The adaptive Lasso for Specifying Autoregressive Moving-Average Models *

Christian Kascha†

February 14, 2016

Abstract

The properties of the adaptive lasso as a method for specifying and estimating autoregressive moving-average models are studied. I develop an estimation method based on the adaptive lasso and previous algorithms and prove some of its asymptotic properties. The small sample properties of the method are studied with a Monte Carlo simulation where various implementations of the algorithm are ranked according to different performance criteria. A general recommendation is given on how to implement the algorithm.

JEL classification: C22, C51, C52, C53
Keywords: Adaptive Lasso, ARMA Models, Model Selection, Forecasting

1 Introduction

In this paper, the properties of the adaptive lasso as a method for specifying and estimating autoregressive moving-average (ARMA) models are studied. I develop an estimation method based on the adaptive lasso of Zou (2006) and the algorithms of Hannan & Rissanen (1982*b*), Hannan & Kavalieris (1984), Kavalieris (1991) and Chan & Chen (2011). The asymptotic properties of the method are established. The small sample properties of the method are studied with a Monte Carlo simulation where various implementations of the algorithm are ranked according to different performance criteria including the precision of the resulting forecasts. A general recommendation is given on how to implement the algorithm.

The problem is that of estimating and specifying the well-known ARMA model

$$\sum_{j=0}^{p} \alpha_j y_{t-j} = \sum_{j=0}^{q} \beta_j \varepsilon_t \tag{1}$$

for a strictly stationary time series $(y_t)_{t\in\mathbb{Z}}$. This is one of the workhorse models for forecasting and its specification is therefore of wide interest.

*I thank participants of the 9th Conference on Computational and Financial Econometrics (CFE 2015) for helpful comments.

†University of Zurich, Chair for Statistics and Empirical Economic Research, Zürichbergstrasse 14, 8032 Zurich, Switzerland; christian.kascha@econ.uzh.ch

Of course, many papers have investigated the same problem. To cite the most prominent, Hannan & Rissanen (1982*b*,*a*) proposed to estimate the orders of an ARMA model by minimizing an information criterion based on least squares regressions. Hannan & Kavalieris (1984) introduced various modifications to the original Hannan-Rissanen method in order to prevent it from overestimating the degrees. Various other works have also analyzed the specification of ARMA models.

Recently, the problem of specifying an ARMA model has been investigated using new techniques taken from the statistical literature. In particular, Tibshirani (1996) introduced the *lasso method* to jointly select the regressors and estimate their parameters in a linear regression model. Zou (2006) proposed the *adaptive lasso method* which is essentially a weighted penalized least squares regression. Various authors derived the properties of the adaptive lasso in a time series context. For example, Medeiros & Mendes (2012), Mendes & Medeiros (2015) develop the asymptotic properties of the adaptive lasso for autoregressive distributed lag (ADL) models with homoscedastic errors and in the presence of generalized autoregressive heteroscedasticity (GARCH), respectively. Ren & Zhang (2010) propose the adaptive lasso for vector autoregressive (VAR) models. Finally, Chan & Chen (2011) combined the original Hannan-Rissanen method with the adaptive lasso method of Zou (2006) in order to specify and estimate the parameters of an ARMA model.

This paper adds to the most recent literature by considering some modifications of the procedure originally proposed by Chan & Chen (2011). In particular I consider a hybrid procedure that pre-selects the maximum degrees using the results in Hannan & Rissanen (1982*b*), Hannan & Kavalieris (1984) and Kavalieris (1991). I modify their proofs and show that this step is necessary in order to ensure the so-called *oracle property* of the estimators. Furthermore, different versions of the algorithm are explored in order to give a recommendation on how to implement the method.

The paper is organized as follows. First, the algorithm is defined. Second, the oracle property is proved for the main variant of the method. Next, different versions of the algorithms are compared in a Monte Carlo study similar to the one of Chan & Chen (2011). The last section concludes and outlines directions of future research.

2 Definition of The Algorithm

I first give the algorithm for the estimation and specification of model (1) in a somewhat general form and then discuss concrete implementations afterwards.

Algorithm 1 (hybrid adaptive lasso) *Suppose there is data* $y_1, \ldots, y_T$*, a fixed constant* $\gamma > 0$*, integers* P*,* Q *and a function* $f(T)$*.*

1. *Choose the lag length* n_T *either determinstcally or by minimizing the AIC information criterion where in any case* $n_T \leq f(T)$*.*

2. *Obtain an estimate of the errors using a "long" autoregression*

$$\hat{\varepsilon}_{t,T} := \sum_{j=0}^{n_T} \hat{a}_{T,j} y_{t-j};\ t = n_T + 1, \ldots, T$$

computed via a Yule-Walker regression (see e.g. Hannan & Rissanen 1982b). Let $m := n_T + \max\{P, Q\} + 1$ *and form the matrix*

$$\underset{(T-m+1)\times(P+Q)}{\hat{X}} := \left((y_{i-j})_{\substack{m\le i\le T\\ 1\le j\le P}} : (\hat{\varepsilon}_{i-j,T})_{\substack{m\le i\le T\\ 1\le j\le Q}} \right)$$
$$= \begin{pmatrix} y_{m-1} & \cdots & y_{m-P} & \hat{\varepsilon}_{m-1,T} & \cdots & \hat{\varepsilon}_{T,m-Q} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y_{T-1} & \cdots & y_{T-P} & \hat{\varepsilon}_{T-1,T} & \cdots & \hat{\varepsilon}_{T-Q,T} \end{pmatrix}$$

for $P \ge 1$ *and* $Q \ge 1$ *and modified accordingly if* $P = 0$ *or* $Q = 0$. *The character* : *means horizontal concatenation of the sub-matrices.*

