Notes on ARIMA Modelling

Brian Borchers

November 22, 2002

Introduction

Now that we understand the theoretical behavior of ARIMA processes, we will consider how to use ARIMA models to fit observed time series data and make forecasts. Before beginning this work, an obvious question needs to be answered. Why should we assume that some random time series can be adequately modeled by an ARIMA process?

The **Wold Decomposition Theorem** provides some theoretical justification. The decomposition theorem asserts that any covariance stationary process Y_t can be written as

$$Y_t = X_t + Z_t \tag{1}$$

where Z_t is a deterministic process, and X_t can be written as the moving average of uncorrelated $N(0, \sigma_A^2)$ shocks. That is,

$$X_t = A_t + \sum_{i=1}^{\infty} \psi_i A_{t-i} \tag{2}$$

where the A_i are independent $N(0, \sigma_A^2)$ random variables.

For our purposes, the deterministic part Z_t can be dealt with separately. In considering X_t, if the ψ_i coefficients decay quickly to 0, then it might be appropriate to truncate the infinite sum after q terms. In this case we would have an MA(q) model. Even if we can't safely truncate the infinite series, we can hope to find an ARMA(p,q) model whose ψ weights are a close match to those of X_t.

Many time series are nonstationary. The only kind of nonstationarity supported by the ARIMA model is simple differencing of degree d. In practice, one or two levels of differencing are often enough to reduce a nonstationary time series to apparent stationarity.

We will use the following procedure to model a time series as an ARIMA process and produce future forecasts.

1. Identify the appropriate degree of differencing d by differencing the time series until appears to be stationary.

2. Remove any nonzero mean from the differenced time series.

3. Estimate the autocorrelation and PACF of the differenced zero mean time series. Use these to determine the autoregressive order p and the moving average order q.

4. Estimate the coefficients $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$. This can be done in a variety of ways. One simple approach is to make sure that the resulting autocorrelations match the observed autocorrelations. However, a more robust method is maximum likelihood estimation.

5. Once the model has been fitted, we can produce future forecasts with associated uncertainties.

Model Identification

The first step in our process is finding the appropriate value of d. It is a good idea to begin this process by simply plotting the data. If the time series shows a strong trend (growth or decline), then the process is clearly not stationary, and it should be differenced at least once. The second test that we will use is to examine the estimated autocorrelation of the time series. For a stationary time series, the autocorrelations will typically decay rapidly to 0. For a nonstationary time series, the autocorrelations will typically decay slowly if at all. Computing the differenced time series is easy to do with the **diff** command in MATLAB.

The second step in our process is removing any nonzero mean from the differenced time series W. This is a very straight forward computation- just compute the mean of the time series, and subtract it from each element of the time series. If the mean is small relative to the standard deviation of the differenced time series W, then it should be safe to simply skip this step.

Example 1 Figures one through three show the analysis of a time series. In figure 1a, notice that there are clear long term shifts in the average. In figure 1b, the autocorrelations do not decay quickly to zero. In figure 2a, there is still a strong trend, and figure 2b shows that the autocorrelations are still not going to 0 very quickly. Finally, figure 3a shows data that appear to be stationary and figure 3b confirms that the autocorrelations die out quickly.

The mean for the second difference of the original time series is 2.12, and the standard deviation for this time series is 1.58, so it seems wise to include the mean in fitting an ARIMA model to the data.

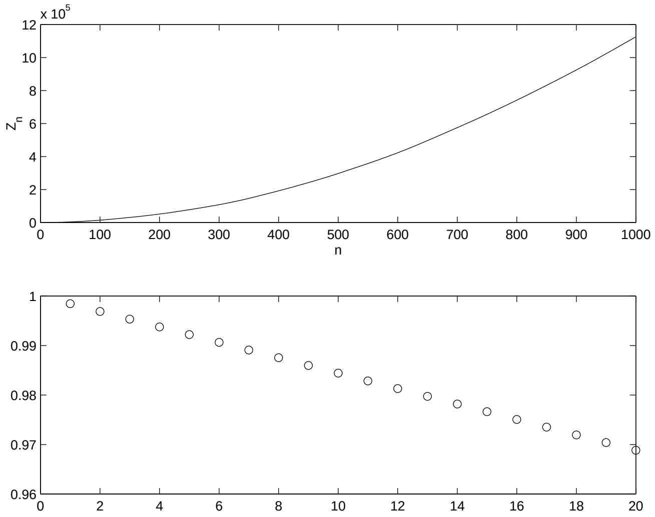

Figure 1: Example time series with no differencing.

