Construction an Informative Prior Distribution of Noise in Seasonal Adjustment

Linyi Guo

Thesis submitted to the University of Ottawa in partial Fulfillment
of the requirements for the Master of Science degree

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

Abstract

Time series data is very common in our daily life. Since they are related to time, most of them show a periodicity. The existence of this periodic influence leads to our research problem, seasonal adjustment. Seasonal adjustment is generally applied around us, especially in areas of economy and finance. Over the last few decades, scholars around the world made a lot of contributions in this area, and one of the latest methods is X-13ARIMA-SEATS, which is built on ARIMA models and linear filters. On the other hand, state space modelling (abbreviated to SSM) is also a popular method to solve this problem and researchers including J. Durbin, S.J. Koopman and and A. Harvery have contributed a lot of work to it. Unlike linear filters and ARIMA models, the study on SSM starts relatively late, thus it has not been studied and developed widely for the seasonal adjustment problem. And SSMs have a lot advantages over those ARIMA-based and filter-based methods such as flexibility, the understandable structure and the potential to do partial pooling, but in practice, its default decomposition result behaves bad in some cases, such as excessively spiky trend series; on the contrary, X-13ARIMA-SEATS could output good decomposition result for us to analyze, but it can't be tweaked or combined as easily as generative models and behaves like a black-box. In this paper, we shall use Bayesian inference to combine both methods' characteristics together. Simultaneously, to show the advantage of using SSMs concretely, we shall give a simple application in partial pooling.

Dedication

This thesis is dedicated to my late father, Guanghua Guo.

Acknowledgements

Firstly, I would like to thank the University of Ottawa to give me this chance to access these wonderful educational resources and strengthen my professional skills. Secondly, I want to thank my supervisor Aaron Smith for his patience, suggestions and guidance during the process of thesis writing. His profound knowledge has inspired me with many good ideas and helped me solve a lot of problems. Then I wan to thank the professors in our department including Mayer Alvo, Raluca Balan, Benoit Dionne, Maia Fraser, Gilles Lamothe, Mahmoud Zarepour and other employees including Mayada El-Maalouf, Diane Demers, etc. I am also glad to make a lot of friends from different countries in the last two years. Finally, I sincerely thank my dear mother Baozhi Sun and my considerate girlfriend Gracie Guo. It is their support and understanding that encourages me along the way.

Contents

1 Introduction **1**

2 ARIMA models and X-11 **5**

2.1 ARIMA models . . . 5

2.1.1 ARMA . . . 5

2.1.2 ARIMA and SARIMA . . . 6

2.2 Techniques used for decomposition . . . 7

2.3 X-11 method . . . 8

3 State space modelling and the Kalman filter **11**

3.1 Introduction to state space modelling . . . 11

3.2 Common state space models . . . 13

3.2.1 Structural time series models . . . 13

3.2.2 ARIMA models . . . 15

3.2.3 Regression models . . . 16

3.3 The Kalman filter . . . 17

3.3.1 Filtering process . . . 18

3.3.2 Smoothing process . . . 19

3.4 Advantages of SSMs over ARIMA models . . . 19

4 Bayesian analysis **23**

4.1 Introduction . . . 23

4.2 Behaviour of maximum likelihood estimators . . . 24

4.3 Loss functions and optimization . . . 27

4.4 Simplification of parameters in SSMs . . . 29

4.5 Bayesian analysis . . . 31

4.6 Empirical prior distributions . . . 33

4.7 Weight adjustment for different lengths . . . 36

4.8 Prediction Comparison . . . 38

5 Application **40**

A Kalman filter **48**

A.1 Filtering process . . . 48

A.2 Smoothing process . . . 49

B Other supplement **51**

List of Figures

1.1 Observed Data Distribution 1
1.2 AirPassengers dataset and its decomposition 2

2.1 Examples of the stationary and nonstationary series 6
2.2 Examples of nonstationary series 7

3.1 State space models 12
3.2 Non-hierarchical and hierarchical models 20

4.1 Decomposition comparison between X-11 and SSM(MLE) 26
4.2 Decomposition comparison between X-11 and SSM(MLE) from 2000 to 2004 26
4.3 Decomposition comparison among X-11, SSM(MLE) and SSM(1,1,1) from 2000 to 2004 27
4.4 Decomposition comparison between X-11 and SSMs from 2000 to 2004 28
4.5 Variability proportions absorbed by three components 29
4.6 Distributions of variance estimators 30
4.7 Comparison of variance distributions 32
4.8 Boxplots of decomposition errors 33
4.9 Densities of decomposition errors 33
4.10 Empirical prior distributions 34
4.11 Comparison of variance distributions (2) 35
4.12 Relations between log-likelihood and length 37

5.1 Classical decomposition of the unemployment dataset 40
5.2 Empirical distributions of the irregular variance from 8 groups 41
5.3 Empirical distributions of the trend variance from 8 groups 42
5.4 Decomposition comparison from 2000 to 2004 42
5.5 Comparison of the variability proportions 43
5.6 Comparison of predictions for the next year 44
5.7 Prediction with 95% confidence intervals 44
5.8 Trends before partial pooling 45
5.9 Trends after complete pooling 46
5.10 Trends after partial pooling 47
5.11 Comparison of trend differences from three pooling methods 47

B.1 Boxplot comparison of decomposition errors 52

List of Tables

3.1 Dimensions of notations 12

4.1 Trend and Seasonal components' error(unemployment) 28
4.2 Information of decomposition error 33
4.3 Log-likelihood under different length and variances of the same dataset 37
4.4 Statistics of decomposition errors 39
4.5 Statistics of prediction errors 39

5.1 Information of SSMs used for simulation 42
5.2 Sum of the squared error 44

B.1 Information of decomposition error(2) 51

Chapter 1

Introduction

Seasonal adjustment is widely applied around us. In the areas of economics and the signal processing of engineering, people need to deal with various time series data almost everyday. The essential of both is to study how to decompose an existing time series and make an accurate prediction. Generally, one time series data in economics could be divided into different *components*, such as the trend, the seasonal and the irregular series. These components in signal processing are also called *signals*. Due to the influence of seasonal movements and some other calendar effects like the Christmas, the Easter and the Chinese New Year, the raw data is usually hard to utilize for analysis directly. Therefore, removing those undesired signals is important for our analysis. Meanwhile, to obtain a good forecasting, ensuring an accurate decomposition of the data is also necessary.

To illustrate the significance of the decomposition, let's look at a simple example. Suppose Figure 1.1 is our observation:

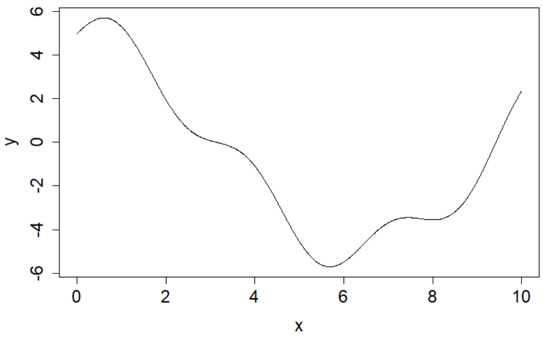


Figure 1.1: Observed Data Distribution

It seems that there is barely no pattern behind it, and one reasonable prediction we could make is that it would increase later. But actually the raw data is from the function:

$$y = sin2x + 5cos\frac{x}{2}, \tag{1.1}$$

which means if we could find the expression of these two parts, then our prediction would be prefect! Therefore, if we could find the accurate expression of our components, it would not only help us with analysis but also for the future prediction. But in the real life, a true time series dataset is always more complicated. The general idea is to write a decomposition of the observed time series $\{y_t\}$ in terms of some easily-interpretable latent variables and some residual noise. For practical purposes we often try to use linear models of the form

$$y_t = \sum_k x_t^k + \epsilon_t,$$

where x_t^k are the easily-interpreted latent variables (such as a periodic function corresponding to seasonal effects, an overall trend, the indicator function for special holidays, and so on) and ϵ_t is some small residual randomness. The most common simple decomposition has two explanatory series, corresponding to an overall trend and to seasonal effects.

For example, *AirPassengers* dataset is the monthly totals of international airline passengers from 1949 to 1960. With the *X-11* method or other methods (see Chapter 2), the original dataset could be decomposed into two components with an residual easily, see Figure 1.2. The purpose of this paper is to explore how to use state space models and the Kalman filter, which we shall introduce later, to generate results similar to those from X-11.

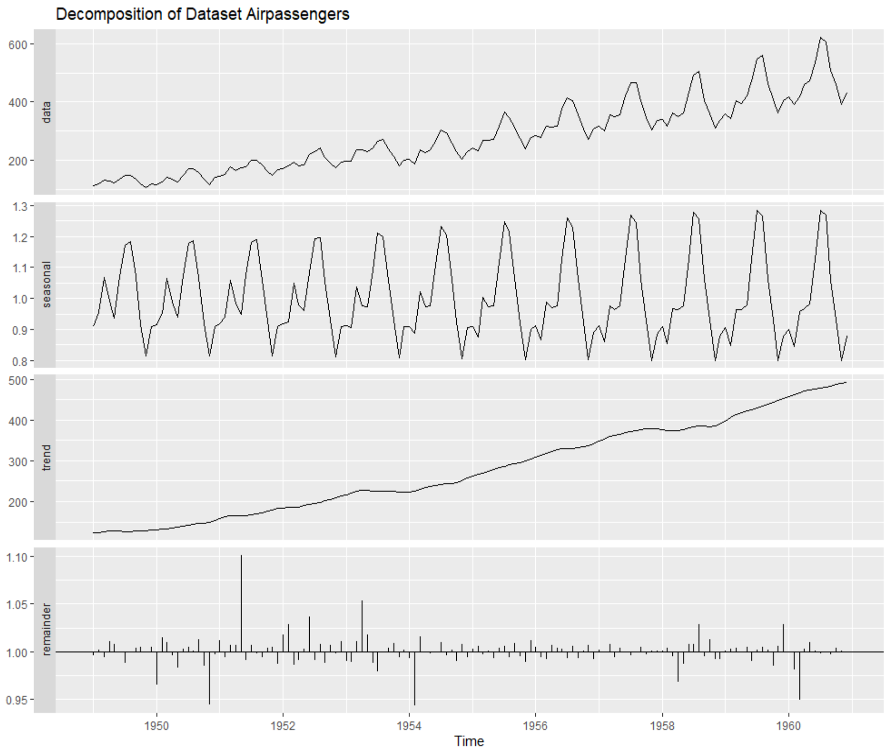


Figure 1.2: AirPassengers dataset and its decomposition

With the appearance and development of ARIMA processes, statistical agencies around the world came up with various ad-hoc fixes for different real census-type data. The history of the seasonal adjustment problem could be traced back to 1960s, at which time the *X-11* program was first proposed by U.S. Bureau of the Census, see [Shiskin et al., 1967]. In 1980, Statistics Canada came up with a new program *X-11-ARIMA* ([Dagum, 1980]), where people could utilize ARIMA models to extend one time series to overcome the inaccuracy of the beginning and ending from *X-11* method. Then U.S. Census Bureau developed an improved version *X-12-ARIMA*

([Findley et al., 1998]) based on X-11-ARIMA. Just two years before that, the Bank of Spain came up with another ARIMA-model-based method called *TRAMO-SEATS*[1] (see [Gomez and Maravall, 1996]), which is used widely in European official statistics agencies at that time. This program is developed from a series of papers by Victor Gómez and Agustin Maravall (see [Gómez and Maravall, 2001] and [Caporello and Maravall, 2004]). In 2007, U.S. Census Bureau brought up *X-13ARIMA-SEATS* (see [Monsell, 2007]), which basically combined all the previous work together and is the up-to-date method used in official statistics agencies around the world.

Generally speaking, because of the existence of outliers, calendar effects and other factors, the first step in seasonal adjustment is to preprocess our raw data. In both widely used methods TRAMO-SEATS and X-13ARIMA-SEATS, this step is achieved by two ARIMA-model-based methods separately, TRAMO and RegARIMA. The second step after preprocessing is usually to decompose the processed dataset. For TRAMO-SEATS, this step is handled by SEATS. For X-13ARIMA-SEATS, you can either choose X-11 or SEATS. The difference is that X-11 is a non-parametric universal method which use linear filters (moving averages) and SEATS is an ARIMA-model-based parametric method. These methods could give us good results in most instances, but still have a few shortcomings (see Section 3.4). The theories behind these methods could be referred to Chapter 2 and the references we mentioned in the last paragraph.

On the other hand, compared with ARIMA models, the state space model (sometimes also known as the *hidden Markov model*, abbreviated to SSM or HMM) is also an efficient modelling method to various problems including seasonal adjustment and was first proposed in [Kalman, 1960]. To fit the state space model, R.E. Kalman came up with the well-known algorithm, the Kalman filter, which is applied well in linear systems and used widely in control theory, signal processing, Guidance, navigation and control and so on. Methods such as the extended Kalman filter (see [Jazwinski, 2007]) and the particle filter (see [Robert and Casella, 2013a]) are developed to solve nonlinear system problems, but they are all applied to the state space model. We shall give a brief introduction of the SSM and the Kalman filter in Chapter 3.

State space models have many advantages over the previous models used in seasonal adjustment. For example, every ARIMA model could be transformed into a state space model but not vice versa, which means we could model more general system with SSMs. And another obvious advantage of SSMs is its interpretable structure. For instance, we could treat the observation $\{y_t\}$ as a combination of the irregular, trend and seasonal components $\{I_t\}$, $\{T_t\}$ and $\{S_t\}$, and build two processes for the trend and seasonality according to our understanding:

$$\begin{aligned} y_t &= T_t + S_t + I_t, \\ T_{t+1} &= T_t + \eta_t, \\ S_{t+1} &= -\sum_{j=1}^{s-1} S_{t+1-j} + \omega_t, \end{aligned}$$

where $t = 1, \dots, n$, I_t, s is the length of each seasonal cycle and η_t, ω_t are independent and identically distributed Gaussian noises with mean 0 and variances σ_I^2, σ_T^2, σ_S^2.

[1] "TRAMO" stands for "Time series Regression with ARIMA noise, Missing values and Outliers" and "SEATS" stands for "Signal Extraction in ARIMA Time Series".

In this model we could tell exactly that:

- the observation y_t has three different parts;
- the trend component is a random walk in fact;
- the summation of seasonal components over one period s follows a normal distribution with mean 0 and variance σ_S^2;
- the irregular component is a gaussian noise with mean 0 and variance σ_I^2.

Therefore, before we write down the state space model for one system, we basically have an intuitive understanding of what our problem and components are. More advantages of SSMs are given in Section 3.4 and [Durbin and Koopman, 2012].

The main purpose and the important achievement of this paper is to apply Bayesian analysis to SSMs to generate the similar decomposition result in terms of those from X-11. Over the last few decades, methods such as X-11 and SEATS have been used widely in government departments and statistics agencies to deal with the decomposition problem and their results have been proved to be useful and convincing by experts. Based on this fact, the main purpose of this paper is to explore how to use SSMs to generate the similar decomposition result compared with X-11's. Specifically, we will first show the deficiency of the maximum likelihood estimate, and then introduce a loss-based method to force our SSMs to generate the satisfying decomposition result. Nevertheless, since our goal is to only apply state space models without using X-11 or other ARIMA-model-based methods, we come up with an empirical-Bayesian-based method to get rid of the dependence on X-11 and compare its results with those from other estimators. All these details are given in Chapter 4. We shall also use a real dataset *unemployment* to verify that the empirical prior distribution does help us to optimize the default decomposition result from SSMs and to make a compromise between X-11 and the standard SSM. In addition, we shall use two real sales datasets to show an important superiority of SSMs, namely partial pooling, which cannot be realized by ARIMA.

Chapter 2

ARIMA models and X-11

In this chapter, we shall talk about the autoregressive integrated moving-average (abbreviated to ARIMA) models and the methodologies used in statistical agencies throughout the world. As mentioned in Chapter 1, the first method proposed to solve the seasonal adjustment problem is *X-11*, which is a combination of linear filters. But after that, almost every new method benefits from ARIMA models, such as X-11-ARIMA, X-12-ARIMA and TRAMO-SEATS. From this point, we could realize the importance of ARIMA models in these conventional seasonal adjustment methods. In Section 2.1, we will give a brief introduction of ARIMA and other related models. More details could be found in [Brockwell and Davis, 2016]. Section 2.2 gives a general introduction of techniques used for decomposing a time series dataset. Then we shall explain the theories behind X-11 in Section 2.3. [Harvey et al., 2018] and [Dagum and Bianconcini, 2016] have given a detailed explanation of other methods besides X-11.

2.1 ARIMA models

ARIMA, the abbreviation of autoregressive integrated moving average, could be treated as one of the most important models in time series area. ARMA is another fundamental model, which first appeared in 1938 (see [Wold, 1938]) and became popular since 1970 (see [Box and Jenkins, 1970]) and could be viewed as a simplified version of ARIMA. It is used to analyse the stationary process, but in practice, most of datasets are not stationary, and ARIMA is designed for these cases. We will talk about them in Subsections 2.1.1 and 2.1.2 respectively. Examples of the stationary and nonstationary series are showed in Figure 2.1.

2.1.1 ARMA

ARMA models could be treated as the combination of an autoregressive model and a moving-average model, which are defined as following:

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = \varepsilon_t, \tag{2.1}$$

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}, \tag{2.2}$$

where $\{X_t\}$ is the observation, $\varepsilon_t \sim N(0, \sigma^2)$, $\{\phi_1, \ldots, \phi_p\}$ and $\{\theta_1, \ldots, \theta_q\}$ are parameters in $AR(p)$ and $MA(q)$ models. For every t, if $\{X_t\}$ meets equation 2.1,

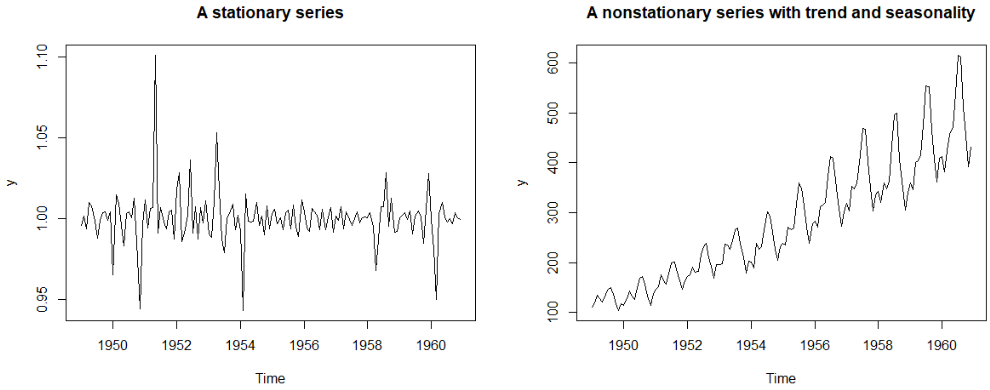


Figure 2.1: Examples of the stationary and nonstationary series

then we call it a *autoregressive model* of order p, denoted as $AR(p)$; if $\{X_t\}$ meets equation 2.2, then we call it a *moving-average model* of order q, denoted as $MA(q)$.

With the same notations, we define $\{X_t\}$ as an $ARMA(p, q)$ process if $\{X_t\}$ is stationary and for every t, $\{X_t\}$ satisfies

$$X_t - \sum_{i=1}^{p} \phi_i X_{t-i} = \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}, \tag{2.3}$$

where polynomials $1 - \phi_1 x - \phi_2 x^2 - \cdots - \phi_p x^p$ and $1 + \theta_1 x + \cdots + \theta_q x^q$ have no common factors. To simplify the formula, we re-write equation 2.3 with a backward operator B:

$$\phi(B)X_t = \theta(B)\varepsilon_t, \tag{2.4}$$

where $B^j X_t = X_{t-j}$, $\{\varepsilon_t\} \sim N(0, \sigma^2)$, $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ and $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_p B^q$. *Note:* the basic definitions such as *stationarity* can be found in Chapters 2 and 3 of [Brockwell and Davis, 2016]. The ARMA model requires the observation is stationary. If not, then we need to use ARIMA models, see Subsection 2.1.2.

In practice, p and q are not very large and we could often determine these coefficients by hand even without a computer. Thus it is easier to fit a stationary process with an ARMA model instead of a general SSM.

