Learning Linear Dynamical Systems from Multivariate Time Series: A Matrix Factorization Based Framework

Zitao Liu* Milos Hauskrecht*

Abstract

The linear dynamical system (LDS) model is arguably the most commonly used time series model for real-world engineering and financial applications due to its relative simplicity, mathematically predictable behavior, and the fact that exact inference and predictions for the model can be done efficiently. In this work, we propose a new generalized LDS framework, gLDS, for learning LDS models from a collection of multivariate time series (MTS) data based on matrix factorization, which is different from traditional EM learning and spectral learning algorithms. In gLDS, each MTS sequence is factorized as a product of a shared emission matrix and a sequence-specific (hidden) state dynamics, where an individual hidden state sequence is represented with the help of a shared transition matrix. One advantage of our generalized formulation is that various types of constraints can be easily incorporated into the learning process. Furthermore, we propose a novel temporal smoothing regularization approach for learning the LDS model, which stabilizes the model, its learning algorithm and predictions it makes. Experiments on several real-world MTS data show that (1) regular LDS models learned from gLDS are able to achieve better time series predictive performance than other LDS learning algorithms; (2) constraints can be directly integrated into the learning process to achieve special properties such as stability, low-rankness; and (3) the proposed temporal smoothing regularization encourages more stable and accurate predictions.

1 Introduction

Recent advances in data collection, data storage and information technologies have resulted in enormous collections of multivariate time series (MTS) data on various aspects of our everyday life. A variety of MTS datasets appear in economy, meteorology, Internet or vehicular traffic, healthcare and many other areas. These MTS data provide us with a unique opportunity to gain novel insights and understanding of the processes and systems generating the data. Developing and learning accurate models of MTS play important roles in forecasting, decision making and/or intelligent system support.

Various mathematical models have been proposed to study real-valued MTS data and processes [27]. Among all of them, the linear dynamical system (LDS) model stands out and is arguably the most commonly used time series model for real-world applications. Basically, an LDS is a Markovian model that assumes the dynamic behavior of the system is captured well using a small set of real-valued hidden-state variables and linear state transitions corrupted by a Gaussian noise. Due to the model's relative simplicity, mathematically predictable behavior, and the fact that exact inference and predictions for the model can be done efficiently, LDS models have been applied widely for time series prediction [16, 18, 21, 22, 28] or object tracking [10, 17] tasks.

While in some rare LDS application scenarios, the LDS models' parameters are known, e.g., they are derived from known physical equations, the majority of real-world applications require them to be learned from MTS data. Expectation-Maximization (EM) [6] and spectral learning [12, 25] algorithms are typically used for this task. Recently in order to meet different characteristics of MTS data, various constraints, such as sparsity, low-rankness, were integrated into the standard LDS models. Constraints are incorporated either in the hidden state inference process [1, 3, 4] or into the parameter learning process [2, 20].

In this work, we propose a new generalized LDS framework, gLDS, for learning LDS models from a collection of MTS data. Our learning framework is based on matrix factorization approach, where each MTS sequence is factorized as a product of a shared emission matrix and a sequence-specific hidden dynamics. In contrast to traditional matrix factorization, the hidden factors in gLDS may evolve in time and individual dynamics is modeled with the help of a shared transition matrix. We use alternating minimization to learn the LDS model from data. In such a case, each parameter can be optimized efficiently and the procedure is flexible enough to incorporate various constraints. Furthermore, we propose a temporal smoothing regularization,

*Computer Science Department, University of Pittsburgh, Pittsburgh, PA USA. Email: {ztliu, milos}@cs.pitt.edu

which penalizes the difference of predictive results from the learned model during the learning phase, to achieve smooth forecasts from the learned LDS models.

In summary, our paper makes the following contributions:

- Compared to LDS spectral learning algorithm, our framework is able to learn the LDS model from a collection of many different MTS sequences. The spectral learning algorithm requires and learns the model from just one sequence. While it is always possible to concatenate multiple MTS sequences into one large sequence, this brute-force concatenation process may introduce a lot of non-smooth transitions leading to model imprecision.
- Compared to the LDS EM learning algorithm, parameter optimization in gLDS can be done efficiently in each iteration. The EM requires one to infer the hidden states for each training sequence. This step is avoided in gLDS.
- Constraints can be easily incorporated into our generalized framework. We show that various existing approaches, such as stable LDS [2], regularized LDS [20] are special cases of our gLDS framework.
- A novel temporal smoothing regularization and a smooth LDS model are proposed which are able to support more smooth and more accurate time series predictions.

The remainder of the paper is organized as follows: Section 2 introduces the basics of the LDS models, standard LDS learning algorithms and some recent work on constrained LDS. In Section 3, we describe our generalized learning framework and show the connections between various existing research and our framework. We proposed a temporal smoothing regularization for LDS to support more smooth and accurate predictions. In Section 4, we show that (1) models learned from gLDS support better predictions compared with standard LDS learning algorithms; and (2) various constraints can be integrated into gLDS and stability and low-rankness can be easily achieved. We summarize our work and outline potential future extensions in Section 5.

2 Background

In this section, we first introduce the notation we will use throughout this paper. Then, we review the basics of the LDS model and its two major learning algorithms: EM and spectral approaches. After that, we discuss some recent LDS extensions that incorporate various constraints into the basic LDS model in order to achieve its better performance.

