Comparison of Estimators for NileMin Series

Justin Veenstra
Western University

A. Ian McLeod
Western University

Abstract

We use the BIC criterion to select the best long-memory ARMA(p, q) model with $p, q = 0, 1, 2, 3$ and long-memory model specifications FD, FGN and PLA. Also the best ARMA(p, q), $p, q = 0, 1, 2, 3$ with no long-memory component is determined using the BIC.

Keywords: BIC, long-memory models .

1. Introduction

The Nile river flow minima, 660-1320, comprising $n = 663$ observations is a famous example of a time series that is well-fit by a long-memory time series model. This series was originally used by Hurst (1951) and some discussion of the data is given in **?**, §5.9 and Beran (1994, §1.4). The Hurst K statistic (Hipel and McLeod 1994), $K = s^{-1}(\max_t R_t / \min_t R_t) = 0.825$, where $R_t, t = 1, \ldots, n$ is the cumulative range and s is the sample standard deviation, provides a simple, fast, and consistent estimate of H in the FGN model (**?**, Corollary 3.6).

Exact maximum likelihood was used to fit the autoregressive moving-average model of order (p, q) by itself as well as convolved with three types of long-memory: fractionally differenced white noise, fractional Gaussian noise, and power-law-decay-autocorrelation. The BIC criterion was used to select the best model among the four types taking $p, q = 0, \ldots, 3$. In all there are 64 models. In the short-memory case the best model was with $p = q = 1$ while in the long-memory case, $p = q = 2$ was best. The best fitting models for each of the four types are compared in Table using their relative plausibility. For likelihoods, the relative plausibility is defined by $R = L/L^*$, where L is the likelihood and L^* is the largest likelihood out of the 64 models fitted. Similarly for the BIC, $R_{\text{BIC}} = \exp\{-0.5 * (\text{BIC} - \text{BIC}^*)\}$. So in terms of the BIC, we see that

The models are compared in Table in terms of their relative likelihoods,

This vignette accompanies our FGN package (McLeod and Veenstra 2013).

Further work is discussed in Veenstra (2013).