2.1.2 ARIMA and SARIMA

ARIMA is meant to solve the obvious problem with ARMA: most series are obviously not stationary. The idea behind ARIMA is that a differenced series $\{(1 - B)^d X_t\}$ would be approximately stationary, even if the original series $\{X_t\}$ isn't. ARIMA is the simplest model one can build on top of ARMA based on this observation. Thus in the ARIMA framework, we shall increase the difference times successively until the result seems to be stationary. We then model the differenced series with an ARMA model. Figure 2.2 shows two different types of nonstationary series.

Suppose $\{X_t\}$ is the observation, and we define $Y_t = (1 - B)^d X_t$, where d is a nonnegative integer. If $\{Y_t\}$ is a causal $ARMA(p, q)$ process, then $\{X_t\}$ is an $ARIMA(p, d, q)$ process (See Chapter 3 in [Brockwell and Davis, 2016] for the defi-

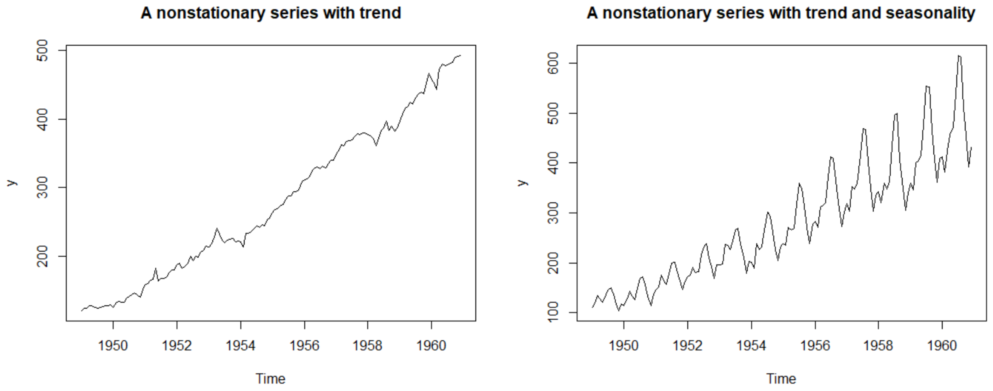


Figure 2.2: Examples of nonstationary series

nition of *causality*). Mathematically, an ARIMA(p,d,q) process $\{X_t\}$ satisfies

$$\phi(B)(1-B)^d X_t = \theta(B)\varepsilon_t, \tag{2.5}$$

where $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$, $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_p B^q$, p and q are nonnegative integers.

Sometimes, the fluctuation of datasets may present a seasonal pattern. For example, the monthly data may have peaks or troughs at the same month of each year, like the second graph in Figure 2.2. In these cases, we could use seasonal ARIMA (SARIMA) models , which could be viewed as an extension of ARIMA.

Suppose $\{X_t\}$ is the observation and $Y_t = (1-B)^d(1-B^s)^D X_t$, where d, D and s are nonnegative integers, and we define

$$\begin{aligned}
\phi(x) &= 1 - \phi_1 x - \phi_2 x^2 - \cdots - \phi_p x^p,\\
\Phi(x) &= 1 - \Phi_1 x - \Phi_2 x^2 - \cdots - \Phi_P x^P,\\
\theta(x) &= 1 - \theta_1 x - \theta_2 x^2 - \cdots - \theta_q x^q,\\
\Theta(x) &= 1 - \Theta_1 x - \Theta_2 x^2 - \cdots - \Theta_Q x^Q,
\end{aligned}$$

where p, q, P and Q are nonnegative integers, then if $\{Y_t\}$ is a causal ARMA process satisfying

$$\phi(B)\Phi(B^s)Y_t = \theta(B)\Theta(B^s)\varepsilon_t, \tag{2.6}$$

where $\{\varepsilon_t\} \sim N(0,\sigma^2)$, then we say $\{X_t\}$ is a $SARIMA(p,d,q) \times (P,D,Q)_s$ process with period s. In this paper, we choose $s = 12$ as the seasonal frequency when simulating datasets. We may note that s itself is determined by data not us in practice, and fixing s at 12 could be a bad choice for some cases, like the monthly data.

2.2 Techniques used for decomposition

In this section, we shall mainly talk about the techniques proposed to decompose a time series dataset. Typically, a time series $\{y_t\}$ is composed by

$$y_t = T_t + S_t + I_t, \qquad or \qquad y_t = T_t \cdot S_t \cdot I_t, \tag{2.7}$$

where $t = 1, \ldots, n$, and T_t, S_t, I_t are trend, seasonal and irregular components separately. If the model is multiplicative like some stock models, then we usually need to take a logarithm transformation if necessary.

To make full use of the properties of the stationary process, we need to eliminate the trend and seasonality. Generally speaking, there are two ways: (1) estimate the trend and seasonality and then extract them out to obtain the stationary process; (2) eliminate them by differencing the original series directly, which is exactly what we introduced in Subsection 2.1.2.

For classical decomposition methods *X-11* and *SEATS*, they both take the first choice. The difference is that X-11 use different *linear filters* to extract the trend and seasonal series (see Section 2.3) whereas SEATS assumes every component could be modelled by an ARIMA model. In another word, if we use SEATS to decompose a time series, we will build m ARIMA models for m latent variables and build another one for our observations, see equation 2.8. For the classical decomposition defined by equation 2.7, $m = 3$. More details of *SEATS* could be found in [Dagum and Bianconcini, 2016].

$$\begin{aligned} \varphi(B)y_t &= \vartheta(B)a_t, \\ \varphi_T(B)T_t &= \vartheta_T(B)b_t, \\ \varphi_S(B)S_t &= \vartheta_S(B)c_t, \\ \varphi_I(B)I_t &= \vartheta_I(B)d_t, \end{aligned} \tag{2.8}$$

where all φ and ϑ are polynomials of B and a_t, b_t, c_t and d_t are identically and independently distributed Gaussian noises with variances σ_a^2, σ_b^2, σ_c^2 and σ_d^2. See [Hillmer and Tiao, 1982] for derivation and analysis with regard to equation 2.8.

Except for X-11 and SEATS, there is also another decomposition method known as *STL* (Seasonal and Trend decomposition using Loess) developed by Cleveland, Cleveland, McRae and Terpenning. If readers are interested, please refer to [Cleveland et al., 1990].

On the other hand, the preprocessing methods *TRAMO* and *regARIMA* are both built on ARIMA models. Hence we can tell that the ARIMA model plays a very crucial role in classical methodologies for seasonal adjustment. Howerve, since this paper didn't apply any ARIMA-related methods, we won't explain more here. [Brockwell and Davis, 2016] has given a detailed introduction to ARIMA and relevant techniques.

2.3 X-11 method

As the earliest method developed to solve seasonal adjustment problem, X-11 is still used as one of the most popular methods nowadays because of its good applicability and simplicity. In this section, we shall see the decomposition steps in X-11. [Ladiray and Quenneville, 2012] is a professional book to explain all procedures in X-11 and other relevant theories.

As what we mentioned in Chapter 1, the centre of X-11 is the *linear filter/moving average*. The general formula of the moving average is

$$\hat{X}_t = \sum_{i=-p}^{q} \theta_i X_{t+i}, \tag{2.9}$$

where $\hat{X}_t$ is the smoothed value at time t. If $p = q$, we call this filter is centred. And if $\theta_{-k} = \theta_k$ in a centred moving average, it is symmetric. For the simplest symmetric moving average with order $P = 2p + 1$, its weight at each point is $\frac{1}{P}$.

In X-11, we mainly use two moving averages: one is the composite moving average, which is a composite of the simplest moving averages, and another one is the Henderson moving average, which is derived by Robert Henderson for actuarial problems in 1916. For example, a 2 × 4 composite moving average at time t is

$$\hat{X}_t = \frac{1}{8}X_{t-2} + \frac{2}{8}X_{t-1} + \frac{2}{8}X_t + \frac{2}{8}X_{t+1} + \frac{1}{8}X_{t+2}.$$

For a Henderson moving average of order $2p + 1$, its weight is computed by

$$\theta_i = \frac{315[(n-1)^2 - i^2][n^2 - i^2][(n+1)^2 - i^2][3n^2 - 16 - 11i^2]}{8n(n^2-1)(4n^2-1)(4n^2-9)(4n^2-25)}, \tag{2.10}$$

where $i = -p, \ldots, 0, \ldots, p$ and $n = p + 2$.

Now, suppose our data is additively composed by the trend, seasonal and irregular series, that is

$$X_t = T_t + S_t + I_t, \tag{2.11}$$

where $t = 1, \ldots, T$. We further assume our observation $\{X_t\}$ has been preprocessed and is monthly data. Then there are 10 main steps in the decomposition procedure:

(i) Estimation of the initial trend by 2 × 12 MA:

$$T_t^{(1)} = \frac{1}{24}X_{t-6} + \frac{1}{12}X_{t-5} + \cdots + \frac{1}{12}X_{t+5} + \frac{1}{24}X_{t+6}; \tag{2.12}$$

(ii) Estimation of the initial seasonal-irregular component:

$$SI_t^{(1)} = X_t - T_t^{(1)}; \tag{2.13}$$

(iii) Estimation of the initial seasonal component by 3 × 3 seasonal moving average:

$$\hat{S}_t^{(1)} = \frac{1}{9}SI_{t-24}^{(1)} + \frac{2}{9}SI_{t-12}^{(1)} + \frac{3}{9}SI_t^{(1)} + \frac{2}{9}SI_{t+12}^{(1)} + \frac{1}{9}SI_{t+24}^{(1)}, \tag{2.14}$$

then using a 2 × 12 moving average to center it:

$$S_t^{(1)} = \hat{S}_t^{(1)} - (\frac{1}{24}\hat{S}_{t-6}^{(1)} + \frac{1}{12}\hat{S}_{t-5}^{(1)} + \cdots + \frac{1}{12}\hat{S}_{t+5}^{(1)} + \frac{1}{24}\hat{S}_{t+6})^{(1)}; \tag{2.15}$$

(iv) Estimation of the initial seasonally adjusted series:

$$SA_t^{(1)} = X_t - S_t^{(1)}; \tag{2.16}$$

(v) Estimation of the intermediate trend by $2H + 1$-term Henderson moving average:

$$T_t^{(2)} = \sum_{j=-H}^{H} h_j^{(2H+1)} SA_t^{(1)}, \tag{2.17}$$

where h_j are weights of $2H + 1$ Henderson MA, and H is determined by users and data;

(vi) Estimation of the intermediate seasonal-irregular component:

$$SA_t^{(2)} = X_t - T_t^{(2)}; \tag{2.18}$$

(vii) Estimation of the seasonal component by 3 × 5 seasonal moving average and centred by 2 × 12 moving average again:

$$\begin{aligned}\hat{S}_t^{(2)} &= \frac{1}{15}SI_{t-36}^{(2)} + \frac{2}{15}SI_{t-24}^{(2)} + \frac{3}{15}SI_{t-12}^{(2)} \\ &+ \frac{3}{15}SI_t^{(2)} + \frac{3}{15}SI_{t+12}^{(2)} + \frac{2}{15}SI_{t+24}^{(2)} + \frac{1}{15}SI_{t+36}^{(2)}, \quad (2.19)\\ S_t^{(2)} &= \hat{S}_t^{(2)} - (\frac{1}{24}\hat{S}_{t-6}^{(2)} + \frac{1}{12}\hat{S}_{t-5}^{(2)} + \cdots + \frac{1}{12}\hat{S}_{t+5}^{(2)} + \frac{1}{24}\hat{S}_{t+6}^{(2)}), \quad (2.20)\end{aligned}$$

where $S_t^{(2)}$ is the seasonal series we obtained from X-11;

(viii) Estimation of the seasonally adjusted series again:

$$SA_t^{(2)} = X_t - S_t^{(2)}, \tag{2.21}$$

which is the seasonally adjusted series from X-11;

(ix) Estimation of the trend series by a $2H' + 1$-term Henderson moving average, where H' is still not fixed, and this output is the final trend series:

$$T_t^{(2)} = \sum_{j=-H'}^{H'} h_j^{(2H'+1)} SA_t^{(2)}; \tag{2.22}$$

(x) Estimation of the irregular series:

$$I_t = SA_t^{(2)} - T_t^{(2)}. \tag{2.23}$$

Steps (i)-(x) above are the main procedure used for decomposition in X-11. Although X-11 is the first method brought up to do seasonal adjustment, given its simplicity and good applicability, people still use it in many statistic agencies. Nonetheless, we should be aware of that X-11 works bad when there are prominent outliers in the data because they will influence our inference a lot, and this is the reason why we assume the dataset has been preprocessed at the beginning. Similar to the missing values. Meanwhile, because of the characteristic of symmetric moving averages and the fact that we don't know the data before the first one and after the last one, X-11 may not have good results around the beginning and the end. The common treatment was to let the observation with negative indices to be 0 in old versions. But this drawback has been fixed after combining it with ARIMA models, because we can fit data with an ARIMA model beforehand and then use this ARIMA model to backcast and forecast the series.

Chapter 3

State space modelling and the Kalman filter

In Chapter 1, we have said R.E. Kalman brought up this statement and the famous algorithm, the Kalman filter in [Kalman, 1960]. The state space model, also known as the hidden Markov model sometimes (see [Rabiner, 1989]), is a powerful modelling method and applied widely in engineering, statistics, economics, etc. We shall introduce this model and explain it with some examples in Section 3.1. And in Section 3.2, we will show its generality by giving three common models that have their state space forms. Although there are many methods to extract hidden states from our observations, the Kalman filter is the most widely used one. Section 3.3 shows the theory of how the Kalman filter works given the structural state space model, which is also what we used in our research. Developed from the Kalman filter, the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) work on nonlinear systems. Meanwhile, the particle filter is a popular Monte Carlo method for SSMs (See Chapter 14 in [Robert and Casella, 2013b]). Section 3.4 will give a detailed comparison of SSMs and ARIMA models to explain why we would like to study SSMs.

3.1 Introduction to state space modelling

State space modelling was first proposed to solve the problems in the area of control theory in 1960s. Then in 1980s and 1990s, with the gradual development of related theories, this model became more and more popular.

For a state space model, the observation is usually composed by one or more components, which is called the *state* in SSMs. For each state space model, both of the observation and the state could be multivariate or univariate. But in practice, at least in seasonal adjustment, we usually deal with cases in which the observation is univariate and the state space is multivariate. If states in a SSM are unobserved and each state is a Markov process, then we call this SSM a hidden Markov model, although some people seem to use SSM and HMM as synonyms. In general cases, what we know about the whole system includes the observations, the relation between observation and states, and the pattern how each state updates.

Figure 3.1 illustrates the HMM vividly (see lecture 19, [Protopapas, 2014]). In this figure, $y_{0:T}$ is the observation and $x_{0:T}$ is our hidden state, which behaves as a Markov chain, that is, the current state only depends on the last state. Suppose T_t

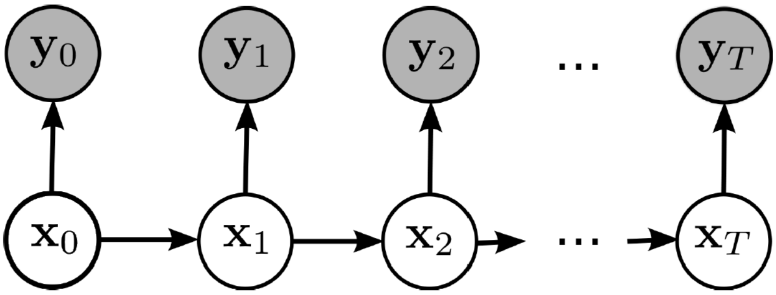


Figure 3.1: State space models

is the transition matrix of the Markov process from time t to $t+1$, then a general discrete hidden Markov model is

$$\begin{aligned} y_t &\sim f(y|X_t, \epsilon_t), \qquad \epsilon \sim (0, H_t), \\ X_{t+1} &= T_t X_t + R_t \eta_t, \qquad \eta_t \sim (0, Q_t), \end{aligned} \tag{3.1}$$

where f is a generic function of state X_t and noise ϵ_t.

By specializing the generic function f and the general noises ϵ_t and η_t to a matrice Z_t and additive Gaussian noises separately, we can obtain the linear gaussian state space model:

$$y_t = Z_t X_t + \epsilon_t, \qquad \epsilon_t \sim N(0, H_t), \tag{3.2}$$

$$X_{t+1} = T_t X_t + R_t \eta_t, \qquad \eta_t \sim N(0, Q_t), \tag{3.3}$$

where $t = 1, ..., n$, and $X_1 \sim N(a_1, P_1)$. Equation 3.2 is called the *measurement equation* and equation 3.3 is called the *transition equation*. Suppose the dimension of our observation is $p \times 1$ and the state is $m \times 1$, then dimensions of above matrices are given in the table 3.1.

Vector	Dimension	Matrix	Dimension
y_t	$p \times 1$	Z_t	$p \times m$
X_t	$m \times 1$	T_t	$m \times m$
ϵ_t	$p \times 1$	R_t	$m \times r$
η_t	$r \times 1$	H_t	$p \times p$
		Q_t	$r \times r$

Table 3.1: Dimensions of notations

Example 3.1.1. In Section 2.1, we have talked about ARIMA models. Here we will show how to transform a $AR(2)$ model to a state space form at first and then introduce the general state space forms for $AR(p)$ models.

Suppose our $AR(2)$ model is

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \epsilon_t, \tag{3.4}$$

where $\epsilon_t \sim N(0, \sigma^2)$, then we can find a new observation is related to the previous two values, therefore when defining this state space model, the transition equation 3.3 should have at least two states to achieve iterations.

Based on this, we will get the following result

$$y_t = \begin{bmatrix}1 & 0\end{bmatrix} x_t, \tag{3.5}$$

$$x_t = \begin{bmatrix}\phi_1 & \phi_2 \\ 1 & 0\end{bmatrix} x_{t-1} + \omega_t, \tag{3.6}$$

where $x_t = \begin{bmatrix}y_t & y_{t-1}\end{bmatrix}^T$ and $\omega_t = \begin{bmatrix}\epsilon_t & 0\end{bmatrix}^T$.

More generally, suppose our model is $AR(p)$, that is

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t, \tag{3.7}$$

where $\epsilon_t \sim N(0, \sigma^2)$. Then the corresponding state space form is

$$y_t = \begin{bmatrix}1 & 0 & \dots & 0\end{bmatrix} x_t, \tag{3.8}$$

$$x_t = \begin{bmatrix}\phi_1 & \phi_2 & \dots & \phi_{p-1} & \phi_p \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0\end{bmatrix} x_{t-1} + \omega_t, \tag{3.9}$$

where $x_t = \begin{bmatrix}y_t & y_{t-1} & \dots & y_{t-p+1}\end{bmatrix}^T_{1\times p}$ and $\omega_t = \begin{bmatrix}\epsilon_t & 0 & \dots & 0\end{bmatrix}^T_{1\times p}$.

3.2 Common state space models

In this section, we shall briefly talk about three common models that can be transformed into state space forms, which are structural time series models, ARIMA models and regression models respectively. Because of these three models, we can tell the generality of SSMs. Durbin and Koopman have detailedly showed them in [Durbin and Koopman, 2012].

3.2.1 Structural time series models

The structural time series model is one of the most frequent types we used in SSM due to its structural characteristic. For one time series data, if we treat it as a combination of the trend, seasonal and irregular components, then we call it a structural time series model, which usually have two modes:

$$y_t = T_t + S_t + I_t, \tag{3.10}$$

$$y_t = T_t \cdot S_t \cdot I_t, \tag{3.11}$$

where T_t, S_t and I_t stand for the *trend*, *seasonal* and *irregular* components respectively. The series without seasonal part is called *seasonally adjusted series*. If the relation is multiplicative, then we usually take the logarithm transformation to put it into a linear SSM. Mathematically, we shall use $log(y_t) = T_t^* + S_t^* + I_t^*$ instead of equation 3.11. Generally speaking, if the fluctuation within each year become greater as time goes on, then this model is multiplicative, like the data we showed in Figure 2.2.

