LA-13292-MS

UC-700
Issued: May 1997

Disaggregating Time Series Data

Shirley Bleasdale Joubert
Tom Burr
James C. Scovel

MASTER

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

HH

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

DISAGGREGATING TIME SERIES DATA

by

Shirley Bleasdale Joubert, Tom Burr, and James C. Scovel

ABSTRACT

This report describes our experiences with disaggregating time series data. Suppose we have gathered data every two seconds and want to guess the data at one-second intervals. Under certain assumptions, there are several reasonable disaggregation methods as well as several performance measures to judge their performance. Here we present results for both simulated and real data for two methods using several performance criteria.

1. Introduction

The reader can probably imagine many situations in which we would be interested in disaggregating time series data. Our interest in disaggregation began while we were analyzing satellite-based sensor data.[1] Some detector counts were not exhibiting the relationships we anticipated, which led us to investigate the time scale. Suppose we expect channel 1 counts to be positively related to channel 2 counts, but that we record channel 2 counts every two seconds and channel 1 counts every second. In our earlier work[1] we aggregated the channel 1 counts to form two-second intervals, and then failed to see the anticipated relationship between channel 1 and channel 2 counts at two-second intervals. There is reason to believe that the relationship between channel 1 and channel 2 counts depends on the time interval, so we would have liked to observe one-second interval counts for both channel 1 and channel 2. The methods presented in this paper are applicable in such a case. In a sense we can consider disaggregation to be interpolation (to “missing data”). Therefore, whether one method is better than another will depend on the model for the process that is generating the data.

Some natural questions to ask about any disaggregation procedure are as follows:

1. How should we measure how well the procedure performs?
2. What are the required sample sizes for the disaggregation method to perform well?
3. What range of time series models can be allowed (theoretically and practically)?
4. How much model departure can be tolerated before the performance degrades too much?

In this report we begin to answer these four questions. In Section 2 we give a simple (naive) method and a model-based method that uses a Kalman filter approach (dynamic linear model, dlm) to represent the aggregation process. In Section 3 we consider performance measures. In Section 4 we give performance results for some simple autoregressive moving average (ARMA) models. In Section 5 we give performance results for some models that differ somewhat from ARMA models but that are treated as though they followed an ARMA model. Section 6 gives results on some real data sets. Section 7 is a summary.

2. Two Disaggregation Methods

In our problem we are given a series of aggregates, *y(t), t=1,2,...,n,* where each aggregate is the nonoverlapping sum of *k* observations from an unobserved series, *x(i)*. Our *y(t)* series can then be written

$$y(t) = \sum_{i=(t-1)k+1}^{tk} x(i) \ . \tag{1}$$

In the work presented here, we summed two values together, so $k = 2$, but these methods can be extended to any *k*. We are interested in estimating the *x(i)*'s given the *y(t)*'s.

Let *y(t)* for *t=1,2,...,n* denote the aggregates for each time step *t*. We assumed that the underlying series *x(i)* can be represented as a stationary ARMA process or a non-stationary auto-regressive integrated moving average (ARIMA) process.[2,3] A stationary ARMA(p,q) process can be expressed as

$$x_i = \phi_1 x_{i-1} + \ldots + \phi_p x_{i-p} + \varepsilon_i + \theta_1 \varepsilon_{i-1} + \ldots + \theta_q \varepsilon_{i-q} \ , \tag{2}$$

where $\phi_1, \ldots, \phi_p$ are the autoregressive parameters, $\theta_1, \ldots, \theta_q$ are the moving average parameters, and $\{\varepsilon_i\}$ is a Gaussian white noise process. Here the level of the underlying process *x(i)* is constant, whereas in an ARIMA model the underlying process is nonconstant. An ARIMA(p,d,q) model can be expressed as

$$(1 - \phi_1 B - \ldots - \phi_p B^p) \cdot (1 - B)^d \cdot x_i = \theta_0 + (1 + \theta_1 B + \ldots + \theta_q B^q) \cdot \varepsilon_i \ , \tag{3}$$

where θ_0 is a constant which is usually referred to as a trend parameter. Thus, if we are summing over two time-steps, $y_1 = x_1 + x_2$, $y_2 = x_3 + x_4$,..., $y_t = x_{2t-1} + x_{2t}$, where the *x(i)*'s are from some ARMA or ARIMA process.

The simple (or naive) method to estimate the *x(i)* series is to divide each value in the *y(t)* series by the number of time-steps aggregated and use this value *k* times as our estimate of the *x(i)*'s. That is, if the *y(t)* series is formed by summing two values together, the naive estimate of the *x(i)* series would be

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

$$x(i) = \left(\frac{y(1)}{2}, \frac{y(1)}{2}, \frac{y(2)}{2}, \frac{y(2)}{2}, \ldots, \frac{y(n)}{2}, \frac{y(n)}{2}\right), i = 1, \ldots, 2n \ . \tag{4}$$

This method is well known and is used in many applications.

Our second approach was to write our model as a state space model with aggregation and then apply the Kalman filter to obtain the *x(i)* series. This method uses recursive estimation to find the disaggregates. The method that we used was outlined by Al-Osh,[4] who referred to it as a dynamic linear model method. The theory behind the dynamic linear model, which is based on the state space representation of models and the Kalman filter, can be found in both the engineering and statistics literature. See References 3, 5, and 6 for examples.

Let *y(t)* for *t=1,2,...,n* denote the aggregate for each time-step *t*. Let *Z(t)* be an $m \times 1$ vector, $m \geq k$, where *k* is the number of disaggregates for each period *t, x(tk), x(tk-1), ..., x(tk-k+1)* and *m* is defined by letting $r = \max(p,q+1)$ and $m = k+r-1$. The remaining $m-k$ elements of *Z(t)* that constitute the state space form can be functions of the disaggregates or constants. The relationship between *y(t)* and its disaggregates *x(tk), ..., x(tk-k+1)* is linear and can be specified by

$$y(t) = H'Z(t), t = 1, 2, \ldots, n \ , \tag{5}$$

where H is a $m \times 1$ vector with the *k* elements that correspond to the disaggregates in *Z(t)* being 1's and the remaining $m-k$ elements being 0's. The unobserved *Z(t)*'s are dynamic and change in accordance with the equation

$$Z(t) = FZ(t-1) + G\xi(t), t = 1, 2, \ldots, n \ , \tag{6}$$

where F and G are $m \times m$ and $m \times k$ matrices of fixed coefficients. The term $\xi(t)$ is a $k \times 1$ vector of multivariate normal noise terms with an expected value of zero and covariance matrix Q. It is also assumed that the $\xi(t)$'s are uncorrelated for $t \neq s$. These equations are usually referred to as the observation (or measurement) equation and the system (or transition) equation, respectively.

