Linear Dynamics: Clustering without identification

Chloe Ching-Yun Hsu[†]
University of California, Berkeley

Michaela Hardt[†]
Amazon

Moritz Hardt[†]
University of California, Berkeley

Abstract

Linear dynamical systems are a fundamental and powerful parametric model class. However, identifying the parameters of a linear dynamical system is a venerable task, permitting provably efficient solutions only in special cases. This work shows that the eigenspectrum of unknown linear dynamics can be identified without full system identification. We analyze a computationally efficient and provably convergent algorithm to estimate the eigenvalues of the state-transition matrix in a linear dynamical system.

When applied to time series clustering, our algorithm can efficiently cluster multi-dimensional time series with temporal offsets and varying lengths, under the assumption that the time series are generated from linear dynamical systems. Evaluating our algorithm on both synthetic data and real electrocardiogram (ECG) signals, we see improvements in clustering quality over existing baselines.

1 Introduction

Linear dynamical system (LDS) is a simple yet general model for time series. Many machine learning models are special cases of linear dynamical systems [Roweis and Ghahramani, 1999], including principal component analysis (PCA), mixtures of Gaussians, Kalman filter models, and hidden Markov models.

When the states are hidden, LDS parameter identification has provably efficient solutions only in special cases, see for example [Hazan et al., 2018, Hardt et al., 2018, Simchowitz et al., 2018]. In practice, the expectation–maximization (EM) algorithm [Ghahramani and Hinton, 1996] is often used for LDS parameter estimation, but it is inherently non-convex and can often get stuck in local minima [Hazan et al., 2018]. Even when full system identification is hard, *is there still hope to learn meaningful information about linear dynamics without learning all system parameters*? We provide a positive answer to this question.

We show that the eigenspectrum of the state-transition matrix of unknown linear dynamics can be identified without full system identification. The eigenvalues of the state-transition matrix play a significant role in determining the properties of a linear system. For example, in two dimensions, the eigenvalues determine the stability of a linear dynamical system. Based on the trace and the determinant of the state-transition matrix, we can classify a linear system as a stable node, a stable spiral, a saddle, an unstable node, or an unstable spiral.

To estimate the eigenvalues, we utilize a fundamental correspondence between linear systems and Autoregressive-Moving-Average (ARMA) models. We establish bi-directional perturbation bounds to prove that two LDSs have similar eigenvalues if and only if their output time series have similar auto-regressive parameters. Based on a consistent estimator for the autoregressive model parameters of ARMA models [Tsay and Tiao, 1984], we propose a regularized iterated least-squares regression method to estimate the LDS eigenvalues. Our method runs in time linear in the sequence length T and converges to true eigenvalues at the rate $O_p(T^{-1/2})$.

As one application, our eigenspectrum estimation algorithm gives rise to a simple approach for time series clustering: First use regularized iterated least-squares regression to fit the autoregressive parameters; then cluster the fitted autoregressive parameters.

This simple and efficient clustering approach captures similarity in eigenspectrums, assuming each time series comes from an underlying linear dynamical system. It is a suitable similarity measure where the main goal for clustering is to characterize state-transition dynamics regardless of change of basis, particularly relevant when

[†]This work was done at Google.
Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

there are multiple data sources with different measurement procedures. Our approach bypasses the challenge of LDS full parameter estimation, while enjoying the natural flexibility to handle multi-dimensional time series with time offsets and partial sequences.

To verify that our method efficiently learns system eigenvalues on synthetic and real ECG data, we compare our approach to existing baselines, including model-based approaches based on LDS, AR and ARMA parameter estimation, and PCA, as well as model-agnostic clustering approaches such as dynamic time warping [Cuturi and Blondel, 2017] and k-Shape [Paparrizos and Gravano, 2015].

Organization. We review LDS and ARMA models in Sec. 3. In Sec. 4 we discuss the main technical results around the correspondence between LDS and ARMA. Sec. 5 presents the regularized iterated regression algorithm, a consistent estimator of autoregressive parameters in ARMA models with applications to clustering. We carry out eigenvalue estimation and clustering experiments on synthetic data and real ECG data in Sec. 6. In the appendix, we describe generalizations to observable inputs and multidimensional outputs, and include additional simulation results.

2 Related Work

Linear dynamical system identification. The LDS identification problem has been studied since the 60s [Kalman, 1960], yet the theoretical bounds are still not fully understood. Recent provably efficient algorithms [Simchowitz et al., 2018, Hazan et al., 2018, Hardt et al., 2018, Dean et al., 2017] require setups that are not best-suited for time series clustering, such as assuming observable states and focusing on predition error instead of parameter recovery.

On recovering system parameters without observed states, Tsiamis et al. recently study a subspace identification algorithm with non-asymptotic $O(T^{-1/2})$ rate [Tsiamis and Pappas, 2019]. While our analysis does not provide finite sample complexity bounds, our simple algorithm achieves the same rate asymptotically.

Model-based time series clustering. Common model choices for clustering include Gaussian mixture models [Biernacki et al., 2000], autoregressive integrated moving average (ARIMA) models [Kalpakis et al., 2001], and hidden Markov models [Smyth, 1997]. Gaussian mixture models, ARIMA models, and hidden Markov models are all special cases of the more general linear dynamical system model [Roweis and Ghahramani, 1999].

Linear dynamical systems have used to cluster video trajectories [Chan and Vasconcelos, 2005, Afsari et al., 2012, Vishwanathan et al., 2007], where the observed time series are higher dimensional than the hidden state dimension. Our work is motivated by the more challenging sitatuion with a single or a few output dimensions, common in climatology, energy consumption, finance, medicine, etc.

Compared to ARMA-parameter based clustering, our method only uses the AR half of the parameters which we show to enjoy more reliable convergence. We also differ from AR-model based clustering because fitting AR to an ARMA process results in biased estimates.

Autoregressive parameter estimation. Existing spectral analysis methods for estimating AR parameters in ARMA models include high-order Yule-Walker (HOYW), MUSIC, and ESPRIT [Stoica et al., 2005, Stoica et al., 1988]. Our method is based on iterated regression [Tsay and Tiao, 1984], a more flexible method for handling observed exogenous inputs (see Appendix C) in the ARMAX generalization.

