution to equation (A.2) to (A.4) is much simpler[28] and is shown below.

If $\Upsilon_{\infty,t} \neq 0$, we have:

$$\begin{aligned}
\Upsilon_t^{(0)} &= 0 \\
\Upsilon_t^{(1)} &= \Upsilon_{\infty,t}^{-1} \\
\Upsilon_t^{(2)} &= -\Upsilon_{\infty,t}^{-1}\Upsilon_{*,t}\Upsilon_{\infty,t}^{-1}
\end{aligned}$$

By the fact that $K_t = P_{t|t-1}H_t^{'}\Upsilon_t^{-1}$, we can write K_t as:

$$K_t = K_t^{(0)} + \kappa^{-1}K_t^{(1)} + \mathcal{O}(\kappa^{-2}) \tag{A.5}$$

$$K_t^{(0)} = P_{\infty,t}H_t^{'}\Upsilon_t^{(1)} \tag{A.6}$$

$$\begin{aligned}
K_t^{(1)} &= P_{*,t}H_t^{'}\Upsilon_t^{(1)} + P_{\infty,t}H_t^{'}\Upsilon_t^{(2)} \\
&= (P_{*,t}H_t^{'} - K_t^{(0)}\Upsilon_{*,t})\Upsilon_{\infty,t}^{-1}
\end{aligned} \tag{A.7}$$

Similarly, we can rewrite $\hat{\xi}_{t|t}$ as:

$$\hat{\xi}_{t|t} = \hat{\xi}_{t|t-1} + K_t^{(0)}d_t + \mathcal{O}(\kappa^{-1}) \tag{A.8}$$

For $P_{t|t}$, we have:

$$\begin{aligned}
P_{t|t} =& [I - K_tH_t]P_{t|t-1}[I - K_tH_t]^{'} + K_tR_tK_t^{'} \\
=& [I - (K_t^{(0)} + \kappa^{-1}K_t^{(1)})H_t](\kappa P_{\infty,t} + P_{*,t})[I - (K_t^{(0)} + \kappa^{-1}K_t^{(1)})H_t]^{'} \\
&+ K_t^{(0)}R_tK_t^{(0)'} + \mathcal{O}(\kappa^{-1}) \\
=& \kappa(I - K_t^{(0)}H_t)P_{\infty,t}(I - K_t^{(0)}H_t)^{'} + (I - K_t^{(0)}H_t)P_{*,t}(I - K_t^{(0)}H_t)^{'} \\
&- K_t^{(1)}H_tP_{\infty,t}(I - K_t^{(0)}H_t)^{'} - (I - K_t^{(0)}H_t)P_{\infty,t}H_t^{'}K_t^{(1)'} \\
&+ K_t^{(0)}R_tK_t^{(0)'} + \mathcal{O}(\kappa^{-1}) \\
=& \kappa(I - K_t^{(0)}H_t)P_{\infty,t}(I - K_t^{(0)}H_t)^{'} + (I - K_t^{(0)}H_t)P_{*,t}(I - K_t^{(0)}H_t)^{'} \\
&- K_t^{(1)}H_tP_{\infty,t} + K_t^{(1)}H_tP_{\infty,t}H_t^{'}K_t^{(0)'} \\
&- P_{\infty,t}H_t^{'}K_t^{(1)'} + K_t^{(0)}H_tP_{\infty,t}H_t^{'}K_t^{(1)'} \\
&+ K_t^{(0)}R_tK_t^{(0)'} + \mathcal{O}(\kappa^{-1})
\end{aligned}$$

Note that $K_t^{(0)} = P_{\infty,t}H_t^{'}\Upsilon_t^{(1)} = P_{\infty,t}H_t^{'}(H_tP_{\infty,t}H_t^{'})^{-1}$, we have:

$$\begin{aligned}
K_t^{(0)}H_tP_{\infty,t}H_t^{'} &= P_{\infty,t}H_t^{'}(H_tP_{\infty,t}H_t^{'})^{-1}H_tP_{\infty,t}H_t^{'} \\
&= P_{\infty,t}H_t^{'}
\end{aligned}$$

Now we have a clean Joseph form for the diffuse filter as:

$$\begin{aligned}
P_{t|t} =& \kappa(I - K_t^{(0)}H_t)P_{\infty,t}(I - K_t^{(0)}H_t)^{'} \\
&+ (I - K_t^{(0)}H_t)P_{*,t}(I - K_t^{(0)}H_t)^{'} \\
&+ K_t^{(0)}R_tK_t^{(0)'} + \mathcal{O}(\kappa^{-1})
\end{aligned} \tag{A.9}$$

[28] In the univariate case, non-singularity is equivalent to being non-zero, and singularity is equivalent to being zero.

If $\Upsilon_{\infty,t} = 0$, then $H_t P_{\infty,t} = 0$ by PSD properties. It is easier to get $\hat{\xi}_{t|t}$ and $P_{t|t}$:

$$\begin{aligned} K_t =& P_{*,t} H_t^{'} \Upsilon_{*,t}^{-1} + \mathcal{O}(\kappa^{-1}) \\ =& K_t^{(*)} + \mathcal{O}(\kappa^{-1}) \end{aligned} \tag{A.10}$$

$$\hat{\xi}_{t|t} = \hat{\xi}_{t|t-1} + K_t^{(*)} d_t + \mathcal{O}(\kappa^{-1}) \tag{A.11}$$

For $P_{t|t}$, we have:

$$\begin{aligned} P_{t|t} =& \kappa (I - K_t^{(*)} H_t) P_{\infty,t} (I - K_t^{(*)} H_t)^{'} \\ &+ (I - K_t^{(*)} H_t) P_{*,t} (I - K_t^{(*)} H_t)^{'} \\ &+ K_t^{(*)} R_t K_t^{(*)'} + \mathcal{O}(\kappa^{-1}) \\ =& \kappa P_{\infty,t} + (I - K_t^{(*)} H_t) P_{*,t} (I - K_t^{(*)} H_t)^{'} \\ &+ K_t^{(*)} R_t K_t^{(*)'} + \mathcal{O}(\kappa^{-1}) \end{aligned} \tag{A.12}$$

A.4 Proof of the Degeneration Algorithm

Define $P_{\infty,t|t}$ as the diffuse part of $P_{t|t}$. From Appendix A.3, we have:

$$P_{\infty,t|t} = P_{\infty,t}[I - H_t^{'}(H_t P_{\infty,t} H_t^{'})^{-1} H_t P_{\infty,t}]$$

Without loss of generality, we can re-arrange indices of ξ_t such that:

$$P_{\infty,1} = \begin{pmatrix} I_q & 0 \\ 0 & 0 \end{pmatrix}$$

In the univariate case, H_t is a row vector, and $H_t P_{\infty,t} H_t^{'}$ is a similarly partitioned matrix as $P_{\infty,1}$. Therefore, any change to the matrix only affects the non-zero partition, and we can focus solely on this partition. In what follows I will provide a proof in the case where $P_{\infty,1}$ has full rank (i.e. the initial condition is completely uninformative), and the conclusion is readily adapted to the mixed diffuse case where $P_{\infty,1}$ does not have full rank through partitioning operations and LDL decomposition of $P_{\infty,t}$.

If $P_{\infty,t} = 0$, $P_{\infty,t|t} = P_{\infty,t}$ and is not updated. If $P_{\infty,t} \neq 0$, $H_t^{'}(H_t P_{\infty,t} H_t^{'})^{-1} H_t P_{\infty,t}$ is an idempotent matrix. If a matrix A is idempotent, then:

1 $Trace(A) = rank(A)$

2 $I - A$ is also idempotent

Using the fact that $Trace(AB) = Trace(BA)$, we have:

$$Trace[H_t^{'}(H_t P_{\infty,t} H_t^{'})^{-1} H_t P_{\infty,t}] = Trace[H_t P_{\infty,t} H_t^{'}(H_t P_{\infty,t} H_t^{'})^{-1}] = 1$$

Using the fact that $Trace(A + B) = Trace(A) + Trace(B)$, we have:

$$Trace[I - H_t^{'}(H_t P_{\infty,t} H_t^{'})^{-1} H_t P_{\infty,t}] = q - 1$$

If A has full rank, then by the fact that $Trace(AB) = Trace(B)$, we have:

$$Trace(P_{\infty,t|t}) = q - 1$$

Therefore, if $\Upsilon_{\infty,t} \neq 0$, the rank of $P_{\infty,t}$ is reduced by 1 with each update. If after period t, $rank(P_{\infty,t|t}) > 0$, we have $P_{\infty,t+1} = F_t P_{\infty,t|t} F_t^{'}$. By the fact that $rank(AB) \leq min[rank(A), rank(B)]$, we have $\hat{q}_{t+1} = min[\hat{q}_t - 1, q_{t+1}]$. I use this updating rule instead of $\hat{q}_{t+1} = \hat{q}_t - 1$ to avoid numerical instability.