For a simple example, let us assume that given our *y(t)* series we have reason to believe that our *x(i)* series can be represented by an AR(1) model, $x_i = \phi x_{i-1} + \varepsilon_i$. The state space representation for an AR(1) series is very simple with $H = 1$, $F = \phi$, and $G = 1$. If we aggregate by two, such that $y_t = x_{2t-1} + x_{2t}$, then our state space model for the aggregation can be formed from

$$X_{tk} = A^k X_{(t-1)k} + \sum_{i=0}^{k-1} (A^i C)\varepsilon_{tk-i} \tag{7}$$

and

$$Z(t) = X_{tk} = FX_{(t-1)k} + G\xi_t \ , \tag{8}$$

where A is a 2×2 matrix formed from the state space representation of the basic model with the other elements picking up the disaggregates in time-step *t* with $A^0 = I_{2 \times 2}$, the 2×2 identity matrix. C is an 2×1 vector with elements $\begin{bmatrix} 1 & 0 \end{bmatrix}'$, G is a 2×2 matrix of constants obtained by stacking the columns of $A^i C$ for $i = 0, 1, \ldots, k-1$ and $\xi_t = (\varepsilon_{tk}, \varepsilon_{tk-1}, \ldots, \varepsilon_{(t-1)k+1})'$. So, for our example

$$A = \begin{bmatrix} \phi & 0 \\ 1 & 0 \end{bmatrix} \tag{9}$$

and

$$X_{2t} = \begin{bmatrix} x_{2t} \\ x_{2t-1} \end{bmatrix}, \tag{10}$$

giving a system equation for Z(t) of

$$Z(t) = X_{2t} = \begin{bmatrix} \phi^2 & 0 \\ \phi & 0 \end{bmatrix} \cdot \begin{bmatrix} x_{2t-2} \\ x_{2t-3} \end{bmatrix} + \begin{bmatrix} 1 & \phi \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \varepsilon_{2t} \\ \varepsilon_{2t-1} \end{bmatrix}. \tag{11}$$

Our observation Equation (5) will have $H = \begin{bmatrix} 1 & 1 \end{bmatrix}'$ because it corresponds to the disaggregates. A generalized formulation for all state space models for disaggregates can be found in Ref. 4.

To estimate the disaggregate series, let us assume that all matrices in Equation (6) are known as well as the covariance matrix Q. Under the assumption of a linear system and Gaussian white noise components, we can compute *Z(t|t)* and its covariance matrix *P(t|t)* by a recursive algorithm known as the Kalman filter. We slightly modified the computational steps set forth in Ref. 4 for our use as follows.

(1) Obtain the starting estimates of $\hat{Z}_{t|t}$, $P_{0|0}$ and estimates of matrices F and Q.

(2) Compute the projected covariance matrix

$$P_{t|t-1} = FP_{t-1|t-1}F' + GQG'. \tag{12}$$

(3) Compute the estimated variance

$$f_t = H'P_{t|t-1}H. \tag{13}$$

(4) Compute the filter gain matrix

$$K_t = P_{t|t-1}H(f_t)^{-1}. \tag{14}$$

(5) Compute the state vector estimate using the previous estimate and the observation at time t

$$\hat{Z}_{t|t} = F\hat{Z}_{t-1|t-1} + K_t \cdot (y_t - H'F\hat{Z}_{t-1|t-1}) \ . \tag{15}$$

(6) Compute the error covariance matrix for $\hat{Z}_{t|t}$

$$P_{t|t} = P_{t|t-1} - K_t H' P_{t|t-1} \ . \tag{16}$$

(7) Set $t = t+1$ and return to step (2).

To find the estimate of the parameters in F, we minimized the likelihood function

$$L(F|Y^n) = \sum_{t=1}^{n} \log f_t + \sum_{t=1}^{n} \frac{(y_t - H'F\hat{Z}_{t-1|t-1})^2}{f_t} \tag{17}$$

using the recursion Equations (12)–(16) given above and a minimization search routine based on the golden section search method. We rewrote the likelihood equation as a function of the ϕ's and θ's for this procedure. Note that the f_i's and the $\hat{Z}_t$'s depend on each model so they must be computed each time the parameter is changed. We assumed that the covariance matrix was known in this work. After obtaining the estimated parameters for the F matrix, the disaggregates are obtained from the values of $\hat{Z}_{t|t}$ found using the recursion with this estimated F matrix.

One of the major challenges with this method is obtaining the correct model for the underlying series *x(i)* when we see only the aggregated series *y(t)* because aggregation changes the model in most cases. There is discussion and guidelines for this in the literature, but further work still needs to be done. Additionally, after a candidate model is obtained, the correct state space formulation for aggregation is also of the utmost importance for the success of this method. Much care must be taken to assure that the state space representation, as well as the likelihood formulation, is correct.

3. Performance Measures

In most instances, the goodness of fit of a statistical model to a data set is evaluated by comparing the observed values with the corresponding fitted values obtained from the model. If the fitted model is suitable, then the difference in the observed values and the fitted values, referred to as residuals, should be acceptably small and exhibit no patterns. However, we have an unusual situation here in which it is unlikely that we can remove the serial correlation in our residual sequences, as we will explain at the end of this section. Also, in our case we have several features in the true series that we would like to preserve in the estimated series, as we now describe.

To measure the performance of the naive method and the dynamic linear model method, we defined three types of goodness of fit. We then used two summary statistics for the first two goodness of fit measures and one summary statistic for the third measure. The three goodness of fit measures we defined are (1) nearness of the estimated *X* values to the true *X* values, (2) nearness of the estimated autocorrelation function of the estimated *X* values to that of the true *X* values, and

(3) nearness of the estimated parameters in a linear model relating the estimated X values to an observed sequence Z to that of the estimated parameters in a linear model relating the true X sequence to Z sequence. We will explain each of the three measures in more detail.

Let the original series before aggregation be denoted by X and the **estimated** values for our two methods be denoted by Xnaive and Xdlm, respectively.