3. *Obtain estimates* $\hat{p}, \hat{q}$ *by minimizing*

$$IC(p, q) = \hat{\tilde{\sigma}}^2_{p,q} + (p + q)\log(T)/T$$

over $p = 0, \ldots, P_T$, $q = 0, 1, \ldots, Q_T$ *where* $\hat{\tilde{\sigma}}^2_{p,q}$ *is the variance estimate based on the recursive calculation of the residuals according to*

$$\begin{aligned} \hat{\tilde{\varepsilon}}_t &= y_t = 0; \quad t < 0 \\ \hat{\tilde{\varepsilon}}_t &= \hat{a}(L)y_t - (\hat{b}(L) - 1)\hat{\tilde{\varepsilon}}_t; \quad t = 1, \ldots, T \\ \hat{\tilde{\sigma}}^2_{p,q} &:= \sum_{t=1}^{T} \hat{\tilde{\varepsilon}}^2_t / T \end{aligned} \tag{2}$$

where $\hat{a}(L)$, $\hat{b}(L)$ *are the least squares estimates of the autoregressive and moving-average polynomials based on residuals from the long autoregression (see Kavalieris 1991).*

4. *Let* $\tau := (\alpha_1, \ldots, \alpha_P, \beta_1, \ldots, \beta_Q)'$. *Estimate* τ *by least squares using* $\hat{p}$, $\hat{q}$ *and* $\hat{\varepsilon}_{t,T}$ *and denote the estimates of the single elements by* $\hat{\tau}_{T,j}$, $j = 1, \ldots, P + Q$.

5. *Set weights* $\hat{\omega}_j := |\hat{\tau}_{T,j}|^{-\gamma}$, *in particular* $\hat{\omega}_j = \infty$ *for* $\hat{\tau}_{T,j} = 0$.

6. *Set a grid of values* $(\lambda_g)_{g=1}^G$; *see section 4.1.*

7. *Estimate* $(\tau(\lambda_g))_{g=1}^G$, *where*

$$\tau(\lambda_g) = \underset{\tau}{\operatorname{argmin}} \left(||y - \hat{X}\,\tau||^2 + \lambda_g \sum_{j=1}^{P+Q} \hat{\omega}_j |\tau_j| \right)$$

and $y := (y_m, \ldots, y_T)'$.

8. *Return* $\tilde{\tau}_T := \tau(\tilde{\lambda}_T)$, *where*

$$\tilde{\lambda}_T = \underset{\lambda}{\operatorname{argmin}} \left(T^{-1}||y - \hat{X}\tau(\lambda)||^2 + |\mathcal{A}_\lambda| \frac{\log(T)}{T} \right)$$

where $|\mathcal{A}_\lambda|$ *is the number of non-zero coefficients in* $\tau(\lambda)$.

General remarks: Steps 1 - 2 recover the residuals from a long autoregression. The important difference to the algorithm proposed by Chan & Chen (2011) is step 3 and the use of its outcome at step 4. These steps are necessary because the matrix $\hat{X}$ defined in step 2 is singular in the limit if the true orders are strictly smaller than P and Q. The rest of the steps are again standard. The steps 1 and 3 are not specified in detail because I try out different versions of them later on in the Monte Carlo study.

Choice of n_T in step 1: Different methods of choosing n_T can lead to consistent estimators as long as assumption (introduced later) are satisfied. We consider choosing n_T deterministically or via the AIC as shown in the following table:

Table 1: Choice of n_T

NT1	$n_T = \text{argmin}_{0\leq p\leq N_T} AIC(p);\ N_T = \max(f(T), P, Q)$
NT2	$n_T = \text{argmin}_{0\leq p\leq N_T} AIC(p);\ N_T = \max(f(T), P, Q), n_{min} = \max(P, Q)$
NT3	$n_T = \text{argmin}_{0\leq p\leq N_T} AIC(p);\ N_T = f(T)$
NT4	$n_T = f(T)$

The first two options take into account that we have given information on the *maximum* lag order of the ARMA polynomials. The second option even enforces a lag length for the vector autoregression which is at least as high as the a priori given maximum lag for the polynomials. The last two options do not use neither P nor Q.

Choice of p, q in step 3: As mentioned above, this step is new relative to the algorithm of Chan & Chen (2011) in that it is necessary to pre-estimate the orders because the regressor matrix $\hat{X}$ is converging to a singular matrix if the orders are over-specified. This argument is outlined in Hannan & Rissanen (1982*b*, page 90). However, the choice of the maximum can be implemented in various ways. We investigate the following options:

Table 2: Choice of p and q

PQ1	$(\hat{p}, \hat{q}) = \text{argmin}_{(p\leq P_T, q\leq Q_T)}(IC(p,q)), (P_T, Q_T) = (\min(n_T, P), \min(n_T, Q))$
PQ2	$(\hat{p}, \hat{q}) = \text{argmin}_{(p\leq P_T, q\leq Q_T)}(IC(p,q)), (P_T, Q_T) = (n_T, n_T)$
PQ3	$(\hat{p}, \hat{q}) = \text{argmin}_{(p\leq P_T, q\leq Q_T)}(IC(p,q)), (P_T, Q_T) = (P, Q)$
PQ4	$(\hat{p}, \hat{q}) = (P, Q)$

where $IC(p,q)$ refers to the information criterion defined in step 3. Both, $PQ1$ and $PQ2$ limit the search over possible orders by the order of the previous step. While $PQ1$ then stops at the maximum lag orders P and Q, the second variant ignores the "information" in P and Q entirely. On the other hand, the last two options ignore that the previous step was based on an order n_T and proceed. The last option does not choose a lag length based on the data but

just uses the maximum orders. It corresponds to the version of Chan & Chen (2011).

Further alternatives: Apart from the variations discussed above, alternative methods could be used at almost every step of the algorithm.

For example, the methods to *estimate* the long autoregression at step 2 could also be varied. As Poskitt (1994) has shown, the Yule-Walker estimator, the OLS estimator or the Burg algorithm are asymptotically equivalent.

The order estimation at stage 3 could be undertaken by the various methods outlined in Hannan & Kavalieris (1984). However, the method of Kavalieris (1991) is much simpler than the alternative methods given in the former paper.

At step 4, one could use other asymptotically efficient estimators such as the one of the third stage in Hannan & Rissanen (1982*b*). This approach is definitely worth exploring.

Finally, steps 7 and 8 could be replaced by a penalized maximum likelihood step.