3 Preliminaries

3.1 Linear dynamical systems

A discrete-time linear dynamical system (LDS) with parameters $\Theta = (A, B, C, D)$ receives inputs $x_1, \cdots, x_T \in \mathbb{R}^k$, has hidden states $h_0, \cdots, h_T \in \mathbb{R}^n$, and generates outputs $y_1, \cdots, y_T \in \mathbb{R}^m$ according to the following time-invariant recursive equations:

$$\begin{aligned} h_t &= Ah_{t-1} + Bx_t + \zeta_t \\ y_t &= Ch_t + Dx_t + \xi_t. \end{aligned} \tag{1}$$

Assumptions. We assume that the stochastic noise ζ_t and ξ_t are diagonal Gaussians. We also assume the system is observable, i.e. $C, CA, CA^2, \cdots, CA^{n-1}$ are linearly independent. When the LDS is not observable, the ARMA model for the output series can be reduced to lower AR order, and there is not enough information in the output series to recover all the full eigenspectrum.

The model equivalence theorem (Theorem 4.1) and the approximation theorem (Theorem 4.2) do not require any additional assumptions for any real matrix A. When additionally assuming A only has simple eigenvalues in $\mathbb{C}$, i.e. each eigenvalue has multiplicity 1, we give a better convergence bound.

Distance between linear dynamical systems. With the main goal to characterize state-transition dynamics, we view systems as equivalent up to change of basis, and use the ℓ_2 distance of the spectrum of the transition matrix A, i.e. $d(\Theta_1, \Theta_2) = \|\lambda(A_1) - \lambda(A_2)\|_2$, where $\lambda(A_1)$ and $\lambda(A_2)$ are the spectrum of A_1 and A_2 in sorted order. This distance definition satisfies non-negativity, identity, symmetry, and triangle inequality.

Two very different time series could still have small distance in eigenspectrum. This is by design to allow the flexiblity for different measurement procedures, which mathematically correspond to different C matricies. When there are multiple data sources with different measurement procedures, our approach can compare the underlying dynamics of time series across sources.

Jordan canonical basis. Every square real matrix is similar to a complex block diagonal matrix known as its Jordan canonical form (JCF). In the special case for diagonalizable matrices, JCF is the same as the diagonal form. Based on JCF, there exists a canonical basis $\{e_i\}$ consisting only of eigenvectors and generalized eigenvectors of A. A vector v is a generalized eigenvector of rank μ with corresponding eigenvalue λ if $(\lambda I - A)^{\mu} v = 0$ and $(\lambda I - A)^{\mu - 1} v \neq 0$.

3.2 Autoregressive-moving-average models

The autoregressive-moving-average (ARMA) model combines the autoregressive (AR) model and the moving-average (MA) model. The AR part involves regressing the variable with respect its lagged past values, while the MA part involves regressing the variable against past error terms.

Autoregressive model. The AR model describes how the current value in the time series depends on the lagged past values. For example, if the GDP realization is high this quarter, the GDP in the next few quarters are likely high as well. An autoregressive model of order p, noted as AR(p), depends on the past p steps,

$$y_t = c + \Sigma_{i=1}^{p} \varphi_i y_{t-i} + \epsilon_t,$$

where $\varphi_1, \cdots, \varphi_p$ are autoregressive parameters, c is a constant, and ϵ_t is white noise.

When the errors are normally distributed, the ordinary least squares (OLS) regression is a conditional maximum likelihood estimator for AR models yielding optimal estimates [Durbin, 1960].

Moving-average model. The MA model, on the other hand, captures the delayed effects of unobserved random shocks in the past. For example, changes in winter weather could have a delayed effect on food harvest in the next fall. A moving-average model of order q, noted as MA(q), depends on unobserved lagged errors in the past q steps,

$$y_t = c + \epsilon_t + \Sigma_{i=1}^{q} \theta_i \epsilon_{t-i},$$

where $\theta_1, \cdots, \theta_q$ are moving-average parameters, c is a constant, and the errors ϵ_t are white noise.

ARMA model. The autoregressive-moving-average (ARMA) model, denoted as ARMA(p, q), merges AR(p) and MA(q) models to consider dependencies both on past time series values and past unpredictable shocks,

$$y_t = c + \epsilon_t + \Sigma_{i=1}^{p} \varphi_i y_{t-i} + \Sigma_{i=1}^{q} \theta_i \epsilon_{t-i}.$$

ARMAX model. ARMA can be generalized to autoregressive–moving-average model with exogenous inputs (ARMAX).

$$y_t = c + \epsilon_t + \Sigma_{i=1}^{p} \varphi_i y_{t-i} + \Sigma_{i=1}^{q} \theta_i \epsilon_{t-i} + \Sigma_{i=0}^{r} \gamma_i x_{t-i},$$

where $\{x_t\}$ is a known external time series, possibly multidimensional. In case x_t is a vector, the parameters γ_i are also vectors.

Estimating ARMA and ARMAX models is significantly harder than AR, since the model depends on unobserved variables and the maximum likelihood equations are intractable [Durbin, 1959, Choi, 2012]. Maximum likelihood estimation (MLE) methods are commonly used for fitting ARMA and ARMAX [Guo, 1996, Bercu, 1995, Hannan et al., 1980], but have converge issues. Although regression methods are also used in practice, OLS is a *biased* estimator for ARMA models [Tiao and Tsay, 1983].

Lag operator. We also introduce the lag operator, a concise way to describe ARMA models [Granger and Morris, 1976], defined as $Ly_t = y_{t-1}$. The lag operator could be raise to powers, or form polynomials. For example, $L^3 y_t = y_{t-3}$, and $(a_2 L^2 + a_1 L + a_0) y_t = a_2 y_{t-2} + a_1 y_{t-1} + a_0 y_t$. The lag polynomials can be multiplied or inverted. An AR(p) model can be characterized by

$$\Phi(L) y_t = c + \epsilon_t,$$

where $\Phi(L) = 1 - \varphi_1 L - \cdots - \varphi_p L^p$ is a polynomial of the lag operator L of degree p. For example, any AR(2) model can be described as $(1 - \varphi_1 L - \varphi_2 L^2) y_t = c + \epsilon_t$.

