Summer 2013

Modelling Locally Changing Variance Structured Time Series Data By Using Breakpoints Bootstrap Filtering

Rajan Lamichhane
Old Dominion University

MODELLING LOCALLY CHANGING VARIANCE STRUCTURED TIME SERIES DATA BY USING BREAKPOINTS BOOTSTRAP FILTERING

by

Rajan Lamichhane
B.S. January 1996, Tribhuvan University, Nepal
M.S. November 1998, Tribhuvan University, Nepal
M.S. May 2007, Southern Methodist University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MATHEMATICS AND STATISTICS

OLD DOMINION UNIVERSITY
August 2013

Approved by:

Norou Diawara (Director)

Larry D. Lee (Member)

Nak-Kyeong Kim (Member)

Cynthia M. Jones (Member)

ABSTRACT

MODELLING LOCALLY CHANGING VARIANCE STRUCTURED TIME SERIES DATA BY USING BREAKPOINTS BOOTSTRAP FILTERING

Rajan Lamichhane
Old Dominion University, 2013
Director: Dr. Norou Diawara

Stochastic processes have applications in many areas such as oceanography and engineering. Special classes of such processes deal with time series of sparse data. Studies in such cases focus in the analysis, construction and prediction in parametric models. Here, we assume several non-linear time series with additive noise components, and the model fitting is proposed in two stages. The first stage identifies the density using all the clusters information, without specifying any prior knowledge of the underlying distribution function of the time series. The effect of covariates is controlled by fitting the linear regression model with serially correlated errors. In the second stage, we partition the time series into consecutive non-overlapping intervals of quasi stationary increments where the coefficients shift from one stable regression relationship to a different one using a breakpoints detection algorithm. These breakpoints are estimated by minimizing the likelihood from the residuals. We approach time series prediction through the mixture distribution of combined error components. Parameter estimation of mixture distribution is done by using the EM algorithm. We apply the method to fish otolith data influenced by various environmental conditions and get estimation of parameters for the model.

ACKNOWLEDGEMENTS

I would like to express my sincere and deepest appreciation to my committee chair Dr. Norou Diawara. This dissertation would not have been possible if it were not by the guidance and motivation rendered to me by him. His consistent help, support and guidance during these three and half years were invaluable to me. His continuous efforts and encouragements to keep me working made all this possible. I could not have imagined having a better advisor and mentor for my Ph.D study.

Also, I would like to thank my committee members Dr. Larry D. Lee, Dr. Nak-Kyeong Kim and Dr. Cynthia M. Jones, for their encouragement, insightful comments, and support.

My Sincere thanks also goes to Dr. N. Rao Chaganty and the late Professor Dr. Dayanand N. Naik for thier invaluable contribution towards my study.

I thank my fellow colleagues and all those who contributed directly and indirectly towards my study and preparation of this dissertation during these years.

last but not the least, I would like to thank my family: my beloved wife Anju, for her love, sacrifice, support and good understanding and my children, Aryan and Aarshi for their unaccountable love. I dedicate this dissertation to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Stochastic processes for longitudinal data are fundamental in probability and statistics and have applications in many areas such as oceanography and engineering. Special classes of such processes deal with time series of sparse data. Studies in such cases focus on the analysis, construction and prediction in parametric models.

In this dissertation, the prediction of time series is revisited and an application based on real data is given. The density uses all the clusters information, without specifying any prior knowledge of the underlying distribution function of time series. The effect of covariates is controlled by fitting the linear regression model with serially correlated errors. The change in stability of regression coefficients during the time course can be accounted by creating different breakpoints. Theories suggest that the modeling of the time series should contain the information about the change intervals, and they should be considered as relevant in many predictions. When high frequency data is available, the prediction can be built on the last recorded values . However, in case of mismatch in recorded data, change in behaviors in temporal problems, one faces challenges as how to summarize the mixture of sample data, and how to make predictions. We partition the time course into consecutive non-overlapping intervals where the coefficients shift from one stable regression relationship to a different one. These breakpoints are estimated by minimizing the residual sum of squares (RSS) using the algorithm described by Bai and Peron (2003). The foundation for estimating breaks in time series regression models was given by Bai (1994) and was extended to multiple breaks by Bai (1997a, 1997b) and Bai and Perron (1998, 2003). The algorithm in selecting the number of change points is based on a simple iterative step in which the maximum difference is less than a critical value of the difference of two consecutive values and is less than an optimal threshold chosen in a Bayesian framework. The partition algorithm fits a different probability model maximizing likelihood within each block interval. In fact, the breakpoints allow us to find the range restriction as a frontier of the blocks.

Since different parts of data fit different models, forecasting depends not just on one model, but on all the relevant models. We develop a method based on mixture of

different distributions to forecast this type of models. The mixture model provides a convenient method to capture the data from different sources. However, such data is incompletely described since the source of the data is not known. The inference is valid only if there is a computationally tractable model to find where each observation originates. We propose a regression based approach in which the data for different blocks are fitted by using different models based on maximum likelihood method, therefore allowing us to focus on the independent errors of the model that we fit with an extended form of the full mixture model. In case, when there are very few observations in some intervals, we improve the parameter estimation by using block bootstrapping. The block bootstrap is the most general method to improve the accuracy of algorithm for the time series data of small scale. We use block bootstrap to generate bootstrap replicates of a statistic applied to time series.

By dividing the data into several blocks, original time series structure as well as the properties of original data generating process are preserved within a block. The Expectation-Maximization (EM) algorithm, with initial values obtained from the empirical estimates, give the estimates of the mixture distribution. Further improvement in the parameter estimation has been observed by using bootstrap re-sampling combined with EM algorithm. For simplicity, we name this method as BreakPoint Bootstrap Filtering (BPBF) method. As discussed earlier, using bootstrapping, we estimate the parameters of the model. Ghysels et al. (2006) show that estimates become inefficient when the mixed nature of the data is ignored, hence the advantage of using BPBF in improving model and forecasting.

This dissertation is an extension of the ideas developed akin to the cited references and related work. It presents a novel concept in time series prediction and some supporting empirical evidence in terms of real data. The concept of using multiple breakpoints based on minimum RSS or Bayesian Information Criteria (BIC) does not always create desirable partitioning of intervals. There could be very few observations in some intervals and the estimates based on those observations may be suspicious. In such cases, the estimation of parameters are improved by using block bootstrap. By dividing the data into different blocks, it can preserve the original time series structure within a block. However, the accuracy of the block bootstrap is sensitive to the choice of block length, and the optimal block length depends on the sample size, the data generating process and the statistic considered. In our examples, we are using the approach proposed by Patton et al. (2009) to identify the optimal block

size. Varying block lengths that follow the geometric distribution are considered, and thus we avoid the problem of non-stationary by its construction (Politis et al. 1994).

The dissertation shows that the dynamical model from different time series models with forecasting is stable and outpreforms classical inference. Over the last decade, there has been much interest in developing breakpoints on time series data in a small sample scale, but one must accomodate the prediction component. There is an ambiguity in the selection of the spliting of the time series and the coefficients used in the prediction are unstable if sub-intervals are not created. Herein we show the results from a combination of time series models, including the non-linearity of the data. We show through numerical simulation that our proposed model accomodates the true distribution closely better than the classical time series model.

The dissertation is organized as follows. Chapter 2 presents the guidelines and theory of the different procedures in model fitting. The distributions of the models are specified, and our new method is provided in Chapter 3. In Chapter 4, we apply our method to simulated and real data and get estimation of parameters as well as model forecasting. Conclusion is presented in Chapter 5.

CHAPTER 2

TIME SERIES AND OTHER LITERATURE REVIEWS

Partially observed time series models are studied under various conditions, e.g. state space models (Durbin and Koopman 2001), dynamic models (West and Harrison 1997), and hidden Markov models (Cappe et al. 2005). All of these methods work if we have regular time series data where the model structure does not change locally. In other words, if the variance changes locally, then it is hard to build the model based on regular time series approach. In many cases, structural changes or breaks appear to affect models. As for example some models related to the evolution in key economic and financial time series such as output growth, inflation, exchange rates, interest rates and stock returns.[1] If data are collected over a long period of time, we are more likely to observe the structural change. Such changes, also called breaks, could be the result of many possible factors such as institutional or technological changes, environmental changes, shifts in economic policy, or could even be due to large macroeconomic shocks such as the doubling or quadrupling of commodity prices experienced over the past decades.

Our main goal is to create reliable models and to incorporate these different model structures to estimate the overall model parameters and forecasting. We assume that if breaks have occurred in the past, surely they are also likely to happen in the future. Approaches that view breaks as being generated deterministically are not applicable when forecasting future events unless, of course, future break dates as well as the size of such breaks are known in advance. In most applications, modelling of the stochastic process underlying the breaks is needed. One of our goals is to create a mixture model for forecasting conditional on the past values. We use a restricted form where the weights are specified from the sizes of the blocks which are constructed using appropriate empirical distribution and smoothness of the data. In this work, we provide a general framework for forecasting time series under structural breaks that is capable of handling the different above scenarios.

[1] A small subset of the many papers that have reported evidence of breaks in economic and financial time series includes Alogouskofis and Smith (1991), Garcia and Perron (1996), Koop and Potter (2001), and Pastor and Stambaugh (2001).

2.1 TIME SERIES ANALYSIS

A time series is a sequence of observations over a time interval $T = [0, t), t > 0$ taken sequentially in time denoted as $\{Y_t\}_{t\epsilon T}$. An intrinsic feature of time series is that, typically, adjacent observations are dependent. Time series analysis is concerned with techniques for the analysis of this dependence. There are many techniques used to analyze the time series data. In this section, we present the time series data and the modelling techniques. Our main concern is in discrete time series in which the set of times at which the observations are made is a discrete set.

A time series linear model of responses $\{Y_t\}$ based on predictor X_t can be defined as

$$Y_t = \beta_0 + \beta_1 X_t + \zeta_t, \qquad t\epsilon T$$

where β_0 and β_1 are the regression coefficients and ζ are the error components. The residuals ζ's are not independent and assume stationarity of the process. $(\zeta_t)_{t\epsilon T}$ could be white noise (WN) or the Gaussian noise. WN are assumed to be a sequence of independent random variables with uniform probability distribution while Gaussian noises are generated from Gaussian distribution. Most of the time, we assume that $(\zeta_t)_{t\epsilon T}$ are from a Gaussian distribution.

A time series $\{Y_t\}$ is said to be weakly stationary (also known as second order stationary or covariance stationary) if and only if

(1) the mean exists, is finite, and does not depend on time t

(2) the covariance $Cov(Y_i, Y_j)$ depend on the absolute value of the lag $h = |i - j|, i.e.$

$$Cov(Y_i, Y_j) = \gamma(h) = Cov(Y_j, Y_i).$$

This implies that $\gamma(i, i) = \sigma^2$ exists for all $i\epsilon T$.

A stronger property is that all higher order moments exist and are constant. A time series process $\{Y_t\}$ is strictly stationary if the joint distribution of $(Y_1, \ldots, Y_n)$ and $(Y_{h+1}, \ldots, Y_{h+n})$ are same for all integers h and $n > 0$. We can also show that for finite second moment, the strictly stationary process is also weakly stationary.

Theorem 1. *If $\{Y_t\}$ is strictly stationary and $E(Y_t^2) < \infty$ for all t, then $\{Y_t\}$ is also weakly stationary.*

Proof. Let $\{Y_t\}$ be strictly stationary process. Then,

$(Y_1, ..., Y_n) \xrightarrow{d} (Y_{1+h}, ..., Y_{n+h})$ for all integers h and $n > 0$.

For $n = 1$, we have $Y_1 \xrightarrow{d} Y_{1+h}$ for all h.

So, the random variables Y_t are identically distributed for all t.

Hence, $E(Y_t)$ is constant and independent of time t.

Again, $E(Y_t^2) < \infty$ so all covariances exist.

Also for $n = 2$, $(Y_1, Y_2) \xrightarrow{d} (Y_{1+h}, Y_{2+h})$ for all integers h.

So the pairs of random variables (Y_t, Y_{t+h}) are identically distributed for all t and h. Hence, $cov(Y_t, Y_{t+h}) = cov(Y_1, Y_{1+h})$ and it is independent of t. So the process $\{Y_t\}$ is weakly stationary.

□

In our case, we are dealing with discrete time series with finite second moment. So whenever we use the term *stationary*, we shall mean weakly stationary. The class of linear time series models, which includes the class of autoregressive moving-average (ARMA) models, provides a general framework for studying stationary processes. In fact, Wold decomposition shows that every second order stationary process is either a linear process or can be transformed to a linear process by subtracting *deterministic* component (Casella et al. 2002).

Let $\{Y_t\}$ be a stationary time series process then
an $AR(1)$ model can be represented as:

$$Y_t - \phi_1 Y_{t-1} = Z_t; \qquad Z_t \sim WN(0, \sigma^2).$$

where WN represents the white noise.
$AR(p)$ can be represented as

$$Y_t - \phi_1 Y_{t-1} - \ldots - \phi_p Y_{t-p} = Z_t; \qquad Z_t \sim WN(0, \sigma^2).$$

A $MA(1)$ can be written as

$$Y_t = Z_t - \theta_1 Z_{t-1}; \qquad Z_t \sim WN(0, \sigma^2).$$

and a $MA(q)$ is

$$Y_t = Z_t - \theta_1 Z_{t-1} - \ldots - \theta_q Z_{t-q}; \qquad Z_t \sim WN(0, \sigma^2).$$

In general $AR(p)$ and $MA(q)$ can be combined thru an autoregressive moving average(ARMA) process for the time series $\{Y_t\}$ of order (p, q), ARMA(p, q), can be represented as

$$Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - ... - \phi_p Y_{t-p} = Z_t - \theta_1 Z_{t-1} - ... - \theta q Z_{t-q}, \tag{1}$$

with $\{Z_t\}$ being the white noise of the process and $\phi_1, ..., \phi_p$ and $\theta_1, ..., \theta_q$ are AR and MA components, respectively.

From 1

$$\phi(B)Y_t = \theta(B)Z_t \ , \tag{2}$$

where $\phi(B) = 1 - \sum_{j=1}^{p} \phi_j B^j$; $\theta(B) = 1 - \sum_{j=1}^{q} \theta_j B^j$ and B is the back shift operator such that $BY_t = Y_{t-1}$ and $B^j Y_t = Y_{t-j}; j \geq 0$.

Hence, from equation 2

$$Y_t = \frac{1}{\phi(B)}\theta(B)Z_t \ ,$$

$$Y_t = \frac{1}{\left(1 - \sum_{j=1}^{p} \phi_j B^j\right)}(1 - \sum_{j=1}^{q} \theta_j B^j)Z_t \ ,$$

Using the power series expansion of

$$\frac{b_0}{b_0 + b_1 x + \ldots + b_m x^m} = a_0 + a_1 x + \ldots + a_n x^n + \ldots$$

where $a_n = -\frac{b_1}{b_0}a_{n-1} - \frac{b_2}{b_0}a_{n-2} - \ldots - \frac{b_m}{b_0}a_{n-m}, n \geq 1,$

We can write,

$$\frac{1}{\phi(B)} = \sum_{j=0}^{\infty} \chi_j B^j = \chi(B).$$

Hence,

$$\begin{aligned} Y_t &= \chi(B)\theta(B)Z_t \ , \\ &= \psi(B)Z_t \ , \\ &= \sum_{j=0}^{\infty} \psi_j Z_{t-j}. \end{aligned} \tag{3}$$

For example, let's consider an AR(1) process

$$\begin{aligned} Y_t &= \phi_1 Y_{t-1} + Z_t \\ &= \phi_1(\phi_1 Y_{t-2} + Z_{t-1}) + Z_t \\ &\vdots \\ &= \sum_{j=0}^{\infty} \phi_1^j Z_{t-j} \\ &= \sum_{j=0}^{\infty} \psi_j Z_{t-j}. \end{aligned}$$

where $\psi_j = \phi_1^j$.

An ARMA(p, q) process $\{Y_t\}$ is said to be causal, if we can express the values Y_t in terms of white noises Z_t only. From equation 3, if there exist constants ψ_j such that $\sum_{j=0}^{\infty} |\psi_j| < \infty$ then the process $\{Y_t\}$ is causal.

Throughout our discussions, we assume that an ARMA(p, q) process $\{Y_t\}$ is causal and it has an unique stationary solution.

Also, we can further extend the model to nonstationary time series. A more general linear model for such process is autoregressive integrated moving average, ARIMA (p, d, q), where Y_t satisfies a difference equation of the form

$$\phi(B)(1-B)^d Y_t = \theta(B) Z_t, \ \{Z_t\} \ \sim WN(0, \sigma^2),$$

with $\phi(z)$ and $\theta(z)$ are polynomials of degrees p and q, respectively,
$\phi(z) \neq 0$ for $|z| \leq 1$, d is the difference indicator and B is the backshift operator. For $d = 0$, an ARIMA(p, d, q) reduces to an ARMA(p, q) process.

2.1.1 FORECASTING USING ARMA MODEL

The simplest and most elegant approach of forecasting stationary time series is based on minimum mean square error (Box et al. 1994). Minimum mean square error may be generated directly from the *difference equation* form of the model. Our goal is to find the *linear combination* of $1, Y_n, ..., Y_1$, that forecasts Y_{n+h} with minimum mean squared error. The best linear predictor in terms of $1, Y_n, ..., Y_1$, will be denoted by $P_n Y_{n+h}$ and clearly has the form

$$P_n Y_{n+h} = a_0 + a_1 Y_n + ... + a_n Y_1. \tag{4}$$

The coefficients $a_0, ..., a_n$ can be determined by finding the values that minimize

$$S(a_0, ..., a_n) = E(Y_{n+h} - a_0 - a_1 Y_n - ... - a_n Y_1)^2.$$

Since, S is a quadratic function of $a_0, ..., a_n$ and is bounded below by zero, it is clear that there is at least one value of $(a_0, ..., a_n)$ that minimizes S and that the minimum $(a_0, ..., a_n)$ satisfies the equations

$$\frac{\partial S(a_0, ..., a_n)}{\partial a_j} = 0, \ j = 0, ..., n.$$

This gives the *difference equations*

$$E\left[Y_{n+h} - a_0 - \sum_{i=1}^{n} a_i Y_{n+1-i}\right] = 0, \tag{5}$$

$$E\left[\left(Y_{n+h} - a_0 - \sum_{i=1}^{n} a_i Y_{n+1-i}\right) Y_{n+1-j}\right] = 0, \ j = 1, ..., n. \tag{6}$$

From Equation 5,

$$E(Y_{n+h}) - a_0 - E\left(\sum_{i=1}^{n} a_i Y_{n+1-i}\right) = 0,$$

For any stationary process with $E(Y_t) = \mu$,

$$\mu - a_0 - \sum_{i=1}^{n} a_i \mu = 0$$

$$a_0 = \mu(1 - \sum_{i=1}^{n} a_i)$$

and from equation 6, substituting value of a_0 from 5

$$E\left[\left(Y_{n+h} - \mu(1 - \sum_{i=1}^{n} a_i) - \sum_{i=1}^{n} a_i Y_{n+1-i}\right) Y_{n+1-j}\right] = 0, \ j = 1, ..., n.$$

$$E\left[\left((Y_{n+h} - \mu) - \sum_{i=1}^{n} a_i (Y_{n+1-i} - \mu)\right) Y_{n+1-j}\right] = 0$$

$$E(Y_{n+h} - \mu, Y_{n+1-j}) - [a_1 E(Y_n - \mu, Y_{n+1-j}) +$$

$$a_2 E(Y_{n-1} - \mu, Y_{n+1-j}) + ... + a_n E(Y_1 - \mu, Y_{n+1-j})] = 0.$$

Here,

$$E[(Y_{n+h} - \mu) Y_{n+1-j}] = \gamma(h + j - 1), j = 1, 2, ..., n.$$

Substituting $j = 1, 2, ..., n$, we get

$$E[(Y_{n+h} - \mu)Y_{n+1-j}] = \gamma_n(h) = \begin{pmatrix} \gamma(h) \\ \gamma(h+1) \\ \vdots \\ \gamma(h+n-1) \end{pmatrix}.$$

where $\gamma(h) = cov(Y_n, Y_{n+h})$.

Also,

$$\sum_{i=1}^{n} a_i E\left[(Y_{n+1-i} - \mu)Y_{n+1-j}\right] = \sum_{i=1}^{n} a_i \gamma(j-i), \;\; j = 1, 2, ..., n.$$

This can be expressed as

$$\Gamma_n \boldsymbol{a} = \begin{bmatrix} \gamma(0) & \gamma(1) & ... & \gamma(n-1) \\ \gamma(1) & \gamma(0) & ... & \gamma(n-2) \\ \vdots & & & \\ \gamma(n-1) & \gamma(n-2) & ... & \gamma(0) \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}.$$

Hence, we have a system of n linear equations in the form

$$\Gamma_n \boldsymbol{a} = \gamma_n(h). \tag{7}$$

However, the direct approach requires the determination of a solution of these n linear equations, which for large n may be difficult and time consuming. For general stationary processes it would be helpful if the one-step predictor $P_n Y_{n+1}$ based on n previous observations could be used to simplify the calculation of $P_{n+1}Y_{n+2}$, the one-step predictor based on $n+1$ previous observations. Prediction algorithms that utilize this idea are said to be recursive. The most common recursive algorithm use in time series analysis is the Durbin-Levinson algorithm and innovations algorithm (Brockwell et al. 2002). Durbin-Levinson algorithm expresses one-step predictor in terms of previous observations $Y_1, Y_2, ..., Y_n$ while innovations algorithm expresses one-step predictor in terms of previous innovations, $Y_1 - \hat{Y}_1, Y_2 - \hat{Y}_2, ..., Y_n - \hat{Y}_n$, that are uncorrelated.

From 16, we have

$$P_n Y_{n+h} = \mu + \sum_{i=1}^{n} a_i (Y_{n+1-i} - \mu).$$

Hence, the expected value of prediction error

$$E(Y_{n+h} - P_n Y_{n+h}) = E(Y_{n+h}) - \mu - \sum_{i=1}^{n} a_i E(Y_{n+1-i} - \mu)$$
$$= 0.$$

And the mean square prediction error is therefore

$$\begin{aligned} E(Y_{n+h} - P_n Y_{n+h})^2 &= E(Y_{n+h})^2 - 2E\left[(Y_{n+h} P_n Y_{n+h}\right] + E(P_n Y_{n+h})^2 \\ &= \gamma(0) - 2\sum_{i=1}^{n} a_i E(Y_{n+h} Y_{n+1-i}) + E\left(\sum_{i=1}^{n} a_i Y_{n+1-i}\right)^2 \\ &= \gamma(0) - 2\sum_{i=1}^{n} a_i \gamma(h+i-1) + \sum_{i=1}^{n}\sum_{j=1}^{n} a_i \gamma(i-j) a_j \\ &= \gamma(0) - 2\boldsymbol{a}'\gamma_n(h) + \boldsymbol{a}'\Gamma_n \boldsymbol{a} \\ &= \gamma(0) - 2\boldsymbol{a}'\gamma_n(h) + \boldsymbol{a}'\gamma_n(h) \qquad [\text{ From 7}] \\ &= \gamma(0) - \boldsymbol{a}'\gamma_n(h). \end{aligned}$$

Each predictor variables $Y_j, j = 1, 2, ..., n$ is uncorrelated with prediction error $(Y_{n+h} - \hat{Y}_{n+h})$ where $\hat{Y}_{n+h}$ is the prediction based on past Y_{n+h-1} values, so Y_j uniquely determines $P_n Y_{n+h}$. As discussed earlier, the most common recursive prediction algorithm that utilize the one-step prediction ideas are Durbin-Levinson algorithm and innovations algorithm. In our discussion we use innovations algorithm since it is applicable to all series with finite second moments, regardless of whether they are stationary or not. Also, innovations algorithm works for mixed models with both autoregressive and moving components while Durbin-Levinson algorithm works for purely autoregressive process. Also, for the state space model Kalman filter equations or simply Kalman filtering is used to define the finite sample optimal (minimum mean square error matrix) estimate of the state vector based on the observations over the finite past time period. Since, model building based on state space model or ARMA are equivalent, ARMA model can be expressed in the state space form and state space model can be transferred to ARMA model. So, in our discussion we focus on the model based on ARMA whose parameters are estimated by using the recursive method called innovations algorithm.

2.1.2 INNOVATIONS ALGORITHM

Suppose that $\{Y_t\}$ is a zero mean time series with finite second moment, $E|Y_t^2| < \infty$, for each t and let

$$E(Y_iY_j) = \kappa(i,j)$$

Let the best one step predictor,

$$\hat{Y}_n = \begin{cases} 0, & \text{if n=1,} \\ P_{n-1}Y_n, & \text{if n=2,3,...,} \end{cases}$$

and the mean squared error

$$v_n = E(Y_{n+1} - P_nY_{n+1})^2$$

Let $U_n = Y_n - \hat{Y}_n$ be the one step prediction error or innovations. Then,

$$\begin{aligned} U_n &= Y_n - P_{n-1}Y_n \\ &= Y_n - (a_1Y_{n-1} + \ldots + a_{n-1}Y_1) \qquad [\because From4] \end{aligned}$$

If $\boldsymbol{U}_n = (U_1, U_2, \ldots, U_n)'$ and $\boldsymbol{Y}_n = (Y_1, Y_2, \ldots, Y_n)'$ then

$$\boldsymbol{U}_n = A_n\boldsymbol{Y}_n, \tag{8}$$

where

$$A_n = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ a_{11} & 1 & 0 & \ldots & 0 \\ a_{22} & a_{21} & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ a_{n-1,n-1} & a_{n-1,n-2} & a_{n-1,n-3} & \ldots & 1 \end{bmatrix}.$$

Since we are using one step prediction, $a_{ij} = -a_j$ as discussed in equation 4 for lag $h = 1$. Also, A_n is non-singular so it's inverse exists.
Define,

$$C_n = A_n^{-1} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ \theta_{11} & 1 & 0 & \ldots & 0 \\ \theta_{22} & \theta_{21} & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \theta_{n-1,n-1} & \theta_{n-1,n-2} & \theta_{n-1,n-3} & \ldots & 1 \end{bmatrix},$$

where θ_{ij} are the ij^{th} components of A_n^{-1}.

Hence, From equation 8

$$\boldsymbol{Y}_n = C_n \boldsymbol{U}_n. \tag{9}$$

Again,

$$\begin{aligned}
\hat{\boldsymbol{Y}}_n &= \boldsymbol{Y}_n - \boldsymbol{U}_n, \\
&= C_n \boldsymbol{U}_n - \boldsymbol{U}_n, && \text{From 9} \\
&= (C_n - I_n)\boldsymbol{U}_n, && I_n \text{ is the identity matrix} \\
&= (C_n - I_n)(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n), && (10) \\
\implies \hat{\boldsymbol{Y}}_n &= \boldsymbol{\Theta}_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n). && (11)
\end{aligned}$$

where

$$\boldsymbol{\Theta}_n = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ \theta_{11} & 0 & 0 & \dots & 0 \\ \theta_{22} & \theta_{21} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ \theta_{n-1,n-1} & \theta_{n-1,n-2} & \theta_{n-1,n-3} & \dots & 0 \end{bmatrix}.$$

So, equation 11 can be written as

$$\hat{Y}_{n+1} = \begin{cases} 0, & \text{if } n = 0, \\ \sum_{j=1}^{n} \theta_{nj} \left(Y_{n+1-j} - \hat{Y}_{n+1-j} \right), & \text{if } n = 1, 2, \dots, \end{cases}$$

Once the coefficients θ_{ij} have been determined, the one step predictors $\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_n$ can be computed, recursively. The following steps show how the coefficients $\theta_{n1}, \dots, \theta_{nn}$ are computed recursively:

Step I: Estimate the initial value of mean squared error,

$$\upsilon_0 = \kappa(1,1) = Var(Y_1) = \sigma^2.$$

Step II: Calculate

$$\theta_{n,n-k} = \upsilon_k^{-1} \left(\kappa(n+1, k+1) - \sum_{j=0}^{k-1} \theta_{k,k-j} \theta_{n,n-j} \upsilon_j \right), 0 \le k < n, and$$

$$\upsilon_n = \kappa(n+1, n+1) - \sum_{j=0}^{n-1} \theta_{n,n-j}^2 \upsilon_j.$$

So, we solve first for v_0, then successively for $\theta_{11}, v_1; \theta_{22}, \theta_{21}, v_2; \theta_{33}, \theta_{32}, \theta_{31}, v_3; \ldots$.

For the causal ARMA process

$$\phi(B)Y_t = \theta(B)Z_t, \qquad \{Z_t\} \sim WN(0, \sigma^2).$$

it is possible to simplify the application of the innovations algorithm . The idea is to not apply it to the process $\{Y_t\}$ itself but to some transferred process $\{W_t\}$ (Ansley 1979).