2. Find the Best Model

These computations take about 2 minutes.

```
R> require("FGN")
R> z <- NileMin
```

```
R> z <- z - mean(z)
R> n <- length(z)
R> P <- Q <- 3
R> #
R> TotalTimes <- numeric(4)
R> names(TotalTimes) <- c("FD", "FGN", "PLA", "NONE")
R> #FD
R> numMod <- (P+1)*(Q+1)
R> outMod <- vector("list", numMod)
R> ii <- 0
R> #takes about 42 seconds on unit
R> startTime <- proc.time()
R> for (p in 0:P)
+     for (q in 0:Q) {
+     ii <- ii+1
+     k <- p+q+2
+     order <- c(p,0,q)
+     ans <- earfima(z, order=order, lmodel="FD")
+     Le <- ans$LL
+     bice <- -2*Le+k*log(n)
+     out <- c(p,q,Le,bice)
+     names(out) <- c("p","q","Le","bice")
+     outMod[[ii]] <- out
+     }
R> endTime <- proc.time()
R> totalTime <- endTime-startTime
R> TotalTimes[1] <- totalTime[1]
R> m<-matrix(unlist(outMod),byrow=TRUE,ncol=4)
R> dimnames(m)[[2]]<- c("p","q","Le","bice")
R> ind1 <- which.min(m[,"bice"])
R> mc<-rep(" ", 16)
R> mc[ind1]<-"*"
R> dimnames(m)[[1]]<-mc
R> mFD<-m
R> #
R> #FGN
R> numMod <- (P+1)*(Q+1)
R> outMod <- vector("list", numMod)
R> ii <- 0
R> #takes about 42 seconds on unit
R> startTime <- proc.time()
R> for (p in 0:P)
+     for (q in 0:Q) {
+     ii <- ii+1
+     k <- p+q+2
+     order <- c(p,0,q)
+     ans <- earfima(z, order=order, lmodel="FGN")
```

```
+     Le <- ans$LL
+     bice <- -2*Le+k*log(n)
+     out <- c(p,q,Le,bice)
+     names(out) <- c("p","q","Le","bice")
+     outMod[[ii]] <- out
+     }
R> endTime <- proc.time()
R> totalTime <- endTime-startTime
R> TotalTimes[2] <- totalTime[1]
R> m<-matrix(unlist(outMod),byrow=TRUE,ncol=4)
R> dimnames(m)[[2]]<- c("p","q","Le","bice")
R> ind1 <- which.min(m[,"bice"])
R> mc<-rep(" ", 16)
R> mc[ind1]<-"*"
R> dimnames(m)[[1]]<-mc
R> mFGN<-m
R> #
R> #PLA
R> numMod <- (P+1)*(Q+1)
R> outMod <- vector("list", numMod)
R> ii <- 0
R> #takes about 42 seconds on unit
R> startTime <- proc.time()
R> for (p in 0:P)
+     for (q in 0:Q) {
+     ii <- ii+1
+     k <- p+q+2
+     order <- c(p,0,q)
+     ans <- earfima(z, order=order, lmodel="PLA")
+     Le <- ans$LL
+     bice <- -2*Le+k*log(n)
+     out <- c(p,q,Le,bice)
+     names(out) <- c("p","q","Le","bice")
+     outMod[[ii]] <- out
+     }
R> endTime <- proc.time()
R> totalTime <- endTime-startTime
R> TotalTimes[3] <- totalTime[1]
R> m<-matrix(unlist(outMod),byrow=TRUE,ncol=4)
R> dimnames(m)[[2]]<- c("p","q","Le","bice")
R> ind1 <- which.min(m[,"bice"])
R> mc<-rep(" ", 16)
R> mc[ind1]<-"*"
R> dimnames(m)[[1]]<-mc
R> mPLA<-m
R> #
R> #
```

```
R> #NONE
R> numMod <- (P+1)*(Q+1)
R> outMod <- vector("list", numMod)
R> ii <- 0
R> #takes about 42 seconds on unit
R> startTime <- proc.time()
R> for (p in 0:P)
+     for (q in 0:Q) {
+     ii <- ii+1
+     k <- p+q+2
+     order <- c(p,0,q)
+     ans <- earfima(z, order=order, lmodel="NONE")
+     Le <- ans$LL
+     bice <- -2*Le+k*log(n)
+     out <- c(p,q,Le,bice)
+     names(out) <- c("p","q","Le","bice")
+     outMod[[ii]] <- out
+     }
R> endTime <- proc.time()
R> totalTime <- endTime-startTime
R> TotalTimes[4] <- totalTime[1]
R> m<-matrix(unlist(outMod),byrow=TRUE,ncol=4)
R> dimnames(m)[[2]]<- c("p","q","Le","bice")
R> ind1 <- which.min(m[,"bice"])
R> mc<-rep(" ", 16)
R> mc[ind1]<-"*"
R> dimnames(m)[[1]]<-mc
R> mNONE<-m
R> #
R> LLs <- c(mFD["*",3],mFGN["*",3],mPLA["*",3],mNONE["*",3])
R> LLmax <- max(LLs)
R> RLs <- exp(LLs-LLmax)
R> names(RLs) <- c("FD","FGN","PLA","NONE")
R> #
R> bics <- c(mFD["*",4],mFGN["*",4],mPLA["*",4],mNONE["*",4])
R> bicmin <- min(bics)
R> RELs <- exp(-0.5*(bics-bicmin))
R> names(RELs) <- c("FD","FGN","PLA","NONE")
R> tb <- matrix(c(RLs,RELs)*100, byrow=TRUE, nrow=2)
R> dimnames(tb) <- list(c("RL","REL"), names(RELs))
R> tbNileMin <- tb
R> #dump("tbNileMin", file="d:/R/CRAN/FGN/vig/tbNileMin.R")
R> #
R> TotalTimes

   FD   FGN   PLA  NONE
12.51 16.18 13.96 15.30
```

```
R> sum(TotalTimes)

[1] 57.95

R> #
R> mFD

  p q        Le      bice
* 0 0 236.0231 -459.0526
  0 1 236.7121 -453.9340
  0 2 237.0574 -448.1277
  0 3 237.0598 -441.6357
  1 0 236.6242 -453.7581
  1 1 236.9508 -447.9146
  1 2 237.0586 -441.6334
  1 3 237.0581 -435.1355
  2 0 237.0768 -448.1665
  2 1 237.0772 -441.6705
  2 2 238.4707 -437.9607
  2 3 241.5985 -437.7195
  3 0 237.0775 -441.6712
  3 1 237.6591 -436.3375
  3 2 197.8448 -350.2121
  3 3 238.6838 -425.3933

R> mFGN

  p q        Le      bice
* 0 0 236.5197 -460.0459
  0 1 236.7758 -454.0612
  0 2 237.0599 -448.1328
  0 3 237.0621 -441.6404
  1 0 236.7350 -453.9797
  1 1 236.9907 -447.9942
  1 2 237.0610 -441.6381
  1 3 237.6699 -436.3592
  2 0 237.0690 -448.1509
  2 1 237.0735 -441.6631
  2 2 240.8156 -442.6506
  2 3 241.5895 -437.7015
  3 0 237.0786 -441.6733
  3 1 237.0923 -435.2040
  3 2 241.5882 -437.6990
  3 3 242.0621 -432.1501

R> mPLA
```

```
  p q       Le      bice
* 0 0 236.3019 -459.6102
  0 1 236.6812 -453.8720
  0 2 237.0585 -448.1298
  0 3 237.0596 -441.6354
  1 0 236.6272 -453.7641
  1 1 236.9213 -447.8555
  1 2 237.0591 -441.6343
  1 3 237.0587 -435.1367
  2 0 237.0782 -448.1693
  2 1 237.0782 -441.6725
  2 2 241.0509 -443.1212
  2 3 241.5781 -437.6788
  3 0 237.0782 -441.6726
  3 1 237.0788 -435.1769
  3 2 241.5333 -437.5891
  3 3 238.8028 -425.6314
```

```
R> mNONE
```

```
  p q          Le      bice
  0 0    79.64748 -146.3014
  0 1   169.28546 -319.0806
  0 2   192.61781 -359.2485
  0 3   203.30791 -374.1319
  1 0   212.56400 -405.6377
  1 1   229.23378 -432.4805
  1 2   236.72533 -440.9668
  1 3   237.29426 -435.6079
  2 0   221.04253 -416.0980
* 2 1   237.61204 -442.7402
  2 2   237.63930 -436.2979
  2 3   237.98662 -430.4958
  3 0   228.14016 -423.7965
  3 1   237.62977 -436.2789
  3 2 -1324.91696 2695.3113
  3 3   237.40950 -422.8448
```

```
R> #
R> tbNileMin
```

```
          FD       FGN      PLA         NONE
RL  20.41404  33.54475 26.97737 100.00000000
REL 60.85614 100.00000 80.42203   0.01746245
```