2.1 Notation Given a collection of N multivariate time series sequences $\{\mathbf{Y}^1, \mathbf{Y}^2, \cdots, \mathbf{Y}^N\}$, we represent each multivariate sequence $\mathbf{Y}^m$ as an $n \times T_m$ matrix, where $\mathbf{Y}^m = [\mathbf{y}_1^m, \cdots, \mathbf{y}_t^m, \cdots, \mathbf{y}_{T_m}^m]$ and n is the number of time series variables and T_m is the length of sequence. $\mathbf{y}_t^m$ is an $n \times 1$ vector observed at time stamp t. Correspondingly, $\{\mathbf{Z}^1, \mathbf{Z}^2, \cdots, \mathbf{Z}^N\}$ represent the collection of hidden state sequences, where $\mathbf{Z}^m = [\mathbf{z}_1^m, \cdots, \mathbf{z}_t^m, \cdots, \mathbf{z}_{T_m}^m]$ is a $d \times T_m$ matrix. d is the dimensionality of hidden state space and $\mathbf{z}_t^m$ is a $d \times 1$ hidden state vector corresponding to $\mathbf{y}_t^m$. Let $\mathbf{Z}_+^m = [\mathbf{z}_2^m, \mathbf{z}_3^m, \cdots, \mathbf{z}_{T_m}^m]$ and $\mathbf{Z}_-^m = [\mathbf{z}_1^m, \mathbf{z}_2^m, \cdots, \mathbf{z}_{T_m-1}^m]$. We use $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{Z}_+$, and $\mathbf{Z}_-$ to denote the horizontal concatenations of $\{\mathbf{Y}^m\}$, $\{\mathbf{Z}^m\}$, $\{\mathbf{Z}_+^m\}$, and $\{\mathbf{Z}_-^m\}$. Here $\mathbf{Y}$ is an $n \times T$ matrix and $\mathbf{Z}$ is a $d \times T$ matrix. $\mathbf{Z}_+$ and $\mathbf{Z}_-$ are $d \times (T - N)$ matrices where $T = \sum_{m=1}^{N} T_m$.

Let $|\cdot|$, $\|\cdot\|_2$ and $\|\cdot\|_F$ be the absolute value, the vector ℓ_2 and matrix Forbenious norms. We use $Tr[\cdot]$ to denote the matrix trace operator. I_d is the $d \times d$ identity matrix. Let $\mathbb{E}_{\mathbf{z}}[f(\cdot)]$ denote the expected value of $f(\cdot)$ with respect to $\mathbf{z}$. For both vectors and matrices, the superscript $(\cdot)^\top$ denotes the transpose. For the sake of notational brevity, we omit the explicit sample index (m) in the rest of Section 2.

2.2 Linear Dynamical System The LDS is an MTS model that represents observation sequences indirectly with the help of hidden states. The LDS models the dynamics of these sequences in terms of the state transition probability $p(\mathbf{z}_t|\mathbf{z}_{t-1})$, and state-observation probability $p(\mathbf{y}_t|\mathbf{z}_t)$. These probabilities are modeled using the following equations:

$$\mathbf{z}_t = A\mathbf{z}_{t-1} + \boldsymbol{\epsilon}_t \tag{2.1}$$

$$\mathbf{y}_t = C\mathbf{z}_t + \boldsymbol{\zeta}_t \tag{2.2}$$

where the transitions among the current and previous hidden states are linear and captured in terms of a $d \times d$ transition matrix A. The stochastic component of the transition, ϵ_t, is modeled by a zero-mean Gaussian noise $\epsilon_t \sim \mathcal{N}(\mathbf{0}, Q)$ with a $d \times 1$ zero mean vector and a $d \times d$ covariance matrix Q. The observational sequence is derived from the hidden states sequence. The dependencies in between the two sequences are linear and modeled using an $n \times d$ emission matrix C. A zero mean Gaussian noise $\boldsymbol{\zeta}_t \sim \mathcal{N}(\mathbf{0}, R)$ models the stochastic relation in between the states and observations. In addition to A, C, Q, R, the LDS is defined by the initial state distribution for $\mathbf{z}_1$ with mean $\boldsymbol{\xi}$ and covariance matrix Ψ, i.e., $\mathbf{z}_1 \sim \mathcal{N}(\boldsymbol{\xi}, \Psi)$. The complete set of the LDS parameters is $\Omega = \{A, C, Q, R, \boldsymbol{\xi}, \Psi\}$.

While in some LDS applications the model parameters are known a priori, in the majority of real-world

applications the model parameters are unknown, and we need to learn them from MTS data. Various learning algorithms have been proposed in the past years. In the following, we briefly discuss the two major categories of these learning algorithms: Expectation-Maximization (EM) [6, 23] (See Section 2.2.1) and spectral learning algorithms [2, 5, 12, 25] (See Section 2.2.2).

2.2.1 EM Learning The EM algorithm is an iterative procedure for finding model parameters that maximizes the likelihood of observations in the presence of hidden variables. In practice, instead of maximizing the data likelihood directly, EM algorithm usually maximizes a $\mathcal{Q}$ function, which is the expectation of the joint probability of both observed and hidden variables with respect to the distribution of hidden variables. The $\mathcal{Q}$ function is a lower bound of the true data likelihood and maximizing it will improve the data likelihood. Under the setting of learning standard LDS defined by eq.(2.1) and eq.(2.2), the $\mathcal{Q}$ function is defined as follows:

$$\begin{aligned}\mathcal{Q} &= \mathbb{E}_{\mathbf{Z}}\Big[\log p(\mathbf{Z},\mathbf{Y})\Big] = \mathbb{E}_{\mathbf{Z}}\Big[\log p(\mathbf{z}_1)\Big] \\ &+ \mathbb{E}_{\mathbf{Z}}\Big[\sum_{t=1}^{T}\log p(\mathbf{y}_t|\mathbf{z}_t)\Big] + \mathbb{E}_{\mathbf{Z}}\Big[\sum_{t=2}^{T}\log p(\mathbf{z}_t|\mathbf{z}_{t-1})\Big]\end{aligned} \tag{2.3}$$

The EM algorithm alternates between maximizing the $\mathcal{Q}$ function with respect to the parameters Ω and with respect to the distribution of hidden states, holding the other quantities fixed. The E-step depends on $\mathbb{E}[\mathbf{z}_t|\mathbf{Y}]$, $\mathbb{E}[\mathbf{z}_t\mathbf{z}_t^\top|\mathbf{Y}]$ and $\mathbb{E}[\mathbf{z}_t\mathbf{z}_{t-1}^\top|\mathbf{Y}]$, which are sufficient statistics to compute eq.(2.3). Kalman filtering [11] and Kalman smoothing [26] algorithms are used for computing the sufficient statistics, which are provided in the supplemental material[1]. The M-step re-estimate each of the parameter in Ω by taking the corresponding partial derivative of the expected log likelihood, setting to zero and solving. Update rules in M-step can be found in [6].