Let

$$W_t = \begin{cases} \sigma^{-1}Y_t, & t = 1, 2, \ldots, m, \\ \sigma^{-1}\phi(B)Y_t, & t > m,. \end{cases} \tag{12}$$

where m=max(p, q).

Applying the innovations algorithm to the process $\{W_t\}$ we obtain

$$\hat{W}_{n+1} = \begin{cases} \sum_{j=1}^{n} \theta_{nj}(W_{n+1-j} - \hat{W}_{n+1-j}), & 1 \leq n < m, \\ \sum_{j=1}^{q} \theta_{nj}(W_{n+1-j} - \hat{W}_{n+1-j}), & n \geq m, \end{cases} \tag{13}$$

Also,

$$Y_t - \hat{Y}_t = \sigma(W_t - \hat{W}_t) \text{ for all } t \geq 1. \qquad [\because \text{From } II.1.11]$$

Replacing $W_t - \hat{W}_t$ by $\sigma^{-1}(Y_t - \hat{Y}_t)$ and simplifying further, we get

$$\hat{Y}_{n+1} = \begin{cases} \sum_{j=1}^{n} \theta_{nj}(Y_{n+1-j} - \hat{Y}_{n+1-j}), & 1 \leq n < m, \\ \phi_1 Y_n + \ldots + \phi_p Y_{n+1-p} + \sum_{j=1}^{q} \theta_{nj}(W_{n+1-j} - \hat{W}_{n+1-j}), & n \geq m, \end{cases} \tag{14}$$

and the mean squared errors

$$E(Y_{n+1} - \hat{Y}_{n+1})^2 = \sigma^2 E(W_{n+1} - \hat{W}_{n+1})^2 = \sigma^2 \boldsymbol{r}_n. \tag{15}$$

2.2 BREAKPOINTS

If the structure of data is such that there is heterogeneous variance structure among different intervals, then parameter estimates based on a regular time series

model is very unrealistic. It is then paramount to assess threshold (or breakpoints) where the effect of the covariates changes. So we divide the data into different parts using multiple breakpoints. The foundation for estimating breaks in time series regression models was given by Bai (1994) and was extended to multiple breaks by Bai (1997a, 1997b) and Bai and Perron (1998). The distribution function used for the confidence intervals for the breakpoints is given in Bai (1997b). The ideas behind this implementation are described in Zeileis et al. (2002). We use the *strucchange* package in R developed by the above mentioned authors to find the breakpoints via a grid search algorithm.

The breakpoints are obtained by testing or assessing deviations from stability in the classical linear regression model. Following idea from Tiwari et al. (2005), we use BIC under MCMC simulation instead of just MCMC simulation, incorporating time factor. Linear regression of covariates is written as:

$$y_j = x_j^T \boldsymbol{\beta} + u_j,$$

where at time j, y_j is the observation of the dependent variable, $x_j = (1, x_{j1}, ..., x_{jk})^T$ is a $(k+1) \times 1$ vector of observations of the independent variables, and u_j are *iid* with 0 mean and variance σ^2, and $\boldsymbol{\beta}$ is the $(k+1) \times 1$ vector of regression coefficients. In many applications, it is reasonable to assume that there are m breakpoints, where the coefficients shift from one stable regression relationship to a different one. Thus, there are $m+1$ segments, $I_1, \cdots, I_{m+1}$ in which the regression coefficients are constant, and the model can be rewritten as:

$$y_j = x_j^T \boldsymbol{\beta_i} + u_j, \tag{16}$$

where $\boldsymbol{\beta_i}, i = 1, 2, \cdots, m+1$ is the vector of regression coefficients within each segment, i denotes the segment index and $j = j_{i-1}+1, ..., j_i$. In practice, the breakpoints are rarely given exogenously, but have to be estimated. They are estimated by minimizing the residual sum of squares (RSS) from equation (16). The algorithm for computing the optimal breakpoints given the number of breaks is based on a dynamic programming approach based on the Bellman principle (Bellman 1952). The main computational effort is to compute a triangular RSS matrix, which gives the RSS for a segment starting at observation indexed j and ending at indexed j' with $j < j'$. Also, the adjacent intervals separated by breakpoints are significantly different. When identifying the breakpoints, we look for both minimum RSS and BIC

associated with the number of breakpoints. A careful consideration should be taken when identifying the breakpoints, since we do not want to divide the data into several small intervals. That it creates a problem of overfitting and overparametrization.

2.3 BLOCK BOOTSTRAP

The bootstrap is a simulation approach to estimate the distribution of test statistics. The original method of bootstrap which was first proposed by Efron (1979) is to create bootstrap samples by resampling the data randomly, and then constructs the associated empirical distribution function. Often, the original bootstrap methods provides improvements to the poor asymptotic approximations when data are independently and identically distributed. However, the performance of the original procedure can be far from satisfactory for time series data with serial correlation and heteroscedsticity of unknown form. The block bootstrap is the most general method to improve the accuracy of bootstrap for the time series data of small scale. We use block bootstrap to generate bootstrap replicates of a statistic applied to time series. By dividing the data into several blocks, original time series structure as well as the properties of original data generating process are preserved within a block.

Let $\{Y_t : t = 1, ..., n\}$ be time series data then we construct bootstrap sample in the following steps:

1. Start by 'wrapping' the data $\{Y_1, ..., Y_n\}$ around a circle.

2. Let $i_0, i_1, ...$ be drawn i.i.d. with uniform distribution on the set $\{1, 2, ..., n\}$; these are the starting points of the new blocks.

3. Pick the optimal block size, l. The accuracy of the block bootstrap is sensitive to the choice of block length, and the optimal block length depends on the sample size, the data generating process, and the statistic considered. Patton et al. (2009) suggested the estimators of optimal block size for block bootstrap methods. These estimators are based on the notion of spectral estimation and are characterized by the fastest possible rate of convergence which is adaptive on the strength of the correlation of the time series. A simple criterion for selecting block size is that it should be at least equal to a cube root of total sample size.

4. Resample the blocks randomly with replacement and generate the bootstrap

sample. The blocks may be overlapping or non-overlappping. According to Lahiri (1999) and Andrews (2002), there is little difference in performance for these two methods. For the overlapping method, we divide the data into $n - l + 1$ blocks, which first block being $\{Y_1, Y_2, ..., Y_l\}$, second block being $\{Y_2, Y_3, ..., Y_{l+1}\}, ..., etc.$ For the non-overlapping method , we divide the data into n/l blocks, which block 1 being $\{Y_1, Y_2, ..., Y_l\}$, block 2 being $\{Y_{l+1}, Y_{l+2}, ..., Y_{2l}\}, ..., etc.$ In our analysis we use overlaping blocks with varying block lengths. Rather than assuming the fixed block length, we assume that the block lengths are the random variables from geometric distribution such that the optimal block size l is the mean of geometric distribution used to generate the block length. This avoids the problem of non-stationarity by construction (Politis and Romano 1994). The resampled blocks are glued together in the order that they were sampled to generate bootstrap sample $\{Y_t^* : t = 1, \cdots, n\}$.

5. Calculate the estimator $\hat{\theta}^{(b)}, b = 1, 2, ..., B$ for all B bootstrap samples. Notice that the bootstrap distribution of $\hat{\theta}_B - \hat{\theta}$ approximates the sampling distribution of $\hat{\theta} - \theta$ fairly well.

   The bootstrap estimate of standard error of an estimator $\hat{\theta}$ is the sample standard deviation of the bootstrap replicates $\hat{\theta}^{(1)}$, ... , $\hat{\theta}^{(B)}$.

$$\hat{se}(\hat{\theta}^*) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}^{(b)} - \overline{\hat{\theta}^*})^2}$$

   where

$$\overline{\hat{\theta}^*} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^{(b)}.$$

In our analysis, we perform block bootstrapping over different groups of time series data which are separated by breakpoints. We call this combination of breakpoints together with bootstrapping as breakpoints Bootstrap Filtering (BPBF) method.

2.4 THE EM ALGORITHM

An expectation-maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a

function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step. The EM algorithm has the ability to deal with missing data and unidentified variables, so it is becoming useful in a variety of incomplete-data problems. EM algorithm was explained and given its name in a classic paper by Dempster et al. (1977).

The EM algorithm has two main applications. The first case occurs when the data has missing values due to limitations or problems with the observation process. The second case occurs when the likelihood function can be obtained and simplied by assuming that there is an additional but missing parameters.

With missing values or parameters in the data which is generated by some distribution under assumption, we call the data, X, the incomplete data. And, we assume that the complete data, $Z = (X; Y)$ exists with Y being missing data and that a joint density function also exists as follows:

$$p(z|\theta) = p(x, y|\theta) = p(y|x, \theta) * p(x|\theta)$$

where θ is a set of unknown parameters from a distribution including a missing parameter.
With the density function, we now define the complete-data likelihood as:

$$L(\theta|Z) = L(\theta|X, Y) = p(X, Y|\theta).$$

And, the original likelihood $L(\theta|X)$ is called the incomplete-data likelihood function. Since the missing data Y is unknown under a certain distribution by assumption, we can think of $L(\theta|X, Y)$ as a function of a random variable, Y , with constant values, X and θ, i.e

$$L(\theta|X, Y) = f_{(x,\theta)}(Y).$$

Using the complete-data log-likelihood function with respect to the missing data Y given the observed data X, the EM algorithm finds its expected value as well as the current parameter estimates at the E Step and maximizes the expectation at the M step. By repeating the E and M step, the algorithm is guaranteed to converge to

a local maximum of the likelihood function with each iteration increasing the log-likelihood.
As mentioned before, EM algorithm involves two steps E and M steps.

2.4.1 EXPECTATION (E) STEP

The complete data sufficient statistics at k cycle is given by the expectation function at k^{th} cycle. The expectation function of the complete data log-likelihood can be defined as

$$Q(\theta; \theta^{(k-1)}) = E[logp(X, Y|\theta)|X, \theta^{(k-1)}],$$

where $\theta^{(k-1)}$ is a set of current parameter estimates after $(k-1)$ cycles that we use to evaluate the expectation and to increase Q with the new θ for optimization. Here, X and $\theta^{(k-1)}$ are known constants and θ is a variable to be adjusted. Since Y is a missing random variable under an assumed distribution, $f(y|X, \theta^{(k-1)})$, the expectation function can be written as:

$$E[logp(X, Y|\theta)|X, \theta^{(k-1)}] = \int_{y\epsilon\Omega} logp(X, y|\theta) * f(y|X, \theta^{(k-1)})dy.$$

where Ω is the space of values where y can take values on and $f(y|X, \theta^{(k-1)})$ is the marginal distribution of the missing data Y depending on observed data and current parameters at $(k-1)$ cycle.

2.4.2 MAXIMIZATION (M) STEP

At the M step, we maximize the expectation we obtain in the E step.

$$\theta^{(k)} = argmax_{\theta} Q(\theta; \theta^{(k-1)}).$$

Maximization step becomes either easy or hard depending on the form of $p(X, y|\theta)$. When the underlying complete data come from an exponential family whose maximum-likelihood estimates are easily computed, then each maximum step of an EM algorithm is likewise easily computed (Dempster et al. 1977).

2.4.3 BOOTSTRAPPING IN EM

Since the likelihood of the segmented data may not be concave (i.e. it may not converge to an unique value) bootstrapping technique offers one option to consider.

The idea is that the data obtained within that interval can be regenerated to get a larger sample size.

2.5 MIXTURE DISTRIBUTION

The idea of the mixture distribution arises when we draw the random variables from more than one parent population. Mixture distributions represent a useful way of describing heterogeneity in the distribution. It arises frequently in practice, particularly in cases where we are observing a composite response from multiple distinct sources. The response x that we observe is modeled as a random variable that has some probability p_1 of being drawn from distribution f_1, probability p_2 of being drawn from distribution f_2, and so forth, with probability p_r of being drawn from distribution f_r, where r is the number of components in our mixture distribution. The key assumption here is one of statistical independence between the process of randomly selecting the component distribution $f_j, j = 1, 2, ..., r$ to be drawn and these distributions themselves. That is, we assume there is a random selection process that first generates the numbers 1 through r with probabilities p_1 through p_r. Then, once we have drawn some number j, we turn to distribution f_j and draw the random variable x from this distribution. So as long as the probabilities or mixing percentages p_j sum to 1, and all of the distributions f_j are proper densities, the combination also defines a proper probability density function, which can be used as the basis for computing expectations, formulating maximum likelihood estimation problems, and so forth.

Let's assume that a sample $X_1, X_2, ..., X_n$ comes from r mixture distributions with density function

$$f(x|\boldsymbol{p}, \boldsymbol{\theta}) = \sum_{j=1}^{r} p_j f_j(x|\boldsymbol{\theta}_j),$$

where $\boldsymbol{\theta}_j$ be the vector of unknown parameters for each distribution f_j and the weights $0 \leq p_j \leq 1$ are unknown and constitute $(r-1)$ dimensional vector $\boldsymbol{p} = (p_1, p_2, ..., p_{r-1})$ such that $\sum_{j=1}^{r} p_j = 1$. Since, we are dealing with the estimation of mixture weights, one main interest lies in the estimation of weights $\boldsymbol{p}$.

The mixture problem has several levels of difficulty: estimation of weights of (i) a fixed number of fully known distributions, (ii) a fixed number of partially specified distributions, and (iii) an unknown number of partially specfied distributions. For cases (ii) and (iii) Gibbs sampling and Markov Chain Monte Carlo(MCMC)

approaches can be implemented in finding expected log-likelihood (Wei and Tanner 1990). But in our case, we are dealing with either iid white noises or normally distributed errors, so we fall in category (i) and we implement the general EM algorithm to estimate the parameters. The likelihood function for independent random variables $X = (X_1, X_2, ..., X_n)$ can be written as

$$L(\boldsymbol{p}; x) = \prod_{i=1}^{n} f(x_i|\boldsymbol{p}, \boldsymbol{\theta}) = \prod_{i=1}^{n} \left(\sum_{j=1}^{r} p_j f_j(x_i|\boldsymbol{\theta}) \right).$$

So, the log likelihood becomes

$$logL(\boldsymbol{p}, \boldsymbol{\theta}; x) = log \left(\prod_{i=1}^{n} f(x_i|\boldsymbol{p}, \boldsymbol{\theta}) \right) = \sum_{i=1}^{n} log(\sum_{j=1}^{r} p_j f_j(x_i|\boldsymbol{\theta})). \tag{17}$$

Since, we are dealing with the estimation of mixture weights $\boldsymbol{p}$, for convenience let's take the log-likelihood with p_js only where all other parameters are known.

$$logL(\boldsymbol{p}; x) = log \left(\prod_{i=1}^{n} f(x_i|\boldsymbol{p}) \right) = \sum_{i=1}^{n} log(\sum_{j=1}^{r} p_j f_j(x_i)).$$

Even in this simplest case when the only parameters are weights $\boldsymbol{p}$, the log-likelihood assumes quite complicated form. The derivatives with respect to p_j lead to the system of equations, not solvable in a closed form.

$$\frac{\partial logL(\boldsymbol{p}; x)}{\partial p_j} = \sum_{i=1}^{n} \left(\frac{f_j(x_i)}{\sum_{j=1}^{n} p_j f_j(x_i)} \right).$$

Here,

$$\frac{\partial logL(\boldsymbol{p}; x)}{\partial p_j} = 0 \ .$$

is not solvable in closed form. This problem can be overcome by using EM algorithm with the introduction of new variable.

We implement the EM algorithm by introducing the unobserved indicators with the goal of simplifying the likelihood. Let's define an unobservable matrix,

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots w_{1j} & \dots & w_{1r} \\ \vdots & & & & \\ w_{i1} & w_{i2} & \dots w_{ij} & \dots & w_{ir} \\ \vdots & & & & \\ w_{n1} & w_{n2} & \dots w_{nj} & \dots & w_{nr} \end{bmatrix} = \begin{pmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_i \\ \vdots \\ \boldsymbol{w}_n \end{pmatrix}. \tag{18}$$

where $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \ldots, w_{w_i r}); \quad i = 1, 2, \ldots, n.$
The values w_{ij} are the indicators such that

$$w_{ij} = \begin{cases} 1, & \text{observation } x_i \text{ comes from the distribution } f_j \\ 0, & \text{else} \end{cases}$$

The unobservable matrix, W tells us where the i^{th} observation x_i comes from. Each row of W contains only one 1 and $(r-1)$ 0's. The augmented data $y_i = (x_i, \boldsymbol{w}_i), i = 1, 2, ..., n$, where $\boldsymbol{w}_i$ is an indicator vector of length r with a 1 in the position corresponding to the component of the mixture which generated x_i, makes the complete data which has quite simple likelihood form.
We have

$$P(w_{ij} = 1|x, \boldsymbol{\theta}) = p_j \ ,$$
$$\Longrightarrow f(w|\boldsymbol{p}, \boldsymbol{\theta}) = \prod_{j=1}^{r} p_j^{w_{ij}} \ .$$

and

$$f(x|\boldsymbol{w}, \boldsymbol{p}, \boldsymbol{\theta}) = \prod_{j=1}^{r} (f_j(x|\boldsymbol{\theta}))^{w_{ij}}.$$

So, the likelihood function for complete data is

$$L_c(\boldsymbol{p}, \boldsymbol{\theta}; x, w) = \prod_{i=1}^{n} \prod_{j=1}^{r} (p_j f_j(x_i|\boldsymbol{\theta}))^{w_{ij}}.$$

Hence, the complete log likelihood is

$$log L_c(\boldsymbol{p}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{r} w_{ij} log p_j + \sum_{i=1}^{n} \sum_{j=1}^{r} w_{ij} log f_j(x_i|\boldsymbol{\theta}). \tag{19}$$

Notice that the second term in the above equation is free of p_j so it's easy to find the conditional mean of w_{ij} given X. Assume that the m^{th} iteration of weights is $\boldsymbol{p}^{(m)}$ is already obtained. The conditional expectation of w_{ij} given x at m^{th} step (E-step) is

$$E_{\boldsymbol{p}^{(m)}}(w_{ij}|x, \boldsymbol{\theta}) = P_{\boldsymbol{p}^{(m)}}(w_{ij} = 1|x, \boldsymbol{\theta}) = w_{ij}^{(m)} \ ,$$

where $w_{ij}^{(m)}$ is the posterior probability of the i^{th} observation coming from the j^{th} mixture component, f_j, in the iterative step m. So, using Bayes rule

$$w_{ij}^{(m)} = \frac{p_j^{(m)} f_j(x_i|\boldsymbol{\theta})}{f(x_i, p^{(m)}|\boldsymbol{\theta})} .$$

Replacing w_{ij} by $w_{ij}^{(m)}$ in the complete log-likelihod equation 19, we get the expectation function $Q(\boldsymbol{p}|\boldsymbol{p}^{(m)})$.

Hence,

$$Q(\boldsymbol{p}|\boldsymbol{p}^{(m)}) = \sum_{i=1}^{n}\sum_{j=1}^{r} w_{ij}^{(m)} log p_j + \sum_{i=1}^{n}\sum_{j=1}^{r} w_{ij}^{(m)} log f_j(x_i|\boldsymbol{\theta}).$$

Since, we are mainly interested in estimation of weights, $\boldsymbol{p}$, the expectation function becomes

$$Q(\boldsymbol{p}|\boldsymbol{p}^{(m)}) = \sum_{i=1}^{n}\sum_{j=1}^{r} w_{ij}^{(m)} log p_j + \text{ Constant.} \tag{20}$$

The equation in 20 is maximized with respect to $\boldsymbol{p}$ to get $(m+1)^{th}$ step. Since, we have a condition $\sum_{j=1}^{r} p_j = 1$, we use the method of Lagrange multiplier to maximize the Lagrange function based on equation 20.

Let λ be a Lagrange multiplier. Then the Lagrange function is

$$\Lambda(\boldsymbol{p}, \lambda) = \sum_{i=1}^{n}\sum_{j=1}^{r} w_{ij}^{(m)} log p_j + Constant + \lambda\left(\sum_{j=1}^{r} p_j - 1\right), \tag{21}$$

$$\begin{aligned} &\frac{\partial \Lambda(\boldsymbol{p}, \lambda)}{\partial p_j} = 0 , \\ \Longrightarrow\ &\frac{1}{p_j}\sum_{i=1}^{n} w_{ij}^{(m)} + \lambda = 0 , \end{aligned} \tag{22}$$

Notice that the straight forward solution of λ is not possible by differentiating 21, i,e; $\dfrac{\partial \Lambda(\boldsymbol{p}, \lambda)}{\partial \lambda} = 0$, so summing over j for 22

$$\begin{aligned} &\sum_{j=1}^{r}\left[\frac{1}{p_j}\sum_{i=1}^{n} w_{ij}^{(m)} + \lambda\right] = 0 , \\ \Longrightarrow\ &\sum_{j=1}^{r}\left[\frac{1}{p_j}\sum_{i=1}^{n} w_{ij}^{(m)}\right] + r\lambda = 0 , \\ \Longrightarrow\ &\left[\frac{\sum_{i=1}^{n} w_{i1}^{(m)}}{p_1} + \frac{\sum_{i=1}^{n} w_{i2}^{(m)}}{p_2} + \ldots + \frac{\sum_{i=1}^{n} w_{ir}^{(m)}}{p_r}\right] + r\lambda = 0. \end{aligned} \tag{23}$$

Here, $w_{ij}^{(m)} = P_{\boldsymbol{p}^{(m)}}(w_{ij} = 1|x, \boldsymbol{\theta})$, is the conditional probability of i^{th} observation coming from j^{th} distribution. So,

$$\frac{w_{ij}^{(m)}}{p_j} = 1 \ ,$$
$$\implies \frac{\sum_{i=1}^{n} w_{ij}^{(m)}}{p_j} = n \ .$$

Hence, from equation 23

$$rn + r\lambda = 0 \ ,$$
$$\implies \lambda = -n \ .$$

Substituting the value of λ in equation 22, we get

$$p_j^{(m+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(m)}}{n} \ .$$

CHAPTER 3

MODEL BUILDING

For the time series data in which the data structure changes over different intervals of time,we suggest to use different time series models for different intervals rather than fitting a single time series model. The intervals are determined based on breakpoints as discussed earlier. Let's assume that we have m breakpoints which creates $m+1$ intervals. Thus, we have $m+1$ time series models, and each model is based only on the data of corresponding interval. So our main challenge is to combine all these models information to create a common model that can be used for forecasting. Several studies have been done in the past to combine the multiple time series regression models. Qin (1993), Qin and Lawless (1994), Qin and Zhang (1997), Gilbert (2000), Zhang (2000) and Fokianos et al. (2001) worked on some semi-parametric methods. Recently, Kedem and Gagnon (2010) further extended those ideas by showing the estimation of the probability distribution of a "reference" time series and using them in conditional prediction. All these aforementioned ideas use multiple time series regressions where different time series structures are related to different covariates but the ideas do not extend into the different time intervals. The method we propose deals with the multiple structured time series data. A joint density function which is a mixture of densities is estimated by using the combined residual data. The parameters of the joint density function is estimated by using EM algorithm. Further improvement in the parameter estimation is done by using bootstrap together with EM algorithm.

3.1 PARAMETER ESTIMATION

Let's assume that there are m breakpoints, so there are $m+1$ time series intervals. We assume that for each interval, different models fit the data so there are $m+1$ distinct models. Let y_{ij} , $i = 1, 2, ..., m, (m+1)$; $j = 1, 2, \ldots, n_i$ be the j^{th} observation in i^{th} interval.

Let $f_i(\boldsymbol{y}_i, \boldsymbol{\theta}_i)$ be the density function at i^{th} interval. Notice that this density function is the function of past values and time series parameters $\boldsymbol{\theta}_i$.

Let $\boldsymbol{t}_i, i = 1, 2, ..., m+1$ be the vector of discrete time components.
Then,

$$\begin{aligned} y_{1,t_1} &= f_1(\boldsymbol{z}_{1,t_1-1}) + \zeta_{t_1}, t_1 = 1, 2, ..., n_1 \ , \\ &\vdots \\ y_{(m+1),t_{m+1}} &= f_{m+1}(\boldsymbol{z}_{m+1,t_{m+1}-1}) + \zeta_{t_{m+1}} \ , \end{aligned} \tag{24}$$

where $t_{m+1} = t_m+1, t_m+2, \ldots, t_m+n_{m+1}$ and $\mathbf{z}_{i,t_i-1}$ contains past values of covariate time series possibly including even past values of y_{1,t_1}, ..., y_{m,t_m}, $y_{m+1,t_{m+1}}$. Also, n_i is the number of observations in the i^{th} interval. Throughout our discussion it will be assumed that data have been "mean corrected" by subtraction of the sample mean, so that it is appropriate to fit a zero-mean ARMA model to the adjusted data.
Since any ARMA model can be expressed in the linear form of $Y_t = \sum_{j=0}^{\infty} \psi_j Z_{t-j}$ where $\boldsymbol{Z} \sim WN(0, \sigma^2)$.
We have

$$f(\boldsymbol{z}_{i,t_i-1}) = \sum_{j=1}^{\infty} \psi_{ij} \zeta_{t_i-j} \ ; \ \boldsymbol{\zeta}_i \sim WN(0, \sigma_i^2) \ , i = 1, 2, \ldots, m+1.$$

If the ARMA process is driven by Gaussian white noise, we can take $\{\zeta_{t_i}\} \overset{iid}{\sim} N(0, \sigma_i^2)$.
So, the predicted values are

$$\begin{aligned} \hat{Y}_{i,t_i} &= \sum_{j=0}^{\infty} \hat{\psi}_{ij} \zeta_{t_i-j} \ , \\ \hat{\gamma}_i(h) &= \widehat{COV}(Y_{i,t_i}, Y_{i,t_i+h}) \ , \\ &= E(Y_{i,t_i} Y_{i,t_i+h}) \ , \qquad \text{[Since } E(Y_{i,t_i}) = 0\text{]} \\ &= E\left[\left(\sum_{j=0}^{\infty} \hat{\psi}_{ij} \zeta_{t_i+h-j} \right) \left(\sum_{k=0}^{\infty} \hat{\psi}_{ik} \zeta_{t_i-k} \right) \right] \ , \\ &= \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \hat{\psi}_{ij} \hat{\psi}_{ik} E(\zeta_{t_i+h-j} \zeta_{t_i-k}) \ , \\ &= \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \hat{\psi}_{ij} \hat{\psi}_{ik} \gamma_i(j-h-k). \end{aligned}$$

Taking $k = h - j$, we get

$$\hat{\gamma}_i(h) = \sum_{j=0}^{\infty} \hat{\psi}_{ij} \hat{\psi}_{i(h-j)} \gamma_i(0)$$

Since, the covariance structure is symmetric so the lag h coefficients $\hat{\psi}_{i(h-j)}$ and $\hat{\psi}_{i(h+j)}$ are equal. Also, $\gamma_i(0) = Var(\zeta_i) = \hat{\sigma}_i^2$.
Hence,

$$\hat{\gamma}_i(h) = \hat{\sigma}_i^2 \sum_{j=0}^{\infty} \hat{\psi}_{ij}\hat{\psi}_{i(j+h)}.$$

$$\Longrightarrow \widehat{VAR}(Y_{i,t_i}) = \hat{\gamma}_i(0) = \hat{\sigma}_i^2 \sum_{j=0}^{\infty} \hat{\psi}_{ij}^2.$$

The parameters $\boldsymbol{\psi}_i$ are composed of both autoregressive components, $\boldsymbol{\phi}_i$ and moving average components, $\boldsymbol{\theta}_i$. The preliminary estimates of parameters $\boldsymbol{\phi}_i$ and $\boldsymbol{\theta}_i$ are obtained by several methods such as Yule-Walker method, Burg procedure, innovations algorithm, Hannan-Rissanen algorithm and maximum likelihood method. Each method has its own advantages and limitations. Apart from the theoretical properties of the estimators such as consistency, efficiency etc., practical issues like the speed of computation and size of the data must also be taken into account in choosing an appropriate method for a given problem.

Yule-Walker and Burg procedures apply to the fitting of pure autoregressive models but innovations algorithm and Hannan-Rissanen algorithm are used for mixed models. Innovations algorithm is applicable to all series with finite second moments, regardless of whether they are stationary or not (Brockwell and Davis 2002). We also prefer innovation algorithm for the preliminary estimation of the parameters. Parameter estimation is improved by using innovations algorithm in conjunction with maximum likelihood method. The maximum likelihood method of estimating model parameters is often favored because it has the advantage among others that its estimators are more efficient (have smaller variance) and many large sample properties are known under rather general conditions. In our case, we do the parameter estimation as follows:

(I) We first identify the order (p, q) of ARMA model based on minimum value of corrected version of Akaike Information Criterion (AICc).