This list could probably be extended. In this paper, however, we focus on variations of the basic algorithm and the $4 \times 4 = 16$ variations that results from combining $NT1 - NT4$ with $PQ1 - PQ4$. The proposed methods that pre-select the orders of the ARMA model have to be contrasted with the originally proposed procedure in Chan & Chen (2011) which corresponds to using $(NT1, PQ4)$ or $(NT3, PQ4)$, depending on the relation between $f(T), P$ and Q.

3 Large-Sample Properties

Let p_0, q_0 and $\alpha_{0,j}, \beta_{0,j}$ denote the true orders and parameters of model (1). In this section, we prove the large-sample properties of the proposed algorithm subject to the following assumptions.

3.1 Assumptions

Assumption A.1 *The assumptions concerning the DGP are:*

1. *The time series $(y_t)_{t\in\mathbb{Z}}$ is generated by an ARMA model (1) with true polynomials*

$$a_0(z) := \sum_{j=0}^{p_0} \alpha_{0,j} z^j \quad b_0(z) := \sum_{j=0}^{q_0} \beta_{0,j} z^j$$

   where $\alpha_{0,0} = \beta_{0,0} := 1$.

2. *$a_0(z) \neq 0$ $(|z| \leq 1)$; $b_0(z) \neq 0$ $(|z| \leq 1)$*

3. *$a_0(z)$ and $b_0(z)$ are co-prime*

4. *$(\varepsilon_t)_{t\in\mathbb{Z}}$ is a strictly stationary martingale difference sequence with respect to $A_t = \sigma(\varepsilon_s, s \leq t)$.*

5. *$E\{\varepsilon_t^2 | A_{t-1}\} = \sigma_0^2$, $E\{\varepsilon_t^4\} < \infty$*

Assumption A.2 *The estimator $\tilde{\tau}_T = \tau(\tilde{\lambda}_T)$ is obtained by an implementation of algorithm 1 such that*

1. $n_T \to \infty$ *monotonically such that* $\log(T) \leq n_T \leq (logT)^b$, *for some* $b > 1$
2. $n_T \geq P_T, n_T \geq Q_T$
3. $P_T = P \geq p_0$ *and* $Q_T = Q \geq q_0$
4. $\frac{\tilde{\lambda}_T}{\sqrt{T}} T^{\eta/2} \to \infty$ *and* $\tilde{\lambda}_T / \sqrt{T} \to 0$.

For the part on choosing the lag order n_T, only the variant NT2 of the implementations mentioned in section 2 guarantees the assumptions are satisfied provided $f(T)$ is chosen accordingly. NT4 might satisfy the assumptions if $f(T)$ does it. For choosing p and q, the variant PQ3 imposes the assumptions while PQ1 and PQ2 satisfy the assumptions if n_T is high enough. All variants with PQ4 do not satisfy the assumptions as it does not choose the orders. In sum, the combinations NT2PQ3 and NT4PQ3 (provided $f(T)$ is high enough) satisfy the assumptions while most other variants do so only when the sample size becomes large. Note that the assumptions are formulated for a generic penalty parameter $\tilde{\lambda}_T$ satisfying the above assumptions. I do not prove that step 8 yields a $\tilde{\lambda}_T$ which satisfies the assumptions. This is left for future research.

Note that the assumptions given here are slightly stricter than the assumptions in Hannan & Rissanen (1982*b*). The only difference is that I assume in addition that (ε_t) is strictly stationary. Therefore, the results of Hannan & Rissanen (1982*b*), Kavalieris (1991) and Chan & Chen (2011) can be invoked in the following proofs.

Notation: We use notation very similar to the used in Chan & Chen (2011). Set $R := P + Q$. Let τ_0 be the $R \times 1$ vector of true parameters and let $\mathcal{A}_0 = \{j : \tau_{0,j} \neq 0\}$ be the set of all truly non-zero coefficients. Similarly denote by $\tilde{\mathcal{A}}_T = \{j : \tilde{\tau}_{T,j} \neq 0\}$ the set associated with the estimator $\tilde{\tau}_T$. For a square matrix C and a set of indices $\mathcal{B}$, denote by $C_{\mathcal{B}}$ the sub-matrix that results from choosing the elements c_{ij} with $i \in \mathcal{B}$ and $j \in \mathcal{B}$. Similarly, for a vector c, let $c_{\mathcal{B}}$ denote the sub-vector containing the elements c_i of c with $i \in \mathcal{B}$. For example, $\tilde{\tau}_{T,\mathcal{A}_0}$ is the sub-vector containing all element $\{\tilde{\tau}_{T,i} : i \in \mathcal{A}_0\}$.

For generic random variables Z_T and Z, the notation $Z_T \xrightarrow{d} Z$, $Z_T \xrightarrow{p} Z$ and $Z_T \xrightarrow{a.s.} Z$ means that Z_T converges in distribution, in probability or almost surely to Z.

Theorem 3.1 *Suppose the estimator $\tilde{\tau}_T$ satisfies A.2 and is applied to a DGP satisfying A.1. Let $x_t' := (y_{t-1}, \dots, y_{t-P}, \varepsilon_{t-1}, \dots, \varepsilon_{t-Q})$ and $C := E(x_t x_t')$. Then*

(i) Asymptotic normality:

$$\sqrt{T}(\tilde{\tau}_{T,\mathcal{A}_0} - \tau_{0,\mathcal{A}_0}) \xrightarrow{d} N(0, \sigma_0^2 C_{\mathcal{A}_0}^{-1})$$

(ii) Selection consistency:

$$\lim_{T\to\infty} P(\tilde{\mathcal{A}}_T = \mathcal{A}_0) = 1$$

3.2 Proofs

The proofs presented here are very similar to the ones in Zou (2006) and Chan & Chen (2011). An important difference is that some results for the initial estimators are proved here. These results play a role in the subsequent, adapted proof on the asymptotic normality of the estimators. Another important difference is that I adopt the proof style of Lamport (1995, 2012). In short, it consists of hierarchically structured claims, where each non-obvious claim is either proved by a short explanation or by a series of subsequent claims on a higher hierarchical level, terminated by Q.E.D. to mark the end of the sub-proof.