While the EM learning algorithm is well studied for LDS models, it is difficult to incorporate constraints on either hidden states estimation or parameter estimation during the process of learning LDS models from the collection of data. Furthermore, when the number of sequential instances increases, the EM algorithm becomes inefficient since it computes the expectation for each sequence.

2.2.2 Spectral Learning Spectral learning methods provide a non-iterative, asymptotically unbiased LDS estimation solution in closed form. They estimate the parameters of an LDS by using singular value decomposition (SVD) to find Kalman filter estimates of the underlying state sequence [5, 12, 25]. Spectral learning methods approximate the observation matrix $\mathbf{Y}$ or its variants (hankel matrix) [2] into $U\Sigma V^\top$ by SVD, where $U \in \mathbb{R}^{n\times d}$ and $V \in \mathbb{R}^{T\times d}$ have orthonormal columns $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ and $\Sigma = \text{diag}\{\delta_1, \cdots, \delta_d\}$ contains the singular values. The emission matrix and state sequence are estimated as $\hat{C} = U$ and $\hat{\mathbf{Z}} = \Sigma V^\top$ and the transition matrix $\hat{A}$ is obtained by solving the least square of $\|A\mathbf{Z}_- - \mathbf{Z}_+\|_F^2$. Based on this SVD intuition, various algorithms have been proposed, such as N4SID [24].

The advantage of spectral learning algorithms is the efficiency due to the non-iterative nature, while the major limitations are (1) it requires the sequence to be long enough in order to maintain its learning consistency; and (2) it has to conduct the SVD over each individual time series sequence and cannot be directly learned from multiple sequences.

2.3 Related Work Recently, in order to improve the model performance and its quality, various researchers have proposed to incorporate different constraints into both the inference process (estimating hidden states $\mathbf{Z}$ from data given known parameters Ω) (See Section 2.3.1) and the learning process (estimating parameters Ω from data) (See Section 2.3.2).

2.3.1 C1: Constraints on LDS Inference Most of the existing refinements focus on enforcing different types of sparsity constraints on the estimates of the hidden state ($\mathbf{Z}$). For example, in [1, 3, 4] the hidden states are sparsified during the Kalman filter inference step. [4] formulates the traditional Kalman filter as a one-step update optimization procedure and incorporates sparsity constraints to achieve a sparse state estimate at each time stamp t. [1] treats all the state estimates as a state estimate matrix and enforces a row-level group lasso on the state estimate matrix.

All the methods in this category try to learn a sparse representation of the hidden-state estimation problem by assuming that all parameters of the LDS are known a priori. Hence they are not directly applicable to the problem of learning LDS models from MTS data.

2.3.2 C2: Constraints on LDS Learning Constraints in this category are incorporated into the LDS learning process in order to achieve special model properties, such as low-rankness [20], stability [2], and others. In the following, we describe two recent constrained LDS models: *regularized LDS* and *stable LDS*.

[1]The supplemental material can be found at `http://www.zitaoliu.com/download/sdm2016_sup.pdf`.

- **Regularized LDS** Regularized LDS models proposed by [20] aim to learn LDS models with a low-rank transition matrix from a limited number of MTS sequences. Compared with the ordinary LDS model, the regularized LDS is able to find the intrinsic dimensionality of the hidden state and prevent the overfitting problem whenever the amount of MTS data is small.
- **Stable LDS** An LDS with dynamic matrix A is stable if all of A's eigenvalues have magnitude at most 1. The stability is crucial when simulating long sequences from LDS models in order to generate representative data or infer stretches of missing values. [2] formulates the optimization of the transition matrix as a quadratic program and keeps generating linear constraints to bound the eigenvalues and hence, is able to guarantee stability.

3 The Generalized LDS Learning Framework

In this section, we propose a generalized framework, gLDS, for learning LDS models based on matrix factorization. In gLDS, (1) the LDS models can be learned from multiple MTS sequences; and (2) various constraints can be easily incorporated into the learning process. More specifically, in Section 3.1, we present the learning of gLDS and the closed form solutions for each parameter. In Section 3.2, we describe the learning process for a ridge regularized LDS model (gLDS-ridge) as one example of the gLDS framework. In Section 3.3, we show the generalization ability of gLDS by illustrating its connections to two popular models: stable LDS and the regularized LDS. In Section 3.4, we propose a novel temporal smooth regularization and show that how to incorporate it into the LDS learning procedure.