(II) Based on order (p, q) from previous step, we use one-step innovations algorithms to get preliminary estimates of $\boldsymbol{\phi}_i$ and $\boldsymbol{\theta}_i$.

(III) One step prediction errors obtained from innovations algorithm by using different values of $\boldsymbol{\phi}_i$ and $\boldsymbol{\theta}_i$ are then used to numerically maximize the likelihood function based on Gaussian noise.

3.1.1 MAXIMUM LIKELIHOOD ESTIMATION OF TIME SERIES PARAMETERS

We fit different time series models for different intervals. Parameters are estimated based on maximum likelihood method in which preliminary estimates are obtained through innovations algorithm. Even though the maximum likelihood method is based on the assumption of Gaussian noise, it still makes sense to use this method as a measure of goodness of fit of the model to the data and it has well defined asymptotic properties. A justification for using maximum Gaussian likelihood estimators of ARMA coefficients is that the large sample distribution of the estimators is the same for white noise $\{\zeta_t\} \sim IID(0, \sigma^2)$, regardless of whether or not Z_t is Gaussian (Brockwell and Davis 2002). For convenience, in our discussion we use a general case to derive the expressions for maximum likelihood rather than defining it for different intervals.

Let $\{Y_t\}$ be causal $ARMA(p, q)$ process so

$$Y_t = \Psi(B)\zeta_t = \sum_{j=0}^{\infty} \psi_j B^j \zeta_t \quad ; \zeta_t \sim WN(0, \sigma^2); \sum_{j=0}^{\infty} |\psi_j| < \infty.$$

Also, from 2,

$$Y_t = \frac{\theta(B)}{\phi(B)}\zeta_t$$

So,

$$\Psi(B) = \frac{\theta(B)}{\phi(B)}$$

$$\Longrightarrow (1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)(\psi_0 + \psi_1 B + \ldots) = 1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q.$$

Equating the like coefficients of B's, we get

$$\begin{aligned}
1 &= \psi_0 \ , \\
\theta_1 &= \psi_1 - \psi_0\phi_1 \Longrightarrow \psi_1 = \theta_1 + \psi_0\phi_1 \ , \\
\theta_2 &= \psi_2 - \psi_1\phi_1 - \psi_0\phi_2 \Longrightarrow \psi_2 = \theta_2 + \psi_1\phi_1 + \psi_0\phi_2 \ , \\
&\vdots \\
\psi_j &= \theta_j + \sum_{i=1}^{min.(j,p)} \phi_i \psi_{j-i} \ , \ j = 0, 1, \ldots, \qquad (25)
\end{aligned}$$

and we define $\theta_0 = 1$ and $\theta_j = 0$ for $j > q$. The innovations estimates $\hat{\theta}_{n1}, \hat{\theta}_{n2}, \ldots, \hat{\theta}_{n,(p+q)}$ obtained in section 2.1.2 are used to estimate $\psi_1, \psi_2, \ldots, \psi_p + q$. Replacing ψ_j by $\hat{\theta}_{nj}$ in equation 25 we get

$$\hat{\theta}_{nj} = \theta_j + \sum_{i=1}^{min.(j,p)} \phi_i \hat{\theta}_{n,(j-i)} \ , \ j = 1, 2 \ldots, p+q, \tag{26}$$

From last q equations we first estimate $\hat{\phi}$ as

$$\begin{pmatrix} \hat{\theta}_{n,q+1} \\ \hat{\theta}_{n,q+2} \\ \vdots \\ \hat{\theta}_{n,q+p} \end{pmatrix} = \begin{pmatrix} \hat{\theta}_{nq} & \hat{\theta}_{n,q-1} & \ldots & \hat{\theta}_{n,q+1-p} \\ \hat{\theta}_{n,q+1} & \hat{\theta}_{n,q} & \ldots & \hat{\theta}_{n,q+2-p} \\ \vdots & \vdots & . & \vdots \\ \hat{\theta}_{n,q+p-1} & \hat{\theta}_{n,q+p-2} & \ldots & \hat{\theta}_{n,q} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{pmatrix}$$

Then $\boldsymbol{\theta}$ can be estimated from equation 25 as

$$\hat{\theta}_j = \hat{\theta}_{nj} - \sum_{i=1}^{min.(j,p)} \hat{\phi}_i \hat{\theta}_{n,j-i} \ , \ j = 1, 2, \ldots, q.$$

After these preliminary estimation of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, we use these values as the initial values to get the maximum likelihood estimates. The maximization is nonlinear in the sense that the function to be maximized is not a quadractic function of the unknown parameters, so the estimators cannot be found by solving a system of linear equations. They are found instead by searching numerically for the maximum of the likelihood surface. When the order p and q of ARMA model is known, good estimators of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ can be found by imagining the data to be observations of a stationary Gaussian time series and maximizing the likelihood with respect to the $p + q + 1$ parameters $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$ and σ^2.

Suppose that $\{Y_t\}$ is a Gaussian time series with mean zero and autocovariance function $\kappa(i, j) = E(Y_i Y_j)$. Let $\boldsymbol{Y}_n = (Y_1, Y_2, \ldots, Y_n)'$ and let the one step predictors $\hat{Y}_n = (\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_n)'$ where $\hat{Y}_1 = 0$ and $\hat{Y}_j = P_{j-1} Y_j, j \geq 2$. Let Γ_n denote the covariance matrix $\Gamma_n = E(\boldsymbol{Y}_n \boldsymbol{Y}_n')$, and it is non-singular.

Then the likelihood of $\boldsymbol{Y}_n$ is

$$L(\Gamma_n; \boldsymbol{Y}_n) = (2\pi)^{-\frac{n}{2}} |\Gamma_n|^{-\frac{1}{2}} exp\left(-\frac{1}{2} \boldsymbol{Y}_n' \Gamma_n^{-1} \boldsymbol{Y}_n\right). \tag{27}$$

The direct calculation of Γ_n is cumbersome and in many situation not possible and it is avoided by using the one step prediction errors $Y_n - \hat{Y}_n$ and mean squared error,

$E(Y_n - \hat{Y}_n)^2$ instead of Y_n and Γ_n. Both of prediction error and mean squared errors are calculated recursively from the innovations algorithm already discussed.

From equation 11,

$$\begin{aligned}
\hat{\boldsymbol{Y}}_n &= \boldsymbol{\Theta}_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n) \\
&= C_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n) - I_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n) \qquad [\text{ From 10}] \\
&= C_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n) - \boldsymbol{Y}_n + \hat{\boldsymbol{Y}}_n \\
\Longrightarrow \boldsymbol{Y}_n &= C_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n) \qquad (28)
\end{aligned}$$

Since the components $Y_n - \hat{Y}_n$ are uncorrelated, the covariance matrix of $Y_n - \hat{Y}_n$ is

$$\Sigma_n = \begin{bmatrix} \upsilon_0 & 0 & \dots & 0 \\ 0 & \upsilon_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \upsilon_{n-1} \end{bmatrix}$$

From 28,

$$\begin{aligned}
Var(\boldsymbol{Y}_n) &= Var[C_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n)] \\
\Longrightarrow \Gamma_n &= C_n \Sigma_n C_n'.
\end{aligned}$$

So,

$$|\Gamma_n| = |C_n|^2 |\Sigma_n| = \upsilon_0 \upsilon_1 \dots \upsilon_{n-1},$$

and

$$\begin{aligned}
\boldsymbol{Y}_n' \Gamma_n^{-1} \boldsymbol{Y}_n &= \left[C_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n)\right]' \Gamma_n^{-1} C_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n) \qquad [\text{From 28}] \\
&= (\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n)' C_n' \Gamma_n^{-1} C_n(\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n) \\
&= (\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n)' \Sigma_n^{-1} (\boldsymbol{Y}_n - \hat{\boldsymbol{Y}}_n) \\
&= \frac{\sum_{j=1}^{n} (Y_j - \hat{Y}_j)^2}{\upsilon_{j-1}}
\end{aligned}$$

Hence, from equation 27 likelihood of vector $\boldsymbol{Y}_n$ reduces to

$$L(\Gamma_n; \boldsymbol{Y}_n) = \frac{1}{\sqrt{(2\pi)^n \upsilon_0 \upsilon_1 \dots \upsilon_{n-1}}} exp \left\{ \frac{-1}{2} \frac{\sum_{j=1}^{n} (Y_j - \hat{Y}_j)^2}{\upsilon_{j-1}} \right\}.$$

The likelihood for data from $ARMA(p,q)$ process is easily computed from the innovations form by replacing $\hat{Y}_j$ by one-step predictor and υ_j by $\sigma^2 r_n$ from 15.

Hence, the Gaussian likelihood for an ARMA process can be written as

$$L(\Gamma_n; \boldsymbol{Y}_n) = \frac{1}{\sqrt{(2\pi\sigma^2)^n r_0 r_1 \dots r_{n-1}}} exp\left\{ \frac{-1}{2\sigma^2} \frac{\sum_{j=1}^{n}(Y_j - \hat{Y}_j)^2}{r_{j-1}} \right\}. \tag{29}$$

So, maximum likelihood estimator of σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n} \frac{\sum_{j=1}^{n}(Y_j - \hat{Y}_j)^2}{r_{j-1}} .$$

and $\hat{\boldsymbol{\phi}}$ and $\hat{\boldsymbol{\theta}}$ are the values of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ that maximize the likelihood in 29.

Also, we used minimum AICc (Akaike Information Criteria Corrected) value as a major criterion for the selection of the orders p and q. AICc criterion can be defined as

$$AICc = -2lnL(\boldsymbol{\phi}_p, \boldsymbol{\theta}_q; Y_n) + \frac{2(p+q+1)n}{n-p-q-2} ,$$

where $lnL(\boldsymbol{\phi}_p, \boldsymbol{\theta}_q; Y_n)$ is the log of likelihood function defined in 29 using maximum likelihood estimators $\hat{\boldsymbol{\phi}}_p$ and $\hat{\boldsymbol{\theta}}_q$. For any fixed p and q it is clear that the AICc is minimized when $\boldsymbol{\phi}_p$ and $\boldsymbol{\theta}_q$ are the maximum likelihood estimators. Final decisions with respect to order selection should therefore be made on the basis of maximum likelihood estimators.

3.1.2 FORECASTING

As we have seen, autoregressive moving average time series models can be regarded as means of transforming the data to white noise, that is, to an uncorrelated sequence of errors. If the appropriate model has been chosen, there will be zero autocorrelation in the errors. For large samples the residuals from a correctly fitted model resemble very closely the true errors of the process (Box and Pierce 1970). Since there are differences in trends, forecasting of multiple time series data based on well behaved residuals and certain joint relationship between their probability density functions are explored by Kedem and Gagnon (2010). We are also exploiting the similar idea but by using the mixture distribution of residual densities as the reference distribution. The mixture parameters are estimated through the distributions of combined noise.

Since, each part of the interval is fitted with different models, the residuals for each part are independent to each other and to the errors from other intervals. So,the error sequence $\{\zeta_{t_i}\}$ is the sequence of *iid* random variables.

Define,

$$\zeta_{t_i} \overset{iid}{\sim} g_i(\varsigma), i = 1, ..., m, m+1 \ ,$$

where $g_i(\varsigma)$ is the density function of ζ_{t_i} for i^{th} interval. We approach time series prediction through the mixture distribution of these error components. Noises from different intervals are combined to form combined noise.

Let

$$\begin{aligned}\boldsymbol{\zeta} &= (\zeta_1, \zeta_2, ..., \zeta_{m+1}) \\ &= \{(\zeta_1, ..., \zeta_{n_1}), ..., (\zeta_{t_i+1}, \ldots, \zeta_{t_i+n_i}), ..., (\zeta_{t_m+1}, \ldots, \zeta_{t_m+n_{m+1}})\},\end{aligned}$$

The joint density of combined noise is the mixture of '$m+1$' noise distributions. So, the joint density of finite mixture of combined noise is

$$g(\varsigma) = \sum_{i=1}^{m+1} p_i g_i(\varsigma).$$

where p_i be the mixing proportion with the constraints $p_i \geq 0, \ i = 1, ..., m, m+1$, and $\sum_{i=1}^{m+1} p_i = 1$.

Hence, the cumulative distribution function of combined error is

$$P(\zeta \leq \varsigma) = G(\varsigma) = \sum_{i=1}^{m+1} p_i G_i(\varsigma), \tag{30}$$

where $G_i(\varsigma)$ is the cumulative distribution function of ζ_i , $i = 1, ..., m, m+1$.
Our main objective is to predict the future reference values $y_{m+2,t_{(m+1)}+1}$ conditional on past values $z_{m+1,t_{m+1}}$. Future probability of events conditional in past values can

be written as

$$
\begin{aligned}
P(y_{m+2,t_{(m+1)}+1} \leq y | \mathbf{z}_{(m+2),t_{m+1}}) &= P(f_{m+2}(\boldsymbol{z}_{(m+2),t_{m+1}}) + \zeta_{t_{m+1}+1} \leq y) \qquad \text{[From 24]} \\
&= P(\zeta_{t_{m+1}+1} \leq y - f_{m+2}(\boldsymbol{z}_{(m+2),t_{m+1}}) \\
&= G(y - f_{m+2}(\boldsymbol{z}_{(m+2),t_{m+1}})) \\
&= \sum_{i=1}^{m+1} p_i G_i(y - f_{m+2}(\boldsymbol{z}_{(m+2),t_{m+1}})) \qquad \text{[From 30]} \\
&= \sum_{i=1}^{m+1} \Bigg[p_i G_i(y - \sum_{j=1}^{p} \phi_{ij} y_{m+2,t_{(m+1)}-j} \\
&\qquad + \sum_{k=1}^{q} \theta_{ik} \zeta_{m+2,t_{(m+1)}-k}) \Bigg]. \qquad (31)
\end{aligned}
$$

Since we are using a sample of observed values, the cumulative distribution function can be approximated by empirical distribution function.

Let $\hat{G}_i(\varsigma)$ be the empirical distribution function of error components in i^{th} interval. Then

$$
\hat{G}_i(\varsigma) = \frac{\sum_{j=1}^{n_i} I(\zeta_{ij} \leq \varsigma)}{n_i},
$$

where

$$
I(\zeta_{ij} \leq \varsigma) = \begin{cases} 1, & \text{if } \zeta_{ij} \leq \varsigma; \qquad j = 1, 2, \ldots, n_i, \\ 0, & Otherwise. \end{cases}
$$

and ζ_{ij} be the j^{th} error component in i^{th} interval. Also, *strong law of large numbers* indicates that empirical cumulative distribution function converges almost surely to cumulative distribution function.

Here, for a fixed point ς the quantity $n_i \hat{G}_i(\varsigma) \sim Bin(n_i, G_i(\varsigma))$. Therefore

$$
E(\hat{G}_i(\varsigma)) = G_i(\varsigma) \text{ and } Var(\hat{G}_i(\varsigma)) = \frac{G_i(\varsigma)(1 - G_i(\varsigma))}{n_i}.
$$

By using Chebyshev's inequality

$$
\begin{aligned}
P(|\hat{G}_i(\varsigma) - G_i(\varsigma)| \geq \epsilon) &\leq \frac{G_i(\varsigma)(1 - G_i(\varsigma))}{n_i \epsilon^2} \qquad \text{for any } \epsilon > 0. \\
&\longrightarrow 0 \qquad \text{as } n_i \to \infty.
\end{aligned}
$$

Hence,

$$
\hat{G}_i(\varsigma) \xrightarrow{a.s} G_i(\varsigma).
$$

So the future probability of events conditional in past values from 31 can be approximated as

$$P(y_{m+2,t_{(m+1)}+1} \leq y | \mathbf{z}_{(m+2),t_{m+1}}) = \sum_{i=1}^{m+1} \left[\hat{p}_i \hat{G}_i \left(y - \sum_{j=1}^{p} \hat{\phi}_{ij} y_{m+2,t_{(m+1)}-j} + \sum_{k=1}^{q} \hat{\theta}_{ik} \zeta_{m+2,t_{(m+1)}-k} \right) \right]. \quad (32)$$

The parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ for each interval are estimated by using the method of maximum likelihood in conjunction with innovations algorithm as discussed in section 3.1.1. Estimation of future values in equation 32 also requires the estimation of p_i which is discussed in the next section.

3.1.3 ESTIMATION OF MIXTURE PROPORTIONS

In general time series, noises are either uncorrelated white noises or Gaussian noises. White noises are assumed to be a sequence of uncorrelated random variables generated from uniform probability distribution while Gaussian noises are generated from Gaussian distribution. The parameter estimation of mixture of Gaussian or other exponential family distribution can be done by using EM algorithm since the likelihood function of these kind of distribution is well defined (Dempster et al. 1977). A general method of parameter estimation for mixture of exponential family distribution is already discussed in section 2.5. But when dealing with the mixture of any location family distribution or particularly white noises EM algorithm may not be the appropriate method to estimate the parameters. The problem of identifiability should also be handled.

In our discussion we will focus more on Gaussian noise since it has some well defined properties. The time series parameter estimation using maximum likelihood method we proposed is based on the assumption of Gaussian noise. Another justification for using Gaussian noise is that, the large sample distribution of estimators is the same whether or not we use Gaussian (Brockwell and Davis 1991). Even though our primary focus is on Gaussian noise, we will also discuss the alternative way of parameter estimation for mixture distribution of white noises.

Gaussian Noise and EM algorithm

For each interval, without loss of generality, assume that $\zeta_i \sim N(0, \sigma_i^2)$. Gaussian mixture model is a simple linear superposition of Gaussian components. In section

2.5, we discussed the general case of parameter estimation of mixture model. In this section we will extend the same idea for the special case of Gaussian probability distributions. Recall from section 2.5 that the Gaussian mixture distribution can be written as linear combination of Gaussians in the form

$$g(\varsigma|\boldsymbol{\theta}) = \sum_{i=1}^{m+1} p_i g_i(\varsigma) \ ,$$

where $\boldsymbol{\theta} = (p_i, \mu_i, \sigma_i^2)$. So,

$$g(\varsigma|\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^{m+1} p_i \frac{1}{\sigma_i} \exp\left[\frac{-1}{2\sigma_i^2}(\varsigma - \mu_i)^2\right] \ , \tag{33}$$

This gives us the incomplete likelihood as

$$L(\boldsymbol{\theta}|\varsigma) = \left(\frac{1}{\sqrt{2\pi}}\right)^n \prod_{j=1}^{n} \left(\sum_{i=1}^{m+1} p_i \frac{1}{\sigma_i} \exp\left[\frac{-1}{2\sigma_i^2}(\varsigma - \mu_i)^2\right]\right) ; n = n_1, \ldots, n_{m+1},$$

and the log likelihood is

$$l(\boldsymbol{\theta}|\varsigma) \propto \sum_{j=1}^{n} log\left(\sum_{i=1}^{m+1} p_i \frac{1}{\sigma_i} \exp\left[\frac{-1}{2\sigma_i^2}(\varsigma_j - \mu_i)^2\right]\right) . \tag{34}$$

Maximizing the log likelihood of 34 turns out to be a more complex problem than for the case of a single Gaussian. The difficulty arises from the presence of summation over i that appears inside the logarithm, so that the logarithm function no longer acts directly on the Gaussian. If we set the derivatives of the log likelihood to zero, we will no longer obtain a closed form solution. Also, the maximum likelihood framework applied to the Gaussian mixture model has significant problem due to the presense of singularities. Whenever one of the Gaussian components collapses onto a specific data point, the log likelihood function will go to infinity as $\sigma_i \to 0$. This creates a singularity problem and inverse covariance matrix, which is often required in maximum likelihood framework, is unattainable. So we consider an alternative approach known as EM algorithm which is an elegant and powerful method for finding maximum likelihood solutions for models with latent variables (Dempster et al. 1977).

Let us introduce a $m+1$ dimensional binary random variable $\boldsymbol{w} = (w_1, w_2, \ldots, w_{m+1})'$ in which a particular element w_i is equal to 1 and all other elements are equal to 0. Matrix representation of W is given in 18. The value of the latent indicator w_i therefore satisfies $w_i \epsilon \{0, 1\}$ and $\sum_{i=1}^{m+1} w_i = 1$.

Also, the probability

$$p(w_i = 1) = p_i,$$

where the parameters $\{p_i\}$ must satisfy

$$0 \leq p_i \leq 1 \qquad \text{and} \qquad \sum_{i=1}^{m+1} p_i = 1.$$

Hence, the marginal density

$$p(\boldsymbol{w}) = \prod_{i=1}^{m+1} p_i^{w_i}.$$

Similarly, the conditional distribution of ς given a particular value of $\boldsymbol{w}$ is

$$p(\varsigma|w_i = 1, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[\frac{-1}{2\sigma_i^2}(\varsigma - \mu_i)^2\right],$$

So

$$p(\varsigma|\boldsymbol{w}, \boldsymbol{\theta}) = \prod_{i=1}^{m+1} \left(\frac{1}{\sqrt{2\pi}\sigma_i}\right)^{w_i} \left[\exp\left\{\frac{-1}{2\sigma_i^2}(\varsigma - \mu_i)^2\right\}\right]^{w_i}.$$

Using conditional probability, the joint density of ς and $\boldsymbol{w}$ is

$$\begin{aligned} p(\varsigma, w|\boldsymbol{\theta}) &= p(\varsigma|w, \boldsymbol{\theta})p(w|\boldsymbol{\theta}) \\ &= \prod_{i=1}^{m+1} \left(\frac{p_i}{\sqrt{2\pi}\sigma_i}\right)^{w_i} \left[\exp\left\{\frac{-1}{2\sigma_i^2}(\varsigma - \mu_i)^2\right\}\right]^{w_i}. \end{aligned}$$

Hence, the complete likelihood is

$$L(\boldsymbol{\theta}; \varsigma, \boldsymbol{w}) = \prod_{j=1}^{n} \prod_{i=1}^{m+1} \left(\frac{p_i}{\sqrt{2\pi}\sigma_i}\right)^{w_i} \left[\exp\left\{\frac{-1}{2\sigma_i^2}(\varsigma_j - \mu_i)^2\right\}\right]^{w_i} \text{ where } n = n_1 + \ldots + n_{m+1}.$$

And complete log likelihood becomes

$$\begin{aligned} l(\boldsymbol{\theta}; \varsigma, \boldsymbol{w}) = \sum_{j=1}^{n} \left(\sum_{i=1}^{m+1} w_i log p_i\right) - \frac{1}{2} \sum_{j=1}^{n} \left(\sum_{i=1}^{m+1} w_i log(2\pi\sigma_i^2)\right) \\ + \sum_{i=1}^{m+1} \sum_{j=1}^{n} w_i \left[\frac{-1}{2\sigma_i^2}(\varsigma_j - \mu_i)^2\right]. \end{aligned} \tag{35}$$

Notice that this log likelihood can be solved easily in the closed form once we identify the conditional distribution of $\boldsymbol{w}$ given $\boldsymbol{\varsigma}$.

The conditional probability of $\boldsymbol{w}$ given $\boldsymbol{\varsigma}$ when $w_i = 1$ plays an important role to define expectation function. We shall view p_i as the prior probability of $w_i = 1$, and the conditional distribution $p(w_i = 1|\boldsymbol{\varsigma}, \boldsymbol{\theta})$ as the corresponding posterior probability once we have observed $\boldsymbol{\varsigma}$ for some known parameter estimates $\boldsymbol{\theta}$. We can use Bayes rule to estimate the conditional probability of $\boldsymbol{w}$ given $\boldsymbol{\varsigma}$ and $\boldsymbol{\theta}$ as follows:

$$p(w_i = 1|\boldsymbol{\varsigma}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{\varsigma}|w_i = 1, \boldsymbol{\theta}).p(w_i = 1|\boldsymbol{\theta})}{p(\boldsymbol{\varsigma}|\boldsymbol{\theta})}$$

$$\left[\because P(A|B) = \frac{P(B|A)P(A)}{P(B)}\right]$$

$$= \frac{\frac{p_i}{\sqrt{2\pi}\sigma_i} \exp\left\{\frac{-1}{2\sigma_i^2}(\varsigma_i - \mu_i)^2\right\}}{\frac{1}{\sqrt{2\pi}} \sum_{j=1}^{m+1} \frac{p_j}{\sigma_j} \exp\left[\frac{-1}{2\sigma_j^2}(\varsigma_j - \mu_j)^2\right]}.$$

Hence,

$$p(w_i = 1|\boldsymbol{\varsigma}, \boldsymbol{\theta}) = \frac{\frac{p_i}{\sigma_i} \exp\left\{\frac{-1}{2\sigma_i^2}(\varsigma_i - \mu_i)^2\right\}}{\sum_{j=1}^{m+1} \frac{p_j}{\sigma_j} \exp\left[\frac{-1}{2\sigma_j^2}(\varsigma_j - \mu_j)^2\right]}. \tag{36}$$

Thus, the conditional expectation of $\boldsymbol{w}$ given $\boldsymbol{\varsigma}$ and $\boldsymbol{\theta}$ is

$$E(\boldsymbol{w}|\boldsymbol{\varsigma}, \boldsymbol{\theta}) = p(w_i = 1|\boldsymbol{\varsigma}, \boldsymbol{\theta})$$

Let's assume that we start with some initial values $\boldsymbol{\theta}^{(0)}$ and cycle up to k^{th} step. Let $\boldsymbol{\theta}^{(k)} = (p_i^{(k}, \mu_i^{(k)}, \sigma_i^{2(k)})$ be the parameter values at k^{th} step. Then, conditional expectation at k^{th} step can be written as

$$w_i^{(k)} = E(\boldsymbol{w}|\boldsymbol{\varsigma}, \boldsymbol{\theta}^{(k)})$$

$$= \frac{\frac{p_i^{(k)}}{\sigma_i^{(k)}} \exp\left\{\frac{-1}{2\sigma_i^{2(k)}}(\varsigma_i - \mu_i^{(k)})^2\right\}}{\sum_{j=1}^{m+1} \frac{p_j^{(k)}}{\sigma_j^{(k)}} \exp\left[\frac{-1}{2\sigma_j^{2(k)}}(\varsigma_j - \mu_j^{(k)})^2\right]}. \tag{37}$$

Now, in E step we replace w_i with the conditional expectation of $\boldsymbol{w}$ at k^{th} step from equation 37 into the complete log likelihood obtained in equation 35. Hence, the expectation function, $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$, becomes

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}) = \sum_{j=1}^{n}\left(\sum_{i=1}^{m+1} w_i^{(k)} log p_i\right) - \frac{1}{2}\sum_{j=1}^{n}\left(\sum_{i=1}^{m+1} w_i^{(k)} log(2\pi\sigma_i^2)\right) + \sum_{i=1}^{m+1}\sum_{j=1}^{n} w_i^{(k)}\left[\frac{-1}{2\sigma_i^2}(\varsigma_j - \mu_i)^2\right]. \tag{38}$$

In the M step, we determine the revised parameter estimate $\boldsymbol{\theta}^{(k+1)}$ by maximizing equation 38 with respect to relative parameters, p_i, μ_i and σ_i^2. The equation 38 can be maximized with respect to p_i under the condition that $\sum_{i=1}^{m+1} p_i = 1$. So we need to maximize the Lagrange function as discussed in section 2.5. From 21, Lagrange function is

$$\Lambda(\boldsymbol{p}, \lambda) = \sum_{j=1}^{n}\sum_{i=1}^{m+1} w_i^{(k)} log p_i + Constant + \lambda\left(\sum_{i=1}^{m+1} p_i - 1\right).$$

Maximizing with respect to p_i and λ and substituting the value of λ we get (from section 2.5) the estimate of p_i at $(k+1)^{th}$ step as

$$p_i^{(k+1)} = \frac{\sum_{j=1}^{n} w_i^{(k)}}{n},$$

where $w_i^{(k)}$ is the conditional expectation as discussed in 37. Also,

$$\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})}{\partial \mu_i} = 0$$

$$\implies \sum_{j=1}^{n} w_i^{(k)}(\varsigma_j - \mu_i) = 0$$

$$\implies \mu_i = \frac{\sum_{j=1}^{n} w_i^{(k)}\varsigma_j}{\sum_{j=1}^{n} w_i^{(k)}}$$

$$\implies \mu_i^{(k+1)} = \frac{\sum_{j=1}^{n} w_i^{(k)}\varsigma_j}{np_i^{(k+1)}}.$$

Similarly,

$$\frac{\partial Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})}{\partial \sigma_i^2} = 0$$

$$\Longrightarrow \qquad \sigma_i^{2(k+1)} = \frac{\sum_{j=1}^{n} \left(\varsigma_j - \mu_i^{(k)}\right)^2 w_i^{(k)}}{np_i^{(k+1)}} .$$

It can be shown that the sequence $\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(k)}\}$ converges to the maximum likelihood estimator of $\boldsymbol{\theta}$, i,e. $\hat{\boldsymbol{\theta}}$ as $k \to \infty$ (Dempster et al. 1977).

