Faster Maximum Likelihood Estimation for AR and ARIMA Models

Author

April 9, 2003
Abstract:

The usual algorithms for the computation of the likelihood of an autoregressive model of order, \( p \), AR(\( p \)), require \( O(np) \) flops per function evaluation. The algorithm given in this article requires only \( O(p) \) flops in repeated likelihood evaluations after an initial setup which requires \( O(np) \) flops. This is useful for maximum likelihood estimation with massive datasets containing long time series. The algorithm is extended to provide the exact maximum likelihood estimate (MLE) for the mean parameter in addition to the \( p \) AR parameters. Using this algorithm it is easy and quick to evaluate the MLE for AR(\( p \)) models with \( p \leq 15 \) using a high level quantitative programming environment such as \textit{Mathematica} or S. The new algorithm estimates are compared with the Yule-Walker (YW), and Burg estimates as well as the exact maximum likelihood estimates obtained from \texttt{arima} in R for an AR(11) fit to the lynx time series data. Simulation experiments which illustrate the usefulness of this algorithm are discussed. The new algorithm is extended to provide an efficient approximate maximum likelihood estimate for ARIMA models. These estimates are compared with those from the R function \texttt{arima}. A simulation experiment demonstrates that this algorithm can fit long ARIMA time series.

**Key Words:** Burg estimator; Exact maximum likelihood estimator; High-order autoregression; Long time series; Massive datasets; Spectral density estimation
1. Introduction

The AR \((p)\) model may be written in operator notation as
\[ \phi(B)(z_t - \mu) = \alpha_t, \]
where \(\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p\) and \(B\) is the backshift operator on \(t\). And \(\alpha_t\) is assumed to be Gaussian white noise with mean zero and variance \(\sigma^2_a\). It is assumed that \(z_t\) is stationary and that all roots of \(\phi(B) = 0\) are outside the unit circle.

Given \(n\) consecutive observations from this time series model, \(z_1, \ldots, z_n\), a method of computing the exact loglikelihood function for this model was given by Champernowne (1948) and Box, Jenkins and Reisel (1994, §A7.5). If \(\mu\) is assumed known, this method requires \(O(p^2)\) flops per loglikelihood evaluation after an initial setup which requires \(O(np^2)\). In this paper, this algorithm is further refined so that the initial setup requires \(O(np)\) flops. A new efficient algorithm for exact MLE of the mean is also developed. By approximating an ARMA model with a high-order AR model, this algorithm may be applied to obtain approximate maximum likelihood estimates for the ARMA and ARIMA models for massively long time series data. With these algorithms exact maximum likelihood for the AR and approximate maximum likelihood for the ARIMA models can easily be efficiently implemented in a high level quantitative programming language such as Mathematica or S.

2. Exact Loglikelihood Function

It follows from Champernowne (1948, eq. 3.5) and Box, Jenkins and Reisel (1994, eqn. A7.5.8) that provided \(n > 2p\), the loglikelihood function may be
written
\[ L(\phi_1, \ldots, \phi_p, \mu, \sigma_a^2) = -\frac{n}{2} \log(\sigma_a^2) - \frac{1}{2} \log(g_p) - S(\phi_1, \ldots, \phi_p, \mu), \] (1)
where,
\[ S(\phi_1, \ldots, \phi_p, \mu) = \beta' D \beta, \] (2)
where \( D \), the Champernowne matrix, is the \((p + 1) \times (p + 1)\) matrix with \((i, j)\)-entry,
\[ D_{i,j} = D_{j,i} = (z_i - \mu)(z_j - \mu) + \ldots + (z_{n+1-j} - \mu)(z_{n+1-i} - \mu) \] (3)
and \( \beta = (-1, \phi_1, \ldots, \phi_p) \).

