Optimal Estimation of Multivariate ARMA Models

Martha White, Junfeng Wen, Michael Bowling and Dale Schuurmans
Department of Computing Science, University of Alberta, Edmonton AB T6G 2E8, Canada
{whitem,junfeng.wen,mbowling,daes}@ualberta.ca

Abstract

Autoregressive moving average (ARMA) models are a fundamental tool in time series analysis that offer intuitive modeling capability and efficient predictors. Unfortunately, the lack of globally optimal parameter estimation strategies for these models remains a problem: application studies often adopt the simpler autoregressive model that can be easily estimated by maximizing (a posteriori) likelihood. We develop a (regularized, imputed) maximum likelihood criterion that admits efficient global estimation via structured matrix norm optimization methods. An empirical evaluation demonstrates the benefits of globally optimal parameter estimation over local and moment matching approaches.

Introduction

A central problem in applied data analysis is time series modeling—estimating and forecasting a discrete-time stochastic process—for which the autoregressive moving average (ARMA) and stochastic ARMA (Thiesson et al. 2012) are fundamental models. An ARMA model describes the behavior of a linear dynamical system under latent Gaussian perturbations (Brockwell and Davis 2002; Lütkepohl 2007), which affords intuitive modeling capability, efficient forecasting algorithms, and a close relationship to linear Gaussian state-space models (Katayama 2006, pp.5-6).

Unfortunately, estimating the parameters of an ARMA model from an observed sequence is a computationally difficult problem: no efficient algorithm is known for computing the parameters that maximize the marginal likelihood of the observed data in an ARMA, stochastic ARMA or linear Gaussian state-space model. Consequently, heuristic local estimators are currently deployed in practice (Hannan and Kavalieris 1984; Durbin 1960; Bauer 2005; Lütkepohl 2007; Thiesson et al. 2012), none of which provide a guarantee of how well the globally optimal parameters are approximated. For estimating linear Gaussian state-space models, it has been observed that local maximization of marginal likelihood tends to find local optima that yield poor results (Katayama 2006, Sec. 1.3).

In response to the difficulty of maximizing marginal likelihood, there has been growing interest in *method of moments* based estimators for state-space models, which offer computationally efficient estimation strategies and sound consistency properties (Andersson 2009; Hsu, Kakade, and Zhang 2012; Anandkumar, Hsu, and Kakade 2012). For ARMA models, the most applicable such estimators are the subspace identification methods for estimating state-space models (Katayama 2006; Moonen and Ramos 1993; Van Overschee and De Moor 1994; Viberg 1995; Song et al. 2010; Boots and Gordon 2012). The statistical efficiency of moment matching, however, generally does not match that of maximum likelihood, which is known to be asymptotically efficient under general conditions (Cramér 1946, Ch. 33). In fact, evidence suggests that the statistical efficiency of current moment matching estimators is quite weak (Foster, Rodu, and Ungar 2012; Zhao and Poupart 2014).

In this paper, we develop a tractable approach to maximum likelihood parameter estimation for stochastic multivariate ARMA models. To efficiently compute a globally optimal estimate, the problem is re-expressed as a regularized loss minimization, which then allows recent algorithmic advances in sparse estimation to be applied (Shah et al. 2012; Candes et al. 2011; Bach, Mairal, and Ponce 2008; Zhang et al. 2011; White et al. 2012). Although there has been recent progress in global estimation for ARMA, such approaches have either been restricted to single-input single-output systems (Shah et al. 2012), estimating covariance matrices for scalar ARMA (Wiesel, Bibi, and Globerson 2013) or using AR to approximate a scalar ARMA model (Anava et al. 2013). By contrast, this paper offers the first efficient maximum likelihood approach to estimating the parameters of a stochastic multivariate ARMA(p, q) model. This convex optimization formulation is general, enabling generalized distributional assumptions and estimation on multivariate data, which has been much less explored than scalar ARMA. An experimental evaluation demonstrates that globally optimal parameters under the proposed criterion yield superior forecasting performance to alternative estimates, including local minimization for ARMA estimation and moment-based estimation methods for state-space models.

Background

An ARMA model is a simple generative model of the form depicted in Figure 1(a), where the innovation variables, $\varepsilon_t \in \mathbb{R}^n$, are assumed to be i.i.d. Gaussian, $\mathcal{N}(0, \Sigma)$, and

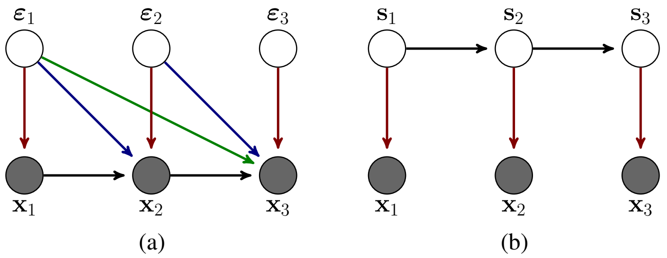


Figure 1: Graphical models depicting the dependence structure of **(a)** an ARMA$(1, 2)$ model, **(b)** a state-space model. These models are equivalent if the state-space model is in observability canonical form (Benveniste, Metivier, and Priouret 2012, Sec. 6.2.1). Distinct methods are used for estimation depending on whether the variables are discrete or continuous.

the observable variables, $\mathbf{x}_t \in \mathbb{R}^n$, are assumed to be generated by the linear relationship

$$\mathbf{x}_t = \sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i} + \sum_{j=1}^{q} B^{(j)}\boldsymbol{\varepsilon}_{t-j} + \boldsymbol{\varepsilon}_t. \tag{1}$$

An ARMA(p, q) model is thus parameterized by $A^{(1)}, ..., A^{(p)} \in \mathbb{R}^{n\times n}$, $B^{(1)}, ..., B^{(q)} \in \mathbb{R}^{n\times n}$, and a positive semi-definite matrix Σ; which we simply collect as $\Theta = (\{A^{(i)}\}, \{B^{(j)}\}, \Sigma)$.[1]

One classical motivation for ARMA models arises from the Wold representation theorem (Wold 1938), which states that any stationary process can be represented as an infinite sum of innovations plus a deterministic process that is a projection of a current observation onto past observations: $\mathbf{x}_t = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \ldots) + \sum_{j=0}^{\infty} B^{(j)}\boldsymbol{\varepsilon}_{t-j}$. Thus the autoregressive component of an ARMA model is often motivated as a more parsimonious representation of this Wold representation (Scargle 1981).

Time series models are used primarily for forecasting: Given an ARMA model with parameters Θ, the value of a future observation $\mathbf{x}_{T+h}$ can be predicted from an observed history $\mathbf{x}_{1:T}$ by evaluating $E[\mathbf{x}_{T+h}|\mathbf{x}_{1:T}, \Theta]$. The key advantage of ARMA is that such forecasts can be computed efficiently; see Appendix F for additional details.

Although forecasting is efficient, the problem of estimating the parameters of an ARMA model raises significant computational challenges, which provides the main focus of this paper. To begin, consider the marginal log-likelihood of an observed history $\mathbf{x}_{1:T}$ given a set of parameters Θ:

$$\log p(\mathbf{x}_{1:T}|\Theta) \quad = \quad \sum_{t=1}^{T} \log p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \Theta). \tag{2}$$

Despite the fact that the conditional expectation $E[\mathbf{x}_t|\mathbf{x}_{1:t-1}, \Theta]$ can be computed efficiently, the quantity $\log p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \Theta)$ is not concave in Θ (Mauricio 1995; Lütkepohl 2007, Sec. 12.2-3), which suggests that maximizing the marginal likelihood is a hard computational problem. Another source of difficulty is that ARMA is a latent variable model, hence marginalizing over the unobserved innovations $\boldsymbol{\varepsilon}_{1:T}$ might also be problematic. Given innovations $\boldsymbol{\varepsilon}_{1:T} = [\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_T]$, however, $p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \boldsymbol{\varepsilon}_{1:t-1}, \Theta)$ is a simple Gaussian with mean

$$\boldsymbol{\mu}_t \quad = \textstyle\sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i} + \sum_{j=1}^{q} B^{(j)}\boldsymbol{\varepsilon}_{t-j} \tag{3}$$

and covariance Σ. To obtain such a simplified form, we will first characterize the entire data likelihood in terms of the innovations which enables application of the widely used expectation-maximization algorithm.

Lemma 1. *For an auxiliary density $q(\cdot)$ over $\boldsymbol{\varepsilon}_{1:T}$, and entropy $H(q(\cdot))$, it follows that (proof given in Appendix A):*

$$\begin{aligned}\log p(\mathbf{x}_{1:T}|\Theta) &= \log \int p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta)\, d\boldsymbol{\varepsilon}_{1:T} \\ &= \max_{q(\cdot)} \int q(\boldsymbol{\varepsilon}_{1:T}) \log p(\mathbf{x}_{1:T}\boldsymbol{\varepsilon}_{1:T}|\Theta)\, d\boldsymbol{\varepsilon}_{1:T} + H(q(\cdot)).\end{aligned} \tag{4}$$

The maximum likelihood problem can now be re-expressed as $\min_\Theta \min_{\{q(\cdot)\}} -\log p(\mathbf{x}_{1:T}|\Theta)$ where in a standard EM algorithm, the M step would consist of optimizing Θ given $\{q(\cdot)\}$, and the E step would consist of (implicitly) optimizing $\{q(\cdot)\}$ given Θ (Neal and Hinton 1998). A standard variant of the log likelihood in (4) can then be obtained simply by dropping the entropy regularizer $H(q(\cdot))$. This leads to the minimization selecting a Dirac delta distribution on $\boldsymbol{\varepsilon}_{1:T}$ and a far simpler formulation, sometimes known as "hard EM" or "Viterbi EM" (Brown et al. 1993):

$$\begin{aligned}&\min_\Theta \min_{\boldsymbol{\varepsilon}_{1:T}} -\log p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta) \\ &= \min_\Theta \min_{\boldsymbol{\varepsilon}_{1:T}} -\sum_{t=1}^{T} \Big[\log p(\boldsymbol{\varepsilon}_t|\mathbf{x}_{1:t}, \boldsymbol{\varepsilon}_{1:t-1}, \Theta) \\ &\qquad\qquad + \log p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \boldsymbol{\varepsilon}_{1:t-1}, \Theta)\Big].\end{aligned} \tag{5}$$