3.1 gLDS Framework Based on the linear assumption in LDS that sequential observation vector is generated by the linear emission transformation C from hidden states at each time stamp (eq.(2.2)), we can formulate the LDS learning problem by using the matrix factorization approach [13, 15] that assumes the collection of MTS sequences are generated by a shared emission matrix and their specific hidden factors.

$$\min_{C,\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 \tag{3.4}$$

However, different from traditional matrix factorization models where the hidden factors are in general time independent, in LDS models, hidden factors evolve with time and are specified by eq.(2.1). Hence, similar to [2, 5], we estimate the transition matrix A by solving another least square problem as follows:

$$\min_{A,\mathbf{Z}} \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 \tag{3.5}$$

In this work, in order to learn the LDS parameters from the data, we jointly optimize both eq.(3.4) and eq.(3.5). Furthermore, in order to incorporate constraints into the learned LDS models, we add regularizations for C, A and $\mathbf{Z}$ into the objective function, shown as follows:

$$\begin{aligned}\min_{A,C,\mathbf{Z}} &\|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha\mathcal{R}_C(C) \\ &+ \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{Z}) + \gamma\mathcal{R}_A(A)\end{aligned} \tag{3.6}$$

Intuitively, this formulation of the problem aims to find an LDS model that is able to fit as accurately as possible the time series in the training data by using a simple (less complex) model.

3.1.1 Learning As we can see from eq.(3.6), the coupling between A, C and $\mathbf{Z}$ makes this problem difficult to find optimal solutions for A, C and $\mathbf{Z}$ simultaneously, so in this work, we adopt the alternating optimization scheme to find the solution iteratively.

Optimization of A, C, and $\mathbf{Z}$ We apply the alternating minimization techniques to eq.(3.6), which leads to the following three optimization problems:

$$\min_{A} \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \gamma/\lambda\mathcal{R}_A(A) \tag{3.7}$$

$$\min_{C} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \alpha\mathcal{R}_C(C) \tag{3.8}$$

$$\min_{\mathbf{Z}} \|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{Z}) \tag{3.9}$$

Since optimization of a hidden state sequence $\mathbf{Z}^m$ is independent of other sequences, we can further decompose the optimization target $\mathbf{Z}$ into $\{\mathbf{Z}^1, \cdots, \mathbf{Z}^m \cdots \mathbf{Z}^N\}$. Due to the asymmetric positions of different $\mathbf{z}_t^m$s in $\mathbf{Z}^m$, we decompose the optimization into three parts: $\mathbf{z}_1^m$, $\mathbf{z}_t^m$ and $\mathbf{z}_{T_m}^m$ $(t = 2, \cdots, T_m - 1)$. The optimization problems for each hidden states sequence $\mathbf{Z}^m$ are defined as follows:

$$\min_{\mathbf{z}_1^m} \|\mathbf{y}_1^m - C\mathbf{z}_1^m\|_2^2 + \lambda\|\mathbf{z}_2^m - A\mathbf{z}_1^m\|_2^2 + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{z}_1^m) \tag{3.10}$$

$$\begin{aligned}\min_{\mathbf{z}_t^m} &\|\mathbf{y}_t^m - C\mathbf{z}_t^m\|_2^2 + \lambda\|\mathbf{z}_t^m - A\mathbf{z}_{t-1}^m\|_2^2 \\ &+ \lambda\|\mathbf{z}_{t+1}^m - A\mathbf{z}_t^m\|_2^2 + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{z}_t^m)\end{aligned} \tag{3.11}$$

$$\min_{\mathbf{z}_{T_m}^m} \|\mathbf{y}_{T_m}^m - C\mathbf{z}_{T_m}^m\|_2^2 + \lambda\|\mathbf{z}_{T_m}^m - A\mathbf{z}_{T_m-1}^m\|_2^2 + \beta\mathcal{R}_{\mathbf{Z}}(\mathbf{z}_{T_m}^m) \tag{3.12}$$

Optimization of R, Q, $\boldsymbol{\xi}$ and Ψ. Once we obtain A, C and $\mathbf{Z}$, the rest of LDS's parameters, R, Q, $\boldsymbol{\xi}$ and Ψ, can be analytically estimated as follows:

$$\hat{Q} = \frac{1}{T-N}(\hat{\mathbf{Z}}_+ - \hat{A}\hat{\mathbf{Z}}_-)(\hat{\mathbf{Z}}_+ - \hat{A}\hat{\mathbf{Z}}_-)^\top \tag{3.13}$$

$$\hat{R} = \frac{1}{T}(\mathbf{Y} - \hat{C}\hat{\mathbf{Z}})(\mathbf{Y} - \hat{C}\hat{\mathbf{Z}})^\top \tag{3.14}$$

$$\hat{\boldsymbol{\xi}} = \frac{1}{N}\sum_{m=1}^{N} \hat{\mathbf{z}}_1^m \tag{3.15}$$

$$\hat{\Psi} = \frac{1}{N}\sum_{m=1}^{N} \hat{\mathbf{z}}_1^m (\hat{\mathbf{z}}_1^m)^\top \tag{3.16}$$

3.1.2 Summary The entire LDS parameter estimation procedure in our gLDS framework is summarized by Algorithm 1.

3.2 The Ridge Model (gLDS-ridge) Ridge regularization, a.k.a, Tikhonov regularization, ℓ_2 regularization [9], is widely used to prevent overfitting since it encourages the sum of the squares of the fitted parameters to be small. Furthermore, it alleviates the ill-posed problems in numerical methods. In our gLDS framework, we achieve the ridge model (gLDS-ridge) by setting $\mathcal{R}_C(C)$, $\mathcal{R}_A(A)$, and $\mathcal{R}_{\mathbf{Z}}(\mathbf{Z})$ to the square of Frobenius norm, i.e., $\|\cdot\|_F^2$.