Maximizing over \( \sigma_a^2 \), the concentrated loglikelihood may be written,
\[ L(\phi, \mu) = -\frac{n}{2} \log(S(\phi_1, \ldots, \phi_p, \mu)/n) - \frac{1}{2} \log(g_p), \] (4)
where \( \phi = (\phi_1, \ldots, \phi_p) \). Since the sample mean, \( \bar{z} \), is an asymptotically fully efficient estimate of \( \mu \), it is often used in place of the exact maximum likelihood estimate. If the sample mean is used, \( \mu \) may be replaced by \( \bar{z} \) in eqn. (3) and so after the initial evaluation, repeated evaluations of eqn. (4) only requires \( O(p^2) \) flops.

Assuming the mean is known or is given by the sample mean, the exact MLE for \( \phi \) may be obtained by maximizing \( L(\phi, \mu) \). Mathematica implements a constrained version of the conjugate direction minimization algorithm of Powell (1964) in its built-in function \texttt{FindMinimum} and this function has proved very effective in obtaining the exact MLE by minimizing the negative of the loglikelihood. The parametrization using
partial autocorrelations (Barndorff-Nielsen and Schou, 1973; Monahan, 1984),

\[(\phi_1, \ldots, \phi_p) \leftrightarrow (\zeta_1, \ldots, \zeta_p)\]  

is used to constrain the minimization to the unit hypercube. In the reparameterized model, it is easily shown that,

\[g_p = \prod_{j=1}^{p} (1 - \zeta_j^2)^{-j}.\]  

The Burg estimators are used as initial estimates since they seem to be more accurate than the YW estimates in many situations (Percival and Walden, 1993, p.414) and like the YW estimates, the Burg estimates are always inside the admissible region. As shown in §6, AR (p) models with \(p \leq 15\) may be fit in a few seconds in Mathematica using this algorithm. Because the sample mean is used, we will refer to this approximate MLE as the AMLE.

3. Faster Champernowne Matrix Computation

In this section we will assume that the data has been corrected by the sample mean and take \(\mu = 0\) for this mean-corrected series, \(z_t, t = 1, \ldots, n\). Note \(D_{i,j}\) has \(n - (i + 1) - (j + 1)\) terms, so direct evaluation of the Champernowne matrix, requires \(O(np^2)\) flops. It may be shown that \(D = C - E\) where the \((i, j)\)-entry of the matrix \(C\) may be written, \(C_{i-j}\), where

\[C_k = z_1z_k - \ldots - z_{n-k}z_n\]  

and the \((i, j)\)-entry for the matrix \(E\) may be computed sequentially

\[E_{i+1,j+1} = E_{i,j} + z_i\bar{z}_j + z_{n+1-i}\bar{z}_{n+1-j}, \ i < j.\]
Using the above results $D$ may be computed in $O(np)$ flops.

4. Exact MLE for the Mean Parameter

Assuming that $(\phi_1, \ldots, \phi_p)$ are known, the exact MLE is given by,

$$\hat{\mu} = \frac{1_n' \Gamma_n^{-1} z}{1_n' \Gamma_n^{-1} 1_n},$$

(9)

where $1_n$ denotes the $n$ dimensional column vector with all entries equal to 1, $z = (z_1, \ldots, z_n)$ and $\Gamma_n^{-1}$ denotes the inverse of the matrix of $n$ successive observations. Since $\hat{\mu}$ does not depend on $\sigma_a^2$, we may assume without loss of generality that $\sigma_a^2 = 1$. Direct evaluation of eqn. (9) using the exact inverse matrix derived by Siddiqui (1958) would require $O(n^2)$ flops.

Zinde-Walsh (1988, eqn. 3.2) showed that

$$\Gamma_n^{-1} = \hat{\Gamma}_n - B$$

(10)

where $\hat{\Gamma}_n$ denotes the $n \times n$ matrix with $(i,j)$-entry given by $\gamma_{i,j}^{(u)}$, where $\gamma_{k}^{(u)} = \text{Cov}(u_t, u_{t-k})$, $u_t = \phi(B)a_t/\sigma_a$ and $B$ is a zero matrix except for $p \times p$ submatrices in the upper-left and lower-right corners. The $(i,j)$-entry of the submatrix of $B$ in the upper-left corner is

$$B_{i,j} = \sum_{k=\min(i,j)}^{p-|i-j|} \phi_k \phi_{k+i-j}. $$

(11)

The matrix in the lower-right corner is, of course, just the transpose of upper-left corner submatrix.