3.2.1 Initial estimators

First, we have to proof some statements about the initial estimators. For this, some additional notation is necessary. Denote the vector containing the true parameters up to order p and q by $\tau_0(p,q) = (\alpha_{0,1}, \dots, \alpha_{0,p}, \beta_{0,1}, \dots, \beta_{0,q})$. Denote the least squares estimate in step 4 based on orders p and q by

$$\begin{aligned}\hat{\tau}_T(p,q) &= (\hat{X}_{(p,q)}'\hat{X}_{(p,q)})^{-1}\hat{X}_{(p,q)}' y_{(p,q)} \\ &= (\hat{X}_{(p,q)}'\hat{X}_{(p,q)})^{-1} T \frac{1}{T}\hat{X}_{(p,q)}' y_{(p,q)} = \hat{C}_{(p,q)}^{-1}\frac{1}{T}\hat{X}_{(p,q)}' y_{(p,q)},\end{aligned}$$

where $\hat{X}_{(p,q)}$ is a version of $\hat{X}$ given the orders p, q, $y_{(p,q)} := (y_{n(p,q)}, \dots, y_T)'$, $n(p,q) := n_T + p + q - 1$ and $\hat{C}_{(p,q)}$ is defined by the above equation.

Denote by $\tau_0 = \tau_0(P,Q)$ and its elements by $\tau_{0,i}$. Denote by $\hat{\tau}_{T,i}$ the least squares estimates of $\tau_{0,i}$ based on the estimates $\hat{p}$, $\hat{q}$ from step 3.

Lemma 3.2 *Suppose that $\hat{\tau}_T(\hat{p},\hat{q})$ is the least squares estimator after selecting the orders p, q according to step 3 and suppose that assumptions A.1 and A.2 hold.*

(i) $\sqrt{T}(\hat{\tau}_T(\hat{p},\hat{q}) - \tau_0(p_0,q_0)) \xrightarrow{d} N(0, C_{\mathcal{A}_0}^{-1})$

(ii) $\hat{\tau}_{T,i} \xrightarrow{p} 0$ *if* $\tau_{0,i} = 0$

The proof of the first part is

ASSUME: Assumptions A.1 and A.2.
PROVE: $\sqrt{T}(\hat{\tau}_T(\hat{p},\hat{q}) - \tau_0(p_0,q_0)) \xrightarrow{d} N(0, \sigma_0^2 C_{\mathcal{A}_0}^{-1})$

⟨1⟩1. $(\hat{p},\hat{q}) \xrightarrow{p} (p_0,q_0)$
PROOF: By A.1 and A.2 and Kavalieris (1991, section 2).

⟨1⟩2. $\sqrt{T}(\hat{\tau}_T(p_0,q_0) - \tau_0(p_0,q_0)) \xrightarrow{d} N(0, \sigma_0^2 C_{\mathcal{A}_0}^{-1})$
⟨2⟩1. $\hat{C}_{(p_0,q_0)} \xrightarrow{a.s.} C_{\mathcal{A}_0}$
PROOF: By A.1, A.2 and Hannan & Rissanen (1982*b*, page 90).
⟨2⟩2. $C_{\mathcal{A}_0}$ is non-singular.
PROOF: By A.1, A.2 and Hannan & Rissanen (1982*b*, page 90).
⟨2⟩3. $\frac{1}{\sqrt{T}}\hat{X}'_{(p_0,q_0)}\hat{\epsilon} \xrightarrow{d} N(0, \sigma_0^2 C_{\mathcal{A}_0})$ where $\hat{\epsilon}$ is the least squares residual.
PROOF: By A.1, A.2 and Chan & Chen (2011, Lemma 3.5)
⟨2⟩4. Q.E.D.
PROOF: By ⟨2⟩1 - ⟨2⟩3 and the fact that $\hat{\tau}_T(p_0,q_0) = \tau + \hat{C}^{-1}_{(p_0,q_0)} \frac{1}{T} \hat{X}'_{(p_0,q_0)} \hat{\epsilon}$.

⟨1⟩3. $\lim_{T\to\infty} P(\sqrt{T}(\hat{\tau}_T(\hat{p},\hat{q}) - \tau_0(p_0,q_0)) = \sqrt{T}(\hat{\tau}_T(p_0,q_0) - \tau_0(p_0,q_0))) = 1$
PROOF: Step ⟨1⟩1 implies $\lim_{T\to\infty} P((\hat{p},\hat{q}) = (p_0,q_0)) = 1$ and Lemma 1 in Pötscher (1991).

⟨1⟩4. Q.E.D.
PROOF: By ⟨1⟩2 and ⟨1⟩3.

The proof of the second part is

ASSUME: Assumptions A.1 and A.2.
PROVE: $\hat{\tau}_{T,i} \xrightarrow{p} 0$ if $\tau_{0,i} = 0$

⟨1⟩1. CASE: $\tau_{0,i} = \alpha_{0,s},\ s \le p_0$ or $\tau_{0,i} = \beta_{0,s},\ s \le q_0$
PROOF: By part (i) of Lemma 3.2.

⟨1⟩2. CASE: $\tau_{0,i} = \alpha_{0,s},\ s > p_0$ or $\tau_{0,i} = \beta_{0,s},\ s > q_0$
⟨2⟩1. $(\hat{p},\hat{q}) \xrightarrow{p} (p_0,q_0)$
PROOF: By A.1 and A.2 and the proof in Kavalieris (1991).
⟨2⟩2. Q.E.D.
PROOF: By ⟨2⟩1 and the definition of $\hat{\tau}_{T,i}$.

⟨1⟩3. Q.E.D.
PROOF: The cases are exhaustive.