Due to the differentiability of ridge regularization, we can take the partial derivatives of eqs.(3.7 - 3.12), set them to zero and solve. The results are shown as follows:

$$\hat{A} = (\mathbf{Z}_+\mathbf{Z}_-^\top)(\mathbf{Z}_-\mathbf{Z}_-^\top + \gamma/\lambda I_d)^{-1} \tag{3.17}$$

$$\hat{C} = (\mathbf{Y}\mathbf{Z}^\top)(\mathbf{Z}\mathbf{Z}^\top + \alpha I_d)^{-1} \tag{3.18}$$

$$\hat{\mathbf{z}}_1^m = (G + \lambda A^\top A)^{-1}(C^\top \mathbf{y}_1^m + \lambda A^\top \mathbf{z}_2^m) \tag{3.19}$$

$$\hat{\mathbf{z}}_t^m = (G + \lambda A^\top A + \lambda I_d)^{-1}(F_t^m + \lambda A^\top \mathbf{z}_{t+1}^m) \tag{3.20}$$

$$\hat{\mathbf{z}}_{T_m}^m = (G + \lambda I_d)^{-1} F_{T_m}^m \tag{3.21}$$

where $G = C^\top C + \beta I_d$ and $F_t^m = C^\top \mathbf{y}_t^m + \lambda A \mathbf{z}_{t-1}^m$.

3.3 Existing Models in gLDS Framework Constraints from both C1 (Section 2.3.1) and C2 (Section 2.3.2) can be easily incorporated into our gLDS framework due to its flexibility and extensibility. Since in this work we focus on learning LDS models from MTS sequences, in the following, we describe how to formulate both the stable LDS and regularized LDS as special instances in our gLDS framework.

3.3.1 Learning Regularized LDS (gLDS-low-rank) In order to obtain a low-rank transition matrix of the LDS model, [20] develops a Maximum a Posteriori learning framework and apply both multivariate Laplacian prior and nuclear norm prior on the A to implicitly shut down spurious and unnecessary dimensions and prevent overfitting problem simultaneously. In gLDS framework, a low-rank transition matrix A can be easily obtained by setting $\mathcal{R}_A(A) = \|A\|_F^2 + \frac{\lambda}{\gamma}\gamma_A\|A\|_*$ in eq.(3.17), which leads to the following objective function (eq.(3.22)). All the others' updates (eqs.(3.18 - 3.16) remain the same.

$$\min_A g(A) + \gamma_A\|A\|_* \tag{3.22}$$

where

$$g(A) = \|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \gamma/\lambda\|A\|_F^2 \tag{3.23}$$

Since $g(A)$ is convex and differentiable with respect to A, we can adopt the generalized gradient descent algorithm to minimize eq.(3.22). The update rule is

$$A^{(k+1)} = \text{prox}_{\rho_k}\Big(A^{(k)} - \rho_k \bigtriangledown g(A^{(k)})\Big) \tag{3.24}$$

where ρ_k is the step size at iteration k and the proximal function $\text{prox}_{\rho_k}(A)$ is defined as the singular value soft-thresholding operator,

$$\text{prox}_{\gamma_A\rho_k}(A) = U \cdot \text{diag}((\sigma_i - \gamma_A\rho_k)_+) \cdot V' \tag{3.25}$$

where $A = U\text{diag}(\sigma_1, \cdots, \sigma_d)V'$ is the singular value decomposition (SVD) of A.

An important open question here is how to set the step size of the generalized gradient method to assure it is well behaved. Theorem 3.1 gives us a simple way to select the step size while also assuring its fast convergence rate.

Algorithm 1 Learn the LDS model in gLDS.

INPUT:
- Initialization $A^{(0)}, C^{(0)}, \mathbf{Z}^{(0)}$.
- Hyper-parameters, γ, λ, β and α.
- A collection of MTS sequences $\mathbf{Y}^1, \cdots, \mathbf{Y}^N$.

PROCEDURE:

```
// Optimize A, C and Z
repeat
    Update A by solving eq.(3.7).
    Update C by solving eq.(3.8).
    for m: 1 → N do
        Update z_1^m by solving eq.(3.10).
        for t: 2 → T_m − 1 do
            Update z_t^m by solving eq.(3.11).
        end for
        Update z_{T_m}^m by solving eq.(3.12).
    end for
until Convergence
// Optimize Q̂, R̂, ξ̂, Ψ̂
Compute Q̂, R̂, ξ̂, Ψ̂ using eqs.(3.13 - 3.16).
```

OUTPUT:
- Learned LDS parameters: $\hat{\Omega} = \{\hat{A}, \hat{C}, \hat{Q}, \hat{R}, \hat{\boldsymbol{\xi}}, \hat{\Psi}\}$.

THEOREM 3.1. *Generalized gradient descent with a fixed step size* $\rho \leq 1/2(\|\mathbf{Z}_-\mathbf{Z}_-^\top\|_F + \gamma/\lambda)$ *for minimizing eq.(3.22) has convergence rate* $O(1/k)$*, where* k *is the number of iterations.*

Proof. The proof of this theorem appears in the supplemental material.

3.3.2 Learning Stable LDS (gLDS-stable) Stability is a desired property for dynamical modeling and it play important roles in tasks such as predictions, long term sequence simulation, etc. [2] proposes a novel method for learning stable LDS models by formulating the problem as a quadratic program. The program starts with a relaxed solution and incrementally add constraints to improve stability. In gLDS framework, by setting $\mathcal{R}_A(A) = \emptyset$, we can easily transform our optimization to the same objective function in [2]. Furthermore, we can apply the following theorem to change eq.(3.7) into the standard quadratic program formulation.

THEOREM 3.2. *Minimizing* A *from eq.(3.7) with* $\mathcal{R}_A(A) = \emptyset$ *is equivalent to minimizing the following problem:*

$$\min_a a^\top B a - 2q^\top a \tag{3.26}$$

where $a = \mathrm{vec}(A^\top)$, $B = I_d \otimes (\mathbf{Z}_-\mathbf{Z}_-^\top)$, $q = (I_d \otimes \mathbf{Z}_-\mathbf{Z}_+^\top)\,\mathrm{vec}(I_d)$.