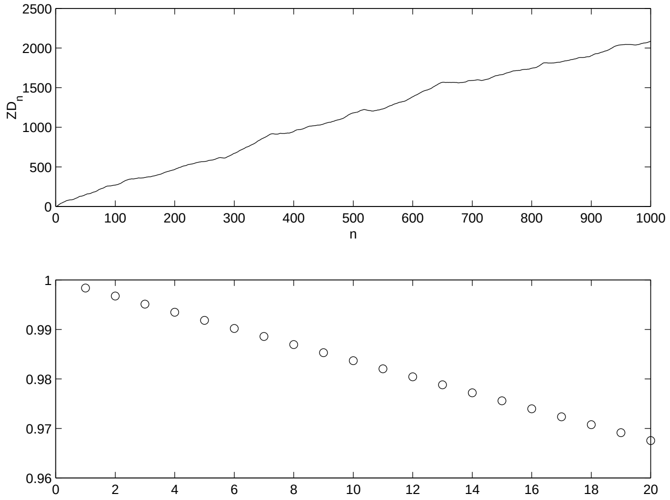

Figure 2: Example time series with $d = 1$.

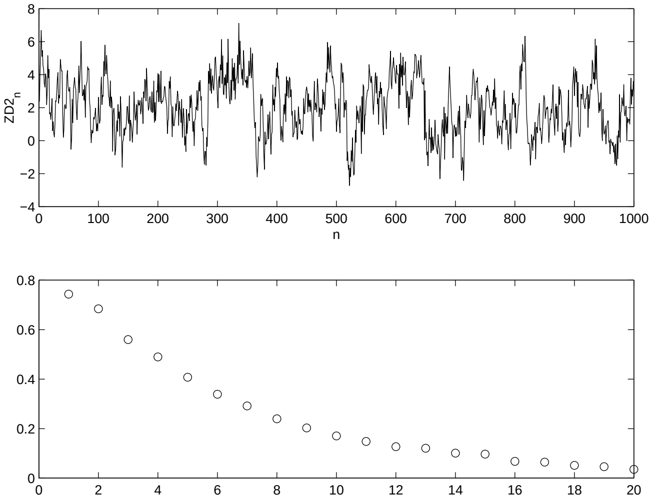

Figure 3: Example time series with $d = 2$.

Order	ρ_k	$\hat{\phi}_{kk}$
(1,0)	exp decay	only $\hat{\phi}_{11}$ nonzero
(0,1)	only ρ_1 nonzero	exp decay
(2,0)	exp or damped sine wave	only $\hat{\phi}_{11}$, $\hat{\phi}_{22}$ nonzero
(0,2)	only $\rho_{1,2}$ nonzero	exp or damped sine wave
(1,1)	exp decay	exp decay

Table 1: Rules for selecting p and q.

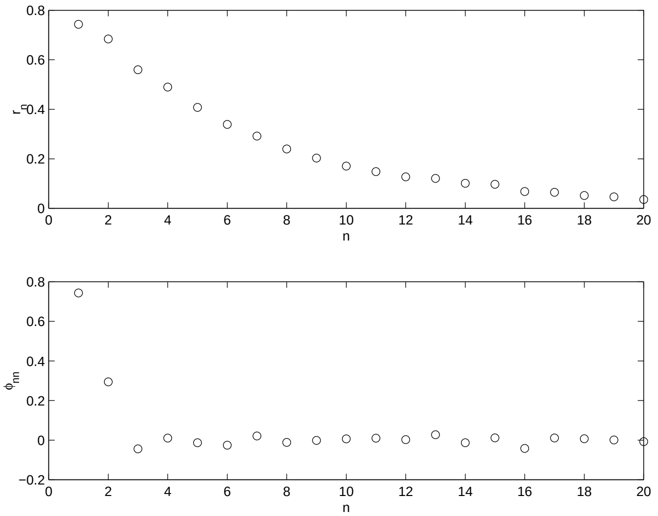

Figure 4: Estimated autocorrelation and PACF for the example time series.

The next step in the process is determining the autoregressive order p and the moving average order q for the differenced time series W. Table 1 (taken from BJR) summarizes the behavior of ARMA(p,q) processes for $p = 0, 1, 2$ and $q = 0, 1, 2$.