3.2.2 Convergence in Distribution

Now we are ready to proof the *first part* of Theorem 3.1:

ASSUME: Assumptions A.1 and A.2
PROVE:

$$\sqrt{T}(\tilde{\tau}_{\mathcal{A}_0,T} \to \tau_{\mathcal{A}_0,0}) \xrightarrow{d} N(0, \sigma_0^2 C_{\mathcal{A}_0}^{-1})$$

⟨1⟩1. DEFINE: 1. $\Psi_T : \mathbb{R}^R \to \mathbb{R}$

$$\Psi_T(u) := ||y - \hat{X}(\tau_0 + u/\sqrt{T})||^2 + \tilde{\lambda}_T \sum_{j=1}^{R} \hat{\omega}_j |\tau_{0,j} + \frac{u_j}{\sqrt{T}}|$$

2. $V_T : \mathbb{R}^R \to \mathbb{R}$; $V_T(u) := \Psi_T(u) - \Psi(0)$

⟨1⟩2. DEFINE: $\tilde{u}_T := \operatorname{argmin} V_T(u)$
PROOF: $V_T(u)$ is convex on $\mathbb{R}^R$.

⟨1⟩3. $\sqrt{T}(\tilde{\tau}_T - \tau_0) = \tilde{u}_T$

⟨2⟩1. $\tilde{u}_T = \operatorname{argmin}_u \Psi_T(u)$

⟨2⟩2. $\tilde{\tau}_T = \operatorname{argmin}_\tau \Psi_T(\sqrt{T}(\tau - \tau_0))$

⟨2⟩3. Q.E.D.

⟨1⟩4.

$$V_T(u) \xrightarrow{d} V(u)$$

$$V(u) := \begin{cases} u'_{\mathcal{A}_0} C_{\mathcal{A}_0} u_{\mathcal{A}_0} - 2u'_{\mathcal{A}_0} W_{\mathcal{A}_0} & \text{if } u_j = 0 \ (j \neq \mathcal{A}) \\ \infty & \text{otherwise} \end{cases}$$

where $W \sim N(0, \sigma_0^2 C)$.

⟨2⟩1.

$$V_T(u) = ||\hat{X}u/\sqrt{T}||^2 - 2\varepsilon' \hat{X}u/\sqrt{T} + \frac{\tilde{\lambda}_T}{\sqrt{T}} \sum_{j=1}^{R} \hat{\omega}_j \sqrt{T} \left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}| \right)$$

⟨3⟩1.

$$
\begin{aligned}
V_T(u) &= ||y - \hat{X}(\tau_0 + u/\sqrt{T})||^2 + \tilde{\lambda}_T \sum_{j=1}^{R} \hat{\omega}_j |\tau_{0,j} + \frac{u_j}{\sqrt{T}}| \\
&\quad - ||y - \hat{X}\tau_0||^2 - \tilde{\lambda}_T \sum_{j=1}^{R} \hat{\omega}_j |\tau_{0,j}| \\
&= \langle y - \hat{X}(\tau_0 + u/\sqrt{T}), y - \hat{X}(\tau_0 + u/\sqrt{T})\rangle - \langle y - \hat{X}\tau_0, y - \hat{X}\tau_0\rangle \\
&\quad + \tilde{\lambda}_T \sum_{j=1}^{R} \hat{\omega}_j \left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}| \right) \\
&= ||y||^2 + ||\hat{X}(\tau_0 + u/\sqrt{T})||^2 - 2y'\hat{X}(\tau_0 + u/\sqrt{T}) \\
&\quad - ||y||^2 - ||\hat{X}\tau_0||^2 + 2y'\hat{X}\tau_0 \\
&\quad + \tilde{\lambda}_T \sum_{j=1}^{R} \hat{\omega}_j \left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}| \right) \\
&= ||\hat{X}(\tau_0 + u/\sqrt{T})||^2 - ||\hat{X}\tau_0||^2 - 2y'\hat{X}u/\sqrt{T} \\
&\quad + \tilde{\lambda}_T \sum_{j=1}^{R} \hat{\omega}_j \left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}| \right) \\
&= ||\hat{X}\tau_0||^2 + 2\tau_0'\hat{X}'\hat{X}u/\sqrt{T} + ||\hat{X}u/\sqrt{T}||^2 - ||\hat{X}\tau_0||^2 - 2(X\tau_0 + \varepsilon)'\hat{X}u/\sqrt{T} \\
&\quad + \tilde{\lambda}_T \sum_{j=1}^{R} \hat{\omega}_j \left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}| \right)
\end{aligned}
$$

⟨3⟩2. Q.E.D.

PROOF: Simple rearranging of terms gives the desired result.

⟨2⟩2.

$$
\frac{\tilde{\lambda}_T}{\sqrt{T}} \sum_{j=1}^{R} \hat{\omega}_j \sqrt{T} \left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}| \right) \xrightarrow{p} \begin{cases} 0 & \text{if } u_j = 0 \ (j \in \mathbb{R}^R \backslash \mathcal{A}_0) \\ \infty & \text{otherwise} \end{cases}
$$

⟨3⟩1. $\frac{\tilde{\lambda}_T}{\sqrt{T}} \hat{\omega}_j \sqrt{T} \left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}| \right) \xrightarrow{p} \infty \quad (j \in \mathbb{R}^R \backslash \mathcal{A}_0), \quad u_j \neq 0$

⟨4⟩1. $\frac{\tilde{\lambda}_T}{\sqrt{T}} \hat{\omega}_j \sqrt{T} \left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}| \right) = \frac{\tilde{\lambda}_T}{\sqrt{T}} \hat{\omega}_j |u_j|$

⟨4⟩2. $\frac{\tilde{\lambda}_T}{\sqrt{T}} \hat{\omega}_j = \frac{\tilde{\lambda}_T}{\sqrt{T}} T^{\eta/2} \frac{1}{T^{\eta/2}|\hat{\tau}_j|^\eta}$

⟨4⟩3. $\frac{\tilde{\lambda}_T}{\sqrt{T}} T^{\eta/2} \to \infty$

PROOF: By assumption A.2 4.

⟨4⟩4. $T^{\eta/2}|\hat{\tau}_j|^\eta = (|T^{1/2}\hat{\tau}_j|)^\eta \xrightarrow{p} 0$

⟨5⟩1. $j \leq (p_0 + q_0)$

PROOF: By Lemma 3.2, (i).