For summary statistics for the first two methods, we used both the average squared residual difference and the average absolute residual difference in residual.

The first performance measure is the most obvious one:

$$pm1_{(mean,square)} = mean((X_i - \hat{X}_j)^2), \tag{18}$$

where $\hat{X}_j$ is either the naive or the dlm estimate of X_j, and the average is over the length of the original series X.

Also, we define

$$pm1_{(mean,absolute)} = mean|X_i - \hat{X}_j|, \tag{19}$$

again where the average is taken over the indices of the original series. Because we performed several simulations, we also average over simulations when appropriate.

Concerning the second performance measure, it is also very important for our estimated series to "model" the original series. In other words, if the underlying series is an AR(1) process, then the estimated series $\hat{Z}_{t|t}$ should also be an AR(1) series. Because the estimated autocorrelation function (acf) provides clues about the amount of differencing needed in an ARIMA process as well as the proper choice of orders for p and q for the autoregressive and moving average parameters, respectively, the difference in the autocorrelation functions was used as a performance measure. The equations are

$$pm2_{(mean,square)} = mean((acf(X) - acf(\hat{X}))^2) \tag{20}$$

and

$$pm2_{(mean,absolute)} = mean|acf(X) - acf(\hat{X})|, \tag{21}$$

where the average was taken over the number of lags we considered. The number of meaningful lags in the acf was only about 10, so the average is over lags 1 to 10.

This third measure was defined as follows. We simulated the Z series according to

$$Z = aX + b + error, \tag{22}$$

where the error was normally distributed with mean 0 and variance 1, $a = 3$, and $b = 1$. We then used both the X series and the estimated X series (Xnaive and Xdlm) to estimate a and b. For each simulated data set we get one estimate of a and b from each method.

The third performance measure had two parts, one for the estimated slope a and one for the estimated intercept b. So we defined

$$pm3.a_{(mean,square)} = (\hat{a}_{naive} - a)^2 \tag{23}$$

and

$$pm3.b_{(mean,square)} = (\hat{b}_{naive} - b)^2 \tag{24}$$

for the naive estimator, and similarly for the dlm estimator, and where in this case there is only one estimated a and b for each data set so there is no need to compute an average. However, we do average over simulations when appropriate, as we report in Section 4.

The fourth performance measure is simply the nearness of the estimated MA(1) or AR(1) coefficient to the true coefficient

$$pm4_{(mean,square)} = (\hat{\theta}_{naive} - \theta)^2 \tag{25}$$

for the MA(1) case, and similarly for the AR(1) case.

In some sense, the “best” method should have the smallest performance measure for all four performance measures.

4. Performance Results on Simulated Data

Individual AR(1) and MA(1) series were simulated with varying sample sizes and parameters to test the methods. The innovations variance for all of the simulated series was equal to one. Series were simulated with different sample sizes to see whether sample size was a factor in the performance of the methods. Original series with sample sizes of 20, 50, 100, 1000, and 2000 were simulated and then aggregated by two to obtain series of sizes 10, 25, 50, 500, and 1000. It was thought that the small sample sizes may give poor results in the dynamic linear model method. Also, series with both positive and negative parameters were simulated because the AR(1) series with negative parameters are much smoother after aggregation, making disaggregation of these series more difficult. For the AR(1) model, the AR coefficient ranged from -0.85 to 0.85 in steps of 0.2 and for the MA(1) model, the MA coefficient ranged from -0.85 to 0.85 in steps of 0.2. We performed 10 simulations for each sample size and parameter value. Because we only performed 10 simulations (computations are slow for our current implementation of the dlm), we computed paired t-tests between the calculated performance measures to see if there was any statistical difference between the two methods in the 10 simulations. A paired difference t-test was used because the individual performance measures were computed using the original series X with both Xnaive and Xdlm, so the independence criteria for two sample t-tests was violated. The differences in the paired performance measures, i.e., $pm1.naive - pm1.dlm$, are used to obtain the statistic which has a student's t-distribution with $n - 1$ degrees of freedom, where n is the number of *pairs* of observations. Consider the paired t-test to be one way to assess the significance of the differences when a relatively small number of simulations is run. A summary of some of our simulations can be found in Appendix A.

The tables in Appendix A give in columns 1-4: the average estimated coefficient values, the standard deviation of the estimated coefficient values, the range of the estimated coefficient values, and the average estimated variance of the innovations process for Xnaive and Xdlm for the 10 simulations. Column 5 is the average sum of squared residuals (performance measure one). Column 6 is the average squared distance between the acf of the true series X and the acf of the estimated series $\hat{X}$ (performance measure one). Columns 7–9 involve performance measure three. Columns 7 and 8 are the means for the estimated coefficients a and b for the regression model, and column 9 gives the mean of the estimated variances from the regression model. The p-values from the paired t-tests are displayed in the line below. We consider a t-test "significant" if the $p - value \leq 0.05$. Thus, a $p - value \leq 0.05$ tells us that there appears to be a significant difference between Xnaive and Xdlm for that performance measure. Even though the mean values are very close in many instances, the paired t-test shows that there is a statistical difference between all the individual pairs in the 10 simulations. All values in Appendix A are rounded to two decimal places. Also, to conserve space, we have not displayed the performance measures that use the mean absolute deviation. For our purposes here it is sufficient to consider the performance measures that use the mean squared deviation. However, in Appendix C we do include some of the results for performance measures that use the absolute deviations.

4.1 Performance Measure One

Performance measure one is the "closeness" of the estimated series to the true series. The dlm is nearly always either closer to the true series or not farther from the true series than is the naive method. The exception is for negative coefficients in the AR(1) case with small sample size (see Tables 3 and 6 for this exception).

4.2 Performance Measure Two

Performance measure two is the "closeness" of the acf of the estimated series to that of the true series. The dlm method consistently is "nearer" in terms of estimated and actual acf than is the naive method. The exceptions appear to be explainable by chance in the case that there is no meaningful difference in the "true" performance measure two.

4.3 Performance Measure Three

Performance measure three measures the ability of the estimated series to estimate the slope, intercept, and residual variance between the original series X and the $Z = aX + b + error$ series.

The naive and the Kalman methods perform quite well and very nearly equally in terms of estimating the slope a and the intercept b. Both methods fail miserably at estimating the variance of the error term, though the dlm method does significantly better than the naive method. We are currently investigating the reason for this poor performance (true value of error variance is 1 and typical estimate is about 10).