The simplest structural time series model is the *local level model*, where we do not have any seasonal or other explanatory variables:

$$\begin{aligned} y_t &= T_t + \varepsilon_t, \\ T_{t+1} &= T_t + \eta_t, \end{aligned} \tag{3.12}$$

where $\varepsilon_t \sim N(0, \sigma_y^2)$ and $\eta_t \sim N(0, \sigma_T^2)$. If we add a slope to the trend component, we will obtain the *local linear trend model*:

$$\begin{aligned} y_t &= T_t + \varepsilon_t, \\ T_{t+1} &= T_t + v_t + \eta_t, \\ v_{t+1} &= v_t + \zeta_t. \end{aligned} \tag{3.13}$$

As for the seasonal component, we usually suppose the sum of its influence over one period is around zero, thus one simple way to model it is:

$$S_{t+1} = -\sum_{j=1}^{s-1} S_{t+1-j} + \omega_t, \tag{3.14}$$

where $\omega_t \sim N(0, \sigma_S^2)$ and s is the seasonal frequency of our data, that is, for weekly and monthly data, $s = 7$ and 12 separately. But sometimes people prefer to use the trigonometric form to express seasonal components (see [Young et al., 1991] for details):

$$\begin{aligned} S_t &= \sum_{j=1}^{[s/2]} (\tilde{S}_{jt} cos\lambda_j t + \tilde{S}_{jt}^* sin\lambda_j t), \\ \tilde{S}_{j,t+1} &= \tilde{S}_{jt} + \tilde{\omega}_{jt}, \\ \tilde{S}_{j,t+1}^* &= \tilde{S}_{jt}^* + \tilde{\omega}_{jt}^*, \end{aligned} \tag{3.15}$$

where $\lambda_j = \frac{2\pi j}{s}$, $j = 1, \dots, [s/2]$ and $\tilde{\omega}_{jt}$, $\tilde{\omega}_{jt}^* \sim N(0, \sigma_\omega^2)$.

And the irregular component in equation 3.10 is generally treated as a normally distributed noise directly with mean 0.

Therefore, if we combine the local linear trend model 3.13 with the seasonal equation 3.14, then we could obtain the following state space form (in Chapter 4, we shall use a similar but easier model composed by the local level model and the seasonal equation 3.14, see equation 4.2):

$$\begin{aligned} y_t &= T_t + S_t + I_t, \\ T_{t+1} &= T_t + v_t + \eta_t, \\ v_{t+1} &= v_t + \zeta_t, \\ S_{t+1} &= -\sum_{j=1}^{s-1} S_{t+1-j} + \omega_t. \end{aligned} \tag{3.16}$$

If we transform it into the general state space form 3.2 and 3.3, then all the notations are defined as following:

$$\begin{aligned}
X_t &= \begin{bmatrix} T_t & v_t & S_t & S_{t-1} & \cdots & S_{t-s+2} \end{bmatrix}^T, \\
Z_t &= \begin{bmatrix} Z_{[T]} & Z_{[S]} \end{bmatrix}, \\
T_t &= diag \begin{bmatrix} T_{[T]} & T_{[S]} \end{bmatrix}, \\
R_t &= diag \begin{bmatrix} R_{[T]} & R_{[S]} \end{bmatrix}, \\
Q_t &= diag \begin{bmatrix} Q_{[T]} & Q_{[S]} \end{bmatrix},
\end{aligned} \tag{3.17}$$

where *diag* means the diagonal matrix and

$$Z_{[T]} = [1 \quad 0], \qquad Z_{[S]} = [1 \quad 0 \quad \cdots \quad 0],$$

$$T_{[T]} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad T_{[S]} = \begin{bmatrix} -1 & -1 & \cdots & -1 & -1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix},$$

$$R_{[T]} = I_2, \qquad R_{[S]} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix},$$

$$Q_{[T]} = \begin{bmatrix} \sigma_\eta^2 & 0 \\ 0 & \sigma_\zeta^2 \end{bmatrix}, \qquad Q_{[S]} = \sigma_\omega^2.$$

3.2.2 ARIMA models

We have introduced the ARIMA model in Section 2.1 and showed how to switch an AR(2) model into the state space form in Section 3.1. In this subsection, we will show how to transform an arbitrary ARIMA model into a state space form.

Let's look at how to transform an ARMA model at first. Suppose we now have an ARMA(p,q) model:

$$\begin{aligned}
y_t &= \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \\
&= \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \\
&= \sum_{i=1}^{r} \phi_i y_{t-i} + \varepsilon_t + \sum_{j=1}^{r-1} \theta_j \varepsilon_{t-j},
\end{aligned} \tag{3.18}$$

where $r = max(p, q+1)$, $\varepsilon_t \sim N(0, \sigma_\varepsilon^2)$ and all the parameters are known. To transform it into the state space form, we define the measurement equation as

$$\begin{aligned}
y_t &= \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} x_t, \\
x_t &= \begin{pmatrix} y_t \\ \phi_2 y_{t-1} + \cdots + \phi_r y_{t-r+1} + \theta_1 \varepsilon_t + \cdots + \theta_{r-1} \varepsilon_{t-r+2} \\ \phi_3 y_{t-1} + \cdots + \phi_r y_{t-r+2} + \theta_2 \varepsilon_t + \cdots + \theta_{r-1} \varepsilon_{t-r+3} \\ \vdots \\ \phi_r y_{t-1} + \theta_{r-1} \varepsilon_t \end{pmatrix}.
\end{aligned} \tag{3.19}$$

And the notations in the transition equation are:

$$T_t = T = \begin{bmatrix} \phi_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \phi_{r-1} & 0 & \cdots & 1 \\ \phi_r & 0 & \cdots & 0 \end{bmatrix}, \qquad R_t = R = \begin{pmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{r-1} \end{pmatrix}, \qquad \eta_t \sim N(0, \sigma_\varepsilon^2). \quad (3.20)$$

By 3.19 and 3.20, we have the capacity to transform every known ARMA model to a corresponding state space model. Similarly, we could put any ARIMA model into a SSM, see Section 3.4 in [Durbin and Koopman, 2012].

Therefore, mathematically we are able to transform every known ARIMA and ARMA model into a state space form, which confirms that state space modelling is a more general and practical method. On the other hand, many but not all state space models have their corresponding ARIMA models unless we take some transformation like the logarithm. One simplest instance is when our model is multiplicative, then people need to take logarithm before fitting it with an ARIMA model, while in SSM, we don't need to do that. Example 3.2.1 shows one simple successful case. More related work could be referred to [Harvey, 1990].

Example 3.2.1. In the local linear trend model 3.13, if we take two difference of observations, we shall get

$$\Delta^2 y_t = \varepsilon_t - 2\varepsilon_{t-1} + \varepsilon_{t-2} + \eta_{t-1} - \eta_{t-2} + \zeta_{t-2}.$$

It is not hard to notice only the first two autocorrelations are nonzero, so we can use a *MA(2)* model to express the right hand side equivalently, that is

$$\Delta^2 y_t = \delta_t + \theta_1^* \delta_{t-1} + \theta_2^* \delta_{t-2},$$

which is the expression of one *ARIMA(0,2,2)* model.

We have been aware of the relation between ARIMA modelling and state space modelling. In the example above, although we did transform the local linear trend model to an ARIMA model, the information with regard to the slope v_t and the level(trend) T_t is lost in this process. And this is one reason why we would like to apply the structural time series SSM in our research instead of the ARIMA model-based methods.

3.2.3 Regression models

As one of the most fundamental concept in statistics, regression models have been studied for a long time. If we could put a regression model into a SSM, then that means we could use the algorithms like the Kalman filter to solve a regression problem. Here, we will show how to transform a linear regression to a linear state space model.

If we consider the *measurement equation* ignoring the subscript t, it is exactly a regression model, which means we could perhaps view a linear regression model as a SSM. Suppose we have a simple regression model for a univariate variable y:

$$y = X\beta + \varepsilon, \qquad \textit{where} \quad \varepsilon \sim N(0, H), \quad (3.21)$$

corresponding to equations 3.2 and 3.3, we let:

$$Z_t = X_t, \qquad T_t = I_t, \qquad R_t = Q_t = 0, \tag{3.22}$$

where $t = 1, 2, \ldots, n$ and n is the number of measurements. If the coefficient is changeable regarding time, then we perhaps need to modify T_t and R_t. For example, if each element in β_t follows a random walk, then it is the multivariate version of the transition equation in 3.12, that is

$$T_t = R_t = I_t, \qquad Q_t = \Sigma_t, \tag{3.23}$$

where Σ_t is the diagonal variance matrix of coefficients.

For regression problems, one important part is to determine the coefficients, which is the *state* in its state space form. And in SSMs, our purpose is to calculate the states. From this perspective, we can use the techniques in SSM to solve a regression problem easily.

3.3 The Kalman filter

We have talked the state space model and its classifications in Sections 3.1 and 3.2, but haven't introduced the algorithms used to solve it so far. In this section, we shall give one of the commonest algorithms, the Kalman filter, which is designed to extract the latent states from linear Gaussian SSMs.

Suppose we have a state space model,

$$\begin{aligned} y_t &= Z_t \alpha_t + \epsilon_t, \qquad & \epsilon_t &\sim (0, H_t), \\ \alpha_{t+1} &= T_t \alpha_t + R_t \eta_t, \qquad & \eta_t &\sim (0, Q_t). \end{aligned}$$

then as we know our final purpose is to estimate/filter the latent state α_t. To measure the accuracy of the estimate, one common error is the *mean squared error* (MSE), which is also what we used in the Kalman filter. In another word, the filtered result from the Kalman filter is the MMSE estimate. Meanwhile, the MSE plays a crucial role in the log-likelihood function. If y_t is Gaussian, then the MLE is also the MMSE estimator. See [Thacker and Lacey, 1998].

The Kalman filter itself is a recursive algorithm mainly composed by prediction and update two steps. In the prediction step, people predict the state of time $t + 1$ only under the state estimate at time t and observed information at time t. When a new observation at time $t + 1$ is available, we update the previous prediction based on it. On the other hand, when talking about the Kalman filter, we usually append a smoother after it. As we just pointed out, the Kalman filter only consider the observation $y_{1:t}$ when estimating the state α_t, whereas we know the other observation could also help us adjust our estimate, and with the smoothers that we shall talk about in Subsection 3.3.2, we could adjust our filtered result and make them smoother and more convincing. Besides these, the filtering process is a forward algorithm but the smoothing process is a backward algorithm. We will expand both algorithms in Subsections 3.3.1 and 3.3.2, more details could be referred to Appendix A and Chapter 4 of [Durbin and Koopman, 2012].

3.3.1 Filtering process

Before showing the filtering algorithm, we need to introduce some new notations to simplify our writing. Given a linear Gaussian state space model,

$$\begin{aligned} y_t &= Z_t\alpha_t + \epsilon_t, & \epsilon_t &\sim N(0, H_t), \\ \alpha_{t+1} &= T_t\alpha_t + R_t\eta_t, & \eta_t &\sim N(0, Q_t). \end{aligned}$$

We denote $Y_t = y_{1:t}$ and define:

$$\begin{aligned} a_{t|t} &= E(\alpha_t|Y_t), & a_{t+1} &= E(\alpha_{t+1}|Y_t), \\ P_{t|t} &= Var(\alpha_t|Y_t), & P_{t+1} &= Var(\alpha_{t+1}|Y_t), \\ v_t &= y_t - Z_t a_t, & F_t &= Var(v_t|Y_{t-1}). \end{aligned} \tag{3.24}$$

The filtering process is mainly composed by update and prediction two parts, which are as following:

Algorithm 1 Filtering process

```
Require: $\alpha_1 \sim N(a_1, P_1)$,   $Y_T$ and matrices $Z, H, T, R, Q$
  for $t \leftarrow 1, \ldots, n$ do
    $v_t \leftarrow y_t - Z_t a_t$,
    $F_t \leftarrow Z_t P_t Z_t^T + H_t$,
    $K_t \leftarrow T_t P_t Z_t^T F_t^{-1}$,
    UPDATE:
    $a_{t|t} \leftarrow a_t + P_t Z_t^T F_t^{-1} v_t$,
    $P_{t|t} \leftarrow P_t - P_t Z_t^T F_t^{-1} Z_t P_t$,
    PREDICT:
    $a_{t+1} \leftarrow T_t a_t + K_t v_t$,
    $P_{t+1} \leftarrow T_t P_t (T_t - K_t Z_t)^T + R_t Q_t R_t^T$.
  end for
```

Usually we do not know the initial distribution $N(a_1, P_1)$, to solve this problem, S.J. Koopman and J. Durbin have presented a diffuse method in [Koopman and Durbin, 2003] and more discussion could be found in Chapter 5 of [Durbin and Koopman, 2012]. In practice, matrices Z_t, T_t, H_t, R_t and Q_t are time-invariant sometimes. For these cases, the following Lemma 3.3.1 could save us considerable computation time, because it tells us that we don't need to compute the matrices P_t, $P_{t|t}$, K_t and F_t in a time-invariant SSM once P_t achieves convergence.

Lemma 3.3.1. If one state space model is time-invariant, that is, Z_t, T_t, H_t, R_t and Q_t are fixed, then the variance matrix P_t converges to the solution of the equation:

$$\bar{P} = T\bar{P}T' - T\bar{P}Z'\bar{F}^{-1}Z\bar{P}T' + RQR', \tag{3.25}$$

where $\bar{F} = Z\bar{P}Z' + H$.

Proof. See [Anderson and Moore, 2012]. □

3.3.2 Smoothing process

We have introduced the filtering process in Subsection 3.3.1, now let's look at the smoothing process. The key point in the state smoothing procedure is to compute $\widehat{\alpha}_t = E(\alpha_t|Y_n)$ for $t = 1, \ldots, n$, where $Y_t = y_{1:t}$. There are many different smoothers including: the fixed-interval smoother $E(\alpha_t|y_t, \ldots, y_s)$ on the interval $Y_{t:s}$; the fixed-point smoother $E(\alpha_t|Y_s)$ for $s = t+1, t+2, \ldots$; and the fixed-lag smoother $E(\alpha_{s-j}|Y_s)$ for a fixed positive j and $s = j+1, j+2, \ldots$. In this paper, we shall only use the fixed-interval smoother over the whole observation, which is also called the *Kalman smoother* and is the choice recommended by S.J. Koopman and J. Durbin. The smoothing algorithm is mainly as following:

Algorithm 2 Smoothing process

```
Require: $r_n = 0$, $N_n = 0$ and results from the filtering process
  for $t \leftarrow n, \ldots, 1$ do
    $L_t \leftarrow T_t - K_t Z_t$,
    $r_{t-1} \leftarrow Z_t' F_t^{-1} V_t + L_t' r_t$,
    $N_{t-1} \leftarrow Z_t' F_t^{-1} Z_t + L_t' N_t L_t$,
    $\widehat{\alpha}_t \leftarrow a_t + P_t r_{t-1}$,
    $V_t \leftarrow P_t - P_t N_{t-1} P_t$.
  end for
```

Here, we introduced several notations including L_t, r_t and N_t in this process and needed the matrices such as K_t and F_t generated in the filtering process. And different from the filtering process, the smoothing process is a backwards algorithm. The final results $\widehat{\alpha}_t$ and V_t are the smoothed states and their variances. The smoothing process is built on the Kalman filter, and optimize one specific state estimate with the observation after it.

3.4 Advantages of SSMs over ARIMA models

In Chapter 1, we have briefly talked about the advantages of SSMs. In this section, we shall give a detailed comparison of some characteristics of SSMs and ARIMA models. In particular, we mainly talk about the advantages of SSMs over ARIMA models to emphasize the reason why we explore the SSM to solve the seasonal adjustment problem.

Firstly, the usage of state space models will benefit us to apply the *hierarchical/multilevel model* and *partial pooling*, one of the most useful points accomplished by hierarchical models. A general interpretation of partial pooling is we shall consider the information from other similar datasets when analyzing the current dataset. Back to the seasonal adjustment problem, we often meet different datasets from the same class and these datasets usually share the similar trend and seasonal patterns. For example, if you look at ice cream sales in a city for several brands over time, you would expect the time series to be related - all of them might have the same broad peak in the summer, and they would also likely share more local peaks and dips caused by hot days or thunderstorms. Fitting these datasets independently with X-11 won't help us to catch this dependence. But with hierarchical model, we can easily achieve it.

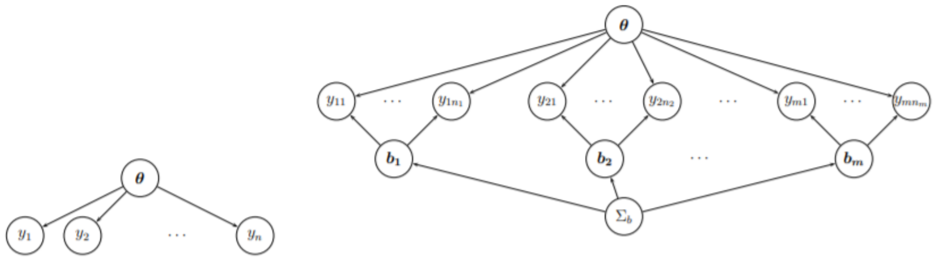
Before giving details of hierarchical SSMs regarding seasonal adjustment, we introduce one classical application of hierarchical models, the radon level estimation problem ([Gelman, 2006]). A simple hierarchical model of this problem is

$$\begin{aligned} y_{ij} &\sim N(\alpha_j, \sigma_y^2), \\ \alpha_j &\sim N(\mu_\alpha, \sigma_\alpha^2), \end{aligned} \tag{3.26}$$

where $j = 1, \dots, J$, $i = 1, \dots, n_j$ and y_{ij} is the logarithm of radon levels in the i^{th} house of the j^{th} county. Here, μ_α and σ_α^2 are hyperparameters that control the average radon level of each county, and σ_y^2 is the common parameter θ in Figure 3.2. Then we could compute the estimators of α_j, σ_y^2, μ_α and σ_α^2 by maximizing

$$p(Y|\mu_\alpha, \sigma_y^2, \sigma_\alpha^2) = (\prod_{j=1}^{J} f(\alpha_j|\mu_\alpha, \sigma_\alpha^2))(\prod_{i=1}^{n_j} f(y_{ij}|\alpha_j, \sigma_y^2)). \tag{3.27}$$

Equation 3.27 indicates that when estimating the average radon level α_j of the county j, we also consider the information of other counties, which will help us deal with the *short/inadequate* datasets better, because we usually cannot make convincing inference with a few points. This is called *partially pooling*. Opposite to partial pooling, if we let $\sigma_\alpha^2 = 0$, then we call it *complete pooling* because the average radon levels for all the counties are the same, which means we put all data together without distinction. If we let $\sigma_\alpha^2 = \infty$, then we call it *no-pooling*, because the average radon levels are uncorrelated to each other, that is to say we build models for each county separately. Figure 3.2 (from Chapter 8, [Levy, 2012]) illustrates the hierarchical model vividly with the comparison of the non-hierarchical model.



(a) A non-hierarchical model (b) A simple hierarchical model, in which observations are grouped into m clusters

Figure 3.2: Non-hierarchical and hierarchical models

Back to the hierarchical SSMs for seasonal adjustment, when we have a bunch of similar and related datasets, we could write down two different hierarchical models regarding the variance and mean respectively. Generally speaking, for the type of variances, we usually believe the fluctuation of these datasets and their components should be related but they could be at different levels; for the type of means, we are supposing that the components of these datasets are around some identical latent series but with different fluctuations.

Now for the hierarchical model regarding variances, suppose we use SSMs defined in equation 4.2, and treat $\sigma = (\sigma_I, \sigma_T, \sigma_S)$ within each SSM as a sample from another distribution such as $N(\mu_\sigma, \Sigma_\sigma)$. Then we could put them together and apply partial pooling easily. Mathematically, our hierarchical model in this case is

$$\begin{cases} y_{jt} \sim N(T_{jt} + S_{jt}, \sigma_{jI}^2), \\ T_{jt} \sim N(T_{j,t-1}, \sigma_{jT}^2), \\ S_{jt} \sim N(-\sum_{i=1}^{s-1} S_{j,t-i}, \sigma_{jS}^2), \end{cases} \qquad \begin{cases} \sigma_{jI} \sim N(\mu_I, \lambda_I^2), \\ \sigma_{jT} \sim N(\mu_T, \lambda_T^2), \\ \sigma_{jS} \sim N(\mu_S, \lambda_S^2), \end{cases} \tag{3.28}$$

where $t = 1, \ldots, n_j$, $j = 1, \ldots, J$ means the j^{th} dataset and for all datasets, the standard deviations of the same component from those datasets follow the same distribution. For instance, for a random j^{th} dataset, the standard deviation of the trend σ_{jT} follows $N(\mu_T, \lambda_T^2)$. Then we could use samples $\{y_{1t}\}, \ldots, \{y_{Jt}\}$ to estimate these parameters including μ_I, μ_T, μ_S, λ_I, etc.