B Kalman Smoother

B.1 Proof of State Smoothing

The backbone of the proof is Lemma 1. Note that $\{Y_T, X_T\}$ is equivalent to $\{Y_{t-1}, d_t, ..., d_T, X_T\}$. In addition, $d_i \perp Y_{t-1}$ from the fact that $\hat{\xi}_{t|t-1}$ is a linear projection of Y_{t-1} with Gaussian distribution. Applying this result iteratively, we can also obtain $d_i \perp d_j \; \forall \{i, j\} \in \{t, ..., T\}$.

Now consider $\hat{\xi}_{t|T}$, denote $\Delta_t \equiv \{d_t^{'}, d_{t+1}^{'}, ..., d_T^{'}\}^{'}$

$$\begin{aligned}\hat{\xi}_{t|T} &= E(\xi_t | Y_{t-1}, \Delta_t, X_T; \theta) \\ &= \hat{\xi}_{t|t-1} + Cov(\xi_t, \Delta_t | Y_{t-1}, X_T; \theta) Var(\Delta_t | Y_{t-1}, X_T; \theta)^{-1} \Delta_t\end{aligned}$$

It is straightforward to show $Var(d_j | Y_{t-1}, X_T; \theta) = \Upsilon_j$. For $Cov(\xi_t, d_j | Y_{t-1}, X_T; \theta)$, we have:

$$\begin{aligned}Cov(\xi_t, d_j | Y_{t-1}, X_T; \theta) &= E(\xi_t d_j^{'} | Y_{t-1}, X_T; \theta) \\ &= E[\xi_t (\xi_j - \hat{\xi}_{j|j-1})^{'} | Y_{t-1}, X_T; \theta] H_j^{'}\end{aligned}$$

If $j = t$, then:

$$E[\xi_t (\xi_t - \hat{\xi}_{t|t-1})^{'} | Y_{t-1}, X_T; \theta] = P_{t|t-1}$$

If $j = t + 1$, then:

$$\begin{aligned}E[\xi_t (\xi_{t+1} - \hat{\xi}_{t+1,t})^{'} | Y_{t-1}, X_T; \theta] &= E[\xi_t (\xi_t - \hat{\xi}_{t|t-1} - K_t d_t)^{'} F_t^{'} | Y_{t-1}, X_T; \theta] \\ &= P_{t|t-1} L_t^{'}\end{aligned}$$

If we have no measurements for time $t + 1$, there is no Kalman updating, and we have:

$$E[\xi_t (\xi_{t+1} - \hat{\xi}_{t+1,t})^{'}] = P_{t|t-1} F_t^{'}$$

For $j > t + 1$, we have:

$$E[\xi_t (\xi_j - \hat{\xi}_{j,j-1})^{'} | Y_{t-1}, X_T; \theta] = P_{t|t-1} \prod_{i=t}^{j-1} L_i^{'}$$

For simplicity, I only use L_i here. If we have no measurements for time i, we replace L_i with F_i.

To compute $\hat{\xi}_{t|T}$, note that $Var(\Delta_t | Y_{t-1}, X_T; \theta)$ is a block diagonal matrix, then we may express $\hat{\xi}_{t|T}$ as:

$$\begin{aligned}\hat{\xi}_{t|T} &= \hat{\xi}_{t|t-1} + \sum_{j=t}^{T} Cov(\xi_t, d_j) \Upsilon_j^{-1} d_j \\ &= \hat{\xi}_{t|t-1} + P_{t|t-1} \sum_{j=t}^{T} \left[\left(\prod_{i=t}^{j-1} L_i^{'} \right) H_j^{'} \Upsilon_j^{-1} d_j \right]\end{aligned}$$

If $j = t$, then we replace the product operation is not carried out and is replace with I. Define r_t as:

$$r_{t-1} \equiv \begin{cases} 0, & t = T + 1 \\ \sum_{j=t}^{T} \left(\prod_{i=t}^{j-1} L_i^{'} \right) H_j^{'} \Upsilon_j^{-1} d_j, & t \leq T \end{cases}$$

We can then write backwards recursive formulation for $\hat{\xi}_{t|T}$ as:

$$\begin{aligned}r_{t-1} &= H_t^{'} \Upsilon_t^{-1} d_t + L_t^{'} r_t \\ \hat{\xi}_{t|T} &= \hat{\xi}_{t|t-1} + P_{t|t-1} r_{t-1}\end{aligned}$$

To calculate $P_{t|T}$, we use Lemma 1 again and have the following result:

$$\begin{aligned} P_{t|T} &= P_{t|t-1} - \sum_{j=t}^{T} Cov(\xi_t, d_j)\Upsilon_j^{-1} Cov(\xi_t, d_j)^{'} \\ &= P_{t|t-1} - P_{t|t-1}\sum_{j=t}^{T}\left[\left(\prod_{i=t}^{j-1} L_i^{'}\right) H_j^{'}\Upsilon_j^{-1} H_j \left(\prod_{i=t}^{j-1} L_i^{'}\right)^{'}\right] P_{t|t-1} \end{aligned}$$

Similarly, we can find $P_{t|T}$ through backwards recursions. Define N_t as:

$$N_{t-1} \equiv \begin{cases} 0, & t = T+1 \\ \sum_{j=t}^{T}\left[\left(\prod_{i=t}^{j-1} L_i^{'}\right) H_j^{'}\Upsilon_j^{-1} H_j \left(\prod_{i=t}^{j-1} L_i^{'}\right)^{'}\right], & t \leq T \end{cases}$$

The recursive formulation for $P_{t|T}$ is:

$$\begin{aligned} N_{t-1} &= H_t^{'}\Upsilon_t^{-1} H_t + L_t^{'} N_t L_t \\ P_{t|T} &= P_{t|t-1} - P_{t|t-1} N_{t-1} P_{t|t-1} \end{aligned}$$

B.2 Proof of Covariance between Smoothed States

First of all, to simplify notations, I define:

$$E[f(t,j)] \equiv E[f(t,j)|Y_{t-1}, X_T; \theta] \;\forall\; 1 \leq t < j \leq T$$

where $f(t,j)$ is a mapping indexed by (t,j). To find the representation of $P_{t,j|T}$, note that:

$$\begin{aligned} P_{t,j|T} &= E[\xi_t(\xi_j - \hat{\xi}_{j|T})^{'}] - E[\hat{\xi}_{t|T}(\xi_j - \hat{\xi}_{j|T})^{'}] \\ &= E[\xi_t(\xi_j - \hat{\xi}_{j|T})^{'}] - \hat{\xi}_{t|T} E(\xi_j - \hat{\xi}_{j|T})^{'} \\ &= E[\xi_t(\xi_j - \hat{\xi}_{j|T})^{'}] \end{aligned}$$

Recall that $\xi_t - \hat{\xi}_{t|T} = \xi_t - \hat{\xi}_{t|t-1} - P_{t|t-1} r_{t-1}$, then:

$$E[\xi_t(\xi_j - \hat{\xi}_{j,T})^{'}] = E[\xi_t(\xi_j - \hat{\xi}_{j|j-1})^{'}] - E(\xi_t r_{j-1}^{'}) P_{j|j-1}$$