Proof. The proof of this theorem appears in the supplemental material.

After the quadratic program transformation, we can apply the same constraints generation techniques described in [2] to optimize the transition matrix and guarantee its stability. Details can be found in [2].

3.4 The Smooth Model (gLDS-smooth) In this section, we propose a novel temporal smoothing regularization (Section 3.4.1), which penalizes the difference of predictive results from the learned model during the learning phase, to achieve smooth forecasts from the learned LDS models. In Section 3.4.2, we show how to incorporate the temporal smoothing regularization into gLDS and describe the corresponding learning procedure.

3.4.1 Temporal Smoothing Regularization To incorporate predictive smoothness property in LDS models for MTS modeling and forecasting, we propose a temporal smoothing regularization term $\mathcal{R}_\mathcal{T}^m$ for each MTS sequence m:

$$\mathcal{R}_\mathcal{T}^m = \frac{1}{2}\sum_{i=1}^{T_m}\sum_{j=1}^{T_m} w_{ij}^m \|\hat{\mathbf{y}}_i^m - \hat{\mathbf{y}}_j^m\|_2^2 \tag{3.27}$$

where $\hat{\mathbf{y}}_t^m$ is the model forecast at time stamp t and w_{ij}^m is the smoothing coefficient balancing the difference between predictions $\hat{\mathbf{y}}_i^m$ and $\hat{\mathbf{y}}_j^m$.

Briefly, the regularization term penalizes the predictions for each time stamp that disagree with other predictions made within the same MTS sequence. The amount of penalty is controlled by a smoothing coefficient w_{ij}^m that is higher for time stamps close to each other and smaller for time stamps further apart. After some algebraic manipulations, the regularization term $\mathcal{R}_m$ can be rewritten as,

$$\begin{aligned}\mathcal{R}_\mathcal{T}^m =& \frac{1}{2}\sum_{i=1}^{T_m}\sum_{j=1}^{T_m} w_{i,j}^m \|\hat{\mathbf{y}}_i^m - \hat{\mathbf{y}}_j^m\|_2^2 \\ =& \sum_{i=1}^{T_m}\sum_{j=1}^{T_m}\sum_{l=1}^{n} w_{i,j}^m (\hat{y}_{l,i}^m)^2 - w_{i,j}^m \hat{y}_{l,i}^m \hat{y}_{l,j}^m \\ =& \sum_{l=1}^{n} \hat{\mathbf{Y}}^m(l,:)(D^m - W^m)\hat{\mathbf{Y}}^m(l,:)^\top \\ =& Tr[C\mathbf{Z}^m L^m (\mathbf{Z}^m)^\top C^\top]\end{aligned} \tag{3.28}$$

where $\hat{\mathbf{Y}}^m(l,:)$ represents the lth row of matrix $\hat{\mathbf{Y}}^m$. L^m is the $T_m \times T_m$ Laplacian matrix for the mth MTS sequence, $L^m = D^m - W^m$. D^m is a diagonal matrix with the ith diagonal element $D_{i,i}^m = \sum_{j=1}^{T_m} w_{i,j}^m$. W^m is the smoothing coefficient matrix among T_m observations and $w_{i,j}^m$ represents the (i,j)th element in W^m.

In order to learn a smooth LDS model, we apply the temporal smooth regularization to each MTS sequence in the training data, which leads to the following compact form of regularization:

$$\mathcal{R}_\mathcal{T} = \sum_{m=1}^{N} \mathcal{R}_\mathcal{T}^m = Tr[C\mathbf{Z}P\mathbf{Z}^\top C^\top] \tag{3.29}$$

where P is the $T \times T$ block diagonal matrix with N blocks and the mth block component is the Laplacian matrix L^m for mth MTS sequence.

3.4.2 Learning Smooth LDS We incorporate the temporal smooth regularization (eq.(3.29)) into the objective function (eq.(3.6)). Here similar to gLDS-ridge, we set $\mathcal{R}_C(C)$, $\mathcal{R}_A(A)$, and $\mathcal{R}_\mathbf{Z}(\mathbf{Z})$ to the ridge regularizations (square of Frobenius norm), which leads to the following new learning objective function:

$$\begin{aligned}\min_{A,C,\mathbf{Z}} &\|\mathbf{Y} - C\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}_+ - A\mathbf{Z}_-\|_F^2 + \alpha\|C\|_F^2 \\ &+ \beta\|\mathbf{Z}\|_F^2 + \gamma\|A\|_F^2 + \delta Tr[C\mathbf{Z}P\mathbf{Z}^\top C^\top]\end{aligned} \tag{3.30}$$

Similar to the gLDS-ridge model learning algorithm (Algorithm 1), we optimize eq.(3.30) in a coordinate descent fashion. Since the temporal smoothing regularization only involves C and $\mathbf{Z}$, the update rules for A, R,

Q, $\boldsymbol{\xi}$ and $\boldsymbol{\Psi}$ remain the same. The update rules for C and $\mathbf{Z}$ are shown as follows:

(3.31)
$$\hat{C} = (\mathbf{Y}\mathbf{Z}^\top)(\mathbf{Z}\mathbf{Z}^\top + \delta\mathbf{Z}P\mathbf{Z}^\top + \alpha I_d)^{-1}$$

(3.32)
$$\hat{\mathbf{z}}_1^m = \left(\Gamma_1^m + \lambda A^\top A\right)^{-1}\left(\Phi_1^m + \lambda A^\top \mathbf{z}_2^m\right)$$

(3.33)
$$\hat{\mathbf{z}}_t^m = \left(\Gamma_t^m + \lambda A^\top A + \lambda I_d\right)^{-1}\left(\Phi_t^m + \lambda A^\top \mathbf{z}_{t+1}^m + \lambda A\mathbf{z}_{t-1}^m\right)$$

(3.34)
$$\hat{\mathbf{z}}_{T_m}^m = \left(\Gamma_{T_m}^m + \lambda I_d\right)^{-1}\left(\Phi_{T_m}^m + \lambda A\mathbf{z}_{T_m-1}^m\right)$$

where

(3.35)
$$\Gamma_t^m = (1 + \delta L_{t,t} - \delta W_{t,t})C^\top C + \beta I_d$$

(3.36)
$$\Phi_t^m = C^\top \mathbf{y}_t^m + \delta C^\top C \sum_{j\neq t} W_{t,j}\mathbf{z}_j^m$$

The entire learning procedure of our smooth LDS model is similar to Algorithm 1 by replacing eq.(3.18 - 3.21) with eqs.(3.31 - 3.34).