On the other hand, instead of working on the variances, we could build another hierarchical model regarding the mean, like the example we gave in equation 3.26:

$$\begin{cases} y_{jt} \sim N(T_{jt} + S_{jt}, \sigma_{jI}^2), \\ T_{jt} \sim N(T_t^*, \sigma_{jT}^2), \\ S_{jt} \sim N(S_t^*, \sigma_{jS}^2), \end{cases} \qquad \begin{cases} T_t^* \sim N(T_{t-1}^*, \sigma_{T^*}^2), \\ S_t^* \sim N(-\sum_{i=1}^{s-1} S_{t-i}^*, \sigma_{S^*}^2), \end{cases} \tag{3.29}$$

where we suppose the trend and seasonal components in each dataset are around another two latent series $\{T_t^*\}$ and $\{S_t^*\}$ but with their own variances, and these two latent series obey the formulas we defined for the trend and seasonality. Compared with the last hierarchical model 3.28, this one requires less computation. We shall talk more about its application in Chapter 5.

However, due to the difference of structural features, the ARIMA models can't do the partial pooling as easily as with the state space model and the Kalman filter. This is one important reason that motivates us choose the SSM as well as a worthy underlying improvement orientation.

Secondly, state space models have easily-understandable structure over ARIMA models. From Subsection 3.2.1, we know a time series data is usually decomposed into three components, and the formula for each component is consistent with our prior knowledge. More generally, we use our prior knowledge to choose and define unobserved processes with reasonable models. On the contrary, from descriptions in Sections 2.1 and 2.2, we can hardly tell where the components come from only according to the ARIMA formula, although the techniques are not very hard compared with the Kalman filter and its derivatives.

Thirdly, missing values won't cause serious problems in state space models. Different from X-11 or SEATS, the value of each time point is not very essential in the Kalman filter. In Subsection 3.3.1, we have seen there are two parts at each recursion - one is to update the prediction of the state α_t and its variance when the new observation y_t is available, and another one is to predict the state α_{t+1} and its variance before we input y_{t+1} to the system. Hence if we do *not* have one particular observation say y_t, then we could just skip the update step and use the prediction from the last step as the updated result.

At last but not least, as what we showed in Section 3.2, state space modelling is a more general modelling methodology compared with ARIMA modelling. Every ARIMA model could be transformed into a state space form but only part of SSMs have their ARIMA forms. In other words, ARIMA models only work for linear processes but SSMs can be used to denote nonlinear processes, which are very common in real problems such as target tracking. And there exist some good methodologies proposed to solve nonlinear SSMs. However, we need to pay the price for the complexity of the model. With the parameter increasing, the computational time is a considerable problem.

Chapter 4

Bayesian analysis

4.1 Introduction

To apply the Kalman filter over a SSM, we need to know all the parameters including Z_t, H_t, T_t, R_t and Q_t. But in practice, we usually don't know the variance matrix Q_t and H_t. One common estimator of them is the maximum likelihood estimator (abbreviated to MLE), but we find the decomposition result from it is not good under the assumption that the decomposition from X-11 is our standard, especially for the trend series. But we find some parameter values chosen manually behaves quite good, thus there is some problem happened with the parameter estimation. So how to fix it?

Because the essential of our problem is to find another appropriate parameter estimation while the MLE is broken, we *guess* the *Bayesian analysis* should be able to improve our estimation a lot if we have a *strong* prior. But how to find such a prior?

As we know, our purpose is to use SSMs to obtain the similar decomposition result compared with those from X-11 with regard to the same dataset. And X-11 has been used since the appearance of the seasonal adjustment problem, which could be viewed as a very reliable method and a combination of expert knowledge. Therefore, we shall build an informative prior upon the information supplied by X-11. In the Bayesian framework, this process is known as *prior elicitation* ([Albert et al., 2012]). On the other hand, we observed that SSMs behave better than X-11 for the prediction problem. Therefore, under the Bayesian analysis with an informative prior, we could *"merge"* the characteristics of X-11 with SSMs.

To be specific, Section 4.2 will introduce the deficiency of maximum likelihood estimators (abbreviated to MLEs) by comparing its decomposition results with the X-11's through a real instance. And Section 4.3 shows that we could force the SSM to generate satisfying result by minimizing some loss functions. Then we use Section 4.4 to explain how we can reduce the computation and illustrate our problem again through one simulated example, which is also used frequently in the following analysis. In Section 4.5, we will explain the intuition why we apply Bayesian inference and utilize a weakly-informative prior to compute posterior estimators and compare them with MLEs. Later in Section 4.6 we shall explain why we would like to use empirical Bayesian analysis and how to make use of the prior knowledge gained from Section 4.3 to build an empirical prior. Meanwhile, we will introduce the weight k of the priors, which is used to control the posterior estimate. Section 4.7 shows if we already have a good weight k_0 upon the current dataset, how we could adjust the weight k when facing datasets of different lengths. In Section 4.8, we shall finally compare all these estimators' decomposition and prediction accuracy together to verify our explanation of the empirical MAP estimators.

Before moving on, let's review the model and notations we used in our work. In Section 3.1 and Section 3.2, we have introduced the general expression of a state space model:

$$\begin{aligned} y_t &= Z_t X_t + \varepsilon_t \qquad \varepsilon_t \sim N(0, H_t), \\ X_{t+1} &= C_t X_t + R_t \eta_t \qquad \eta_t \sim N(0, Q_t), \end{aligned} \tag{4.1}$$

where $t = 1, \dots, n$, and $X_1 \sim N(a_1, P_1)$. *Note:* To avoid the confusion, we *re-denote* the *transition* matrix as C_t in model 4.1, because we need to use T_t to express the *trend* series. In this chapter, we let:

$$X_t = \begin{bmatrix} T_t & S_t & S_{t-1} & \cdots & S_{t-s+2} \end{bmatrix}', \qquad \varepsilon_t = I_t,$$

$$Z_t = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \end{bmatrix}, \qquad C_t = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & -1 & \cdots & -1 & -1 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix},$$

$$R_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}, \qquad Q_t = \begin{bmatrix} \sigma_T^2 & 0 \\ 0 & \sigma_S^2 \end{bmatrix}, \qquad H_t = \sigma_I^2,$$

where $t = 1, \dots, n$. Then we could derive the state space model applied in our research:

$$\begin{aligned} y_t &= T_t + S_t + I_t, \\ T_{t+1} &= T_t + \eta_t, \\ S_{t+1} &= -\sum_{j=1}^{s-1} S_{t+1-j} + \omega_t, \end{aligned} \tag{4.2}$$

where I_t, η_t and ω_t are independent and identically distributed gaussian noises with mean 0 and variances σ_I^2, σ_T^2, σ_S^2. In this chapter, we shall use $SSM(a, b, c)$ to express model 4.2 with given variances $\sigma_I^2 = a$, $\sigma_T^2 = b$, $\sigma_S^2 = c$ and σ^2 to express the variance combo $(\sigma_I^2, \sigma_T^2, \sigma_S^2)$.

4.2 Behaviour of maximum likelihood estimators

In Chapter 3 and Section 4.1, we have noticed that there are some parameters in the state space model such as σ_I^2, σ_T^2 and σ_S^2 in model 4.2 and the variance matrix H_t and Q_t in model 4.1. According to the algorithms showed in Section 3.3, the Kalman filter only works when these parameters are known, so we need some approaches to infer them if they are not given. In this section, we shall show the common estimator, the maximum likelihood estimate, behaves bad in SSMs when used for decomposition in some cases, see Figures 4.1 and 4.2. As we will see, the fluctuation of the trend series is too large sometimes, which is not consistent with people's prior understanding of the trend, that is, the trend component should be smooth relatively. Mathematically speaking, the variance of the differenced series $\{T_{t+1}-T_t\}$ from SSM 4.2 is too large. More illustration can be found after Figure 4.2.

Now let's look at the log-likelihood expression. Suppose we have a general state space model:

$$\begin{aligned} y_t &= Z_t\alpha_t + \epsilon_t, \qquad & \epsilon_t &\sim N(0, H_t), \\ \alpha_{t+1} &= T_t\alpha_t + R_t\eta_t, \qquad & \eta_t &\sim N(0, Q_t), \end{aligned}$$

where $t = 1, \ldots, n$, and $\alpha_1 \sim N(a_1, P_1)$, the log-likelihood of $\{y_1, \ldots, y_n\}$ given $\theta = (\{Z_t\}, \{H_t\}, \{T_t\}, \{R_t\}, \{Q_t\})$ is

$$\ell(\theta) = log(p(y_1, y_2, ...y_n|\theta)) = log(p(y_1|\theta)\prod_{t=2}^{n} p(y_t|Y_{t-1}, \theta)) = \sum_{t=1}^{n} log(p(y_t|Y_{t-1}, \theta)), \tag{4.3}$$

where $Y_t = y_1, \cdots, y_t$ and $p(y_1|Y_0, \theta) = p(y_1)$. In Section 3.3 and Appendix A, we could obtain $y_t|Y_{t-1}, \theta \sim N(Z_t a_t, F_t)$ and $v_t = y_t - Z_t a_t$, where a_t is the prediction of states at time t and $F_t = Var(y_t|Y_{t-1}, \theta) = Var(v_t|Y_{t-1}, \theta) = Z_t P_t Z_t^T + H_t$, thus equation 4.3 could be written as

$$\ell(\theta) = -\frac{np}{2} log2\pi - \frac{1}{2}\sum_{t=1}^{n}(log|F_t| + v_t' F_t^{-1} v_t), \tag{4.4}$$

where p is the dimension of the state α_t. For a univariate problem, equation 4.3 would be

$$\ell(\theta) = -\frac{n}{2} log2\pi - \frac{1}{2}\sum_{t=1}^{n} log(F_t + v_t^2 F_t^{-1}). \tag{4.5}$$

As in most cases, the default estimator in SSMs is the value that maximize the log-likelihood 4.5. To compute the MLE, there are several methods introduced in Chapter 7, [Durbin and Koopman, 2012]. Here, we use the function *fitSSM* provided in package **KFAS** to compute the MLEs, which is wrapped by functions *optim* and *logLik*, see [Helske, 2016] for details. In practice, we usually do not know the variance matrices Q_t and H_t, and the others such as Z_t, T_t and R_t are usually known.

Let's take the unemployment data (by the thousand) of the United States from 1990 to 2016 as an example. Figure 4.1 is the comparison of decomposition results from X-11 and the SSM with MLEs. To have a closer look, we extracted the interval from 2000 to 2004, see Figure 4.2.

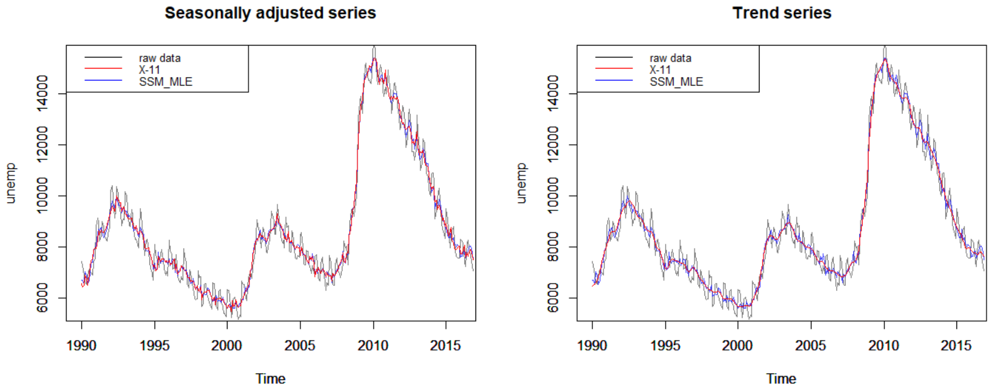


Figure 4.1: Decomposition comparison between X-11 and SSM(MLE)

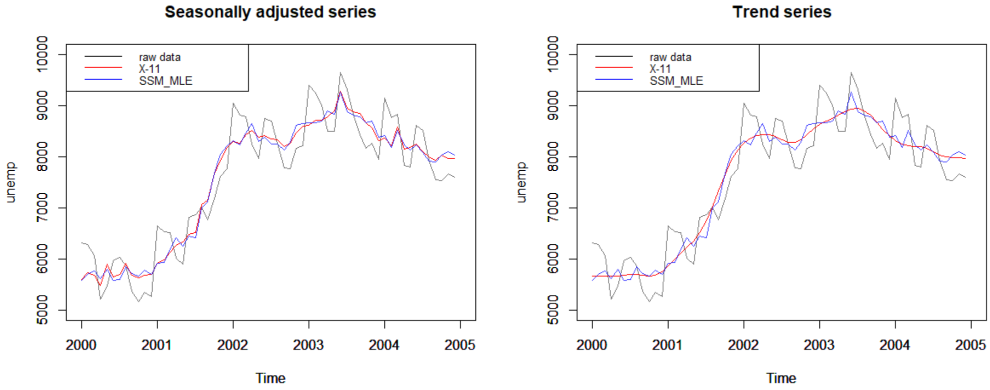


Figure 4.2: Decomposition comparison between X-11 and SSM(MLE) from 2000 to 2004

As we can see, the difference of the trend series from X-11 and SSM (MLE) is obvious, where the result from the SSM is much spikier. In economics, people would like to believe and see a relatively smooth trend instead of a spiky one, and the regular and irregular fluctuation should be mainly absorbed by the seasonal and irregular components separately. Meanwhile, when analysing one time series dataset, people usually care more about the seasonally adjusted and trend series. If the dataset is highly volatile, the seasonally adjusted series may not be enough to analyze or make a decision since the irregular series influence too much. In those cases, we need to use the trend series for analysis. However, as we have seen, the decomposition based on maximum likelihood estimation is apparently not good enough.

From Figures 4.1 and 4.2, we may wonder whether this problem is raised by SSMs inherently or the parameter estimation we chose. The latter turns out to be the case: if we let $\sigma^2 = (1, 1, 1)$, decomposition results would be closer to those from X-11 (see Figure 4.3). More generally, adjusting our parameters allows us to shift fluctuations between these three components, allowing us to arbitrarily increase or decrease spikiness.

This case tells us that we can use SSMs to get close to X-11 but can't achieve it with MLEs. Consequently, if we want to use the SSM to obtain decomposition

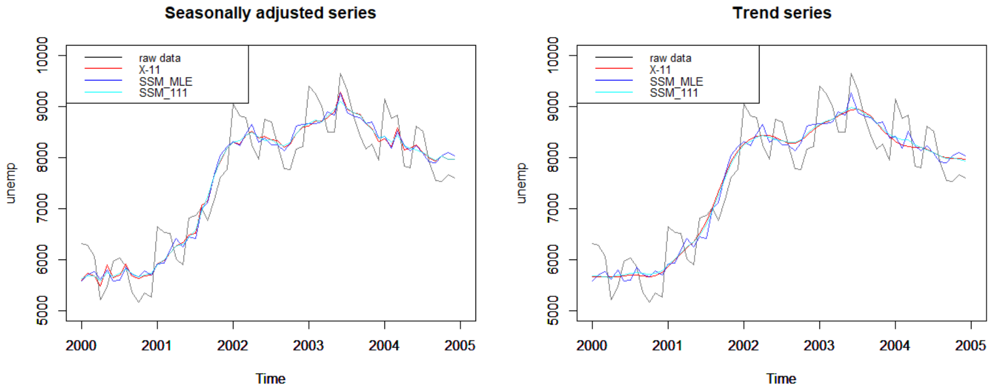


Figure 4.3: Decomposition comparison among X-11, SSM(MLE) and SSM(1,1,1) from 2000 to 2004

results similar to those from X-11, the maximum likelihood estimator may not be a good choice, which means we need to find other approaches to replace MLEs in the SSM.

4.3 Loss functions and optimization

In this section, we shall explore a loss-based method to generate decomposition results similar to those from X-11 instead of choosing numbers manually like we did at the end of Section 4.2, where we have also showed the default (MLE) decomposition result from the SSM and the Kalman filter is not satisfying. And in Chapter 1 and Section 3.1, we said our final goal is to only use SSMs and the Kalman filter to generate the seasonally adjusted and trend series as close as possible to X-11 decomposition. Here, we first propose some loss functions to check whether we could obtain ideal parameter estimation by minimizing the loss. In Sections 4.5 and 4.6, we shall further utilize the loss function to transform experts' knowledge and build a prior.

We have mentioned that we mainly care about the seasonally adjusted and trend series in practice. Since the seasonally adjusted series is the original dataset minus the seasonal component, thus their absolute values of differences of each two adjacent points are equal. Then we define our first loss function as:

$$L_1(\sigma^2) = \|T_{X11} - T_{SSM}(\sigma^2)\|_2^2 + \|S_{X11} - S_{SSM}(\sigma^2)\|_2^2, \tag{4.6}$$

where $\sigma^2 = (\sigma_I^2, \sigma_T^2, \sigma_S^2)$. $T_{SSM}(\sigma^2)$, $S_{SSM}(\sigma^2)$ are the trend and seasonal series we obtained from the state space model with corresponding variance σ^2, and T_{X11}, S_{X11} are results from X-11 with the same observation.

To optimize the loss function, the common technique is the gradient descent. As what we proved in Section 3.3, the trend and seasonal series from the Kalman filter is obtained from two recursive and complicated processes, the filtering and smoothing process. Although $T_{SSM}(\sigma^2)$ and $S_{SSM}(\sigma^2)$ could be expressed with regard to σ^2 theoretically, the expression would be very complicated. Thus it is difficult to take the derivative, which means the gradient descent is hard to be applied here.

An alternative choice is to use the grid search to find the best value but it is too time-consuming if we want to have a good precision. Finally, to accelerate our calculation, we adopt one derivative-free optimization algorithm, *Hooke-Jeeves* algorithm to solve this black-box optimization problem, see [Varadhan et al., 2016].

Now let's still focus on the monthly unemployment data of U.S. from 1990 to 2016. After calculating regarding L_1, the values of parameters σ_I^2, σ_T^2 and σ_S^2 with the lowest loss are 3.93750, 2.90625 and 1.87500, whereas the MLEs are 2.664035, 64895.19 and 0.01197881 separately. As we will see in Figure 4.4, parameters obtained by optimizing loss functions behave better than MLEs (we shall give another loss function 4.7 later).

In Section 4.2, we have seen one main problem is that the trend series from MLEs is not smooth enough, so we add a new term to penalize the smoothness of the trend series. If we view the series $y_1, y_2, \ldots, y_t$ as a function $y_i = f(i)$ of the index i, then we define the operator $D(y_i, y_{i+1}) = y_{i+1} - y_i$, which could be viewed as a natural discrete analogue of the derivative of f at i. Therefore, to force the smoothness of the trend series from SSM to be similar to that from X-11, we could use the penalty term $\|D(T_{X11}) - D(T_{SSM(\sigma^2)})\|_2^2$ as a measurement of the difference between the derivative of two trend series. To sum up, we introduce a new loss function:

$$L_2(\sigma^2) = \|T_{X11} - T_{SSM}(\sigma^2)\|_2^2 + \|S_{X11} - S_{SSM}(\sigma^2)\|_2^2 + \|D(T_{X11}) - D(T_{SSM}(\sigma^2))\|_2^2, \tag{4.7}$$

where $\sigma^2 = (\sigma_I^2, \sigma_T^2, \sigma_S^2)$. The optimal parameter values from loss function L_2 are 4.46875, 3.00000 and 2.31250. Table 4.1 gives the squared L^2 norms of the trend and seasonal component errors between the selected object (MLE, LOSS1 or LOSS2) and X-11. For example, the number 7525182 located at (Trend, MLE) is the sum of squared error of the trend series from the X-11 and the SSM with MLEs. Figure 4.4 is the comparison of the decomposition results from 2000 to 2004, and Figure 4.5 gives the comparison of variability proportions from three components.

