

CARLETON UNIVERSITY

SCHOOL OF
MATHEMATICS AND STATISTICS

HONOURS PROJECT



TITLE: A Comparison Between Multivariate and Univariate Models of Time Series

AUTHOR: Scott McCalla

SUPERVISOR: Song Cai, Shirley Mills (second reader)

DATE: Saturday, December 28, 2019

Contents

1 Introduction	1
2 The VAR(p) Model	2
2.1 Introduction to VAR(p) Models	2
2.2 Features and Conditions of the VAR(p) Model	3
2.3 Estimation Under the VAR(p) Model	5
2.4 Model Checking for VAR(p) Model	6
2.5 Forecasting With the VAR(p) Model	7
3 The VARMA(p,q) Model	9
3.1 Introduction to the VARMA(p,q) Model	9
3.2 The VMA(q) model	9
3.3 The VARMA(p,q) Model	10
3.4 Estimation and Forecasting with the VARMA(p,q) Model	11
4 Bivariate VAR(1) Model Example	12
4.1 VAR(1) Model Procedure and Prediction	12
4.2 The True Model	13
4.3 The Procedure in R	14
4.4 10 Sample Procedures	17
5 Trivariate VARMA(1,1) Model Example	18
5.1 VARMA(1,1) Model Procedure and Prediction	18
5.2 The True Model	19
5.3 The Procedure in R	20
5.4 20 Sample Procedures	22

6 Trivariate ARMA(2,1) Model Example	23
6.1 The ARMA(2,1) Procedure and Prediction	23
6.2 The True Models	24
6.3 The Procedure in R	25
6.4 20 Sample Procedures	27
7 Bivariate AR(1) Model Example	29
7.1 The AR(1) Procedure and Prediction	29
7.2 The True Models	30
7.3 The Procedure in R	30
7.4 10 Sample Procedures	31
8 Conclusion and Further Study	33
Appendix	34

1 Introduction

In this project, we will consider multivariate time series. We strive to compare the use of multivariate models, particularly the VARMA(p,q) model, to several univariate models. We will do this using R to sample data from various different processes, then fitting both a multivariate model and several univariate models to the same sample. We will then compare the mean squared error of the forecasted values of each fitted model.

We will first explain how the VAR(p) model is derived and how it is used in forecasting. The goal of this is to understand the process R will use to estimate and forecast the values we are interested in. It will also allow us to construct a “true” model that we will use R to generate a sample from.

Next, we will briefly explain a few specifications we will need to use the VARMA(p,q) model. This will be needed to construct a true model to generate from, as well as understanding how R will forecast with the given fitted model.

Finally, we will give 4 different examples where we compare a multivariate model to several univariate models. Our goal is to see whether a multivariate model is truly a better option than several univariate models in all cases.

2 The VAR(p) Model

2.1 Introduction to VAR(p) Models

To start off our fitting of multivariate time series using the VARMA(p,q) model, we begin with the special case in which the parameter $q = 0$. This model is formally known as a VAR(p) process. We will denote the k -dimensional time series under consideration as

$$X_t = \{X_{1t}, X_{2t}, \dots, X_{kt}\},$$

where t is time (integer valued), and $1, \dots, k$ denote the dimensions.

Most of the material covered in this section may also be found in *Multivariate Time Series Analysis*, (R. S. Tsay), and some of it will not be used in the examples portion of this project. This information is still useful when applying VAR(p) models to real data sets with no known distribution, but is unnecessary when the “true” distribution is known. In the case of our examples, we will be constructing the true distributions to our specified standards, so we will know the “true” distribution.

With this, we will define our VAR(p) equation as follows:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + Z_t,$$

where c is a k -dimensional vector of constants, ϕ_i 's are the k by k coefficient matrices of constants, for all $1 \leq i \leq p$, and $Z_t = \{Z_{1t}, Z_{2t}, \dots, Z_{kt}\}$ is a multivariate white noise process with mean 0 and constant covariance matrix Σ_z . One can see that this is very similar to the univariate AR(p) equation. As with the univariate case, we can rewrite the VAR(p) equation using the backward shift operator B , where $BX_t = X_{t-1}$, and $B^i X_t = X_{t-i}$. We will also assume the process $\{X_t\}$ has 0 mean, since we could easily centre the process around its mean with a few basic operations. With that, we end up with:

$$\phi(B)X_t = Z_t,$$

$$\text{where } \phi(B) = I_k - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p,$$

where I_k is the k by k identity matrix. This $\phi(u)$ is known as the characteristic polynomial.

For notation, we refer to the individual elements of each ϕ_i as $\phi_{i, mn}$ to represent the m 'th row and n 'th column element of the i 'th coefficient matrix. It is

As an example, here is the expanded form of a bivariate VAR(1) process:

$$X_{1,t} = c_1 + \phi_{1,11}X_{1,t-1} + \phi_{1,12}X_{2,t-1} + Z_{1,t},$$

$$X_{2,t} = c_2 + \phi_{1,21}X_{1,t-1} + \phi_{1,22}X_{2,t-1} + Z_{2,t},$$

where as the original matrix form would be written as:

$$X_t = c + \phi_1 X_{t-1} + Z_t$$

Note that $\phi_{1,12}$ is the relationship between $X_{1,t}$ and $X_{2,t-1}$, and $\phi_{1,21}$ for $X_{2,t}$ and $X_{1,t-1}$, and is the cause of the dynamic correlation between $X_{1,t}$ and $X_{2,t-1}$. This means that if $\phi_{1,12} = \phi_{1,21} = 0$, then X_{1t} and X_{2t} would not be influenced by past X_{2t} and X_{1t} , respectively. If this were the case, then we would treat both time series as univariate AR(1) models. This can extend to general AR(p) with any number of time series in X_t , with all ϕ_i 's being diagonal meaning each X_{it} can be treated as a univariate AR(p) model. An interesting case also arises if, without loss of generality, $\phi_{1,21} = 0$, but $\phi_{1,12}$ is not zero. Then, we would treat X_{1t} as a univariate AR(1) model, but may be able to reduce variance in forecasting X_{2t} by using a bivariate model with X_{1t} included, which could prove useful if X_{1t} could be controlled. This concept can be extended to general VAR(1) models with $k \geq 1$, but requiring ϕ_1 to be lower triangular to variance in X_{kt} by introducing $k - 1$ control variables into the model.

2.2 Features and Conditions of the VAR(p) Model

Similar to the univariate model, a process satisfying the VAR(p) equation is only useful to us if it is stationary. Fortunately, it is rather simple to determine if a process X_t that satisfies the VAR(p) equation is stationary, and is also a biconditional. This sufficient condition is that all solutions to the following equation:

$$\det[\phi(B)] = 0$$

must have modulus greater than 1. This result is proved by Tsay.

As a result of the definition of the VAR(p) equation, any process X_t satisfying it will also be invertible.