Similarly, an MA(q) can be characterized by a polynomial $\Psi(L) = \theta_q L^q + \cdots + \theta_1 L + 1$ of degree q,

$$y_t = c + \Psi(L) \epsilon_t.$$

For example, for an MA(2) model the equation would be $y_t = c + (\theta_2 L^2 + \theta_1 L + 1) \epsilon_t$.

Merging the two and adding dependency to exogenous input, we can write an ARMAX(p, q, r) model as

$$\Phi(L) y_t = c + \Psi(L) \epsilon_t + \Gamma(L) x_t \qquad (2)$$

where Φ, Ψ, and Γ are polynomials of degree p, q and r. When the exogenous time series x_t is multidimensional, $\Gamma(L)$ is a vector of degree-r polynomials.

4 Learning eigenvalues without system identification

This section provides theoretical foundations for learning LDS eigenvalues from autoregressive parameters without full system identification. While general model equivalence between LDS and ARMA(X) is known [Åström and Wittenmark, 2013, Kailath, 1980], we provide detailed analysis of the exact correspondence between the LDS characteristic polynomial and the ARMA(X) autoregressive parameters along with perturbation bounds.

4.1 Model equivalence

We show that the output series from any LDS can be seen as generated by an ARMAX model, whose AR parameters contain full information about the LDS eigenvalues.

Theorem 4.1. *Let $y_t \in \mathbb{R}^m$ be the outputs from a linear dynamical system with parameters $\Theta = (A, B, C, D)$, hidden dimension n, and inputs $x_t \in \mathbb{R}^k$. Each dimension of y_t can be generated by an ARMAX($n, n, n-1$) model, whose autoregressive parameters $\varphi_1, \cdots, \varphi_n$ can recover the characteristic polynomial of A by $\chi_A(\lambda) = \lambda^n - \varphi_1 \lambda^{n-1} - \cdots - \varphi_n$.*

In the special case where the LDS has no external inputs, the ARMAX model is an ARMA(n, n) model.

See Appendix A for the full proof.

As a high-level proof sketch: We first analyze the hidden state projected to (generalized) eigenvector directions in Lemma A.2. We show that for a (generalized) eigenvector e of the adjoint A^* of the transition matrix with eigenvalue λ and rank μ, the time series obtained from applying the lag operator polynomial $(1-\lambda L)^\mu$ to $\langle h_t, e\rangle$ can be expressed as a linear combination of the past k inputs $x_t, \cdots, x_{t-k+1}$. Since A is real-valued, A and its adjoint A^* share the same characteristic polynomial χ_A.

We then consider the lag operator polynomial $\chi_A^\dagger(L) = L^n \chi_A(L^{-1})$, and show that the time series obtained from applying $\chi_A^\dagger(L)$ applied to any (generalized) eigenvector direction is a linear combination of the past k inputs. We use this on the Jordan canonical basis for A^* that consists of (generalized) eigenvectors. From there, we conclude $\chi_A^\dagger$ is the autoregressive lag polynomial that contains the autoregressive coefficients.

The converse of Theorem 4.1 also holds. An ARMA(p, q) model can be seen as a ($p+q$)-dimensional LDS where the state encodes the relevant past values and error terms.

Corollary 4.1. *The output series of two linear dynamical systems have the same autoregressive parameters if and only if they have the same non-zero eigenvalues with the same multiplicities.*

Proof. By Theorem 4.1, the autoregressive parameters are determined by the characteristic polynomial. Two LDSs of the same dimension have the same autoregressive parameters if and only if they have the same characteristic polynomials, and hence the same eigenvalues with the same multiplicities. Two LDSs of different dimensions $n_1 < n_2$ can have the same autoregressive parameters if and only if $\chi_{A_1}(\lambda) = \chi_{A_2}(\lambda)\lambda^{n_2-n_1}$ and $\varphi_{n_1+1} = \cdots = \varphi_{n_2} = 0$, in which case they have the same non-zero eigenvalues with same multiplicities. □

It is possible for two LDSs with different dimensions to have the same AR coefficients, if the higher-dimensional system has additional zero eigenvalues. Whether over-parameterized models indeed learn additional zero eigenvalues requires further empirical investigation.

4.2 Approximation theorems for LDS eigenvalues

We show that small error in the AR parameter estimation guarantees a small error in the eigenvalue estimation. This implies that an effective estimation algorithm for the AR parameters in ARMAX models leads to effective estimation of LDS eigenvalues.

General $(1/n)$-exponent bound

Theorem 4.2. *Let y_t be the outputs from an n-dimensional linear dynamical system with parameters $\Theta = (A, B, C, D)$, eigenvalues $\lambda_1, \cdots, \lambda_n$, and hidden inputs. Let $\hat{\Phi} = (\hat{\varphi}_1, \cdots, \hat{\varphi}_n)$ be the estimated autoregressive parameters for $\{y_t\}$ with error $\|\hat{\Phi} - \Phi\| = \epsilon$, and let $r_1, \cdots, r_n$ be the roots of the polynomial $1 - \hat{\varphi}_1 z - \cdots - \hat{\varphi}_n z^n$.*

Assuming the LDS is observable, the roots converge to the true eigenvalues with convergence rate $\mathcal{O}(\epsilon^{1/n})$. If all eigenvalues of A are simple (no multiplicity), then the convergence rate is $\mathcal{O}(\epsilon)$.

Without additional assumptions, the $\frac{1}{n}$-exponent in the above general bound is tight. As an example, $z^2 - \epsilon$ has roots $z \pm \sqrt{\epsilon}$. The general phenomenon that a root with multiplicity m can split into m roots at rate $O(\epsilon^m)$ is related to the regular splitting property [Hryniv and Lancaster, 1999, Lancaster et al., 2003].

Linear bound for simple eigenvalues

Under the additional assumption that all the eigenvalues are simple (no multiplicity), we derive a better $O(\epsilon)$ bound instead of $O(\epsilon^{1/n})$. We show small perturbation in AR parameters results in small perturbation in companion matrix, and small perturbation in companion matrix results in small perturbation in eigenvalues.

We defer the full proofs to Appendix B, but describe the proof ideas here.