This formulation suggests an approach where one successively imputes values $\boldsymbol{\varepsilon}_{1:T}$ for the unobserved innovation variables, then optimizes the parameters. Interestingly, $p(\boldsymbol{\varepsilon}_t|\mathbf{x}_{1:t}, \boldsymbol{\varepsilon}_{1:T-1}, \Theta)$ is a Dirac delta distribution; $\boldsymbol{\varepsilon}_t$ must be the residual, $\boldsymbol{\varepsilon}_t = \mathbf{x}_t - \boldsymbol{\mu}_t$, otherwise the loss becomes unbounded. This distribution, therefore, imposes a constraint on the minimization. To maximize likelihood under this constraint, we optimize

$$\min_\Theta \min_{\boldsymbol{\varepsilon}_{1:T}:\boldsymbol{\varepsilon}_t = \mathbf{x}_t - \boldsymbol{\mu}_t} -\sum_{t=1}^{T} \log p(\mathbf{x}_t|, \mathbf{x}_{1:t\text{–}1}, \boldsymbol{\varepsilon}_{1:t\text{–}1}, \Theta) \tag{6}$$

$$= \min_{\substack{\Theta, \boldsymbol{\varepsilon}_{1:T}: \\ \boldsymbol{\varepsilon}_t = \mathbf{x}_t - \boldsymbol{\mu}_t}} \frac{T}{2} \log((2\pi)^n|\Sigma|) + \frac{1}{2}\sum_{t=1}^{T} \left\| \Sigma^{-\frac{1}{2}}(\mathbf{x}_t - \boldsymbol{\mu}_t) \right\|^2. \tag{7}$$

Unfortunately, this optimization raises an important challenge. Due to the direct interaction between Σ, B and $\boldsymbol{\varepsilon}_{1:T}$ the final form of the problem (7) is still not convex in the parameters Θ and $\boldsymbol{\varepsilon}_{1:T}$ jointly. A typical strategy is therefore to first estimate the innovations $\boldsymbol{\varepsilon}_{1:T}$ directly from data, for example by using the errors of a learned autoregressive model, then observing that with the innovation variables fixed and Σ approximated from the innovations, the problem becomes convex in Θ (Hannan and Kavalieris 1984; Lütkepohl 2007). Another more contemporary approach is to convert the ARMA model into a state-space model (see

[1] Note that we use the term ARMA to include vector ARMA, with no restriction to scalar time series.

Figure 1(b)) and then solve for parameters in that model using system identification approaches (Bauer 2005). Though this has been an important advance for efficient ARMA estimation, these approaches still result in local minima.

Regularized ARMA modeling

To develop a likelihood based criterion that admits efficient global optimization, we begin by considering a number of extensions to the ARMA model. First, notice that the ARMA model in (1) can be equivalently formulated by introducing a $B^{(0)}$ and taking $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\boldsymbol{\mu} = \mathbf{0}, \Sigma = I)$, giving $B^{(0)}\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, B^{(0)}B^{(0)\prime})$. Second, following Thiesson et al. (2012), an independent noise term $\boldsymbol{\eta}_t$ can be added to (1) to obtain the *stochastic ARMA* model

$$\mathbf{x}_t = \sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i} + \sum_{j=0}^{q} B^{(j)}\boldsymbol{\varepsilon}_{t-j} + \boldsymbol{\eta}_t. \qquad (8)$$

A key challenge in estimating the parameters of a classical ARMA model (1) is coping with the deterministic constraint that $\boldsymbol{\eta}_t = 0$, which forces the innovations $\boldsymbol{\varepsilon}_t$ to match the residuals (7). The stochastic ARMA model (8) relaxes this assumption by allowing $\boldsymbol{\eta}_t$ to be generated by a smooth exponential family distribution, such as $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, Q_t)$ for covariance Q_t; smaller Q_t yields a closer approximation to the original ARMA model. Thiesson et al. (2012) have shown that expectation-maximization (EM) updates are only meaningful for non-zero Q_t; else, EM stops after one iteration. EM is not however guaranteed to find a globally optimal parameter estimate for the stochastic ARMA model. A key advantage of this model, however, is that it allows a convenient re-expression of the marginal log-likelihood (6) by applying the chain rule in the opposite order for $\mathbf{x}_t$ and $\boldsymbol{\varepsilon}_t$:

$$\begin{aligned}(6) = &\min_{\Theta}\min_{\boldsymbol{\varepsilon}_{1:T}} \\ &-\sum_{t=1}^{T}\Big[\log p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \boldsymbol{\varepsilon}_{1:t}, \Theta) + \log p(\boldsymbol{\varepsilon}_t|\mathbf{x}_{1:t-1}, \boldsymbol{\varepsilon}_{1:t-1}, \Theta)\Big] \\ &= \min_{\Theta}\min_{\boldsymbol{\varepsilon}_{1:T}} -\sum_{t=1}^{T}\Big[\log p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \boldsymbol{\varepsilon}_{1:t}, \Theta) + \log p(\boldsymbol{\varepsilon}_t|\Theta)\Big],\end{aligned}$$

since $\boldsymbol{\varepsilon}_t$ is independent of past innovations and data without $\mathbf{x}_t$. Furthermore, $p(\boldsymbol{\varepsilon}_t|\Theta) = p(\boldsymbol{\varepsilon}_t)$ since the covariance was moved into $B^{(0)}$ to make $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, I)$, yielding

$$\begin{aligned}-\sum_{t=1}^{T}\log p(\boldsymbol{\varepsilon}_t) &= \frac{nT}{2}\log(2\pi) + \frac{1}{2}\sum_{t=1}^{T}\|\boldsymbol{\varepsilon}_t\|_2^2 \\ &= \frac{nT}{2}\log(2\pi) + \frac{1}{2}||\mathcal{E}||_F^2 \qquad (9)\end{aligned}$$

for $\mathcal{E} = \boldsymbol{\varepsilon}_{1:T}$. The constant is ignored in the optimization.

Third, rather than merely consider a maximum likelihood objective, we can consider the maximum a posteriori (MAP) estimate given by the introduction of a prior $\log p(\Theta)$ over the model parameters $\Theta = (A, B, \Sigma)$. Since the parameters A and B do not typically have distributional assumptions, we view the choice of priors rather as regularizers:

$$-\log p(\Theta) = -\log p(A) - \log p(B) = R(A) + G(B),$$

for convex functions R and G. Any convex regularizer on A is acceptable. The choice of regularizer on B is more subtle, since for any s, $B\mathcal{E} = (Bs^{-1})(s\mathcal{E})$: $G(B)$ is required to prevent B from being scaled up, pushing $||\mathcal{E}||_F^2$ to zero. We consider $G(B) = ||B||_F^2$ for $B = [B^{(0)}; \ldots; B^{(q)}]$, which effectively controls the size of B and, importantly, also results in a global reformulation given in Theorem 2.

Finally, as noted, we can consider any natural exponential family distribution for $\boldsymbol{\eta}_t$ rather than merely assuming Gaussian. The negative log-likelihood for such a distribution corresponds to a regular Bregman divergence (see Appendix D), allowing one to write the final estimation criterion in terms of a convex loss function $L(\cdot|\mathbf{x}_t)$ as

$$\begin{aligned}\min_{A,B,\mathcal{E}} \sum_{t=1}^{T} L\Big(\sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i} + \sum_{j=0}^{q} B^{(j)}\boldsymbol{\varepsilon}_{t-j} \;\Big|\; \mathbf{x}_t\Big) \qquad (10) \\ +\alpha\Big(||\mathcal{E}||_F^2 + G(B)\Big) + \gamma R(A),\end{aligned}$$

for regularization parameters α and γ.

Efficient parameter estimation

Although $L(\cdot|\mathbf{x}_t)$ is convex, (10) is not jointly convex due to the coupling between B and $\mathcal{E}$. However, using recent insights from matrix factorization (Bach, Mairal, and Ponce 2008; Dudik, Harchaoui, and Malick 2012; White et al. 2012) one can reformulate (10) as a convex optimization.

For fixed autoregressive parameters, A, let $L_{A,t}(\mathbf{z}) = L(\mathbf{z} + \sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i} \mid \mathbf{x}_t) + \gamma R(A)$, which is still convex in $\mathbf{z}$.[2] By introducing the change of variables, $Z = B\mathcal{E}$, the optimization over B and $\mathcal{E}$ given A can be written as

$$\begin{aligned}&\min_{A,B,\mathcal{E}} \sum_{t=1}^{T} L_{A,t}\Big(\sum_{j=0}^{q} B^{(j)}\mathcal{E}_{:,t-j}\Big) + \alpha\Big(||\mathcal{E}||_F^2 + G(B)\Big) \qquad (11) \\ &= \min_{A,Z} \sum_{t=1}^{T} L_{A,t}\Big(\sum_{j=0}^{q} Z^{(j)}_{:,t-j}\Big) + \alpha \min_{\substack{B,\mathcal{E} \\ B\mathcal{E}=Z}} \Big(||\mathcal{E}||_F^2 + G(B)\Big).\end{aligned}$$

Surprisingly, this objective can be re-expressed in a convex form since

$$|||Z||| = \min_{B,\mathcal{E}: B\mathcal{E}=Z} \Big(||\mathcal{E}||_F^2 + G(B)\Big) \qquad (12)$$

defines an induced norm on Z (established in Theorems 2 and 3 below) allowing (11) to be equivalently expressed as:[3]

$$\sum_{t=1}^{T} L\Big(\sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i} + \sum_{j=0}^{q} Z^{(j)}_{:,t-j} \;\Big|\; \mathbf{x}_t\Big) + \alpha|||Z||| + \gamma R(A)$$

Therefore, one can alternate between A and Z to obtain a globally optimal solution, then recover B and $\mathcal{E}$ from Z, as discussed in the next section. Proofs for the following two theorems are provided in Appendix B.

Theorem 2. *The regularized ARMA(p, q) estimation problem for $G(B) = ||B||_F^2$ is equivalent to*

$$(11) = \min_{A,Z} \sum_{t=1}^{T} L_{A,t}\Big(\sum_{j=0}^{q} Z^{(j)}_{:,t-j}\Big) + 2\alpha||Z||_{tr}$$

with a singular value decomposition recovery: $Z = U\Sigma V'$ giving $B = U\sqrt{\Sigma}$ and $\mathcal{E} = \sqrt{\Sigma}V'$.

[2] Proved in Lemma 4, Appendix B for completeness

[3] Proved in Corollary 5, Appendix B for completeness.

The estimation problem is more difficult with the second choice of regularizer; to get an exact formulation, we need to restrict $q = 1$, giving $Z = \begin{bmatrix} B^{(0)} \\ B^{(1)} \end{bmatrix} \mathcal{E}$.