Now that we can determine whether X_t is stationary or not, we will assume X_t is stationary when determining the moments of a VAR(p) process. We will start with a VAR(1) process, since we will soon learn that any VAR(p) process can be condensed into a VAR(1) process. Since we already assumed that $\mu = E(X_t) = 0$ for simplicity, and X_t is stationary, we now know that the covariance matrix of X_t must be constant. Let us denote this covariance as Γ_0 , and we will have that:

$$\Gamma_0 = \phi_1 \Gamma_0 \phi_1' + \Sigma_Z,$$

Where A' is the transpose of matrix A . This, through many theorems and operations detailed by Tsay, becomes solvable for Γ_0 using the following equation:

$$[I_{k^2} - \phi_1 \otimes \phi_1] \text{vec}(\Gamma_0) = \text{vec}(\Sigma_Z),$$

where \otimes is the Kronecker product, and $\text{vec}(A)$ is the column-stacked vector form of matrix A . Then, for any lag $\tau > 0$, we may find cross covariance matrices using this equation:

$$\Gamma_\tau = \phi_1 \Gamma_{\tau-1}$$

As mentioned before, any VAR(p) process can be expressed as a VAR(1) process using the following equation:

$$Y_t = \Phi Y_{t-1} + b_t,$$

Where $Y_t' = \{X_t', X_{t-1}', \dots, X_{t-p+1}'\}$ is a pk dimensional time series, $b_t' = \{Z_t', 0\}$, where 0 is a $k(p-1)$ dimensional vector of zeros, and Φ is a kp by kp matrix defined as:

$$\Phi = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_p \\ I_k & 0 & \dots & 0 \\ 0 & I_k & \dots & 0 \\ \dots & & & 0 \\ 0 & \dots & I_k & 0 \end{bmatrix}$$

Therefore, we can find Γ_0 for any VAR(p) process by using the formulae for a VAR(1) process.

As an extra note, one can express any single time series X_{it} that is a VAR(p) process as an ARMA($kp, (k-i)p$) univariate process, which follows from the identity:

$$\det(\phi(B))X_t = \text{adj}(\phi(B))Z_t$$

where $\text{adj}(A)$ is the adjoint of matrix A . Using this, we can expand each X_{it} to model a univariate ARMA $(kp, (k-1)p)$ model.

2.3 Estimation Under the VAR(p) Model

Through this section we will assume that Z_t follows a normal distribution with mean 0 and constant covariance Σ_Z . Now, we suppose that a sample $\{X_1, X_2, \dots, X_T\}$ is observed from X_t , where T is an integer.

Recall that our VAR(p) model is:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + Z_t,$$

Here, the parameters we are interested in are c, ϕ_i , for all $1 \leq i \leq p$, and Σ_Z . It is important to note that we have $T - p$ data points to work with, and that we are truly estimating $k(k(p + 1) + 1)$ parameters, with k contained in c , k^2 contained in all $p \phi_i$'s, and k^2 in Σ_Z .

Using the Least Squares method, we can rewrite the VAR(p) model as follows:

$$X = Y\beta + Z,$$

where X is a $(T - p)$ by k matrix whose i 'th row is X_{p+i}' , Y is a $(T - p)$ by $(kp + 1)$ matrix whose i 'th row is $\{1, X_{p+i-1}', X_{p+i-2}', \dots, X_i'\}$, Z is a $(T - p)$ by k matrix whose i 'th row is Z_{p+i}' , and $\beta = \{c, \phi_1, \phi_2, \dots, \phi_p\}$ is a k by $(kp + 1)$ matrix containing model parameters. This may sound obscure, but we can deduce that the j 'th column of β contains all the parameters for X_{jt} . After many operations detailed by Tsay, we get the LS estimator of β , $\hat{\beta}$, as follows:

$$\hat{\beta} = (Y'Y)^{-1}(Y'X),$$

which contains our LS estimates of c , and all ϕ_i 's. This is an unbiased estimator of β .

We also learn that $\text{vec}(\hat{\beta})$ is distributed as a multivariate normal with mean $\text{vec}(\beta)$ and covariance $\Sigma_Z \otimes (Y'Y)^{-1}$.

Using maximum likelihood method, as detailed by Tsay, we find our estimator of Σ_Z to be the following:

$$\hat{\Sigma}_Z = (T-p)^{-1} \hat{Z}' \hat{Z},$$

where $\widehat{Z} = X - Y\widehat{\beta}$. It is shown by Tsay that $\widehat{\Sigma}_Z$ is asymptotically unbiased and consistent estimator of Σ_Z , and is independent of $\widehat{\beta}$. We also learn that $\widehat{\Sigma}_Z$ is distributed as a $W_{k, T - (k + 1)p - 1}(\Sigma_Z)$, a Wishart distribution.

Finally, to test for the order of the VAR(p) process, p, we have a shortcut available for easy use. This shortcut comes in the form of Bayesian information criterion, or BIC for short. One could also use Akaike information criteria, but the BIC is a stronger test. The BIC test statistic is defined as follows for VAR order u:

$$\text{BIC}(u) = \ln(\det(\widehat{\Sigma}_{Z,u})) + \ln(T)uk^2/T,$$

where $\widehat{\Sigma}_{Z,u}$ is the estimated sample covariance when using a VAR(u) model. The general idea to estimate the order is to find the BIC test statistic for the sample using VAR(1), then repeat until some maximum order VAR(P), where P is reasonably high, usually around 10 to 15 for lower k. Then, the model VAR(u) that produces the lowest BIC test statistic will be our chosen model of order u.

2.4 Model Checking for VAR(p) Model

We will not go into too much detail about model checking in this study, but a few things will be introduced to satisfy a few requirements before we get into forecasting. Model checking is explored in great detail by Tsay, but here we will skip the intermediate steps and get to the final result.

First, we start off by defining a few new statistics of interest. Let \widehat{C}_τ be the cross covariance residual series at lag τ , and \widehat{R}_τ be the cross correlation residual series at lag τ . From Tsay, we get that

$$\begin{aligned}\widehat{C}_0 &= \Sigma_Z, \\ \widehat{C}_\tau &= (T - p)^{-1} \widehat{Z}' B \widehat{Z}, \\ \widehat{R}_\tau &= \widehat{D}^{-1} \widehat{C}_\tau \widehat{D}^{-1},\end{aligned}$$

where B is a (T - p) by (T - p) back shift matrix and is defined as:

$$B = \begin{bmatrix} 0 & 0'_{T-p-1} \\ I_{T-p-1} & 0_{T-p-1} \end{bmatrix}$$

and \widehat{D} is a diagonal matrix whose i,i 'th entries are the square root of the i,i 'th entries of \widehat{C}_0 .

Through many theorems explored by Tsay, we get our hypothesis of interest:

$$H_0: R_1 = R_2 = \dots = R_u = 0, \text{ versus}$$

$$H_a: R_i \text{ is not zero, for some } 1 \leq i \leq u,$$

where R_τ is the “true” cross correlation matrix at lag τ , and u is some positive integer.