It is worth noting that the parameters are estimated quite well for all of the MA(1) series with the Kalman method except for the case where $n \leq 100$. The innovations variance is not estimated very accurately by either the dynamic linear model method or the naive method for these series.

The AR(1) models gave similar results. The dynamic linear model method consistently did better than the naive method, except occasionally when $n \leq 100$, with our first four performance measures. See Appendix A for an example. Once again, none of the series showed a significant difference in the estimation of the coefficients in the regression performance measure. The parameter values for the underlying models are estimated quite well for series with $n \geq 100$ and above. The dynamic linear model method seems to do a better job of estimating the innovations variance than the naive method does in all of these series, although neither do really well. (Fifteen simulations for each sample size with various negative parameter values were done to determine whether simulation size may be a factor here. Results similar to those stated were obtained.)

4.4 Performance Measure Four

The naive method is not designed to be good at estimating the MA or AR coefficient. For example, if the true model were MA(1) with a coefficient nearly equal to 0, then the original data behaves almost like independent and identically distributed observations from a Gaussian distribution. However, the naive method would introduce strong serial correlation because successive estimates are identical. And, in our simulations, the dynamic linear model method consistently performed better than the naive method in all of the moving average cases and in all of the autoregressive cases. It was a surprise, however, to observe the good performance of the dlm method for $n \geq 100$ and fairly good performance even for the $n = 20$ case. The dlm method estimates the AR(1) or MA(1) coefficient very well. It should be noted that we could also estimate these coefficients using an algebraic relation between the coefficient of the aggregated series and the original series. We have done this for the MA(1) case and can estimate the MA(1) parameter reasonably well, though not as well as the dlm method can. Also, the tables clearly indicate that the variance of the dlm-based estimate of the coefficients depends on the true coefficient as well as on the sample size. The standard deviation (column 2) for the dlm method for $\theta = 0.65$ is 0.20, while that for $\theta = 0.65$ is 0.34.

The dlm method estimates the innovations variance better than the naive method does in all of these series although both methods underestimate the variance. The true variance is 1 and the estimated variance is typically about 0.5 or less. Fifteen simulations for each sample size with various negative parameter values were done to determine whether simulation size may be a factor. Results similar to those stated were obtained.

5. Performance Results in Presence of Model Departure

Because the dlm relies heavily on the assumed ARIMA model, it is important to study how much model departure can be tolerated. We have begun an investigation of the sensitivity to model departure by simulating two models: (1) a random coefficient AR model and (2) a random coefficient MA model (the AR or MA coefficients are random variables having a long-term stationary mean and variance). In both cases we treat the data as if it arose from non-random coefficient AR and MA models. Our interest here is not in developing disaggregation methods for random coefficient AR or MA models, but in studying how much model departure our method for standard AR or MA models can tolerate.

In the results reported in Appendix B, we considered random coefficient MA(1) ($x_i = \theta_1 \varepsilon_{i-1} + \varepsilon_i$, with θ_1 random) and random coefficient AR(1) ($x_i = \phi_1 x_{i-1} + \varepsilon_i$ with ϕ_1 random) models. We used the same average values for the coefficients as we did for the results described in Section 4 (the MA and AR coefficients ranged from nearly -1 to nearly 1 in steps of 0.2), and at each time step we generated a new (independent from all previous values) normally distributed coefficient. We used two variances for the normal distribution of the random coefficients: 0.1 and 1.0 for all cases.

By comparing the tables in Appendix B to the appropriate tables in Appendix A (for example, compare Table 7 in Appendix B to Table 1 in Appendix A), we see that the dlm results for the lower variance case (.1) are quite similar to those for the ordinary AR and MA results, but that the results for the higher variance case (1.0) are quite different from those for the ordinary AR and MA results. Qualitatively, we can say that for the cases considered, it is not possible to use performance on the ordinary AR and MA models to predict performance on the random coefficient AR or MA models if the variance of the random coefficients is as large as 1.0. It would be useful to include additional cases in future work.

6. Performance Results on Real Data

Two of the series which motivated this work were counts from satellite-based gamma detectors. These gamma counts were reasonably well fit by an ARIMA(0,1,1) model.[7] Both series contained approximately 1800 counts before aggregation. The counts in the individual series were summed by two to form the aggregated series. As before, the estimates obtained for the *x(i)* series from the two methods were compared with the original series to obtain the performance measures.

The first series (gamma1) before aggregation was fit with an ARIMA(0,1,1) model where θ was estimated to be -0.97 and the variance of the innovations, ε, was estimated to be 87.37. The dynamic linear model method estimated θ to be -0.97 and the estimated innovations variance was 39.78. A coefficient for the naive method was estimated to be 1.0. The innovation variance for this method was estimated at 14.48. The dynamic linear model method gave lower estimates on all of the performance measures. There was an extremely large difference in the variance of the regression models. Results are displayed in Appendix C, and Figure 1 shows the first 50 counts and the estimates of those counts using both the naive and the dlm methods. We see in Figure 1 that the dlm method tends to produce estimates that are closer to *X* than does the naive method.

The gamma2 series was originally fit with an ARIMA(0,1,1) model where θ was estimated to be -0.93 with an innovations variance of 126.76. The estimate for the dynamic linear model method was $\theta = -0.93$ with an innovations variance of 65.12. Once again the naive method estimated θ to be 1.0. The innovations variance for this model was estimated at 30.82. This is essentially the same kind of performance as with the gamma1 data set. The dynamic linear model method gave lower estimates on all of the performance measures. The estimates from the dynamic linear model method are closer to those from the original series than are the naive estimates for both of these gamma series.