Example 2 Continuing with the time series from the last example, we removed the mean of 2.12, and then estimated the autocorrelations and partial autocorrelation function. Figure 4 shows the estimated autocorrelations and partial autocorrelation. Notice that both $\hat{\phi}_{11}$ and $\hat{\phi}_{22}$ are nonzero, and that the r_n appear to decay exponentially. This suggests that an ARIMA(2,2,0) model would be appropriate.

Once we've determined p, d, and q, the final step is to estimate the actual parameters ϕ_1, …, ϕ_p, θ_1, …, θ_q. One very simple approach can be used if we have formulas for the autocorrelations in terms of the parameters. For example,

Order	ρ_1	ρ_2
(1,0)	$\rho_1 = \phi_1$	
(0,1)	$\rho_1 = -\theta_1/(1+\theta_1^2)$	
(2,0)	$\rho_1 = \phi_1/(1-\phi_2)$	$\rho_2 = (\phi_1^2)/(1-\phi_2) + \phi_2$
(0,2)	$\rho_1 = -\theta_1(1-\theta_2)/(1+\theta_1^2+\theta_2^2)$	$\rho_2 = -\theta_2/(+\theta_1^2+\theta_2^2)$
(1,1)	$\rho_1 = (1-\theta_1\phi_1)(\phi_1-\theta_1)/(1+\theta_1^2-2\phi_1\theta_1)$	$\rho_2 = \rho_1\phi_1$

Table 2: Rules for selecting p and q.

for an AR(2) process, we know that

$$\rho_1 = \frac{\phi_1}{1-\phi_2} \tag{3}$$

and

$$\rho_2 = \frac{\phi_1^2}{1-\phi_2} + \phi_2. \tag{4}$$

We can substitute our estimates r_1 and r_2 for the differenced time series and solve these equations to obtain ϕ_1 and ϕ_2. Table 2 summarizes the equations to be solved for the (1,0), (0,1), (2,0), (0,2) and (1,1) cases.

Example 3 Continuing our earlier example, we've decided to fit an AR(2) model to w_n. We have $r_1 = 0.7434$ and $r_2 = 0.6844$. Solving the above equations for ϕ_1 and ϕ_2, we get the estimates $\phi_1 = 0.5246$ and $\phi_2 = 0.2943$. In fact, this series was generated using $\phi_1 = 0.5$ and $\phi_2 = 0.3$.

This simple procedure for fitting the model parameters may be adequate for some simple models, but it is not precise enough for some purposes, and it cannot easily be extended to models in which p or q is large. We'll move on to the more sophisticated technique of maximum likelihood estimation.

Maximum Likelihood Estimation and Least Squares

Given a set of observations x_1, x_2, …, x_n, which come from a probability distribution with a parameter θ, how can we estimate the parameter θ?

Let $f(x;\theta)$ be the probability distribution function. Given θ, what is the probability of obtaining a particular value x_i?

$$P(X = x_i) = \int_{x_i}^{x_i} f(x;\theta)dx = 0. \tag{5}$$

Because f is a *probability density function*, $f(x;\theta_i)$ is not a probability value. However, we can integrate $f(x;\theta_i)dx$ over any interval to find the probability that X lies in that interval. In particular, we could integrate over a small interval of width 2δ. The probability of obtaining a value of X in such an interval around x_i is roughly $2f(x_i;\theta)\delta$. If $f(x_i;\theta)$ is very small, then the probability of obtaining a value near x_i is very small.

The value of $f(x_i;\theta)$ is called the **likelihood** of x_i. More generally, given a value of the parameter θ, the likelihood of the data set x_1, x_2, $\ldots$, x_n is

$$L(\theta) = f(x_1;\theta)f(x_2;\theta)\cdots f(x_n;\theta). \tag{6}$$

The **maximum likelihood principle** asserts that we should select the value $\hat{\theta}$ that maximizes $L(\theta)$. Maximizing $L(\theta)$ is a problem that depends very much on the particular probability density function $f(x;\theta)$. In many cases, it is possible to find an explicit formula for $\hat{\theta}$. In other cases, a general purpose optimization algorithm can be applied to estimate $\hat{\theta}$.

Example 4 Suppose that the observations x_1, x_2, $\ldots$, x_n come from a normal distribution known standard deviation σ and unknown mean μ. We wish to estimate the mean μ.