Through some simplification shown in fsgd, we get our test statistic, $Q_k(u)$, which is specifically the Portmanteau statistic, in this equation:

$$Q_k(u) = T^2 \sum_{\tau=1}^u (T - \tau) \text{tr}(\widehat{C}_\tau' \widehat{C}_0^{-1} \widehat{C}_\tau \widehat{C}_0^{-1})$$

Through a theorem proved by Tsay, we get that under certain conditions, $Q_k(u)$ asymptotically follows a chi-Square distribution with $(u-p)k^2$ degrees of freedom. These conditions are that i) X_t follows a stationary VAR(p) model, ii) Z_t follows a white noise process with 0 mean and covariance Σ_z , iii) the fourth moment of X_t is finite, and iv) β is consistently estimated. When testing a model, you set the p-values $Q_k(u)$ to 1 for any $u \leq p$, since $Q_k(u)$ requires $u > p$ for positive degrees of freedom. We want the p-values of $Q_k(u)$, for $u > p$, to be higher than our specified margin of error, which would mean our model is an acceptable fit.

There are ways to test whether a parameter is zero or not, but we will not consider this in our study. For more information, Tsay has many details pertaining to the testing of zero parameters in the VAR(p) model. We will also not consider linear constraints in our examples, so we will skip them.

2.5 Forecasting With the VAR(p) Model

For this section, we must define a few new values and properties.

First, let h be the forecast origin, τ be the number of forward steps ahead we are forecasting, and F_h be the information available up to and including time h . Then, we define the τ -step forward prediction of $X_{h+\tau}$ as $X_h(\tau)$. Through Tsay, we are told that the minimum mean squared error forecast of $X_{h+\tau}$ is:

$$X_h(\tau) = E(X_{h+\tau} | F_h)$$

For our VAR(p) model, we can expand this for $X_h(1)$ and a general $X_h(\tau)$, for $\tau > 1$:

$$X_h(1) = c + \phi_1 X_h + \phi_2 X_{h-1} + \dots + \phi_p X_{h+1-p}, \text{ and}$$

$$X_h(\tau) = c + \sum_{i=1}^p \phi_i X_h(\tau - i), \text{ for } \tau > 1$$

where for $j < 1$, $X_h(j) = X_{h+j}$. It is shown in dgsdfgsd that for X_t with mean zero, $X_h(\tau)$ converges to zero as τ gets large.

Next, we define the forecast error $e_h(\tau)$ as:

$$e_h(\tau) = X_{h+\tau} - X_h(\tau)$$

Clearly, the expectation of $e_h(\tau)$ is zero, and it is shown by Tsay that $\text{Cov}(e_h(\tau))$ converges to $\text{Cov}(X_t)$.

Now that we have these defined, we can move on to the estimates:

$$\hat{X}_h(\tau) = \hat{c} + \sum_{i=1}^p \phi_i \hat{X}_h(\tau - i),$$

$$\hat{e}_h(\tau) = X_{h+\tau} - \hat{X}_h(\tau)$$

Through Tsay, we find that the process to find $\text{Cov}(\hat{X}_h(\tau))$ and $\text{Cov}(\hat{e}_h(\tau))$ is very technical, and the result for $\tau = 1$ is:

$$\text{Cov}(\hat{e}_h(1)) = (T - p + kp + 1)(T - p)^{-1} \hat{\Sigma}_Z,$$

We will not be requiring this in our examples, so we will stop here for our VAR(p) model.

3 The VARMA(p,q) Model

3.1 Introduction to the VARMA(p,q) Model

For the most part, the VARMA(p,q) model is very similar to the VAR(p) model. Since we went into great detail for the VAR(p) model, we will go through the VARMA(p,q) model much quicker so we can get to the comparison examples.

We will start by introducing the VMA(q) model, where we will quickly go over stationarity conditions and forecasting. We will then move onto the VARMA(p,q) model, and cover much of the same.

Once again, most of the information provided in this section is also described by Tsay. However, this time we will only cover the material required for our examples.

3.2 The VMA(q) model

We will start off by introducing the VMA(q) model. If X_t follows a VMA(q) model, then X_t can be represented as:

$$X_t = Z_t - \theta_1 Z_{t-1} - \theta_2 Z_{t-2} - \dots - \theta_q Z_{t-q},$$

where Z_t is a white noise process with mean 0 and covariance Σ_Z positive definite, and θ_i 's are all k by k matrices of constants. Here, we are also assuming that $E(X_t) = 0$, for simplicity. Using the backward shift operator, as seen in the VAR(p) process, we may also write X_t as:

$$X_t = \theta(B)Z_t,$$

where $\theta(B) = I_k - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$ is the characteristic polynomial of X_t .

Quickly moving through properties of VMA(q), we have that X_t is stationary when Z_t is a white noise process with mean 0 and covariance Σ_Z positive definite. We also have that X_t is invertible if and only if all of the solutions to $\det[\theta(B)] = 0$ have modulus strictly greater than 1.

Unfortunately, in order to estimate all of the θ_i 's and Σ_Z , it is required to use an iterative method. Suppose the observed sample is $\{x_1, x_2, \dots, x_T\}$, of length T . We must first estimate z_0 through z_T , and then estimate the θ_i 's. However, estimating the z_i 's

requires the use of θ_i 's, so we must not have a closed form solution. This is also the case for a VARMA(p,q) process, which is why we will be using programming software to fit the VARMA(p,q) model to our simulated data.

Finally, the forecasting of VMA(q) processes from forecast origin h is as follows:

$$X_h(\tau) = - \sum_{i=h}^q \theta_i Z_{t+h-i},$$

where $X_h(\tau)$ is our τ -step forward forecast of X_h . For the estimator $\hat{X}_h(\tau)$, we replace the θ_i 's and the Z_i 's by their estimated values. Note that for $\tau > q$, we have that $\hat{X}_h(\tau) = 0$. The forecasting error is then defined as:

$$e_h(\tau) = X_h(\tau) - X_{h+\tau}, \text{ with estimator}$$

$$\hat{e}_h(\tau) = \hat{X}_h(\tau) - X_{h+\tau}.$$

3.3 The VARMA(p,q) Model

For the VARMA(p,q) model, we will go into as much detail as we did for the VMA(q) model. To start off, if X_t follows a VARMA(p,q) process, then X_t can be written as:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + Z_t - \theta_1 Z_{t-1} - \theta_2 Z_{t-2} - \dots - \theta_q Z_{t-q},$$

where Z_t is a white noise process with mean 0 and covariance Σ_Z positive definite, and θ_i 's and ϕ_i 's all k by k matrices of constants. As with the VMA(q) and VAR(p) processes, we may condense the VARMA(p,q) equation using the backward shift operator, giving us the following new equation:

$$\phi(B)X_t = \theta(B)Z_t,$$

where $\phi(B) = I_k - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$, and $\theta(B) = I_k - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$.