	MLE	LOSS1	LOSS2
Trend	7525182	822060	823149
Seasonal	3333002	1010155	1022638

Table 4.1: Trend and Seasonal components' error(unemployment)

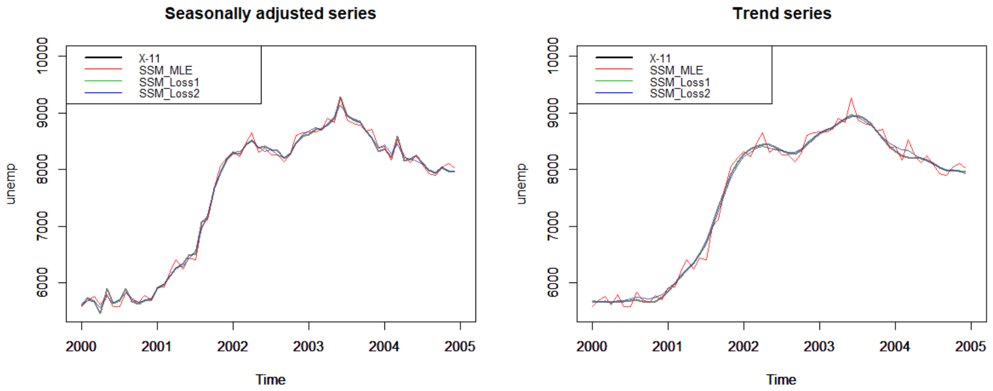


Figure 4.4: Decomposition comparison between X-11 and SSMs from 2000 to 2004

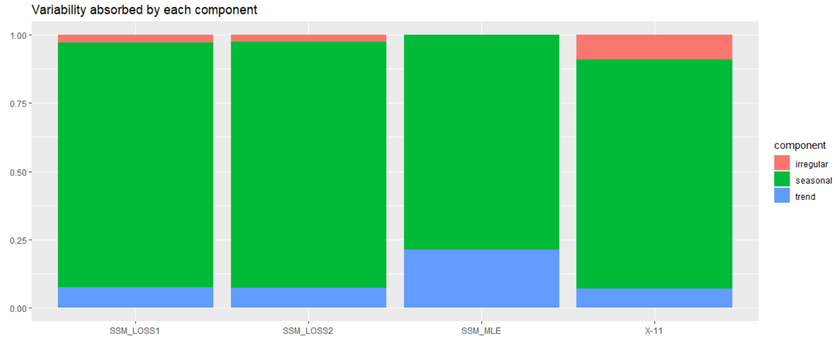


Figure 4.5: Variability proportions absorbed by three components

Table 4.1 and Figure 4.4 show that the decomposition from both loss functions fits better compared with the result from MLEs, and the distinction between the two loss functions is not obvious either. *Note:* in the following text, we adopted L_2 as our default loss function if not specified.

So far we have realized defining an appropriate loss function allows us to reproduce the classical decomposition result but we also notice that our loss function is dependent on the first fitting the dataset to X-11. If we stop here and utilize the loss function to find the optimal values of parameters in the state space model, essentially speaking we are just putting another model and methodology around the X-11. The nature of it is still X-11 instead of state space models and the Kalman filter. Therefore, how to avoid using X-11 to obtain the same or similar estimators from the loss function is our main problem now. We shall talk about it in Sections 4.5 and 4.6.

4.4 Simplification of parameters in SSMs

Recall the model applied in our paper is

$$\begin{aligned} y_t &= T_t + S_t + I_t, & I_t &\sim N(0, \sigma_I^2) \\ T_{t+1} &= T_t + \eta_t, & \eta_t &\sim N(0, \sigma_T^2) \\ S_{t+1} &= -\sum_{j=1}^{s-1} S_{t+1-j} + \omega_t, & \omega_t &\sim N(0, \sigma_S^2) \end{aligned}$$

where $t = 1, \ldots, n$, $\{y_t\}$ is our observation and $\{T_t\}$, $\{S_t\}$ and $\{I_t\}$ are the trend, seasonal and irregular components. In Section 4.3, we computed three variances when optimizing the loss function and maximizing the likelihood. In this section, we shall talk about the reason why we could only estimate two instead of all the three variances σ_I^2, σ_T^2 and σ_S^2. Although we only reduce one parameter here, this would benefit our computation a lot. Then we give an example to pave the way for Sections 4.5 and 4.6.

In signal processing, the *signal-to-noise ratio* ρ is defined as the ratio of the signal variance and noise variance. In [Skagen, 1988], D.W. Skagen pointed out if the signal-to-noise ratio ρ stays constant, the different values of the signal and noise variances will have the same decomposition result after applying the Kalman filter. Corresponding to our model, ρ is $(\frac{\sigma_T^2}{\sigma_I^2}, \frac{\sigma_S^2}{\sigma_I^2})$. In fact, it is not hard to understand the ratio of these components' variances plays a more crucial role when decomposing, because no matter how large the variances are, the sum of components in every moment is fixed. Thus, without loss of generality, we fixed $\sigma_S^2 = 1$ and focus on the estimation of σ_T^2 and σ_I^2 in this paper.

Until now we only worked on a certain dataset *'unemployment'*. To make our final conclusion more convincing, in the following analysis, we shall compare the overall performance on many datasets. Specifically, we shall compare the distributions of the parameter estimations of numerous datasets. At the same time, to guarantee the comparability and make these distributions sensible, all datasets used in each experiment will be simulated from the same or similar state space models. We shall talk more in Section 4.5.

In the following figure, we simulated 1000 monthly time series data sets at length 180 (15 years) from SSM(20,10,1) and then use the first 14 years' data to compute the MLEs and optimal values of σ_I^2 and σ_T^2 with regard to the L_2 (we need the data of the last year to test the prediction accuracy in Section 4.8. In the following decomposition analysis and examples regarding these datasets, we will only use the first 14 years unless otherwise noted). Then we obtained their distributions, see Figure 4.6.

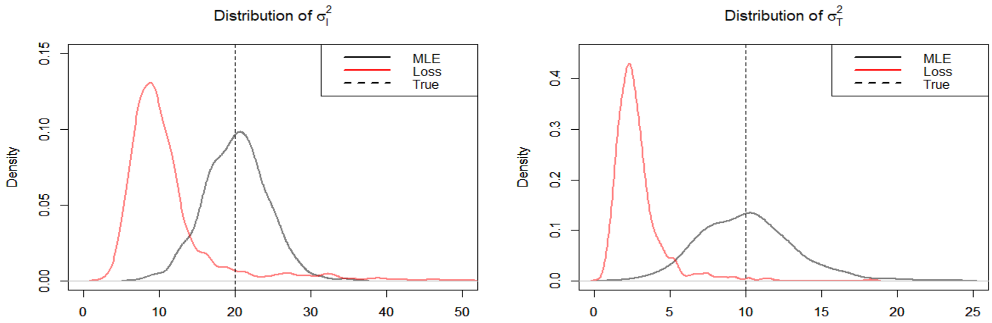


Figure 4.6: Distributions of variance estimators

As we can see, differences of estimators from two methods are prominent in this case. To be specific, because our datasets are simulated from $SSM(20, 10, 1)$, the distributions of MLEs of σ_I^2 and σ_T^2 are approximately normal with mean 20 and 10 separately. However, when computing the loss-based estimators, our standard is the decomposition result from X-11, whose theory is totally different. Thus we could see the distributions of these estimators seems to be irrelevant to the the true value 20 and 10. In fact, after a lot of simulations, we found the distributions of optimal parameters with regard to our loss function do not change too much (See Section 4.6 and Chapter 5) even though variance values $\sigma_I^2, \sigma_T^2, \sigma_S^2$ used for simulation differ a lot.

Since our goal is to use SSMs to obtain the similar decomposition in terms of X-11, the next two sections basically talk about how we *push* the black lines (MLE) to the red line (optimal) without actually fitting an X11 model to each dataset.

4.5 Bayesian analysis

In Section 4.2, we have showed the drawback of the classical estimate MLE. In Section 4.3, we have proved the loss function could help us to find suitable parameter estimation, but we still need to count on X-11 or other existing methods. In this section, we will consider our problem from the Bayesian perspective to avoid the dependence of other methods. Specifically, we shall talk about why we adopt Bayesian analysis and how we use the *prior elicitation* to build our own priors. Then we will check whether the Bayesian inference works through a simple example in the end.

Let's first review Bayesian inference briefly: suppose $g(\theta)$ is the prior distribution of parameter θ and the likelihood function of observations $\{y_1, \ldots, y_n\}$ given θ is $f(y_{1:n}|\theta)$, then the posterior distribution of θ is proportional to the product of them, that is:

$$g(\theta|y_{1:n}) = \frac{g(\theta)f(y_{1:n}|\theta)}{f(y_{1:n})}, \tag{4.8}$$

which is equivalent to

$$log(g(\theta|y_{1:n})) = log(g(\theta)) + log(f(y_{1:n}|\theta)) + constant. \tag{4.9}$$

In this paper, we will use the maximum a posterior estimator (abbreviated to MAP) as the posterior parameter estimator, which could be viewed as the the analogue to the maximum likelihood estimator:

$$\begin{aligned}\theta_{MAP} &= \arg\max_{\theta} g(\theta|y_{1:n}) \\ &= \arg\max_{\theta} g(\theta)f(y_{1:n}|\theta) \\ &= \arg\max_{\theta}[log(g(\theta)) + log(f(y_{1:n}|\theta))], \\ \theta_{MLE} &= \arg\max_{\theta} log(f(y_{1:n}|\theta)).\end{aligned} \tag{4.10}$$

In Bayesian inference, we treat each parameter as a variable and this allows parameters to have their own distribution. At the same time, we are allowed to consider our prior knowledge of parameters when estimating them. Generally speaking, the prior knowledge is what we already know from history or experience before dealing with real observations. From this point of view, we may make use of the information of loss-based optimal estimators to build our prior distributions.

On the other hand, it is not hard to understand that datasets belonging to the same category in economics usually have the similar trend and seasonal patterns like different brands of electronic products usually achieve sales peak in December every year and ice-cream manufacturers usually need to produce more ice-cream every summer. Furthermore, if the magnitudes of these datasets don't have a huge difference, like the sales of these different brands belong to the same level, then we have reasons to believe these datasets should share similar parameters, or their parameters should follow some particular distribution. In fact, this is exactly the key idea of *partial pooling*, see Section 3.4 and [Guerzhoy, 2016].

Therefore, suppose we have abundant datasets from the same economic category, by computing the optimal loss-based estimators in Section 4.2, we could obtain the distributions of these estimators. And when we meet new datasets from the same category, we could use the distributions derived before as our priors. In Bayesian analysis, we call them empirical prior distributions, because they are obtained from data directly. We shall talk more about them in Section 4.6.

Before applying empirical prior distributions, let's first use the weakly-informative prior as a simple example to check the effect of Bayesian analysis and see what will happen. Readers should notice that we don't expect the influence to be *obvious* at this time and this weakly-informative prior attempt doesn't have a relation to our core work. Even if the weak prior does influence our inference in some cases, we should remember it won't always work when encountering larger datasets, since the MAP converges to MLE as sample size increases for a fixed prior, see Section 4.7. An illustrative example 4.5.1 is given below.

Example 4.5.1. In Section 3.4, we have said the state space model could be regarded as a special type of hierarchical models. In [Gelman et al., 2006], Gelman talked about prior distribution choices for variances in hierarchical models. Based on his conclusions, we shall use the half-normal distribution as weakly-informative priors for standard deviations σ_I and σ_T. The reason why we didn't adopt the recommended half-Cauchy is that distributions of loss-based estimators do not have a heavy tail (this argument could also be verified by Figures 5.2 and 5.3). *Note:* the distribution we drew before is for the variance σ^2 instead of the standard deviation.

Specifically, for datasets simulated from the $SSM(20, 10, 1)$, according to the *red* line in Figure 4.6, we know the optimal loss-based estimators for σ_I^2 and σ_T^2 are mainly distributed over [0,40] and [0,10]. Thus with *three-sigma rule*, the variances we set up for half-normal distributions are $\frac{\sqrt{40}}{3}$ and $\frac{\sqrt{10}}{3}$. Then we computed corresponding MAP estimators. The distribution comparison of MLEs, posterior estimators and optimal values is showed in Figure 4.7.

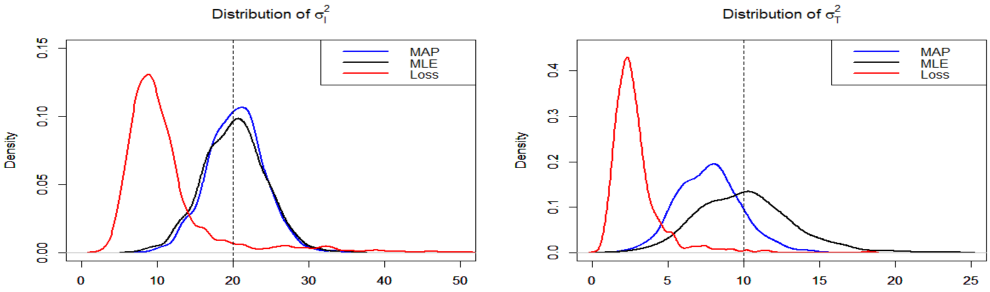


Figure 4.7: Comparison of variance distributions

To further show the difference between three estimators, we drew Table 4.2, Figures 4.8 and 4.9 by defining the decomposition error $Er(\sigma^2)$, where Table 4.2 shows the median, mean and standard error of three different errors, Figure 4.8 is the corresponding box plot and Figure 4.9 is the comparison of three density curves (*Note:* MAP(hnormal) in Figure 4.8 means the MAP estimator from the half-normal priors).

$$Er(\sigma^2) = \|T_{X11} - T_{SSM(\sigma^2)}\|_2^2 + \|S_{X11} - S_{SSM(\sigma^2)}\|_2^2. \tag{4.11}$$

	MLE	Loss	MAP
Median	761.9	645.2	715.2
Mean	785.2	657.1	733.3
sd	207.11	150.54	179.84

Table 4.2: Information of decomposition error

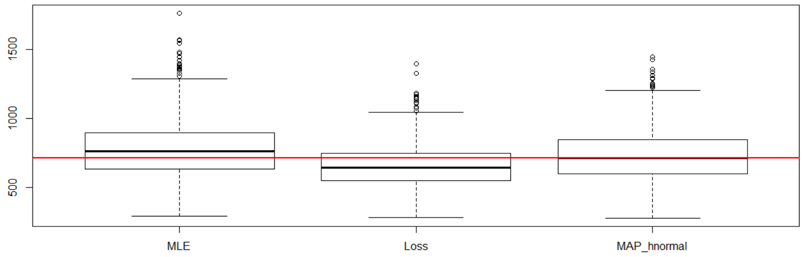


Figure 4.8: Boxplots of decomposition errors

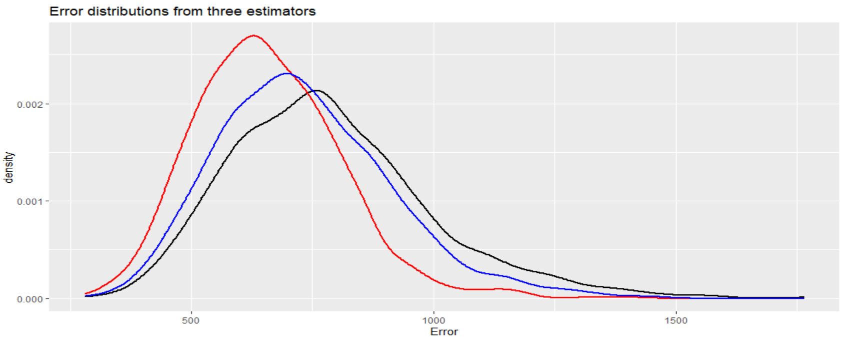


Figure 4.9: Densities of decomposition errors

Black, red and blue curves stand for MLE, Loss and MAP separately

We could tell that the maximum a posterior estimators does improve the decomposition results compared with MLEs, and to back up our argument here we also used the *Friedman* test and *Mann–Whitney U* test (see [Mangiafico, 2016]) to check whether the difference of errors from MLE and MAP are prominent or not. Both results showed that the difference is prominent (See Appendix B).

4.6 Empirical prior distributions

In Section 4.5, we have said we could compute the distributions of loss-based estimators as empirical priors and apply them when meeting new similar datasets. And we also have seen the weakly-informative prior, the half-normal distribution could help improve the decomposition results. In this section, we shall further explore the MAP estimators from the empirical priors and continue with Example 4.5.1 to test the effect of empirical priors. As we know, in Bayesian analysis, with the amount of data increasing, the influence of the prior distribution upon the posterior estimators will be weaker, see Example 4.7.1. To control its influence, we will introduce a user-defined weight k regarding the log-prior in the end of this section. Mathematically, k corresponds to the effective sample size of the prior.

As what we will see in Section 4.8, SSMs seem to have better predictive accuracy, while experts at Statistics Canada tell us that they prefer the decompositions returned by X-11. So how to find a method that has the best properties of both X11 and SSMs? Since SSMs could match X-11 for some choice of parameters, the obvious approach is to try to *encode* the expert opinion as a prior for the SSMs, and then apply the Bayesian analysis in our problem. As what we have said in Section 4.5, the simplest approach is to build a prior with the parameters whose decomposition result is ”close” to the result of X-11, where ”close” is measured by some loss function of statistical relevance. This is very similar to the general approach of Approximate Bayesian Computation (see [Turner and Van Zandt, 2012]) and more specifically to previous work on prior elicitation from experts (see [Albert et al., 2012]).

Example 4.6.1. Continuing with Example 4.5.1, we simulated another 3000 datasets from $SSM(20, 10, 1)$ as the history data and computed the distributions of the estimators with the lowest loss defined by the loss function 4.7 as empirical priors. The 1000 simulated datasets mentioned in Section 4.3 will be used as *new* datasets and to calculate different types of estimators. Then we compared these estimators' distribution to check the effect of empirical priors.

Figure 4.10 is the empirical distributions of σ_I^2 and σ_S^2 we obtained from 3000 simulated datasets after using the *Gaussian kernel density estimation* and their parametric approximations. Here, we use two piecewise functions to approximate them:

$$g(\sigma_I^2) = \begin{cases} \frac{1}{\sqrt{2\pi}\cdot 2.9} exp(-\frac{(\sigma_I^2 - 8.8)^2}{2\cdot 2.9^2}) \quad if \quad 0 < \sigma_I^2 < 14.5, \\ 0.2 \cdot exp(-0.2\sigma_I^2) \quad if \quad \sigma_I^2 \geq 14.5, \end{cases} \tag{4.12}$$

$$g(\sigma_T^2) = \begin{cases} \frac{1}{\sqrt{2\pi}\cdot 0.83} exp(-\frac{(\sigma_T^2 - 2.46)^2}{2\cdot 0.83^2}) \quad if \quad 0 < \sigma_T^2 < 4.2, \\ exp(-\sigma_T^2) \quad if \quad \sigma_T^2 \geq 4.2, \end{cases} \tag{4.13}$$

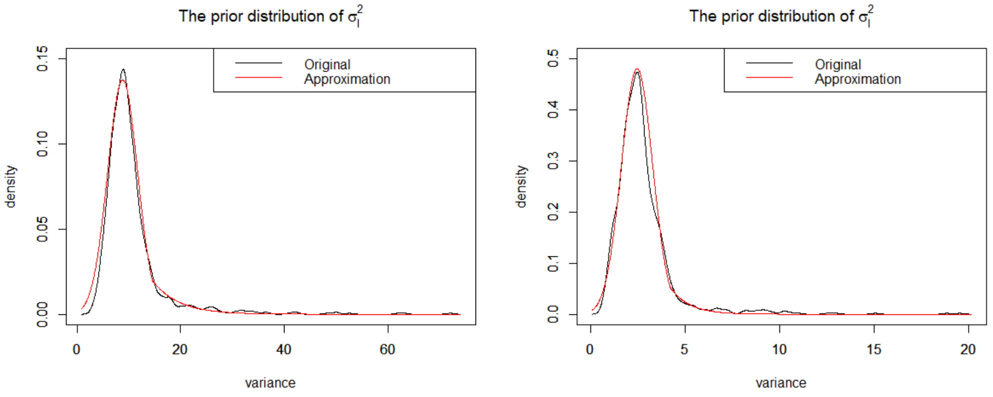


Figure 4.10: Empirical prior distributions

Note: The raw empirical distributions have a few discontinuity points at the tails. If we use them as priors directly, the posterior estimator will not take these points and this would make the distribution of the MAP estimator having a lot of different peaks. Hence without loss of generality, we approximate them with parametric functions to make them smooth. Although both parametric approximations are improper (the integral of each piecewise function is not 1 over the domain), this won't influence our final inference, because we need to take the logarithm to compute the MAP estimator in the end.