In this case, the probability density function is

$$f(x;\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \tag{7}$$

The likelihood function is

$$L(\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x_i-\mu)^2/2\sigma^2} \tag{8}$$

Since the natural logarithm is a monotone increasing function, we can just as easily maximize the logarithm of $L(\mu)$.

$$l(\mu) = \log L(\mu) = n \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{2\sigma^2}. \tag{9}$$

Since the first term is a constant, and the factor of $2\sigma^2$ in the denominator is constant, maximizing $L(\mu)$ is equivalent to minimizing

$$S(\mu) = \sum_{i=1}^{n} (x_i - \mu)^2. \tag{10}$$

This is a least squares problem. By differentiating with respect to μ and setting the derivative equal to 0, we find that the optimal value of μ is

$$\hat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n}. \tag{11}$$

Maximum Likelihood Estimation for an ARMA Model

Now, consider the problem of fitting an ARM(p,q) model to a sequence w_1, w_2, $\ldots$, w_n. Suppose that we also knew the values of w and a that preceeded the time series, as well as σ_A^2. By using the model equations, we could compute a_1, a_2, …, a_n.

The likelihood function associated with these a values is

$$L(\phi, \theta) = \prod_{t=1}^{n} \frac{1}{\sqrt{2\pi\sigma_A^2}} e^{\frac{-a_t^2}{2\sigma_A^2}} \tag{12}$$

Ignoring constant factors and taking the logarithm, we're left with the least squares problem of minimizing

$$S(\phi, \theta) = \sum_{t=1}^{n} a_t^2. \tag{13}$$

In practice, we don't have all of the previous values of a and w. We will use a crude approximation that works well when time series is long compared to p and q. We simply set a_1, a_2, …, a_q to 0. If $p > q$, we also set all values of w before w_1 to 0. We then use the model equations to compute the remaining values of a, and let

$$S^*(\phi, \theta) = \sum_{t=q+1}^{n} a_t^2. \tag{14}$$

Other more sophisticated estimates of the sum of squares are possible. See Box, Jenkins, and Reinsel for details.

Now that we have an approximate sum of squares $S^*(\phi, \theta)$, we're left with the problem of minimizing the sum of squares. This is a nonlinear least squares optimization problem which we won't discuss in this course. Suffice it to say that algorithms such as the Levenberg-Marquardt method can be used to find values of ϕ and θ that minimize S^*.

In performing this minimization, one problem that can occur is that we may find an "optimal" model which is non stationary. In some cases it is possible to introduce constraints in the optimization problem to eliminate these false solutions. In practice, it's often simpler to just check the resulting model for stability.

Once optimal values of the parameters $\hat{\phi}$ and $\hat{\theta}$ have been found, we still need to estimate σ_A^2. Since there are $n - q$ squares of a_t values in S^*, a reasonable estimate is

$$\hat{\sigma}_A^2 = \frac{S^*(\hat{\phi}, \hat{\theta})}{n - q}. \tag{15}$$

Depending on the amount of available time series data, there may be considerable uncertainty in the model parameters ϕ and θ. The nonlinear least squares procedure will also a parameter covariance matrix which gives the parameter uncertainties. For convenience, we group the parameters into a single vector β.

$$\beta = \begin{bmatrix} \phi \\ \theta \end{bmatrix} \tag{16}$$

The parameter covariance matrix is

$$C = 2\hat{\sigma}_A^2 (J^T(\hat{\beta})J(\hat{\beta}))^{-1} \tag{17}$$

where $J(\beta)$ is the Jacobian matrix for the a_t.

$$J_{i,k}(\beta) = \frac{\partial a_i}{\partial \beta_k} \tag{18}$$

Once we've found the parameter covariance matrix, an approximate 95% confidence interval for a parameter β_k is

$$\beta_k = \hat{\beta}_k \pm 1.96\sqrt{C_{k,k}} \tag{19}$$

Be aware that these confidence intervals are often quite large. Unless we have a very long time series, and σ_A^2 is relatively small, it can be impossible to obtain precise estimates of the model parameters.