We now go over some properties of the VARMA(p,q) model. First, we have, for X_t satisfying the VARMA(p,q) equation, that X_t is stationary if and only if all of the solutions to the equation $\det[\phi(B)] = 0$ have modulus strictly greater than 1. Furthermore, if X_t is stationary, then it has an MA representation as follows:

$$X_t = Z_t + \Psi_1 Z_{t-1} + \Psi_2 Z_{t-2} + \dots ,$$

where each Ψ_i is a k by k matrix of constants, and can be found using the following equation:

$$\phi(B)\Psi(B) = \theta(B),$$

where $\Psi(B) = I_k + \Psi_1 B + \Psi_2 B^2 + \dots$.

The next property of the VARMA(p,q) model is invertibility. For X_t that satisfies the VARMA(p,q) equation, we have that X_t is invertible if and only if all solutions to the equation $\det[\theta(B)] = 0$ have modulus strictly greater than 1. Furthermore, if X_t is invertible, then X_t has an AR representation as follows:

$$X_t = Z_t + \pi_1 X_{t-1} + \pi_2 X_{t-2} + \dots,$$

Where each π_i is a k by k matrix of constants, and can be found using the following equation:

$$\theta(B)\pi(B) = \phi(B),$$

where $\pi(B) = I_k - \pi_1 B - \pi_2 B^2 - \dots$.

3.4 Estimation and Forecasting Under the VARMA(p,q) Model

As was said in the VMA(q) model, we will be using R to fit our VARMA(p,q) model, since what we are interested in is the difference between our forecasted values and the true values in our simulation. For more details on the estimation of parameters in the VARMA(p,q) model, see Tsay.

As for the forecasting, let h be the forecast origin, and $\{x_1, x_2, \dots, x_T\}$ be our observed sample. Then, our τ -step forward forecast of X_h , $X_h(\tau)$, is:

$$X_h(\tau) = \sum_{i=1}^p \phi_i X_h(\tau - i) - \sum_{j=1}^q \theta_j Z_{h+\tau-j},$$

Note that the second summation is always zero when $\tau > q$. Now, our estimator of $X_h(\tau)$, $\hat{X}_h(\tau)$, is just $X_h(\tau)$ with our fitted estimates of ϕ_i 's and θ_j 's replacing the unknown true values, as well as our estimates of the Z_i 's. Now, our forecast error is defined as always, and the estimator for it is as follows:

$$\hat{e}_h(\tau) = \hat{X}_h(\tau) - X_{h+\tau}$$

4 Bivariate VAR(1) Model Example

The main purpose of this example is to compare the multivariate model to several univariate models when there are few parameters that cause dynamic correlation in the “true” model. It is also an introduction and explanation of the code we will be using throughout all 3 examples.

In this example, the multivariate model will be estimating 4 parameters, where each univariate model will be estimating only 1 parameter.

4.1 VAR(1) Model Procedure and Prediction

We will start off our examples with a comparison between a bivariate VAR(1) model and two univariate AR(1) models fit on the same data set. We will conduct the following procedure:

1. Using R, we will generate a sample that follows a bivariate VAR(1) process using our predetermined “true” model. Our sample size will be 120 values. That is, we will generate x_1, \dots, x_{120} , each of which is a 2-dimensional vector. For clarity, we will write each value of x_i as $x_{1,i}$ and $x_{2,i}$.
2. We will fit a VAR(1) model to the first 100 observation, that is, fit our model to x_1, \dots, x_{100} . Then, we will fit 2 univariate AR(1) models to the first 100 observations as well, one to $x_{1,1}, x_{1,2}, \dots, x_{1,100}$, and one to $x_{2,1}, x_{2,2}, \dots, x_{2,100}$.
3. Using our fitted models, we will then forecast x_{101}, \dots, x_{120} . We will then have 2 forecasts of each x_i , where $101 \leq i \leq 120$. Let us denote the values forecasted by the VAR(1) model as M, for multivariate, and the values forecast by the 2 AR(p) processes as U, for univariate.
4. We will then calculate the following value for both forecasts M and U:

$$\text{MSE} = \sum_{i=101}^{120} (\text{norm}[\hat{x}_{100}(i) - x_{100+i}])^2$$

5. We will then repeat steps 1 to 4 100 times, and in the end, taking the sum of the MSE of M and the MSE of U. This is the final comparison we are looking for.

Clearly, our prediction is that the MSE of M will be smaller than that of U, as M has more information used in its forecasts than U does. In all of the examples, we will be using R to fit the VAR(1) model and AR(1) models to the sample. For this study, we will be using the “MTS” R package, which will allow us to do so.

4.2 The True Model

First, we have to come up with our “true” parameters that will follow a bivariate VAR(1) process. This will require us to have Σ_Z be positive definite, and ϕ_1 such that all eigenvalues of ϕ_1 have modulus less than 1, but not zero. The second requirement is equivalent to $\det[\phi(B)] = 0$ in the case where $p = 1$. If we fulfill these 2 restriction, we will have an X_t such that a) it satisfies to VAR(1) equation, b) it is stationary and invertible. Otherwise, there would not be any merits in using the model to predict the future.

Let us make this as simple as possible, and say that our 2 white noises, $Z_{1,t}$ and $Z_{2,t}$, are uncorrelated and identically distributed as normal random variables, with mean 0 and variance 4. That gives us our Σ_Z as follows:

$$\Sigma_Z = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

which is also a positive definite matrix, thus making it fit for our model.

It is also rather simple to find a ϕ_1 such that all of its eigenvalues have modulus less than 1. Let us use the following ϕ_1 for our example:

$$\phi_1 = \begin{bmatrix} 1/4 & 2/3 \\ -5/6 & 1/5 \end{bmatrix}$$

which can be shown to have eigenvalues $0.225 \pm 0.745i$, meaning both eigenvalues have modulus 0.60565. This confirms that we can use this arbitrary ϕ_1 for our model.

4.3 The Procedure in R

As mentioned before, we will be using the “MTS” package that is not installed by default in R. Some of the new commands we have access to in MTS are used to fit a multivariate time series using a VARMA(p,q) model, and generating a sample that follows a VARMA(p,q) distribution. These will prove invaluable in this study, as it was mentioned before that to fit a VMA(q) model, we require an iterative algorithm.

This particular example is a very basic model, as it is a VAR(1) model and X is only 2-dimensional, and would not benefit from a multivariate fitting as much as, say, a trivariate VARMA(1,1) model would. However, we will show that it is still more efficient to use a multivariate fit regardless of how many covariance parameters are in the model.