In our applications, we are using the mixture of two Gaussian distributions in chapter 4. The mixture of two Gaussian population is given as:

$$g(\varsigma|\boldsymbol{\theta}) = p\frac{1}{\sigma_1}\varphi\left(\frac{\varsigma - \mu_1}{\sigma_1}\right) + (1-p)\frac{1}{\sigma_2}\varphi\left(\frac{\varsigma - \mu_2}{\sigma_2}\right),$$

where φ is the cumulative distribution function of the standard normal distribution and $\boldsymbol{\theta} = (p, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$; $0 < p < 1$.
Then, the indicator variable W be treated as missing data information such that:

$$W = \begin{cases} 1, & \text{if} \quad \zeta_j \text{ belongs to first interval} \\ 0, & \text{if} \quad \zeta_j \text{ belongs to second interval,} \end{cases}$$

where W_i is Bernoulli distributed with parameter p.
Therefore, the likelihood expression for complete data becomes:

$$L_n(\theta|\varsigma, w) = \prod_{j=1}^{n} p^w(1-p)^{1-w}\frac{1}{\sigma_1^w}\varphi\left(\frac{\varsigma_j - \mu_1}{\sigma_1}\right)^w \frac{1}{\sigma_2^{1-w}}\varphi\left(\frac{\varsigma_j - \mu_2}{\sigma_2}\right)^{1-w}.$$

And the corresponding log-likelihood function for the density becomes:

$$\begin{aligned} l_n(\theta|\varsigma, w) \;=\; & \sum_{j=1}^{n} w log(p) + \sum_{j=1}^{n}(1-w)log(1-p) - \frac{1}{2}\sum_{j=1}^{n} w log(2\pi\sigma_1^2) \\ & -\frac{1}{2\sigma_1^2}\sum_{j=1}^{n} w(\varsigma_j - \mu_1)^2 - \frac{1}{2}\sum_{j=1}^{n}(1-w)log(2\pi\sigma_2^2) \\ & -\frac{1}{2\sigma_2^2}\sum_{j=1}^{n}(1-w)(\varsigma_j - \mu_2)^2. \end{aligned}$$

The conditional distribution of W given ζ is:

$$W|\varsigma_j, \theta^{(k)} \sim Bin(1, w^{(k)}),$$

with

$$w^{(k)} = \frac{p^{(k)} \frac{1}{\sigma_1^{(k)}} \varphi(\frac{\varsigma_j - \mu_1^{(k)}}{\sigma_1^{(k)}})}{p^{(k)} \frac{1}{\sigma_1^{(k)}} \varphi(\frac{\varsigma_j - \mu_1^{(k)}}{\sigma_1^{(k)}}) + (1 - p^{(k)}) \frac{1}{\sigma_2^{(k)}} \varphi(\frac{\varsigma_j - \mu_2^{(k)}}{\sigma_2^{(k)}})},$$

where $p^{(k)}$ is a set of known or estimated parameters at k^{th} step. The initial value $p^{(0)}$ can be obtained from the empirical distribution.
Hence, the conditional mean at k^{th} step is:

$$E(w|\varsigma_j, \theta^{(k)}) = w^{(k)}.$$

The expectation function becomes

$$\begin{aligned} Q(\theta|\theta^{(k)}) &= \sum_{j=1}^{n} w^{(k)} log(p) + \sum_{j=1}^{n} (1 - w^{(k)}) log(1-p) - \frac{1}{2} \sum_{j=1}^{n} w^{(k)} log(2\pi\sigma_1^2) \\ &\quad - \frac{1}{2\sigma_1^2} \sum_{j=1}^{n} w^{(k)} (\varsigma_j - \mu_1)^2 - \frac{1}{2} \sum_{j=1}^{n} (1 - w^{(k)}) log(2\pi\sigma_2^2) \\ &\quad - \frac{1}{2\sigma_2^2} \sum_{j=1}^{n} (1 - w^{(k)}) (\varsigma_j - \mu_2)^2. \end{aligned}$$

Now, we maximize the expectation function as discusssed above.

Hence, the parameter estimates at the $(k+1)^{th}$ step are:

$$p^{(k+1)} = \frac{1}{n} \sum_{j=1}^{n} w^{(k)},$$

$$\mu_1^{(k+1)} = \frac{\sum_{j=1}^{n} w^{(k)} \varsigma_j}{\sum_{j=1}^{n} w^{(k)}}, \quad \mu_2^{(k+1)} = \frac{\sum_{j=1}^{n} (1 - w^{(k)}) \varsigma_j}{\sum_{j=1}^{n} (1 - w^{(k)})},$$

$$\sigma_1^{(k+1)^2} = \frac{\sum_{j=1}^{n} w^{(k)} (\varsigma - \mu_1^{(k+1)})^2}{\sum_{j=1}^{n} w^{(k)}}, \quad \text{and} \quad \sigma_2^{(k+1)^2} = \frac{\sum_{j=1}^{n} (1 - w^{(k)}) (\varsigma_j - \mu_2^{(k+1)})^2}{\sum_{j=1}^{n} (1 - w^{(k)})}.$$

The initial values of $\boldsymbol{\theta} = (p^{(0)}, \mu_1^{(0)}, \mu_2^{(0)}, \sigma_1^{2\ (0)}, \sigma_2^{2\ (0)})$ are obtained from the empirical distribution.

Each update to the parameters resulting from an E step followed by M step is guaranteed to increase the log likelihood function. In practice, the algorithm is deemed to have converged when the change in the log likelihood function, or alternatively in the parameters, falls below some threshold.

Mixture of White Noises

In the previous section, we have discussed more general case of mixture of Gaussian noises. As mentioned earlier, Gaussian noise have some well defined properties and they are easy to deal with. But it's not uncommon to assume time series as a linear combination of white noises. White noises are the random variables from some uniform distribution defined in a particular interval. Teicher (1963) showed that univariate normal mixtures are identifiable, while in general mixture of uniform distributions are not. Identiability is a necessary condition for the possibility to estimate the parameters of a mixture model consistently. It makes sure that no two essentially different mixture parameter vectors parameterize the same distribution. According to Casella and Berger (2002)

"A parameter θ for a family of distributions $\{f(x|\theta) : \theta \epsilon \Theta\}$ is *identifiable* if distinct values of θ correspond to distinct probability density or mass functions. That is, if $\theta \neq \theta'$, then $f(x|\theta)$ is not the same function of x as $f(x|\theta')$."

Maximum likelihood method of parameter estimation for exponential family distribution give robust estimates. EM algorithm can also be thought as an adjusted maximum likelihood method. So the parameter estimates of mixture of Gaussian are robust and identifiable. Maximum likelihood estimates for a wide class of location-scale mixtures are not robust (Hennig 2004). So the parameter estimation based on EM algorithm may not be the appropriate choice when we deal with mixture of white noises.

Parameter estimation of mixture of uniform distributions using the method of moments and method of maximum likelihood is discussed in Craigmile and Titterington (1997). In this section we briefly discuss the alternative way of estimating the parameters for mixture of white noises. Since our main interest lies in the Gaussian noise and parameter estimation through EM algorithm, in this section we just outline some of the parameter estimates based on method of moments for mixture of uniform distribution without giving detailed explanation.

As discussed earlier, the apparent simplicity of uniform mixtures conceals a hidden

danger of non-identifiability.

For example, let us assume a two component mixture of uniform distribution

$$f(\varsigma|p,\theta) = pU(\varsigma;0,\theta) + (1-p)U(\varsigma;\theta,1),$$

where $U(\varsigma;a,b)$ denotes the uniform density on the interval $[a,b)$.

Let us take $p=\theta$. Then

$$f(\varsigma|p,p) = U(0,1).$$

So, for any values of p, we get the same distribution function. The problem of non-identifiability arises and the estimate is not consistent. But if p or θ is known, the mixture is identifiable even if the true values of p and θ are equal. So care should be taken when dealing with mixture of uniform distribution and we should avoid the condition of non-identifiability.

For convenience, let us assume that we have the mixture of two white noises from each of the interval, so $\zeta_i \overset{iid}{\sim} WN(0,\sigma_i^2); i=1,2$. Notice that σ_i^2 is the variance of the uniform distribution in the i^{th} interval. Since, uniform distribution is a location-scale family, we can consider two disjoint intervals for uniform mixture and can write the density function as

$$f(\varsigma|p,\theta) = pU(\varsigma;0,\theta) + (1-p)U(\varsigma;\theta,1), \tag{39}$$

As discussed earlier, this is non-identifiable when $p=\theta$, so we take the cases when $p \neq q$. Equation 39 can be written as

$$f(\varsigma|p,\theta) = \frac{p}{\theta} I(0 \leq \varsigma < \theta) + \frac{(1-p)}{(1-\theta)} I(\theta \leq \varsigma < 1),$$

where I be the indicator function. Since, k^{th} raw moment of uniform distribution $U(\varsigma;a,b)$ is

$$\frac{1}{k+1}\frac{b^{k+1}-a^{k+1}}{b-1}$$

we can write the k^{th} raw moment of the mixture density as

$$\begin{aligned} m_k = E(\zeta^k) &= \frac{p\theta^k}{k+1} + \frac{(1-p)(1-\theta^{k+1})}{(1-\theta)(k+1)} \\ &= \frac{1-\theta^{k+1}-p(1-\theta^k)}{(1-\theta)(k+1)}. \end{aligned} \tag{40}$$

Also, k^{th} sample moment can be represented as

$$M_k = \frac{1}{n}\sum_{j=1}^{n} x_j^k \tag{41}$$

By equating k^{th} raw moment with k^{th} sample moment we estimate the parameters of mixture distribution. As discussed earlier, we choose the cases when $p \neq \theta$ to avoid non-identifiability. There are three cases of parameter estimation:

(i) θ known, p unknown

(ii) p known, θ unknown and

(iii) both p and θ are unknown

But in our situation, we don't have known p, so second case is irrelevant to our discussion.Here we'll discuss case (i) and case (iii) briefly.

Case I: θ known, p unknown

When θ is known, p is estimated by equating k^{th} raw moment with k^{th} sample moment as

$$\begin{aligned} & M_k = m_k, \\ & M_k = \frac{p\theta^k}{k+1} + \frac{(1-p)(1-\theta^{k+1})}{(1-\theta)(k+1)}, \qquad \text{[From 40]} \\ \Longrightarrow \quad & \tilde{\mathrm{p}} = \frac{-(1-\theta)(k+1)}{(1-\theta^k)} M_k + \frac{(1-\theta^{k+1})}{(1-\theta^k)} , \end{aligned}$$

where M_k is the k^{th} sample moment defined in equation 41. For simplicity we can write

$$\tilde{\mathrm{p}} = c_k M_k + d_k.$$

The order k is determined based on the optimal variance of $\tilde{\mathrm{p}}$.

$$\begin{aligned} var(\tilde{\mathrm{p}}) &= c_k^2 var\left(\frac{1}{n}\sum_{j=1}^{n} x_j^k\right) \\ &= \frac{c_k^2}{n^2}\left[E\left(\left\{\sum_{i=1}^{n} x_i^k\right\}^2\right) - E\left(\sum_{i=1}^{n} x_i^k\right)^2\right] \\ &= \frac{c_k^2(m_{2k} - m_k^2)}{n} . \end{aligned}$$

We choose k and estimate of p in such a way that variance of $\tilde{\mathrm{p}}$ will be minimum. Gupta and Miyawaki (1978) suggested $k = 1$ for estimation of p.

Case III: Both p and θ unknown

Gupta and Miyawaki (1978) has suggested using first and second order moments to estimate the parameters of θ and p. Here, we will derive the expression based on first and second order moments. As suggested by method of moments,

$$\begin{aligned} M_1 &= m_1 \\ M_1 &= \frac{p\theta}{2} + \frac{(1-p)(1-\theta^2)}{2(1-\theta)} \\ 2M_1 &= 1 + \theta - p. \end{aligned} \tag{42}$$

Again,

$$\begin{aligned} M_2 &= m_2 \\ M_2 &= \frac{p\theta^2}{3} + \frac{(1-p)(1-\theta^3)}{3(1-\theta)} \\ 3M_2 &= 1 + \theta - p + \theta^2 - p\theta. \end{aligned} \tag{43}$$

From equation 42 and 43,

$$\tilde{\theta} = \frac{3M_2 - 2M_1}{2M_1 - 1},$$

and

$$\tilde{\mathrm{p}} = 1 - \frac{4M_1^2 - 3M_2}{2M_1 - 1}.$$

Caveat: Mixture of white noises

The extension of more than 2 component mixture of uniform distribution and their parameter estimation is discussed briefly by Craigmile and Titterington (1997) giving an example for mixture of 3 component uniform distributions. Even for three component mixture distribution, there are several cases of non-identifiability and if we have higher component mixture distributions we will encounter multiple cases of non-identifiability. So the parameter estimation is restricted by several conditions. It is not possible to track all the restricted conditions so higher component mixture of uniform distribution is not suggested. The parameter estimation could be very inconsistent and in many cases not possible. Also, one should be very careful when assuming the mixture of uniform distribution, since mixture is defined as the combination of non-overlapping uniform distribution. In the cases with large number of breakpoints, the number intervals $m + 1$ may not be equal to the number of clusters for the mixture of uniform distributions. If this situation arises, the method of

forecasting based on $m+1$ mixtures that we have proposed will not be appropriate and some other methods with reduced dimension should be considered. But this is not the case for mixture of Gaussian noises. So, we consider the case of mixture of white noises as just an alternative approach and preference is given to the framework based on mixture of Gaussian noises.

3.2 CONFIDENCE INTERVAL ESTIMATON AND LARGE SAMPLE PROPERTIES

In sections 2.1.2 and 3.1.1, we discussed the time series parameter estimation using innovations algorithm together with the maximum likelihood method. We can use the asymptotic distribution of ARMA parameters $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}})$ to derive approximate large sample confidence regions for the true coefficient vectors $(\boldsymbol{\phi}, \boldsymbol{\theta})$.

Let Y_t be the stationary and invertible time series process. An ARMA model can be written as

$$Y_t - \sum_{i=1}^{p} \phi_i Y_{t-i} = \zeta_t - \sum_{j=1}^{q} \theta_j \zeta_{t-j}; \qquad \zeta \overset{iid}{\sim} N(0, \sigma^2)$$

This is equivalent to

$$\prod_{i=1}^{p} (1 - A_i) Y_t = \prod_{j=1}^{q} (1 - M_j) \zeta_t, \tag{44}$$

$$\Longrightarrow \qquad \zeta_t = \prod_{i=1}^{p} (1 - A_i) \prod_{j=1}^{q} (1 - M_j)^{-1} Y_t \ .$$

For example, if we have $ARMA(2,2)$ model

$$Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} = \zeta_t - \theta_1 \zeta_{t-1} - \theta_1 \zeta_{t-1},$$

Then we can easily derive

$$\phi_1 = A_1 + A_2 \qquad \phi_2 = -A_1 A_2$$
$$\theta_1 = M_1 + M_2 \qquad \theta_2 = -M_1 M_2.$$

ARMA model are the superposition of both AR and MA models, so we can write AR and MA components of equation 44 in terms of past errors as

$$u_{t,i} = -\frac{\partial \zeta_t}{\partial A_i} = (1 - A_i B)^{-1} \zeta_{t-1}$$
$$v_{t,i} = -\frac{\partial \zeta_t}{\partial M_i} = -(1 - M_j B)^{-1} \zeta_{t-1} \tag{45}$$

For the mixed $ARMA$ models, the information matrix can be written as:

$$I(\phi,\theta) =$$

$$\frac{n}{\sigma^2}\left[\begin{array}{cccc:cccc} \gamma_{uu}(0) & \gamma_{uu}(1) & \dots & \gamma_{uu}(p-1) & \gamma_{uv}(0) & \gamma_{uv}(-1) & \dots & \gamma_{uv}(1-q) \\ \gamma_{uu}(1) & \gamma_{uu}(0) & \dots & \gamma_{uu}(p-2) & \gamma_{uv}(1) & \gamma_{uv}(0) & \dots & \gamma_{uv}(2-q) \\ & \vdots & & \vdots & & \vdots & & \vdots \\ \gamma_{uu}(p-1) & \gamma_{uu}(p-2) & \dots & \gamma_{uu}(0) & \gamma_{uv}(p-1) & \gamma_{uv}(p-2) & \dots & \gamma_{uv}(p-q) \\ \hdashline \gamma_{uv}(0) & \gamma_{uv}(-1) & \dots & \gamma_{uv}(1-q) & \gamma_{uu}(0) & \gamma_{uu}(1) & \dots & \gamma_{uu}(p-1) \\ \gamma_{uv}(1) & \gamma_{uv}(0) & \dots & \gamma_{uv}(2-q) & \gamma_{uu}(1) & \gamma_{uu}(0) & \dots & \gamma_{uu}(p-2) \\ & \vdots & & \vdots & & \vdots & & \vdots \\ \gamma_{uu}(1-q) & \gamma_{uu}(2-q) & \dots & \gamma_{uu}(p-q) & \gamma_{uv}(q-1) & \gamma_{uv}(q-2) & \dots & \gamma_{uv}(0) \end{array}\right] \tag{46}$$

where $\gamma_{uu}(h) = E(u_t, u_{t+h})$,$\gamma_{uv}(h) = E(u_t, v_{t+h})$ and so on. u and v are related to autoregressive and moving average components of equation 45. Using equation 46 and 45, for the large sample we get the information matrix in terms of A_i and M_j as $I(\phi,\theta) =$

$$n\left[\begin{array}{cccc:ccc} (1-A_1^2)^{-1} & (1-A_1A_2)^{-1} & \dots & (1-A_1A_p)^{-1} & -(1-A_1M_1)^{-1} & \dots & -(1-A_1M_q)^{-1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ (1-A_1A_p)^{-1} & (1-A_2A_p)^{-1} & \dots & (1-A_p^2)^{-1} & -(1-A_pM_1)^{-1} & \dots & -(1-A_pM_q)^{-1} \\ \hdashline -(1-A_1M_1)^{-1} & -(1-A_2M_1)^{-1} & \dots & -(1-A_pM_1)^{-1} & (1-M_1^2) & \dots & -(1-M_1M_q)^{-1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ -(1-A_1M_q)^{-1} & -(1-A_2M_q)^{-1} & \dots & -(1-A_pM_q)^{-1} & (1-M_1M_q)^{-1} & \dots & (1-M_q^2)^{-1} \end{array}\right].$$

This marix can be partitioned after p^{th} row and q^{th} column. So,

$$I(\phi,\theta) = n\left[\begin{array}{c:c} I_{AA} & I_{AM} \\ \hdashline I'_{AM} & I_{MM} \end{array}\right].$$

Notice that matrix $I^{-1}(\phi,\theta)$ is non-singular, so the covariance matrix of estimators of ϕ and θ for ARMA model is

$$\Sigma(\hat{\phi},\hat{\theta}) = I^{-1}(\phi,\theta).$$

In case of pure AR and MA models, this covariance matrix can further split by removing cross covariance matrices of AR and MA comonents (i.e. $I_{AM} = \mathbf{0}$). This

gives

$$\begin{aligned}\Sigma(\hat{\phi}) &= I^{-1}(\phi) \\ &= \frac{1}{n} \quad I_{AA}^{-1} \ ,\end{aligned}$$

$$\begin{aligned}\Sigma(\hat{\theta}) &= I^{-1}(\theta) \\ &= \frac{1}{n} \quad I_{MM}^{-1} \ .\end{aligned}$$

Let $\beta = (\phi, \theta)$ be the vector of $ARMA$ model parameters.
The large sample distribution of maximum likelihood estimators of $ARMA(p, q)$ can be written as

$$\hat{\beta} \sim N_{p+q}(\beta, n^{-1}\Sigma(\hat{\phi}, \hat{\theta}) \ ,$$

And for pure autoregressive(AR) and moving average (MA) models,

$$\hat{\beta}_p \sim N_p(\beta_p, n^{-1}\Sigma(\hat{\phi}))$$

and

$$\hat{\beta}_q \sim N_q(\beta_q, n^{-1}\Sigma(\hat{\theta})) \ .$$

As we have seen that any ARMA(p, q) model can be obtained using the linear filter ψ from white noise or Gaussian noise, $\zeta \overset{iid}{\sim} N(0, \sigma^2)$.

$$Y_t = \psi(B)\zeta_t = \sum_{j=0}^{\infty} \psi_j \zeta_{t-j}.$$

Let $\hat{Y}_{t+h}$ be the h step predictor as discussed in sections 3.1.2 and 2.1.2. Then

$$\hat{Y}_{t+h} = \sum_{j=0}^{h-1} \hat{\psi}_j \zeta_{t+h-j}.$$

Hence, the mean squared error is

$$\begin{aligned}S^2 = E(Y_t - \hat{Y}_{t+h})^2 &= \sum_{j=0}^{h-1} \hat{\psi}_j^2 \ \mathrm{Var}(\zeta_{t+h-j}). \\ &= \hat{\sigma}^2 \sum_{j=0}^{h-1} \hat{\psi}_j^2 \ . \qquad (47)\end{aligned}$$

Here, $\hat{\psi}_j^2$ is the function of estimators $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ such that

$$\hat{\psi}_j = \sum_{k=1}^{p} \hat{\phi}_k \hat{\psi}_{j-i} - \hat{\theta}_j, j = 0, 1, 2, \ldots,$$

where $\hat{\theta}_0 = 1$, $\hat{\theta}_j = 0$ if $j > q$, $\hat{\psi}_0 = 1$, and $\hat{\psi}_j = 0$ if $j < 0$.
Other estimators $\hat{\boldsymbol{\phi}}_k, k = 1, 2, \ldots, p$ and $\hat{\boldsymbol{\theta}}_j, j = 1, 2, \ldots, q$ are estimated using innovations algorithm together with maximum likelihood method as discussed in section 2.5. And the estimator $\hat{\sigma}^2$ is obtained from the empirical data.

If the ARMA(p, q) process $\{Y_t\}$ (for each interval separated by breakpoints) is driven by Gaussian white noise, then the prediction error $Y_{t+k} - \hat{Y}_{t+k}$ is normally distributed with mean 0 and variance S^2 given by the equation 47. In our case we are using one-step prediction so $k = 1$. Hence, the prediction interval of Y_{t+k} is

$$Y_{t+k} = \hat{Y}_{t+k} \pm \Phi_{1-\alpha/2} S$$

Let us assume that there are $m+1$ mixture components, then for the forecasting based on mixture distribution we can rewrite the h step prediction as

$$\hat{Y}_{n+h} = \sum_{i=1}^{m+1} p_i \sum_{j=0}^{h-1} \hat{\boldsymbol{\psi}}_{ij} \zeta_{n+h-j}.$$

where $\sum_{i=1}^{m+1} p_i = 1$ and $0 \leq p_i \leq 1$, $n = n_1 + n_2 + \ldots + n_{m+1}$, n_i be the number of observations in each component of mixtures and $\hat{\boldsymbol{\psi}}_i$ is composed of ARMA parameters $(\hat{\boldsymbol{\phi}}_i, \hat{\boldsymbol{\theta}}_i)$ from each breakpoint groups. Hence, the mean squared error is

$$S_m^2 = \hat{\sigma}_{m+1}^2 \sum_{i=1}^{m+1} \sum_{j=1}^{h-1} p_i^2 \hat{\boldsymbol{\psi}}_{ij}^2 \ ,$$

where $\hat{\sigma}_{m+1}^2$ be the white noise variance estimator of $(m+1)^{th}$ component.

Assuming that the ARMA process $\{Y_n\}$ is driven by Gaussian white noise so if $\zeta_t \overset{iid}{\sim} N(0, \sigma^2)$, then for each $h \geq 1$ the prediction error is normally distributed with mean 0 and variance S_m^2. It follows that Y_{n+h} lies between the bounds

$$\hat{Y}_{n+h} \pm \Phi_{1-\alpha/2} S_m. \tag{48}$$

with probability $(1 - \alpha)$. In the above equation $\Phi_{1-\alpha/2}$ is the $(1 - \alpha/2)$ quintile of standard normal distribution. We can call this bound as prediction bound for Y_{n+h}.

3.3 DIAGNOSTIC CHECKING

Once the model is identified and the parameters are estimated, diagnostic checks are then applied to the fitted model. It is an important step to conduct various diagnostic tests in time series modeling. Most of the diagnostic tests are designed to examine the auto correlation structure of the time series data itself and the residuals. So we perform diagnostic tests on both the data before identifying the best model and the residuals after fitting the best model to the data. If a time series is serially uncorrelated, no linear function of the lagged variables can account for the behavior of the current variable. For a serially independent time series, there is no relationship between the current and past variables. Diagnostic testing on data series thus provides information regarding how these data might be modeled. When a model is estimated, diagnostic tests can be applied to evaluate model residuals, which also serve as tests of model adequacy. We implement diagnostic tests to the data and residuals in each interval separated by breakpoints and to the overall data. The diagnosis of the model requires confirming that basic hypotheses made with respect to the residuals are true. We do diagnostic checking on the residuals to see whether the following assumptions are met or not.

(1) No correlation for any lag (Autocorrelation Test)
(2) Constant marginal variance and zero marginal mean.
(3) Gaussian distribution

Let $\{Y_t\}$ be any ARMA process. Then it can be written as

$$(1 - \phi(B))Y_t = (1 - \theta(B))\zeta_t \ .$$

The residuals are computed recursively as

$$\hat{\zeta}_t = Y_t - \sum_{i=1}^{p} \hat{\phi}_i Y_{t-i} + \sum_{j=1}^{q} \hat{\theta}_j \hat{a}_{t-j}, t = 1, 2, \ldots, n.$$

If the model is adequate, then

$$\hat{\zeta}_t = \zeta_t + O\left(\frac{1}{n}\right). \qquad \text{(Box et al. 1994)}$$

As the series length increases, the $\hat{\zeta}_t$'s become close to the white noise ζ_t's. Therefore, one might expect that study of the $\hat{\zeta}_t$'s could indicate the existence and nature

of model inadequacy. In particular, recognizable patterns in the estimated autocorrelation function (ACF) of the $\hat{\zeta}_t$ could point to appropriate modification in the model.

3.3.1 AUTOCORRELATION TESTS

For an adequate model, the residuals should behave as white noise and should be *iid.* Portmanteau lack of fit test which was first proposed by Box and Pierce (1970) is the most common method to test the autocorrelation. However, Ljung and Box (1978) argued that the Q statistics provided by Box and Pierce tending to be somewhat smaller than expected under the chi-squared distribution. Other empirical evidences also reinforced their argument. Ljung and Box proposed a modified form of the Box and Pierce statistics which is more popular and assumed to be the global test.

Let $\hat{r}_j$ be the residual autocorrelation function (ACF) of order h, then

$$\hat{r}_j = \frac{\sum_{t=1}^{n-j}(\hat{\zeta}_t - \overline{\zeta})(\hat{\zeta}_{t+j} - \overline{\zeta})}{\sum_{t=1}^{n}(\hat{\zeta}_t - \overline{\zeta})^2}; \qquad j = 1, 2, \ldots, h,$$

where $\overline{\zeta}$ is the mean of n residuals.

We wish to check simultaneously that first h residual ACFs are not significantly different from zero,

$$H_0 : r_1 = r_2 = \ldots = r_h = 0.$$

If the residuals are really white noise, then the estimated correlation coefficients are asymptotically normal, with zero mean and variance $\frac{(n-h)}{n(n+2)}$.

Then, modified Ljung-Box test statistics is

$$Q(h) = n(n+2)\sum_{j=1}^{h}\frac{\hat{r}_j^2}{n-j}$$

$Q(h)$ is asymptotically distributed as χ^2 distribution with degrees of freedom $h - p - q$ for ARMA(p, q) process.

The Ljung-Box $Q(h)$ statistic is asymptotically equivalent to testing for an ARMA(p, q) model against ARMA$(p, q + h)$ or ARMA$(p + h, q)$ model. So Ljung-Box test can also be considered as an alternative for testing model overfitting. The

power of Ljung-Box depends on the choice of h. In the literature, simulation results by Davies and Newbold (1979) suggested that $h = O(ln\ \mathrm{n})$ is close to an optimal choice. Other alternative of portmanteau test, Durbin-Watson test can also be performed but Ljung-Box test is preferred since Durbin-Watson test tests the null hypothesis of absence of autocorrelation against the alternative hypothesis of first-order autocorrelation only, while real time series often present autocorrelation of higher order.

3.3.2 TESTING HOMOSCEDASTICITY AND ZERO MEAN OF ERRORS

Test of homoscedasticity and zero mean of residuals should be applied after checking that the residuals are uncorrelated, to ensure that $\hat{\sigma}^2$ is a reasonable estimator of the variance. Our interest is to test the hypothesis of

$$H_0 : \mu_\zeta = 0.$$

where μ_ζ is the error mean in the actual population.