From Appendix B.1, we have:

$$E[\xi_t(\xi_j - \hat{\xi}_{j,j-1})^{'}] = P_{t|t-1}\prod_{i=t}^{j-1} L_i^{'}$$

To compute $E(\xi_t r_{j-1}^{'})$, consider the following:

$$\begin{aligned} E(\xi_t d_j^{'}) &= E[\xi_t(\xi_j - \hat{\xi}_{j|T})^{'}] H_j^{'} \\ &= P_{t|t-1}\left(\prod_{i=t}^{j-1} L_i^{'}\right) H_j^{'} \end{aligned}$$

Then using definition of r_t, we have:

$$\begin{aligned}
E(\xi_t r_{j-1}^{'}) &= \sum_{i=j}^{T} \left[E(\xi_t d_i^{'}) \Upsilon_i^{-1} H_i \left(\prod_{k=j+1}^{i} L_{k-1}^{'} \right)^{'} \right] \\
&= P_{t|t-1} \left(\prod_{i=t}^{j-1} L_i^{'} \right) \sum_{i=j}^{T} \left[\left(\prod_{k=j+1}^{i} L_{k-1}^{'} \right) H_i^{'} \Upsilon_i^{-1} H_i \left(\prod_{k=j+1}^{i} L_{k-1}^{'} \right)^{'} \right] \\
&= P_{t|t-1} \left(\prod_{i=t}^{j-1} L_i^{'} \right) N_{j-1}
\end{aligned}$$

Now using the expressions for $E[\xi_t(\xi_j - \hat{\xi}_{j,j-1})^{'}]$ and $E(\xi_t r_{j-1}^{'})$, we have:

$$P_{t,j|T} = P_{t|t-1} \left(\prod_{i=t}^{j-1} L_i^{'} \right) (I - N_{j-1} P_{j|j-1})$$

In particular, for $j = t+1$, we have:

$$P_{t,t+1|T} = P_{t|t-1} L_t^{'} (I - N_t P_{t+1|t})$$

B.3 Proof of Diffuse Smoothing

B.3.1 General Expressions for State Smoothing:

We use similar techniques to derive diffuse smoothers as we do diffuse filters. I follow closely (Durbin and S.J. Koopman 2003) and (Siem Jan Koopman 1997) in deriving diffuse smoothers, and fill the gaps in proofs omitted by the original paper.

Following the practice in deriving diffuse Kalman filters, we can write r_{t-1} and N_{t-1} as:

$$\begin{aligned}
r_{t-1} &= r_{t-1}^{(0)} + \kappa^{-1} r_{t-1}^{(1)} + \mathcal{O}(\kappa^{-2}) \\
N_{t-1} &= N_{t-1}^{(0)} + \kappa^{-1} N_{t-1}^{(1)} + \kappa^{-2} N_{t-1}^{(2)} + \mathcal{O}(\kappa^{-3})
\end{aligned}$$

To compute $\hat{\xi}_{t|T}$ for $t < t_q$, note that $Var(\xi_t | Y_{t_q}, X_T; \theta) < \infty$ by definition of t_q. If we conditioned on Y_t, we have $Var(\xi_t | Y_T, X_T; \theta) < \infty$, which means $\hat{\xi}_{t|T} < \infty$. We have:

$$\begin{aligned}
\hat{\xi}_{t|T} &= \hat{\xi}_{t|t-1} + P_{t|t-1} r_{t-1} \\
&= \hat{\xi}_{t|t-1} + \kappa P_{\infty,t} r_{t-1}^{(0)} + P_{*,t} r_{t-1}^{(0)} + P_{\infty,t} r_{t-1}^{(1)} + \mathcal{O}(\kappa^{-1}) \\
&= \hat{\xi}_{t|t-1} + P_{*,t} r_{t-1}^{(0)} + P_{\infty,t} r_{t-1}^{(1)} + \mathcal{O}(\kappa^{-1})
\end{aligned}$$

The last equality holds because we have established finiteness of $\hat{\xi}_{t|T}$.

To compute $P_{t|T}$, I first denote $\langle A \rangle$ as $A + A^{'}$. Then for $P_{t|T}$, we have:

$$\begin{aligned}
P_{t|T} =& \kappa P_{\infty,t} + P_{*,t} - (\kappa P_{\infty,t} + P_{*,t})(N_{t-1}^{(0)} + \kappa^{-1} N_{t-1}^{(1)} + \kappa^{-2} N_{t-1}^{(2)})(\kappa P_{\infty,t} + P_{*,t}) \\
=& -\kappa^2 P_{\infty,t} N_{t-1}^{(0)} P_{\infty,t} \\
&+ \kappa (P_{\infty,t} - \langle P_{\infty,t} N_{t-1}^{(0)} P_{*,t} \rangle - P_{\infty,t} N_{t-1}^{(1)} P_{\infty,t}) \\
&+ P_{*,t} - \langle P_{\infty,t} N_{t-1}^{(1)} P_{*,t} \rangle - P_{*,t} N_{t-1}^{(0)} P_{*,t} - P_{\infty,t} N_{t-1}^{(2)} P_{\infty,t} + \mathcal{O}(\kappa^{-1}) \\
=& P_{*,t} - \langle P_{\infty,t} N_{t-1}^{(1)} P_{*,t} \rangle - P_{*,t} N_{t-1}^{(0)} P_{*,t} - P_{\infty,t} N_{t-1}^{(2)} P_{\infty,t} + \mathcal{O}(\kappa^{-1})
\end{aligned}$$

The last equality holds because $P_{t|T} < \infty$, and terms that associate with κ^2 and κ must be 0.

Finally, for $P_{t,t+1|T}$, we have:

$$P_{t,t+1|T} = P_{t|t-1}L_t^{'}(I - N_t P_{t+1|t})$$

Note that $(I - N_t P_{t+1|t})$ is $\mathcal{O}(\kappa^0)$ because $P_{\infty,t+1}N_t^{(0)} = 0$. Therefore, we only need to expand $P_{t|t-1}$ and L_t as:

$$\begin{aligned} P_{t|t-1} &= \kappa P_{\infty,t} + P_{*,t} + \mathcal{O}(\kappa^{-1}) \\ L_t &= L_t^{(0)} + \kappa^{-1}L_t^{(1)} + \mathcal{O}(\kappa^{-2}) \end{aligned}$$

By the same reasoning, we only need to expand $P_{t|t-1}L_t^{'}$ up to $\mathcal{O}(\kappa^0)$:

$$P_{t|t-1}L_t^{'} = \kappa P_{\infty,t}L_t^{(0)'} + P_{*,t}L_t^{(0)'} + P_{\infty,t}L_t^{(1)'} + \mathcal{O}(\kappa^{-1})$$

For $I - N_t P_{t+1|t}$, we have:

$$\begin{aligned} I - N_t P_{t+1|t} =& I - N_t^{(1)}P_{\infty,t+1} - N_t^{(0)}P_{*,t+1} \\ & - \kappa^{-1}(N_t^{(2)}P_{\infty,t+1} + N_t^{(1)}P_{*,t+1}) + \mathcal{O}(\kappa^{-2}) \end{aligned}$$

by the fact that $P_{t|T}P_{t+1|T} - P_{t,t+1|T}P_{t,t+1|T}^{'}$ is PSD and that $P_{t|T}$ and $P_{t+1|T}$ are both finite, we know that $P_{t,t+1|T}$ is finite as well. Note that we do not expand $P_{t+1|t}$ beyond $\mathcal{O}(\kappa^0)$ in $N_t P_{t+1|t}$, because by Lemma 2 in Appendix B.3.2, $P_{\infty,t}L_t^{(0)'}N_t^{(0)} = 0$. Now we can express $P_{t,t+1|T}$ as:

$$\begin{aligned} P_{t,t+1|T} =& (P_{*,t}L_t^{(0)'} + P_{\infty,t}L_t^{(1)'})(I - N_t^{(1)}P_{\infty,t+1} - N_t^{(0)}P_{*,t+1}) \\ & - P_{\infty,t}L_t^{(0)'}(N_t^{(2)}P_{\infty,t+1} + N_t^{(1)}P_{*,t+1}) + \mathcal{O}(\kappa^{-1}) \end{aligned}$$

If $\Upsilon_{\infty,t} = 0$, $P_{t,t+1|T}$ becomes:

$$\begin{aligned} P_{t,t+1|T} =& P_{*,t}L_t^{(0)'}(I - N_t^{(1)}P_{\infty,t+1} - N_t^{(0)}P_{*,t+1}) \\ & - P_{\infty,t}F_t^{'}(N_t^{(2)}P_{\infty,t+1} + N_t^{(1)}P_{*,t+1}) + \mathcal{O}(\kappa^{-1}) \end{aligned}$$

B.3.2 Recursive Formula for r_{t-1} and N_{t-1}

Having derived expressions for $\hat{\xi}_{t|T}$, $P_{t|T}$ and $P_{t,t+1|T}$, I proceed to derive the backward recursive formula for r_{t-1} and N_{t-1}. The derivation follows closely (Durbin and S.J. Koopman 2003) with some additional details.

