R-package `armiss`: A Tutorial

William Weimin Yoo

January 29, 2013

1 Introduction

Missing values are a common phenomenon in data analysis, and is particular so for data collected over time. Time series with missing observations are irregularly spaced, and as a result standard estimation methods in time series analysis are not directly applicable. The base `R` package `stats` contains many functions that will do parameter estimation if we are willing to assume certain parametric models for our time series data. For example, the `ar.mle` function fits an autoregressive (AR) process using maximum likelihood where the order is selected using the Akaike Information Criterion (AIC). However by default, these functions do not accept data with missing values. There are various methods proposed in the literature to deal with missing values in time series analysis. However, most of these methods are computationally expensive and are not viable for long data series. Therefore, the author and Dr. Sujit Ghosh have proposed a simple and effective method to deal with this problem. This method first reorders the series into the observed and missing parts respectively using a permutation matrix, missing data are then imputed using the conditional Gaussian distribution based on the observed data. The reconstructed series is then used to do parameter estimation. An `R` package called `armiss` was written for this proposed methodology, and is based on the working paper titled "Gaussian Imputation of an ARMA Process with Missing Values".

2 Imputation via conditional Gaussian distribution

We will give a brief overview of our proposed imputation method. Given a time series $\{X_t, t = 1, 2, \ldots, T\}$, suppose we only get to observe a subset of this series due to missing values. Specifically, for some indices $1 \leq t_1 \leq t_2 \leq \ldots \leq t_n \leq T$, we observed $\{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\}$. We first assume that there is an underlying process $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_T)'$ that generated the observed series $\{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\}$. That is for $t \in \{t_1, t_2, \ldots, t_n\}$, we have $Y_t \overset{d}{=} X_t$ with $\overset{d}{=}$ denoting equal in distribution. While for $t \notin \{t_1, t_2, \ldots, t_n\}$, we do not observe Y_t due to missing values. For simplicity, we assume that $\mathbf{Y}$ is generated from an autoregressive process of some order p, denoted as AR(p),

$$Y_t = \mu + \sum_{i=1}^{\min\{p,t-1\}} \phi_i (Y_{t-i} - \mu) + \epsilon_t, \tag{2.1}$$

where $\mu = \mathrm{E}(Y_t)$ is the process mean. Also, we assume that $\{\epsilon_t\}$ are independent and identically (i.i.d) normally distributed with mean zero and finite variance σ^2. The AR coefficients $\{\phi_i\}$ satisfy the necessary constraints for weak stationarity. Therefore, $\mathbf{Y}$ is normally distributed with mean $\mu \mathbf{1}_T$ and covariance matrix $\mathbf{\Sigma}$, where the i, j element of $\mathbf{\Sigma}$ is given by $\mathbf{\Sigma}_{ij} = \gamma(|i-j|)$, and $\gamma(h)$ is the autocovariance function at time lag h for an AR(p) process. Let $\boldsymbol{\theta} = (\mu, \phi_1, \ldots, \phi_p, \sigma^2, p)'$ be the parameter vector for this model setting. Our aim is to estimate $\boldsymbol{\theta}$ given the incomplete observations $\{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\}$.

Now, for any given $\boldsymbol{\theta}$, we begin by reordering the elements of $\mathbf{Y}$ using a permutation matrix $\mathbf{P}$, so as to partition $\mathbf{Y}$ into the vector of observed values $\mathbf{Y}_O$ and the vector of missing values $\mathbf{Y}_M$ respectively

$$\mathbf{PY} = \begin{pmatrix} \mathbf{P}_O \\ \mathbf{P}_M \end{pmatrix} \mathbf{Y} = \begin{pmatrix} \mathbf{Y}_O \\ \mathbf{Y}_M \end{pmatrix} \overset{d}{=} \begin{pmatrix} \mathbf{X}_O \\ \mathbf{Y}_M \end{pmatrix}, \tag{2.2}$$

where $\mathbf{X}_O = (X_{t_1}, X_{t_2}, \ldots, X_{t_n})'$ denotes the observed values. An `R` function `elem` was written to accomplish this task. The `elem(data, sym = NA)` function has two arguments, the first is the data itself and the second `sym` is symbol or indicator used to represent missing values.