As for the code we are using, let us separate the code into chunks corresponding to the steps in the procedure. Note that the blue text is the code, and the red text will describe the code in the line above. First, we have input the “true” parameters of the VAR(1) process we will be simulating from, as well as define the statistics that will carry over the for loop:

```
library(MTS);                # this loads the MTS package
MSEUavg = 0;
MSEMAvg = 0;
phi1 = matrix(c(1/4, -5/6 , 2/3, 1/5), nrow = 2, ncol = 2);    # the true  $\phi_1$ 
sigmaZ = matrix(c(4, 0, 0, 4), nrow = 2, ncol = 2);           # the true  $\Sigma_Z$ 
```

Next is step 1 of our procedure, where we will generate a sample that follows a bivariate VAR(1) process using the VARMAsim() command from the MTS package, as well as start the loop. Note that a VAR(1) process is a special case of the VARMA(p,q) process, where $q = 0$, so we may use the VARMAsim to simulate a VAR(1) process.

```

for (i in c(1:100)) {
  X = VARMAsim(120 , arlags = 1, malags = NULL, cnst = NULL, phi = phi1, theta
  = NULL, skip = 0, sigma = sigmaZ)$series;
  # simulate the observations  $x_1, \dots, x_{120}$ 
  X1 = X$series[1:100];
  # this will be used to fit a univariate AR(1) model to  $x_{1,1}, \dots, x_{1,100}$ 
  X2 = X$series[121:220];
  # this will be used to fit a univariate AR(1) model to  $x_{2,1}, \dots, x_{2,100}$ 
  Xfit = matrix(c(X1, X2), nrow = 100, ncol = 2);
  # this will be used to fit the VAR(1) model to  $x_1, \dots, x_{100}$ 
  Xfuture = matrix(c(X$series[101:120], X$series[221:240]), nrow = 20, ncol = 2);
  # the true future values  $x_{101}, \dots, x_{120}$ 

```

For step 2 of the procedure, we fit the VAR(1) model to “Xfit”, and then fit 2 univariate AR(1) models to both “X1” and “X2”. Note that an AR(1) process is a special case of ARIMA(p,d,q) where $d = q = 0$, so we may use the default R command `arima0()` to fit an AR(1) model.

```

Mfit = VARMA(Xfit, p = 1, q = 0, include.mean = FALSE);
# this fits a VAR(1) model with mean 0 to  $x_1, \dots, x_{100}$ 
X1fit = arima0(X1, order = c(1, 0, 0), include.mean = FALSE);
# this fits an AR(1) model with mean 0 to  $x_{1,1}, \dots, x_{1,100}$ 
X2fit = arima0(X2, order = c(1, 0, 0), include.mean = FALSE);
# this fits an AR(1) model with mean 0 to  $x_{2,1}, \dots, x_{2,100}$ 

```

For step 3 of the procedure, the MTS package has a command that forecasts our fitted VAR(1) model “Mfit” using the same methods as described in the VAR section of

this essay. We use this, as well as R's default ARIMA prediction tool, to forecast x_{101}, \dots, x_{110} using both the multivariate and the 2 univariate methods.

```
M = VARMApred(Mfit, h = 20, orig = 0)$'pred';
# this gives us our multivariate forecasts of  $\hat{x}_{100}(1), \dots, \hat{x}_{100}(20)$ 
X1p = predict(X1fit, n.ahead = 20)$'pred';
# this gives us our univariate forecasts of  $\hat{x}_{1,100}(1), \dots, \hat{x}_{1,100}(20)$ 
X2p = predict(X2fit, n.ahead = 20)$'pred';
# this gives us our univariate forecasts of  $\hat{x}_{2,100}(1), \dots, \hat{x}_{2,100}(20)$ 
U = matrix(c(X1p, X2p), nrow = 20, ncol = 2);
# this gives us our univariate forecasts of  $\hat{x}_{100}(1), \dots, \hat{x}_{100}(20)$ 
```

For step 4 of our procedure, we simply take the norms of the prediction errors for the 2 methods and add them up, giving us our MSE for each method. We will also be adding our MSE for each method to the total MSE for each method from previous generations.

```
MSEM = sum((M - Xfuture)^2)/20;
# the MSE of the VAR(1) fitted model
MSEU = sum((U - Xfuture)^2)/20;
# the MSE of the AR(1) fitted models
MSEUavg = MSEUavg + MSEU;
# takes a running total of the MSE of our multivariate fitting
MSEMavg = MSEMavg + MSEM;
# takes a running total of the MSE of our univariate fitting
} # ends the for loop
MSEUavg = MSEUavg/100
# gives us our final MSE for the multivariate model
MSEMavg = MSEMavg/100
# gives us our final MSE for the univariate models
```

4.4 10 Sample Procedures

Now, we will run this code 10 times, so we can have our total average be of 1000 samples. We will record the MSE of the multivariate and univariate models each time in the following table:

Sample	Multivariate MSE	Univariate MSE
1	18.73076	19.64518
2	19.33851	21.03417
3	18.87541	20.13732
4	18.14727	19.12028
5	19.14115	20.71006
6	18.5274	19.59322
7	17.30565	18.96717
8	19.67333	20.87682
9	18.53624	19.94728
10	19.77226	21.82406
Average	18.80480	20.18556

Therefore, over 1000 samples of a VAR(1) process, the MSE from using a VAR(1) model was 18.80480, and the MSE from using two univariate models was 20.18556. This is a relatively good indication that using a multivariate model is “better” than using several univariate models when considering a time series with a dimension greater than

1. In this particular case, our MSE for the multivariate model was about 6.84% less than the MSE for the univariate models.

It is important to note that in this case, and the rest of the examples, the mean of X_t is zero. This means that forecasts produced by the AR or VAR models will decay towards zero as the forecast horizon gets larger. As a result of this, both models will be relatively close in their estimation, which means that their forecast error and MSE will also be fairly close. This can be seen in the results, where a higher MSE of M will mean a higher MSE of U. This is the case for any mean μ of X_t , except instead of decaying to zero, the forecasted values will converge to the mean.

In conclusion, we now have much greater confidence in the multivariate model, as even a model with very little dynamic correlation between $X_{1,t}$ and $X_{2,t}$ benefits when using a multivariate model over several univariate models. Next, we will compare the multivariate model with a lot of dynamic correlation to several univariate models.

5 Trivariate VARMA(1,1) Model Example

The main purpose of this example is to compare the multivariate model to several univariate models when there are many parameters that cause dynamic correlation in the “true” model.

In this example, the multivariate model will be estimating 18 parameters, where each univariate model will be estimating only 2 parameters.

5.1 VARMA(1,1) Model Procedure and Prediction

For the most part, the procedure will be the same as the procedure for the VAR(1) example, only we will be fitting 3 ARMA(1,1) models as our univariate models instead of 2 AR(1) models. Therefore, our procedure will be nearly identical to the previous procedure, with all differences noted as follows:

1. Using R, we will generate a trivariate VARMA(1,1) process with a predetermined “true” model, with sample size of 120 again. We will then have x_1, \dots, x_{120} , each a 3-dimensional vector.