First of all, consider r_{t-1}:

$$\begin{aligned} r_{t-1} =& r_{t-1}^{(0)} + \kappa^{-1}r_{t-1}^{(1)} + \mathcal{O}(\kappa^{-2}) \\ =& H_t^{'}(\Upsilon_t^{(0)} + \kappa^{-1}\Upsilon_t^{(1)})d_t \\ & + (L_t^{(0)} + \kappa^{-1}L_t^{(1)})^{'}(r_t^{(0)} + \kappa^{-1}r_t^{(1)}) + \mathcal{O}(\kappa^{-2}) \end{aligned}$$

After some algebra, we have:

$$\begin{aligned} r_{t-1}^{(0)} &= H_t^{'}\Upsilon_t^{(0)}d_t + L_t^{(0)'}r_t^{(0)} \\ r_{t-1}^{(1)} &= H_t^{'}\Upsilon_t^{(1)}d_t + L_t^{(0)'}r_t^{(1)} + L_t^{(1)'}r_t^{(0)} \\ L_t^{(0)} &= F_t(I - K_t^{(0)}H_t) \\ L_t^{(1)} &= -F_t K_t^{(1)}H_t \end{aligned}$$

Similarly, for N_t:

$$\begin{aligned}
N_{t-1} =& N_{t-1}^{(0)} + \kappa^{-1} N_{t-1}^{(1)} + \kappa^{-2} N_{t-1}^{(2)} + \mathcal{O}(\kappa^{-3}) \\
=& H_t^{'}(\Upsilon_t^{(0)} + \kappa^{-1}\Upsilon_t^{(1)} + \kappa^{-2}\Upsilon_t^{(2)}) H_t \\
&+ [I - (K_t^{(0)} + \kappa^{-1} K_t^{(1)} + \kappa^{-2} K_t^{(2)}) H_t]^{'} F_t^{'} \\
&\times (N_t^{(0)} + \kappa^{-1} N_t^{(1)} + \kappa^{-2} N_t^{(2)}) \\
&\times F_t [I - (K_t^{(0)} + \kappa^{-1} K_t^{(1)} + \kappa^{-2} K_t^{(2)}) H_t] + \mathcal{O}(\kappa^{-3})
\end{aligned}$$

After a lot of algebra, we have:

$$\begin{aligned}
N_{t-1}^{(0)} =& H_t^{'} \Upsilon_t^{(0)} H_t + L_t^{(0)'} N_t^{(0)} L_t^{(0)} \\
N_{t-1}^{(1)} =& H_t^{'} \Upsilon_t^{(1)} H_t + \langle L_t^{(1)'} N_t^{(0)} L_t^{(0)} \rangle + L_t^{(0)'} N_t^{(1)} L_t^{(0)} \\
N_{t-1}^{(2)} =& H_t^{'} \Upsilon_t^{(2)} H_t + \langle L_t^{(1)'} N_t^{(1)} L_t^{(0)} \rangle + L_t^{(0)'} N_t^{(2)} L_t^{(0)} \\
&- \langle H_t^{'} K_t^{(2)'} F_t^{'} N_t^{(0)} L_t^{(0)} \rangle + L_t^{(1)'} N_t^{(0)} L_t^{(1)}
\end{aligned}$$

Before I proceed, I will prove the following auxiliary result[29]:

Lemma 2 (Law of Equivalent Recursion) *Consider the following system:*

$$\begin{aligned}
c_{t-1} &= a_t + b_t + L_t^{(0)'} c_t \\
e_t &= f_t + P_{\infty,t} c_{t-1}
\end{aligned}$$

If $P_{\infty,t} a_t = 0$, then for the purpose of calculating e_t, recursion for c_{t-1} can be simplified to:

$$c_{t-1} = b_t + L_t^{(0)'} c_t$$

To prove Lemma 2, note that if $P_{\infty,t} a_t = 0$, then we have for any $j < t$:

$$\begin{aligned}
& F_{t-1} ... F_j P_{\infty,j} L_j^{(0)'} ... L_{t-1}^{(0)'} a_t = 0 \\
\Rightarrow & a_t^{'} F_{t-1} ... F_j P_{\infty,j} L_j^{(0)} ... L_{t-1}^{(0)} a_t = 0
\end{aligned}$$

Note that $F_j P_{\infty,j} L_j^{(0)'}$ is PSD, then if $a_t F_{t-1} P_{\infty,t-1} L_{t-1}^{(0)'} a_t = 0$, we have:

$$\begin{aligned}
a_t^{'} F_t (P_{\infty,t-1} - P_{\infty,t-1} H_{t-1}^{'} \Upsilon_{\infty,t-1}^{-1} H_{t-1} P_{\infty,t-1}) F_t^{'} a_t = 0 \ , \text{ if } \Upsilon_{\infty,t-1} > 0 \\
a_t^{'} F_{t-1} P_{\infty,t-1} F_{t-1}^{'} a_t = 0 \ , \text{ if } \Upsilon_{\infty,t-1} = 0
\end{aligned}$$

By properties of PSD matrices, we have:

$$\begin{aligned}
P_{\infty,t-1} L_{t-1}^{(0)'} a_t = 0 \ , \text{ if } \Upsilon_{\infty,t-1} = 0 \\
P_{\infty,t-1} F_t^{'} a_t = 0 \ , \text{ if } \Upsilon_{\infty,t-1} = 0
\end{aligned}$$

By applying the above result recursively, we have the desired result. For c_{j-1}, the term that involves a_t is $P_{\infty,j} L_j^{(0)} ... L_{t-1}^{(0)} a_t = 0$. Therefore, we can ignore a_t in the recursion formula for c_{t-1}.

[29] I am embarrassing myself, but you get the point...

Now we can derive the recursion formula for r_{t-1} and N_{t-1}. First consider the case $\Upsilon_{\infty,t} \neq 0$[30]. Recall that $r_{t-1} = H_t^{'}\Upsilon_t^{-1}d_t + L_t^{'}r_t$. If $\Upsilon_{\infty,t} \neq 0$, we have the recursion formula for r_{t-1} as:

$$
\begin{aligned}
r_{t-1}^{(0)} &= L_t^{(0)'} r_t^{(0)} && \text{(B.1)}\\
r_{t-1}^{(1)} &= H_t^{'}\Upsilon_t^{(1)}d_t + L_t^{(0)'} r_t^{(1)} + L_t^{(1)'} r_t^{(0)} \\
&= H_t^{'}(\Upsilon_{\infty,t}^{-1}d_t - K_t^{(1)'} F_t^{'} r_t^{(0)}) + L_t^{(0)'} r_t^{(1)} && \text{(B.2)}\\
L_t^{(0)} &= F_t(I - K_t^{(0)} H_t) \\
L_t^{(1)} &= -F_t K_t^{(1)} H_t
\end{aligned}
$$

We can perform backwards recursion for $t \in \{t_q, ..., 1\}$ with $r_{t_q}^{(0)} = r_{t_q}$ and $r_{t_q}^{(1)} = 0$, where $K_t^{(0)}$ and $K_t^{(1)}$ are defined in equation (A.5) through (A.7).