The input for `data` argument must be a vector with numeric entries. If missing values are encoded with characters, then a `list` object may be used for the data, in which case the input for `sym` must be delimited by quotation marks, i.e. "missing". The default symbol used is NA. As an example, suppose our data is `c(1, 2, 8888, 4, 8888)` with `8888` denoting missing observations. Then this function will create the corresponding permutation matrix, P that will separate the observed and missing values. Multiplying P and `(1, 2, 8888, 4, 8888)` will give us `(1, 2, 4, 8888, 8888)`.

```
> library(armiss)
> x <- c(1, 2, 8888, 4, 8888)
> P <- elem(data = x, sym = 8888)
> P  #permutation matrix
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0    0    1    0
[4,]    0    0    1    0    0
[5,]    0    0    0    0    1
```

```
> as.vector(t(P %*% x))
```

```
[1]    1    2    4 8888 8888
```

It then follows that $\mathbf{PY}$ is normally distributed with distribution given by,

$$\mathbf{PY} \sim \mathrm{N}\left(\mu\mathbf{1}_T, \begin{bmatrix} \mathbf{P}_O\mathbf{\Sigma}\mathbf{P}_O^{'} & \mathbf{P}_O\mathbf{\Sigma}\mathbf{P}_M^{'} \\ \mathbf{P}_M\mathbf{\Sigma}\mathbf{P}_O^{'} & \mathbf{P}_M\mathbf{\Sigma}\mathbf{P}_M^{'} \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_{OO} & \mathbf{\Sigma}_{OM} \\ \mathbf{\Sigma}_{MO} & \mathbf{\Sigma}_{MM} \end{bmatrix}\right). \tag{2.3}$$

An `R` function was written to construct $\mathbf{\Sigma}$, which is the covariance matrix for an AR(p) process. The function is `covmat(phi, sigma2)` with two arguments. The input for `phi` is a vector of AR coefficients and for `sigma2` is the innovation variance σ^2. The AR coefficients must satisfy the necessary stationarity conditions. This function calls upon the `ARMAacf` function from the `stats` package that computes the autocorrelation function for a given set of AR coefficients. Continuing from the previous example above, let `phi = c(0.5, 0.2)` and `sigma2 = 1` for an AR(2) process,

```
> N <- length(x)  #data length
> phi = c(0.5, 0.2)  #AR(2)
> sigma2 = 1  #innovation variance
> covmat(phi = phi, sigma2 = sigma2, N = N)
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 1.7094017 1.0683761 0.8760684 0.6517094 0.5010684
[2,] 1.0683761 1.7094017 1.0683761 0.8760684 0.6517094
[3,] 0.8760684 1.0683761 1.7094017 1.0683761 0.8760684
[4,] 0.6517094 0.8760684 1.0683761 1.7094017 1.0683761
[5,] 0.5010684 0.6517094 0.8760684 1.0683761 1.7094017
```

Therefore, the conditional distribution of $\mathbf{Y}_M$ given $\mathbf{Y}_O$ is normally distributed, and since $\mathbf{Y}_O$ and $\mathbf{X}_O$ are equal in distribution, we can write

$$\mathbf{Y}_M | \mathbf{X}_O \sim \mathrm{N}(\boldsymbol{\mu}_{M|O}, \boldsymbol{\Sigma}_{M|O}), \tag{2.4}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{M|O} &= \boldsymbol{\mu}_M + \boldsymbol{\Sigma}_{MO}\boldsymbol{\Sigma}_{OO}^{-1}(\mathbf{X}_O - \boldsymbol{\mu}_O) \\ \boldsymbol{\Sigma}_{M|O} &= \boldsymbol{\Sigma}_{MM} - \boldsymbol{\Sigma}_{MO}\boldsymbol{\Sigma}_{OO}^{-1}\boldsymbol{\Sigma}_{OM}, \end{aligned} \tag{2.5}$$