For a monic polynomial $\Phi(u) = z^n + \varphi_1 z^{n-1} + \cdots + \varphi_{n-1} z + \varphi_n$, the *companion matrix*, also known as the controllable canonical form in control theory, is the square matrix

$$C(\Phi) = \begin{bmatrix} 0 & 0 & \dots & 0 & -\varphi_n \\ 1 & 0 & \dots & 0 & -\varphi_{n-1} \\ 0 & 1 & \dots & 0 & -\varphi_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & -\varphi_1 \end{bmatrix}.$$

The matrix $C(\Phi)$ is the companion in the sense that it has Φ as its characteristic polynomial.

In relation to an autoregressive AR(p) model, the companion matrix corresponds to the transition matrix in the linear dynamical system when we encode the values form the past p lags as a p-dimensional state

$$h_t = \begin{bmatrix} y_{t-p+1} & \cdots & y_{t-1} & y_t \end{bmatrix}^T.$$

If $y_t = \varphi_1 y_{t-1} + \cdots + \varphi_p y_{t-p}$, then $h_t =$

$$\begin{bmatrix} y_{t-p+1} \\ y_{t-p+2} \\ \cdots \\ y_{t-1} \\ y_t \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ \varphi_p & \varphi_{p-1} & \varphi_{p-2} & \dots & \varphi_1 \end{bmatrix} \begin{bmatrix} y_{t-p} \\ y_{t-p+1} \\ \cdots \\ y_{t-2} \\ y_{t-1} \end{bmatrix}$$
$$= C(-\Phi)^T h_{t-1}. \tag{3}$$

We then use matrix eigenvalue perturbation theory results on the companion matrix for the desired bound.

Lemma 4.1 (Theorem 6 in [Lancaster et al., 2003])**.** *Let $L(\lambda, \epsilon)$ be an analytic matrix function with semisimple eigenvalue λ_0 at $\epsilon = 0$ of multiplicity M. Then there are exactly M eigenvalues $\lambda_i(\epsilon)$ of $L(\lambda, \epsilon)$ for which $\lambda_i(\epsilon) \to \lambda_0$ as $\epsilon \to 0$, and for these eigenvalues*

$$\lambda_i(\epsilon) = \lambda_0 + \lambda_i' \epsilon + o(\epsilon). \tag{4}$$

Explicit bound on condition number

When the LDS has all simple eigenvalues, we provide a more explicit bound on the condition number.

Theorem 4.3. *In the same setting as above in Theorem 4.2, when all eigenvalues of A are simple, $|r_j - \lambda_j| \le \kappa\epsilon + o(\epsilon^2)$, then the condition number κ is bounded by*

$$\frac{1}{\prod_{k \neq j} |\lambda_j - \lambda_k|} \le \kappa \le$$
$$\frac{\sqrt{n}}{\prod_{k \neq j} |\lambda_j - \lambda_k|} \left(\max(1, |\lambda_j|)\right)^{n-1} \left(1 + \rho(A)^2\right)^{\frac{n-1}{2}},$$

where $\rho(A)$ is the spectral radius, i.e. largest absolute value of its eigenvalues.

In particular, when $\rho(C) \le 1$, i.e. when the matrix is Lyapunov stable, then the absolute difference between the root from the auto-regressive method and the eigenvalue is bounded by $|r_j - \lambda_j| \le \frac{\sqrt{n}(\sqrt{2})^{n-1}}{\prod_{k \neq j} |\lambda_j - \lambda_k|}\epsilon + o(\epsilon^2)$.

In Appendix B, we derive the explicit formula in Theorem 4.3 by conjugating the companion matrix by a Vandermonde matrix to diagonalize it and invoking the explicit inverse formula of Vandermonde matrices.

5 Estimation of ARMA autoregressive parameters

In general, learning ARMA models is hard, since the output series depends on unobserved error terms. Fortunately, for our purpose we are only interested in the autoregressive parameters, that are easier to learn since the past values of the time series are observed.

The autoregressive parameters in an ARMA(p, q) model are not equivalent to the pure AR(p) parameters for the same time series. For AR(p) models, ordinary least squares (OLS) regression is a consistent estimator of the autoregressive parameters [Lai and Wei, 1983]. However, for ARMA(p, q) models, due to the serial correlation in the error term $\epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$, the OLS estimates for autoregressive parameters can be biased [Tiao and Tsay, 1983].

Regularized iterated regression. Iterated regression [Tsay and Tiao, 1984] is a consistent estimator for the AR parameters in ARMA models. While iterated regression is theoretically well-grounded, it tends to over-fit and results in excessively large parameters. To avoid over-fitting, we propose a slight modification with regularization, which keeps the same theoretical guarantees and yields better practical performance.

We also generalize the method to handle *multidimensional outputs* from the LDS and *observed inputs* by using ARMAX instead of ARMA models, as described in details in Appendix C as Algorithm 2.

The i-th iteration of the regression only uses error terms from the past i lags. The initial iteration is an ARMA($n, 0$) regression, the first iteration is an ARMA($n, 1$) regression, and so forth until ARMA(n, n) in the last iteration.

Algorithm 1: Regularized iterated regression for autoregressive parameter estimation

Input: Time series $\{y_t\}_{t=1}^T$, target hidden state dimension n, and regularization coefficient α.
Initialize error term estimates $\hat{\epsilon}_t = 0$ for $t = 1, \dots, T$;
for $i = 0, \cdots, n$ **do**
  Perform ℓ_2-regularized least squares regression to estimate $\hat{\varphi}_j$, $\hat{\theta}_j$, and $\hat{c}$ in $y_t = \sum_{j=1}^n \hat{\varphi}_j y_{t-j} + \sum_{j=1}^i \hat{\theta}_j \hat{\epsilon}_{t-j} + \hat{c}$ with regularization strength α only on the $\hat{\theta}_j$ terms;
  Update $\hat{\epsilon}_t$ to be the residuals from the most recent regression;
end
Return $\hat{\varphi}_1, \cdots, \hat{\varphi}_n$.

Time complexity. The iterated regression involves $n+1$ steps of least squares regression each on at most $2n+1$ variables. Therefore, the total time complexity of Algorithm 1 is $O(n^3T+n^4)$, where T is the sequence length and n is the hidden state dimension.