Example 5 For this example, we generated 5000 points from an ARMA(1,1) process with $\phi_1 = 0.6$ and $\theta_1 = 0.4$. Figure 5 shows a contour plot of $S^*(\phi, \theta)$. We then applied the least squares parameter estimation technique. The output, including 95% confidence intervals for the parameters and an estimate of σ_A^2 is shown below. Note that the optimal values of ϕ_1 and θ_1 are reasonably close to the true values that were used in generating the data. Also notice that the resulting ARMA(1,1) model satisfies the stationarity condition.

```
>> lsexamp

sigmaa2 =

    0.1006

parameter estimates with 95\% confidence intervals

ans =

    0.4389    0.5521    0.6654
    0.1859    0.3148    0.4438

optimal sum of squares was

ans =

  503.1670
```

There is quite a bit of uncertainty in the parameter estimates, even though our time series had 5000 points and σ_A^2 was relatively small. The problem here is that many acceptable ARMA(1,1) models. This is shown in Figure 5 by the

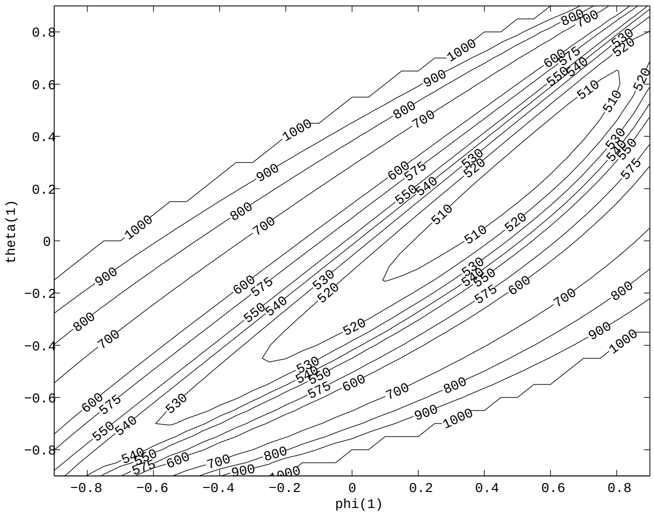

Figure 5: Contour plot of $S^*(\phi, \theta)$.

large "flat spot" in the contour map. Any combination of ϕ_1 and θ_1 in this area results in an ARMA(1,1) model with a small sum of squares. The available data simply are not adequate to precisely pin down the parameter values.

Forecasting

Now that we can fit an ARIMA model to an observed time series, we will discuss the problem of forecasting future observations.

Suppose that we are now at time n, and wish to predict the observations at times $n+1$, $n+2$, ..., $n+l$. We will use the notation z_{n-k} for the observations up to time n. As before, we will use Z_{n+k} for random future values of the time series. We will use $\hat{z}_n(l)$ for the predicted observation at time $n+l$ based on observations through time n.

In making a forecast $\hat{z}_n(l)$, we want to minimize the expected value of the square of the error in the forecast.

$$\min E[(Z_{n+l} - \hat{z}_n(l))^2] \tag{20}$$

where this expected value is conditioned on all of the observations through time n. Obviously, we should pick $\hat{z}_n(l) = E[Z_{n+l}]$. Using the $\psi()$ function, we know that

$$Z_{n+l} = \sum_{j=0}^{\infty} \psi_j A_{n+l-j} = A_{n+l} + \psi_1 A_{n+l-1} + \ldots \tag{21}$$

This infinite sum contains some terms which correspond to times up to time n and other terms which lie in the future and are still random. To get the expected value of Z_{n+l}, we take the expected value of each term on the right hand side. For A_{n+l} and other future inputs from the white noise, this expected value is 0. For A_n and other past white noise inputs, the expected value of A_{n-k} is the actual value a_{n-k} that was observed. Thus our prediction is given by

$$\hat{z}(l) = \sum_{j=l}^{\infty} \psi_j a_{n+l-j}. \tag{22}$$

The random error associated with our forecast is

$$e_n(l) = A_{n+l} + \psi_1 A_{n+l-1} + \ldots + \psi_{l-1} A_{n+1}. \tag{23}$$

Clearly, the expected value of $e_n(l)$ is 0. Furthermore, we can work out the variance associated with our prediction.

$$Var(e_n(l)) = Var(A_{n+l}) + \psi_1^2 Var(A_{n+l-1}) + \ldots + \psi_{l-1}^2 Var(A_{n+1}) \tag{24}$$

$$Var(e_n(l)) = \sigma_A^2 (1 + \psi_1^2 + \ldots + \psi_{l-1}^2). \tag{25}$$

There are three important practical issues that we need to resolve before we can actually start computing forecasts. The first problem is that we have a time series z_k, but not the corresponding a_k series. To compute the a sequence, notice that

$$e_n(1) = A_{n+1} \tag{26}$$

Thus

$$z_n - \hat{z}_{n-1}(1) = a_n. \tag{27}$$

We can use this to compute the values a_k for $k \leq n$. Just compute the lag 1 predictions, and subtract them from the actual values. In doing this, we may have to refer to z_k and a_k values from before the start of our observations. Set these to 0. In practice, the 0 initial conditions will have little effect on the forecasts.