To compute N_{t-1}, note that we do not have expressions for $K_t^{(2)}$, but by Lemma 2, we know that N_{t-1} is pre and post-multiplied by $P_{\infty,t}$ (or by $P_t L_t^{(0)'}$ or $P_t F_t^{'}$ or $P_{\infty,t+1}$ in the case of $P_{t,t+1|T}$), and we have established that $P_{\infty,t} L_t^{(0)'} N_t^{(0)} = 0$, we have the simplified formula for N_{t-1}:

$$
\begin{aligned}
N_{t-1}^{(0)} =& L_t^{(0)'} N_t^{(0)} L_t^{(0)} && \text{(B.3)}\\
N_{t-1}^{(1)} =& H_t^{'}\Upsilon_t^{(1)} H_t + \langle L_t^{(1)'} N_t^{(0)} L_t^{(0)} \rangle + L_t^{(0)'} N_t^{(1)} L_t^{(0)} && \text{(B.4)}\\
N_{t-1}^{(2)} =& H_t^{'}\Upsilon_t^{(2)} H_t + \langle L_t^{(1)'} N_t^{(1)} L_t^{(0)} \rangle + L_t^{(0)'} N_t^{(2)} L_t^{(0)} + L_t^{(1)'} N_t^{(0)} L_t^{(1)} && \text{(B.5)}
\end{aligned}
$$

Now consider the case where $\Upsilon_{\infty,t} = 0$. Write r_t as:

$$
\begin{aligned}
r_{t-1} =& H_t^{'}\Upsilon_t^{-1}d_t + L_t^{'}r_t \\
=& H_t^{'}(\Upsilon_{*,t}^{-1} + \kappa^{-1}\Upsilon_t^{(1)})d_t \\
&+ (I - K_t^{(0)}H_t - \kappa^{-1}K_t^{(1)}H_t)^{'} F_t^{'}(r_t^{(0)} + \kappa^{-1}r_t^{(1)}) + \mathcal{O}(\kappa^{-2})
\end{aligned}
$$

where $K_t^{(0)} \equiv K_t^{(*)} = P_{*,t}H_t^{'}\Upsilon_{*,t}^{-1}$ from Appendix A.3.

Rearrange terms and we have:

$$
\begin{aligned}
r_{t-1}^{(0)} &= H_t^{'}\Upsilon_{*,t}^{-1}d_t + (I - K_t^{(0)}H_t)^{'} F_t^{'} r_t^{(0)} \\
r_{t-1}^{(1)} &= H_t^{'}(\Upsilon_t^{(1)}d_t - K_t^{(1)'} F_t^{'} r_t^{(0)} - K_t^{(0)'} F_t^{'} r_t^{(1)}) + F_t^{'} r_t^{(1)}
\end{aligned}
$$

By Lemma 2, we can drop the term that involves $H_t^{'}$ in $r_{t-1}^{(1)}$. The simplified recursive formula for r_{t-1} are:

$$
\begin{aligned}
r_{t-1}^{(0)} &= H_t^{'}\Upsilon_{*,t}^{-1}d_t + (I - K_t^{(*)}H_t)^{'} F_t^{'} r_t^{(0)} && \text{(B.6)}\\
r_{t-1}^{(1)} &= F_t^{'} r_t^{(1)} && \text{(B.7)}
\end{aligned}
$$

Now consider N_{t-1}. Expand N_{t-1} as:

$$
\begin{aligned}
N_{t-1} =& H_t^{'}(\Upsilon_{*,t}^{-1} + \kappa^{-1}\Upsilon_t^{(1)} + \kappa^{-2}\Upsilon_t^{(2)})H_t \\
&+ [I - (K_t^{(*)} + \kappa^{-1}K_t^{(1)} + \kappa^{-2}K_t^{(2)})H_t]^{'} F_t^{'} \\
&\times (N_t^{(0)} + \kappa^{-1}N_t^{(1)} + \kappa^{-2}N_t^{(2)}) \\
&\times F_t[I - (K_t^{(*)} + \kappa^{-1}K_t^{(1)} + \kappa^{-2}K_t^{(2)})H_t] + \mathcal{O}(\kappa^{-3})
\end{aligned}
$$

[30] Here we consider the univariate case.

After some algebra, we have:

$$\begin{aligned}
N_{t-1}^{(0)} =& H_t^{'}\Upsilon_{*,t}^{-1}H_t + (I - K_t^{(*)}H_t)^{'}F_t^{'}N_t^{(0)}F_t(I - K_t^{(*)}H_t) \\
N_{t-1}^{(1)} =& H_t^{'}\Upsilon_t^{(1)}H_t + (I - K_t^{(*)}H_t)^{'}F_t^{'}N_t^{(1)}F_t(I - K_t^{(*)}H_t) \\
& - \langle H_t^{'}K_t^{(1)}F_t^{'}N_t^{(0)}F_t(I - K_t^{(*)}H_t)\rangle \\
N_{t-1}^{(2)} =& H_t^{'}\Upsilon_t^{(2)}H_t + (I - K_t^{(*)}H_t)^{'}F_t^{'}N_t^{(2)}F_t(I - K_t^{(*)}H_t) \\
& - \langle H_t^{'}K_t^{(1)^{'}}F_t^{'}N_t^{(1)}F_t(I - K_t^{(*)}H_t)\rangle - \langle H_t^{'}K_t^{(2)^{'}}F_t^{'}N_t^{(0)}F_t(I - K_t^{(*)}H_t)\rangle \\
& + H_t^{'}K_t^{(1)^{'}}F_t^{'}N_t^{(0)}F_tK_t^{(1)}H_t^{'} + \mathcal{O}(\kappa^{-1})
\end{aligned}$$

By similar argument, we can simplify the recursive formula for N_{t-1} as:

$$N_{t-1}^{(0)} = H_t^{'}\Upsilon_{*,t}^{-1}H_t + (I - K_t^{(*)}H_t)^{'}F_t^{'}N_t^{(0)}F_t(I - K_t^{(*)}H_t) \tag{B.8}$$

$$N_{t-1}^{(1)} = (I - K_t^{(*)}H_t)^{'}F_t^{'}N_t^{(1)}F_t(I - K_t^{(*)}H_t) \tag{B.9}$$

$$N_{t-1}^{(2)} = F_t^{'}N_t^{(2)}F_t \tag{B.10}$$

Here I use a less simplified expressions for $N_{t-1}^{(1)}$ than provided in (Durbin and S.J. Koopman 2003) to maintain numerical PSD of $N_{t-1}^{(1)}$. It is important to keep in mind that recursion for $N_{t-1}^{(2)}$ involves $N_{t-1}^{(1)}$ as well, we need to make sure dropping terms in $N_{t-1}^{(1)}$ does not affect recursion for $N_{t-1}^{(2)}$.

B.4 Smoothed Distribution of Missing Measurements

Suppose at time t, we only partially observe y_t. Denote $y_t^{(1)}$ as the observed part with size n_t, and $y_t^{(2)}$ the unobserved part. I assume y_t are nicely partitioned[31], then we have:

$$\begin{aligned}
y_t^{(1)} &= H_t^{(1)}\xi_t + D_t^{(1)}x_t + w_t^{(1)} \\
y_t^{(2)} &= H_t^{(2)}\xi_t + D_t^{(2)}x_t + w_t^{(2)}
\end{aligned}$$

where $(D_t^{(1)}, H_t^{(1)})$ and $(D_t^{(2)}, H_t^{(2)})$ are the corresponding part of H_t and D_t for $y_t^{(1)}$ and $y_t^{(2)}$, respectively. Similarly, we can rewrite R_t as:

$$R_t = \begin{pmatrix} R_t^{(1,1)} & R_t^{(1,2)} \\ R_t^{(2,1)} & R_t^{(2,2)} \end{pmatrix}$$

Next, I define $Y_t^{(1)}$ as the observed part of the measurement sequence Y_t. It is straight-forward to obtain:

$$E(y_t^{(1)}|Y_t^{(1)}) = y_t^{(1)} \tag{B.11}$$

$$Var(y_t^{(1)}|Y_t^{(1)}) = 0 \tag{B.12}$$

For simplicity, I omit X_t and θ, and use $E(*|Y_t^{(1)})$ to denote smoothed values. To obtain $E(y_t^{(2)}|Y_t^{(1)})$, consider the following:

$$\begin{aligned}
E(y_t^{(2)}|Y_t^{(1)}) &= E(H_t^{(2)}\xi_t + D_t^{(2)}x_t + w_t^{(2)}|Y_t^{(1)}) \\
&= H_t^{(2)}\hat{\xi}_{t|T} + D_t^{(2)}x_t + E(w_t^{(2)}|Y_t^{(1)})
\end{aligned}$$

[31] In practice, we can always permute y_t to achieve the same effect.