Using the above results it was found after some computation that,

$$1_n' \Gamma_n^{-1} = 1_n \phi^2(1) - (\epsilon_1, \ldots, \epsilon_p, 0, \ldots, 0, \epsilon_p, \ldots, \epsilon_1)$$

$$- (\kappa_1, \ldots, \kappa_p, 0, \ldots, 0, \kappa_p, \ldots, \kappa_1), $$

(12)
where \( \phi(1) = 1 - \phi_1 - \ldots - \phi_p, \epsilon = 1_n^t B, \)

\[
\epsilon = (\epsilon_1, \ldots, \epsilon_p, 0, \ldots, 0, \epsilon_p, \ldots, \epsilon_1)
\]

(13)

and

\[
\kappa_i = \sum_{k=1}^{i} \gamma_k(u).
\]

(14)

Using eqn. (12), \( \hat{\mu} \) can now be evaluated in \( O(n) \) flops.

An iterative algorithm is used for simultaneous joint MLE of

\((\phi_1, \ldots, \phi_p, \mu),\)

1. Correct the series for sample mean, \( \bar{z} \).

2. Find the exact mle for \( \phi \) for the mean corrected series

3. Using \( \hat{\phi} \) from Step 2, evaluate eqn. (9) using (12), (13) and (14) to
   obtain a revised estimate for exact mle of the mean.

4. Repeat Steps 2 and 3.

In practice, this algorithm rarely takes more than about four or five
iterations to converge. This algorithm will be denoted by EMLE in our
discussion below.

5. Lynx Data Comparisons

In this section we compare the estimates given by our algorithms,
AMLE and EMLE, with estimates obtained using the YW and Burg
algorithms. In addition a comparison is made with the exact maximum
likelihood algorithm available in the R algorithm \texttt{arima}, denoted for brevity by RMLE.

Tong (1990, p.335) selected an AR (11) model to fit to the logarithms of the well-known lynx time series data. In Table 1, we compare AR (11) models fitted the five different algorithms. The differences between the estimates can be summarized using the Piccolo-distance metric (Piccolo, 1990) which in this case is just the Euclidean norm of the vector difference of the $\phi$ parameters. The Piccolo-distances, shown in Table 2, indicate that there is a relatively big difference between the Burg and Yule-Walker. It may be shown that all roots corresponding to the models fitted in Table 1 are quite close to the unit circle and so it may be expected that YW estimates are severely biased (Tjøstheim and Paulsen, 1983) and this explains why there is such a difference.

The difference between the Burg estimates and the various MLE estimators is relatively smaller and the differences between the three MLE estimators are all relatively very small.

[ Table 1 here ]

6. Simulation Comparison of AR Estimators

A simulation experiment was done to compare the EMLE with the AMLE and the Burg estimate for the AR (4) time series model used by Percival and Walden (1993, eqn. 46A). The parameters of this model are given by $\phi_1 = 2.7607, \phi_2 = -3.8106, \phi_3 = 2.6535$ and $\phi_4 = -0.9238$. This model has a
very sharp spectral peak and the roots of the equation $\phi(B) = 0$ are close the admissible boundary. Percival and Walden (1993, §9.5) show that for this model the Burg estimates provide a much better estimate of the spectral density function than the Yule-Walker estimates. In the simulation we simulated 1,000 replications of this model for a series of length $n = 1024$ and computed the Burg, AMLE and MLE estimators for each simulation. Denoting the estimator by $\hat{\phi}$, the mean-square error may be written