2. We will fit a VARMA(1,1) model to the first 100 x_i 's, and 3 univariate ARMA(1,1) models to each $X_{i,t}$, for $i=1, 2, 3$.
3. We will then forecast x_{101}, \dots, x_{120} with both of our fitted models
4. We will then calculate the MSE of each method as before
5. We will repeat steps 1 to 4 10 times, and not 100, since the R command to fit a VARMA model takes a long time when there are so many parameters.

As before, our goal is to show that the MSE of multivariate models will be lower than that of several univariate models. This time, however, we expect the difference to be much larger than before, with multivariate being a much better alternative to several univariate models when there are many dynamic correlation parameters.

5.2 The True Model

This time, we need several more conditions to be met before we can say our true model is stationary and invertible. Once again, we will require Σ_Z to be positive definite, and we will set the mean of X_t to zero. However, now we have more parameters than before. We now have ϕ_1 and θ_1 , each a 3 by 3 matrix of constants. We now require that all eigenvalues of θ_1 have modulus less than 1 but not zero, for invertibility, as well as all eigenvalues of ϕ_1 have modulus less than 1 but not zero, for stationarity.

As before, for simplicity, we will assume that all $Z_{i,t}$'s are uncorrelated, identically distributed normal random variables with mean 0 and variance 4. That will give us Σ_Z as follows:

$$\Sigma_Z = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

which is a positive definite matrix.

All we require from our true θ_1 is that all of its eigenvalues have modulus less than 1. Therefore, we will use the following θ_1 :

$$\theta_1 = \begin{bmatrix} 1/5 & 1/2 & -2/3 \\ 2/5 & 1/3 & 1/3 \\ -2/5 & -2/5 & 1/6 \end{bmatrix}$$

which can be shown to have eigenvalues 0.848, -0.315, and 0.167, all having modulus less than 1.

Lastly, all we require from our ϕ_1 is that all eigenvalues have modulus less than 1. Therefore, we will use the following ϕ_1 :

$$\phi_1 = \begin{bmatrix} 4/5 & -1/10 & 3/10 \\ 7/20 & 1/2 & -3/10 \\ 2/5 & -1/5 & 2/7 \end{bmatrix}$$

which can be shown has eigenvalues 0.895, 0.679, and 0.012, which all have modulus less than 1.

Therefore, our true model for X_t will be stationary and invertible.

5.3 The Procedure in R

Since the code is very similar to that of the VAR(1) example, we will not separate the code into chunks corresponding to each step in the procedure. We will also only comment on the differences between the VAR(1) code and this new code.

```
library(MTS);
MSEUavg = 0;
MSEMavg = 0;
phi1 = matrix(c(4/5, -1/10, 3/10, 7/20, 1/2, -3/10, 2/5, -1/5, 2/7), nrow = 3, ncol = 3,
byrow = TRUE);          #  $\phi_1$  is now a 3 by 3 matrix
theta1 = matrix(c(1/5, 1/2, -2/3, 2/5, 1/3, 1/3, -2/5, -2/5, 1/6), nrow = 3, ncol = 3, byrow =
TRUE);          # this is defining the MA coefficient matrix,  $\theta_1$ 
sigmaZ = 4 * diag(3);    # the  $\Sigma_z$  is now a 3 by 3 matrix
for (i in c(1:10)) {
    # we will be looping only 10 times instead of 100 times, since the VARMA fit
    takes a long time when there are a lot of parameters
}
```

```

X = VARMAsim(120, arlags = 1, malags = 1, cnst = NULL, phi = phi1, theta =
theta1, skip = 0, sigma = sigmaZ);
# malags and theta are no longer NULL
X1 = X$series[1:100];
X2 = X$series[121:220];
X3 = X$series[241:340];          #we have a third univariate model now
Xfit = matrix(c(X1, X2, X3), nrow = 100, ncol = 3);
Xfuture = matrix(c(X$series[101:120], X$series[221:240], X$series[341:360]),
nrow = 20, ncol = 3);
Mfit = VARMA(Xfit, p = 1, q = 1, include.mean = FALSE);
# we now fit a VARMA(1,1) model to the sample
X1fit = arima0(X1, order = c(1, 0, 1), include.mean = FALSE);
# we now fit a univariate ARMA(1,1) model to the sample
X2fit = arima0(X2, order = c(1, 0, 1), include.mean = FALSE);
X3fit = arima0(X3, order = c(1, 0, 1), include.mean = FALSE);
# we now fit a third univariate model
M = VARMApred(Mfit, h = 20, orig = 0)$'pred';
X1p = predict(X1fit, n.ahead = 20)$'pred';
X2p = predict(X2fit, n.ahead = 20)$'pred';
X3p = predict(X3fit, n.ahead = 20)$'pred';
U = matrix(c(X1p, X2p, X3p), nrow = 20, ncol = 3);
MSEM = sum((M - Xfuture)^2)/20;
MSEU = sum((U - Xfuture)^2)/20;
MSEMavg = MSEMavg + MSEM;
MSEUavg = MSEUavg + MSEU;
}          # ends the for loop
MSEMavg = MSEMavg/10;
MSEUavg = MSEUavg/10;

```

5.4 20 Sample Procedures

Now, we will run the code multiple times, recording the final MSE of the multivariate fit and the univariate fit each time. This time the loop size is 10 instead of 100. The number of loops is 20, giving us an average over 100 samples twice, where we then take the average of our 2 samples of size 100.

Sample	Multivariate MSE	Univariate MSE
1	68.83437	68.22207
2	56.73507	72.76644
3	104.6958	100.111
4	60.13573	50.72712
5	64.21538	66.74007
6	55.64048	63.77421
7	69.50781	78.49815
8	86.1785	89.81336
9	45.30169	45.03916
10	58.21322	67.64536
Average	66.94581	70.33369
11-20 Average	64.61206	71.00506
Average	65.778935	70.66938

As the sample results show, the VARMA multivariate fit seems to not be as consistent as the VAR multivariate fit, as there are several instances of the univariate fit having less MSE than the multivariate fit. This is a result of the smaller loop size this time around, so the MSE of M and U have more variance. As expected, the MSE of the multivariate model when the loop size was 100 is decisively smaller than that of the multiple univariate models. The MSE of the multivariate models was on average 6.92% smaller than that of the univariate models, yielding about the same result as for the VAR model fit.

In conclusion, we do not have strong evidence to support the prediction that a multivariate process with a lot of dynamic correlation will benefit more from a multivariate model than a multivariate process with little dynamic correlation. However, we now have more evidence that multivariate models will be better than several univariate models when there is dynamic correlation.

6 Trivariate ARMA(2,1) Model Example

The main purpose of this example is to compare the multivariate model to several univariate model when there are no parameters that cause dynamic correlation in the “true” model.

In this example, the multivariate model will be estimating 27 parameters, where each univariate model will be estimating only 3 parameters.