The second issue is that we may not know σ_A^2. In this case, we use the sample variance of the a values that we have computed as an estimate for σ_A^2.

The third issue is that evaluating the infinite sum

$$\hat{z}(l) = \sum_{j=l}^{\infty} \psi_j a_{n+l-j}. \tag{28}$$

may be impractical. If the ψ_j weights decay rapidly, we can safely truncate the series, but if the ψ_j weights decay slowly this may be impractical. Furthermore, in the case of an ARIMA process with $d > 0$, the series will not converge! Fortunately, it is also possible to use the two other main forms of the model

$$Z_n = A_n + \pi_1 Z_{n-1} + \pi_2 Z_{n-2} + \ldots \tag{29}$$

or

$$\varphi(B) Z_n = \theta(B) A_n \tag{30}$$

for forecasting. If the model is purely autoregressive, then the π weights are the way to go. If the model is purely moving average, then it's best to use the ψ weights. For mixed models, the form $\varphi(B)Z_n = \theta(B)A_n$ is usually the easiest to work with.

In making a forecast using any of the three forms of the model, we use the same basic idea. We start by computing the previous a_k values. Next, we substitute observed or expected values for all terms in the model to get $\hat{z}_n(l)$. The expected values of all future a_{n+k} values are 0. The expected values of future z_{n+k} values are given by our predictions $\hat{z}_n(k)$, $k = 1, 2, \ldots$. We compute $\hat{z}_n(1), \hat{z}_n(2), \ldots, \hat{z}_n(l)$, and then use the variance formula to get confidence intervals for our predictions.

Example 6 The following time series was randomly generated according to an ARIMA(1,1,1) process with $\phi_1 = 0.3$ and $\theta_1 = 0.1$. (Read across the rows.)

```
-0.4326   -2.1847   -2.4184   -2.2134   -3.3271   -2.3557   -0.9942
-0.7423   -0.3356   -0.0717   -0.1967    0.5102    0.0614    2.1688
 2.4463    2.6571    3.7757    4.0639    4.0488    3.2215    3.3509
 2.0241    2.4740    4.1611    3.8131    4.6359    6.0510    4.7563
 3.0864    3.3006    2.9079    3.5201    4.4503    5.3598    6.8516
 7.8388    9.2589    8.3634    8.1952    7.9900
```

We will use the first 35 points to predict the last five values. First, we write out the model in the mixed form.

$$(1 - 0.3B)(1 - B)Z_n = (1 - 0.1B)A_n \tag{31}$$

$$(1 - 1.3B + 0.3B^2)Z_n = (1 - 0.1B)A_n \tag{32}$$

$$Z_n = 1.3Z_{n-1} - 0.3Z_{n-2} + A_n - 0.1A_{n-1} \tag{33}$$

We'll also need the first four ψ weights.

$$\psi(B) = 1 + 1.2B + 1.26B^2 + 1.278B^3 + 1.2834B^4 + \ldots \tag{34}$$

Notice that since this is a nonstationary series, this power series will not converge for B=1. That is OK, because we're only going to use the first four ψ weights.

Next, we have to compute the a_k for $k = 1, 2, \ldots, 35$. The lag 1 prediction of $\hat{z}_{n-1}(1)$ is given by

$$\hat{z}_{n-1}(1) = 1.3z_{n-1} - 0.3z_{n-2} - 0.1a_{n-1} \tag{35}$$

We then use the formula $a_n = z_n - \hat{z}_{n-1}(1)$ to compute the unknown a_n values.

```
%
% Compute a(1) through a(35) from the time series.
%
a(1)=z(1);
a(2)=z(2)-1.3*z(1)+0.1*a(1);
```

```
for i=3:35,
  a(i)=z(i)-1.3*z(i-1)+0.3*z(i-2)+0.1*a(i-1);
end;
%
% We’ll need an estimate of sigma_A^2.
%
sigmaaest=std(a(1:35))
%
% Also set future a(i) values to 0.
%
for i=36:40,
  a(i)=0.0;
end;
%
% And print out everything.
%
a
```