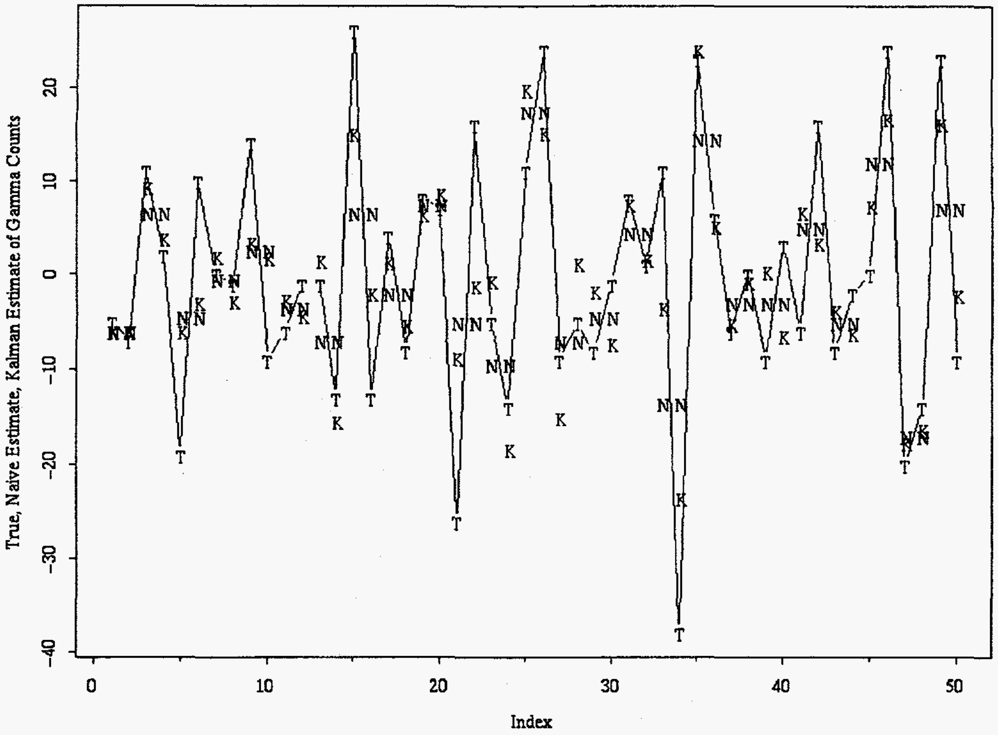

FIGURE 1. Gamma counts from gamma1 data. T is true gamma counts, N is naive estimate, and K is Kalman filter estimate.

Several other data sets, which are available with the standard release of Splus, were also analyzed. The results from three of these are also displayed in Appendix C.

The data set denoted *air* has 110 observations and is reasonably well fit by an MA(1) model where θ was estimated to at -0.39 and the innovations variance was 34.10. The dynamic linear model method gave an estimate of -0.41 for θ and 20.44 for the innovations variance. The naive method gave an estimate of 1.00 for θ and an innovations variance of 7.84. The dynamic linear model method gave lower estimates in all of the performance measures once again.

The last two data sets were reasonably well fit by AR(1) models. The data set *lh* is taken from Venables and Ripley[8] and has 48 observations of luteinizing hormone levels for a human female. The dynamic linear model method's estimate gave lower estimates for all the performance measures.

The only real data set in Appendix C in which the dynamic linear model method did not perform well is the *geyser* data set. The AR(1) parameter was not estimated well, and consequently, the dlm method did very slightly worse in estimating the original data as the naive method did (compare 11.23 to 11.39). This may be due to the fact that this data set is better fit with an AR process of order 2 than an AR process of order 1. In other words, the AR(1) fit was only marginally acceptable on the *geyser* data, so it is likely that the model departure is more than the dlm method can tolerate.

On the whole, the dynamic linear model method seems to provide fairly good estimates of the original series even in series with $n \approx 50$. Further testing needs to be done on this, however.

7. Summary

The dynamic linear model method seems to perform well in the estimation of the disaggregates when the underlying model is an MA(1) or an AR(1) in aggregated series with $n \geq 50$. Also, we have seen that small amounts of model departure seem to be tolerated reasonably well, though further work needs to be done in this area. We also plan to extend the method to multi-dimensions (higher lags) and to estimate the innovations variance for the underlying white noise series. We could also ask the following questions:

1. Why does an iterative procedure such as the one outlined in Ref. 4 fail to find the parameter values that minimize the sum of squared errors (this failure is why we resorted to a direct search for the minimizing parameters)? Is there a form of the EM algorithm that could be implemented instead of the grid search currently used to find the parameters? Define the conditions needed for an iterative procedure such as outlined in Ref. 4 for the EM algorithm to find the global minimum.
2. How is this problem related to hidden Markov models?
3. Apply the method to more real data sets in a variety of settings, especially ones in which it makes sense to aggregate the original data, then try to recover the original data. How well will this work on a wider variety of real data sets? Can we anticipate the performance on the basis of some measure of goodness of fit to the assumed ARMA model?

References

1. S. Bleasdale, T. Burr, A. Coulter, J. Doak, B. Hoffbauer, D. Martinez, J. Prommel, C. Scovel, R. Strittmatter, T. Thomas, and A. Zardecki, "Knowledge Fusion: Analysis of Vector-Based Time Series with an Example from the SABRS Project," Los Alamos National Laboratory report LA-12931-MS (April 1995).

2. G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, 3rd Ed. (Holden-Day, San Francisco, 1994).

3. C. Chatfield, *The Analysis of Time Series, An Introduction*, 4th Ed. (Chapman and Hall, London, 1989).

4. M. Al-Osh, "A Dynamic Linear Model Approach for Disaggregating Time Series Data," *Journal of Forecasting,* Vol. 8(2), 1989.

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

6. B. D. O. Anderson and J. B. Moore, *Optimal Filtering* (Prentice-Hall, 1979).

7. S. Bleasdale, T. Burr, C. Scovel, and R. Strittmatter, "An Approach to Time Series Model Selection Followed by Pattern Recognition," Los Alamos National Laboratory report LA-13095-MS (March 1996).