Then we used approximated distributions as our priors $g(\theta)$ and computed the corresponding MAP estimators by equation 4.10, that is

$$\theta_{MAP} = \arg\max_{\theta}[log(g(\theta)) + log(f(Y_n|\theta))].$$

As shown in Figure 4.11, the red line is our target, the black line is the MLEs' distribution and the two blue lines are MAPs' distributions from weakly-informative priors (denoted by MAP_{hnor}) and empirical priors (denoted by MAP_{emp}):

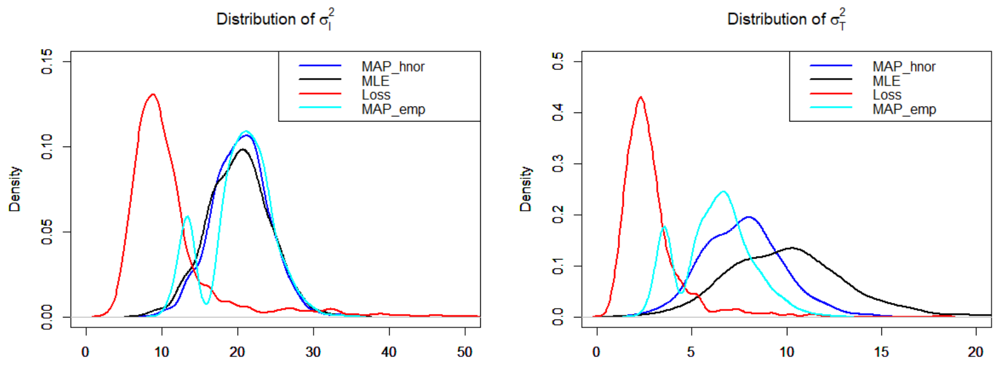


Figure 4.11: Comparison of variance distributions (2)

It seems for some datasets, their MAP estimators have changed because of the influence of priors, but for the others, the MAP estimators barely changed especially for σ_I^2. The reason for this phenomenon is the magnitude of the log-likelihood is too large and as a result the posterior estimate is not very sensitive to small numbers. That is: the log-prior is much smaller than the log-likelihood (in absolute value).

To test the difference among these estimators, we also computed the median, mean and standard error of the decomposition error, and did the same hypothesis tests in Example 4.5.1. And it showed that the decomposition result from the empirical posterior estimator did improve the decomposition result compared with MLEs, but didn't have an obvious improvement compared with results from the half-normal distribution (see Appendix B).

Like we said in Example 4.6.1, our MAP estimator is mainly controlled by the (log)likelihood, in another word, the influence of priors is too weak. Suppose we have more confidence with our prior compared with the likelihood, then to enlarge the influence of priors, we could put more *weights* on the prior distributions, that is:

$$\theta^*_{MAP} = \arg\max_{\theta}[k \cdot log(g(\theta)) + log(f(Y_n|\theta))], \tag{4.14}$$

where k is the weight we put on the prior distributions. Later in Section 4.8, we shall see the prediction of SSMs is more accurate than results from X-11. Thus if $k = 1$,

θ^*_{MAP} is the MAP estimator from the standard Bayesian inference; if $k > 1$, we put more weights on the prior distributions, in another word, we think more of X-11 and the decomposition result; if $k < 1$, we put more weights on the likelihood of sample Y_n, or we could say we care more about the prediction. Users could tweak k according to their demand. Tables 4.4 and 4.5 in Chapter 4.8 will give the variation of the means and standard deviation of the decomposition error and the prediction error for different k. We will talk more about the weight k in Section 4.7.

4.7 Weight adjustment for different lengths

Suppose we already know the empirical priors $g(\theta)$ for one *specific* dataset and a good weight k, but when facing a new dataset, the previous setting could be useless because the likelihood changes. So we need to extend our method to more general cases if we don't want to re-calculate the empirical prior or the prior weight endlessly. In this section, we shall talk about how to deal with a new dataset at a different length. In the real life, we often have various datasets from the same category, but due to the different start date recorded in history, they often don't have the exactly same length. Meanwhile, for one specific dataset, its length is also increasing as time goes on. In Section 4.6, we realized we may achieve a satisfying result by adjusting weights on the prior distribution. But the log-likelihood is related to the sample size n, as we can tell from equation 4.5. Thus we need to tweak k when datasets have different length. But we do not expect to spend time seeking a good weight k every time. Ideally speaking, if we find the rules of log-likelihood changes for different lengths, we will know how to change k as well. An illustrative example is given below.

Example 4.7.1. Suppose we have the sample $x_1, x_2, \ldots, x_n$ from the distribution $Bernoulli(\theta)$, that is, $P(x_1, x_2, \ldots, x_n|\theta) = \theta^{\sum x_i}(1-\theta)^{n-\sum x_i}$, and the prior on θ is $Beta(\alpha, \beta)$, where $P(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$, and α and β are constants. Then we can show the MAP estimator of θ is

$$\begin{aligned}\theta^{(0)}_{MAP} &= \arg\max_\theta[log(\theta^{\alpha-1}(1-\theta)^{\beta-1}) + log(\theta^{\sum x_i}(1-\theta)^{n-\sum x_i})] \\ &= \frac{\alpha - 1 + \sum_{i=1}^n x_i}{\alpha+\beta-2+n}. \end{aligned} \tag{4.15}$$

If we extend the length of our sample to $2n$, the MAP estimator would be

$$\theta^{(1)}_{MAP} = \frac{\alpha - 1 + \sum_{i=1}^{2n} x_i}{\alpha+\beta-2+2n}. \tag{4.16}$$

Out of some purposes, suppose we want to keep the new estimator $\theta^{(1)}_{MAP}$ around the value of $\theta^{(0)}_{MAP}$, then by multiplying the log-prior probability by 2, we could obtain

$$\begin{aligned}\theta^{(2)}_{MAP} &= \arg\max_\theta[2\cdot log(\theta^{\alpha-1}(1-\theta)^{\beta-1}) + log(\theta^{\sum x_i}(1-\theta)^{2n-\sum x_i})] \\ &= \frac{2(\alpha-1) + \sum_{i=1}^{2n} x_i}{2(\alpha+\beta-2)+2n}. \end{aligned} \tag{4.17}$$

Since x is a Bernoulli distribution, $\sum_{i=1}^{2n} x_i$ should approximate $2\sum_{i=1}^{n} x_i$. Thus, $\theta^{(2)}_{MAP}$ approach to $\theta^{(0)}_{MAP}$ as n increases. In this example, we could find if we don't want to change the existing posterior estimator when meeting another similar dataset at a different length, we could use the ratio of two datasets' length as the weight for our prior.

Remark: For the Bernoulli distribution, we know the MLE of θ is $\frac{\sum_{i=1}^{n} x_i}{n}$, so based on equation 4.15, $\theta_{MAP} \to \theta_{MLE}$, as $n \to \infty$. This also explains why we want to put a weight on the prior distribution when we have more confidence with our prior information instead of the likelihood function.

Now let's look at the log-likelihood of our state space model. We have showed in Section 4.2, for univariate cases, the log-likelihood is

$$\ell(\theta) = -\frac{n}{2} log 2\pi - \frac{1}{2} \sum_{t=1}^{n} log(F_t + v_t^2 F_t^{-1}).$$

And based on Lemma 3.3.1, we know $P_t \to \bar{P}$ as t increases, when matrices such as Z_t and H_t in SSMs are time-invariant. Since $F_t = Z_t P_t Z_t'$, thus $F_t \to \bar{F}$ as P_t converge to $\bar{P}$. And in Section 3.3, it is not hard to derive $E(v_t) = 0$. Therefore, intuitively speaking, the log-likelihood $\ell(\theta)$ should be a linear function regarding length n approximately. To check our hypothesis, for the same dataset, we fixed variances in the state space model at different sets and plotted the figure of its log-likelihood with regard to its length, see Figure 4.12 (the dataset is simulated from $SSM(20, 10, 1)$ at length 360). Table 4.3 shows the log-likelihood at different length for these SSMs.

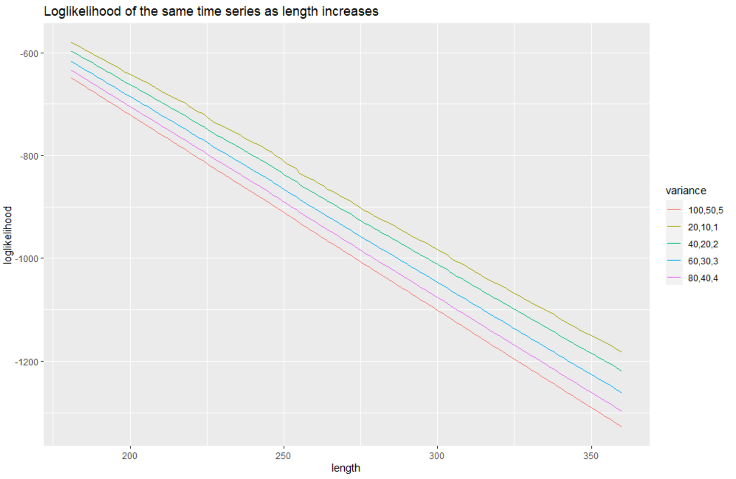


Figure 4.12: Relations between log-likelihood and length

	(20,10,1)	(40,20,2)	(60,30,3)	(80,40,4)	(100,50,5)
180 (15 years)	-580.7678	-597.334	-617.5942	-634.9026	-649.5577
360 (30 years)	-1182.5489	-1218.846	-1261.2929	-1297.2977	-1327.6936

Table 4.3: Log-likelihood under different length and variances of the same dataset

As we have seen, for the state space model with fixed variances, the log-likelihood is approximately direct proportional to the length n.

Combined with the conclusion we drew in Section 4.6, given the sample Y_n at length n, an empirical prior distribution $g(\theta)$ and the log-likelihood function $\ell(\theta)$ as $log(f(Y_n|\theta))$ in equation

$$\theta_{MAP} = \arg\max_{\theta}[k \cdot log(g(\theta)) + log(f(Y_n|\theta))],$$

suppose we already know k_0 could help us have a good result under Y_n, $g(\theta)$ and $\ell(\theta)$, then when we meet a new dataset Y_{n^*} from the same category, to make the MAP estimator stable, we could let $k = \lambda k_0$ as the new weight on the prior $g(\theta)$, where $\lambda = \frac{n^*}{n}$, while keep others the same.

So far, we have showed we could transform the information of the linear-filter-based method X-11 to a likelihood function by building the empirical prior distributions. And to generate decomposition results by SSMs and the Kalman filter similar to those from X-11, we put a weight k to the empirical priors and users could tweak k as they want. We will use Tables 4.4 and 4.5 to show the effect of different k in Section 4.8. In this section, we showed how to tweak an existing good weight k_0 when facing another new dataset at different length, if we hope our priors play the same important role as before. In Section 4.8, we shall compare X-11 and the SSMs with different estimators' behaviours for prediction problem, which is another very important part in practice as we said in Chapter 1.

4.8 Prediction Comparison

Recall in Section 4.4, we said we saved the data of the last year for the prediction problem when using the simulated 1000 datasets from $SSM(20, 10, 1)$. In this section, we shall continue with Example 4.6.1, and take the weight k and the prediction error into consideration.

Suppose $y_1, \ldots, y_s$ is the real data of one series over one period s, and $x_1, \ldots, x_s$ is the prediction we obtained from one specific model, then we define the prediction error as

$$\sum_{i=1}^{s}(y_i - x_i)^2. \tag{4.18}$$

Example 4.8.1. Continued with Example 4.6.1, remember our definition of decomposition error in Section 4.5 is

$$Er(\sigma^2) = \|T_{X11} - T_{SSM(\sigma^2)}\|_2^2 + \|S_{X11} - S_{SSM(\sigma^2)}\|_2^2,$$

where σ^2 is the variance estimator. Here, we first computed decomposition errors under different estimators over the first 14 years' part of the same 1000 datasets simulated previously, including MLEs, loss-based optimal estimators, and MAP estimators under different weight k. Then we could obtain Table 4.4:

	median	mean	sd
MLE	761.9	785.2	207.1050
k=0.1	747.8	767.4	197.3232
k=0.5	716.9	734.5	180.6615
k=1	709.2	728.6	181.3250
k=2	714.8	738.5	190.3706
k=5	675.2	700.6	179.6753
k=10	662.3	680.8	155.7274
k=50	662.8	679.1	152.1393
Loss	645.2	657.1	150.5360

Table 4.4: Statistics of decomposition errors

Apparently, as we put more weight on the prior, the decomposition error of corresponding posterior estimators would be smaller and closer to the optimal value.

For the prediction problem, we don't need to rely on the X-11 or other conventional methods anymore, thus we could add the X-11 as a new candidate besides those in Table 4.4. And as we mentioned in Section 4.6, we found the SSMs usually have more accurate predictions than X-11, see Table 4.5.

	median	mean	sd
X-11	1044.4	1499.1	1319.804
MLE	947.4	1310.2	1121.327
k=0.1	944.73	1309.62	1119.985
k=0.5	843.51	1310.07	1119.356
k=1	940.13	1314.05	1123.506
k=2	945.64	1316.53	1126.389
k=5	935.54	1319.82	1129.023
k=10	935.38	1321.46	1132.451
k=50	942.5	1323.6	1133.275
Loss	946.37	1327.50	1135.614

Table 4.5: Statistics of prediction errors

As we can see, the X-11 did the worst job for prediction compared with other state space models. And the MLEs behaves very well on the whole although the differences among these SSMs is very *tiny*. There are also some existing discussion with regard to the prediction comparison, see [Ellis, 2015].

In Section 4.6, we have explained the real meaning of the MAP estimator from the empirical prior distribution. Now, Tables 4.4 and 4.5 could help us understand the word *trade-off* better.

In conclusion, X-11 could give us the better decomposition results that are preferred by experts while the SSMs with MLEs behave better for prediction. Both goals are reasonable but it seems we have to make a trade-off between them - we can't get smooth decompositions and prediction accuracy at the same time.

Chapter 5

Application

In the previous chapters, we have introduced the seasonal adjustment problem, different methodologies, the problem we found regarding MLEs in SSMs and our approach to improve it. In this chapter, we shall apply our method to a real dataset *unemployment* (see Section 4.2) and compare its result with those from other estimators or models. In the end, we will use a simple example to show how to achieve the partial pooling with the Kalman filter and SSM.

The original dataset and its decomposition result from X-11 given in Section 2.3 are showed in Figure 5.1.

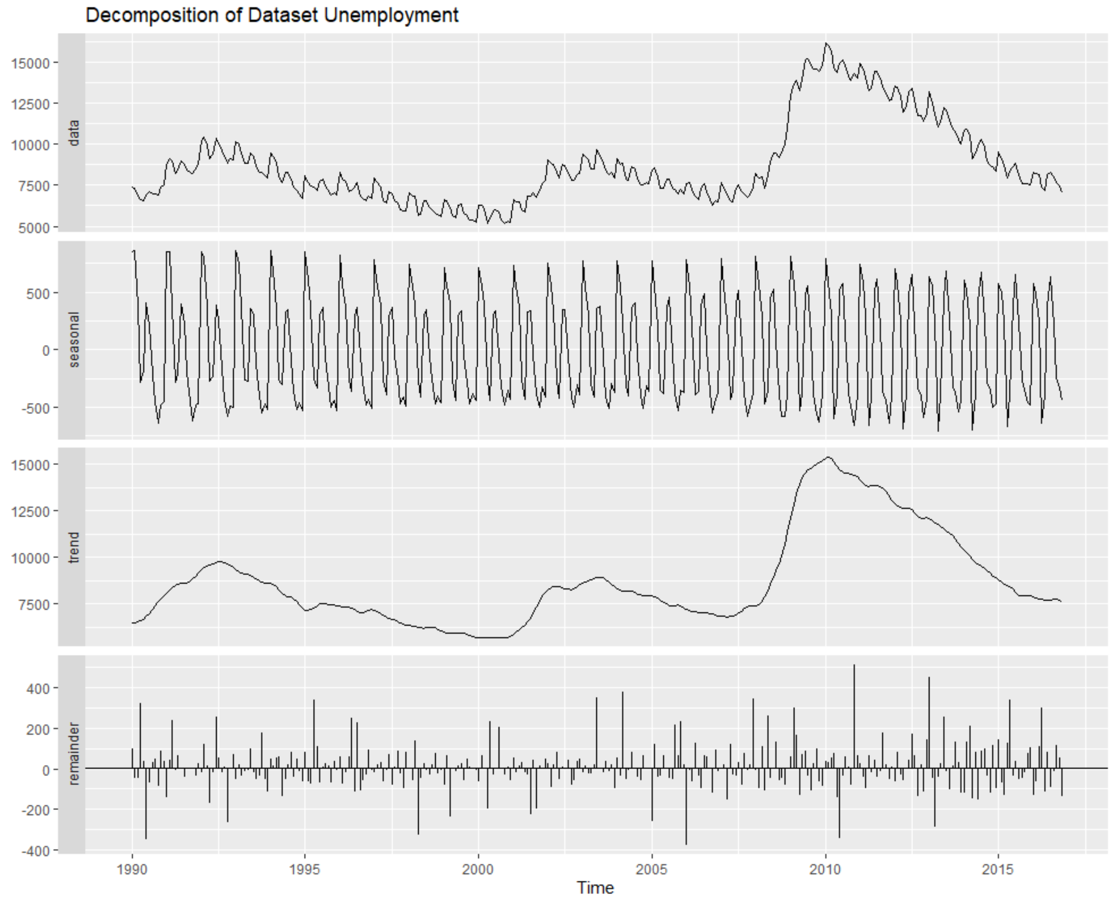


Figure 5.1: Classical decomposition of the unemployment dataset

In Section 4.2, we have seen one important motivation behind our work, which is the decomposition result from the state space model with MLEs is not friendly for analysis, especially for the trend component. And we also have seen the comparison regarding the seasonally adjusted series and the trend from SSMs with different estimators and X-11 in Figure 4.4, where we found by using a sensible loss function, we could obtain parameters with better decomposition results.

Now, to realize our method, we need two empirical prior distributions for σ_I^2 and σ_T^2. You may worry that we don't have a bunch of datasets similar to *unemployment* to generate empirical priors. This is a real and crucial problem in our research. One choice to solve it is that we could use the MLEs of *unemployment* dataset as parameters in the SSM, then use this SSM to simulate thousands of datasets as the history data, and finally construct the empirical prior over them, but the disadvantage of this method is the computation is too time-consuming and one specific MLE may not be accurate enough to describe some class of data. Thus it is not very practical.

On the other hand, after plenty of simulations, we found the distinction among the empirical distributions under different cases are *not* very much. Figures 5.2 and 5.3 are the distributions of σ_I^2 and σ_T^2, and Table 5.1 is the information of SSMs we used for simulation. Therefore, without loss of generality, we could use the empirical priors defined in equations 4.12 and 4.13 as the prior distributions for the unemployment dataset. *Note*: the legend *idemat* in Figures 5.2 and 5.3 is the abbreviation for the *ideal value matrix*, which is used to build empirical priors. And *simlist* in Table 5.1 is the list of simulated datasets used to compute the corresponding *idemat* and each list contains 1000 datasets.