⟨5⟩2. $(p_0 + q_0) < j \leq R$

⟨6⟩1. $(\hat{p}, \hat{q}) \xrightarrow{p} (p_0, q_0)$

⟨6⟩2. $\exists T_0 : P((\hat{p}, \hat{q}) = (p_0, q_0)) = 1, (T \geq T_0)$

⟨6⟩3. $\{(\hat{p}, \hat{q}) = (p_0, q_0))\} \subset \{|T^{1/2}\hat{\tau}_j| = 0\}$

⟨6⟩4. $\exists T_0 : P(|T^{1/2}\hat{\tau}_j| = 0) = 1, (T \geq T_0)$

⟨6⟩5. Q.E.D.

⟨5⟩3. Q.E.D.

⟨4⟩5. Q.E.D.

PROOF: By ⟨4⟩1 - ⟨4⟩4 and Slutsky's theorem.

⟨3⟩2. $\frac{\tilde{\lambda}_T}{\sqrt{T}}\hat{\omega}_j\sqrt{T}\left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}|\right) \xrightarrow{p} 0 \quad (j \in \mathcal{A}_0)$

⟨4⟩1.

$$\frac{\tilde{\lambda}_T}{\sqrt{T}}\hat{\omega}_j\sqrt{T}\left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}|\right) = \frac{\tilde{\lambda}_T}{\sqrt{T}}\hat{\omega}_j u_j \left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}|\right) / (\frac{u_j}{\sqrt{T}})$$

⟨4⟩2. $\tilde{\lambda}_T/\sqrt{T} \xrightarrow{p} 0$

⟨4⟩3. $\frac{1}{|\hat{\tau}_j|^\eta} \xrightarrow{p} \frac{1}{|\tau_j|^\eta}$

⟨4⟩4.

$$\left(|\tau_{0,j} + \frac{u_j}{\sqrt{T}}| - |\tau_{0,j}|\right) / (\frac{u_j}{\sqrt{T}}) \to \text{sign}(\tau_{0,j})$$

⟨4⟩5. Q.E.D.

PROOF: By Slutsky's theorem.

⟨3⟩3. Q.E.D.

⟨2⟩3.

$$||\hat{X}u/\sqrt{T}||^2 \xrightarrow{p} \begin{cases} u_{\mathcal{A}_0}^T C_{\mathcal{A}_0} u_{\mathcal{A}_0} & \text{if } u_j = 0 \; \forall j \neq \mathcal{A}_0 \\ u^T C u & \text{otherwise} \end{cases}$$

PROOF: $\hat{X}'\hat{X}/T \xrightarrow{p} C$ by A.1, A.2 and Hannan & Rissanen (1982*b*, page 90).

⟨2⟩4. $2\varepsilon'\hat{X}u/\sqrt{T} \xrightarrow{d} 2Wu$

⟨3⟩1. $X'\varepsilon/\sqrt{T} \xrightarrow{d} N(0, \sigma_0^2 C)$

⟨4⟩1.

$$X'\varepsilon/\sqrt{T} = (1/\sqrt{T})\sum_{t=1}^{T} x_t\varepsilon_t$$

⟨4⟩2. $(x_t\varepsilon_t)$ is a martingale difference series wrt. $\mathcal{F}_t = \sigma(\varepsilon_t, \varepsilon_{t-1}, \ldots)$.

⟨4⟩3. $var(x_t\varepsilon_t) = \sigma_0^2 E(x_t x_t') = \sigma_0^2 C < \infty$

PROOF: By A.1.5.

⟨4⟩4. $(x_t\varepsilon_t)$ is strictly stationary and ergodic

⟨4⟩5. Q.E.D.

PROOF: ⟨4⟩2 - ⟨4⟩4 imply that also every linear combination of $(x_t\varepsilon_t)$ is a strictly stationary and ergodic series. Then one can use the central limit theorem given in (Pötscher & Prucha 2001, Theorem 31) in conjunction with the Cramér-Wold device (e.g. Pötscher & Prucha 2001, Theorem 13) for the vector case.

⟨3⟩2. $\hat{X}'\varepsilon - X'\varepsilon \xrightarrow{p} 0$

⟨4⟩1.
$$\hat{X}'\varepsilon - X'\varepsilon = (\hat{X} - X)'\varepsilon$$
$$= \begin{pmatrix} 0 & \dots & 0 & \hat{\varepsilon}_{m-1} - \varepsilon_{m-1} & \dots & \hat{\varepsilon}_{m-q} - \varepsilon_{m-q} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \hat{\varepsilon}_{T-1} - \varepsilon_{T-1} & \dots & \hat{\varepsilon}_{T-q} - \varepsilon_{T-q} \end{pmatrix}' \varepsilon$$
$$= \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sum_{t=m}^{T} (\hat{\varepsilon}_{m-1} - \varepsilon_{m-1})\varepsilon_{m-1} \\ \vdots \\ \sum_{t=m}^{T} (\hat{\varepsilon}_{m-q} - \varepsilon_{m-q})\varepsilon_{m-q} \end{pmatrix}$$
PROOF: By the definition of X and $\hat{X}$.

⟨4⟩2. $\sum_{t=m}^{T} (\hat{\varepsilon}_{m-j} - \varepsilon_{m-j})\varepsilon_{m-j} = o_p(1) \quad (j \in \mathbb{N}_Q^0)$
PROOF: By Lemma 3.3 in Chan & Chen (2011).

⟨4⟩3. Q.E.D.
PROOF: Convergence of the single components implies convergence of the vector.

⟨3⟩3. Q.E.D.
PROOF: Steps ⟨3⟩1 and ⟨3⟩2 imply the result by a standard argument (e.g. Lütkepohl 2005, Proposition C.2, (2))

⟨2⟩5. Q.E.D.