Convergence rate. The consistency and the convergence rate of the estimator is analyzed in [Tsay and Tiao, 1984]. Adding regularization does not change the asymptotic property of the estimator.

Theorem 5.1 ([Tsay and Tiao, 1984])**.** *Suppose that y_t is an ARMA(p, q) process, stationary or not. The estimated autoregressive parameters $\hat{\Phi} = (\hat{\varphi}_1, \cdots, \hat{\varphi}_n)$ from iterated regression converges in probability to the true parameters with rate*

$$\hat{\Phi} = \Phi + O_p(T^{-1/2}),$$

or more explicitly, convergence in probability means that for all ϵ, $\lim_{T\to\inf} \Pr(T^{1/2}|\hat{\Phi} - \Phi| > \epsilon) = 0$.

5.1 Applications to clustering

The task of clustering depends on an appropriate distance measure for the clustering purpose. When there are multiple data sources for time series with comparable dynamics but measured by different measurement procedures, one might hope to cluster time series based only on the state-transition dynamics.

We can observe from the LDS definition (1) that two LDSs with parameters (A, B, C, D) and (A', B', C', D') are equivalent if $A' = P^{-1}AP, B' = P^{-1}B, C' = CP, D' = DP$, and $h'_t = P^{-1}h_t$ under change of basis by some non-singular matrix P. Therefore, to capture the state-transition dynamics while allowing for flexibility in the measurement matrix C, the distance measure should be invariant under change of basis. We choose to use the eigenspectrum distance between state-transition matrices, a natural distance choice that is invariant under change of basis.

In previous sections, our theoretical analysis shows that AR parameters in ARMA time series models can effectively estimate the eigenspectrum of underlying LDSs. We therefore propose a simple time series clustering algorithm: 1) first use iterated regression to estimate the autoregressive parameters in ARMA models for each times series, and 2) then apply any standard clustering algorithm such as K-means on the distance between autoregressive parameters.

Our method is very flexible. It handles multi-dimensional data, as Theorem 4.1 suggests that any output series from the same LDS should share the same autoregressive parameters. It can also handle exogenous inputs as illustrated in Algorithm 2 in Appendix C. It is scale, shift, and offset invariant, as the autoregressive parameters in ARMA models are. It accommodates missing values in partial sequences as we can still perform OLS after dropping the rows with missing values. It also allows sequences to have different lengths, and could be adapted to handle sequences with different sampling frequencies, as the compound of multiple steps of LDS evolution is still linear.

6 Experiments

We experimentally evaluate the quality and efficiency of the clustering from our method and compare it to existing baselines. The source code is available online at `https://github.com/chloechsu/ldseig`.

6.1 Methods

- `ARMA`: K-means on AR parameters in ARMA(n, n) model estimated by regularized iterated regression as we proposed in Algorithm 1.
- `ARMA_MLE`: K-means on AR parameters in ARMA(n, n) model estimated by the MLE method using statsmodels [Seabold and Perktold, 2010].
- `AR`: K-means on AR parameters in AR(n) model estimated by OLS using statsmodels.
- `LDS`: K-means on estimated LDS eigenvalues. We estimate the LDS eigenvalues with the pylds package [Johnson and Linderman, 2018], with 100 EM steps initialized by 10 Gibbs iterations.
- `k-Shape`: A shape-based time series clustering method [Paparrizos and Gravano, 2015], using the tslearn package [Tavenard, 2017].
- `DTW`: K-medoids on dynamic time warping distance, using the dtaidistance [Meert, 2018] and pyclustering [Novikov, 2019] packages.
- `PCA`: K-means on the first n PCA components, using sklearn [Pedregosa et al., 2011].

# Clusters	Method	Adj. Mutual Info.	Adj. Rand Score	V-measure	Runtime (secs)
2	`AR`	0.06 (0.04-0.08)	0.07 (0.05-0.09)	0.07 (0.05-0.09)	1.09 (1.01-1.17)
	`ARMA`	**0.13 (0.11-0.16)**	**0.16 (0.13-0.19)**	**0.14 (0.11-0.16)**	**0.44 (0.40-0.47)**
	`ARMA_MLE`	0.02 (0.01-0.03)	0.02 (0.01-0.03)	0.03 (0.02-0.04)	70.64 (68.01-73.28)
	`DTW`	0.02 (0.01-0.03)	0.02 (0.01-0.03)	0.03 (0.02-0.04)	6.60 (6.34-6.86)
	`k-Shape`	0.03 (0.02-0.04)	0.03 (0.02-0.05)	0.04 (0.03-0.05)	28.58 (25.15-32.01)
	`LDS`	0.09 (0.06-0.12)	0.09 (0.06-0.12)	0.10 (0.07-0.12)	341.08 (328.24-353.93)
	`PCA`	-0.00 (-0.00-0.00)	-0.00 (-0.00-0.00)	0.02 (0.02-0.02)	0.45 (0.43-0.47)
3	`AR`	0.11 (0.09-0.12)	0.09 (0.07-0.10)	0.12 (0.11-0.14)	1.02 (0.93-1.10)
	`ARMA`	0.18 (0.16-0.20)	0.16 (0.14-0.18)	0.19 (0.17-0.21)	**0.42 (0.38-0.46)**
	`ARMA_MLE`	0.04 (0.03-0.05)	0.04 (0.03-0.05)	0.06 (0.05-0.07)	72.58 (69.63-75.52)
	`DTW`	0.04 (0.03-0.05)	0.03 (0.02-0.03)	0.07 (0.06-0.08)	6.63 (6.37-6.90)
	`k-Shape`	0.06 (0.05-0.07)	0.04 (0.04-0.05)	0.08 (0.07-0.09)	40.10 (34.65-45.55)
	`LDS`	**0.20 (0.18-0.23)**	**0.17 (0.15-0.20)**	**0.22 (0.19-0.24)**	338.67 (325.66-351.67)
	`PCA`	0.00 (-0.00-0.00)	0.00 (-0.00-0.00)	0.04 (0.04-0.04)	0.47 (0.45-0.49)
5	`AR`	0.17 (0.16-0.18)	0.11 (0.10-0.12)	0.22 (0.21-0.23)	0.91 (0.83-1.00)
	`ARMA`	0.22 (0.21-0.23)	0.15 (0.14-0.16)	0.26 (0.25-0.28)	**0.40 (0.35-0.45)**
	`ARMA_MLE`	0.08 (0.07-0.09)	0.05 (0.04-0.05)	0.14 (0.13-0.15)	74.00 (71.70-76.30)
	`DTW`	0.05 (0.04-0.06)	0.03 (0.02-0.03)	0.11 (0.10-0.12)	6.20 (5.86-6.55)
	`k-Shape`	0.08 (0.07-0.09)	0.05 (0.04-0.05)	0.14 (0.13-0.15)	97.83 (84.97-110.68)
	`LDS`	**0.25 (0.23-0.26)**	**0.17 (0.16-0.18)**	**0.29 (0.28-0.30)**	321.40 (304.17-338.64)
	`PCA`	0.01 (0.01-0.01)	0.00 (0.00-0.01)	0.08 (0.07-0.08)	0.52 (0.49-0.55)
10	`AR`	0.22 (0.21-0.22)	0.11 (0.11-0.12)	0.38 (0.37-0.38)	0.87 (0.77-0.98)
	`ARMA`	**0.24 (0.23-0.25)**	**0.14 (0.13-0.15)**	**0.39 (0.39-0.40)**	**0.42 (0.37-0.48)**
	`ARMA_MLE`	0.11 (0.10-0.12)	0.06 (0.05-0.06)	0.29 (0.28-0.30)	63.39 (60.49-66.30)
	`DTW`	0.06 (0.06-0.07)	0.03 (0.02-0.03)	0.25 (0.24-0.25)	5.51 (5.03-5.98)
	`k-Shape`	0.08 (0.07-0.08)	0.04 (0.03-0.04)	0.26 (0.26-0.27)	108.79 (93.41-124.18)
	`LDS`	0.23 (0.22-0.24)	0.13 (0.12-0.13)	**0.39 (0.38-0.40)**	277.45 (253.38-301.51)
	`PCA`	0.02 (0.01-0.02)	0.01 (0.00-0.01)	0.14 (0.13-0.15)	0.47 (0.44-0.51)