6.1 The ARMA(2,1) Procedure and Prediction

This time, we will be comparing a multivariate model to several univariate models by fitting both of them to a sample that is 3 univariate ARMA(2,1) processes with no dynamic correlation. The procedure will be as follows:

1. Using R, we will generate 3 univariate ARMA(2,1) samples of size 120, each from a predetermined “true” model. That is, we will generate $x_{1,1}, \dots, x_{1,120}, x_{2,1}, \dots, x_{2,120}$, and $x_{3,1}, \dots, x_{3,120}$.

2. We will fit a VARMA(2,1) model to the first 100 values of each $X_{i,t}$, as well as 3 univariate ARMA(2,1) models to the same X_i 's..
3. We will forecast the next 20 values for each model. That is, we will have 2 forecasts of X_{101}, \dots, X_{120} .
4. We will calculate the MSE of the multivariate fit, and call it M. We will also do so for the univariate fits, and call it U.
5. We will repeat steps 1 to 4 20 times, the same as for the VARMA(1,1) example.

This time, our prediction will be that the MSE of the multivariate fit will be around the same as the MSE of the univariate fit.

6.2 The True Models

This time, we will have 3 true models. Since they will all be ARMA(2,1) processes with mean zero, they will all have the following representation, for all $i = 1, 2, 3$:

$$X_{i,t} = \phi_{i,1}X_{i,t-1} + \phi_{i,2}X_{i,t-2} + Z_{i,t} - \theta_1 Z_{i,t-1},$$

where θ_1 , $\phi_{i,1}$ and $\phi_{i,2}$ are constants, and $Z_{i,t}$ is a sequence of normal random variables with mean zero and variance 4.

For stationarity, we require the roots of each $\phi_i(B) = 0$ to have modulus greater than 1. We will keep these backward shift polynomials simple, and make each of them a perfect square. Let the following be our ϕ_i 's:

$$\phi_1(B) = 1 - (2/3)B + (1/9)B^2,$$

$$\phi_2(B) = 1 - (1/2)B + (1/16)B^2,$$

$$\phi_3(B) = 1 - (4/5)B + (4/25)B^2,$$

where it can easily be shown that all of these polynomials have roots that are greater than 1 in modulus.

For invertibility, we require that the root of $\theta_i(B) = 0$ must have modulus greater than 1. WE also require this root to differ from the roots of each corresponding ϕ_i , otherwise the model would be overparameterized. Therefore, we will use the following for our θ_i 's:

$$\theta_1(B) = 1 - (1/2)B,$$

$$\theta_2(B) = 1 - (1/3)B,$$

$$\theta_3(B) = 1 - (1/4)B,$$

where it can easily be shown that the roots have modulus greater than 1.

6.3 The Procedure in R

Once again, the R code used in this procedure is similar to that of the previous examples. We will only highlight the differences between this code and the previous ones.

```
library(MTS);
MSEUavg = 0;
MSEMAvg = 0;
phi1 = matrix(c(2/3, 0, 0, 0, 1/2, 0, 0, 0, 4/5), nrow = 3, ncol = 3);
# this time, the AR coefficient matrices are diagonal, since the "true" distributions
# are univariate, and have no dynamic correlation
phi2 = matrix(c(-1/9, 0, 0, 0, -1/16, 0, 0, 0, -4/25), nrow = 3, ncol = 3);
# there are also 2 AR coefficient matrices, as each univariate process is a
# ARMA(2,1) process
theta1 = matrix(c(-1/2, 0, 0, 0, -1/3, 0, 0, 0, -1/4), nrow = 3, ncol = 3);
# the MA coefficient matrix is also diagonal now
sigmaZ = 4*diag(3);
```

```

for (i in c(1:10)) {
  X = VARMAsim(120, arlags = 2, malags = 1, cnst = NULL, phi = matrix(c(phi1,
    phi2), nrow = 3, ncol = 6), theta = theta1, skip = 0, sigma = sigmaZ);
  X1 = X$series[1:100];
  X2 = X$series[121:220];
  X3 = X$series[241:340];
  Xfit = matrix(c(X1, X2, X3), nrow = 100, ncol = 3);
  Xfuture = matrix(c(X$series[101:120], X$series[221:240], X$series[341:360]),
    nrow = 20, ncol = 3);
  Mfit = VARMA(Xfit, p = 2, q = 1, include.mean = FALSE);
  X1fit = arima0(X1, order = c(2,0,1), include.mean = FALSE);
  X2fit = arima0(X2, order = c(2,0,1), include.mean = FALSE);
  X3fit = arima0(X3, order = c(2,0,1), include.mean = FALSE);
  M = VARMAPred(Mfit, h = 20, orig = 0)$'pred';
  X1p = predict(X1fit, n.ahead = 20)$'pred';
  X2p = predict(X2fit, n.ahead = 20)$'pred';
  X3p = predict(X3fit, n.ahead = 20)$'pred';
  U = matrix(c(X1p, X2p, X3p), nrow = 20, ncol = 3);
  MSEM = sum((M - Xfuture)^2)/20;
  MSEU = sum((U - Xfuture)^2)/20;
  MSEMavg = MSEMavg + MSEM;
  MSEUavg = MSEUavg + MSEU;
}      # the end of the for loop
MSEMavg = MSEMavg/10;
MSEUavg = MSEUavg/10;

```

6.4 20 Sample Procedures

As before, we will run the code multiple times, and record the MSE of the multivariate and univariate fit each time. Note that occasionally, the VARMA fit of the sample would produce arbitrarily high forecasts, which would result in arbitrarily high MSE. In those cases, I did not record the MSE, as it was anywhere between 10^{39} and 10^{86} . It occurred in 2 out of the 22 instances of running the procedure. It would also be obvious to any researcher that case that something went wrong and that they should not use that particular fitted model for predicting the future.

It is also worth noting that one can specify which parameters would be zero when fitting the VARMA model in R. The reason we did not do that is because one would not know for certain in a real situation if the dynamic correlation was zero. Also, if we did that, it would yield the exact same result as the univariate fit, since it would be the same process as a univariate fit.

Sample	Multivariate MSE	Univariate MSE
1	24.15139	22.53709
2	25.66828	23.63895
3	27.14961	27.40922
4	30.58018	23.21319
5	24.59105	24.17727
6	23.60594	22.88871
7	29.44406	26.31264
8	26.30852	24.4379
9	31.30392	28.5527
10	24.96626	23.08676
Average	26.77692	24.62544
11-20 Average	24.91205	23.191218
Average	25.84449	23.90833

As the results show, the multivariate fit is decisively worse than the several univariate fits. This shows that our prediction was wrong, and that it may not always be better to use the multivariate fit over several univariate fits. This could be the result of a small sample size not giving enough information to the VARMA estimation to determine that the dynamic correlation parameters are near zero. The MSE of the univariate fit was 7.5% lower than the MSE of the multivariate fit on average. This is almost the same

as how much better the MSE of the multivariate fit from the previous examples was compared to the MSE of the univariate fits.