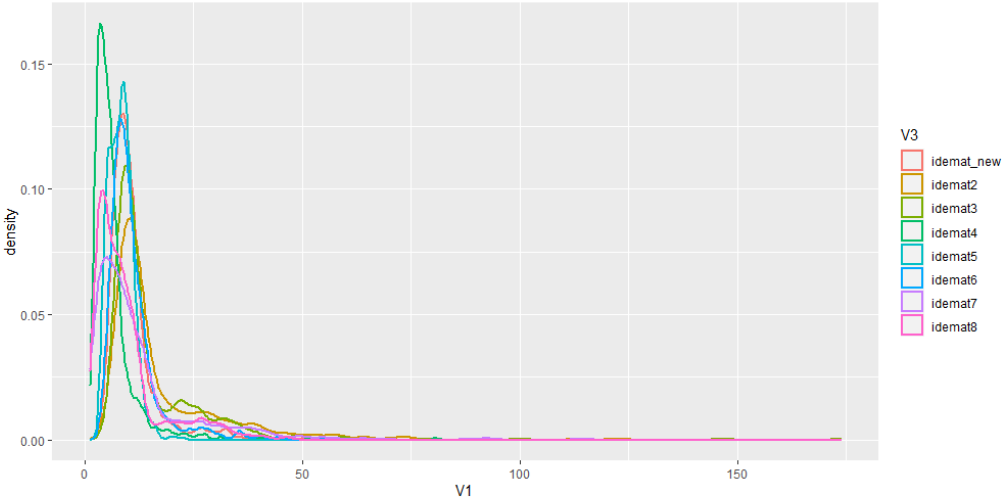


Figure 5.2: Empirical distributions of the irregular variance from 8 groups

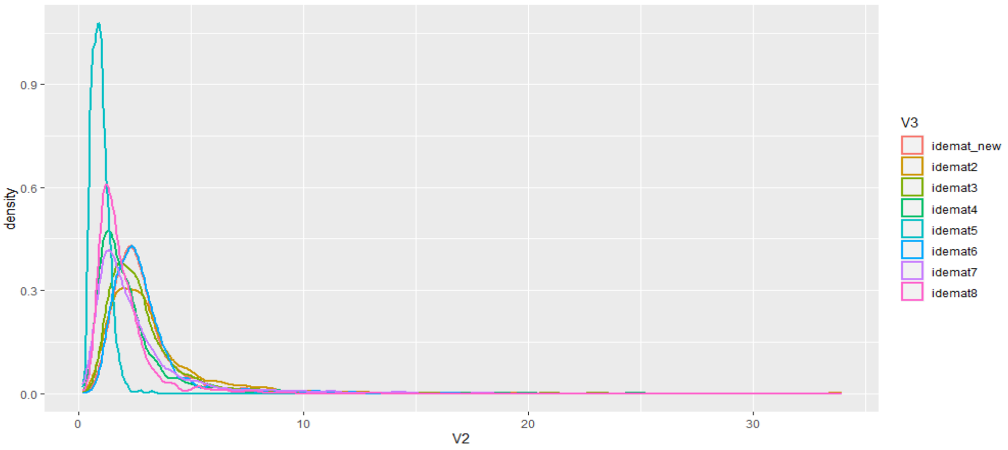


Figure 5.3: Empirical distributions of the trend variance from 8 groups

Name	Length(yrs)	σ_I^2	σ_T^2	σ_S^2
simlist_new	15	20	10	1
simlist2	15	100	25	1
simlist3	20	100	25	1
simlist4	15	25	100	1
simlist5	15	1	0.25	1
simlist6	15	200	100	10
simlist7	15	$(N(0,10))^2$	$(N(0,10))^2$	1
simlist8	30	$(N(0,10))^2$	$(N(0,10))^2$	1

Table 5.1: Information of SSMs used for simulation

After figuring out the empirical prior problem, we first adopted the standard Bayesian analysis by setting weight k equal to 1. The posterior estimators of σ_I^2 and σ_T^2 under this case are 3715.0938 and 746.4023, while the MLEs are 19.18455 and 66778.52, and the loss-based optimal parameter values (denoted as *IDEAL* in Figure 5.4) are 1.855240 and 1.268692 (Note: σ_S^2 is always fixed at 1 in these cases). As we can see, the MLE of the noise variance for the trend is very large but the loss-based estimator is very small. This is the reason why the smoothness of their curves is different. Figure 5.4 gives the partial comparison of the seasonally adjusted series and trend series from SSMs with different estimators and X-11.

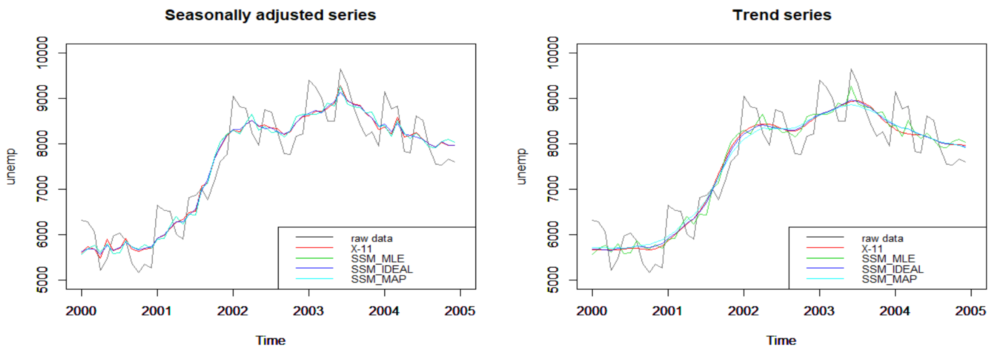


Figure 5.4: Decomposition comparison from 2000 to 2004

Visually, the trend series from the MAP estimator does look better than that from the MLE, and is closer to X-11 result, although the difference of seasonally adjusted series between them is not obvious. To better present the transformation, we drew a bar diagram regarding the proportion of variability absorbed by different components, see Figure 5.5.

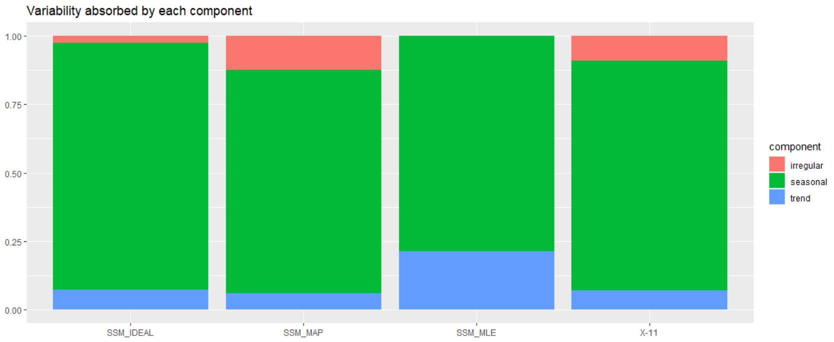


Figure 5.5: Comparison of the variability proportions

Apparently, in our case, the irregular component from the state space model with MLEs is too weak and too much variability is assigned to the trend component, whereas the X-11 method takes care of both series and does a good trade-off! For the empirical MAP estimator, by considering the information from X-11 as the prior, we transformed some variability from the trend to the irregular in the context of the state space models. Thus the trend curve is smoother and easier for analysis.

Now let's look at their behaviours for prediction. Figures 5.6 gives the comparison of predictions from three SSMs with different estimators and X-11, and the true value from December 2015 to November 2016. And Figure 5.7 compares their prediction with the 95% confidence interval and the true data separately. Based on the plot, the X-11 did a bad job for prediction compared with the other two SSMs. In addition, Table 5.2 gives their sums of the squared error denoted by Er. *Note:* we used the optimal ratios to compute the decomposition before, where the scaling is not significant, but here when drawing the confidence intervals, the scaling matters. Recall that we fixed σ_S^2 at 1 previously. Naturally, to determine the scaling, we could use the MLE of σ_S^2 given the optimal ratios obtained before. After calculation, the MLEs of σ_S^2 given the ratios of the ideal and posterior cases (which are (1.855240, 1.268692) and (3715.0938, 746.4023) respectively) are 20861.78 and 31.32604 respectively (we computed the MLEs of three variances directly and obtained $\sigma^2 = (7.420803, 66084.02, 0.399136)$).

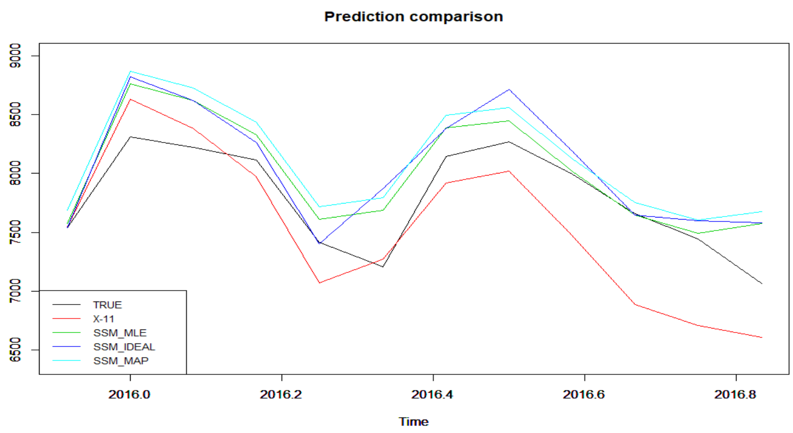


Figure 5.6: Comparison of predictions for the next year

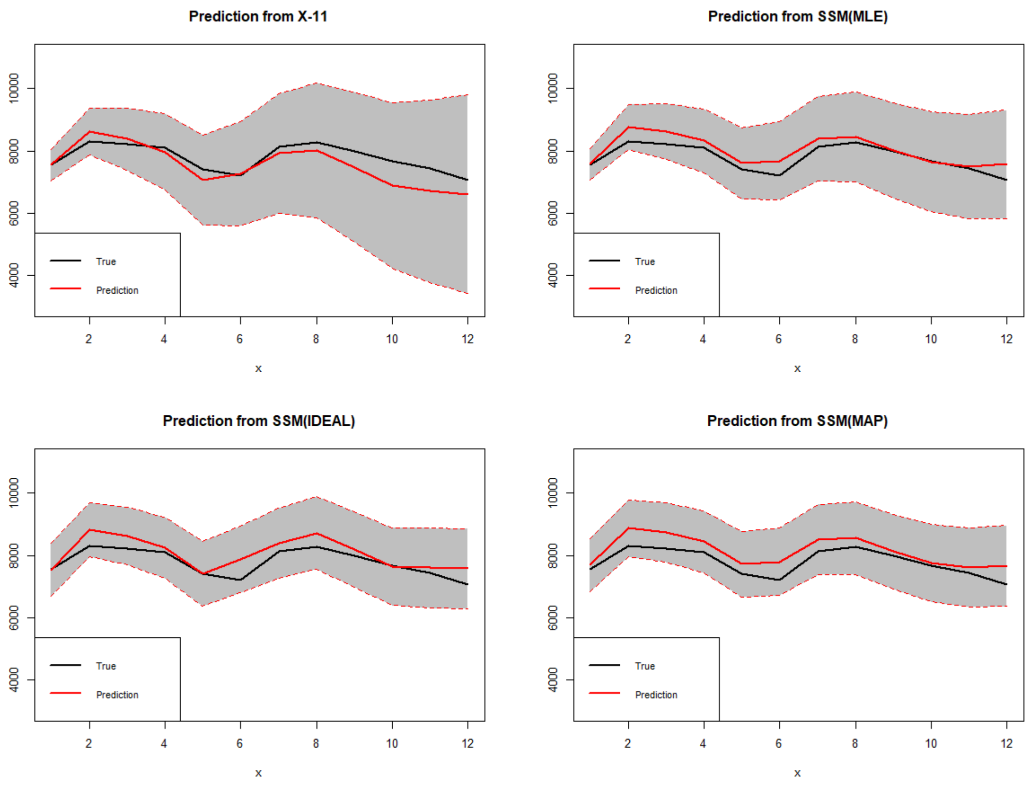


Figure 5.7: Prediction with 95% confidence intervals

	X-11	SSM(MLE)	SSM(IDEAL)	SSM(MAP)
Er	2009890	1026737	1470003	1771164

Table 5.2: Sum of the squared error

As we indicated in Section 4.8, compared with X-11's prediction result, the state space model using MLEs usually have better performance, while the empirical MAP estimator is a compromise between them. But we also find one problem with the MAP estimator now is that the true value doesn't fall into its 95% confidence interval.

In this real case, we proved that we could utilize the empirical priors to generate decomposition results more similar to those from X-11 compared with the default output. And this process could be achieved without using X-11 or other conventional methods, which is the main purpose of our research. At the same time, we showed for the prediction problem, the state space model with MLEs behaves better than X-11, and the effect of the empirical posterior estimator is better than X-11 but poorer than MLEs.

Now, let's look how we can achieve partial pooling in seasonal adjustment with the state space model and the Kalman filter. In Section 3.4, we have seen two types of the hierarchical model for the seasonal adjustment problem. Here, we will only use the second model 3.29 to study a simple case where the data has been seasonally adjusted, that is, the data only contains the trend and irregular series. *Reminder:* This example here may not be very appropriate to show the real functionality of partial pooling for seasonal adjustment. Our main purpose is to prove we could do partial pooling with SSMs and the Kalman filter for the seasonal adjustment problem and give readers an intuitive understanding of the partial pooling.

Suppose we already know the seasonally adjusted sales data of two companies A and B, and their trends gained from two independent SSMs are given in Figure 5.8.

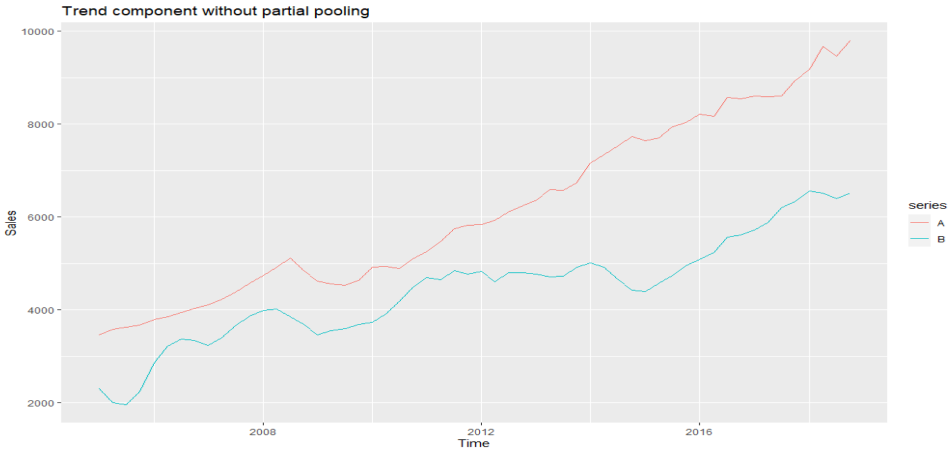


Figure 5.8: Trends before partial pooling

Now let's say we want to do complete pooling with regard to the trend at first. In another word, we suppose both trends $\{T_{1t}\}$ and $\{T_{2t}\}$ follow the same model. For a single dataset, suppose our initial model is

$$\begin{cases} y_{it} &= T_{it} + \varepsilon_{it}, \qquad \varepsilon_{it} \sim N(0, \sigma_{iI}^2), \\ T_{it} &= T_t^* + \eta_{1t}, \qquad \eta_{1t} \sim N(0, \sigma_T^2), \\ T_{t+1}^* &= T_t^* + \eta_{2t}, \qquad \eta_{2t} \sim N(0, \sigma_{T^*}^2), \end{cases} \tag{5.1}$$

where $i = 1, 2$ in our instance. *Note:* This original model above is not in a state space form technically. Thus to solve it with the Kalman filter, we need to transform it to a state space form, see equations 5.2.

$$\begin{gathered}
\begin{pmatrix} y_{1t}\\ y_{2t}\end{pmatrix} = \begin{pmatrix}1&0&0\\0&1&0\end{pmatrix}\begin{pmatrix}T_{1t}\\T_{2t}\\T^*_{t+1}\end{pmatrix} + \begin{pmatrix}\varepsilon_{1t}\\ \varepsilon_{2t}\end{pmatrix}, \qquad \begin{pmatrix}\varepsilon_{1t}\\ \varepsilon_{2t}\end{pmatrix} \sim N(0, \begin{pmatrix}\sigma^2_{1I}&0\\0&\sigma^2_{2I}\end{pmatrix})\\
\begin{pmatrix}T_{1t}\\T_{2t}\\T^*_{t+1}\end{pmatrix} = \begin{pmatrix}0&0&1\\0&0&1\\0&0&1\end{pmatrix}\begin{pmatrix}T_{1,t-1}\\T_{2,t-1}\\T^*_{t}\end{pmatrix} + \begin{pmatrix}1&0\\1&0\\0&1\end{pmatrix}\begin{pmatrix}\eta_{1t}\\ \eta_{2t}\end{pmatrix}, \quad \begin{pmatrix}\eta_{1t}\\ \eta_{2t}\end{pmatrix} \sim N(0, \begin{pmatrix}\sigma^2_{T}&0\\0&\sigma^2_{T^*}\end{pmatrix}).
\end{gathered} \tag{5.2}$$

To apply the Kalman filter, we computed the MLEs of these variance parameters first, and obtained 803198.2, 6664.782, 3.328507×10^{-6} and 37241.81 respectively for σ^2_{1I}, σ^2_{2I}, σ^2_{T} and $\sigma^2_{T^*}$. Then plugged them into the Kalman filter to compute two trend series $\{T_{1t}\}$ and $\{T_{2t}\}$, as showed in Figure 5.9. *Note:* Because the new SSM contains some hyperparameters and is different from our SSM 4.2 before, we didn't use any empirical prior here and took the MLE as the estimator.

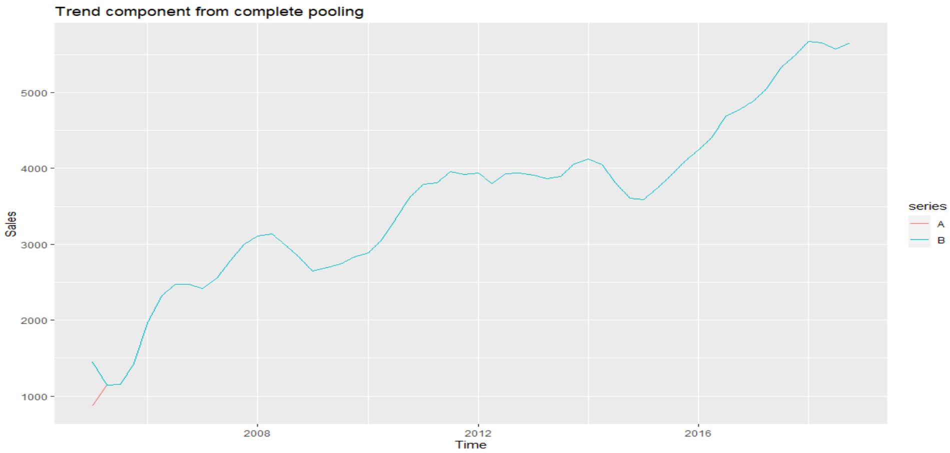


Figure 5.9: Trends after complete pooling

As we can see, the problem with this figure is that two trends are almost the same to each other, which is exactly the result of complete pooling but usually not what we want in practice.

To avoid the exactly same decomposition result but make them relevant to each other, we allow the variances of two trend components σ^2_T in equation 5.2 to be different, then we shall have the state space model for *partial pooling*:

$$\begin{gathered}
\begin{pmatrix} y_{1t}\\ y_{2t}\end{pmatrix} = \begin{pmatrix}1&0&0\\0&1&0\end{pmatrix}\begin{pmatrix}T_{1t}\\T_{2t}\\T^*_{t+1}\end{pmatrix} + \begin{pmatrix}\varepsilon_{1t}\\ \varepsilon_{2t}\end{pmatrix}, \qquad \begin{pmatrix}\varepsilon_{1t}\\ \varepsilon_{2t}\end{pmatrix} \sim N(0, \begin{pmatrix}\sigma^2_{1I}&0\\0&\sigma^2_{2I}\end{pmatrix})\\
\begin{pmatrix}T_{1t}\\T_{2t}\\T^*_{t+1}\end{pmatrix} = \begin{pmatrix}0&0&1\\0&0&1\\0&0&1\end{pmatrix}\begin{pmatrix}T_{1,t-1}\\T_{2,t-1}\\T^*_{t}\end{pmatrix} + \eta_t, \qquad \eta_t \sim N(0, \begin{pmatrix}\sigma^2_{1T}&0&0\\0&\sigma^2_{2T}&0\\0&0&\sigma^2_{T^*}\end{pmatrix}.
\end{gathered} \tag{5.3}$$

Similarly, we still adopt the MLEs (which are 7.288344, 1.842641×10^{-8}, 7644.67, 854570.8 and 28875.73 respectively for σ^2_{1I}, σ^2_{2I}, σ^2_{1T}, σ^2_{2T} and $\sigma^2_{T^*}$) to apply the Kalman filter over the SSM 5.3, then two trend components we obtained are showed in Figure 5.10.

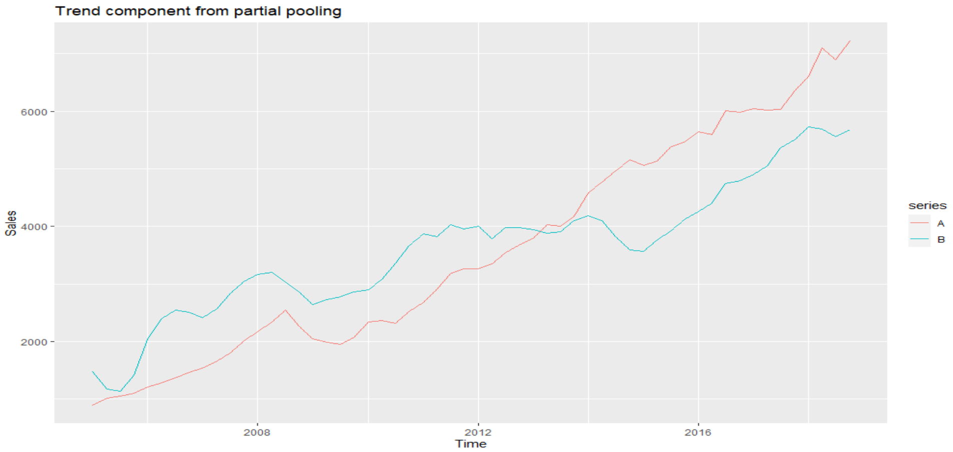


Figure 5.10: Trends after partial pooling

Compared with Figure 5.8, the effect of partial pooling is to pull the previous trends to each other; compared with Figure 5.9, it tries to make a difference between two trend series. To show the difference of three pooling methods from another way, we took the difference of two trends in each figure, and plotted them in Figure 5.11. As what we expect, the difference of two trends from no pooling is very large, the difference from complete pooling is almost zero and partial pooling's result is greater than zero but less than that from no pooling (in absolute value). The surprising finding is that the correlation of the differences from no pooling and partial pooling is 1!