Also, it's reasonable to assume that $\overline{\zeta}$ follows normal distribution with mean μ_ζ and variance $\dfrac{\hat{\sigma}^2}{n}$.

Then the test statistic is

$$Z = \sqrt{n}\left(\frac{\overline{\zeta}}{\hat{\sigma}}\right).$$

where $Z \sim N(0,1)$ and we reject null hypothesis for the large values of Z.

The homoscedasticity of the marginal variance of the residuals is confirmed by studying the graph of the residuals over time. If in the view of the estimated residuals there seems to be a change of variance from a point $t = n_1$ on, we can divide the sample interval into two parts and apply the test of equality of variances (variance ratio test) based on the F distribution. In the hypothesis under which both sections have the same variance,

$$H_0 : \sigma_1^2 = \sigma_2^2$$

the test statistic

$$F = \frac{\sum_{t=1}^{n_1} \hat{\zeta}^2/n_1}{\sum_{t=n_1+1}^{n} \hat{\zeta}^2/(n-n_1)}$$

will be distributed approximately as an F distribution with n_1 and $(n - n_1)$ degrees of freedom.

In the same way, if we suspect there are m changes in variance in the periods $n_1, \ldots, n_m$ then we want to test

$$H_0 : \sigma^2 = \sigma_1^2 = \ldots = \sigma_m^2.$$

simultaneously. The test statistic is

$$\lambda = nlog\hat{\sigma}^2 - \sum_{i=1}^{m} n_i log s_i^2$$

where $\hat{\sigma}^2$ is the variance of the residuals in the entire sample and s_i^2 is the variance in the section i of length n_i observations.

Test statistics λ is asymptotically a chi-square distribution with $m - 1$ degrees of freedom. To apply this test, it is advisable to have at least 10 observations in each section.

3.3.3 CHECKING NORMALITY OF RESIDUALS

Since our model is based on the assumption of normality of residuals, it is important to check whether the residuals from each intervals are normal or not. We can examine the normalized residual plots or normal quintile plots to see the discrepancy of the data from Gaussian distribution assumption. Together with these graphical methods we also need to use some other statistical methods to confirm the nature of the residuals. Shapiro-Wilk test and Kolmogorov-Smirnov tests are two common tests to check the normality of the data. Two sample Kolmogorov-Smirnov test is also used to compare the cumulative distribution function of two different samples. Both Shapiro-Wilk and Kolmogorov-Smirnov tests for normality calculate the probability that the sample was drawn from a normal distribution.

The hypothesis of interest is

$$H_0 : \zeta \sim GaussianDistribution.$$

The test statistic for Shapiro-Wilk test is

$$W = \frac{\left(\sum_{i=1}^{n} a_i \zeta_{(i)}\right)^2}{\sum_{i=1}^{n} \left(\zeta_i - \overline{\zeta}\right)^2},$$

where $\zeta_{(i)}$ is the i^{th} order statistic, and

$$\overline{\zeta} = \frac{\zeta_1 + \ldots + \zeta_n}{n}.$$

The coefficients a_i's are given by the mean and covariance matrix of i^{th} order statistic.

$$a_1, \ldots, a_n = \frac{m'V^{-1}}{(m'V^{-1}V'^{-1}m)^2}$$

where $m_1, \ldots, m_n$ are expectations and V is the covariance matrix of i^{th} order statistic. For more details readers are suggested to refer the paper by Shapiro and Wilks (1965). Small values of W are significant and indicate non-normality. Critical values of rejection region for distribution of W are given in their paper.

Shapiro-Wilks test does not work very well if several ties are present in the data and this test is very sensitive to the problems in the tail. So, Kolmogorov-Smirnov test is better option than Shapiro-Wilks test. Kolmogorov-Smirnov test is more robust and works reasonably well for the small data set also.

We use two sample Kolmogorov-Smirnov test to compare the empirical distribution functions of estimated residuals and original mixture white noise. We are interested in testing whether or not two sample distributions are same.

$$H_0 : \hat{F}_{1,n_1}(y) = \hat{F}_{2,n_2}(y)$$

where $F_{1,n_1}(y)$ and $F_{2,n_2}(y)$ are empirical cumulative distribution functions of two samples. The test statistic is

$$D = Sup.|\hat{F}_1(y) - \hat{F}_2(y)|.$$

The null hypothesis is rejected if

$$\sqrt{\frac{n_1 n_2}{n_1 + n_2}} D > k_\alpha \ ,$$

where n_1 and n_2 are sample sizes and k_α is such that $P(k < k_\alpha) = 1 - \alpha$.

It is suggested that all the diagnostic tests should be performed simultaneously one after another. The test of autocorrelation of residuals is the most important test should be performed first and other tests such as homoscedasticity with zero mean

and normality of residuals should be performed later.

We explained the key ideas for a breakpoint time series model. In the next section, we show how such model can be fitted to provide a reasonable approximation to the underlying time variation of error, and we also compare it with the classical approach where the measures of goodness of fit of data using AICc and BIC will be given, and implemented in R and SAS.

CHAPTER 4

APPLICATION

In this section, we test our proposed methodology on two different types of simulation data and also apply it to fit the actual data related to fish otoliths. The simulation will allow us to justify our methodology. As for simulated data, we simulate two different types of data. The first kind is a single ARMA model with mixture of Gaussian noise while the other type is the mixture of different ARMA models from different intervals. We implement our proposed method for forecasting and predictionn of both types of data and compare the result with the classical time series approach. Also, we used the proposed method for forecasting and model fitting of real data related to fish otoliths. Otoliths are the organs that detect sound and assist balance that are found in the inner ear. They are composed of calcium carbonate ($CaCO_3$) and trace elements which reflect environmental conditions. Otoliths accrete daily bands for the first year of life and yearly bands thereafter (Jones 2002). Each band contains a fingerprint of the water chemistry to which the fish was exposed, and thus provides a chronology of changing habitat (Campana 1999 and Dorval et al. 2007). Our otolith data spans fish-birth years from 1967 to 2000. Otoliths were measured for $\delta^{18}O$ (the ratio of the stable isotopes ^{18}O:^{16}O), a measure of the oxygen isotopes contained in their $CaCO_3$, that mirrors water temperatures and origin. In our example, we use fish-otolith data collected from Lake Tasiat in eastern Canada, near the Arctic Circle. This region has experienced varying temperatures and precipitation that may reflect climate change.

4.1 SIMULATED DATA

4.1.1 ARMA MODEL WITH MIXTURE OF GAUSSIAN NOISE

We simulate an ARMA model with mixture of Gaussian noise. An AR model with zero mean, AR component (ϕ) = 0.4 with mixture of two component Gaussian is simulated. The simulated mixture of Gaussian noise has the following parameters

$$p = 0.3, \mu_1 = 1, \sigma_1 = 0.6, \mu_2 = 3, \sigma_2 = 2.$$

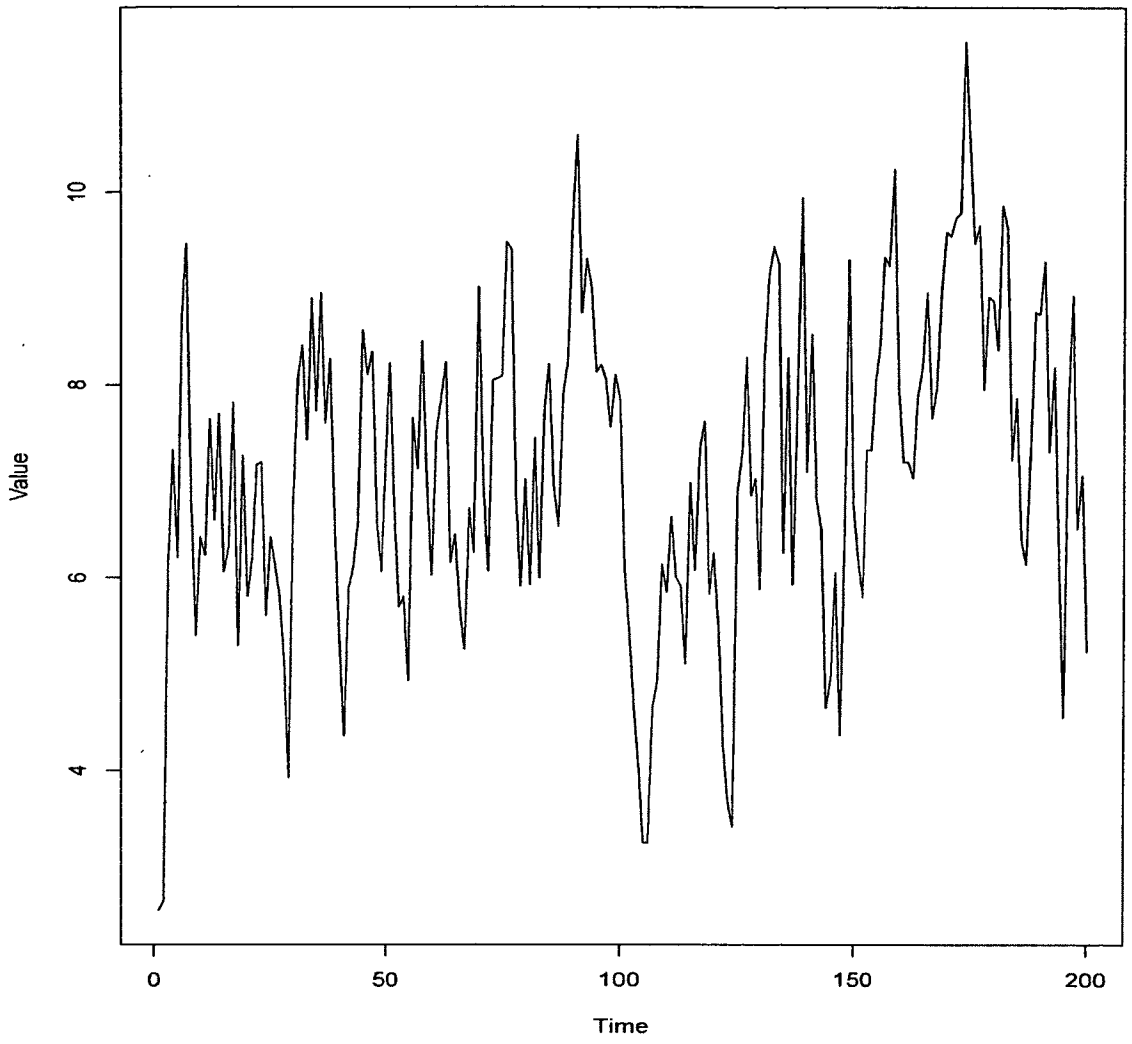

Figure 1. AR(0.4) model with noises from mixture of two Gaussians.

First, we fit the model based on classical approach, which fits an ARMA model for the entire data. Looking at figure 1 we can see that the data structure does not look same and it changes over two different time intervals. Also, there is no seasonal component associated with these data. So, we don't need to worry about fitting an ARIMA model. Also, both augmented Dickey-Fuller test and Philips-Perron unit root tests suggest that data is stationary (p-value=0.01). Once stationarity is established, now we want to see which model best fit the data. Based on maximum

likelihood method and minimum Akaike Information Criterion (AICc), we choose MA(4) model. The estimated parameters for this model are given in the table below. The values in the parenthesis are the standard error estimates.

Table 1. Parameter estimates and se() of MA(4) model for simulated data

$\hat{\mu}$	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$	$\hat{\sigma}^2$	AIC	BIC	log-lik
3.97	0.28	0.13	0.14	-0.12	3.05	802.90	822.69	-395.45
(0.18)	(0.07)	(0.08)	(0.08)	(0.07)				

Also, table 2 shows the AICs using maximum likelihood method for different combinations or AR(p) and MA(q) components. Notice that for MA(4) we achieve minimum AIC.

Table 2. Summary of AICs using different combinations of p and q for best model selection of simulated data

q→ / p↓	0	1	2	3	4	5
0	0.0	806.8	807.6	804.0	**802.9**	803.9
1	804.9	806.6	808.5	804.1	803.5	805.7
2	806.6	808.6	808.6	805.5	805.5	805.6
3	808.6	807.4	806.1	805.1	806.8	801.5
4	803.7	803.9	805.8	806.6	807.9	806.4
5	803.7	805.4	806.4	802.5	802.5	804.5

Also, predicted values and twenty future forecast values using MA(4) model together with the original data are plotted in figure 2. In the figure the yellow band after time 200 represents the prediction bound for forecasting.We can clearly see that model fitting is not very good and many cyclic variations are not captured. Forecasting is even worse, it has large prediction bounds and forecast looks constant. Here, the model based on classical approach fails to incorporate the cyclic variation in the forecast and it's not capturing the change of data structure over different intervals

of time. We overcome these problems by using breakpoints and mixture distribution based forecasting discussed in previous chapters.

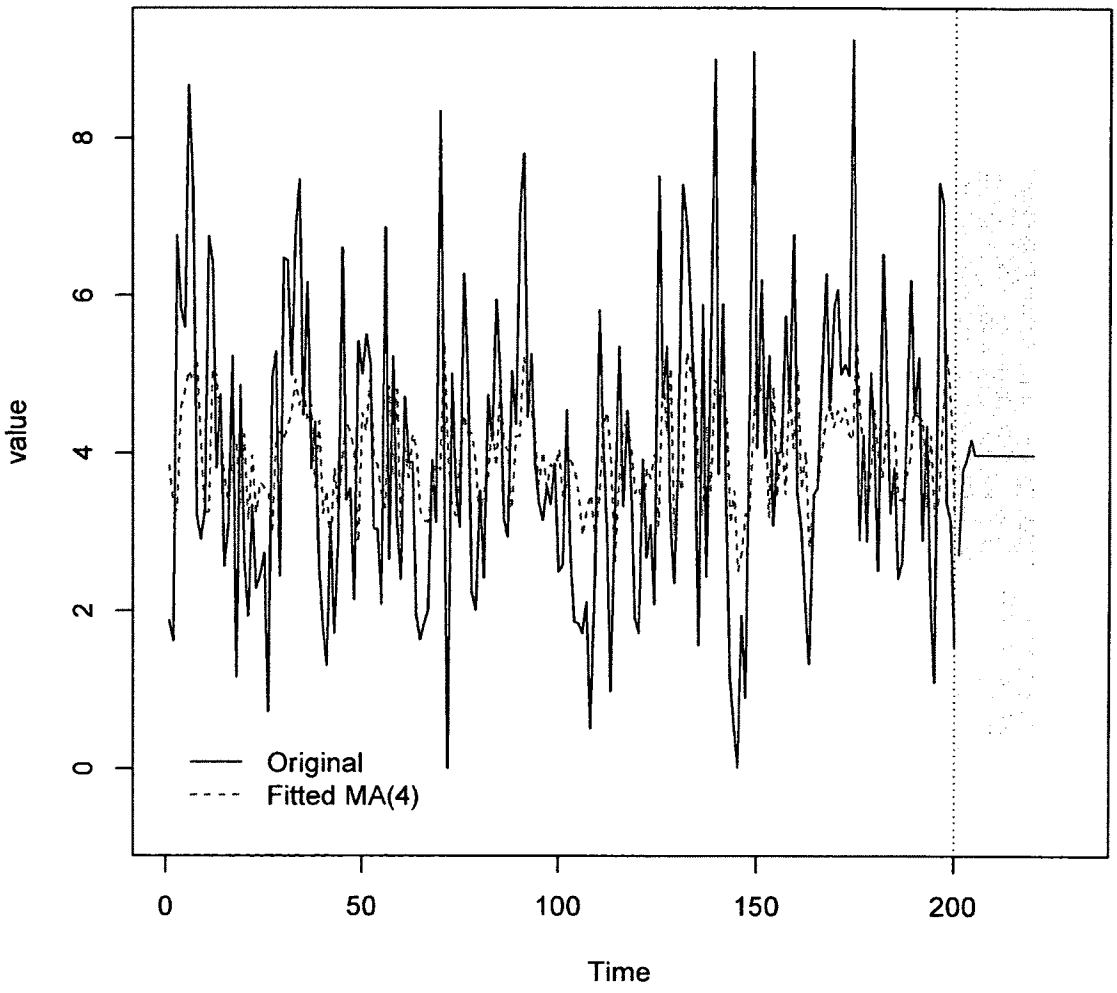

Figure 2. Forecasting and model fitting of simulated data .

In the next step, we use our proposed method to the data. First, we identify the breakpoints in the data set and divide it into different intervals. Then we fit separate models for the data in each interval. Breakpoints are identified according to the method discussed in the section 2.2. Using R package *strucchange* it is reasonable to use one breakpoint in the data set. Figure 3 shows different values of Bayesian

Information Criterion (BIC) and Residual Sums of Squares (RSS) for different breakpoints. Our goal is to take the optimal solution and it is reasonable to consider one breakpoint. Also, if we choose more than one breakpoint, we may encounter the problem of overfitting.

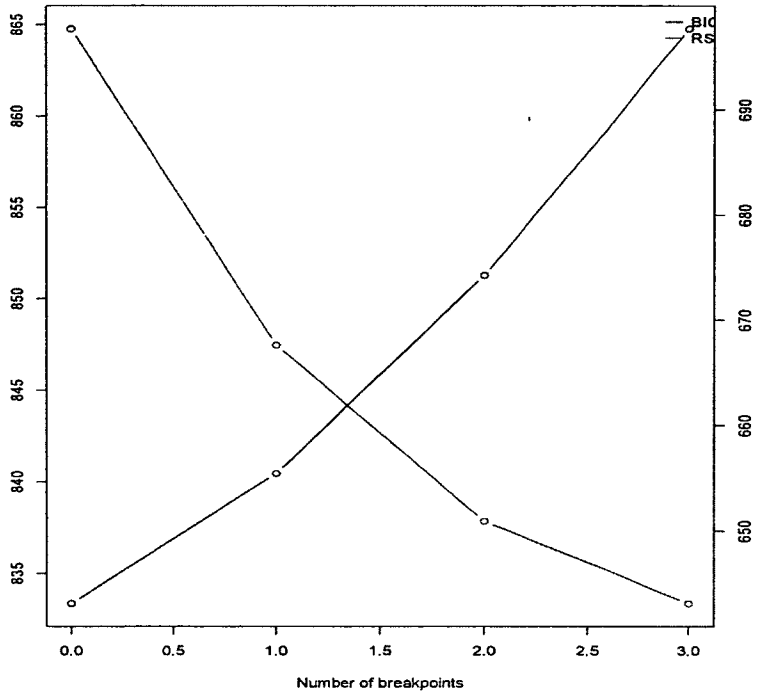

Figure 3. Breakpoint identification of simulated data.

From figure 3 it is clear that at one breakpoint we get BIC= 840.4 and RSS=667.6. These values are close to the possible minimum values of BIC (= 833.4) and RSS (= 697.7). In the data set, this breakpoint lies in the 124^{th} observation, so we divide the data into two parts, 1-124 and 125-200. Now, we fit different ARMA models to these two parts and combine the error distributions.

In the data of both parts no non-stationarity is evident. Phillips-Perron Unit root tests suggest the stationarity of the data in both intervals. Also, there are no seasonal or periodic components in the data set, so we use ARMA based models on both parts. Based on the method of maximum likelihood and minimum AIC, We choose MA(3) and ARMA$(2, 2)$ models for first and second parts respectively.

Tables 3 and 4 show the AIC values for different combinations of AR (p) and MA(q) components.

Table 3. Summary of AICs using different combinations of p and q for first part (1-124 observations) of simulated data

q→ / p↓	0	1	2	3	4	5
0	0.0	487.8	489.6	**486.1**	486.7	488.5
1	486.8	488.3	490.2	487.0	488.6	490.5
2	488.5	490.3	488.9	487.7	489.1	489.8
3	489.2	488.0	490.0	488.1	489.6	491.2
4	488.3	489.8	490.3	489.4	490.9	492.9
5	489.3	491.0	486.5	487.0	492.9	494.9

Table 4. Summary of AICs using different combinations of p and q for second part (125-200 observations) of simulated data

q→ / p↓	0	1	2	3	4	5
0	0.0	319.5	320.7	320.9	319.4	320.1
1	318.9	320.9	322.5	321.6	316.3	318.3
2	320.9	322.9	**315.2**	321.3	318.3	320.3
3	321.9	318.3	317.2	317.9	320.2	322.3
4	316.2	317.9	319.7	319.5	315.8	317.5
5	318.0	319.8	321.7	320.9	317.4	316.7

Parameter estimates of best models are given in tables 5 and 6.

Table 5. Parameter estimates and se() of MA(3) model for part 1 of simulated data

$\hat{\mu}$	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\sigma}^2$	AIC	BIC	log-lik
3.76	0.35	0.14	0.24	2.72	486.06	500.16	-238.03
(0.25)	(0.09)	(0.11)	(0.10)				

Table 6. Parameter estimates and se() of ARMA(2,2) model for part 2 of simulated data

$\hat{\mu}$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\sigma}^2$	AIC	BIC	log-lik
4.30	1.34	-0.94	-1.13	0.87	3.08	315.23	329.21	-151.61
(0.25)	(0.07)	(0.06)	(0.12)	(0.15)				

Notice that all comparative measures such as AIC, BIC and Log-likelihood of the models obtained by using breakpoints (tables 5 and 6) are significantly improved compared to the model obtained by using classical approach (table 1).

We also check the residuals from each of these fits to see whether or not they meet the autocorrelation test and normality test with homoscedastic variance. Both Box-Pierce and Ljung-Box portmanteau tests suggest independence and white noise property of the data. P-values for Box-Pierce and Ljung-Box tests are 0.79 and 0.78, so we fail to reject the null hypothesis that "data are independently distributed". Figures 4 and 5 show there is no serious autocorrelation between the residuals. Also, residuals from both intervals meet the criterion of normality separately. Also, for the model based on classical time series approach, the residuals are not normal. Several model selection methods based on AIC, BIC and minimum variance were tried and in all cases residuals were not normally distributed. This is reasonable, because we intentionally simulated the model with mixture Gaussian noise and classical time series approach fails to handle this situation.

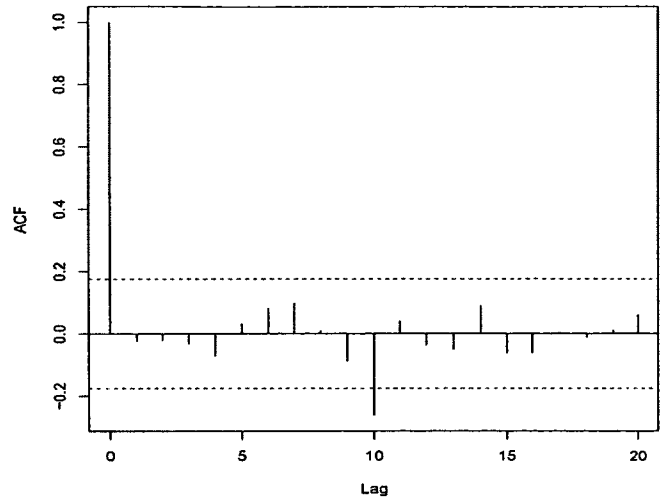

Figure 4. Autocorrelation function of residuals from part 1 (1-124 observations).

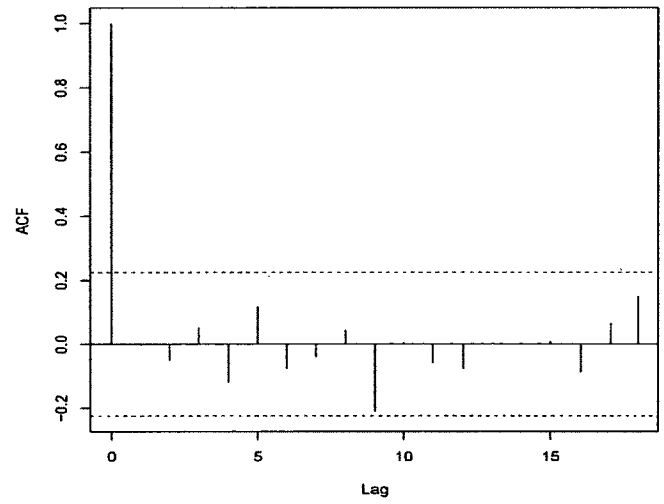

Figure 5. Autocorrelation function of residuals from part 2 (125-200 observations).

The process is invertible, so we can get the actual data from the errors. So for forecasting we combine the errors from both parts and estimate the mixture parameters of mixture of two component Gaussian distributions using the EM algorithm. Figure 6 shows the histogram of combined noise which seems right skewed from normal distribution, infact it is the mixture of normal distribution. Also, for combined noise we don't see significant autocorrelation (figure 7), we fail to reject both Box-Pierce

(p-value=0.74) and Box-Ljung(p-value=0.74).

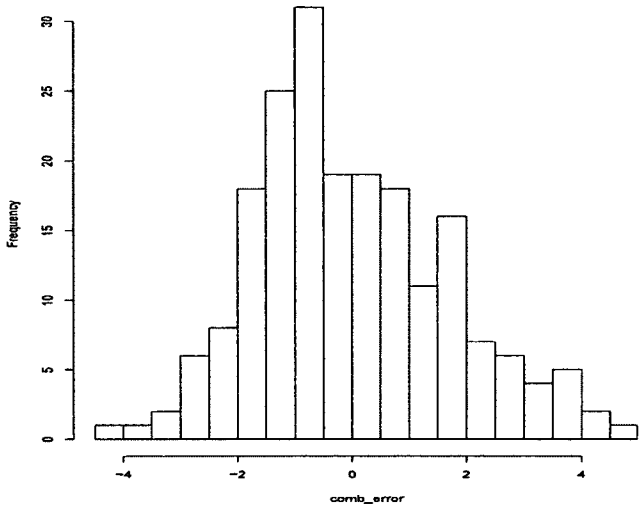

Figure 6. Histogram of combined residuals of simulated data.

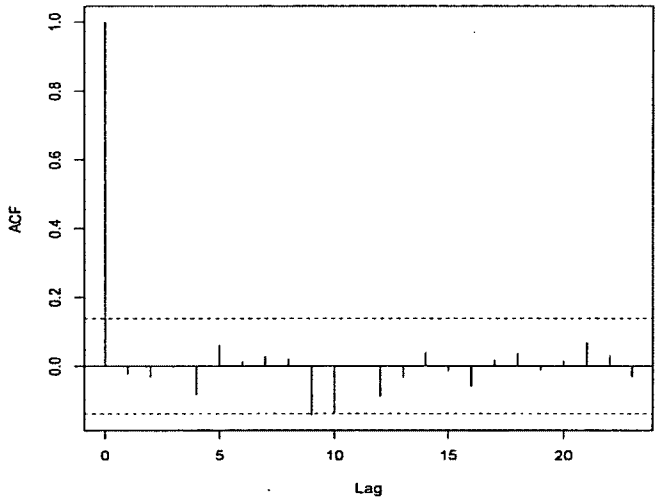

Figure 7. Autocorrelation of combined residuals of simulated data.

It's a reasonable assumption to consider that the joint distribution is the mixture of Gaussian distribution since the source of residuals are different. We use EM algorithm to estimate the model parameters of this mixture distribution. Parameter estimation of mixture of two component Gaussian using EM algorithm is presented

in table 7. Also, figure 8 shows the estimated mixture density tothether with individual Gaussian component density for combined data.