In conclusion, contrary to our prediction, we now have evidence that it is not always better to use a multivariate model over several univariate models. However, it should be noted that this conclusion was derived from a special case where the “true” distribution had zero dynamic correlation.

7 Bivariate AR(1) Model Example

The main purpose of this example is to compare the multivariate model to several univariate models when the “true” distribution has very few dynamic correlation parameters, and they are equal to zero.

In this example, the multivariate model will be estimating 4 parameters, where each univariate model will be estimating 1 parameter.

7.1 The AR(1) Procedure and Prediction

This procedure is very similar to the previous example. We will fit a bivariate VAR(1) model and two univariate AR(1) models to a sample that is 2 AR(1) processes with no dynamic correlation.

1. Using R, we will generate 2 univariate AR(1) samples of size 120, each from a predetermined “true” model. That is, we will generate $X_{1,1}, \dots, X_{1,120}, X_{2,1}, \dots, X_{2,120}$.
2. We will fit a VAR(1) model to the first 100 values of each X_i , as well as 2 univariate AR(1) models to the same X_i 's.
3. We will forecast the next 20 values for each method, giving us 2 forecasts of X_{101}, \dots, X_{120} .
4. We will calculate the MSE of both forecasts, M for the multivariate, and U for the univariate.
5. Repeat steps 1 to 4 10 times.

Our prediction is that the MSE of the multivariate fit will be less than 7% less efficient than the MSE of the univariate fit. That is, M will be closer to U than it was in the previous example, where there were a lot of dynamic correlation parameters equal to zero.

7.2 The True Models

This time, we have 2 true models. Since they will both be AR(1) processes with mean zero, they will have the following representation:

$$X_{i,t} = \phi_{i,1}X_{i,t-1} + Z_{i,t},$$

where $\phi_{i,1}$ is a constant, and $Z_{i,t}$ is a sequence of normal random variables with mean 0 and variance 4.

All that we require of our $\phi_{i,1}$'s is that they are less than 1 in modulus. Therefore, let us use the following as our true model:

$$\phi_{1,1} = 3/4$$

$$\phi_{2,1} = 4/7$$

7.3 The Procedure in R

This R code will be nearly identical to the last example, with the exception that we can run the loop 100 times now. So, we will just state it:

```
library(MTS)
MSEUavg = 0;
MSEMAvg = 0;
phi1 = matrix(c(3/4, 0, 0, 4/7), nrow = 2, ncol = 2)
sigmaZ = 4 * diag(2)
for (i in c(1:100)) {
  X = VARMAsim(120, arlags = 1, malags = NULL, cnst = NULL, phi = phi1,
  theta = NULL, skip = 0, sigma = sigmaZ)
```

```

X1 = X$series[1:100]
X2 = X$series[121:220]
Xfit = matrix(c(X1, X2), nrow = 100, ncol = 2)
Xfuture = matrix(c(X$series[101:120], X$series[221:240]), nrow = 20, ncol = 2)
Mfit = VARMA(Xfit, p = 1, q = 0, include.mean = FALSE)
X1fit = arima0(X1, order = c(1, 0, 0), include.mean = FALSE)
X2fit = arima0(X2, order = c(1, 0, 0), include.mean = FALSE)
M = VARMApred(Mfit, h = 20, orig = 0)$'pred'
X1p = predict(X1fit, n.ahead = 20)$'pred'
X2p = predict(X2fit, n.ahead=20)$'pred'
U = matrix(c(X1p, X2p), nrow = 20, ncol = 2)
MSEM = sum((M - Xfuture)^2)/20
MSEU = sum((U - Xfuture)^2)/20
MSEMavg = MSEMavg + MSEM
MSEUavg = MSEUavg + MSEU
}
MSEMavg = MSEMavg/100
MSEUavg = MSEUavg/100

```

7.4 10 Sample Procedures

As usual, we will run the process 10 times, and record the MSE of the univariate and multivariate fits in the following chart. We are back to a loop size of 100 again, due to the “true” model having few parameters.

Sample	Multivariate MSE	Univariate MSE
1	13.34784	13.29545
2	13.77279	13.74112
3	13.02134	12.99223
4	14.61936	14.63598
5	14.23693	14.18709
6	14.19143	14.15773
7	13.84568	13.84071
8	14.23717	14.26158
9	14.00746	13.96312
10	14.70702	14.67039
Average	13.99870	13.97454

Clearly, these results show that the multivariate fit is only marginally worse, if not the same, as the univariate fit. This shows that our prediction was correct, and that a multivariate fit can be a better general model for processes with a low number of parameters. It is also noteworthy that in samples 4 and 8, the MSE of the multivariate fit was less than the MSE of the univariate fit, further suggesting the two models are almost the same in this case.

In this case, the MSE of the multivariate fit was only 0.17% worse than the MSE of the univariate fit. This is most likely due to the fewer number of estimated parameters compared to the last example.

8 Conclusion and Further Study

In this study, we have seen many unexpected results. Firstly, we saw a sample that follows a multivariate time series process with many dynamic correlation parameters may not benefit more from a multivariate fit than a time series with very few dynamic correlation parameters. This could have been caused by the relatively low size of the generated sample, as we fitted the models to a sample that was only of size 100. This could also be caused by the VARMA(p,q) model being relatively simple for a multivariate time series model. One could further research more complex models used for fitting a multivariate time series, such as one that does not converge so quickly to the mean when forecasting more than 10 steps ahead. Regardless, we saw that a multivariate model was decidedly more efficient than several univariate models when dealing with arbitrary multivariate time series processes with dynamic correlation.

Secondly, we saw that a multivariate fit on a multivariate time series with no dynamic correlation, but many parameters, was actively worse than several univariate models when used to forecast. This result was most likely caused by a low sample size, as we were using 98 data points to estimate 27 parameters. Also, we did not apply any testing for zero parameters, which in any normal circumstances, we would have done. This is because if we had done so, we would have yielded the exact same fit as the univariate fits, where we were trying to compare the two in their most basic form. This result shows that a multivariate model should not always be the “default” method used when forecasting any multivariate time series, particularly ones that are suspected of having many parameters.

Lastly, our most predictable result was that a multivariate model on a process with no dynamic correlation, but few parameters, was just as good as several univariate models when used to forecast. This result shows that a multivariate model should be used on any time series where the planned model has few parameters.

For further study, one could research model checking and order selection for the VARMA(p,q) process. This would allow one to apply the VARMA(p,q) model to a real data set, and make specifications of their own. One could also research the VARMAX model, where an exogenous regressor is added to the VARMA model.

Appendix

Tsay, R. S. (2014). *Multivariate Time Series Analysis, with R and Financial Applications*. Hoboken, New Jersey: John Wiley and Sons, Inc.