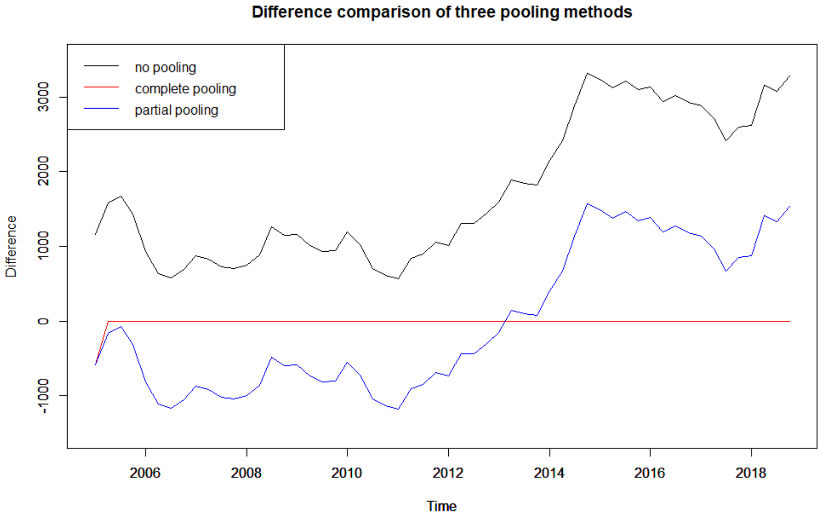


Figure 5.11: Comparison of trend differences from three pooling methods

As what we said in Section 3.4, if one time series is too short, by partial pooling, we could make use of other series information to avoid an extreme/unreliable inference. In a word, partial pooling is very useful for analysis and with SSM and the Kalman filter, we could solve its relevant computation and make an inference easily.

Appendix A

Kalman filter

Given the content in Section 3.3, we shall show how to derive the Kalman filtering step by step based on the general expression of a state space model below. The whole process could also be found in Durbin and Koopman, 2012.

$$y_t = Z_t\alpha_t + \epsilon_t \qquad \epsilon_t \sim NID(0, H_t) \tag{A.1}$$

$$\alpha_{t+1} = T_t\alpha_t + R_t\eta_t \qquad \eta_t \sim NID(0, Q_t) \tag{A.2}$$

Before giving the derivation procedure, we post a known conclusion from multivariate analysis:

Lemma A.0.1. Suppose X and Y are jointly normally distributed as following,

$$E[(x \quad y)^T] = (\mu_x \quad \mu_y)^T \qquad Var\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_{yy} \end{pmatrix} \tag{A.3}$$

then the conditional distribution of X given Y is also normal with mean

$$E[x|y] = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y) \tag{A.4}$$

and variance matrix

$$Var[x|y] = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{xy}^T \tag{A.5}$$

A.1 Filtering process

It's not hard to show the expectation of v_t given Y_{t-1} is 0, then with Lemma A.0.1 applying on α_t and v_t given Y_{t-1}, we could show

$$a_{t|t} = E(\alpha_t|Y_{t-1}) + Cov(\alpha_t, v_t)Var(v_t)^{-1}v_t$$

where

$$\begin{aligned} Cov(\alpha_t, v_t) &= E(\alpha_t(Z_t\alpha_t + \varepsilon_t - Z_t a_t)'|Y_{t-1}) \\ &= E(\alpha_t(\alpha_t - a_t)'Z_t'|Y_{t-1}) \\ &= P_t Z_t' \\ Var(v_t|Y_{t-1}) &= Var(Z_t\alpha_t + \varepsilon_t - Z_t a_t|Y_{t-1}) \\ &= Z_t P_t Z_t' + H_t \\ &= F_t \end{aligned}$$

thereby,

$$a_{t|t} = a_t + P_t Z_t' F_t^{-1} v_t$$

Similarly, by Lemma A.0.1 we derive another update equation

$$\begin{aligned} P_{t|t} &= Var(\alpha_t|Y_t) \\ &= Var(\alpha_t|Y_{t-1}, v_t) \\ &= Var(\alpha_t|Y_{t-1}) - Cov(\alpha_t, v_t)Var(v_t)^{-1}Cov(\alpha_t, v_t)' \\ &= P_t - P_t Z_t' F_t^{-1} Z_t P_t \end{aligned}$$

Now let's look at how to predict the state at time t+1:

$$\begin{aligned} a_{t+1} &= E(\alpha_{t+1}|Y_t) \\ &= E(T_t\alpha_t + R_t\eta_t|Y_t) \\ &= T_t E(\alpha_t|Y_t) \\ &= T_t a_{t|t} \\ P_{t+1} &= Var(T_t\alpha_t + R_t\eta_t|Y_t) \\ &= T_t Var(\alpha_t|Y_t)T_t' + R_t Q_t R_t' \\ &= T_t P_{t|t} T_t' + R_t Q_t R_t' \end{aligned}$$

With update equations we obtained above and the Kalman gain $K_t = T_t P_t Z_t' F_t^{-1}$, we could have the final version of our prediction equation:

$$\begin{aligned} a_{t+1} &= T_t a_t + K_t v_t \\ P_{t+1} &= T_t P_t (T_t - K_t Z_t)' + R_t Q_t R_t' \end{aligned}$$

Sometimes Z_t, T_t, H_t, R_t and Q_t are time-invariant, then we can show that the variance matrix P_t converges to a constant matrix $\bar{P}$, which is the solution to

$$\bar{P} = T\bar{P}T' - T\bar{P}Z'\bar{F}^{-1}Z\bar{P}T' + RQR' \tag{A.6}$$

where $\bar{F} = Z\bar{P}Z' + H$.

A.2 Smoothing process

Define $x_t = \alpha_t - a_t$, then

$$v_t = y_t - Z_t a_t = Z_t(\alpha_t - a_t) + \varepsilon_t = Z_t x_t + \varepsilon_t \tag{A.7}$$

Meanwhile,

$$\begin{aligned} x_{t+1} &= \alpha_{t+1} - a_{t+1} \\ &= T_t\alpha_t + R_t\eta_t - T_t a_t - k_t v_t \\ &= T_t(\alpha_t - a_t) + R_t\eta_t - K_t Z_t x_t - K_t\varepsilon_t \\ &= T_t x_t + R_t\eta_t - K_t Z_t x_t - K_t\varepsilon_t \\ &= L_t x_t + R_t\eta_t - K_t\varepsilon_t \end{aligned} \tag{A.8}$$

where $L_t = T_t - K_t Z_t$

Denote $v_{t:n} = (v_t', \dots, v_n')$. We apply Lemma A.0.1 for α_t and $v_{t:n}$, then we have

$$\begin{aligned} \widehat{\alpha}_t &= E(\alpha_t|Y_n) = E(\alpha_t|Y_{t-1}, v_{t:n}) \\ &= a_t + \sum_{j=t}^{n} Cov(\alpha_t, v_j) F_j^{-1} v_j \end{aligned} \tag{A.9}$$

where

$$\begin{aligned} Cov(\alpha_t, v_j) &= E(\alpha_t v_j'|Y_{t-1}) - E(\alpha_t|Y_{t-1}) \cdot E(v_j'|Y_{t-1}) \\ &= E(\alpha_t \cdot (Z_j x_j + \varepsilon_j)'|Y_{t-1}) \\ &= E(\alpha_t \cdot x_j'|Y_{t-1}) \cdot Z_j' \end{aligned} \tag{A.10}$$

Meanwhile,

$$\begin{aligned} E(\alpha_t x_t'|Y_{t-1}) &= E(\alpha_t(\alpha_t - a_t)'|Y_{t-1}) = P_t \\ E(\alpha_t x_{t+1}'|Y_{t-1}) &= E(\alpha_t(L_t x_t + R_t \eta_t - K_t \varepsilon_t)'|Y_{t-1}) = P_t L_t' \\ &\vdots \\ E(\alpha_t x_n'|Y_{t-1}) &= P_t L_t' L_{t+1}' \cdots L_{n-1}' \end{aligned} \tag{A.11}$$

Substituting Equation A.9 with A.10 and A.11, we shall have

$$\widehat{\alpha}_n = a_n + P_n Z_n' F_n^{-1} v_n \tag{A.12}$$

$$\widehat{\alpha}_{n-1} = a_{n-1} + P_{n-1} Z_{n-1}' F_{n-1}^{-1} v_{n-1} + P_{n-1} L_n' Z_n' F_n^{-1} v_n \tag{A.13}$$

$$\vdots \tag{A.14}$$

$$\widehat{\alpha}_t = a_t + P_t Z_t' F_t^{-1} v_t + P_t L_t' Z_{t+1}' F_{t+1}^{-1} v_{t+1} + \cdots + P_t L_t' L_{t+1}' \cdots L_{n-1}' Z_n' F_n^{-1} v_n \tag{A.15}$$

Let $r_{n-1} = Z_n' F_n^{-1} v_n$, $r_{n-2} = Z_{n-1}' F_{n-1}^{-1} v_{n-1} + L_{n-1}' Z_n' F_n^{-1} v_n$, ..., $r_{t-1} = Z_t' F_t^{-1} v_t + L_t' Z_{t+1}' F_{t+1}^{-1} v_{t+1} + \cdots + L_t' L_{t+1}' \cdots L_{n-1}' Z_n' F_n^{-1} v_n$, then we could derive the relation between r_{t-1} and r_t is

$$r_{t-1} = Z_t' F_t^{-1} v_t + L_t' r_t \qquad where t = n, \dots, 1 \tag{A.16}$$

then we could substitute Equation A.9 with

$$\widehat{\alpha}_t = a_t + P_t r_{t-1} \tag{A.17}$$

$$r_{t-1} = Z_t' F_t^{-1} v_t + L_t' r_t \tag{A.18}$$

where $t = n, \dots, 1$ and $r_n = 0$. This is the derivation for the state smoothing process.

Now let's look at the variance matrix. We still rely on Lemma A.0.1, it's not hard to know

$$V_t = Var(\alpha_t|Y_{t-1}, v_{t:n}) = P_t - \sum_{j=t}^{n} Cov(\alpha_t, v_j) F_j^{-1} Cov(\alpha_t, v_j)' \tag{A.19}$$

then we could substitute with Equation A.10 and A.11 again, and repeat the similar smoothing treatment as what we did for the state α_t. In the end we could replace Equation A.19 with

$$V_t = P_t - P_t N_{t-1} P_t \tag{A.20}$$

$$N_{t-1} = Z_t' F_t^{-1} Z_t + L_t' N_t L_t \tag{A.21}$$

where $t = n, \dots, 1$ and $N_n = 0$. This is the smoothing process for the state variance matrix.

Appendix B

Other supplement

The following is the hypothesis test results between the weakly-informative prior and MLEs:

Friedman rank sum test
Friedman chi-squared = 583.7, df = 1, p-value < 2.2e-16

Wilcoxon rank sum test with continuity correction
W = 570603, p-value = 4.565e-08
alternative hypothesis: true location shift is not equal to 0

The decomposition error comparison among MLEs, the posterior estimators from weakly-informative and empirical priors is:

	MLE	Loss	MAP(hnormal)	MAP(empirical)
Median	761.9	645.2	715.2	709.2
Mean	785.2	657.1	733.3	728.6
sd	207.1	150.5	179.8	181.3

Table B.1: Information of decomposition error(2)

where *hnormal* is the weakly-informative prior half-normal distribution. Figure B.1 is the box plot of their decomposition errors:

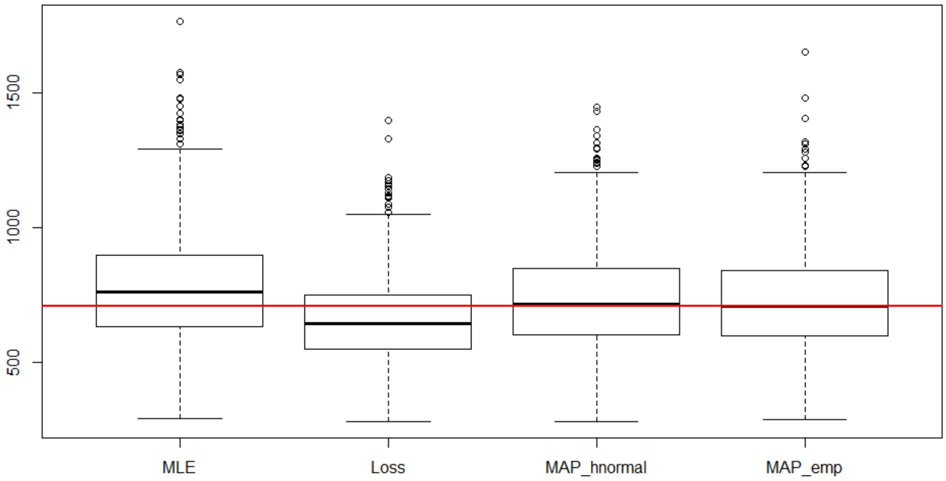


Figure B.1: Boxplot comparison of decomposition errors

Part of testing result w.r.t the posterior estimators from the empirical prior:

Friedman rank sum test
data: MLE,MAP(hnormal),MAP(empirical)
Friedman chi-squared = 690.82, df = 2, p-value < 2.2e-16

Friedman rank sum test
data: MAP(hnormal),MAP(empirical)
Friedman chi-squared = 65.536, df = 1, p-value = 5.706e-16

Friedman rank sum test
data: MLE,MAP(empirical)
Friedman chi-squared = 336.4, df = 1, p-value < 2.2e-16

Wilcoxon signed rank test with continuity correction
data: MLE,MAP(empirical)
V = 500500, p-value < 2.2e-16
alternative hypothesis: true location is not equal to 0

Wilcoxon signed rank test with continuity correction
data: MAP(hnormal),MAP(empirical)
V = 500500, p-value < 2.2e-16
alternative hypothesis: true location is not equal to 0

Bibliography

[Albert et al., 2012] Albert, I., Donnet, S., Guihenneuc-Jouyaux, C., Low-Choy, S., Mengersen, K., Rousseau, J., et al. (2012). Combining expert opinions in prior elicitation. *Bayesian Analysis*, 7(3):503–532.

[Anderson and Moore, 2012] Anderson, B. D. and Moore, J. B. (2012). *Optimal filtering*. Courier Corporation.

[Box and Jenkins, 1970] Box, G. E. and Jenkins, G. M. (1970). Time series analysis: Forecasting and control Holden-Day. *San Francisco*, page 498.

[Brockwell and Davis, 2016] Brockwell, P. J. and Davis, R. A. (2016). *Introduction to time series and forecasting*. springer.

[Caporello and Maravall, 2004] Caporello, G. and Maravall, A. (2004). Program tsw: Revised reference manual. *Banco de España*.

[Cleveland et al., 1990] Cleveland, R. B., Cleveland, W. S., McRae, J. E., and Terpenning, I. (1990). Stl: A seasonal-trend decomposition. *Journal of official statistics*, 6(1):3–73.

[Dagum, 1980] Dagum, E. (1980). The X-II-ARIMA seasonal adjustment method. *Statistics Canada*.

[Dagum and Bianconcini, 2016] Dagum, E. B. and Bianconcini, S. (2016). *Seasonal adjustment methods and real time trend-cycle estimation*. Springer.

[Durbin and Koopman, 2012] Durbin, J. and Koopman, S. J. (2012). *Time series analysis by state space methods*. Oxford university press.

[Ellis, 2015] Ellis, P. (2015). X13-SEATS-ARIMA as an automated forecasting tool.

[Findley et al., 1998] Findley, D. F., Monsell, B. C., Bell, W. R., Otto, M. C., and Chen, B.-C. (1998). New Capabilities and Methods of the X-12-ARIMA Seasonal-Adjustment Program. *Journal of Business I& Economic Statistics*, 16(2):127–152.

[Gelman, 2006] Gelman, A. (2006). Multilevel (hierarchical) modeling: what it can and cannot do. *Technometrics*, 48(3):432–435.

[Gelman et al., 2006] Gelman, A. et al. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian analysis*, 1(3):515–534.

[Gomez and Maravall, 1996] Gomez, V. and Maravall, A. (1996). Programs SEATS and TRAMO: Instructions for the User. *Bank of Spain*.

[Gómez and Maravall, 2001] Gómez, V. and Maravall, A. (2001). Seasonal adjustment and signal extraction in economic time series. *A course in time series analysis*, pages 202–247.

[Guerzhoy, 2016] Guerzhoy, M. (2016). STA303 Methods of Data Analysis II: Multilevel/Hierarchical Models.

[Harvey et al., 2018] Harvey, A., Ladiray, D., and etc, T. M. (2018). *Handbook on Seasonal Adjustment.* Eurostat.

[Harvey, 1990] Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter.* Cambridge university press.

[Helske, 2016] Helske, J. (2016). KFAS: Exponential family state space models in R. *arXiv preprint arXiv:1612.01907.*

[Hillmer and Tiao, 1982] Hillmer, S. C. and Tiao, G. C. (1982). An ARIMA-model-based approach to seasonal adjustment. *Journal of the American Statistical Association*, 77(377):63–70.

[Jazwinski, 2007] Jazwinski, A. H. (2007). *Stochastic processes and filtering theory.* Courier Corporation.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

[Koopman and Durbin, 2003] Koopman, S. J. and Durbin, J. (2003). Filtering and smoothing of state vector for diffuse state-space models. *Journal of Time Series Analysis*, 24(1):85–98.

[Ladiray and Quenneville, 2012] Ladiray, D. and Quenneville, B. (2012). *Seasonal adjustment with the X-11 method*, volume 158. Springer Science & Business Media.

[Levy, 2012] Levy, R. (2012). Probabilistic models in the study of language. *Online Draft, Nov.*

[Mangiafico, 2016] Mangiafico, S. (2016). Summary and analysis of extension program evaluation in r, version 1.15. 0. *URL https://rcompanion. org/handbook.*

[Monsell, 2007] Monsell, B. (2007). The X-13AS Seasonal Adjustment Program. *Bureau of the Census.*

[Protopapas, 2014] Protopapas, P. (2014). AM207: Lecture 19: Hidden Markov Models.

[Rabiner, 1989] Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Robert and Casella, 2013a] Robert, C. and Casella, G. (2013a). *Monte Carlo statistical methods.* Springer Science & Business Media.

[Robert and Casella, 2013b] Robert, C. and Casella, G. (2013b). *Monte Carlo statistical methods.* Springer Science & Business Media.

[Shiskin et al., 1967] Shiskin, J., Young, A., and Musgrave, J. (1967). The X-11 Variant of the Census Method II Seasonal Adjustment Program. *Bureau of the Census*, Technical Paper 15.

[Skagen, 1988] Skagen, D. (1988). Estimation of running frequency spectra using a Kalman filter algorithm. *Journal of biomedical engineering*, 10(3):275–279.

[Thacker and Lacey, 1998] Thacker, N. and Lacey, A. (1998). Tutorial: The kalman filter. *Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester*, page 61.

[Turner and Van Zandt, 2012] Turner, B. M. and Van Zandt, T. (2012). A tutorial on approximate bayesian computation. *Journal of Mathematical Psychology*, 56(2):69–85.

[Varadhan et al., 2016] Varadhan, R., Borchers, H. W., and Varadhan, M. R. (2016). Package 'dfoptim'.

[Wold, 1938] Wold, H. (1938). *A study in the analysis of stationary time series.* PhD thesis, Almqvist & Wiksell.

[Young et al., 1991] Young, P. C., Ng, C. N., Lane, K., and Parker, D. (1991). Recursive forecasting, smoothing and seasonal adjustment of non-stationary environmental data. *Journal of Forecasting*, 10(1-2):57–89.