Theorem 3. *The regularized ARMA($p, 1$) estimation problem for $G(B) = \max_{j=0,\ldots,q} ||B^{(j)}||_F^2$ is equivalent to*

$$(11) = \min_{A,Z} \sum_{t=1}^{T} L_{A,t}\Big(\sum_{j=0}^{q} Z^{(j)}_{:,t-j} \Big) + \max_{0 \leq \rho \leq 1} ||W_\rho^{-1} Z||_{tr} \quad (13)$$

where $W_\rho := \begin{bmatrix} 1/\sqrt{\rho}\, I_n & 0 \\ 0 & 1/\sqrt{1-\rho}\, I_n \end{bmatrix}$. *Moreover* $||W_\rho^{-1} Z||_{tr}$ *is concave in* ρ *over* $[0, 1]$*, enabling an efficient line-search.*

Identifiability and optimal parameter recovery

One desirable ideal for estimation is identifiability: being able to “identify” parameters uniquely. For a strictly convex loss function, L, the convex regularized ARMA optimization in (13) produces a unique moving average variable, Z. This identifiable matrix is sufficient for correctly estimating the autoregressive parameters for the ARMA model, which can be all that is required for forecasting in expectation.

It might be desirable, however, to recover the factors B and $\mathcal{E}$ to gain further insight into the nature of the time series. Unfortunately, unlike Z, the factors that satisfy $B\mathcal{E} = Z$ are not unique. Worse, if one simply recovers any B and $\mathcal{E}$ that satisfies $B\mathcal{E} = Z$, the recovered innovations $\mathcal{E}$ need not be Gaussian distributed. This issue can be addressed via a careful recovery procedure that finds a particular pair B and $\mathcal{E}$ with the same regularization penalty as Z. Let

$$\text{Factors}(Z) = \big\{ (B, \mathcal{E}) : B\mathcal{E} = Z \text{ and } G(B) + ||\mathcal{E}||_F^2 = |||Z||| \big\}$$

This set of solutions satisfies the desired distributional properties, but is invariant under scaling and orthogonal transformations: for any $(B, \mathcal{E}) \in$ Factors(Z), (i) for $s = G(B)/||\mathcal{E}||_F$, $(B(1/s), s\mathcal{E}) \in$ Factors(Z) and (ii) for any orthogonal matrix $P \in \mathbb{R}^{n \times n}$, $(BP, P'\mathcal{E}) \in$ Factors(Z) since the Frobenius norm is invariant under orthogonal transformations. When $G(B) = ||\mathcal{E}||_F$, a solution from Factors(Z) can be computed from the singular value decomposition of Z, as shown in Theorem 2. When $G(B) = \max_j ||B^{(j)}||_F^2$, a boosting approach can be used for recovery; see Appendix C for details as well as a discussion on recovering Laplacian instead of Gaussian innovations.

Computational complexity

The overall estimation procedure is outlined in Algorithm 1 for $G(B) = ||B||_F^2$; the approach is similar for the other regularizer, but with an outer line search over ρ. The computational complexity is governed by the matrix multiplication to compute the autoregressive and moving average components, and by the use of the singular value decomposition. The matrix multiplication for AX_p is $O(Tpn^2)$, which dominates the cost of computing the autoregressive loss, corresponding to ARloss in Algorithm 1. For the moving average loss, MALoss in Algorithm 1, the thin SVD of $Z \in \mathbb{R}^{qn \times T}$ has complexity $O(Tq^2n^2)$ and the multiplication of UV' is also $O(Tq^2n^2)$. Thus, each call to ARloss is $O(Tpn^2)$ and each call to MAloss is $O(Tq^2n^2)$. The initial solution of A, which involves solving a basic vector autoregressive model, is $i_1 O(Tpn^2)$ where i_1 is the number of iterations (typically small). For i_2 the number of iterations between A and B: RARMA cost = VAR cost + $i_2(O(Tpn^2) + O(Tq^2n^2)) = (i_1 + i_2)O(Tpn^2) + i_2 O(Tq^2n^2) \approx O(T(p + q^2)n^2)$.

Algorithm 1 RARMA(p, q)

```
Input: $X, p, q, \alpha, \gamma$
Output: $A, B, \mathcal{E}$
$X_p = \mathbf{0}$      // History matrix, $X_p \in \mathbb{R}^{np \times T}$
for $i = 1, \ldots, p$, $X_p(:, t) = [X(:, t-1); \ldots ; X(:, t-p)]$
[f,g] = MAloss($Z$, $A$):
    $[U, \Sigma, V] = \text{svd}(Z)$
    $Y = AX_p + \sum_{j=1}^{q} Z(j : n(j+1) - 1, :)$
    f = $L(Y; X) + \alpha$ sum(diag($\Sigma$))
    g = repmat($\nabla_Y L(Y; X)$, $q$, 1)
    // Zero out unused parts of $Z$
    for $j = 1, \ldots, q$, g($j : n(j+1), (t-j+1) : t$) = 0
    g = g $+ \alpha UV'$
[f,g] = ARloss($A$, $Z$):
    $Y = AX_p + \sum_{j=1}^{q} Z(j : n(j+1) - 1, :)$
    $f = L(Y; X) + \alpha R(A)$
    $g = (\nabla_Y L(Y; X))X_p + \gamma \nabla R(A)$
Initialize $Z = \mathbf{0}, A = \mathbf{0}$
// Apply your favourite optimizer to the AR search
$A$ = lbfgs(ARloss($\cdot$, $Z$), $A$)
// Iterate between $A$ and $Z$
$[A, Z]$ = iterate(ARloss, MAloss, $A$, $Z$)
// Recover the optimal $B$ and $\mathcal{E}$
$[U, \Sigma, V] = \text{svd}(Z)$
Return: $A, B = U\Sigma^{1/2}, \mathcal{E} = \Sigma^{1/2}V'$
```

Experimental evaluation

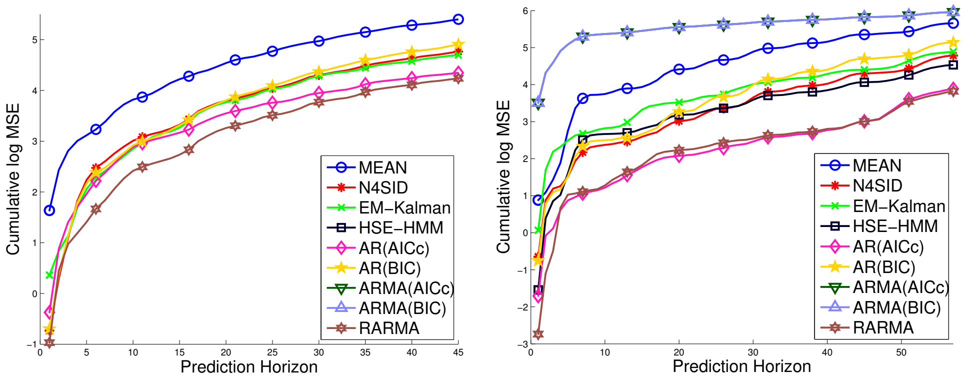
In this section, regularized ARMA is compared to a wide range of time series methods for the task of forecasting future observations on both synthetic and real-world data. As discussed in Appendix F, forecasts are performed using only the autoregressive parameters. In the final section, there is also a comparison on estimating the underlying autoregressive parameters of RARMA using $q = 0$ versus $q > 0$.

Several algorithms are compared: MEAN, which uses the mean of the training sequence as the prediction, a popular subspace identification method (N4SID) (Van Overschee and De Moor 1994), expectation-maximization to learn the parameters for a Kalman filter (EM-Kalman), Hilbert space embeddings of hidden Markov models (HSE-HMM) (Song et al. 2010; Boots and Gordon 2012),[4] maximum likelihood estimation of vector AR (AR), the Hannan-Rissanen method for ARMA (ARMA) (Hannan and Kavalieris 1984) and global estimation of regularized ARMA (RARMA). We also compared to local alternation of the RARMA objective

[4] In addition to the two method-of-moments approaches, N4SID and HSE-HMM, we tried a third state-space technique (Anandkumar, Hsu, and Kakade 2012), with no previous published empirical demonstrations. It performed poorly and so is omitted.

Table 1: For each dataset, the first column contains the test MSE (with standard error in parentheses) and the second the percentage of trials that were stable. The stability rates are measured using a threshold: eigenvalues $< 1+\epsilon = 1.01$. The method(s) with the most t-test wins with significance level of 5% are bold for each dataset. Stable rates are key for iterated prediction performance; large MSE is mainly due to unstable trials.

Algorithm	N6-P2-Q2		N9-P3-Q3		N12-P3-Q3		N15-P3-Q3		N12-P4-Q4		CAC		Atlantic	
MEAN	2.85(0.13)	1.00	6.64(0.27)	1.00	12.5(0.44)	1.00	21.2(1.06)	1.00	6.81(0.19)	1.00	4.92(0.17)	1.00	5.03(0.39)	1.00
N4SID	3.23(0.21)	1.00	6.82(0.30)	1.00	12.9(0.58)	1.00	24.7(1.46)	1.00	6.85(0.19)	1.00	2.60(0.35)	1.00	2.10(0.98)	1.00
EM-Kalman	4.27(0.30)	0.97	11.7(1.17)	0.94	19.0(1.19)	0.95	32.8(3.03)	0.89	15.9(1.45)	0.88	2.43(0.26)	1.00	2.33(0.66)	1.00
HSE-HMM	13.5(12.8)	0.95	1070(1469)	0.95	353(514)	0.95	2017(3290)	0.95	31.8(28.6)	0.91	7637(1433)	0.88	1.63(0.53)	1.00
AR(AICc)	1.83(0.29)	0.96	8.99(2.38)	0.88	16.9(2.69)	0.84	80.2(24.7)	0.69	24.8(29.6)	0.69	**1.71**(0.31)	1.00	**0.85**(0.37)	1.00
AR(BIC)	1.67(0.25)	0.97	6.42(1.27)	0.91	10.7(1.22)	0.93	34.0(5.20)	0.85	5.25(0.54)	0.82	3.00(0.27)	1.00	2.99(0.81)	1.00
ARMA(AICc)	1.63(0.2)	0.98	4.93(0.62)	0.93	8.52(0.73)	0.96	27.7(3.64)	0.86	5.49(0.48)	0.88	101(42.9)	1.00	6.80(2.85)	1.00
ARMA(BIC)	1.68(0.25)	0.98	4.81(0.58)	0.95	8.40(0.59)	0.97	29.4(5.76)	0.91	5.26(0.48)	0.97	107(49.3)	1.00	6.80(2.85)	1.00
RARMA	**1.29**(0.08)	1.00	**4.10**(0.30)	0.97	**7.49**(0.41)	0.99	**15.7**(0.92)	0.98	**4.69**(0.21)	0.99	**1.53**(0.27)	1.00	**0.80**(0.35)	1.00



(a) CAC.