⟨1⟩5. $\tilde{u}_{T,\mathcal{A}_0} \xrightarrow{d} C_{\mathcal{A}_0}^{-1} W_{\mathcal{A}_0}$

⟨2⟩1. $V_T(u)$ is convex

⟨2⟩2. V has a unique minimum with $\operatorname{argmin}(V) = C_{A_0}^{-1} W_{A_0}$

⟨2⟩3. Q.E.D.
By ⟨1⟩4, ⟨2⟩1, ⟨2⟩2 and Knight & Fu (2000, Proof of Theorem 2, page 1360) we have that
$$\operatorname{argmin}(V_T) \xrightarrow{d} \operatorname{argmin}(V)$$

⟨1⟩6. Q.E.D.
PROOF: By ⟨1⟩3, ⟨1⟩5 and $(\sqrt{T}(\tilde{\tau}_T - \tau_0))_{\mathcal{A}_0} = \sqrt{T}(\tilde{\tau}_{\mathcal{A}_0,T} - \tau_{\mathcal{A}_0,0})$.

3.2.3 Selection Consistency

The proof of the second part of Theorem 3.1 is identical to the proof in Chan & Chen (2011) and is therefore omitted.

4 Monte Carlo Simulations

In this section, we investigate the properties of the proposed algorithms using various measures. The first concern is whether the methods are able to find the true model - at least in large samples. Thus, the number of times the correct model is chosen is one of our metrics. I estimate this quantity with the Monte Carlo average, $R^{-1} \sum_{r=1}^{R} 1(\hat{\mathcal{A}}_r = \mathcal{A}_0)$ (`APC`), where r is an index for the Monte

Carlo replication and $\hat{\mathcal{A}}_r$ is the estimated set of indices. Throughout, the total number of simulations is $R = 1000$. Since these models are predominantly used for forecasting, we estimate the mean squared prediction error and the mean absolute prediction error with their corresponding averages, that is

$$\begin{aligned} \texttt{ASE} &:= R^{-1} \sum_{r=1}^{R} (y_{T+1} - \hat{y}_{T+1|T})^2 \\ \texttt{AAE} &:= R^{-1} \sum_{r=1}^{R} |y_{T+1} - \hat{y}_{T+1|T}|, \end{aligned}$$

where $\hat{y}_{T+1|T}$ stands for a generic forecast based on information up to time T.

The first four data-generating processes (DGPs) are taken from Chan & Chen (2011). The last four DGPs are versions of the first four with smaller lag orders.

DGP I:	$(1 - 0.8B)(1 - 0.7B^6)y_t = \varepsilon_t$
DGP II:	$(1 - 0.8B)(1 - 0.7B^6)y_t = (1 + 0.8B)(1 + 0.7B^6)\varepsilon_t$
DGP III:	$y_t = (1 + 0.8B)(1 + 0.7B^6)\varepsilon_t$
DGP IV:	$y_t = (1 - 0.6B - 0.8B^{12})\varepsilon_t$
DGP V:	$(1 - 0.8B)(1 - 0.7B^3)y_t = \varepsilon_t$
DGP VI:	$(1 - 0.8B)(1 - 0.7B^3)y_t = (1 + 0.8B)(1 + 0.7B^3)\varepsilon_t$
DGP VII:	$y_t = (1 + 0.8B)(1 + 0.7B^3)\varepsilon_t$
DGP VIII:	$y_t = (1 - 0.6B - 0.8B^6)\varepsilon_t$

Table 3: Simulated DGPs

where $(\varepsilon_t) \sim i.i.d.\, N(0, 1)$. For each DGP, we simulate 1000 time series of length $T = 120, 240, 360$.

The first four DGPs have in common that relatively few parameters describe them but the associated lags are quite long such that only an extensive search could potentially discover the relevant regressors. The methods, however, do not exploit the seasonal patterns of the data.

4.1 Implementation Details

All algorithms work on the mean-adjusted data even though the mean adjustment was omitted from the presentation. The choice of $f(T)$ is $f(T) = [10 \cdot log_{10}(T)]$ in accordance with the Monte Carlo study in Chan & Chen (2011). We also set $P = Q = 14$ as in Chan & Chen (2011). The computations were done in MATLAB (The Mathworks Inc. 2015) and OCTAVE (John W. Eaton, David Bateman, S\oren Hauberg, and Rik Wehbrin 2014). I also use the `glmnet` code of Qian, J., Hastie, T., Friedman, J., Tibshirani, R. and Simon, N. (2014) for the computation of the adaptive lasso. All code can be found on the homepage of the author.

4.2 Results

The results are expressed as average ranks in table (4). The ranks are computed for each combination of a particular DGP, a sample size and a criterion. For

example, all methods are ranked according to the computed average squared (prediction) error for DGP 1, for $T = 120$. These rankings are averaged over all DGPs and sample sizes in the columns with the heading *Overall* - for each criterion separately. The other columns show averages computed over all eight DGPs for the indicated sample size. Therefore, a low number indicates that a certain method is ranked among the better methods.

First, the forecasting measures rank the methods quite differently than the probability of getting the correct model. Which forecasting measure is employed is, however, not very important for the ranking of the methods.

For choosing the correct model, the methods **NT2PQ1**, **NT2PQ2** and **NT2PQ3** perform best overall. There is little variation in the ranking throughout the considered sample sizes, though it seems that when the sample size is sufficiently high then the methods **NT4PQ1**, **NT4PQ2** and **NT4PQ3** become competitive or even preferable. All methods with **PQ4** perform quite badly.

For the forecasting measures, the methods **NT1PQ1** and **NT1PQ2** perform best overall. The average rankings differ, however, when they are computed over different sample sizes. For low sample sizes, **NT1PQ1** and **NT1PQ2** perform quite well. For medium sample sizes **NT4PQ1**, **NT4PQ2** and **NT4PQ3** perform best. For larger sample sizes, **NT1PQ2** and **NT2PQ2** perform best. Overall, taking the time variation into account, one can recommend the combination **NT1PQ2**. Interestingly, the method performs reasonably when the purpose of the modeling is finding the true model. Thus **NT1PQ2** seems overall a good default choice if one insists one a default that is independent of the modeling purpose.

5 Conclusion

In this paper, I investigated the adaptive lasso estimation method for ARMA models. I proposed an algorithm and derived its large sample properties. I also examined how one could implement the algorithm and found that it was possible to find good default values.