3. Summary

	FD	FGN	PLA	NONE
RL	20.4	33.5	27.0	100.0
REL	60.9	100.0	80.4	0.0

Table 1: Relative Plausibility for Fitted Models

References

Beran J (1994). *Statistics for Long-Memory Processes*. Chapman & Hall.

Hipel KW, McLeod AI (1994). *Time Series Modelling of Water Resources and Environmental Systems*. Elsevier. URL http://www.stats.uwo.ca/faculty/aim/1994Book/default.htm.

Hurst HE (1951). "Long-term Storage Capacity of Reservoirs." *Transactions of the American Society of Civil Engineers*, **116**, 770–808.

McLeod AI, Veenstra J (2013). *Title: Fractional Gaussian Noise and hyperbolic decay time series model fitting*. R package version 2.0-11, URL http://CRAN.R-project.org/package=FGN.

Veenstra JQ (2013). *Persistence and Anti-persistence: Theory and Software*. Ph.D. thesis, Western University. URL http://ir.lib.uwo.ca/etd/1119.

Affiliation:

A. Ian McLeod
Department of Statistical and Actuarial Sciences
Western University
E-mail: aim@stats.uwo.ca
URL: http://www.stats.uwo.ca/mcleod
Justin Veenstra
Department of Statistical and Actuarial Sciences
Western University