(b) Atlantic.

Figure 2: Cumulative test MSE in log scale on two real-world datasets. Each model is iterated for (a) 40 and (b) 60 steps, respectively. AR(BIC) and AR(AICc) have significantly different performance, indicating the importance in selecting a good lag length for AR. HSE-HMM is unstable for CAC, but performs reasonably well for Atlantic. The best performing methods are AR(AICc) and RARMA. RARMA has strong first step prediction in both cases and has very good early predictions in CAC.

(11); for both the best and worst random initializations, however, the results were always worse that global RARMA, slower by more than an order of magnitude and often produced unstable results. Therefore, these local alternator results are omitted, with the focus instead on the comparison between the RARMA objective and the other approaches. The HR method is used for the ARMA implementation, because a recent study (Kascha 2012) shows that HR is reasonably stable compared to other vector ARMA learning algorithms. A third step was added for further stability, in which the $A^{(i)}$ are re-learned from the observations with the MA component (from the second step) removed. The built-in Matlab implementations were used for AR and N4SID.

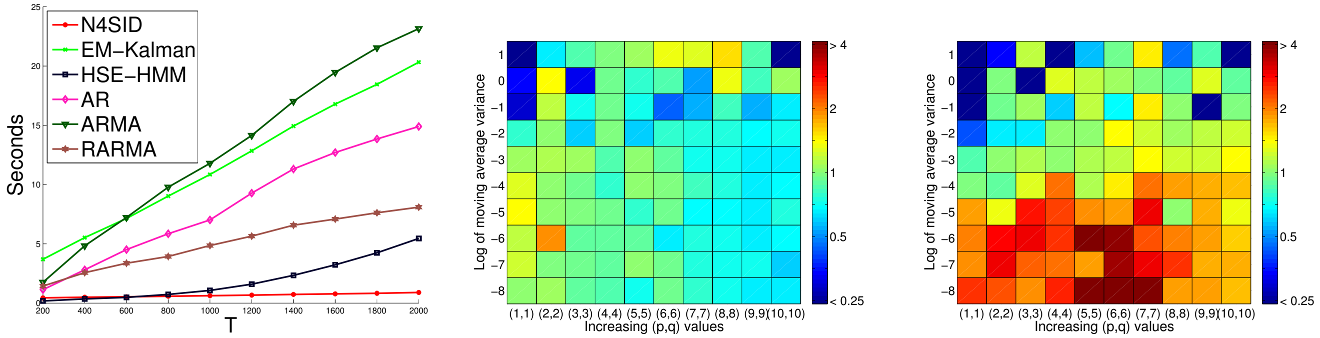
The lag parameters p and q were selected according to standard criteria in time series. For AR and ARMA, the parameters p and q are chosen using BIC and AICc, and reported separately due to interesting differences in their performance. For N4SID, the built-in Matlab implementation chose the best order. For RARMA, because of the temporal structure in the data, parameters were chosen by performing estimation on the first 90% of the training sequence and evaluating model performance on the last 10% of the training sequence. We use a robust loss, the Huber loss, for RARMA, which is easily incorporated due to the generality of RARMA. Autoregressive models can also easily use the Huber loss; we therefore directly compare RARMA to only using an autoregressive component in the last section.

Synthetic experiments: Synthetic data sets are generated from an ARMA(p, q) model. An ARMA model is called unstable if the spectra of the AR matrices exceeds the unit circle on the complex plane (Lütkepohl 2007); intuitively, iterating a dynamics matrix $A = U\Sigma V'$ that has any $\Sigma(i, i) > 1$ for t steps, $A^t = U\Sigma^t V'$, is expansive. See Appendix E for details about the procedure for generating stable ARMA models. For each (p, q, n) configuration, 500 data sequences are generated, each with 300 samples partitioned into 200 training points followed by 100 test points.

Table 1 shows the results, including the test mean squared error (MSE) and the stability rates over all trials. RARMA with global optimization is the best model on each data set in terms of MSE. Learning is generally more difficult as the dimension increases, but RARMA performs well even when most algorithms fail to beat the baseline (MEAN) and maintains a reasonably stable rate.

Figure 3(a) illustrates a runtime comparison, in CPU seconds. The synthetic model is fixed to $n = 9$ and $p = q = 3$, with an increasing number of training points. RARMA is comparable to other methods in terms of efficiency and scales well with increasing number of samples T.

Experiments on real time series: To see how our method performs on real-world data, we experimented on two real-



(a) Runtimes for increasing T

(b) Parameter recovery for $q = 0$ vs. $q > 0$

(c) Forecasting for $q = 0$ vs. $q > 0$

Figure 3: **(a)** Runtimes for an increasing number of samples, for multivariate series with $n = 9$ and $p = q = 3$. For **(b)** and **(c)**, the relative error is reported between RARMA(p, q) and the best RARMA(m,0) model, with $m = p, p{+}q$ (with results for each m separately in Appendix E, Figure 4). The plot is symmetric, where at 4, RARMA(p, q) has 4x lower error (good), and at 0.25, has 4x higher error (bad). The x-axis shows increasing lag and the y-axis increasing moving average variance. As the variance is increased beyond $\exp(-3)$, using the MEAN as a predictor begins to outperform all three methods and the moving average component begins to dominate the autoregressive component. For **(b)**, the comparison is the ℓ_1 error between the recovered A parameters and the true parameters, cut-off at p for RARMA($p + q, 0$). As p, q increase, RARMA($p + q$,0) becomes the dominant method for obtaining the underlying parameters, despite the fact that only the first p components of the learned A are used. For **(c)**, the comparison is with forecasting accuracy for a horizon of 10, measured with ℓ_1 error. Using $q > 0$ clearly improves performance for lower variance levels; the mean predictor begins to outperform all three at e^{-2}.

world datasets from IRI.[5] These climate datasets consist of monthly sea surface temperature on the tropical Pacific Ocean (CAC) and the tropical Atlantic Ocean (Atlantic). The area size of CAC is 84×30 with 399 points, while the area of Atlantic is 22×11 with 564 points. We use the first 30×30 locations for CAC and the first 9×9 locations for Atlantic. These areas are further partitioned into grids of size 3×3 and vectorized to obtain observations $\mathbf{x}_t \in \mathbb{R}^9$. The data for each location is normalized to have sample mean of zero and sample standard deviation of one in the experiments. The first 90% of the sequence is used as training set, the last 10% as the test set.

Table 1 shows the test MSE and Figure 2 shows the cumulative MSE in log scale. As in the synthetic experiments, RARMA is consistently among the best models in terms of MSE. Moreover, when examining iterated forecasting accuracy in Figure 2, RARMA is better for early predictions (about the first 30 predictions) on the real datasets.

Investigating the moving average component: The final comparison is an investigation into the importance of the moving average component, versus simply using an AR model. RARMA(p, q) is compared to RARMA($p, 0$) and RARMA($p + q, 0$) for two settings: recovering the true autoregressive parameters, A, and accuracy in forecasting. The same code is run for all three methods, simply with different p, q settings. The comparison is over increasing lag and increasing variance of the moving average component. The results are presented in Figure 3, with a more complete figure in Figure 4, Appendix E. The heat map presents the relative error between RARMA(p, q) and the best of RARMA($p, 0$) and RARMA($p{+}q, 0$); values greater than 1 indicate superior performance for RARMA(p, q).

These results indicate two interesting phenomena. First, including the moving average component significantly improves forecasting performance when the variance is relatively small. As the variance reached levels where the MEAN began to outperform all three techniques, the models with $q = 0$ were slightly better. Second, RARMA with $q > 0$ performs noticeably better for forecasting but typically performed about the same or worse for extracting the underlying autoregressive parameters. This result suggests that, if the ultimate goal is forecasting, we need not focus so much on identification. Importantly, because vector ARMA models can now be solved globally and efficiently, there is little disadvantage in using this more powerful model, and strong benefits in some cases.

Conclusion

This paper tackles a long-standing problem in time series modeling: tractable maximum likelihood estimation of multivariate ARMA models. The approach involves three key components: (1) estimating stochastic ARMA models, which relaxes the requirement that the innovations exactly equal the residuals, (2) characterizing the independent Gaussian structure of the innovations using a regularizer and (3) developing a theoretically sound convex reformulation of the resulting stochastic multivariate ARMA objective. Solving this convex optimization is efficient, guarantees global solutions and outperformed previous ARMA and state-space methods in forecasting on synthetic and real datasets.

These results suggest stochastic regularized ARMA is a promising direction for time series modeling, over conventional (deterministic) ARMA. Stochastic ARMA is similarly motivated by the Wold representation, but is amenable to optimization, unlike deterministic ARMA. Moreover, the regularized ARMA objective facilitates development of estimation algorithms under generalized innovations. Though the focus in this work was on a convex formulation for Gaussian innovations, it extends to Laplacian innovations for a $(2, 1)$-block norm regularizer. Advances in optimizing structured norm objectives advance global estimation of regularized ARMA models for novel innovation properties.

[5] http://iridl.ldeo.columbia.edu/SOURCES/

Acknowledgements

This work was supported by NSERC, Alberta Innovates Technology Futures and the University of Alberta.

References

Anandkumar, A.; Hsu, D.; and Kakade, S. M. 2012. A method of moments for mixture models and hidden Markov models. *arXiv:12030683v3 [cs.LG]*.

Anava, O.; Hazan, E.; Mannor, S.; and Shamir, O. 2013. Online learning for time series prediction. In *Proceedings of the 26th Annual Conference on Learning Theory*, 1–13.

Andersson, S. 2009. Subspace estimation and prediction methods for hidden Markov models. *The Annals of Statistics* 4131–4152.

Arslan, O. 2010. An alternative multivariate skew Laplace distribution: properties and estimation. *Statistical Papers* 865–887.

Bach, F.; Mairal, J.; and Ponce, J. 2008. Convex sparse matrix factorizations. *arXiv:0812.1869v1 [cs.LG]*.

Banerjee, A.; Merugu, S.; Dhillon, I. S.; and Ghosh, J. 2005. Clustering with Bregman divergences. *Journal of Machine Learning Research* 6:1705–1749.

Bauer, D. 2005. Estimating linear dynamical systems using subspace methods. *Econometric Theory* 21(01).

Benjamin, M. A.; Rigby, R. A.; and Stasinopoulos, D. M. 2003. Generalized autoregressive moving average models. *Journal of the American Statistical Association* 98(461):214–223.

Benveniste, A.; Metivier, M.; and Priouret, P. 2012. *Adaptive Algorithms and Stochastic Approximations*. Springer.