To derive an expression for $E(w_t^{(2)}|Y_t^{(1)})$, consider the following:

$$\begin{aligned}
E(w_t^{(2)}|Y_t^{(1)}) &= E[E(w_t^{(2)}|\Xi_t, Y_t^{(1)})|Y_t^{(1)}] \\
&= E[E(w_t^{(2)}|\Xi_t, W_t^{(1)})|Y_t^{(1)}] \\
&= E[E(w_t^{(2)}|\xi_t, w_t^{(1)})|Y_t^{(1)}] \\
&= E[\mathcal{B}_t(y_t^{(1)} - H_t^{(1)}\xi_t - D_t^{(1)}x_t)|Y_t^{(1)}] \\
&= \mathcal{B}_t\epsilon_t \\
\mathcal{B}_t &\equiv R_t^{(2,1)}\left(R_t^{(1,1)}\right)^{-1} \\
\epsilon_t &\equiv y_t^{(1)} - H_t^{(1)}\hat{\xi}_{t|T} - D_t^{(1)}x_t
\end{aligned}$$

where $W_t^{(1)} \equiv \{w_1^{(1)}, ..., w_T^{(1)}\}$. The second equality follows because knowing $(\Xi_t, X_t, Y_t^{(1)})$ is the same as knowing $(\Xi_t, X_t, W_t^{(1)})$. The third equality follows from the fact that conditioned on $(\xi_t, x_t, w_t^{(1)})$, $w_t^{(2)}$ is independent of measurements and state values in other periods. The fourth equality holds by applying Lemma 1:

$$\begin{aligned}
E(w_t^{(2)}|\xi_t, w_t^{(1)}) &= E(w_t^{(2)}|\xi_t) + \mathcal{B}_t[w_t^{(1)} - E(w_t^{(1)}|\xi_t)] \\
&= \mathcal{B}_t(y_t^{(1)} - H_t^{(1)}\xi_t - D_t^{(1)}x_t)
\end{aligned}$$

Now we have expression for $\hat{y}_{t|T}^{(2)} \equiv E(y_t^{(2)}|Y_t^{(1)})$:

$$E(y_t^{(2)}|Y_t^{(1)}) = H_t^{(2)}\hat{\xi}_{t|T} + D_t^{(2)}x_t + \mathcal{B}_t\epsilon_t \tag{B.13}$$

To derive $Var(y_t^{(2)}|Y_t^{(1)}) \equiv E[(y_t^{(2)} - \hat{y}_{t|T}^{(2)})(y_t^{(2)} - \hat{y}_{t|T}^{(2)})'|Y_t^{(1)}]$, consider the following:

$$\begin{aligned}
Var(y_t^{(2)}|Y_t^{(1)}) =& E\{[H_t^{(2)}(\xi_t - \hat{\xi}_{t|T}) + w_t^{(2)} - \mathcal{B}_t\epsilon_t][H_t^{(2)}(\xi_t - \hat{\xi}_{t|T}) + w_t^{(2)} - \mathcal{B}_t\epsilon_t]'|Y_t^{(1)}\} \\
=& H_t^{(2)}P_{t|T}H_t^{(2)'} + \langle H_t^{(2)}E[(\xi_t - \hat{\xi}_{t|T})w_t^{(2)'}|Y_t^{(1)}]\rangle + Var(w_t^{(2)}|Y_t^{(1)}) \\
Var(w_t^{(2)}|Y_t^{(1)}) =& E(w_t^{(2)}w_t^{(2)'}|Y_t^{(1)}) - \mathcal{B}_t\epsilon_t\epsilon_t'\mathcal{B}_t'
\end{aligned}$$

We need to find expressions for $E(\xi_t w_t^{(2)'}|Y_t^{(1)})$ and $Var(w_t^{(2)}|Y_t^{(1)})$. The derivation follows closely from (R. H. Shumway 2000). First consider $E(\xi_t w_t^{(2)'}|Y_t^{(1)})$:

$$\begin{aligned}
E(\xi_t w_t^{(2)'}|Y_t^{(1)}) &= E\left[\xi_t E\left(w_t^{(2)'}|\xi_t, w_t^{(1)}\right)|Y_t^{(1)}\right] \\
&= \hat{\xi}_{t|T}\epsilon_t'\mathcal{B}_t' - P_{t|T}H_t^{(1)'}\mathcal{B}_t'
\end{aligned}$$

Next, by Lemma 1, we have:

$$\begin{aligned}
Var(w_t^{(2)}|w_t^{(1)}, \xi_t) &= Var(w_t^{(2)}|\xi_t) - Cov(w_t^{(2)}, w_t^{(1)}|\xi_t)Var(w_t^{(1)}|\xi_t)^{-1}Cov(w_t^{(2)}, w_t^{(1)}|\xi_t)' \\
&= R_t^{(2,2)} - R_t^{(2,1)}\left(R_t^{(1,1)}\right)^{-1}R_t^{(1,2)}
\end{aligned}$$

Since we have already known $E(w_t^{(2)}|\Xi_t, W_t^{(1)}) = \mathcal{B}_t w_t^{(1)}$, we have:

$$\begin{aligned}
Var(w_t^{(2)}|Y_t^{(1)}) &= E\left[E\left(w_t^{(2)}w_t^{(2)'}|\xi_t, w_t^{(1)}\right)|Y_t^{(1)}\right] - \mathcal{B}_t\epsilon_t\epsilon_t'\mathcal{B}_t' \\
&= R_t^{(2,2)} - R_t^{(2,1)}\left(R_t^{(1,1)}\right)^{-1}R_t^{(1,2)} + \mathcal{B}_t H_t^{(1)}P_{t|T}H_t^{(1)'}\mathcal{B}_t'
\end{aligned}$$

Now we have all the elements, and $Var(y_t^{(2)}|Y_t^{(1)})$ is:

$$Var(y_t^{(2)}|Y_t^{(1)}) = R_t^{(2,2)} - R_t^{(2,1)}\left(R_t^{(1,1)}\right)^{-1}R_t^{(1,2)} + \left(H_t^{(2)} - \mathcal{B}_t H_t^{(1)}\right)P_{t|T}\left(H_t^{(2)} - \mathcal{B}_t H_t^{(1)}\right)' \tag{B.14}$$

C Parameter Estimation

C.1 Derivation of Marginal Likelihood

First I derive the regular log likelihood $l(Y_T) \equiv logL(Y_T|X_T;\theta)$:

$$l(Y_T) = log\mathbb{P}(y_1|x_T;\theta) + \sum_{t=2}^{T} log\mathbb{P}(y_t|Y_{t-1},X_T;\theta)$$

Since $y_t|Y_{t-1},X_T;\theta$ is normally distributed with conditional mean and variance as:

$$\begin{aligned} E(y_t|Y_{t-1},X_T;\theta) &= H_t\hat{\xi}_{t|t-1} + D_tX_t \\ Var(y_t|Y_{t-1},X_T;\theta) &= \Upsilon_t \end{aligned}$$

Now we can write $l(Y_T)$ as:

$$l(Y_T) = Const - \frac{1}{2}\sum_{t=1}^{T}(log|\Upsilon_t| + d_t^{'}\Upsilon_t^{-1}d_t)$$

where $|\cdot|$ denotes pseudo-determinant. In the univariate case, $|\cdot|$ is reduced to calculating the absolute value.