Table 1: Performance of clustering 100 random 2-dimensional LDSs based on their output series of length 1000, with 95% confidence intervals from 100 trials. AMI, Adj. Rand, and V-measure are the adjusted mutual information score, adjusted Rand score, and V-measure between ground truth cluster labels and learned clusters. The runtime is on an instance with 12 CPUs and 40 GB memory running Ubuntu 18.

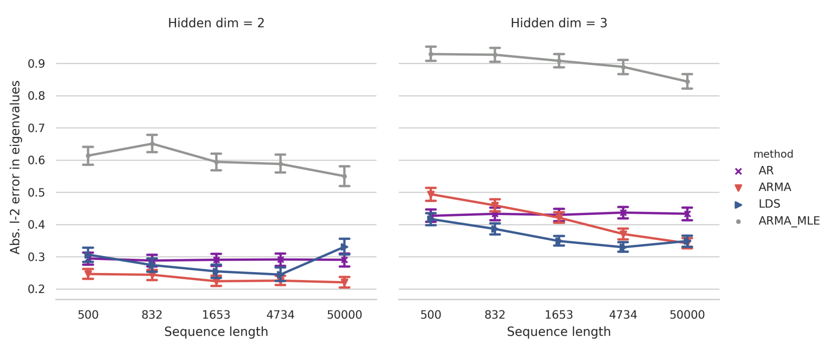

Figure 1: Absolute ℓ_2-error in eigenvalue estimation for 2-dimensional and 3-dimensional LDSs, with 95% confidence interval from 500 trials.

Method	AMI	Adj. Rand Score	V-measure
`ARMA`	0.12 (0.10-0.13)	0.12 (0.11-0.14)	0.14 (0.12-0.16)
`AR`	0.10 (0.09-0.12)	0.10 (0.09-0.12)	0.13 (0.11-0.14)
`PCA`	0.04 (0.03-0.05)	0.02 (0.02-0.03)	0.07 (0.06-0.08)
`LDS`	0.09 (0.07-0.10)	0.11 (0.10-0.13)	0.10 (0.09-0.11)
`k-Shape`	0.08 (0.07-0.10)	0.09 (0.07-0.11)	0.10 (0.09-0.12)
`DTW`	0.03 (0.02-0.04)	0.02 (0.01-0.03)	0.06 (0.05-0.07)

Table 2: Clustering performance on electrocardiogram (ECG) data separating segments of normal sinus rhythm from supraventricular tachycardia. 95% Confidence intervals are from 100 bootstrapped samples of 50 series.

6.2 Metrics of Cluster Quality

We measure cluster quality using three metrics in sklearn: V-measure [Rosenberg and Hirschberg, 2007], adjusted mutual information [Vinh et al., 2010], and adjusted Rand score [Hubert and Arabie, 1985].

6.3 Simulation

Dataset. We generate LDSs representing cluster centers with random matrices of i.i.d. Gaussian entries. From the cluster centers, we derive LDSs that are close to the cluster centers. From each LDS, we generate time series of length 1000 by drawing inputs from standard Gaussians and adding noise to the output sampled from $N(0, 0.01^2)$. More details in Appendix D.1.

Clustering performance. The iterated `ARMA` regression method and the `LDS` method yield the best clustering quality, while the iterated `ARMA` regression method is significantly faster. These results hold up for choices of different cluster quality metrics and number of clusters.

Eigenvalue Estimation. Good clustering results rely on good approximations of the LDS eigenvalue distance. Our analyses in Theorem 5.1 and Theorem 4.2 proved that the iterative `ARMA` regression algorithm can learn the LDS eigenvalues with converge rate $O_p(T^{-1/2})$. In Figure 1, we see that the observed convergence rate in simulations roughly matches the theoretical bound.

Each EM step in `LDS` runs in $O(n^3T)$. When running a constant number of EM steps, `LDS` has the same total complexity as iterated `ARMA`. We chose 100 steps based on empirical evaluation of convergence for sequence length 1000. However, longer sequences may need more EM steps to converge, which would explain the increase in `LDS` eigenvalue estimation error for sequence length 50000 in Figure 1. Depending on different implementations and initialization schemes, it is possible that the `LDS` performance can be further optimized.