4 Experiments

In this experiments, we (1) qualitatively show the prediction results from our gLDS-smooth model; (2) quantitatively show that the superior predictive performance of models from our framework (gLDS-ridge and gLDS-smooth) compared with traditional LDS learning algorithms (EM and spectral algorithms); (3) stability and sparsification effects achieved by our framework. Experiments are conducted on four real-world datasets across different domains. The hyper parameters (α, β, λ and γ) used in our methods are selected (in all experiments) by the internal cross validation approach while optimizing models' predictive performances. In the following experiments, we smooth only the pairs of two consecutive forecasts, which leads to the following smoothing coefficient matrix W^m for each sequence m: $w_{ij}^m = 1$ if $|i-j| = 1$; otherwise, $w_{ij}^m = 0$.

4.1 Datasets

- Flour price data (*flourprice*). It is a monthly flour price indices data from [27], which contains the flour price series in Buffalo, Minneapolis and Kansas City, from August 1972 to November 1980.
- Evap data (*evap*). The evaporation data contains the daily amounts of water evaporated, temperature, and barometric pressure from 10/11/1692 to 09/11/1693 [7] [2].
- H2O evap data (*h2o_evap*). It contains six MTS variables: the amount of evaporation, total global radiation, estimated net radiation, saturation deficit at max temperature, mean daily wind speed and saturation deficit at mean temperature [14] [3].
- Clinical data (*clinical*). We test our gLDS on a MTS clinical data obtained from electronic health records of post-surgical cardiac patients in PCP database [8, 19]. We take 500 patients from the database who had their *Complete Blood Count* (CBC) tests [4] done during their hospitalization. The MTS data consists of 6 individual CBC lab time series: mean corpuscular hemoglobin concentration, mean corpuscular hemoglobin, mean corpuscular volume, mean platelet volume, red blood cell and red cell distribution width.

In order to get a comprehensive evaluations of the proposed methods, in the following experiments, we vary both the training sizes and the number of hidden states. For *flourprice*, *evap* and *h2o_evap* datasets, we conduct the experiments on both 80% and 90% data for training and use both 5 and 10 as the hidden state size. For *clinical* data, we have randomly selected 100 patients out of 500 as a test set and used the remaining 400 patients for training the models and vary the hidden states from 10 to 30 with a step increase of 5.

4.2 Evaluation Metric We use the average mean absolute percentage error (Avg-MAPE) to measure how accurate the predictions are made by the LDS models in MTS forecasting. The Avg-MAPE is defined as follows:

$$\text{Avg-MAPE} = \frac{1}{nTN}\sum_{m=1}^{N}\sum_{l=1}^{n}\sum_{t=1}^{T_m}|1 - \hat{y}_{l,t}^m / y_{l,t}^m| \times 100\%$$

Usually in MTS modeling, the different individual time series are in different scales and simply averaging the error values themselves is not appropriate. Averaging MAPE measures the prediction deviation proportion in terms of the true values, which is more sensible than computing the average of root mean square errors(RMSE), mean square errors(MSE) or mean absolute errors(MAE) of each time series' forecasts.

4.3 Qualitative Prediction Analysis We qualitatively show the prediction effectiveness of our gLDS-smooth model. Figure 1 shows the predictions results for the flour price series in Buffalo. (Due to space limit,

[2] http://www.stat.ufl.edu/~winner/data/evap.dat

[3] http://www.stat.ufl.edu/~winner/data/h2o_evap.dat

[4] CBC panel is used as a broad screening test to check for such disorders as anemia, infection, and other diseases.

more results can be found in the supplemental material). 80% of the MTS is used for training and 20% is used for testing. As we can see from Figure 1, the gLDS-smooth is able to well capture the ups and downs of the time series and makes the accurate predictions.

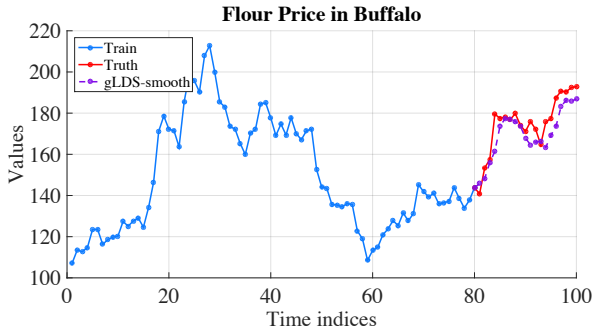

Figure 1: Predictions for flour price series in Buffalo.

4.4 Quantitative Prediction Analysis In this section, we quantitatively compute and compare the prediction accuracy of the proposed methods (gLDS-ridge and gLDS-smooth) with the standard LDS learning approaches: EM (Section 2.2.1) and spectral algorithms (Section 2.2.2). The results are shown in Table 1 and Table 2. Due to the space limit, prediction results results of *flourprice* and *h20_evap* datasets are shown in the supplemental material. As we can see, methods based our framework (gLDS-ridge and gLDS-smooth) perform significantly better than all the other methods. Furthermore, due to the smooth effect of the temporal smooth regularization, gLDS-smooth supports better predictions than gLDS-ridge, which gives the best predictive performance.