Next we derive diffuse likelihood $l_d(Y_T) \equiv logL(Y_T|X_T;\theta)$ with diffuse priors. First consider $\Upsilon_{\infty,t} > 0$ (again we only consider the univariate case). In this case we have:

$$\begin{aligned} \Upsilon_t^{-1} &= \kappa^{-1}\Upsilon_{\infty,t}^{-1} + \mathcal{O}(\kappa^{-2}) \\ log|\Upsilon_t^{-1}| &= log|\kappa^{-1}\Upsilon_{\infty,t}^{-1} + \mathcal{O}(\kappa^{-2})| \\ &= -log(\kappa) + log|\Upsilon_{\infty,t}^{-1} + \mathcal{O}(\kappa^{-1})| \\ \lim_{\kappa\to\infty}(-log|\Upsilon_t| + log(\kappa)) &= -log|\Upsilon_{\infty,t}| \\ \lim_{\kappa\to\infty} d_t^{'}\Upsilon_t^{-1}d_t &= 0 \end{aligned}$$

If $\Upsilon_{\infty,t} = 0$, we have:

$$\begin{aligned} \lim_{\kappa\to\infty}(-log|\Upsilon_t|) &= -log\Upsilon_{*,t} \\ \lim_{\kappa\to\infty}(d_t^{'}\Upsilon_t^{-1}d_t) &= d_t^{'}\Upsilon_{*,t}^{-1}d_t \end{aligned}$$

The overall expression of $l_d(Y_T)$ is:

$$\begin{aligned} logl_d(Y_T) =& Const - \frac{1}{2}\sum_{t=1}^{t_q}\Psi_t - \frac{1}{2}\sum_{t=t_q+1}^{T}(log|\Upsilon_t| + d_t^{'}\Upsilon_t^{-1}d_t) \\ \Psi_t =& \begin{cases} log|\Upsilon_{\infty,t}| & \Upsilon_{\infty,t} > 0 \\ log|\Upsilon_{*,t}| + d_t^{'}\Upsilon_{*,t}^{-1}d_t & \Upsilon_{\infty,t} = 0 \end{cases} \end{aligned}$$

Finally, I will briefly discuss marginal likelihood $l_m(Y_T)$. (Marc K Francke, Siem Jan Koopman, and De Vos 2010) provides an excellent overview of profile likelihood, diffuse likelihood, and marginal likelihood. The primary advantage of using marginal likelihood is its robustness against H_t, F_t and A in diffuse Kalman filters being dependent on θ, in which case diffuse likelihood is not appropriate for parameter estimation. In other words, marginal likelihood properly integrates out initial values δ. For example, marginal likelihood returns sensible estimates when we allow the possibility of unit roots (see (Marc K

Francke and Aart F de Vos 2007) for more details). In addition, (Marc K Francke, Siem Jan Koopman, and De Vos 2010) shows:

$$
\begin{aligned}
l_m(Y_T) &= l_d(Y_T) + \frac{1}{2} log \left| \sum_{t=1}^{T} (Z_t^{'} Z_t) \right| \\
Z_t &= H_t \phi_t \\
\phi_{t+1} &= F_t \phi_t \\
\phi_1 &= A
\end{aligned}
$$

Here I omit the proof as it is straightforward in (Marc K Francke, Siem Jan Koopman, and De Vos 2010). Instead I will explain two key intermediate steps in their proof. In Section 2.3 of (Marc K Francke, Siem Jan Koopman, and De Vos 2010), the following claim:

$$
\begin{aligned}
&(\Omega A, X)^{'} A (A \Omega A)^{-1} A^{'} = (A, 0)^{'} \Leftrightarrow (\Omega A, X)^{'} \Omega^{-1} M_{\Omega} = (A, 0)^{'} \\
\Rightarrow &A (A \Omega A)^{-1} A^{'} = \Omega^{-1} M_{\Omega}
\end{aligned}
$$

holds because of the uniqueness of the hat matrix in the case of generalized linear regression.

For the second claim:

$$
\begin{aligned}
|\Omega| \cdot |A^{'} A| \cdot |X^{'} X| &= |(A, X)^{'} \Omega (A, X)| \\
&= |A^{'} \Omega A| \cdot |X^{'} \Omega X - X^{'} \Omega A (A^{'} \Omega A)^{-1} A^{'} \Omega X|
\end{aligned}
$$

To establish the first equality, note that:

$$
|(A, X)^{'} (A, X)| = |A^{'} A| \cdot |X^{'} X|
$$

because $A^{'} X = 0$, and we can use the properties of block diagonal matrix. In addition, (A, X) is a square matrix of the same dimension as Ω, so we can write:

$$
\begin{aligned}
|(A, X)^{'} (A, X)| \cdot |\Omega| &= |(A, X)^{'}| \cdot |\Omega| \cdot |(A, X)| \\
&= |(A, X)^{'} \Omega (A, X)|
\end{aligned}
$$

To establish the second equality, we can simply apply another trick of block matrices described in (Petersen, Pedersen, et al. 2008).

C.2 EM Algorithm Premier

Denote Y as observed measurements, ξ as hidden states, $g(\theta)$ as some function parameterized by θ, and θ as parameters to be estimated. We want to optimize:

$$
\begin{aligned}
L(\theta) &= log[P(Y|\theta)] + g(\theta) \\
&= log \left[\frac{P(Y, \xi|\theta)}{P(\xi|Y, \theta)} \right] + g(\theta) \\
&= log[P(Y, \xi|\theta)] - log[P(\xi|Y, \theta)] + g(\theta)
\end{aligned} \tag{C.1}
$$

Take expectation of equation (C.1) wrt. some distribution of ξ with pdf $f(\xi)$ and get:

$$
\begin{aligned}
L(\theta) = &\int f(\xi) log[P(Y, \xi|\theta)] d\xi + g(\theta) \\
&- \int f(\xi) log[P(\xi|Y, \theta)] d\xi
\end{aligned}
$$

To optimize $L(\theta)$ we can iterate through the E steps and M steps to achieve a local maximum. By Jensen's inequality, we have the second term in equation (C.1) minimized when $f(\xi) = P(\xi|Y,\theta)$ (E-step). If we define:

$$Q(\theta) = \int log[P(Y,\xi|\theta)]f(\xi|Y,\theta)d\xi + g(\theta) \tag{C.2}$$

then maximizing $Q(\theta)$ is equivalent to maximizing $L(\theta)$. For a given $\hat{\theta}$ and $P(\xi|Y,\hat{\theta})$, we find θ to optimize the first term (M-step). Use the new θ as $\hat{\theta}$ for the next iteration, and we will reach a local maximum. It is important to note that for a given $\hat{\theta}$, $f(\xi|Y,\hat{\theta})$ is a given quantity and does not change wrt. θ.

C.3 Derivation of Log-likelihood for $G(\theta,\theta_i)$

From Section 7.3 we know that:

$$G(\theta,\theta_i) \equiv \int log[\mathbb{P}(Y_T,\Xi_T|X_T,\theta)]\mathbb{P}(\Xi_T|Y_T,X_T,\theta_i)d\Xi_T + \frac{1}{2}log\left|\sum_{t=1}^{T}(Z_t^{'}Z_t)\right|$$

Since the second term does not depend on θ_i, here we only discuss the first term.