Table 7. Parameter estimates of combined residuals using EM algorithm

p	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}_1$	$\hat{\sigma}_1$
0.32	-1.01	0.48	0.76	1.80

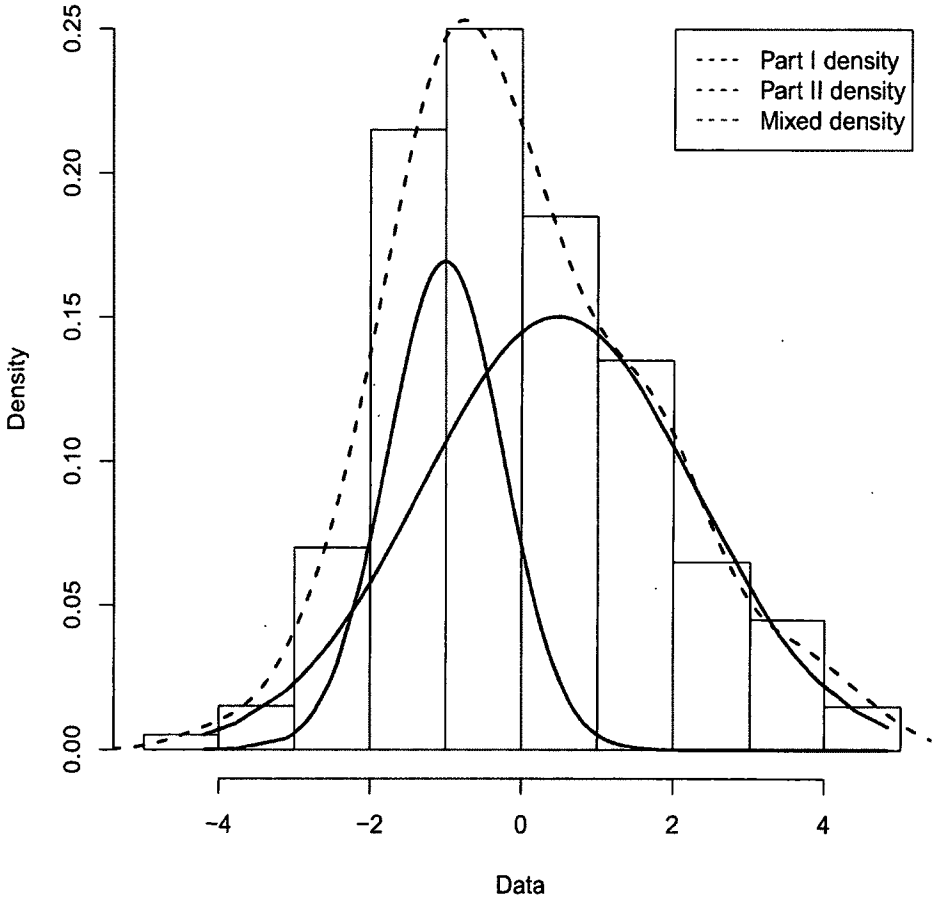

Figure 8. Density of the mixture distribution together with individual component distribution for combined residuals.

Now we forecast the next 20 future values using the theory already discussed in

section 3.1.3.

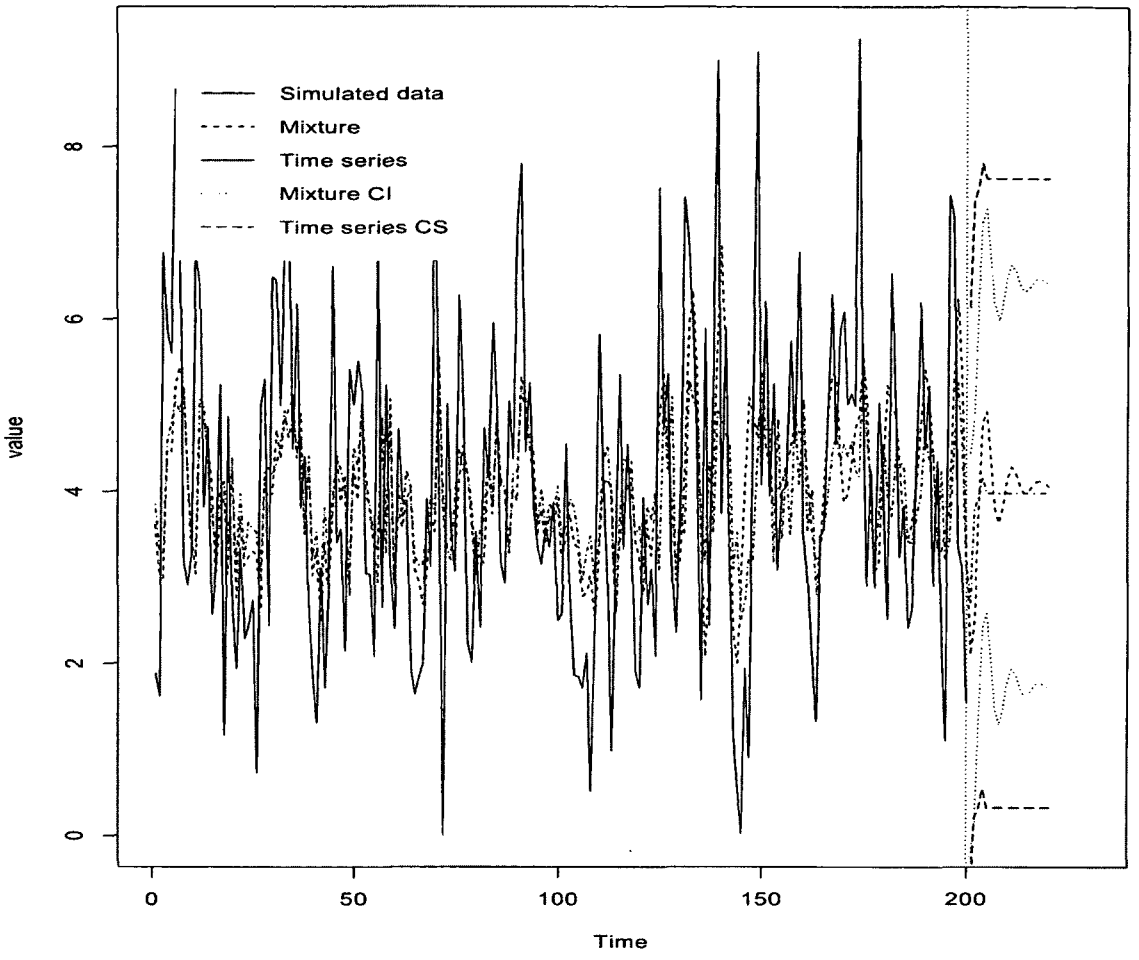

Figure 9. Model fitting and forecasting of simulated data by using classical time series and mixture model approaches.

In figure 9 we can see that the model fitting has improved significantly by using the mixture model. Most importantly, forecasting of the data has significantly improved. The mixture model forecasting also incorporates the cyclic factor of the data and the prediction intervals are narrower than those of classical approach.

4.1.2 MIXTURE OF DIFFERENT ARMA MODELS

We simulate a time series data with different covariance structures in two different intervals. Combination of AR(1), and MA(2) is simulated. For the first interval we assume an AR(1) model, and for the second interval, MA(2). The two models are generated with equal sample size, $n_1 = n_2 = 100$. For the AR(1) model, we use AR component $\phi_1 = 0.7$, zero mean with variance $\sigma_1^2 \approx 1$. For MA(2) model, we use the moving average components $\theta_1 = 0.5$ and $\theta_2 = 0.4$, mean value of 3 and variance $\sigma_2^2 \approx 1$.

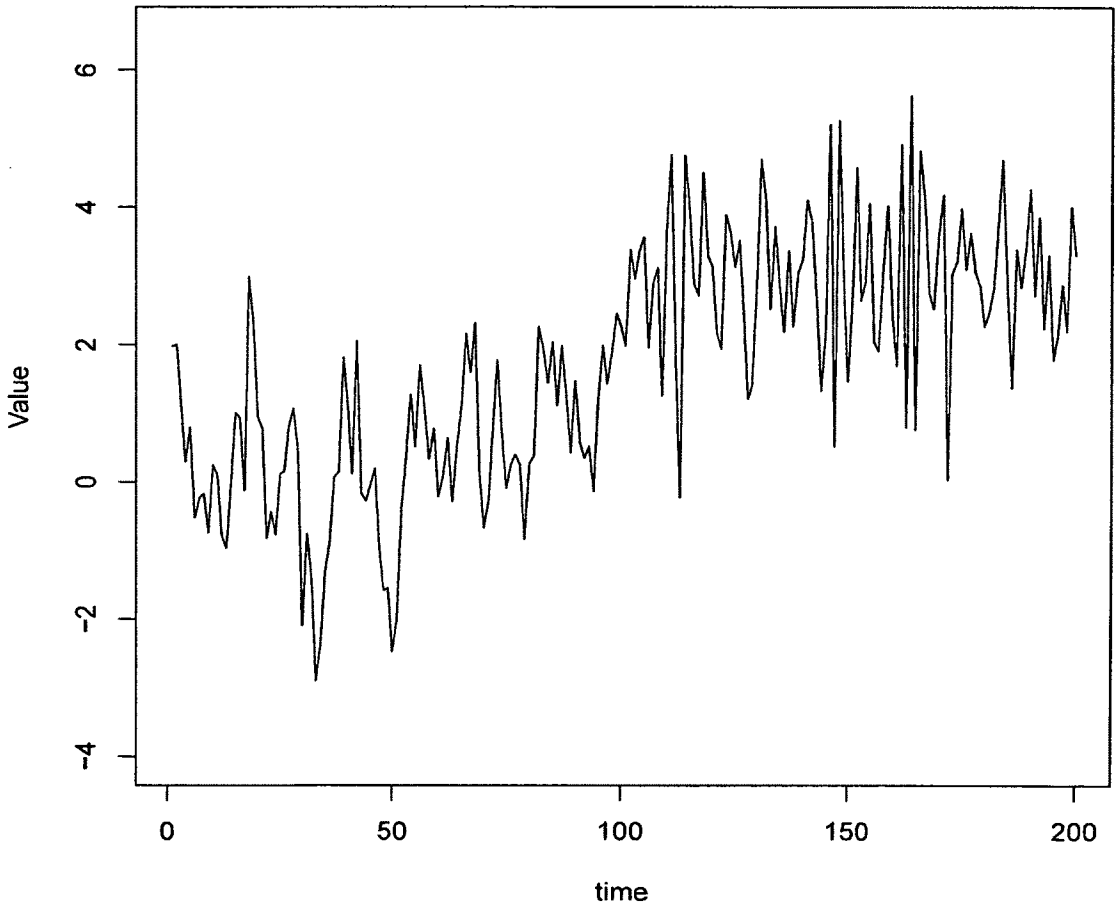

Figure 10. Simulated data using AR(1) and MA(2) mixtures .

We fit the regular time series data based on optimal values of AIC, BIC and variance. We used SAS to identify the best model to fit the data. Based on classical

time series approach, AR(3) is the most reasonable model with minimum variance to fit the combined data. Parameter estimates based on AR(3) model is presented in table 8.

Table 8. Parameter estimates and se() of AR(3) model for simulated data

$\hat{\mu}$	$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\phi}_3$	$\hat{\sigma}^2$	AIC	BIC	log-lik
1.78	0.34	0.26	0.23	1.44	644.66	0.40	-318.33
(0.46)	(0.07)	(0.07)	(0.07)				

Figure 11 shows the actual simulated data together with the fitted values and 25 future forecast based on classical time series approach.

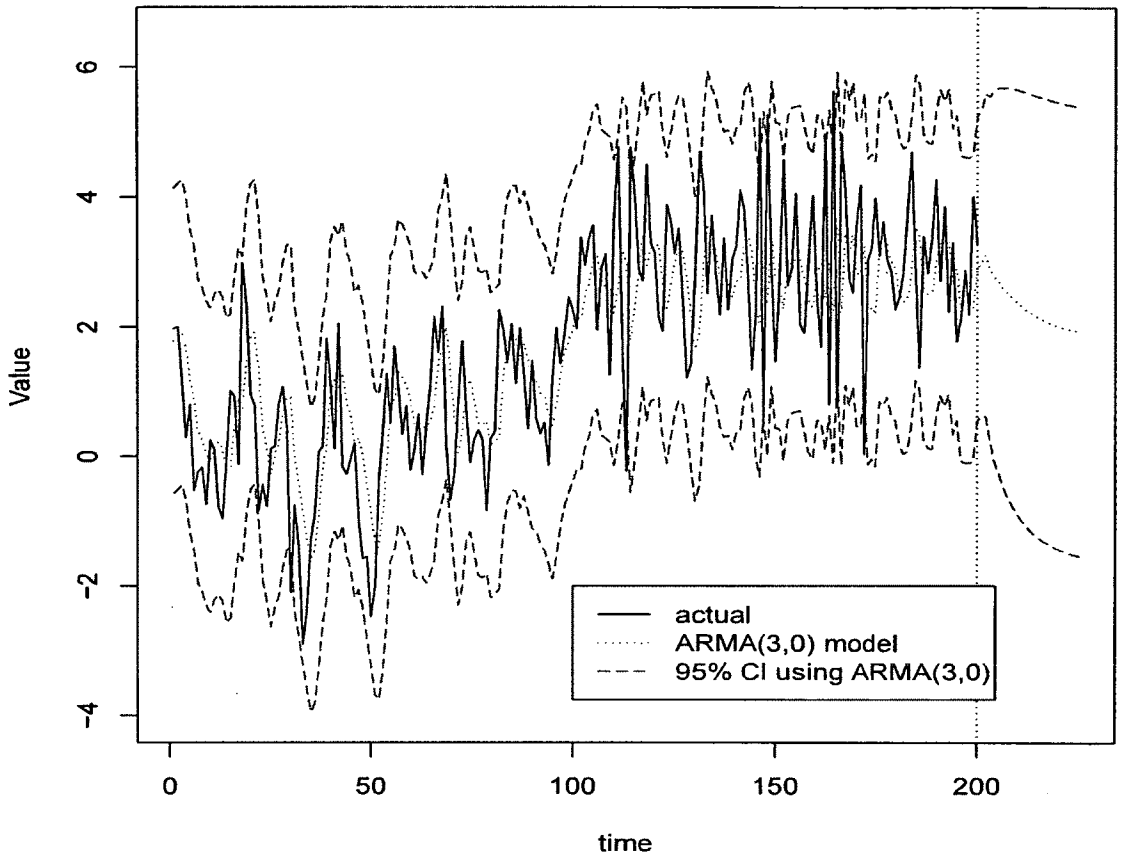

Figure 11. Model fit and forecasting for simulated data using AR(3) model.

This simulation is chosen intentionally so that the model based on classical time series approach fits the data very well. Our goal is to show that even for such data, our proposed method outperforms the classical method.

We improved the forecasting by identifying the breakpoints where the data structures were different. Then, we fit different time series models for each intervals. Figure 12 shows that one breakpoint is reasonable. There is no significant difference in the BIC for choosing one or two breakpoints. Also, choosing more breakpoints may result in the over parametrization of the problem, so we choose one breakpoint. The breakpoint is observed at 98^{th} observation which is very close to our simulation grouping of 100-100 observations.

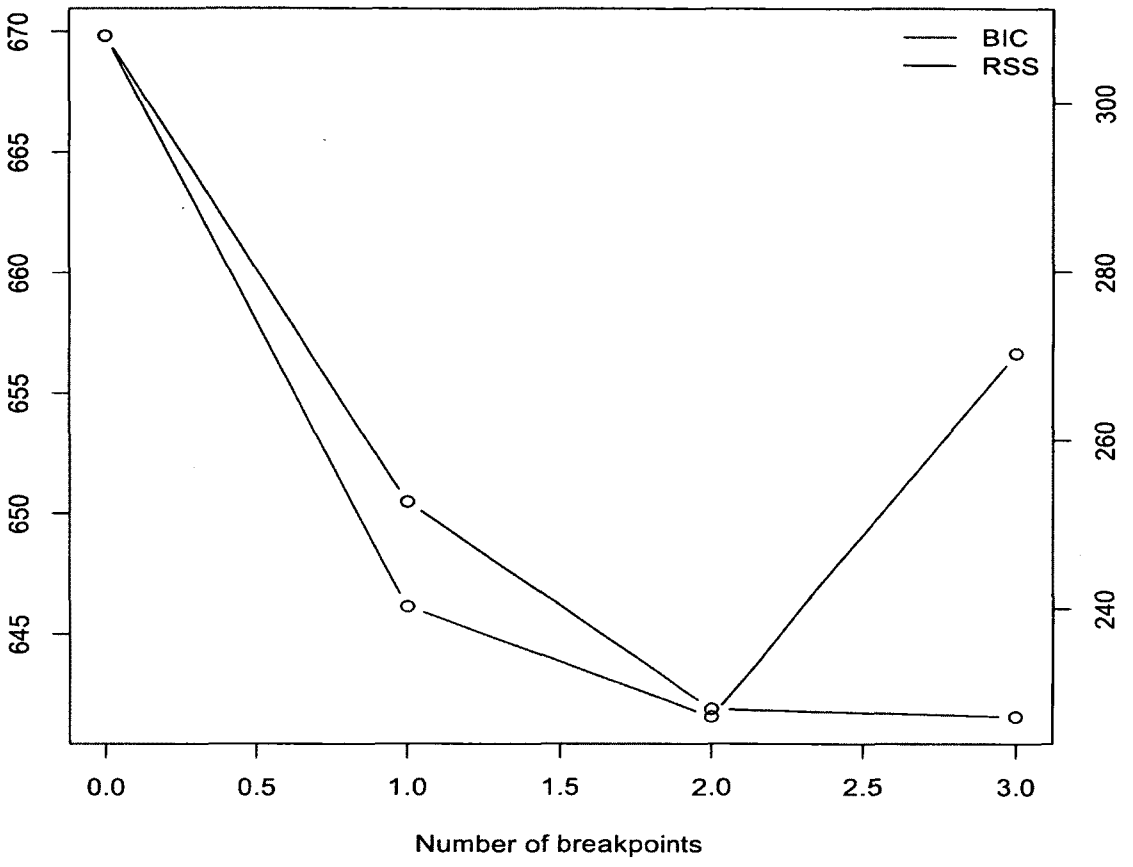

Figure 12. Identification of breakpoints for simulated data.

Once the breakpoint is identified, we fit different time series models for each intervals and the residuals from each intervals are combined and their joint density is estimated. Table 9 and 10 show the parameter estimates of the best models based on minimum AIC for the first 97 observations and last 103 observations of the simulated data. For the first interval (1-97 observations), AR(1) fits the data very well and for the second interval (98-200 observations) , MA(2) fits the data.

Table 9. Parameter estimates and se() of AR(1) model for first part (1-97 observations) of simulated data

$\hat{\mu}$	$\hat{\phi}_1$	$\hat{\sigma}^2$	AIC	log-lik
0.53	0.67	0.81	264.61	-130.31
(0.27)	(0.08)			

Table 10. Parameter estimates and se() of MA(2) model for second part(98-200 observations) of simulated data

$\hat{\mu}$	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\sigma}^2$	AIC	log-lik
3.0	0.51	0.31	1.03	290.14	-142.07
(0.02)	(0.10)	(0.10)			

Assume that the residuals from two intervals are normally distributed. The parameters of mixture distribution are estimated by the EM algorithm. In our simulated data, the initial values for EM algorithm for mixture of two normal densities are taken as sample mean and variance of two error components. In our case, the two error components have means close to zero and variances are 0.80 and 1.01 for the first and second intervals, respectively. The estimated weights (proportions) using EM algorithm for both groups is 0.50.

These estimates are used to generate the mixture distribution for forecasting. Model fit by using breakpoints and forecasting is shown in Figure 13. Figure 14 shows the comparison between mixture model and classical time series model.

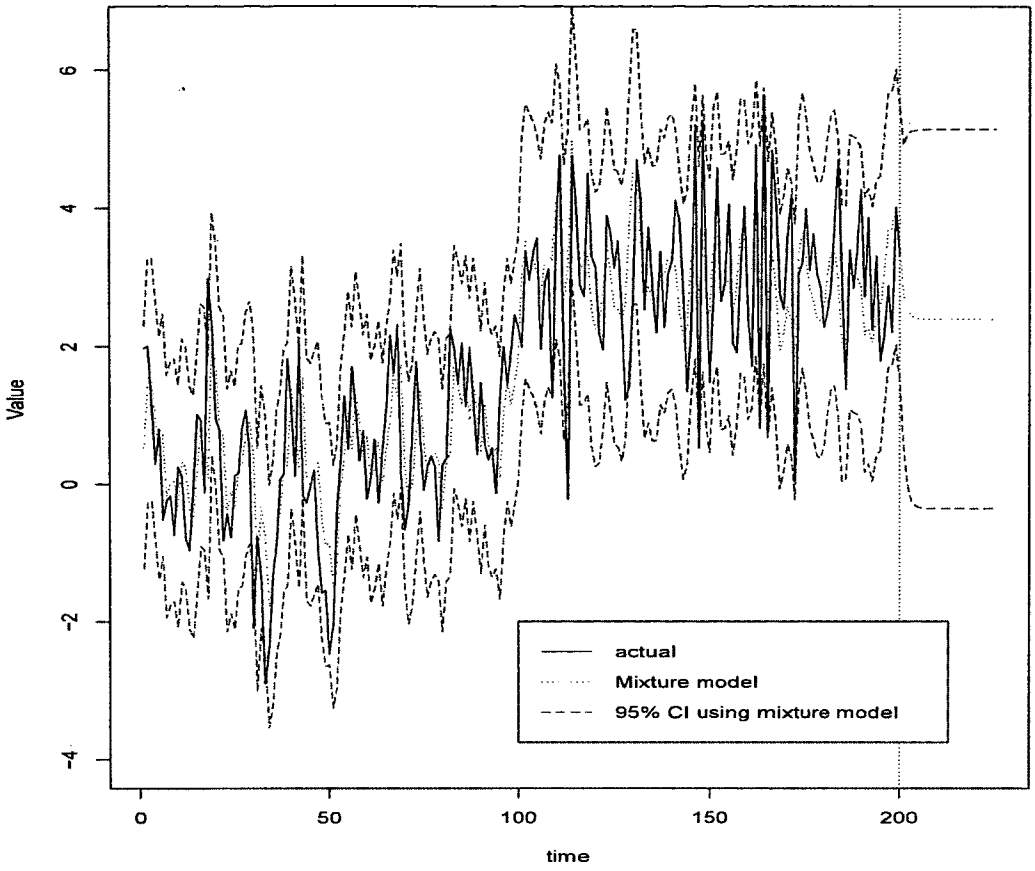

Figure 13. Model fit for simulated data using mixture model.

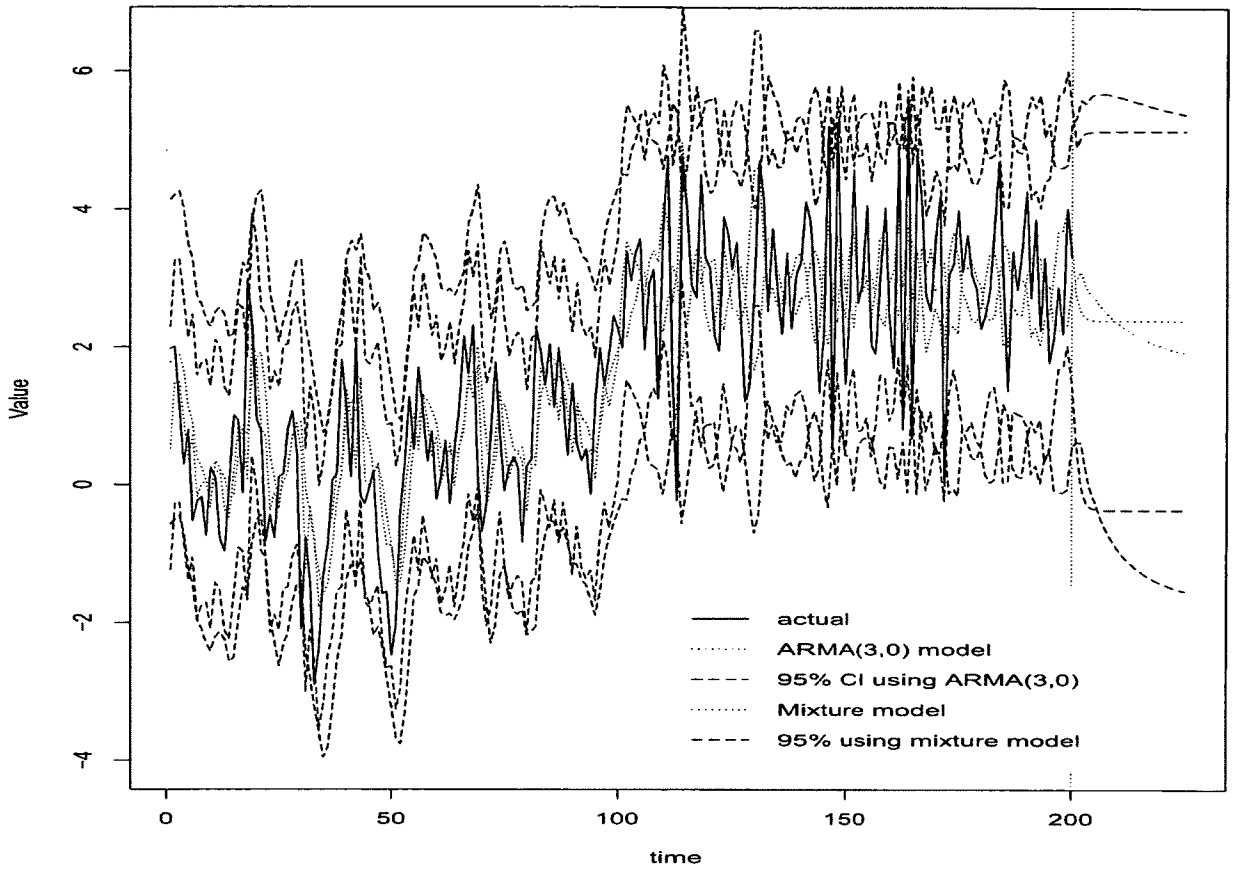

Figure 14. Model fit for simulated data.

The Mixture model has significantly improved model fitting and forecasting. The overall variance of the combined residual is 0.90, and the confidence intervals generated by using mixture model is narrower than that generated by regular time series model, AR(3). We also compared the similarity of the theoretical and actual cumulative distribution function (CDF) of the combined error. Kolmogorov-Smirnov test shows that theoretical CDF based on mixture model is not significantly different (p=0.167) from actual combined error at 5 percent level of significance. Figure 15 compares the empirical CDF of actual error with theoretical CDFs based on classical time series model and proposed mixture model.

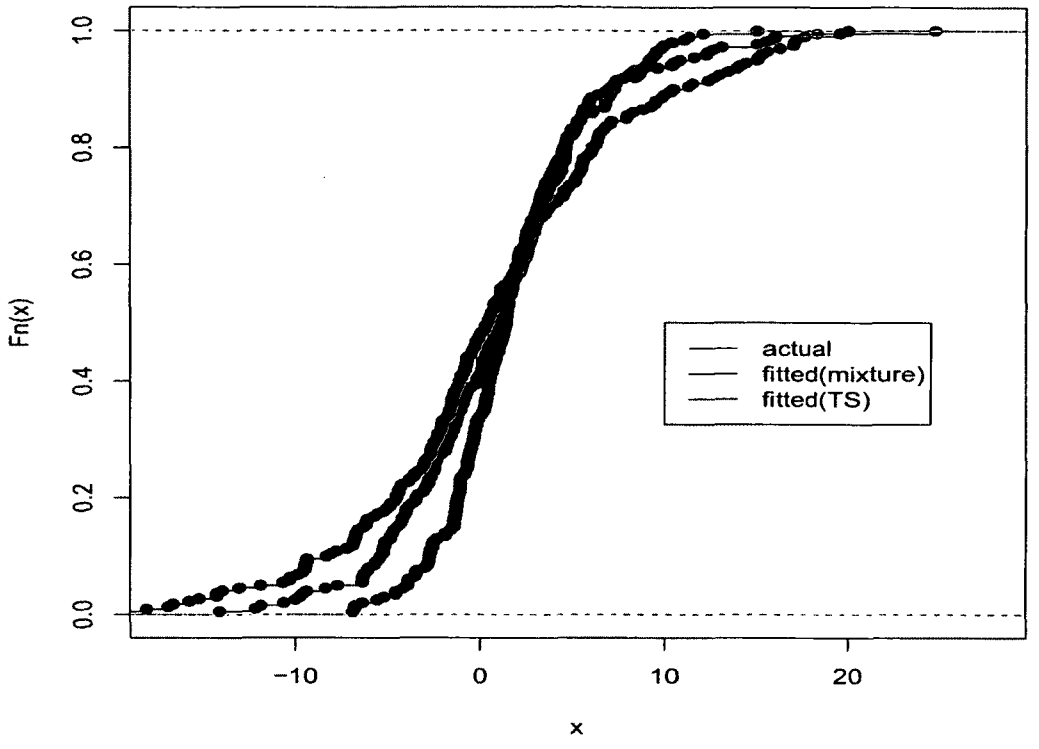

Figure 15. Comparison of empirical CDFs for simulated data.

4.1.3 OTOLITH DATA

We implement our methodology using to otolith data obtained from Lake Tasiat, Canada. The study of O18 (Oxygen Isotope $\delta^{18}O$) from fish otoliths is useful in estimating historical water temperature and weather. Lake Tasiat has information from years 1967 to 2000. Appendix A shows Tasiat data with other covariates. Our main interest is to see the overall change of O18 over time and predict how it will behave in future. Average precipitation, average temperature and average rain are the available covariates which may cause changes in O18. Table 11, shows the summary of data from Lake Tasiat. The effects of these covariates are not significant in the linear regression model. The regression coefficients of average precipitation, average temperature and average rain are 0.0355, -0.00035 and 0.021, respectively.