for $\boldsymbol{\mu}_M = \mu\mathbf{P}_M\mathbf{1}_T$ and $\boldsymbol{\mu}_O = \mu\mathbf{P}_O\mathbf{1}_T$. Based on this setup, we impute the missing values by generating samples from˜(2.4), that is, $\mathbf{Y}_O^+ \sim \mathrm{N}(\boldsymbol{\mu}_{M|O}, \boldsymbol{\Sigma}_{M|O})$. We then append the imputed values $\mathbf{Y}_O^+$ to $\mathbf{X}_O$. Applying the matrix inverse of $\mathbf{P}$ to this appended series will yield the reconstruction of $\mathbf{Y}$, which we will denote as $\mathbf{Y}^+$.

In real life applications, the parameters $\boldsymbol{\theta}$ is not known and have to be estimated from the data. Now, given time series data with missing observations, our proposed method can be summarized in an algorithm.

1. Obtain initial estimates of $\boldsymbol{\theta}$ based on $\mathbf{X}_O$ by treating $\mathbf{X}_O$ as the complete series. We used the R function ar.yw to get these estimates using Yule-Walker (methods of moments) by fitting an AR(1) process. Let $\hat{\boldsymbol{\theta}}^{(0)}$ be the initial estimates

2. Use $\hat{\boldsymbol{\theta}}^{(0)}$ to conduct the imputation method as described above to get the reconstructed series $\mathbf{Y}^+$

3. Use $\mathbf{Y}^+$ to reestimate $\boldsymbol{\theta}$ using ar.mle, which will simultaneously estimate $\boldsymbol{\theta}$ and choose the optimal order p using AIC. Update $\hat{\boldsymbol{\theta}}^{(0)}$ to $\hat{\boldsymbol{\theta}}^{(1)}$

4. Repeat steps 2 and 3 until the parameter estimates converged, where the convergence criterion is judged by (at the $k+1$ iterate),

   $$|l(\hat{\boldsymbol{\theta}}^{(k+1)}; \mathbf{X}_O) - l(\hat{\boldsymbol{\theta}}^{(k)}; \mathbf{X}_O)| < \epsilon |l(\hat{\boldsymbol{\theta}}^{(k)}; \mathbf{X}_O)|.$$

   Here, $l(\hat{\boldsymbol{\theta}}^{(k+1)}; \mathbf{X}_O)$ is the log likelihood function of $\hat{\boldsymbol{\theta}}^{(k+1)}$ evaluated at the observed data $\mathbf{X}_O$, and ϵ is a tuning parameter that controls the rate of convergence.

For an illustration, we use the data provided in the package. This data is a simulated time series from an AR(1) process $\mu = 0, \phi = 0.5, \sigma^2 = 1$ with 365 data points. Missing values are assumed to be missing at random. There are 165 randomly missing observations among the 365 data points. Missing values are encoded with NA.

```
> data(ar1sim)
> ar1sim[1:20]  #first 20 observations

 [1]          NA          NA  0.92885445          NA  0.01189859  0.50705157
 [7] -1.29052433          NA  0.30966335          NA          NA -0.51284701
[13]          NA -2.13178384          NA          NA -1.33412779  0.87742481
[19]  1.10040857  2.17439932
```

The R function ar.miss implements the proposed method/algorithm described above. The arguments of this function are ar.miss(data, epsilon = 0.001, order = NULL, max.iter

`= 100, sym = NA, control.optim = list(maxit = 200))`. The input for the first argument is the time series data. Currently, only a single time series is supported. We are now working on extending this proposed method to multiple time series. The observed part of the data must be numeric, which can be a vector or a time series (`ts`) object. However, as mentioned before, if missing values are encoded as characters, then a list might be employed. The next argument controls the parameter convergence in Step 4 of the algorithm, where the default value is 0.001. If we have some a priori knowledge of the process order, then we can specify it in the `order` argument. In this case, the AIC selection procedure in Step 3 is skipped.