8. W. N. Venables and B. D. Ripley, *Modern Applied Statistics with Splus* (Springer-Verlag, 1994).

Appendix A

TABLE 1. MA(1) with θ = -0.65.

	avg	std dev	range						
n=20	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.99	0.00	.99,.99	0.13	0.80	0.34	2.97	0.98	8.72
Xdlm	-.77	0.20	-.99,-.40	0.27	0.74	0.08	2.61	0.96	7.90
P-val	0.00	NA	NA	0.00	0.31	0.00	0.26	0.12	0.14

	avg	std dev	range						
n=100	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.99	0.04	.99,.99	0.22	1.02	0.08	3.00	1.00	10.65
Xdlm	-.68	0.08	-.76,-.52	0.52	0.93	0.02	2.93	1.00	9.57
P-val	0.00	NA	NA	0.00	0.01	0.00	0.37	0.61	0.00

	avg	std dev	range						
n=1000	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	1.00	0.00	1.0,1.0	0.18	1.01	0.07	3.01	0.99	10.10
Xdlm	-.65	0.02	-.68,-.62	0.47	0.91	0.15	2.99	0.99	9.16
P-val	0.00	NA	NA	0.00	0.00	0.00	0.34	0.44	0.00

	avg	std dev	range						
n=2000	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	1.00	0.00	1.0,1.0	0.20	1.02	0.07	3.00	1.00	10.24
Xdlm	-.65	0.03	-.70,-.60	0.50	0.91	0.02	3.01	1.00	9.20
P-val	0.00	NA	NA	0.00	0.00	0.00	0.77	0.80	0.00

Column 1: Average (over all simulations) of $\hat{\theta}$. Column 2: Standard deviation of θ's over all simulations. Column 3: Range of θ's over all simulations. Column 4: Average Estimate of innovations variance (true is 1). Column 5: Average rss between *XHAT* and *X*. Column 6: Average rss between acf (lags 1–10 only) of *XHAT* and acf of *X*. Columns 7, 8, and 9: Average of estimated intercept, slope, and residual variance in linear model, $Z = aX + b + error$. True values are 3, 1, and 1, respectively.

TABLE 2. MA(1) with $\theta = 0.65$.

	avg	std dev	range						
n=20	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.98	0.07	.78,1.0	0.50	0.47	0.02	2.99	1.00	5.74
Xdlm	0.65	0.34	-.17,.99	0.55	0.46	0.03	2.97	1.01	5.59
P-val	0.02	NA	NA	0.12	0.55	0.01	0.69	0.65	0.54

	avg	std dev	range						
n=100	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.97	0.04	.88,1.0	0.53	0.37	0.01	3.00	1.04	4.42
Xdlm	0.65	0.21	.20,.94	0.61	0.37	0.01	2.97	1.04	4.42
P-val	0.00	NA	NA	0.00	0.66	0.07	0.01	0.63	0.00

	avg	std dev	range						
n=1000	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	1.0	0.00	.99,1.0	0.53	0.40	0.00	2.98	1.0	4.67
Xdlm	0.66	0.05	.58,.73	0.61	0.40	0.00	2.98	1.0	4.44
P-val	0.00	NA	NA	0.00	0.00	0.00	0.52	0.92	0.00

	avg	std dev	range						
n=2000	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	1.0	0.00	.99,1.0	0.51	0.38	0.00	3.01	1.01	4.44
Xdlm	0.63	0.05	.56,.71	0.60	0.35	0.00	3.01	1.01	4.21
P-val	0.00	NA	NA	0.00	0.00	0.00	0.79	0.08	0.00

TABLE 3. AR(1) with θ = -0.65.

	avg	std dev	range						
n=20	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.47	0.15	.31,.76	0.22	1.45	0.53	2.98	0.89	14.90
Xdlm	-.68	0.36	-.99,-.09	0.38	1.44	0.19	2.98	0.85	13.85
P-val	0.00	NA	NA	0.00	0.04	0.01	0.05	0.66	0.05

	avg	std dev	range						
n=100	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.43	0.07	.33,.54	0.26	1.60	0.18	3.00	0.99	15.70
Xdlm	-.59	0.21	-.97,-.28	0.51	1.66	0.11	2.71	0.99	14.98
P-val	0.00	NA	NA	0.00	0.69	0.03	0.39	0.98	0.08

	avg	std dev	range						
n=1000	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.45	0.02	.41,.48	0.25	1.46	0.15	3.01	1.00	14.11
Xdlm	-.63	0.15	-.83,-.42	0.54	1.35	0.05	2.90	1.00	13.04
P-val	0.00	NA	NA	0.00	0.00	0.00	0.50	0.91	0.00

	avg	std dev	range						
n=2000	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	.44	0.01	.42,.47	0.22	1.42	0.14	3.01	1.01	13.88
Xdlm	-.66	0.08	-.78,-.54	0.54	1.31	0.04	2.91	1.01	12.87
P-val	0.00	NA	NA	0.00	0.00	0.00	0.19	0.73	0.00

TABLE 4. AR(1) with θ = 0.65.

	avg	std dev	range						
n=20	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.75	0.11	.63,.95	0.67	0.25	0.03	3.02	0.96	3.17
Xdlm	0.53	0.25	-.11,.78	0.59	0.23	0.03	3.02	0.97	3.03
P-val	0.01	NA	NA	0.07	0.24	0.02	0.74	0.54	0.49

	avg	std dev	range						
n=100	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.79	0.07	.66,.90	0.55	0.26	0.00	3.02	1.01	3.42
Xdlm	0.67	0.09	.53,.84	0.49	0.24	0.00	3.01	1.01	3.20
P-val	0.00	NA	NA	0.00	0.02	0.59	0.72	0.92	0.01

	avg	std dev	range						
n=1000	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.77	0.02	.75,.79	0.58	0.30	0.00	3.02	0.98	3.79
Xdlm	0.65	0.02	.61,.67	0.54	0.28	0.00	3.02	0.98	3.56
P-val	0.00	NA	NA	0.00	0.00	0.00	0.89	0.79	0.00

	avg	std dev	range						
n=2000	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.76	0.01	.74,.78	0.58	0.30	0.00	3.00	1.00	3.67
Xdlm	0.64	0.02	.61,.67	0.54	0.27	0.00	3.00	1.00	3.44
P-val	0.00	NA	NA	0.00	0.00	0.07	0.93	0.64	0.00

TABLE 5. MA(1) with θ = -0.85, -0.45, -0.25, 0.25, 0.45, 0.85 and n = 100.