Table 11. Mean and standard deviation of covariates

Avg. O18		Avg. Temp.(^oC)		Avg. Rain (mm)		Avg. Prec. (mm)	
Mean	sd	Mean	sd	Mean	sd	Mean	sd
-12.62	0.33	-5.70	1.25	22.81	4.34	43.88	5.02

Since, none of the covariates are significant in the linear regression, we fit the time series model only for average O18 and the effect of other covariates are ignored. Classical time series approach based on minimum AIC identifies AR(1) as the best model to fit the data. Estimate of model parameters are listed in Table 12.

Table 12. Parameter estimates and se() of AR(1) model forTasiat data.

$\hat{\mu}$	$\hat{\phi}_1$	$\hat{\sigma}^2$	AIC	BIC	log-lik
-12.60	0.47	0.09	15.50	-2.76	-5.75
(0.09)	(0.17)				

Figure 16 shows the original otolith data together with the fitted AR(1) model based on classical time series approach. We see that the model based on the classical time series approach does not fit the data very well.

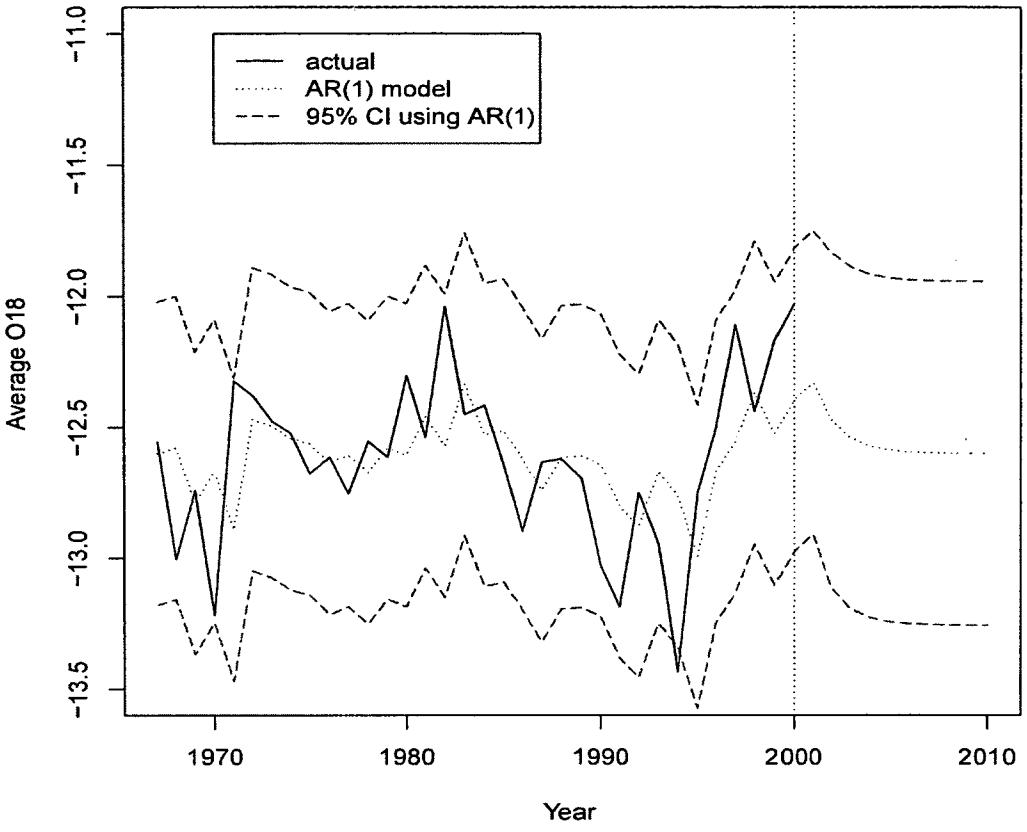

Figure 16. Otolith data and predicted values using AR(1) model.

In order to improve the parameter estimation and for better prediction, our proposed model based on mixture distribution is used. Based on minimal RSS and BIC, one breakpoint is identified. Figure 17 shows the breakpoint identification with possible RSS and BIC. Breakpoint is identified on 23^{rd} observation with corresponding break date 1989. The data are divided into two groups: group 1 with first $1-22$ observations corresponding to years $1967-1988$, and group 2 with observations corresponding to years $1989-2000$. These consecutive groups are significantly different in their regression parameters. The horizontal line specifies the RSS and BIC corresponding to the break date 1989.

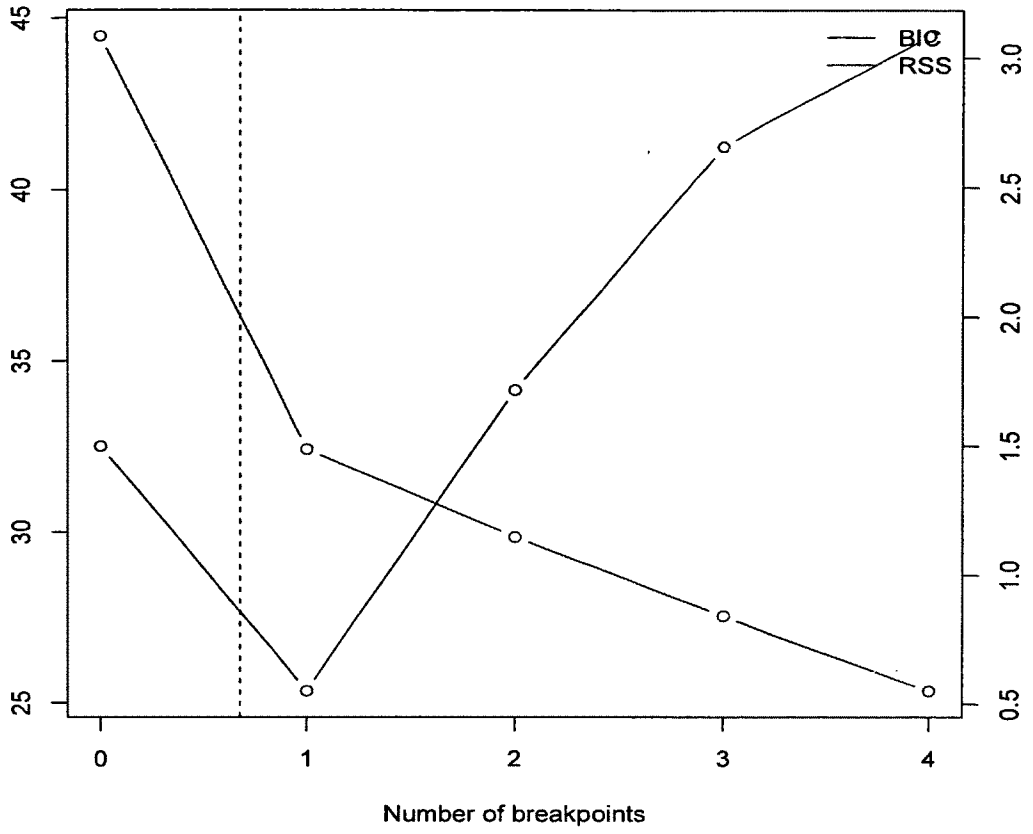

Figure 17. Breakpoint identification of otolith data using window width $h = 0.2$.

Different time series models based on maximum likelihood method are fitted to the intervals separated by the breakpoint. Using minimum AIC criterion ARMA(1,1) fits the first part of the otolith data while AR(1) fits the last part. Parameter estimation of fitted models are presented in Tables 13 and 14. The variance of the combined residual is 0.07. Since there are few observations in the last part (only 12 observarions) of the otolith data, we investigate the sampling distribution of mean and AR(1) parameters using block bootstrap. Figure 18 shows the sampling distribution of AR(1) parameters for Tasiat data (1989-2000).

Table 13. Parameter estimates and se() of ARMA(1,1) model for the first part of otolith data

$\hat{\mu}$	$\hat{\theta}_1$	$\hat{\phi}_1$	$\hat{\sigma}^2$	AIC	BIC	log-lik
-12.59	0.24	0.42	0.07	5.36	-6.17	0.32
(0.07)	(1.12)	(1.05)				

Table 14. Parameter estimates and se() of AR(1) model for the second part of otolith data

$\hat{\mu}$	$\hat{\phi}_1$	$\hat{\sigma}^2$	AIC	BIC	log-lik
-12.61	0.68	0.13	11.58	-5.57	-3.79
(0.02)	(0.10)	(0.10)			

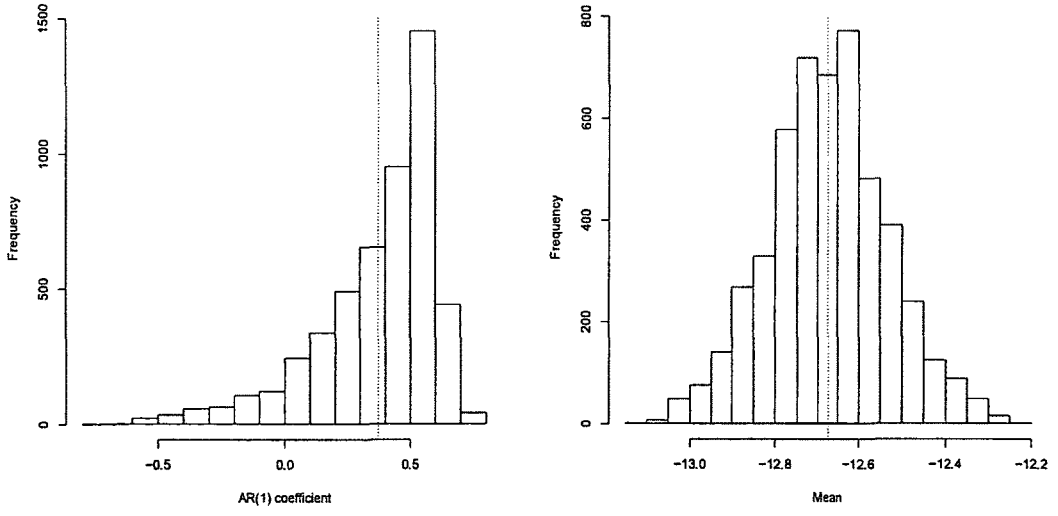

Figure 18. Sampling distribution of AR(1) parameters for Tasiat 1989-2000 data using block bootstrapping with block length 6.

Also, Figure 19 shows the prediction based on proposed mixture model. Model comparison between proposed mixture model and classical time series ARMA model

is shown in Figure 20. It is noticeable that the model based on our proposed mixture model has significantly improved the forecasting of the data. The confidence intervals based on mixture model is smaller than those based on classical time series model. Also, Figure 21 compares the empirical CDFs of Tasiat data with classical time series model and proposed mixture model. The CDF of fitted model based on mixture distribution is significantly closer to the true data.

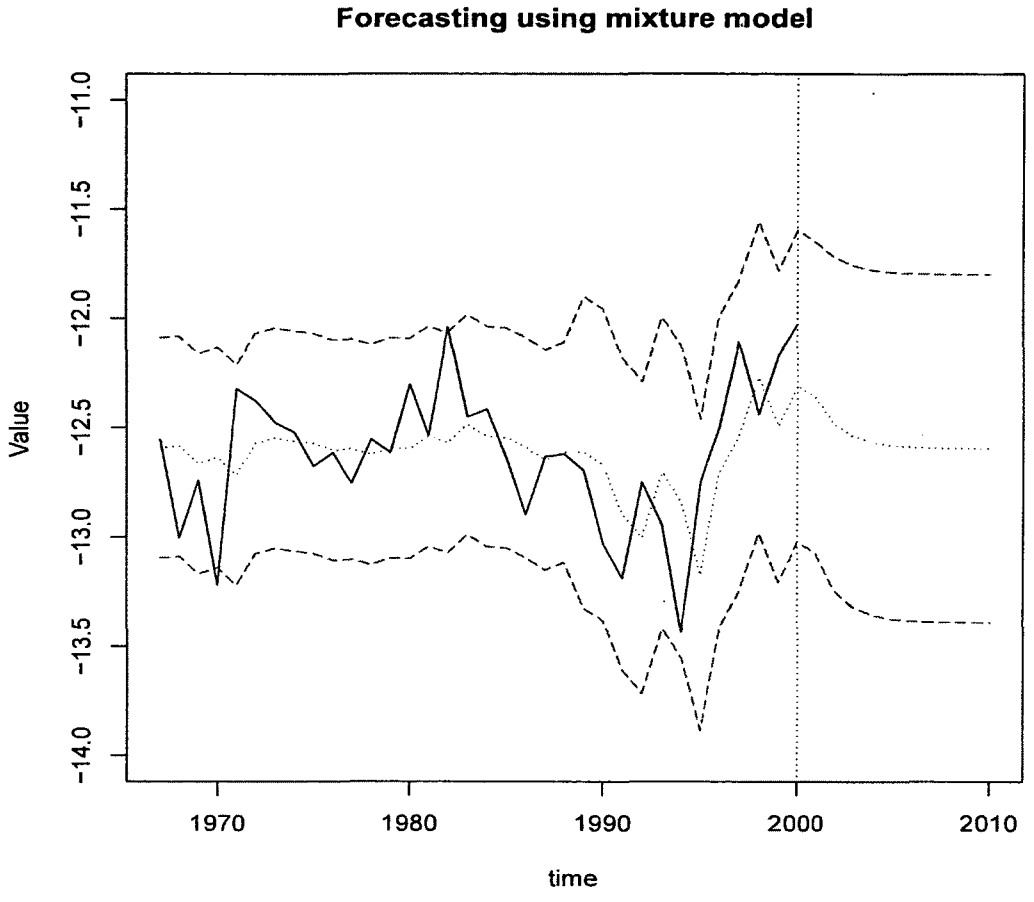

Figure 19. Actual data and fitted model using proposed mixture method, BPBF.

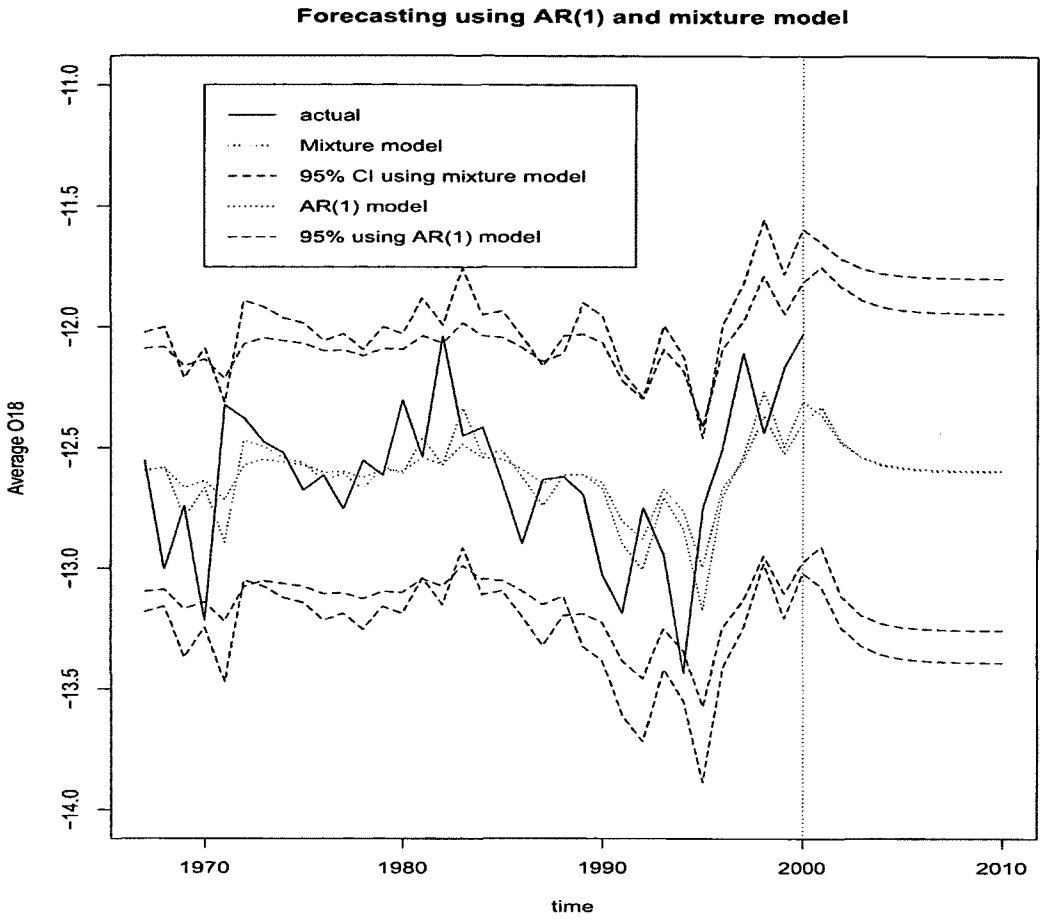

Figure 20. Actual data and fitted model using classical time series model AR(1) and proposed mixture model, BPBF.

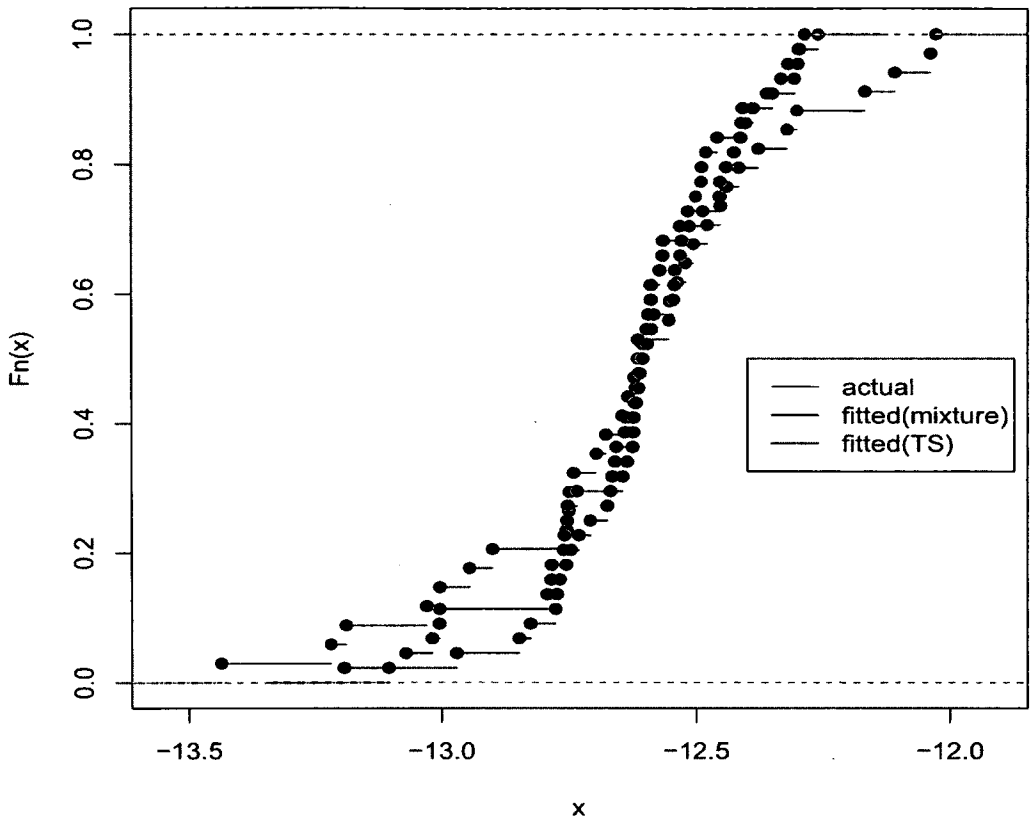

Figure 21. Empirical cumulative distribution functions of otolith data together with classical time series model and the proposed mixture model.

CHAPTER 5

CONCLUSION

We have introduced a non-linear dynamical probability time series model which exploits the idea of breakpoints together with bootstrapping and mixture distribution. Breakpoints partition the time course into consecutive non-overlapping intervals where the coefficients shift from one stable regression relationship to a different one. Also, because there are limited observations in some intervals, we use block bootstrapping to improve the parameter estimates. The optimal size of the blocks needed is chosen such that the RSS will be minimum. Once we fit the model for different intervals, such information is combined and used in forecasting.

Forecasting partitioned data which has different model structures at different partitions is a challenging task. Over the last decade there has been much interest in developing breakpoints to time series data in a small sample scale. To our knowledge, there are no existing methods that discuss the prediction of this type of data. We have shown numerically that the model accomodates data with different variance structures with the introduction of the breakpoints. The regression and the dependency of the parameters in the model have been included in an consistent and efficient manner. Regression models with unequal mixed sample frequencies and their advantages is still relatively the unexplored area (Andreou et al. 2002). Consistency is guaranteed since we are using the maximum likelihood method, efficiency is guaranteed since the bootstrap method is used to resample the data once the blocks have been identified and the predictions lie within the smaller range intervals than the classical time series modelling.

Our proposed method is different from other existing methods that are based on time series data in which different covariates have different covariance structures. Typically, for the models that are built with the predictors without the breakpoint inclusion does not provide substantial forecasting (Stock 2008). We have developed a new approach which advances previous concepts with new ideas for forecasting time series data that are subject to the structural breaks and non-equidistant time. Our approach is based on the mixture distribution where the parameters are estimated by using EM algorithm combined with bootstrapping. Our approach together with

block bootstrapping performs very well when faced with small and sparse data sets as we have shown in our real example. Our approach is quite general and can be implemented in different ways other than those documented. Pesaran et al. (2006) discussed similar type of data by using Bayesian approach by allowing the possibility of new breaks occurring over the forecast horizon. We assume that the existence of breakpoints in the forecast horizon is some how unrealistic. Our approach is based on past data within the intervals and we do not use the information of systematic breakpoints in the forecast horizon.

Further questions are being explored. One of the questions is related to the identification of optimal block size for block bootstrapping as discussed in Patton et al. (2009). Another concern is related to finding a procedure of choosing initial value in EM algorithm for faster convergence.

REFERENCES

Alogoskoufis, G.S. and Smith, R. (1991), "The Phillips Curve, the persistence of Inflation, and the Lucas Critique: Evidence from Exchange Rate Regimes," *American Economic Review*, 81, 1254-1275.

Andreou, E., Ghysels, E., and Kourtellos, A. (2002), "Regression Models with Mixed Sampling Frequencies," *Journal of Econometrics*, Vol. 158, Issue 2, 246-261.

Andrews, D.W.K. (2002), "Higher-order Improvements of a Computationally Attractive k-step Bootstrap for Extremum Estimators," *Econometrica*, 70, 119-162.

Ansley, C.F. (1979), "Least Squares Estimation of a Shift in Linear Processes," *Journal of Time Series Analysis*, 15, 453-472.

Bai, J. (1994), "An algorithm for the exact likelihood of a mixed autoregressive moving average process," *Biometrica*, 66, 59-65.

Bai, J. (1997a), "Estimating Multiple Breaks One at a Time," *Econometric Theory*, 13, 315-352.

Bai, J. (1997b), "Estimation of a Change Point in Multiple Regression Models," *Review of Economics and Statistics*, 79, 551-563.

Bai J., and Perron, P. (1998),"Estimating and Testing Linear Models With Multiple Structural Changes," *Econometrica*, 66, 47-78.

Bai, J., and Perron, P. (2003), "Computation and Analysis of Multiple Structural Change Models," *Journal of Applied Econometrics*, 18, 1-22.

Bellman R. (1952), " On the Theory of Dynamic Programming," *Proceedings of the National Academy of Sciences*, 1952.

Box, G.E.P., Pierce, D.A.(1970), "Distribution of Residual Autocorrelations in Autoregressive Integrated Moving Average Time Series Models," *Journal of the American Statistical Association*, Vol. 65, Number 32.

Box, G.E.P., Jenkins, G.M., and Reinsel, G.C.(1994), *Time Series Analysis: Forecasting and Control*, Prentice-hall Inc., 1994 (3rd Ed.).

Brockwell, P.J., and Davis, R.A.(2002)), *Introduction to Time Series and Forecasting*, Springer-Verlag, New York (2nd Ed.).

Brockwell, P.J., and Davis, R.A.(1991)), *Time Series: Theory and Methods*, Springer-Verlag, New York (2nd Ed.).

Campana, S.E. (1999), "Chemistry and composition of fish otoliths: pathways, mechanisms and applications," *Marine Ecology Progress Series*, 188, 263-297.

Cappè, O., Moulines, E. and Rydèn T. (2005), *Inference in Hidden Markov Models*, Springer, 2005.

Casella, G., Fienberg, S., and Olkin, I. (2002), *Introduction to Time Series and Forecasting*,Wardsworth Group, Duxbury (2nd Ed.).

Casella, G. and Berger R.L. (2002), *Statistical Inference*, Springer-Verlag, New York (2nd Ed.).

Craigmile, P.F. and Titterington, D.M. (1997), "Parameter Estimation For Finite Mixtures of Uniform Distributions," *Communications in Statistics-Theory and Methods*,26(8), 1981-1995.

Davies, N. and Newbold, P. (1979), "Some power studies of a portmanteau test of time series model specication," *Biometrika*, 66, 153-155.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, B, 39,1-38.

Dorval, E., Jones, C.M., Hannigan, R., and Montfrans, J.V. (2007), "Relating otolith chemistry to surface water chemistry in a coastal plain estuary," *Canadian Journal of Fisheries Aquatic Sciences*, 64, 1-14.

Durbin, J. and Koopman, S.J. (2001), *Time Series Analysis by State Space Methods*, Oxford University Press, 2001.

Efron, B. (1979),*Bootstrap methods: Another look at the jackknife.* Ann. Statist. 7, 1-26

Fokianos, K., Kedem, B., Qin, J. , and Short, D.A. (2001), "A Semiparametric Approach to the One-Way Layout," *Technometrics*, 43, 56-65.

Garcia, R. and Perron, P. (1996), "An Analysis of the Real Interest Rate under Regime Shifts," *Review of Economics and Statistics*, 78, 111-125.

Ghysels, E., Santa, P.C., and Valkanov, R. (2006), "Predictiong Volatility: Getting the Most Out of Return Data Sampled at Different Frequencies," *Journal of Econometrics*, 131, 59-95.

Gilbert, P.B. (2000), "Large Sample Theory of Maximum Likelihood Estimation in Semiparametric Biased Sampling Models," *Annals of Statistics*, 28, 151-194.

Gupta, A.K. and Miyawaki, T. (1978), "On a Uniform Mixture Model," *Biometrica Journal*, 20, 631-637.

Hennig, C. (2004), "Breakdown Points For Maximum Likelihood Estimators of Location Scale Mixtures," *Annals of Statistics*, 32, 1313-1340.

Jones, C.M. (2002), *Age and Growth*, in Fisheries Science, [Editors] L.A. Fuiman and R.G. Werner, Blackwell Scientific, 30-63.

Kedem, B. and Gagnon, R. (2010), "Semiparametric Distribution Forecasting," *Journal of Statistical Planning and Inference*, 140 (2010) 3734-3741.

Koop, G. and Potter, S. (2001), "Are Apparent Findings of Nonlinearity Due to Structural Instability in Economic Time Series?," *Econometric Journal*, 4, 37-55.

Lahiri, S.N. (1999), "Theoretical Comparisons of Block Bootstrap Methods", *The Annals of Statistics*, Vol. 27, N0. 1, 384-404.

Ljung, G.M. and Box, G.E.P. (1978), "On the Measure of Lack of Fit in Time Series Models," *Biometricka*, 65, 297-303.

Pastor, L. and Stambaugh, R.F. (2001), "The Equity Premium and Structural Breaks," *Journal of Finance*, 56, 1207-1239.

Patton, A. , Politis D. N., and White H. (2009), "CORRECTION TO "Automatic block-length selection for the dependent bootstrap by D. Politis and H. White," *Econometric Reviews* 28 (4), 372-375.

Pesaran, M.H., Pettenuzzo, D., and Timmermann, A. (2006) "Forecasting Time Series Subject to Multiple Structural Breaks", *Review of Economic Studies* Wiley Blackwell, vol. 73(4), pages 1057-1084, October..

Politis, D.N. and Romano, J.P. (1994), "The stationary bootstrap," *Journal of American Statistical Association*, 89: 1303-1313.

Qin, J. (1993), "Empirical Likelihood in Biased Sampling Problems," *Annals of Statistics*, 21, 1182-1186.

Qin, J. and Lawless, J.F. (1994), "Empirical Likelihood and General Estimating Equations," *Annals of Statistics*, 22, 300-325.

Qin, J. and Zhang, B. (1997), "A Goodness of Fit Test for Logistic Regression Models Based on Case-Control Data," *Biometrica*, 84, 609-618.

Shapiro, S.S. and Wilk, M.B. (1965), "An Analysis of Variance Test for Normality (Complete Samples)," *Biometrika*, 52, 591-611.

Stock, J.H., and Watson, M.W. (2008), "Phillips Curve Inflation Forecasts", *Conference Series [Proceedings]*, Federal Reserve Bank of Boston.