`ARMA` and `LDS` have comparable eigenvalue estimation error for most configurations. While the pure `AR` approach also gives comparable estimation error on relatively short sequences, its estimation is biased, and the error does not go down as sequence length increases.

6.4 Real-world ECG data

While our simulation results show the efficacy of our method, the data generation process satisfy assumptions that may not hold on real data. As a proof-of-concept, we also test our method on real electrocardiogram (ECG) data.

Dataset. The MIT-BIH [Moody and Mark, 2001] dataset in PhysioNet [Goldberger et al., 2000] is the most common dataset for evaluating algorithms for ECG data [De Chazal et al., 2004, Yeh et al., 2012, Özbay et al., 2006, Ceylan et al., 2009].

It contains 48 half-hour recordings collected at the Beth Israel Hospital between 1975 and 1979. Each two-channel recording is digitized at a rate of 360 samples per second per channel. 15 distinct rhythms are annotated in recordings including abnormalities of cardiac rhythm (arrhythmias) by two cardiologists.

Detecting cardiac arrhythmias has stimulated research and product applications such as Apple's FDA-approved detection of atrial fibrillation [Turakhia et al., 2018]. ECG data have been modeled with AR and ARIMA models [Kalpakis et al., 2001, Corduas and Piccolo, 2008, Ge et al., 2002], and more recently convolutional neural networks [Hannun et al., 2019].

We bootstrap 100 samples of 50 time series; each bootstrapped sample consists of 2 labeled clusters: 25 series with supraventricular tachycardia and 25 series with normal sinus rhythm. Each series has length 500 which adequately captures a complete cardiac cycle. We set the `ARMA` ℓ_2-regularization coefficient to be 0.01, chosen based on our simulation results.

Results. Comparing methods outlined in Section 6.1, Table 2 shows that our method achieves the best quality closely followed by the `AR` and `LDS` methods, according to adjusted mutual information, adjusted Rand score and V-measure, while being computationally efficient.

7 Conclusion

We give a fast, simple, and provably effective method to estimate linear dynamical system (LDS) eigenvalues based on system outputs. The algorithm combines statistical techniques from the 80's with our insights on the correspondence between LDSs and ARMA models.

As a proof-of-concept, we apply the eigenvalue estimation algorithm to time series clustering. The resulting clustering approach is flexible to handle varying lengths, temporal offsets, as well as multidimensional inputs and outputs. Our efficient algorithm yields high quality clusters in simulations and on real ECG data.

While LDSs are general models encompassing mixtures of Gaussian and hidden Markov models, they may not fit all applications. It would be interesting to extend the analysis to non-linear models, and to consider model overparameterization and misspecification.

Acknowledgments We thank the anonymous reviewers for thoughtful comments, and Scott Linderman, Andrew Dai, and Eamonn Keogh for helpful guidance.

References

[Afsari et al., 2012] Afsari, B., Chaudhry, R., Ravichandran, A., and Vidal, R. (2012). Group action induced distances for averaging and clustering linear dynamical systems with applications to the analysis of dynamic scenes. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2208–2215. IEEE.

[Åström and Wittenmark, 2013] Åström, K. J. and Wittenmark, B. (2013). *Computer-controlled systems: theory and design*. Courier Corporation.

[Beauzamy, 1999] Beauzamy, B. (1999). How the roots of a polynomial vary with its coefficients: A local quantitative result. *Canadian Mathematical Bulletin*, 42(1):3–12.

[Bercu, 1995] Bercu, B. (1995). Weighted estimation and tracking for armax models. *SIAM Journal on Control and Optimization*, 33(1):89–106.

[Biernacki et al., 2000] Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7):719–725.

[Ceylan et al., 2009] Ceylan, R., Özbay, Y., and Karlik, B. (2009). A novel approach for classification of ecg arrhythmias: Type-2 fuzzy clustering neural network. *Expert Systems with Applications*, 36(3):6721–6726.

[Chan and Vasconcelos, 2005] Chan, A. B. and Vasconcelos, N. (2005). Probabilistic kernels for the classification of auto-regressive visual processes. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 846–851. IEEE.

[Choi, 2012] Choi, B. (2012). *ARMA model identification*. Springer Science & Business Media.

[Corduas and Piccolo, 2008] Corduas, M. and Piccolo, D. (2008). Time series clustering and classification by the autoregressive metric. *Computational Statistics & Data Analysis*, 52:1860–1872.

[Cuturi and Blondel, 2017] Cuturi, M. and Blondel, M. (2017). Soft-dtw: a differentiable loss function for time-series. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 894–903. JMLR. org.

[De Chazal et al., 2004] De Chazal, P., O'Dwyer, M., and Reilly, R. B. (2004). Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *IEEE transactions on biomedical engineering*, 51(7):1196–1206.

[Dean et al., 2017] Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. (2017). On the sample complexity of the linear quadratic regulator. *arXiv preprint arXiv:1710.01688*.

[Durbin, 1959] Durbin, J. (1959). Efficient estimation of parameters in moving-average models. *Biometrika*, 46(3/4):306–316.

[Durbin, 1960] Durbin, J. (1960). Estimation of parameters in time-series regression models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 22(1):139–153.

[El-Mikkawy, 2003] El-Mikkawy, M. E. (2003). Explicit inverse of a generalized vandermonde matrix. *Applied mathematics and computation*, 146(2-3):643–651.

[Ge et al., 2002] Ge, D., Srinivasan, N., and M Krishnan, S. (2002). Cardiac arrhythmia classification using autoregressive modeling. *Biomedical engineering online*, 1:5.

[Ghahramani and Hinton, 1996] Ghahramani, Z. and Hinton, G. E. (1996). Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science.

[Goldberger et al., 2000] Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220. Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[Granger and Morris, 1976] Granger, C. W. and Morris, M. J. (1976). Time series modelling and interpretation. *Journal of the Royal Statistical Society: Series A (General)*, 139(2):246–257.

[Greenbaum et al., 2019] Greenbaum, A., Li, R.-c., and Overton, M. L. (2019). First-order perturbation theory for eigenvalues and eigenvectors. *arXiv preprint arXiv:1903.00785*.