Boots, B., and Gordon, G. 2012. Two-manifold problems with applications to nonlinear system identification. In *Proceedings of the 29th International Conference on Machine Learning*, 623–630.

Brockwell, P. J., and Davis, R. A. 2002. *Introduction to Time Series and Forecasting*. Springer.

Brown, P. F.; Pietra, V. J. D.; Pietra, S. A. D.; and Mercer, R. L. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics* 19(2):263–311.

Candes, E. J.; Li, X.; Ma, Y.; and Wright, J. 2011. Robust principal component analysis? *Journal of the ACM* 58:1–37.

Cramér, H. 1946. *Mathematical Methods of Statistics*. Princeton University Press.

Dudik, M.; Harchaoui, Z.; and Malick, J. 2012. Lifted coordinate descent for learning with trace-norm regularization. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, 327–336.

Durbin, J. 1960. The fitting of time-series models. *Revue de l'Institut International de Statistique* 28:233–244.

Foster, D. P.; Rodu, J.; and Ungar, L. H. 2012. Spectral dimensionality reduction for HMMs. *arXiv:1203.6130v1*.

Hannan, E. J., and Kavalieris, L. 1984. Multivariate linear time series models. *Advances in Applied Probability* 16:492–561.

Hsu, D.; Kakade, S.; and Zhang, T. 2012. A spectral algorithm for learning Hidden Markov Models. *Journal of Computer and System Sciences* 1460–1480.

Kascha, C. 2012. A comparison of estimation methods for vector Autoregressive Moving-Average Models. *Econometric Reviews* 31(3):297–324.

Katayama, T. 2006. *Subspace Methods for System Identification*. Springer.

Lütkepohl, H. 2007. *New Introduction to Multiple Time Series Analysis*. Springer.

Mauricio, J. A. 1995. Exact maximum likelihood estimation of stationary vector ARMA models. *Journal of the American Statistical Association* 282–291.

Moonen, M., and Ramos, J. 1993. A subspace algorithm for balanced state space system identification. *IEEE Transactions on Automatic Control* 38(11):1727–1729.

Neal, R. M., and Hinton, G. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models* 355–368.

Scargle, J. D. 1981. Studies in astronomical time series analysis. *Astrophysical Journal Supplement Series* 835–853.

Shah, P.; Bhaskar, B. N.; Tang, G.; and Recht, B. 2012. Linear system identification via atomic norm regularization. *arXiv:1204.0590v1 [math.OC]*.

Song, L.; Boots, B.; Siddiqi, S.; Gordon, G.; and Smola, A. 2010. Hilbert space embeddings of hidden Markov models. In *Proceedings of the 27th International Conference on Machine Learning*, 991–998.

Thiesson, B.; Chickering, D. M.; Heckerman, D.; and Meek, C. 2012. ARMA time-series modeling with graphical models. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 552–560.

Van Overschee, P., and De Moor, B. 1994. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica* 30(1):75–93.

Viberg, M. 1995. Subspace-based methods for the identification of linear time-invariant systems. *Automatica* 31(12):1835–1851.

White, M.; Yu, Y.; Zhang, X.; and Schuurmans, D. 2012. Convex multi-view subspace learning. In *Advances in Neural Information Processing Systems*, 1673–1681.

Wiesel, A.; Bibi, O.; and Globerson, A. 2013. Time varying autoregressive moving average models for covariance estimation. *IEEE Transactions on Signal Processing* 61(11):2791–2801.

Wold, H. O. A. 1938. *A Study in The Analysis of Stationary Time Series*. Almqvist & Wiskell.

Zhang, X.; Yu, Y.; White, M.; Huang, R.; and Schuurmans, D. 2011. Convex sparse coding, subspace learning, and semi-supervised extensions. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, 567–573.

Zhao, H., and Poupart, P. 2014. A sober look at spectral learning. *arXiv:1406.4631v1 [cs.LG]*.

Optimal Estimation of Multivariate ARMA Models Appendix

A Proof of Lemma 1

Proof: A standard argument follows (Neal and Hinton 1998). First note that a lower bound on $\log p(\mathbf{x}_{1:T}|\Theta)$ can be easily obtained:

$$\begin{aligned}
&\log p(\mathbf{x}_{1:T}|\Theta) = \log \int p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta)\, d\boldsymbol{\varepsilon}_{1:T} \\
&= \log \int q(\boldsymbol{\varepsilon}_{1:T}) \frac{p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta)}{q(\boldsymbol{\varepsilon}_{1:T})}\, d\boldsymbol{\varepsilon}_{1:T} \\
&\qquad \triangleright \text{ for any density } q(\cdot) \\
&\geq \int q(\boldsymbol{\varepsilon}_{1:T}) \left(\log \frac{p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta)}{q(\boldsymbol{\varepsilon}_{1:T})} \right) d\boldsymbol{\varepsilon}_{1:T} \\
&\qquad \triangleright \text{ by Jensen's inequality (since } \log \text{ is concave)} \\
&= \int q(\boldsymbol{\varepsilon}_{1:T}) \log p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta)\, d\boldsymbol{\varepsilon}_{1:T} \\
&\qquad - \int q(\boldsymbol{\varepsilon}_{1:T}) \log q(\boldsymbol{\varepsilon}_{1:T})\, d\boldsymbol{\varepsilon}_{1:T} \\
&= \int q(\boldsymbol{\varepsilon}_{1:T}) \log p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta)\, d\boldsymbol{\varepsilon}_{1:T} + H(q(\cdot)). \quad (14)
\end{aligned}$$

It remains to show that the maximization of the lower bound attains the original value; that is:

$$\log p(\mathbf{x}_{1:T}|\Theta) = \max_{q(\cdot)} \int q(\boldsymbol{\varepsilon}_{1:T}) \log p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta)\, d\boldsymbol{\varepsilon}_{1:T} + H(q(\cdot))$$

over densities $q(\cdot)$. This can be verified merely by choosing the particular density $q(\boldsymbol{\varepsilon}_{1:T}) = p(\boldsymbol{\varepsilon}_{1:T}|\mathbf{x}_{1:T}, \Theta)$ and

$$\begin{aligned}
(14) &= \int p(\boldsymbol{\varepsilon}_{1:T}|\mathbf{x}_{1:T}, \Theta) \log p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta)\, d\boldsymbol{\varepsilon}_{1:T} \\
&\quad - \int p(\boldsymbol{\varepsilon}_{1:T}|\mathbf{x}_{1:T}, \Theta) \log p(\boldsymbol{\varepsilon}_{1:T}|\mathbf{x}_{1:T}, \Theta)\, d\boldsymbol{\varepsilon}_{1:T} \quad (15) \\
&= \int p(\boldsymbol{\varepsilon}_{1:T}|\mathbf{x}_{1:T}, \Theta) \log \frac{p(\mathbf{x}_{1:T}, \boldsymbol{\varepsilon}_{1:T}|\Theta)}{p(\boldsymbol{\varepsilon}_{1:T}|\mathbf{x}_{1:T}, \Theta)}\, d\boldsymbol{\varepsilon}_{1:T} \\
&= \int p(\boldsymbol{\varepsilon}_{1:T}|\mathbf{x}_{1:T}, \Theta) \log p(\mathbf{x}_{1:T}|\Theta)\, d\boldsymbol{\varepsilon}_{1:T} \\
&= \log p(\mathbf{x}_{1:T}|\Theta) \int p(\boldsymbol{\varepsilon}_{1:T}|\mathbf{x}_{1:T}, \Theta)\, d\boldsymbol{\varepsilon}_{1:T} \\
&= \log p(\mathbf{x}_{1:T}|\Theta), \quad (16)
\end{aligned}$$

implying that the upper bound can always be attained by $q(\boldsymbol{\varepsilon}_{1:T}) = p(\boldsymbol{\varepsilon}_{1:T}|\mathbf{x}_{1:T}, \Theta)$. ■

B Proof of Theorems 2 and 3

Lemma 4. *Given a convex loss function $L(\cdot, \mathbf{x})$ and convex regularizer $R(\cdot)$ with $\gamma \geq 0$, the following loss is convex in Z for all $\mathbf{x}_t \in \mathbb{R}^n$*

$$L_A\left(\sum_{j=0}^{q} Z^{(j)}{}_{:,t-j}, \mathbf{x}_t\right) = L\left(\sum_{j=0}^{q} Z^{(j)}{}_{:,t-j}, + \sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i}, \mathbf{x}_t\right) + \gamma R(A)$$

Proof: Let $g_t((Z,A)) = \sum_{j=0}^{q} Z^{(j)}_{:,t-j} + \sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i}$. We need to show that $L(g_t((Z,A)), \mathbf{x}_t)$ is convex for any $\mathbf{x}_t$.

1. Clearly $g_t((Z,A))$ is linear in (Z,A).
2. Since $L(\cdot, \mathbf{x}_t)$ is convex, and the composition of a convex function with a linear function is convex, then $L(g_t(\cdot), \mathbf{x}_t)$ is convex.

■

Corollary 5. *The parameter estimation problem with a convex regularizer $R(\cdot)$ on the autoregressive parameters, A,*

$$\min_A \min_Z \sum_{t=1}^{T} L\left(Z^{(j)}_{:,t-j} + \sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i}, \mathbf{x}_t\right) + \alpha |\!|\!| Z |\!|\!| + \gamma R(A)$$

is jointly convex in A and Z for $\alpha, \gamma \geq 0$.

Theorem 2 The regularized ARMA estimation problem for $G(B) = ||B||_F^2$ is equivalent to the following convex optimization problem

$$\begin{aligned}
&\min_{A,B,\mathcal{E}} \sum_{t=1}^{T} L_{A_t}\left(\sum_{j=0}^{q} B^{(j)}\mathcal{E}_{:,t-j}\right) + \alpha||\mathcal{E}||_F^2 + \alpha||B||_F^2 \\
&= \min_{A,Z} \sum_{t=1}^{T} L_{A,t}\left(\sum_{j=0}^{q} Z^{(j)}_{:,t-j}\right) + 2\alpha||Z||_{tr}
\end{aligned}$$

with a singular value decomposition recovery: $Z = U\Sigma V'$ giving $B = U\sqrt{\Sigma}$ and $\mathcal{E} = \sqrt{\Sigma}V'$.