Teicher, H. (1963), "Identifiability of Mixtures", *The Annals of Mathematical Statistics*, 32, 244-248.

Tiwari, R.C., Cronin, K.A., Fener, E.J., Yu B., and Chib, S. (2005), "Bayesian Model Selection For Join Point Regression with Application to Age-adjusted Cancer Rates", *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54, 919-939.

Wei, G.C.G, and Tanner, M. (1990), "A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation," *Journal of American Statistical Association*, 85, 699-704.

West, M. and Harrison, J. (1997), *Bayesian Forecasting and Dynamic Models*, Springer, 1997 (2nd Ed.).

Zeileis, A., Leisch, F., Hornik, K., Kleiber, C. (2002), "strucchange: An R Package for Testing for Structural Change in Linear Regression Models," *Journal of Statistical Software*, 7 (2), 1-38.

Zhang, B. (2000), "M-Estimation Under a Two Sample Semiparametric Model," *Scandinavian Journal of Statistics*, 27, 263-280.

APPENDIX A

OTOLITH DATA

Year	Average O18	Average Temp. (^0C)	Average Rain (mm.)	Average Precipitation (mm.)
1,967	-12.55	-6.62	22.18	42.39
1,968	-13.00	-4.77	27.19	47.73
1,969	-12.74	-4.98	18.18	45.35
1,970	-13.22	-5.95	17.37	42.57
1,971	-12.32	-5.84	24.05	41.60
1,972	-12.38	-9.08	20.70	39.90
1,973	-12.48	-4.58	24.49	50.02
1,974	-12.52	-6.35	17.22	39.06
1,975	-12.68	-5.74	23.89	42.96
1,976	-12.61	-6.78	20.53	38.48
1,977	-12.75	-4.34	24.42	41.93
1,978	-12.55	-6.98	28.06	49.53
1,979	-12.61	-5.28	27.15	51.91
1,980	-12.30	-5.06	16.74	38.08
1,981	-12.54	-3.21	23.46	44.63
1,982	-12.04	-6.43	22.37	42.68
1,983	-12.45	-6.23	18.41	31.84
1,984	-12.42	-5.73	23.32	40.91
1,985	-12.64	-5.22	21.85	45.32
1,986	-12.90	-6.63	17.56	43.23
1,987	-12.63	-5.75	20.60	43.04
1,988	-12.62	-5.88	22.32	38.97
1,989	-12.70	-6.90	15.22	40.13
1,990	-13.03	-6.68	16.66	38.66
1,991	-13.19	-6.50	20.33	37.39
1,992	-12.75	-7.88	27.63	48.42
1,993	-12.95	-6.75	26.99	50.33
1,994	-13.44	-5.75	19.66	39.78
1,995	-12.75	-4.94	27.69	44.65
1,996	-12.51	-4.19	24.60	48.73
1,997	-12.11	-5.20	33.67	53.08
1,998	-12.44	-4.04	30.02	46.58

Year	Average O18	Average Temp. (^0C)	Average Rain (mm.)	Average Precipitation (mm.)
1,999	-12.17	-3.44	26.43	50.60
2,000	-12.03	-4.10	24.51	51.58

APPENDIX B

R AND SAS CODES

```
##########################################
##EXAMPLE I
###Simulation of time series model with mixture distribution;
##Mixture of Gaussians N(0,9) and N(0,25).
 n <- 200 #sample size
#true values
a <- c(0.3, 0.7) #proportions
m <- c(1, 3.0) #m: mu
s <- c(0.6, 2) #s: sigma
gm2sim <- function(n, a, m, s) {
set.seed(123)
x <- rnorm(n) #standard normal
y <- 2 - (runif(n) < a[1]) #1 for true, 2 for false
x <- x * s[y] + m[y] #pick the right mu and sigma
return(list(x=x, y=y))
}
w<-gm2sim(n,a,m,s)$x
#########################
##AR(1) model with mixture noise;
ar_mix<-function(n)
{
set.seed(123)
y1<-w
for (t in 2:n)
y1[t]<-0.4*y1[t-1]+w[t]
return(y1)
}
ar1_mix<-ts(ar_mix(200))
### Testing stationarity of the data
```

```
PP.test(ar1_mix)
plot(ar1_mix,xlab="Time", ylab="Value")
decompose(ar1_mix)
##Decomposition of the model,  There are no or less than 2 periods,
so no seasonal model, i.e. d=0, only ARMA model.
####################################
###Finding the best model based on minimum AICc using ML method;
aics <- matrix(0,11,11, dimnames=list(p=0:10, q=0:10))
for(q in 1:10) aics[1, 1+q] <- arima(ar1_mix,
 c(0,0,q),method="ML")$aic
for(p in 1:10)
for(q in 0:10) aics[1+p, 1+q] <- arima(ar1_mix,
 c(p,0,q),method="ML")$aic
aics
indicator<-which(aics !=0)
round(aics - min(aics[indicator]))  ## See where the 0 is.
###AR(1) came out the best model;
##Fitting the best AR(1) model for the data;
library(forecast)
sim_fit<-forecast.Arima(arima(ar1_mix,order=c(0,0,4),method="ML"),
h=20,level=95)
 ##  for simulated fit based on AIC
fitted.sim<-sim_fit$fitted
residual.sim<-residuals(sim_fit)
lower.sim<-sim_fit$lower
upper.sim<-sim_fit$upper
forcast.sim<-sim_fit$mean
summary(sim_fit)   ##gives model fit statistics
####Plot the data
##########################################
plot(sim_fit,type="l",main="", ylab="value",xlab="Time"
,xlim=c(1,230))
lines(fitted.sim,type="l",col=2,lty=2)
abline(v=200, lty=3,col="blue")
```

```
legend(0,0.7,c("Original","Fitted MA(4)"),lty=c(1,2)
,col=c("black","red"),box.lwd = 0,box.col = "white"
,bg = "white")
###################
## Residual diagnostic
#############################################
###Introduction of breakpoints;
###Identification of breakpoints
##Create a matrix of time and simulated value;
ar1_simat<-matrix(0,200,2)
ar1_simat[1:200,1]<-c(1:200)
ar1_simat[1:200,2]<-ar_mix(200)
colnames(ar1_simat)<-c("time","value")
library(strucchange)
brk1 <- breakpoints(ar1_simat[1:200,2]~ar1_simat[1:200,1],h=0.2)
summary(brk1)
plot(brk1,main="")
lines(brk1)
#########Break point at observation 124;
ar1_mix1<-ts(ar1_mix[1:124],start=1)
ar1_mix2<-ts(ar1_mix[125:200],start=125)
##Identification of best models for each part;
#Part I
#Finding the best model based on minimum AICc using ML method;
aics <- matrix(0,11,11, dimnames=list(p=0:10, q=0:10))
for(q in 1:10) aics[1, 1+q] <- arima(ar1_mix1, c(0,0,q),
method="ML")$aic
for(p in 1:10)
for(q in 0:10) aics[1+p, 1+q] <- arima(ar1_mix1, c(p,0,q),
method="ML")$aic
aics
indicator<-which(aics !=0)
round(aics - min(aics[indicator]))  ## See where the 0 is.
#####################
```

```
## Fitting the best model MA(3)
sim_fit1<-forecast.Arima(arima(ar1_mix1,order=c(0,0,3),
method="ML")
,h=20,level=95)   ##  for simulated fit based on AIC
fitted.sim1<-sim_fit1$fitted
residual.sim1<-residuals(sim_fit1)
lower.sim1<-sim_fit1$lower
upper.sim1<-sim_fit1$upper
forcast.sim1<-sim_fit1$mean
summary(sim_fit1)   ##gives model fit statistics
####DIagnostic tests
##Auto correlation tests
plot(acf(residual.sim1),main="")
plot(acf(residual.sim2),main="")
Box.test(residual.sim1,type=c("Box-Pierce","Ljung-Box"))
##Ljung box autocorrelation tests
Box.test(residual.sim2,type=c("Box-Pierce","Ljung-Box"))
library(fBasics)
hist(residual.sim1,breaks=15)
hist(residual.sim2,breaks=15)

ksnormTest(residual.sim1)
shapiroTest(residual.sim1)
ksnormTest(residual.sim2)
shapiroTest(residual.sim2)
##homoscedasticity
####
###Part 2
aics <- matrix(0,11,11, dimnames=list(p=0:10, q=0:10))
for(q in 1:10) aics[1, 1+q] <- arima(ar1_mix2, c(0,0,q),
method="ML")$aic
for(p in 1:10)
for(q in 0:10) aics[1+p, 1+q] <- arima(ar1_mix2, c(p,0,q),
method="ML")$aic
```

```
aics
indicator<-which(aics !=0)
round(aics - min(aics[indicator]))  ## See where the 0 is.
##############################################
####################
## Fitting the best model ARMA(2,2)
sim_fit2<-forecast.Arima(arima(ar1_mix2,order=c(2,0,2),method="ML")
,h=20,level=95)   ##  for simulated fit based on AIC
fitted.sim2<-sim_fit2$fitted
residual.sim2<-residuals(sim_fit2)
lower.sim2<-sim_fit2$lower
upper.sim2<-sim_fit2$upper
forcast.sim2<-sim_fit2$mean
summary(sim_fit2)   ##gives model fit statistics
##Combined error and parameter estimatiion for mixture distribution
comb_error<-c(residual.sim1,residual.sim2)
hist(comb_error,breaks=15,main="")
ksnormTest(comb_error)
Box.test(comb_error,type="Box-Pierce")
Box.test(comb_error,type="Ljung-Box")
plot(acf(comb_error),main="")
library(mixtools)
mix_param<-normalmixEM(comb_error,k=2)
mix_param
plot(mix_param,which=2,main1="", main2="")
lines(density(comb_error),lty=2,lwd=2,main="")
legend(2,0.25,c("Part I density","Part II density","Mixed density")
,lty=c(2,2,2),col=c("red","green","black"))
# Empirical CDF comparison of breakpoint mixtures and true mixtures
### Adequacy of mixture model
##############################################
## See how many mixture components are good
#Evaluate various numbers of Gaussian components by
#data-set splitting (i.e., very crude cross-validation)
```

```
#Log likelihood function for a Gaussian mixture,
# potentially on new data
dnormalmix <- function(x,mixture,log=FALSE) {
  lambda <- mixture$lambda
  k <- length(lambda)
  # Calculate share of likelihood for all data for one component
  like.component <- function(x,component) {
    lambda[component]*dnorm(x,mean=mixture$mu[component],
                            sd=mixture$sigma[component])
  }
 #Create array with likelihood shares from all components
# over all data
  likes <- sapply(1:k,like.component,x=x)
  # Add up contributions from components
  d <- rowSums(likes)
  if (log) {
    d <- log(d)
  }
  return(d)
}
#Log likelihood function for a Gaussian mixture, potentially
# on new data
loglike.normalmix <- function(x,mixture) {
  loglike <- dnormalmix(x,mixture,log=TRUE)
  return(sum(loglike))
}
############################
n <- length(comb_error)
data.points <- 1:n
data.points <- sample(data.points) # Permute randomly
train <- data.points[1:floor(n/2)] # First random half is training
test <- data.points[-(1:floor(n/2))] # 2nd random half is testing
candidate.component.numbers <- 2:10
loglikes <- vector(length=1+length(candidate.component.numbers))
```

```
# k=1 needs special handling
mu<-mean(comb_error[train]) # MLE of mean
sigma <- sd(comb_error[train])*sqrt((n-1)/n) # MLE of sd
loglikes[1] <- sum(dnorm(comb_error[test],mu,sigma,log=TRUE))
for (k in candidate.component.numbers) {
  mixture <- normalmixEM(comb_error[train],k=k,maxit=400,epsilon=1e-2)
  loglikes[k] <- loglike.normalmix(comb_error[test],mixture=mixture)
}
### Figure of identification of mixture components
plot(x=1:10, y=loglikes,xlab="Number of mixture components",
     ylab="Log-likelihood on testing data")
##############################################
 # Comparison of ECDF of mixture of Gaussian and combined Residuals
# Function to calculate the cumulative distribution function of a
#Gaussian mixture model
# Presumes the mixture object has the structure used by mixtools
# Doesn't implement some of the usual options for CDF functions in R,
# like returning the log probability, or the upper tail probability
pnormmix <- function(x,mixture) {
  lambda <- mixture$lambda
  k <- length(lambda)
  pnorm.from.mix <- function(x,component) {
    lambda[component]*pnorm(x,mean=mixture$mu[component],
                            sd=mixture$sigma[component])
  }
  pnorms <- sapply(1:k,pnorm.from.mix,x=x)
  return(rowSums(pnorms))
}
######## Comparison of Mixture cdf(Theoretical) and empirical cdf
distinct.val <- sort(unique(comb_error))
tcdfs <- pnormmix(distinct.val,mixture=mix_param)
ecdfs <- ecdf(comb_error)(distinct.val)
plot(tcdfs,ecdfs,xlab="Theoretical CDF",ylab="Empirical CDF"
,xlim=c(0,1),ylim=c(0,1))
```

```
abline(0,1)
########################
plot(mix_param,which=2,main1="", main2="")
lines(density(comb_error),lty=2,lwd=2,main="")
curve(dnormalmix(x,mix_param),add=TRUE,lty=4,col="purple")
legend(2,0.25,c("N(-1.00,0.58) density","N(0.48,3.22) density"
,"Mixture density",
 "Combined error density"),lty=c(1,1,4,2),col=c("red","green"
,"purple","black")
,box.lwd = 0,box.col = "white",bg = "white")
##Kolmogorov-Smirnov test for two sample density test
ks.test(comb_error,w)
ks.test(residuals.sim,w)
########################################
### Forecasting
forecast_matrix<-matrix(0,220,9)
forecast_matrix[1:200,1]<-c(1:220)
forecast_matrix[1:200,2]<-ar1_mix
forecast_matrix[1:124,3]<-fitted.sim1    ##Mixture fitted
forecast_matrix[125:200,3]<-fitted.sim2
### Forecasting
forecast_matrix[201,3]<-0.32*(3.7639+0.3484*residual.sim2[76]+
0.1411*residual.sim2[75]+0.2350*residual.sim2[74])+0.68*(4.3021
-1.3395*4.3021+4.3021*0.9434+1.3395*ar1_mix[200]-0.9434
*ar1_mix[199]-1.1308*residual.sim2[76]+0.8701*residual.sim2[75])
forecast_matrix[202,3]<-0.32*(3.7639+0.1411*residual.sim2[76]
+0.2350*residual.sim2[75])+ 0.68*(4.3021-1.3395*4.3021+4.3021*0.9434
+1.3395*forecast_matrix[201,3]-0.9434*ar1_mix[200]+0.8701
*residual.sim2[76])forecast_matrix[203,3]<-0.32*(3.7639+0.2350
*residual.sim2[76])+0.68*(4.3021-1.3395*4.3021+4.3021*0.9434
+1.3395*forecast_matrix[202,3]-0.9434*forecast_matrix[201,3])
for (i in 204:220){
forecast_matrix[i,3]<-0.32*3.7639+0.68*(4.3021-1.3395*4.3021
+4.3021*0.9434+1.3395*forecast_matrix[i-1,3]-0.9434
```

```
*forecast_matrix[i-2,3])}
forecast_matrix[201:220,4]<-sqrt(0.32^2*0.03124*3.083+0.68^2*3.083)
## 0.03124 is the variance  of theta1+var_theta2+var_theta3
for(i in 201:220)
{
forecast_matrix[i,5]<-forecast_matrix[i,3]-1.96*forecast_matrix[i,4]
forecast_matrix[i,6]<-forecast_matrix[i,3]+1.96*forecast_matrix[i,4]
}
forecast_matrix[1:200,7]<-fitted.sim  ##Regular time series fit
forecast_matrix[201:220,7]<-forcast.sim
forecast_matrix[201:220,8]<-lower.sim
forecast_matrix[201:220,9]<-upper.sim
colnames(forecast_matrix)<-c("time","actual","forecast_mix","se_mix",
"mix_ll","mix_ul","ts_fit","ts_ll","ts_ul")
dim(forecast_matrix)
############################
is.na(forecast_matrix)<-(forecast_matrix==0) ##convert 0's to NA.
####Plot the graph
plot(ar1_mix,type="l",main="", ylab="value",xlab="Time"
,xlim=c(1,230))
lines(forecast_matrix[,3],col="red",lty=2)
lines(forecast_matrix[,7],col="green",lty=4)
lines(forecast_matrix[,5],col="brown",lty=3)
lines(forecast_matrix[,6],col="brown",lty=3)
lines(forecast_matrix[,8],col="blue",lty=5)
lines(forecast_matrix[,9],col="blue",lty=5)
abline(v=200, lty=3)
legend(8,9.5,c("Simulated data","Mixture", "Time series","Mixture CI"
,"Time series CS"),lty=c(1,2,7,3,5),col=c("black","red","green"
,"brown","blue"),box.lwd = 0,box.col = "white",bg = "white")
```

```
#####################################
## Example 2, (SAS Codes)
###Mixture of AR(1) and MA(2) models
#############################################
/*AR(1) with model phi=0.7 and mean=0;
*/
libname diss "G:\Research\Rajan AISC paper\data";
run;
**
data diss.ar1;
title Simulated AR(1) Series;
u1 = 0;
do i = -50 to 100;
time=i;
u = 0.7*u1 + rannor( 32565 ) ;
if i > 0 then output;
u1=u;
end;
run;
proc arima data=diss.ar1;
identify var=u minic  nlag=15;
run;
estimate p=1  ;
 forecast lead=25 id=time out=p1forecast printall;
run;
**Simulating MA (2) with parameters theta1=0.5 and theta2=0.4,
mean=3;
data diss.MA2;
title 'Simulated MA(2) model';
a1=0;
a2=0;
do i=-50 to 100;
time=i+100;
e=rannor(1234);
```

```
u=3+e-0.5*a1-0.4*a2;
if i>0 then output; **First 50 observations are removed;
a2=a1;
a1=e;
end;
run;
proc arima data=diss.ma2;
identify var=u minic nlag=15;
run;
estimate q=2 ;
forecast lead=25 id=time out=p2forecast printall;
run;
quit;
** Combined data;
data diss.combined;
set diss.ar1 diss.ma2;
keep time u;
run;
***Plot the data;
AXIS1 LABEL=(ANGLE=90 'value') ;
AXIS2 LABEL=('Time') order=(0 to 250 by 20)minor=none ;
SYMBOL1 V=DOT C=black I=JOIN H=0.1 W=0.5;
 PROC GPLOT DATA=diss.combined;
 where time<=200;
title 'simulated data';
PLOT u*time/ VAXIS=AXIS1 HAXIS=AXIS2 href=100 ;
RUN; QUIT;
************************************;
**Fit the best model for combined data using regular time series
 modeling;
* we fit the model for first 200 observations and remaining 30
observations are forecasted;
****************************************************;
proc arima data=diss.combined;
```

```
identify var=u minic scan esacf;
run;
estimate p=3;
run;
forecast lead=25 id=time out=reg_results printall;
run;
quit;
data diss.all_withforecast_ts;
merge  reg_results diss.combined;
by time;
run;
************************************************;
***Plot the data with original values;
*********************************************;
 Legend1 label=(height=1 position=Top justify=center '')
value=("Actual" "Forecast" "95% CI" "" )
across=1 down=4
position = (bottom center inside)
mode=protect;
AXIS1 LABEL=( "Time") order=(0 to 250 by 20)minor=none;
AXIS2 LABEL=(ANGLE=90 "Value") ;
 SYMBOL1 V=DOT I=JOIN H=0.1 W=0.5;
 symbol2 V=DOT C=blue I=JOIN H=0.1 W=0.5 line=3;
SYMBOL3 V=none I=j C=red  line=2;
 symbol4 V=none C=red I=j  line=2;
**
 Proc gplot data=all_withforecast_ts;
 title 'Simulated data';
 plot (u forecast l95 u95)*time/overlay href=(100,200)  haxis=axis1
vaxis=axis2 legend=legend1;
run;
quit;
** Residual plot;
proc gplot data=all_withforecast_ts;
```

```
title "Residual Plot";
plot residual*time;
run;
quit;
**************************************
**Chow test to see the significant difference at the break points;
proc autoreg data=part;
model u=time/chow=(100);
run;
 /* For forecasting , we need to get STD from part 2 by fitting
 both models;*/
    data part1;
   set p1forecast;
   if time>100;
   rename std=std1;
   time1=time+100;
   drop time;
   rename time1=time;
   drop forecast u L95 U95 residual;
   run;
    data part2;
   set p2forecast;
   if time>200;
   rename std=std2;
   drop forecast u L95 U95 residual;
   run;
   data new;
   merge part1 part2;
   /* the variance of forecast values depends on the variance
 of last part times weighted sum of parameter estimests phi_i  so*/
  std=(0.5)*(std1*1.200599/0.899629)+(0.5)*std2; **1.200=std of
 part 2, and 0.899=std of part 1;
   drop std1 std2;
   run;
```

```
   **** combined forecast for each segment data;
   data part2forecast;
   set p2forecast;
 if time<=200;
   run;
   data part1forecast;
   set p1forecast;
 if time<=100;
   run;
   **Concatenating all results from two parts;
   data diss.all_part_forecast;
   set part1forecast part2forecast new;
     run;
*****************************************
***Forecasting  ********************************
*********************************************;
Proc IML;
use diss.all_part_forecast;
show names;
show datasets;
show contents;
read all;
all_pred=time||u||forecast||std||L95||U95||residual;
print all_pred;
/*nr= nrow(forecast);
nco=ncol(forecast);
las=forecast[782];*/
forecast0=j(225,7,0);
/* j(a,b,c) creates the matrix of a rows,
b cols with c values, 52 forecast weeks*/
print forecast0;
do i=1 to 200;     *Replacing 1-783 with previous predictions;
forecast0[i,1]=forecast[i];
forecast0[i,2]=L95[i];
```

```
forecast0[i,3]=U95[i];
forecast0[i,4]=u[i];
forecast0[i,5]=time[i];
forecast0[i,6]=std[i];
forecast0[i,7]=residual[i];
end;*/
print forecast;
/* estimates from two parts of weekly data*/
mu1=0.53140;
mu2=2.99696       ;
phi=0.67236       ;
theta1=0.51074       ;
theta2=0.30737       ;
p1=0.5;  /*Proportion of first part*/
p2=0.5;  /* Proportion of second part*/
/**Forecasting based on weighted estimates*/
forecast0[201,1]=p1*(mu1+phi*(u[200]-mu1))+p2*(mu2
-(theta1*residual[200])-(theta2*residual[199]));
forecast0[202,1]=p1*(mu1+phi*(forecast0[201,1]-mu1))
+p2*(mu2-(theta2*residual[200]));

do i=203 to 225;
forecast0[i,1]=p1*(mu1+phi*(forecast0[i-1,1]-mu1))+p2*(mu2);
end;
do i=201 to 225;
forecast0[i,2]=forecast0[i,1]-1.96*std[i];
forecast0[i,3]=forecast0[i,1]+1.96*std[i];
forecast0[i,5]=i;
forecast0[i,6]=std[i];
end;
print forecast0;
show contents;
create mydata1 from forecast0;
Append from forecast0 ;* VAR{forecast L95 U95 avg week};
```

```
quit;
data diss.mixtureforecast;
set mydata1;
rename col1=mixfrcst col2=mix_L95 col3=mix_U95 col4=u col5=time
col6=mix_std col7=mix_residual;
run;
*****************************************;
** Combining all results from ARMA and Mixture model into one  data;
data armamodel;
set diss.all_withforecast_ts;
run;

data diss.all_forecast_mix_simulation;
merge armamodel diss.mixtureforecast;
by time;
if time=0 then u=.;
if u=0 then u=.;
if mix_residual=0 then mix_residual=.;
run;
 **Testing the autocorrelatio of combined residuals of mixture model;
 proc autoreg data=diss.all_forecast_mix_simulation;
   model mix_residual = time / dw=6 dwprob;
run;
***Variace estimation of combined residual of mixture model;
proc means data=diss.all_forecast_mix_simulation;
var mix_residual;
run;
#####################################
##Example 3 (Otolith Data), Block Bootstrap and EM algorithms only
#####################################
###Identification of break points ;
tasiat<-read.table("E:/Research/Rajan AISC paper/data/tasiat.txt"
,header=TRUE,blank.lines.skip = FALSE);
library(strucchange)
```

```
### We want to see the empirical fluctation;
efp(avg_o18 ~ avg_temp + avg_precip + avg_rain, data =tasiat)
fluct1<-efp(avg_o18 ~ YR_FORM, data =tasiat)
plot(fluct1,main="Fluctuation for lake Tasiat")
## Identification of breaking points;
brk1 <- breakpoints(avg_o18~YR_FORM, data =tasiat,h=0.2)
summary(brk1)
plot(brk1,main="Breaking points for lake Tasiat by taking h=0.2")
lines(brk1)
##################################
# AR(1) fits the part 2 data, but there are only 12 observations,
# so we use Block bootstrap to improve the parameter
# estimation
#Parameter estimation using Bootstrap;
###################################;
### First we estimate the optimal block size within each partition;
##Second Part of the data;
tasiat2<-ts(tasiat[23:34,3],frequency=1, start=1989)#Only Average o18;
library(np)
b.star(tasiat2,round=TRUE)
##Gives optimal block length=2, but it's less than 1/3 of the size,
#so we take block length=6 for better estimate;
#First we create a function that output the parameters for AR(1) model;
boot.func<-function(y){
{
store<-rep(0,61)   ## we store output in this vector, 6:12 are related
 to residuals of fitted model;
set.seed(1234)
aics <- arima(y, c(1,0,0),method="ML")
## Forecast standard errors;
library(forecast)
sim1.pred<-forecast.Arima(aics,level=95)
est<-as.vector(sim1.pred$mean)    ##10 forecasts
actual<-as.vector(tasiat2)    ## Actual values
```

```
fit<-as.vector(sim1.pred$fitted)
resi<-as.vector(sim1.pred$residuals)
std1<-(sim1.pred$lower-sim1.pred$mean)/(-1.96)
 ##Lower limit-est/-1.96 gives st. error for 95 % CI.
###### Parameter estimates based on Minimum AIC and ML method;
##Storing all values in a vector;
store[1]<-aics$aic  ## It stores Minimum AIC value;
store[2]<-aics$sigma2  ## It stores samplign variance;
store[3]<-aics$loglik  ## it stores log likelihood ;
store[4:5]<-aics$coef
#it stores coefficients in the order of AR and intercept;
store[6:17]<-actual  ## Actual values
store[18:29]<-fit  ##it stores the residuals
 based on the model with min. variance.;
store[30:39]<-est    ## forecast values
store[40:49]<-std1    ##Standard errors
store[50:61]<-resi     ### Residuals
}
store
}
######################;
library(boot)
#### block bootstrappping estimating AR1 parameters only;
block.ar1boot <-tsboot(tasiat2, boot.func, R=5000, l=6, sim = "geom")
 ##mean block length l=6;
block.ar1boot
block.ar1result<-block.ar1boot$t
###################
boot.theta1<-block.ar1result[,4]
boot.mean<-block.ar1result[,5]
density.theta1<-density(boot.theta1)
density.mean<-density(boot.mean)
##Parameter estimateion using EM on combined errors;
 error<-as.vector(c(error1,error2))
```

```
 hist(error)
 hist(error1)
 hist(error2)
 ####EM function;
 W = error
s = c(0.65,mean(error1),mean(error2), var(error1),0.012522239  )
  ##Initial value of p=22/34.
 em = function(W,s) {
 Ep = s[1]*dnorm(W, s[2], sqrt(s[4]))/(s[1]*dnorm(W, s[2], sqrt(s[4]))
 +(1-s[1])*dnorm(W, s[3], sqrt(s[5])))
 s[1] = mean(Ep)
s[2] = sum(Ep*W) / sum(Ep)
 s[3] = sum((1-Ep)*W) / sum(1-Ep)
s[4] = sum(Ep*(W-s[2])^2) / sum(Ep)
 s[5] = sum((1-Ep)*(W-s[3])^2) / sum(1-Ep)
s
 }
iter = function(W, s) {
s1 = em(W,s)
 for (i in 1:5) {
if (abs(s[i]-s1[i]) > 0.0001) {
 s=s1
 iter(W,s)
 }
 else s1
 }
 s1
}
 iter(W,s)
 options(expressions=100000)  ## Improves memory to run the program
 result<-iter(W,s)
 result
```

VITA

Rajan Lamichhane
Department of Mathematics and Statistics
Old Dominion University
Norfolk, VA 23529

Education

Tribhuvan University, Kathmandu, Nepal, M.S., Statistics, 1996-1998.
Southern Methodist University, Dallas, Texas, M.S., Statistics, 2005-2007.

Typeset using LaTeX.