[Guo, 1996] Guo, L. (1996). Self-convergence of weighted least-squares with applications to stochastic adaptive control. *IEEE transactions on automatic control*, 41(1):79–89.

[Hannan et al., 1980] Hannan, E. J., Dunsmuir, W. T., and Deistler, M. (1980). Estimation of vector armax models. *Journal of Multivariate Analysis*, 10(3):275–295.

[Hannun et al., 2019] Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., Tison, G. H., Turakhia, M. P., Bourn, C., and Ng, A. Y. (2019). Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine*, 15.

[Hardt et al., 2018] Hardt, M., Ma, T., and Recht, B. (2018). Gradient descent learns linear dynamical systems. *The Journal of Machine Learning Research*, 19(1):1025–1068.

[Hazan et al., 2018] Hazan, E., Lee, H., Singh, K., Zhang, C., and Zhang, Y. (2018). Spectral filtering for general linear dynamical systems. In *Advances in Neural Information Processing Systems*, pages 4634–4643.

[Hryniv and Lancaster, 1999] Hryniv, R. and Lancaster, P. (1999). On the perturbation of analytic matrix functions. *Integral Equations and Operator Theory*, 34(3):325–338.

[Hubert and Arabie, 1985] Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

[Ipsen and Rehman, 2008] Ipsen, I. C. and Rehman, R. (2008). Perturbation bounds for determinants and characteristic polynomials. *SIAM Journal on Matrix Analysis and Applications*, 30(2):762–776.

[Johnson and Linderman, 2018] Johnson, M. and Linderman, S. (2018). Pylds: Bayesian inference for linear dynamical systems. `https://github.com/mattjj/pylds`.

[Kailath, 1980] Kailath, T. (1980). *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

[Kalpakis et al., 2001] Kalpakis, K., Gada, D., and Puttagunta, V. (2001). Distance measures for effective clustering of arima time-series. In *Proceedings 2001 IEEE international conference on data mining*, pages 273–280. IEEE.

[Kalpakis et al., 2001] Kalpakis, K., Gada, D., and Puttagunta, V. (2001). Distance measures for effective clustering of arima time-series. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 273–280.

[Lai and Wei, 1983] Lai, T. and Wei, C. (1983). Asymptotic properties of general autoregressive models and strong consistency of least-squares estimates of their parameters. *Journal of multivariate analysis*, 13(1):1–23.

[Lancaster et al., 2003] Lancaster, P., Markus, A., and Zhou, F. (2003). Perturbation theory for analytic matrix functions: the semisimple case. *SIAM Journal on Matrix Analysis and Applications*, 25(3):606–626.

[Meert, 2018] Meert, Wannes, e. a. (2018). Dtaidistance. `https://dtaidistance.readthedocs.io/en/latest/`.

[Moody and Mark, 2001] Moody, G. and Mark, R. (2001). The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine : the quarterly magazine of the Engineering in Medicine & Biology Society*, 20:45–50.

[Novikov, 2019] Novikov, A. (2019). Pyclustering: data mining library. *Journal of Open Source Software*, 4(36):1230.

[Özbay et al., 2006] Özbay, Y., Ceylan, R., and Karlik, B. (2006). A fuzzy clustering neural network architecture for classification of ecg arrhythmias. *Computers in Biology and Medicine*, 36(4):376–388.

[Paparrizos and Gravano, 2015] Paparrizos, J. and Gravano, L. (2015). k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1855–1870. ACM.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

[Rosenberg and Hirschberg, 2007] Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

[Roweis and Ghahramani, 1999] Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345.

[Seabold and Perktold, 2010] Seabold, S. and Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, volume 57, page 61. Scipy.

[Simchowitz et al., 2018] Simchowitz, M., Mania, H., Tu, S., Jordan, M. I., and Recht, B. (2018). Learning without mixing: Towards a sharp analysis of linear system identification. *arXiv preprint arXiv:1802.08334*.

[Smyth, 1997] Smyth, P. (1997). Clustering sequences with hidden markov models. In *Advances in neural information processing systems*, pages 648–654.

[Stoica et al., 1988] Stoica, P., Friedlander, B., and Söderström, T. (1988). A high-order yule-walker method for estimation of the ar parameters of an arma model. *Systems & control letters*, 11(2):99–105.

[Stoica et al., 2005] Stoica, P., Moses, R. L., et al. (2005). *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ.

[Tavenard, 2017] Tavenard, R. (2017). tslearn: A machine learning toolkit dedicated to time-series data. *URL https://github. com/rtavenar/tslearn*.

[Tiao and Tsay, 1983] Tiao, G. C. and Tsay, R. S. (1983). Consistency properties of least squares estimates of autoregressive parameters in arma models. *The Annals of Statistics*, pages 856–871.

[Tsay and Tiao, 1984] Tsay, R. S. and Tiao, G. C. (1984). Consistent estimates of autoregressive parameters and extended sample autocorrelation function for stationary and nonstationary arma models. *Journal of the American Statistical Association*, 79(385):84–96.

[Tsiamis and Pappas, 2019] Tsiamis, A. and Pappas, G. J. (2019). Finite sample analysis of stochastic system identification. *arXiv preprint arXiv:1903.09122*.

[Turakhia et al., 2018] Turakhia, M., Desai, M., Hedlin, H., Rajmane, A., Talati, N., Ferris, T., Desai, S., Nag, D., Patel, M., Kowey, P., Rumsfeld, J., M. Russo, A., True Hills, M., B. Granger, C., W. Mahaffey, K., and Perez, M. (2018). Rationale and design of a large-scale, app-based study to identify cardiac arrhythmias using a smartwatch: The apple heart study. *American Heart Journal*, 207.

[Vinh et al., 2010] Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854.

[Vishwanathan et al., 2007] Vishwanathan, S., Smola, A. J., and Vidal, R. (2007). Binet-cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. *International Journal of Computer Vision*, 73(1):95–119.

[Yeh et al., 2012] Yeh, Y.-C., Chiou, C. W., and Lin, H.-J. (2012). Analyzing ecg for cardiac arrhythmia using cluster analysis. *Expert Systems with Applications*, 39(1):1000–1010.