Proof: This result is well known (Bach, Mairal, and Ponce 2008; Zhang et al. 2011), but we include some details here for completeness. First, $||Z||_{tr} = \min_{B,\mathcal{E}:B\mathcal{E}=Z} \frac{1}{2}(||B||_F^2 + ||\mathcal{E}||_F^2)$ because the induced norm is equal to the dual of the spectral norm, which is the trace norm. For the recovery,

$$\begin{aligned}
&||B||_F^2 + ||\mathcal{E}||_F^2 = ||U\sqrt{\Sigma}||_F^2 + ||\sqrt{\Sigma}V'||_F^2 \\
&= ||\sqrt{\Sigma}||_F^2 + ||\sqrt{\Sigma}||_F^2 = 2||\sqrt{\Sigma}||_F^2 = 2\operatorname{tr}\Sigma = 2||Z||_{tr}
\end{aligned}$$

■

For Theorem 3, we will first proof the following lemma.

Lemma 6.

$$\begin{aligned}
&\min_{B,\mathcal{E}:Z=B\mathcal{E}} \frac{1}{2}\left(||\mathcal{E}||_F^2 + \max_{j=0,\ldots,q} ||B^{(j)}||_2^2\right) \\
&= \min_{B,\mathcal{E}:Z=B\mathcal{E}\forall i,j||B^{(j)}{}_{:,i}||_2\leq 1} ||\mathcal{E}||_{2,1}
\end{aligned}$$

Proof: From Equations 2,3 and 4 in (Bach, Mairal, and Ponce 2008), we know that the induced norm on $Z = B\mathcal{E}$ can be written in either of these two forms. This results follows from the fact that for $||B_{:,i}||_C^2 = \max_{j=0,\dots,q} ||B^{(j)}{}_{:,i}||_2^2$ and $||\mathcal{E}_{i,:}||_R^2 = ||\mathcal{E}_{i,:}||_2^2$, any $(B, \mathcal{E})$, can be rescaled by $s_i = \frac{\|\mathcal{E}_{:,i}\|_R}{\|B_{:,i}\|_C}$ to get:

$$
\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n} ||B_{:,i}s_i||_C^2 + ||s_i^{-1}\mathcal{E}_{i,:}||_R^2 \\
&= \frac{1}{2}\sum_{i=1}^{n} ||B_{:,i}||_C^2 \frac{\|\mathcal{E}_{:,i}\|_R}{\|B_{:,i}\|_C} + ||\mathcal{E}_{i,:}||_R^2 \frac{\|B_{:,i}\|_C}{\|\mathcal{E}_{:,i}\|_R} \\
&= \frac{1}{2}\sum_{i=1}^{n} ||B_{:,i}||_C \|\mathcal{E}_{:,i}\|_R + ||\mathcal{E}_{i,:}||_R \|B_{:,i}\|_C \\
&= \sum_{i=1}^{n} ||B_{:,i}||_C ||\mathcal{E}||_R
\end{aligned}
$$

Then, the norm of B can be scaled to 1, with the scale being pushed into the other parameter, Z, giving the final form. ■

Theorem 3 The regularized ARMA estimation problem for $G(B) = \max_{j=0,\dots,q} ||B^{(j)}||_F^2$ is equivalent to the following convex optimization problem

$$
\begin{aligned}
&\min_{A,B,\mathcal{E}} \sum_{t=1}^{T} L_{A_t}\left(\sum_{j=0}^{q} B^{(j)}\mathcal{E}_{:,t-j}\right) \\
&\qquad\qquad + \alpha||\mathcal{E}||_F^2 + \alpha \max_{j=0,\dots,q} ||B^{(j)}||_F^2 \\
&= \min_{A,Z} \sum_{t=1}^{T} L_{A,t}\left(\sum_{j=0}^{q} Z^{(j)}_{:,t-j}\right) + \max_{0\le\rho\le1} ||W_\rho^{-1}Z||_{tr}
\end{aligned}
$$

where

$$
W_\rho := \begin{bmatrix} 1/\sqrt{\rho}\, I_n & 0 \\ 0 & 1/\sqrt{1-\rho}\, I_n \end{bmatrix}.
$$

Moreover, $||W_\rho^{-1}Z||_{tr}$ is concave in ρ over $[0, 1]$.

Proof: From Lemma 6, we find an equivalent definition for the induced norm on Z

$$
\begin{aligned}
|\!|\!|Z|\!|\!| &= \min_{B,\mathcal{E}:Z=B\mathcal{E}} \frac{1}{2}\sum_{i=1}^{n}\left(||\mathcal{E}_{i,:}||_2^2 + \max_{j=0,\dots,q} ||B_{:,i}||_2^2\right) \\
&= \min_{\substack{B,\mathcal{E}:Z=B\mathcal{E} \\ \forall i,j \|B^{(j)}{}_{:,i}\|_2 \le 1}} ||\mathcal{E}||_{2,1}
\end{aligned}
$$

Usefully, this equivalent form with $(2, 1)$-block norm regularizer and constraint on B has recently been convexly reformulated for $q = 1$ (White et al. 2012). Therefore, since our main loss is convex by Lemma 4, we can directly apply their proof; see Proposition 2, Lemma 3, Lemma 4 and Theorem 5 in (White et al. 2012) for the result. In our case, the weights on each of the views is simply $\beta_1 = \beta_2 = 1$. Therefore, we obtain

$$
|\!|\!|Z|\!|\!| = \max_{0\le\rho\le1} ||W_\rho^{-1}Z||_{tr}
$$

From (White et al. 2012, Lemma 4), we know that $||W_\rho^{-1}Z||_{tr}$ is concave in ρ over $[0, 1]$, for any Z.

■

C Boosting recovery procedure for regularized ARMA

For the goal of recovering Gaussian distributed innovation variables, we provide the following boosting procedure that iteratively generates columns of B and rows of $\mathcal{E}$ until $B\mathcal{E} = Z$.

1. First rescale Z such that $|\!|\!|Z|\!|\!| = 1$, since after recovering B and $\mathcal{E}$ we can simply multiply them both by $\sqrt{|\!|\!|Z|\!|\!|}$.

2. For any matrices B and $\mathcal{E}$, we can write $B = [\mathbf{b}_1, \dots, \mathbf{b}_n]\operatorname{diag}(\mathbf{s}_1)$ and $\mathcal{E} = \operatorname{diag}(\mathbf{s}_2)[\boldsymbol{\varepsilon}_1; \dots; \boldsymbol{\varepsilon}_n]$ for $\|\mathbf{b}_i\|_2 = 1$ and $\|\boldsymbol{\varepsilon}_i\|_2 = 1$ and scale vectors $\mathbf{s}_1, \mathbf{s}_2 \ge 0$.

3. Generate unit vectors and scales to optimize $f(K) = \|Z - K\|_F^2$, i.e.,

$$
\min_{B,\mathcal{E}} \|Z - B\mathcal{E}\|_F^2 = \min_{\mathbf{b}_1,\mathbf{b}_2,\dots,\mathbf{s}_1,\mathbf{s}_2,\boldsymbol{\varepsilon}_1,\boldsymbol{\varepsilon}_2,\dots,\mathbf{s}_2} \|Z - B\mathcal{E}\|_F^2
$$

in a repeated two step boosting approach, starting with $K_0 = 0$:

3.1 Weak learning step: greedily pick $(\mathbf{b}_t, \boldsymbol{\varepsilon}_t) \in \operatorname{argmin}_{\|\mathbf{b}\|_2=1, \|\boldsymbol{\varepsilon}\|_2=1} \langle \nabla f(K_{t-1}), \mathbf{b}_t\boldsymbol{\varepsilon}_t'\rangle$. This step can be computed efficiently by using the procedure in (White et al. 2012).

3.2 "Totally corrective" step: $\boldsymbol{\mu}^{(t)} = \operatorname*{argmin}_{\boldsymbol{\mu}\ge\mathbf{0},\sum_i \mu_i=1} f\Big(\sum_{i=1}^{t} \mu_i \mathbf{b}_i\boldsymbol{\varepsilon}_i'\Big)$, then $K_t = \sum_{i=1}^{t} \mu_i^{(t)}\mathbf{b}_i\boldsymbol{\varepsilon}_i'$.

Notice that the scales are reoptimized after each basis is added, meaning that unuseful bases will have their scale set to zero. This procedure will find a K_t satisfying $\|Z - K_t\|_F^2 < \epsilon$ within $O(1/\epsilon)$ iterations (White et al. 2012).

4. Set $\mathbf{s}_1 = \mathbf{s}_2 = \sqrt{|\!|\!|Z|\!|\!|}\sqrt{\boldsymbol{\mu}}$, $B = [\mathbf{b}_1, \dots, \mathbf{b}_t]\operatorname{diag}(\mathbf{s}_1)$. and $\mathcal{E} = \operatorname{diag}(\mathbf{s}_2)[\boldsymbol{\varepsilon}_1; \dots; \boldsymbol{\varepsilon}_t]$. Then we can see that

$$
\begin{aligned}
&\frac{1}{2}(\max_{j=0,\dots q} ||B^{(j)}||_2^2 + ||\mathcal{E}||_F^2) \\
&= \frac{1}{2}(|\!|\!|Z|\!|\!| ||[\sqrt{\mu_1}\mathbf{b}_1, \dots, \sqrt{\mu_t}\mathbf{b}_t]||_F^2 \\
&\qquad\qquad + |\!|\!|Z|\!|\!| ||[\sqrt{\mu_1}\boldsymbol{\varepsilon}_1; \dots; \sqrt{\mu_t}\boldsymbol{\varepsilon}_t]||_F^2) \\
&= \frac{|\!|\!|Z|\!|\!|}{2}(||\sqrt{\operatorname{diag}\boldsymbol{\mu}}||_F^2 + ||\sqrt{\operatorname{diag}\boldsymbol{\mu}}||_F^2) \\
&= |\!|\!|Z|\!|\!| \|\sqrt{\boldsymbol{\mu}}\|_2^2 = |\!|\!|Z|\!|\!|
\end{aligned}
$$

since $\|\sqrt{\boldsymbol{\mu}}\|_2^2 = 1$ by the constraints in step 3.2.

To recover **Laplacian innovations** instead of Gaussian innovations, the only difference is the rescaling. For the first setting, $G(B) = ||B||_F^2$, the recovery for $Z = U\Sigma V'$ is $B = U$ and $\mathcal{E} = \Sigma V'$, to obtain Laplacian distributed $\mathcal{E}$. For the second setting, we simply set $\mathbf{s}_1 = \mathbf{1}$ and $\mathbf{s}_2 = |\!|\!|Z|\!|\!|\boldsymbol{\mu}$. See the last section in Appendix D to see how the $||\mathcal{E}||_{2,1}$ regularizer corresponds to a Laplacian distribution on innovations across time.