Future research could investigate more thoroughly the merits of the method in empirical research. Furthermore, the method can be extended in various directions. Among these, the extension to a penalized likelihood method would be most interesting.

Table 4: Average Ranks

	Overall			$T = 120$			$T = 240$			$T = 360$		
	ASE	**AAE**	**APC**	**ASE**	**AAE**	**APC**	**ASE**	**AAE**	**APC**	**ASE**	**AAE**	**APC**
NT1PQ1	5.67	5.83	5.88	4.62	4.75	6.25	6.50	7.00	5.81	5.88	5.75	5.56
NT1PQ2	5.58	5.50	5.92	5.00	4.88	6.38	6.81	6.69	6.00	4.94	4.94	5.38
NT1PQ3	6.83	6.96	5.52	6.50	6.75	5.75	7.56	7.56	5.31	6.44	6.56	5.50
NT1PQ4	6.44	6.77	10.75	7.06	7.31	10.50	5.62	6.38	10.62	6.62	6.62	11.12
NT2PQ1	7.21	6.94	4.17	7.19	6.69	4.06	7.25	6.88	3.56	7.19	7.25	4.88
NT2PQ2	6.79	6.29	4.60	6.62	6.50	4.69	8.00	6.75	4.31	5.75	5.62	4.81
NT2PQ3	7.21	6.94	4.17	7.19	6.69	4.06	7.25	6.88	3.56	7.19	7.25	4.88
NT2PQ4	6.40	6.73	10.75	7.44	7.69	10.50	5.38	6.12	10.62	6.38	6.38	11.12
NT4PQ1	6.06	6.27	4.96	5.44	5.44	5.06	5.50	6.00	5.25	7.25	7.38	4.56
NT4PQ2	6.46	6.12	6.19	7.12	7.25	6.62	5.62	4.75	7.06	6.62	6.38	4.88
NT4PQ3	6.06	6.27	4.96	5.44	5.44	5.06	5.50	6.00	5.25	7.25	7.38	4.56
NT4PQ4	7.29	7.38	10.15	8.38	8.62	9.06	7.00	7.00	10.62	6.50	6.50	10.75

Note: Average ranks for the different method when evaluated according to the average squared error (ASE), the average absolute error (AAE) or the average probability of choosing the correct model (APC). The methods and DGPs are explained in the text.

References

Chan, K.-S. & Chen, K. (2011), 'Subset ARMA selection via the adaptive lasso', *Stat. Interface* **4**(2), 197–205.
URL: *http://dx.doi.org/10.4310/sii.2011.v4.n2.a14*

Hannan, E. J. & Kavalieris, L. (1984), 'A method for autoregressive-moving average estimation', *Biometrika* **72**(2), 273–280.

Hannan, E. J. & Rissanen, J. (1982*a*), 'Errata to recursive estimation of mixed autoregressive-moving average order', *Biometrika* **69**, 1–17.

Hannan, E. J. & Rissanen, J. (1982*b*), 'Recursive estimation of mixed Autoregressive-Moving average order', *Biometrika* **69**(1), 81–94.
URL: *http://www.jstor.org/stable/2335856*

John W. Eaton, David Bateman, S\oren Hauberg, and Rik Wehbrin (2014), *GNU Octave version 3.8.1 manual: a high-level interactive language for numerical computations*, CreateSpace Independent Publishing Platform.
URL: *http://www.gnu.org/software/octave/doc/interpreter*

Kavalieris, L. (1991), 'A note on estimating autoregressive-moving average order', *Biometrika* **78**(4), 920–922.
URL: *http://biomet.oxfordjournals.org/content/78/4/920.abstract*

Knight, K. & Fu, W. (2000), 'Asymptotics for Lasso-Type estimators', *Ann. Stat.* **28**(5), 1356–1378.
URL: *http://www.jstor.org/stable/2674097*

Lamport, L. (1995), 'How to write a proof', *Am. Math. Mon.* **102**(7), 600–608.
URL: *http://www.jstor.org/stable/2974556*

Lamport, L. (2012), 'How to write a 21st century proof', *J. Fixed Point Theory Appl.* **11**(1), 43–63.
URL: *http://link.springer.com/article/10.1007/s11784-012-0071-6*

Lütkepohl, H. (2005), *New introduction to multiple time series analysis*, Springer Science & Business Media.

Medeiros, M. C. & Mendes, E. (2012), 'Estimating high-dimensional time series models', *CREATES Research Paper* **37**.

Mendes, E. F. & Medeiros, M. C. (2015), Adaptive lasso estimation for ARDL models with garch innovations, Technical report.

Poskitt, D. S. (1994), 'A note on autoregressive modeling', *Econometric Theory* **10**(5), 884–899.
URL: *http://www.jstor.org/stable/3532858*

Pötscher, B. M. (1991), 'Effects of model selection on inference', *Econometric Theory* **7**(2), 163–185.
URL: *http://www.jstor.org/stable/3532042*

Pötscher, B. M. & Prucha, I. R. (2001), Basic elements of asymptotic theory, *in* B. H. Baltagi, ed., 'A companion to theoretical econometrics', Blackwell Publishing Ltd, pp. 201–229.

Qian, J., Hastie, T., Friedman, J., Tibshirani, R. and Simon, N. (2014), 'Glmnet for Matlab'.
URL: *https://web.stanford.edu/ hastie/glmnet_matlab/*

Ren, Y. & Zhang, X. (2010), 'Subset selection for vector autoregressive processes via adaptive lasso', *Stat. Probab. Lett.* **80**(23–24), 1705–1712.
URL: *http://www.sciencedirect.com/science/article/pii/S0167715210002075*

The Mathworks Inc. (2015), 'MATLAB'.
URL: *https://ch.mathworks.com/*

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society. Series B* **58**(1), 267–288.

Zou, H. (2006), 'The adaptive lasso and its oracle properties', *J. Am. Stat. Assoc.* **101**(476), 1418–1429.
URL: *http://www.jstor.org/stable/27639762*