	avg	std dev	range						
-0.85	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	1.0	0.00	1.0,1.0	0.17	1.19	0.11	2.97	1.00	11.83
Xdlm	-.84	0.08	-.99,-.73	0.38	0.95	0.02	3.07	1.00	9.59
P-val	0.00	NA	NA	0.00	0.00	0.00	0.45	0.17	0.00

	avg	std dev	range						
-0.45	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	1.0	0.01	.97,1.0	0.18	0.84	0.08	3.03	0.98	8.64
Xdlm	0.52	0.17	-.80,-.25	0.45	0.81	0.03	2.92	0.98	8.28
P-val	0.00	NA	NA	0.00	0.06	0.00	0.32	0.88	0.02

	avg	std dev	range						
-0.25	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.99	0.02	.95,1.0	0.19	0.66	0.07	2.95	0.95	6.90
Xdlm	0.20	0.15	-.43,.09	0.44	0.64	0.05	2.97	0.95	6.78
P-val	0.00	NA	NA	0.00	0.12	0.01	0.59	0.67	0.14

	avg	std dev	range						
0.25	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.99	0.02	.94,1.0	0.35	0.38	0.01	2.99	0.98	4.36
Xdlm	0.28	0.26	0.00,.70	0.52	0.38	0.01	2.98	0.99	4.36
P-val	0.00	NA	NA	0.00	0.88	0.01	0.51	0.09	0.98

	avg	std dev	range						
0.45	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.98	0.04	.87,1.0	0.40	0.36	0.01	2.92	1.02	4.28
Xdlm	0.41	0.18	0.03,.66	0.53	0.35	0.01	2.91	1.02	4.17
P-val	0.00	NA	NA	0.00	0.16	0.07	0.60	0.49	0.06

	avg	std dev	range						
0.85	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.97	0.03	.94,1.0	0.65	0.48	0.01	3.02	1.04	5.40
Xdlm	0.82	0.17	.55,.99	0.70	0.44	0.01	3.02	1.04	5.05
P-val	0.00	NA	NA	0.01	0.00	0.24	0.83	0.46	0.00

TABLE 6. AR(1) with θ = -0.85, -0.45, -0.25, 0.25, 0.45, 0.85 and n = 100.

	avg	std dev	range						
-0.85	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.49	0.07	.39,.61	0.20	3.78	0.39	2.99	1.05	35.94
Xdlm	-.85	0.12	-.99,-.63	0.52	4.21	0.11	2.73	1.03	29.62
P-val	0.00	NA	NA	0.00	0.70	0.00	0.55	0.44	0.01

	avg	std dev	range						
-0.45	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.44	0.10	.29,.59	0.28	0.83	0.09	3.00	0.98	8.76
Xdlm	-.46	0.20	-.85,-.11	0.52	0.83	0.06	2.77	0.97	8.63
P-val	0.00	NA	NA	0.00	0.97	0.00	0.18	0.39	0.44

	avg	std dev	range						
-0.25	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.48	0.07	.37,.56	0.30	0.60	0.06	2.97	0.97	6.54
Xdlm	-.32	0.26	-.95,-.02	0.50	0.64	0.05	2.76	0.98	6.59
P-val	0.00	NA	NA	0.00	0.34	0.10	0.25	0.29	0.67

	avg	std dev	range						
0.25	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.59	0.06	.49,.68	0.42	0.40	0.21	2.99	0.96	4.54
Xdlm	0.21	0.15	-.05,.42	0.48	0.40	0.22	2.99	0.96	4.51
P-val	0.00	NA	NA	0.00	0.10	0.00	0.59	0.03	0.39

	avg	std dev	range						
0.45	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.67	0.05	.60,.75	0.53	0.34	0.01	2.97	0.94	4.15
Xdlm	0.47	0.09	.34,.61	0.53	0.33	0.01	2.96	0.94	4.05
P-val	0.00	NA	NA	0.46	0.17	0.00	0.62	0.15	0.28

	avg	std dev	range						
0.85	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.89	0.04	.83,.95	0.66	0.28	0.00	3.02	0.98	3.56
Xdlm	0.85	0.04	.78,.92	0.55	0.24	0.00	3.02	0.97	3.23
P-val	0.00	NA	NA	0.00	0.00	0.00	0.91	0.18	0.00

Appendix B

Results with a random coefficient MA(1) model treated as if it were an ordinary MA(1) model.

TABLE 7. MA(1) with θ ~ *N*(-0.65, 0.1).

n=20	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	1.0	0.00	1.0,1.0	0.20	0.97	0.35	3.10	1.04	10.88
Xdlm	-.67	0.29	-.99,-.12	0.37	0.90	0.13	2.91	1.03	10.15
P-val	0.00	NA	NA	0.00	0.24	0.00	0.00	0.33	0.22

n=100	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	1.0	0.01	.97,1.0	0.23	1.07	0.08	2.94	1.04	10.72
Xdlm	-.60	0.09	-.73,-.44	0.56	0.95	0.03	3.02	1.04	9.49
P-val	0.00	NA	NA	0.00	0.00	0.00	0.25	0.23	0.00

TABLE 8. MA(1) with θ ~ *N*(-0.65, 1.0).

n=20	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.98	0.05	.87,1.0	0.48	1.54	0.28	3.13	1.11	17.37
Xdlm	0.12	0.81	-.99,.99	0.71	1.57	0.21	2.88	1.08	17.77
P-val	0.00	NA	NA	0.00	0.72	0.23	0.07	0.04	0.58

n=100	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.99	0.12	.96,1.0	0.44	1.62	0.07	2.98	1.02	15.76
Xdlm	0.43	0.39	-.31,.84	0.62	1.67	0.08	2.88	1.02	16.15
P-val	0.00	NA	NA	0.00	0.07	0.22	0.02	0.83	0.08

Column 1: Average (over all simulations) of $\hat{\theta}$. Column 2: Standard deviation of θ' s over all simulations. Column 3: Range of θ' s over all simulations. Column 4: Average estimate of innovations variance (true is 1). Column 5: Average rss between *XHAT* and *X*. Column 6: Average rss between acf (lags 1-10 only) of *XHAT* and acf of *X*. Columns 7, 8, and 9: Average of estimated intercept, slope, and residual variance in linear model, $Z = aX + b + error$. True values are 3, 1, and 1, respectively.