D Generalizations for regularized ARMA

There are many generalizations to ARMA models that are important for practical applications. Of particular importance is generalizing the Gaussian distributional assumptions on $\mathbf{x}_t$ in the estimation of ARMA models (Benjamin, Rigby, and Stasinopoulos 2003), the generalization to ARMA with exogenous, input variables (ARMAX) and the generalization to non-stationary series (ARIMA). In this section, we indicate how the regularized ARMA formulation can be generalized to include these three important settings.

Generalized distributional assumptions

We can relax the Gaussian assumption on observations $\mathbf{x}_t$ by moving to natural exponential family distributions.[6] A natural exponential family distribution is a distribution parametrized by θ as follows

$$P_F(\mathbf{x}|\boldsymbol{\theta}) = \exp(\mathbf{x}^T\boldsymbol{\theta} - F(\boldsymbol{\theta}))p_0(\mathbf{x}) \tag{17}$$

where F is commonly thought of as the cumulant function. Examples of natural exponential families include the Gaussian, gamma, chi-square, beta, Weibull, Bernoulli and Poisson distributions (Banerjee et al. 2005). Many distributions can be approximated with exponential families, further generalizing the distributional assumptions.

With this generalized (noise) model, we can write the log likelihood as

$$\begin{aligned} \log P_F(\mathbf{x}_t|\boldsymbol{\theta}) &= \mathbf{x}_t^T\boldsymbol{\theta}_t - F(\boldsymbol{\theta}_t) + \log p_0(\mathbf{x}_t) \qquad (18) \\ &= -D_F(\boldsymbol{\theta}_t||\mathbf{x}_t) + F(\mathbf{x}_t) + \log p_0(\mathbf{x}_t) \end{aligned}$$

where D_F is a Bregman divergence for strictly convex potential function $F : \mathbb{R}^n \to \mathbb{R}$

$$D_F(\boldsymbol{\theta}_t||\mathbf{x}_t) = F(\boldsymbol{\theta}_t) - F(\mathbf{x}_t) - \nabla F(\mathbf{x}_t)'(\boldsymbol{\theta}_t - \mathbf{x}_t).$$

Bregman divergences are not true metrics: they do not generally satisfy the triangle inequality nor symmetry. They are, however, convex in the first argument and encompass many useful losses, including the Euclidean loss (Gaussian distribution with $F(\mathbf{x}) = \frac{1}{2}\mathbf{x}'\mathbf{x}$), relative entropy loss (Poisson distribution with $F(\mathbf{x}) = \ln(\mathbf{1}'\exp(\mathbf{x}))$) and Itakura-Saito distance (exponential distribution for scalar variables x with $F(\lambda) = \log(\lambda)$) (Banerjee et al. 2005).

The resulting minimization of the negative log likelihood with respect to $\boldsymbol{\theta}_t$ now corresponds to a minimization of the Bregman divergence[7], since the terms $F(\mathbf{x}_t) + \log p_0(\mathbf{x}_t)$ in (18) do not affect the minimization over parameter $\boldsymbol{\theta}_t$. Previously, we restricted ourselves to $F(\mathbf{x}) = \frac{1}{2}\mathbf{x}'\mathbf{x}$, which gives the Bregman divergence $D_F(\boldsymbol{\theta}_t||\mathbf{x}_t) = ||\boldsymbol{\theta}_t - \mathbf{x}_t||_2^2$. Though the Euclidean loss is symmetric, in general, Bregman divergences are only convex in the first argument. Therefore, it is crucial that we minimize $D_F(\boldsymbol{\theta}_t||\mathbf{x}_t)$ with $\boldsymbol{\theta}_t$ as the first argument to achieve a convex formulation of the parameter estimation problem.

Generalization to ARMAX

We can trivially add exogenous variables because, like the autoregressive part, they are included in the loss additively:

$$L\left(\sum_{i=1}^{p} A^{(i)}\mathbf{x}_{t-i} + \sum_{j=0}^{q} Z^{(j)}_{:,t-j} + \sum_{i=1}^{s} C^{(i)}\mathbf{u}_{t-i}\ ;\ \mathbf{x}_t\right)$$

where $\mathbf{u}_t \in \mathbb{R}^d$ is an input control vector or exogenous vector. As with the autoregressive component, we can add a convex regularizer on $C \in \mathbb{R}^{n \times ds}$ to avoid overfitting. The resulting optimization is an alternation over the three parameters, A, Z and C.

Generalization to ARIMA models

This generalization is similarly simple, because an autoregressive integrated moving average, ARIMA(p,d,q), model is simply an ARMA(p,q) model of the time series differenced d times. Differencing is a form of taking the derivative, with the assumption that the time lag is appropriately small. As a result, the more times the differencing is applied, the more likely we are to reach a stationary distribution.

Regularized ARMA models with other regularizers

In Lemma 6 and Theorem 3, we indicated that we can also solve the following objective:

$$\min_{B,\mathcal{E},\forall i,j\|B^{(j)}{}_{:,i}\|_2 \le 1} \sum_{t=1}^{T} L_A\left(\sum_{j=1}^{q} B^{(j)}\mathcal{E}_{:,t-j}, \mathbf{x}_t\right) + \alpha||\mathcal{E}||_{2,1}$$

Below, we show that this block $2,1$-norm corresponds to assuming a prior on the innovations that is a Laplacian distribution across time.

There are several multivariate extensions of Laplace distributions; we choose a multivariate Laplace, parametrized by a mean, $\boldsymbol{\mu}_i$, and scatter matrix Σ_i, with the convenient pdf (Arslan 2010):

$$p_L(\mathcal{E}_{i,:}|\boldsymbol{\mu}_i, \Sigma_i) = \frac{|\Sigma_i|^{-1/2}}{2^T\pi^{\frac{T-1}{2}}\Gamma(\frac{T+1}{2})} e^{-\sqrt{(\mathcal{E}_{i,:}-\boldsymbol{\mu}_i)\Sigma_i^{-1}(\mathcal{E}_{i,:}-\boldsymbol{\mu}_i)'}}$$

As before, where the covariance was pushed into the B parameters, we assume $\boldsymbol{\mu} = \mathbf{0}$ and $\Sigma = I$, giving

$$\begin{aligned} -\log p_L(\mathcal{E}_{i,:}|\boldsymbol{\mu}_i, \Sigma_i) &= \tfrac{1}{2}\log\left(|\Sigma_i|\right) + T\log(2) + \tfrac{T-1}{2}\log(\pi) \\ &\quad + \log\Gamma\left(\frac{T+1}{2}\right) + \sqrt{(\mathcal{E}_{i,:} - \boldsymbol{\mu}_i)\Sigma_i^{-1}(\mathcal{E}_{i,:} - \boldsymbol{\mu}_i)'} \\ \Longrightarrow \min_{\mathcal{E}} &\sum_{i=1}^{n} -\log p_L(\mathcal{E}_{i,:}|\boldsymbol{\mu}_i = \mathbf{0}, \Sigma_i = I) \\ &= \min_{\mathcal{E}} \sum_{i=1}^{n} \sqrt{\mathcal{E}_{i,:}\mathcal{E}_{i,:}'} = \min_{\mathcal{E}} \|\mathcal{E}\|_{2,1} \end{aligned}$$

We can now simplify the relationship between the hidden variables because this multivariate Laplace distribution decomposes nicely into the multiplication of a scalar gamma-distributed variable, $S_i \sim G(\frac{t+1}{2}, \beta = \frac{1}{2})$ the covariance matrix $\Sigma \in \mathbb{R}^{t \times t}$ and independent, standard normal variables, $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, I)$ (Arslan 2010):

$$\mathcal{E}_{i,:} = \sqrt{S_i}\Sigma\boldsymbol{\epsilon}_i \quad \triangleright\, p_{S_i}(s) = \frac{1}{\Gamma(\frac{t+1}{2})2^{\frac{t+1}{2}}} s^{\frac{t-1}{2}} \exp\left(-\tfrac{s}{2}\right).$$

Interestingly, this makes the connection with the Frobenius norm formulation more clear, since once the scale is fixed, we have independent innovations. The scalar across time acts like a shared scale on the covariance of the innovations.

[6] This equates to generalizing the assumptions on the noise.

[7] Banerjee et al. (2005) proved there is an isomorphic equivalence between regular Bregman divergences and natural (regular) exponential family distributions.

Algorithm 2 ARMA synthetic data generation

```
Input: $p$, $q$, dimension of series $n$, number of samples $T$
Output: $A, B, \mathbf{x}_1, \ldots, \mathbf{x}_T$
$m \leftarrow n/p$    // Size of partition in $\mathbf{x}$
$s_0 \leftarrow 0.999$    // Scale $s_0 < 1$
$\tau \leftarrow$ floor$(T/3)$    // Permutation period
for $i = 1 : p$ do
    $\mathbf{d} \leftarrow []$    // Eigenvalues of permutation matrix $\tilde{A}^{(i)}$
    $V = []$    // Eigenvectors of permutation matrix $\tilde{A}^{(i)}$
    count $\leftarrow 0$
    if $m$ is odd then
        $\mathbf{d} \leftarrow s_0$
        $V =$ randn$(m, 1)$
        count $\leftarrow 1$
    end if
    while count $< m$ do
        d_new $\leftarrow s_0 \cdot \exp(2\pi\sqrt{-1}/\tau)$
        $\mathbf{d} \leftarrow [\mathbf{d}$, d_new, conj(d_new)$]$
        $\mathbf{v}_1 \leftarrow$ randn$(m, 1), \mathbf{v}_2 \leftarrow$ randn$(m, 1)$
        $V = [V,\ \mathbf{v}_1 + \sqrt{-1}\mathbf{v}_2,\ \mathbf{v}_1 - \sqrt{-1}\mathbf{v}_2]$
        count $\leftarrow$ count + 2
    end while
    $A^{(i)} = \mathbf{0}$
    $D = $ diag$(\mathbf{d})$
    $A^{(i)}{}_{((i-1)m+1:im),((i-1)m+1:im)} = $ real$(VDV^{-1})$
end for
$B \leftarrow$ randn$(qn, n)$,    $B_{:j} \leftarrow B_{:j}/\sqrt{\sum_{i=1}^{qn} B_{ij}^2}$
$\mathcal{E} \leftarrow \nu \cdot$ randn$(n, T)$ // Draw $\mathcal{E}$ with variance $\nu$
$(\mathbf{x}_1, \ldots, \mathbf{x}_T) \leftarrow$ simulate$(A, B, \mathcal{E})$ // Simulate data
    from generated $A$, $B$ and $\mathcal{E}$
Return: $A, B, (\mathbf{x}_1, \ldots, \mathbf{x}_T)$
```

In general, there are potentially many other distributional assumptions we can make on the innovation variables that could be efficiently solvable, depending on advances in convex reformulations of matrix factorization.