The `max.iter` argument controls the number of iteration of the algorithm, and has an upper limit of 100 by default. The iteration for the proposed algorithm terminates when the convergence criterion is satisfied. However, in some series that resemble a unit root process, it takes a substantial amount of iteration (> 100) to reach convergence. Hence, this is put in place to ensure reasonable running time for a variety of series. The `sym` argument is the same as discussed for the `elem` function, which is the symbol encoding missing data. The last argument is the list of control variables for the `optim` function, embedded in the `ar.mle` function to maximize the likelihood. By default, `ar.mle` uses BFGS updating. Here we set BFGS iteration limit to 200. To conduct the imputation, the function calls upon `elem` to compute the permutation matrix P and `covmat` to construct the covariance matrix $\mathbf{\Sigma}$.

Using the `ar1sim` data, we then try to estimate $\boldsymbol{\theta}$ with the `ar.miss` function,

```
> set.seed(2345) #to get same answers as in this example
> ar.miss(data = ar1sim)

         mu         phi      sigma2
0.007578813 0.566670041 0.923626563
```

Recall the true values are $\mu = 0, \phi = 0.5, \sigma^2 = 1$. We see that this proposed method seems to work fine, with estimates close to the population values. In fact, even with missing data, the method was able to detect the true AR order. We also note that this method is fast and efficient, at least for this example.

3 Conclusions and discussions

The package `armiss` implements the method/algorithm described in the previous section. The proposed method offers a simple and statistically sound way to do imputation in time series analysis, if we are willing to adopt certain parametric models (an AR(p) for our package). However, we can easily generalize to include the autoregressive moving average (ARMA) models by modifying the `covmat` function to construct ARMA covariance matrices, and use the `arima` function to do estimation. As mentioned before, we are currently working on extending to multiple time series collected over multiple locations, each with different patterns of missing values. Also, another direction that we are working on is to use the same imputation framework in a non-parametric time series model. We are hoping to extend the basic functionalities of the `armiss` package to include these more general cases in the future.

References

P.~J. Brockwell and R.~A. Davis. *Time series: theory and methods*. Springer-Verlag New York, Inc., New York, NY, USA, 1986.

W.~Chen. *Techniques in Aquatic Toxicology, Volume 2*, chapter 21. Simple methods for estimating exposure concentrations of pesticide resulting from non-point source applications in agricultural drainage networks, pages 357–382. CRC Press, 2005.

G.~Gardner, A.~C. Harvey, and G.~D.~A. Phillips. Algorithm as154. an algorithm for exact maximum likelihood estimation of autoregressive-moving average models by means of kalman filtering. *Applied Statistics*, 29:311–322, 1980.

P.~Hall. *The Boostrap and Edgeworth Expansion*. Springer-Verlag New York, Inc., New York, NY, USA, 1992.

P.~Hall, W.~Hardle, and L.~Simar. On the inconsistency of bootstrap distribution estimators. *Computational Statistics & Data Analysis*, 16(1):11–18, June 1993.

R.~H. Jones. Maximum likelihood fitting of arma models to time series with missing observations. *Technometrics*, 22(3):pp. 389–395, 1980.

M.~R. Leadbetter, G.~Lindgren, and H.~Rootzén. *Extremes And Related Properties Of Random Sequences And Processes*. Springer-Verlag New York, Inc., New York, NY, USA, 1983.

J.~Lindström, A.~A. Szpiro, P.~D. Sampson, L.~Sheppard, A.~P. Oron, M.~Richards, and T.~Larson. A flexible spatio-temporal model for air pollution: Allowing for spatio-temporal covariates. *UW Biostatistics Working Paper Series*, page 370, 2011.

U.~Miroslava. The extreme value distribution of rainfall data at belgrade, yugoslavia. *Atmósfera*, 5(1), 2009.

J.~W. Park, M.~G. Genton, and S.~K. Ghosh. Censored time series analaysis with autoregressive moving average models. *The Canadian Journal of Statistics*, 35(1):151–168, 2007.

J.~W. Park, M.~G. Genton, and S.~K. Ghosh. Nonparametric autocovariance estimation from censored time series by gaussian imputation. *Journal of Nonparametric Statistics*, 21(2):241–259, 2009.