$$d = (\hat{\beta} - \beta)' \mathcal{I}(\hat{\beta} - \beta),$$

where $\mathcal{I}$ is the Fisher-information matrix. For each simulation, $i = 1, ..., 1000$, the value of $d$ was computed for the Burg, AMLE and EMLE. Denote these results by $d_{\text{Burg}}$, $d_{\text{AMLE}}$ and $d_{\text{EMLE}}$. Figure 1 and 2 show the Quantile plot (Cleveland, 1993) for these results for selected percentiles from 0.01 to 0.99. From Figure 1 we see that the AMLE outperforms the Burg almost always. The comparison between the EMLE and AMLE is less clear cut. Most of the time the EMLE is either better or not much different but the upper 10% quantiles for $d_{\text{EMLE}}-d_{\text{AMLE}}$ indicate that sometimes the $d_{\text{EMLE}}$ is much worse than the corresponding lower 10% quantiles. For the Wilcoxon signed-rank test, the signed-rank normal statistic for $d_{\text{EMLE}}-d_{\text{AMLE}}$ is $Z = -4.8121$ so this strongly rejects the null hypothesis of equality of the medians and favors the EMLE.

[ Figures 1 and 2 ]
7. ARIMA Approximate Maximum Likelihood Estimation

Present approximate and exact likelihood evaluation algorithms for the general ARMA require $O(n)$ flops. An approximate ARMA maximum likelihood estimation was implemented by approximating the ARMA model by a high order autoregression of order $P$. In practice $P = 30$ seems adequate for many models but for complicated or high order models such as seasonal models, $P$ may need to increased.

It is well known that the exact maximum likelihood estimator for an MA (1) model may have a significant probability of attaining the boundary value of $\pm 1$ (Cryer and Ledholter, 1983). This effect also occurs for the moving-average component in mixed ARMA models. However it is known that this effect diminishes when the series length is increased and this effect appears to be negligible when $n \geq 200$ unless the true parameter value is very close to the moving-average boundary. This consideration suggestions that the AMLE is suitable provided the series length is not too small.

For long time series which occur in massive datasets in high frequency financial data and radio physics, the AMLE algorithm is much faster. As an example a series of length $n = 10^6$ was simulated and fit using RMLE. This R function is interfaced to optimized C code and so it runs very efficiently. However RMLE required about 91 seconds of cpu time whereas the AMLE algorithm, coded entirely in the high-level Mathematica programming language, required only 16 seconds on the same PC.

A simulation experiment was conducted to compare the distribution of our AMLE estimates with those of the R arima function for the
ARMA (1, 1) model with $\phi = 0.9$ and $\theta = 0.5$ in the case of $n = 200$. Then our AMLE was used to examine the distribution of the estimates with $n = 10^6$. Let $\hat{\beta} = (\hat{\phi}_1 - \phi_1, \hat{\theta}_1 - \theta_1)$, the asymptotic distribution of $\sqrt{n}\hat{\beta}$ is normal with mean zero and covariance matrix $I^{-1}$, where

$$I = \begin{pmatrix} 100/19 & -20/11 \\ -20/11 & 4/3 \end{pmatrix}$$

(16)

Hence, asymptotically, $d = n\beta' I^{-1} \beta$ is $\chi^2$ distributed with 2 df. In our simulations, 500 time series of length $n = 200$ were fit. A QQ plot of the distribution of $d$ vs. the $\chi^2$ distributed with 2 df is shown in Figure 3. From this plot we see that for $n = 200$, there is a major departure from the asymptotic approximation for both the AMLE algorithm and for the algorithm `arima` in R. Surprisingly, the AMLE algorithm seems to work better than RMLE when $n = 200$. When $n$ was increased to $n = 10^6$, the AMLE estimates agreed well with the asymptotic distribution as was hoped.

[ Figure 3 ]

With the software that we have provided further simulation experiments could be done. Our experience suggests that this algorithm is quite reasonable for many time series models where $n \geq 200$.

8. Concluding Remarks

A complete Mathematica package for ARIMA and multiplicative seasonal ARIMA time series modelling has been developed using the algorithms outlined in this article. This package will be made available in the software section of this journal. One advantage of Mathematica over R is its
symbolic capabilities which are used in our package in functions to compute theoretical autocorrelations and the large-sample Fisher information matrix. *Mathematica* notebooks showing all computations reported in this paper are available with this package (anonymous, 2003A).

An R package is also under development will also be available soon (anonymous, 2003B).

The algorithms in this paper can also be extended to subset AR and subset ARIMA modelling (anonymous, 2003C).
REFERENCES


Applications, Cambridge University Press.