E Details for the algorithms and experiments

For the autoregressive part, we need to choose the parameters $A^{(i)}$ carefully, otherwise the system will be unstable and the generated sequences will diverge (Lütkepohl 2007). For vector ARMA in particular, there are not many approaches to generating stable ARMA models and most experiments involve small, specific known systems. We use an approach where each $A^{(i)}$ acts as a permutation matrix on a sub-part of $\mathbf{x}$. The observation $\mathbf{x}$ is partitioned into p subvectors of size $m = n/p$. Each $A^{(i)}$ permutes the ith block in $\mathbf{x}$, with a slow decay for numerical stability. Therefore, $A^{(i)}$ has zeros in entries corresponding to blocks $1, \ldots, i-1$ and $i+1, \ldots, n$ and a slowly decaying permutation matrix at row and columns entries $(i-1)m+1$ to im. This permutation matrix is generated using randomly generated eigenvectors and eigenvalues, as well as their conjugates, to give $\tilde{A}^{(i)} = VDV^{-1}$. The conjugates are used to ensure that the generated matrix A is a real matrix. The decay rate s_0 is set to 0.999 for numerical stability and the period of the permutation matrix determine by τ.

For the moving average part, the entries of $B^{(j)}$ and $\boldsymbol{\varepsilon}_t$ are drawn from standard normal, $\mathcal{N}(\mathbf{0}, I)$. The matrix B is normalized to have unit variance. To generate data from this model, the initial $\mathbf{x}_{-p}, \ldots, \mathbf{x}_0$ are sampled from the unit ball in $\mathbb{R}^n$. Then the model is iterated from this initial sample. See Algorithm 2 for the complete generation procedure.

F Forecasting in ARMA models

A main use of ARMA models is for forecasting, once the model parameters have been estimated. In particular, given the parameters, Θ, we would like to predict the value of a future observation, $\mathbf{x}_{t+h}$, given observations of a history $\mathbf{x}_1, \ldots, \mathbf{x}_t$. Under the Gaussian assumption (and exponential family distributions more generally, see Appendix D) the optimal point predictor, $\hat{\mathbf{x}}_{t+h}$, is given by the conditional expectation

$$\tilde{\mathbf{x}}_{t+h} = E[\mathbf{x}_{t+h} | \mathbf{x}_1, \ldots, \mathbf{x}_t, \Theta], \tag{19}$$

which can be easily computed from the observed history and the parameters. To understand ARMA forecasting in a little more detail, first consider the one step prediction case. If the innovation variables are included in the observed history, then from the conditional independence properties depicted in Figure 1(a) and (1) the one step conditional expectation is easily determined to be

$$\begin{aligned}\hat{\mathbf{x}}_{t+1} &= E[\mathbf{x}_{t+1} | \mathbf{x}_1, \ldots, \mathbf{x}_t, \boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_t, \Theta] \\ &= \sum_{i=1}^{p} A^{(i)} \mathbf{x}_{t+1-i} + \sum_{j=1}^{q} B^{(j)} \boldsymbol{\varepsilon}_{t+1-j}.\end{aligned}$$

For the h step forecast, since the expected innovations for $\mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+h-1}$ are zero given previous innovations, we obtain

$$\begin{aligned}\hat{\mathbf{x}}_{t+h} &= E[\mathbf{x}_{t+h} | \mathbf{x}_1, \ldots, \mathbf{x}_t, \boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_t, \Theta] \\ &= \sum_{i=1}^{p} A^{(i)} \hat{\mathbf{x}}_{t+h-i} + \sum_{j=h}^{q} B^{(j)} \boldsymbol{\varepsilon}_{t+h-j}\end{aligned}$$

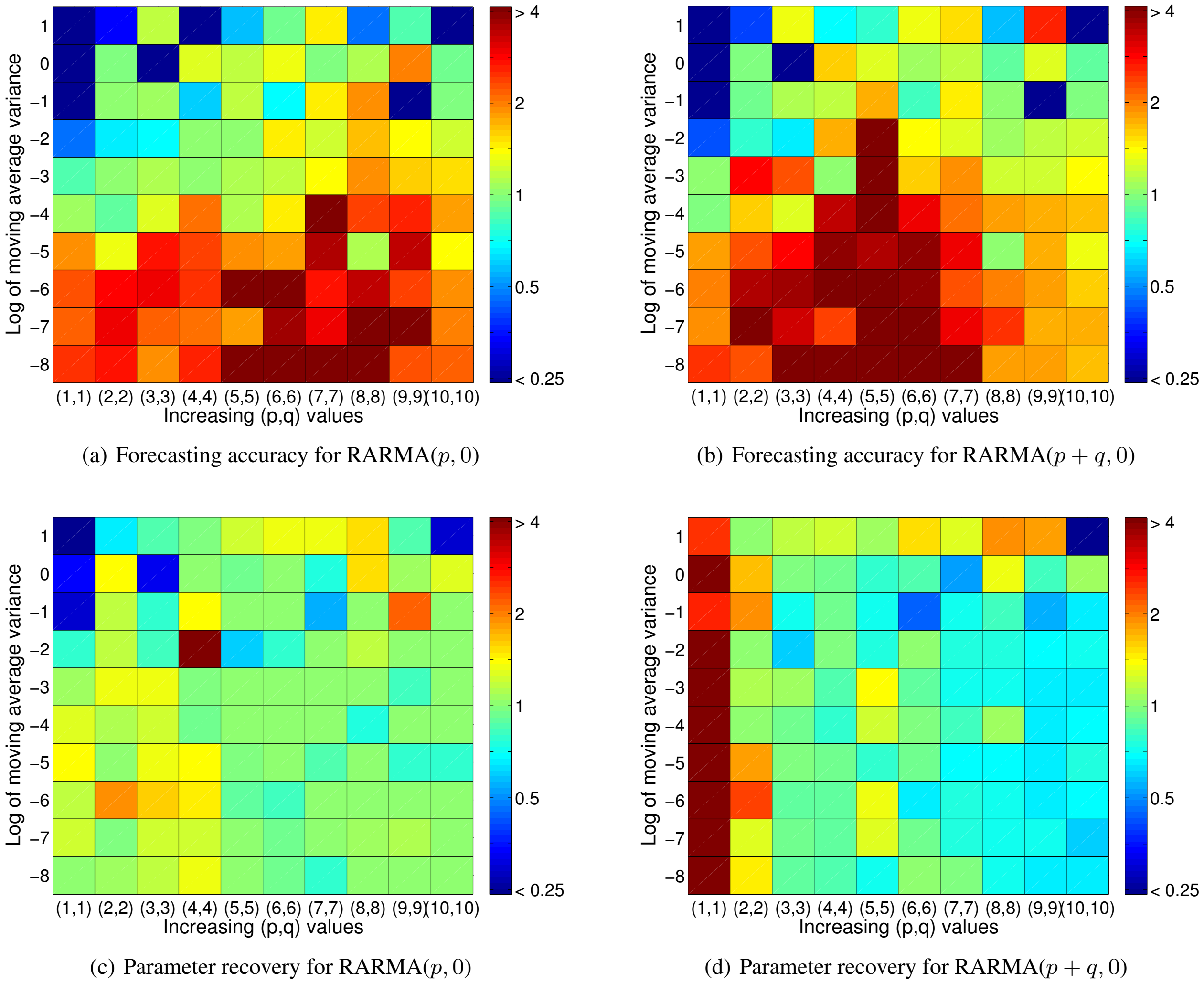
where for $h > q$, the moving average term is no longer used, and $\hat{\mathbf{x}}_{t+h-j} = \mathbf{x}_{t+h-j}$ for $t + h - j \leq t$.

If, however, the innovation variables $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_t$ are not observed, then $\mathbf{x}_{t+1}$ becomes dependent on the entire history $\mathbf{x}_1, \ldots, \mathbf{x}_t$. The one step conditional expectation then becomes

$$\begin{aligned}\tilde{\mathbf{x}}_{t+1} &= E[\mathbf{x}_{t+1} | \mathbf{x}_1, \ldots, \mathbf{x}_t, \Theta] \\ &= \sum_{i=1}^{p} A^{(i)} \mathbf{x}_{t+1-i} + \sum_{j=1}^{q} \tilde{B}_{(t)}^{(j)} \tilde{\boldsymbol{\varepsilon}}_{t+1-j}\end{aligned}$$

where $\tilde{\boldsymbol{\varepsilon}}_t = \mathbf{x}_t - \tilde{\mathbf{x}}_t$, and the $\tilde{B}_{(t)}^{(j)}$ can be efficiently computed, recursively, from the previous $\tilde{B}_{(n)}^{(k)}$ and the original parameters; see for example (Brockwell and Davis 2002, Sec. 3.3) for details. This leads to the optimal h step predictor that can be efficiently computed using the same recursively updated quantities

$$\begin{aligned}\tilde{\mathbf{x}}_{t+h} &= E[\mathbf{x}_{t+h} | \mathbf{x}_1, \ldots, \mathbf{x}_t, \Theta] \\ &= \sum_{i=1}^{p} A^{(i)} \tilde{\mathbf{x}}_{t+h-i} + \sum_{j=h}^{t+h-1} \tilde{B}_{(t+h-1)}^{(j)} \tilde{\boldsymbol{\varepsilon}}_{t+h-j}.\end{aligned}$$



(a) Forecasting accuracy for RARMA($p, 0$)

(b) Forecasting accuracy for RARMA($p + q, 0$)

(c) Parameter recovery for RARMA($p, 0$)

(d) Parameter recovery for RARMA($p + q, 0$)

Figure 4: The relative error is reported between RARMA(p, q) and the RARMA(p,0) and RARMA($p + q$,0): err(RARMA($q = 0$)) / err(RARMA($q > 0$)). The plot is symmetric, where at 4, RARMA(p, q) has 4x lower error (good), and at 0.25, has 4x higher error (bad). The dimension is set to $n = p$ and $50 + p + q$ training samples. The x-axis shows increasing lag and the y-axis increasing moving average variance. As the variance is increased beyond $\exp(-3)$, using the MEAN as a predictor begins to outperform all three methods and the moving average component begins to dominate the autoregressive component. For **(a)** and **(b)**, the comparison is with respect to forecasting accuracy for a horizon of 10, measured with ℓ_1 error. For **(c)** and **(d)**, the comparison is with respect to the ℓ_1 error between the recovered A parameters and the true parameters, cut-off at p for RARMA($p + q, 0$). Interestingly, it appears that the accuracy of A is not crucial for forecasting performance, as RARMA(p, q) outperforms RARMA($p, 0$) for most reasonable innovation variance in terms of forecasting error, but not in terms of accuracy of the underlying A.