The output from this script is as follows:

```
sigmaaest =

    0.9423


a =

  Columns 1 through 7

   -0.4326   -1.6656    0.1253    0.2877   -1.1465    1.1909    1.1892

  Columns 8 through 14

   -0.0376    0.3273    0.1746   -0.1867    0.7258   -0.5883    2.1832

  Columns 15 through 21

   -0.1364    0.1139    1.0668    0.0593   -0.0956   -0.8323    0.2944

  Columns 22 through 28

   -1.3362    0.7143    1.6236   -0.6918    0.8580    1.2540   -1.5937

  Columns 29 through 35

   -1.4410    0.5711   -0.3999    0.6900    0.8156    0.7119    1.2902
  Columns 36 through 40

         0         0         0         0         0
```

Now that we have the previous z and a values in place, we can go ahead and compute the forecasts and the confidence intervals. The following script does the work.

```
%
% Compute the forecasts for z(36) through z(40)
%
for i=36:40,
  z(i)=1.3*z(i-1)-0.3*z(i-2)+a(i)-0.1*a(i-1);
end;
%
% Compute the variances.
%
psi=[1.2 1.26 1.278 1.2834];
v(1)=sigmaa^2;
v(2)=sigmaa^2*(1+psi(1)^2);
v(3)=sigmaa^2*(1+psi(1)^2+psi(2)^2);
v(4)=sigmaa^2*(1+psi(1)^2+psi(2)^2+psi(3)^2);
v(5)=sigmaa^2*(1+psi(1)^2+psi(2)^2+psi(3)^2+psi(4)^2);
%
% 95% CI's.
%
disp('    lower      mean      upper');
[z(36:40)-1.96*sqrt(v') z(36:40) z(36:40)+1.96*sqrt(v')]
```

The output from this script is as follows:

```
>> forecast
    lower       mean     upper

ans =

    5.3232    7.1702    9.0172
    4.3806    7.2657   10.1509
    3.5877    7.2944   11.0011
    2.9085    7.3030   11.6975
    2.3125    7.3056   12.2987
```

Notice that the confidence intervals get wider as we move forward in time. For this particular data set, it is impossible to make very precise predictions.

In fact, the next five points in the time series were

```
    7.8388
    9.2589
    8.3634
    8.1952
    7.9900
```

These points lie well within the 95% confidence intervals.

Sunspot Data

In this section of the notes, we'll analyze the number of sun spots reported during each year from 1770 to 1865. The raw data are shown in Figure 6. Estimated auto correlations out to $k = 30$ are shown in Figure 7. The PACF is shown in Figure 8. The sample spectrum is shown in Figure 9. Notice that both the autocorrelations and the sample spectrum clearly show periodicity with a period of about 10 or 11 years.

The initial data set appears to be stationary, so we won't try any differencing. The damped sine wave pattern in the autocorrelations, along with the first two nonzero PACF coefficients suggest that an ARMA(2,0) model would be appropriate for this data. We used the least squares technique to fit this model to the data, and obtained the following model:

```
sigmaa2 is estimated as

sigmaa2 =

  268.9646

phi parameters have 95\% confidence intervals

ans =

    1.1389    1.3524    1.5660
   -0.8742   -0.6606   -0.4471
```

Next, we forecast the sunspot numbers for 1866 through 1869. The forecasts (with 95% confidence intervals) were:

```
Predictions for 1866-1869, with confidence intervals

ans =

   -7.9842   24.1601   56.3044
  -26.5730   27.4924   81.5577
  -29.9734   35.8568   89.9222
  -24.4648   44.9676  110.7978
```

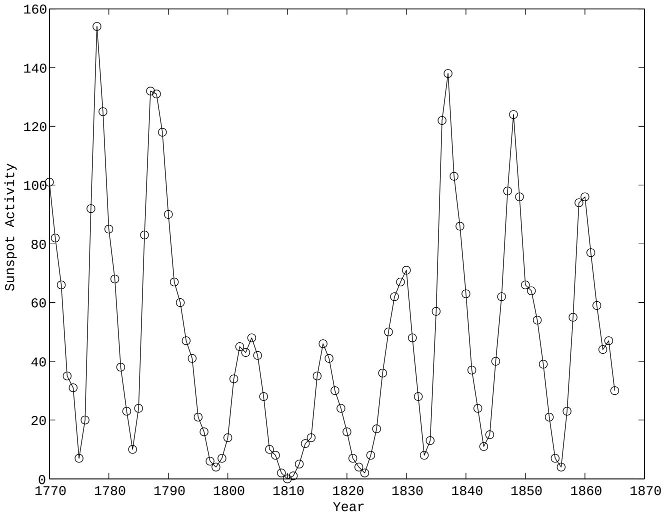

Figure 6: Sunspot Data.