TABLE 9. MA(1) with $\theta \sim N(0.65, 0.1)$.

n=20	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.96	0.07	.82,1.0	0.41	0.45	0.05	3.14	1.12	5.23
Xdlm	0.40	0.50	-.50,.99	0.52	0.47	0.06	2.98	1.12	5.51
P-val	0.01	NA	NA	0.04	0.40	0.19	0.03	0.78	0.21

n=100	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.97	0.03	.91,1.0	0.57	0.43	0.01	2.95	1.00	5.19
Xdlm	0.74	0.14	.52,.99	0.63	0.41	0.01	2.93	1.01	5.09
P-val	0.00	NA	NA	0.00	0.18	0.23	0.22	0.24	0.42

TABLE 10. MA(1) with $\theta \sim N(0.65, 1.0)$.

n=20	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.93	0.11	.71,1.0	0.59	0.77	0.08	2.97	1.03	8.59
Xdlm	0.65	0.45	-.44,.99	0.74	0.79	0.08	2.84	1.06	8.69
P-val	0.07	NA	NA	0.01	0.75	0.81	0.15	0.29	0.85

n=100	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.99	0.04	.89,1.0	0.72	1.01	0.02	3.00	0.95	10.09
Xdlm	0.91	0.15	.62,.99	0.78	0.95	0.02	3.02	0.95	9.50
P-val	0.17	NA	NA	0.00	0.01	0.01	0.39	0.49	0.01

Results with a random coefficient AR(1) model treated as if it were an ordinary AR(1) model.

TABLE 11. AR(1) with θ ~ *N*(-0.65, 0.1).

	avg	std dev	range						
n=20	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.51	0.11	.35,.66	0.27	1.49	0.47	3.07	1.04	16.5
Xdlm	-.55	0.46	-.99,.33	0.44	1.58	0.20	2.46	1.09	16.3
P-val	0.00	NA	NA	0.00	0.46	0.16	0.09	0.33	0.69

	avg	std dev	range						
n=100	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.45	0.05	.36,.54	0.31	1.76	0.17	3.02	1.04	16.76
Xdlm	-.37	0.29	-.96,-.06	0.52	1.68	0.12	3.02	1.04	15.92
P-val	0.00	NA	NA	0.00	0.03	0.15	0.96	0.95	0.03

TABLE 12. AR(1) with θ ~ *N*(-0.65, 1.0).

	avg	std dev	range						
n=20	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.55	0.20	.20,.93	113.91	198.92	0.27	3.07	0.87	1986.74
Xdlm	0.30	0.60	-.97,.95	109.48	212.21	0.23	2.77	0.71	2112.83
P-val	0.16	NA	NA	0.33	0.34	0.56	0.20	0.25	0.34

	avg	std dev	range						
n=100	**coeff**	**coeff**	**coeffs**	**avg var**	**avg rss**	**avg acf**	**coeff a**	**coeff b**	**var reg**
Xnaive	0.42	0.10	.31,.55	7.58	18.49	0.11	2.99	0.97	171.41
Xdlm	0.38	0.19	0.65	7.12	20.50	0.13	2.83	1.00	188.42
P-val	0.48	NA	NA	0.20	0.16	0.00	0.08	0.23	0.16

TABLE 13. AR(1) with θ ~ $N(0.65, 0.1)$.

n=20	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.71	0.14	.51,.88	0.68	0.41	0.02	2.91	0.90	4.67
Xdlm	0.51	0.27	0,.84	0.59	0.38	0.02	2.91	0.90	4.44
P-val	0.00	NA	NA	0.04	0.09	0.01	0.94	0.88	0.18

n=100	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.75	0.04	.70,.84	0.61	0.36	0.01	3.00	1.02	4.26
Xdlm	0.61	0.11	.44,.74	0.56	0.33	0.01	3.00	1.01	4.03
P-val	0.00	NA	NA	0.02	0.02	0.05	0.90	0.59	0.04

TABLE 14. AR(1) with θ ~ $N(0.65, 1.0)$.

n=20	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.68	0.15	.43,.90	2.78	2.14	0.05	2.97	1.08	21.2
Xdlm	0.60	0.21	.33,.90	2.26	1.99	0.07	2.98	1.08	19.8
P-val	0.12	NA	NA	0.21	0.23	0.00	0.72	0.85	0.21

n=100	avg coeff	std dev coeff	range coeffs	avg var	avg rss	avg acf	coeff a	coeff b	var reg
Xnaive	0.61	0.14	.44,.88	8.88	3.48	0.01	3.00	0.99	32.83
Xdlm	0.67	0.12	.55,.91	8.05	3.85	0.01	2.92	0.99	36.07
P-val	0.00	NA	NA	0.01	0.12	0.01	0.00	0.74	0.12

Appendix C

TABLE 15. Summary of Results on Real Data.

gam1	coeff	var	avg rss	abs rss	avg acf	abs acf	coeff a	coeff b	var reg
Xnaive	1.0	14.48	127.15	9.02	0.08	0.16	2.99	1.01	1147.13
Xdlm	-.97	39.78	93.96	7.80	0.00	0.02	2.93	1.00	84.75
X	-.97	87.47	0	0	0	0	3.00	1.01	1.07

gam2	coeff	var	avg rss	abs rss	avg acf	abs acf	coeff a	coeff b	var reg
Xnaive	1.0	30.82	173.31	10.26	0.08	0.16	3.00	0.96	1562.02
Xdlm	-.93	65.12	125.46	8.77	0.00	0.03	2.96	0.96	1130.11
X	-.93	126.76	0	0	0	0	3.00	0.96	1.06

air	coeff	var	avg rss	abs rss	avg acf	abs acf	coeff a	coeff b	var reg
Xnaive	1.00	7.84	22.85	3.48	0.06	0.16	3.00	1.08	214.43
Xdlm	-.41	20.44	20.67	3.24	0.03	0.13	3.09	1.08	193.87
X	-.39	34.10	0	0	0	0	3.02	1.08	1.01

lh	coeff	var	avg rss	abs rss	avg acf	abs acf	coeff a	coeff b	var reg
Xnaive	0.65	0.14	0.06	0.18	0.01	0.08	3.25	0.96	1.71
Xdlm	0.52	0.13	0.05	0.15	0.01	0.07	3.29	0.96	1.58
X	0.58	0.21	0	0	0	0	3.25	0.96	1.02

geyser	coeff	var	avg rss	abs rss	avg acf	abs acf	coeff a	coeff b	var reg
Xnaive	1.00	25.06	167.62	11.23	0.23	0.37	3.00	0.97	1520.65
Xdlm	0.99	24.5	173.31	11.39	0.24	0.37	2.58	3.11	1567.54
X	-.70	97.86	0	0	0	0	3.00	0.91	0.94

Note: The first three data sets are acceptably well fit with MA(1) models and have n = 1770, 1770, and 100, respectively. The last two data sets are acceptably well fit with AR(1) models and have n = 48 and 299, respectively. The geyser data set is better fit by an order 2 AR process, but the fit with order 1 is acceptable.