Table 1: Comparison of Yule-Walker (Y-W), Burg, Approximate MLE (AMLE) and MLE estimators for the AR(11) model fitted to the logged lynx data.

<table>
<thead>
<tr>
<th></th>
<th>YW</th>
<th>Burg</th>
<th>AMLE</th>
<th>EMLE</th>
<th>RMLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi_1$</td>
<td>1.1387</td>
<td>1.1745</td>
<td>1.1678</td>
<td>1.1683</td>
<td>1.1675</td>
</tr>
<tr>
<td>$\phi_2$</td>
<td>-0.5078</td>
<td>-0.5511</td>
<td>-0.5447</td>
<td>-0.5450</td>
<td>-0.5443</td>
</tr>
<tr>
<td>$\phi_3$</td>
<td>0.2123</td>
<td>0.2686</td>
<td>0.2657</td>
<td>0.2661</td>
<td>0.2657</td>
</tr>
<tr>
<td>$\phi_4$</td>
<td>-0.2698</td>
<td>-0.3179</td>
<td>-0.3097</td>
<td>-0.3102</td>
<td>-0.3088</td>
</tr>
<tr>
<td>$\phi_5$</td>
<td>0.1124</td>
<td>0.1675</td>
<td>0.1548</td>
<td>0.1540</td>
<td>0.1537</td>
</tr>
<tr>
<td>$\phi_6$</td>
<td>-0.1239</td>
<td>-0.1582</td>
<td>-0.1465</td>
<td>-0.1459</td>
<td>-0.1461</td>
</tr>
<tr>
<td>$\phi_7$</td>
<td>0.0678</td>
<td>0.0712</td>
<td>0.0571</td>
<td>0.0567</td>
<td>0.0569</td>
</tr>
<tr>
<td>$\phi_8$</td>
<td>-0.0401</td>
<td>-0.0462</td>
<td>-0.0290</td>
<td>-0.0295</td>
<td>-0.0294</td>
</tr>
<tr>
<td>$\phi_9$</td>
<td>0.1337</td>
<td>0.1437</td>
<td>0.1348</td>
<td>0.1347</td>
<td>0.1346</td>
</tr>
<tr>
<td>$\phi_{10}$</td>
<td>0.1853</td>
<td>0.2182</td>
<td>0.2017</td>
<td>0.2019</td>
<td>0.2021</td>
</tr>
<tr>
<td>$\phi_{11}$</td>
<td>-0.3110</td>
<td>-0.3486</td>
<td>-0.3385</td>
<td>-0.3396</td>
<td>-0.3395</td>
</tr>
</tbody>
</table>

Table 2: Piccolo Distances Between Fitted AR(11) Models

<table>
<thead>
<tr>
<th></th>
<th>YW</th>
<th>Burg</th>
<th>AMLE</th>
<th>EMLE</th>
<th>RMLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>YW</td>
<td>0.0000</td>
<td>0.1245</td>
<td>0.1012</td>
<td>0.1017</td>
<td>0.1003</td>
</tr>
<tr>
<td>Burg</td>
<td>0.1245</td>
<td>0.0000</td>
<td>0.0375</td>
<td>0.0373</td>
<td>0.0378</td>
</tr>
<tr>
<td>AMLE</td>
<td>0.1012</td>
<td>0.0375</td>
<td>0.0000</td>
<td>0.0018</td>
<td>0.0020</td>
</tr>
<tr>
<td>MLE</td>
<td>0.1017</td>
<td>0.0373</td>
<td>0.0018</td>
<td>0.0000</td>
<td>0.0018</td>
</tr>
<tr>
<td>RMLE</td>
<td>0.1003</td>
<td>0.0378</td>
<td>0.0020</td>
<td>0.0018</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Figure 1: Quantile plot of dAMLE-dBurg.
Figure 2: Quantile plot of dEMLE-dAMLE.
Figure 3: QQ plots of $d$, where $d = n\beta'\mathcal{I}^{-1}\beta$, and $\hat{\beta} = (\hat{\phi}_1 - \phi_1, \hat{\theta}_1 - \theta_1)$. 