It's clear that this time period covers a relatively low part of the sunspot cycle, with the count starting to go up by 1869. However, the confidence intervals are quite large. In fact, the actual sunspot counts for these years were 16, 7, 37, and 74.

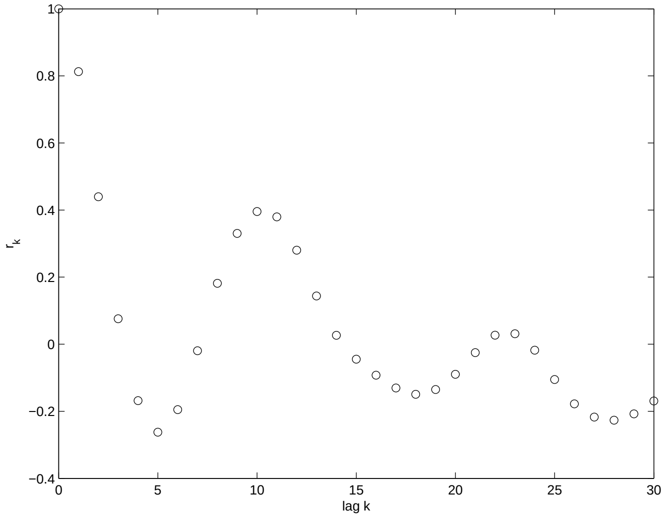

Figure 7: Autocorrelations.

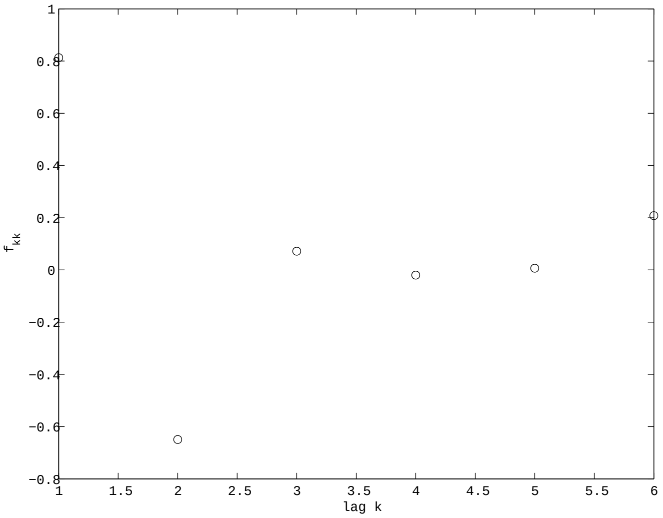

Figure 8: Partial Autocorrelation Function.

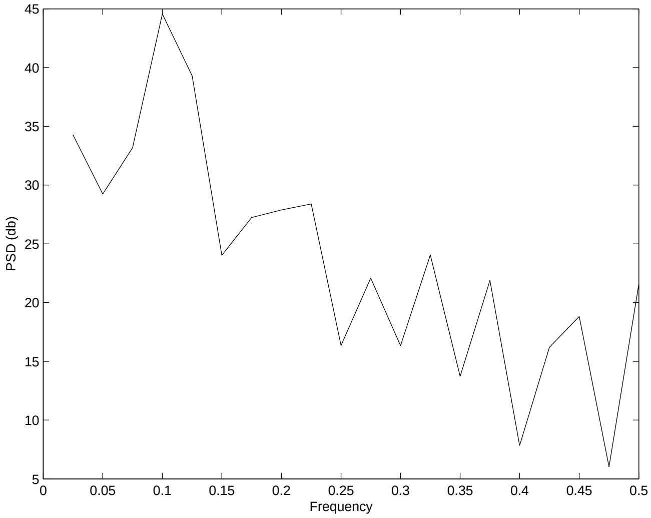

Figure 9: Sample Spectrum.