By Markov property, we can rewrite $log[\mathbb{P}(Y_T,\Xi_T|X_T,\theta)]$ as:

$$\begin{aligned} log[\mathbb{P}(Y_T,\Xi_T|X_T;\theta)] &= \sum_{t=1}^{T} log[\mathbb{P}(\xi_t|\xi_{t-1},x_{t-1};\theta)] + \sum_{t=1}^{T} log[\mathbb{P}(y_t|\xi_t,x_t;\theta)] \\ &= log[\mathbb{P}(\xi_1;\theta)] + \sum_{t=2}^{T} log[\mathbb{P}(\xi_t|\xi_{t-1},x_{t-1};\theta)] \\ &+ \sum_{t=1}^{T} log[\mathbb{P}(y_t|\xi_t,x_t;\theta)] \end{aligned} \tag{C.3}$$

I define $G_1^t(\theta,\theta_i)$, $G_2^t(\theta,\theta_i)$, and $G_0(\theta,\theta_i)$ as:

$$\begin{aligned} G_1^t(\theta,\theta_i) &\equiv \int log[\mathbb{P}(\xi_t|\xi_{t-1},\theta)]\mathbb{P}(\Xi_T|Y_T,X_T,\theta_i)d\Xi_T \\ &= const - \frac{1}{2}log(|\mathbb{Q}_{t-1}|) - \frac{1}{2}Tr[E(\mathbb{Q}_{t-1}^{-1}\delta_t\delta_t^{'}|Y_T,X_T;\theta_i)] \\ &= const - \frac{1}{2}log(|\mathbb{Q}_{t-1}|) - \frac{1}{2}Tr[\mathbb{Q}_{t-1}^{-1}E(\delta_t\delta_t^{'}|Y_T,X_T;\theta_i)] \end{aligned} \tag{C.4}$$

$$\begin{aligned} G_2^t(\theta,\theta_i) &\equiv \int log[\mathbb{P}(y_t|\xi_t,x_t;\theta)]\mathbb{P}(\Xi_T|Y_T,X_T;\theta_i)d\Xi_T \\ &= const - \frac{1}{2}log(|\mathbb{R}_t|) - \frac{1}{2}Tr[\mathbb{R}_t^{-1}E(\chi_t\chi_t^{'}|Y_T,X_T;\theta_i)] \end{aligned} \tag{C.5}$$

$$\begin{aligned} G_0(\theta,\theta_i) &\equiv const - \lim_{\kappa\to\infty}\left\{\frac{1}{2}log(|\mathbb{P}_{1|0}|) + \frac{1}{2}Tr[\mathbb{P}_{1|0}^{-1}E(\delta_*\delta_*^{'}|Y_T,X_T;\theta_i)] - \frac{q}{2}log(\kappa)\right\} \\ &= const - \frac{1}{2}log(|\mathbb{P}_*|) - \frac{1}{2}Tr(\mathbb{P}_*^{-1}E(\delta_*\delta_*^{'}|Y_T,X_T;\theta_i)) \end{aligned} \tag{C.6}$$

$$\begin{aligned} \delta_t &\equiv \xi_t - \mathbb{F}_{t-1}\xi_{t-1} - \mathbb{B}_{t-1}x_{t-1} \\ \delta_* &\equiv \xi_1 - \mathbb{A} \\ \chi_t &\equiv y_t - \mathbb{H}_t\xi_t - \mathbb{D}_t x_t \end{aligned}$$

If $\mathbb{Q}_t$ does not have full rank, I use `scipy.linalg.pinvh` and `scipy.linalg.eigh`[32] to find the pseudo inverse and the pseudo determinant of $\mathbb{Q}_t$.

[32] After finding all eigenvalues of Q_t, multiply non-zero eigenvalues to get the determinant.

Denote $\Theta_i \equiv (Y_T, X_T; \theta_i)$, I now derive expressions for $E(\delta_t\delta_t^{'}|\Theta_i)$, $E(\chi_t\chi_t^{'})|\Theta_i)$, and $E(\delta_*\delta_*^{'}|\Theta_i)$. Following (Siem Jan Koopman and Shephard 1992b), we have:

$$
\begin{aligned}
E(\delta_t\delta_t^{'}|\Theta_i) =& E(\xi_t\xi_t^{'}|\Theta_i) - \langle \mathbb{F}_{t-1}E(\xi_{t-1}\xi_t^{'}|\Theta_i)\rangle - \langle \mathbb{B}_{t-1}x_{t-1}[E(\xi_t|\Theta_i) - \mathbb{F}_{t-1}E(\xi_{t-1}|\Theta_i)]^{'}\rangle \\
&+ \mathbb{F}_{t-1}E(\xi_{t-1}\xi_{t-1}^{'}|\Theta_i)\mathbb{F}_{t-1}^{'} + \mathbb{B}_{t-1}x_{t-1}x_{t-1}^{'}\mathbb{B}_{t-1}^{'} \\
=& (\hat{\xi}_{t|T} - \mathbb{F}_{t-1}\hat{\xi}_{t-1|T} - \mathbb{B}_{t-1}x_{t-1})(\hat{\xi}_{t|T} - \mathbb{F}_{t-1}\hat{\xi}_{t-1|T} - \mathbb{B}_{t-1}x_{t-1})^{'} \\
&+ P_{t|T} - \langle \mathbb{F}_{t-1}P_{t-1,t|T}\rangle + \mathbb{F}_{t-1}P_{t-1|T}\mathbb{F}_{t-1}^{'} \qquad \text{(C.7)} \\
E(\chi_t\chi_t^{'}|\Theta_i) =& (y_t - \mathbb{D}_t x_t)(y_t - \mathbb{D}_t x_t)^{'} - \langle \mathbb{H}_t\hat{\xi}_{t|T}(y_t - \mathbb{D}_t x_t)^{'}\rangle + \mathbb{H}_t E(\xi_t\xi_t^{'}|\Theta_i)\mathbb{H}_t^{'} \\
=& (y_t - \mathbb{H}_t\hat{\xi}_{t|T} - \mathbb{D}_t x_t)(y_t - \mathbb{H}_t\hat{\xi}_{t|T} - \mathbb{D}_t x_t)^{'} + \mathbb{H}_t P_{t|T}\mathbb{H}_t^{'} \qquad \text{(C.8)} \\
E(\delta_*\delta_*^{'}|\Theta_i) =& E(\xi_1\xi_1^{'}|\Theta_i) - \langle \hat{\xi}_{1|T}\mathbb{A}^{'}\rangle + \mathbb{A}\mathbb{A}^{'} = (\hat{\xi}_{1|T} - \mathbb{A})(\hat{\xi}_{1|T} - \mathbb{A})^{'} + P_{1|T} \qquad \text{(C.9)} \\
E(\xi_t\xi_t^{'}|\Theta_i) =& \hat{\xi}_{t|T}\hat{\xi}_{t|T}^{'} + P_{t|T} \\
E(\xi_{t-1}\xi_t^{'}|\Theta_i) =& \hat{\xi}_{t-1|T}\hat{\xi}_{t|T}^{'} + P_{t-1,t|T}
\end{aligned}
$$

where $\mathbb{M}_t \equiv (\mathbb{Q}_t, \mathbb{R}_t, \mathbb{F}_t, \mathbb{H}_t, \mathbb{B}_t, \mathbb{D}_t, \mathbb{P}_*, \mathbb{A})$ are the system matrices parameterized by θ. Now $G(\theta, \theta_i)$ is:

$$
G(\theta, \theta_i) = G_0(\theta, \theta_i) + \sum_{t=2}^{T} G_1^t(\theta, \theta_i) + \sum_{t=1}^{T} G_2^t(\theta, \theta_i) + \frac{1}{2} log \left| \sum_{t=1}^{T} (\mathbb{Z}_t^{'}\mathbb{Z}_t) \right|
$$

where $\mathbb{Z}_t = \mathbb{H}_t\phi_t$ and $\phi_1 = \mathbb{S}$, with $\mathbb{S}$ as the parameterized counterpart of selection matrix A in equation (5.9). Using equation (C.4) through (C.9), we have an expression for $G(\theta, \theta_i)$. We can then use numerical methods[33] to find θ_{i+1} that maximizes $G(\theta, \theta_i)$.

For systems with missing measurements, the univariate smoothing technique remains valid for calculating state estimates and $E(\delta_t\delta_t^{'}|\Theta_i)$, but we need to pay special attention to $E(\chi_t\chi_t^{'}|\Theta_i)$. Here y_t are observed measurements at time t. Therefore, it is important in practice to align the measurement index of y_t and $\mathbb{M}_t$[34].

[33] If system matrices are constant, we have a closed form solution. But I decide to use numerical optimizations so that the EM is also able to solve a BSTS model with complex system matrices.

[34] (R. H. Shumway and D. S. Stoffer 1982), (R. H. Shumway 2000), and (Robert H Shumway and David S Stoffer 2017) offers alternative derivations without changing the size of $\mathbb{M}_t$. The primary advantage is to computing score function more easily, but as I am using non-gradient based algorithms for `linkalman`, I opt for calculating χ_t only for observed measurements, which is computationally more efficient.