Table 1: Average-MAPE results on *clinical* dataset.

# of states	10	15	20	25	30
Spectral	6.29	6.24	6.32	6.04	6.00
EM	3.97	3.54	3.54	3.53	3.53
gLDS-ridge	3.22	3.21	3.21	3.21	3.21
gLDS-smooth	**3.21**	**3.20**	**3.20**	**3.19**	**3.19**

Table 2: Average-MAPE results on *evap* dataset.

	Training: 80%		**Training: 90%**	
# of states	5	10	5	10
Spectral	24.62	24.85	25.08	26.28
EM	17.68	14.45	16.32	17.35
gLDS-ridge	10.58	10.35	13.60	14.05
gLDS-smooth	**10.35**	**10.27**	**13.39**	**13.68**

4.5 Stability Effects of gLDS-stable In this section, similar to [2], we show the stability effects of the gLDS-stable model learned using our framework by generating the simulated sequences in the future. The results of *evap* are shown in Figure 2. Due to the space limit, results of *flourprice*, *h20_evap* and *clinical* datasets are shown in the supplemental material. We can found that in Figure 2, the LDS models learned from EM and spectral algorithms fail to guarantee the system stability and the generated values go to infinities in the future.

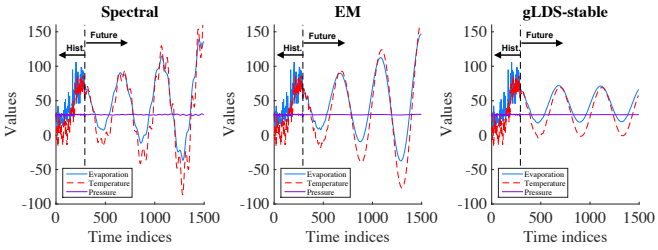

Figure 2: Training data and simulated sequences from gLDS-stable model in *evap* data. "Hist." represents the historical observations and "Future" represents the sequence generated by LDS.

4.6 Sparsification Effects of gLDS-low-rank In this section, we show the sparsification effects of the gLDS-low-rank model learned using our framework. The gLDS-low-rank model is able to identify the intrinsic dimensionality of the hidden state space. The results are shown in Figure 3. As we can see, similar to the experimental results in [20], gLDS-low-rank model is able to find the intrinsic dimension of the hidden state space. Due to the space limit, results of other datasets are shown in the supplemental material.

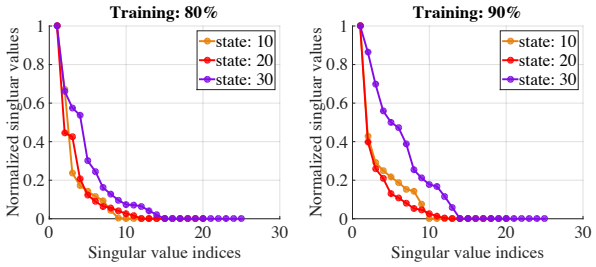

Figure 3: Intrinsic dimensionality recovery of the hidden state space in *evap* dataset.

5 Conclusion

In this paper, we presented a generalized LDS framework for learning LDS models from a collection of MTS sequences. Compared to the traditional LDS learning algorithms, the advantages of our gLDS framework are: (1) the LDS models can be learned efficiently from multiple MTS sequences; (2) constraints on both the hidden states and the parameters can be easily incorporated

into the learning process; (3) it is able to support accurate MTS prediction. Furthermore, we propose a novel temporal smoothing regularization for learning the LDS models, which stabilizes the model, its learning algorithm and predictions it makes. Experimental results on several real-world datasets demonstrated that (1) gLDS are able to achieve better time series predictive performance when compared to other LDS learning algorithms; (2) the proposed temporal smoothing regularization encourages more stable and accurate predictions; and (3) constraints can be directly integrated in the learning process and special designed system properties such as stability, low-rankness can be easily achieved.

Acknowledgment

The work in this paper was supported by grant R01GM088224 from the NIH and by the Predoctoral Andrew Mellon Fellowship awarded to Zitao Liu for the school year 2015-2016. Its content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

[1] D. Angelosante, S. Roumeliotis, and G. Giannakis, *Lasso-kalman smoother for tracking sparse signals*, in Asilomar Conference on Signals, Systems and Computers, 2009, pp. 181–185.

[2] B. Boots, G. Gordon, and S. Siddiqi, *A constraint generation approach to learning stable linear dynamical systems*, in NIPS, 2007, pp. 1329–1336.

[3] A. Carmi, P. Gurfil, and D. Kanevsky, *Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms*, IEEE Transactions on Signal Processing, 58 (2010), pp. 2405–2409.

[4] A. Charles, M. S. Asif, J. Romberg, and C. Rozell, *Sparsity penalties in dynamical system estimation*, in The 45th Annual Conference on Information Sciences and Systems, IEEE, 2011, pp. 1–6.

[5] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto, *Dynamic textures*, IJCV, 51 (2003), pp. 91–109.

[6] Z. Ghahramani and G. Hinton, *Parameter estimation for linear dynamical systems*, tech. report, CRG-TR-96-2, University of Totronto, 1996.

[7] E. Halley, *An account of the evaporation of water, as it was experimented in gresham colledge in the year 1693. with some observations thereon. by edm. halley*, Philosophical Transactions, 18, pp. 183–190.

[8] M. Hauskrecht, M. Valko, I. Batal, G. Clermont, S. Visweswaran, and G. Cooper, *Conditional outlier detection for clinical alerting*, in AMIA Annual Symposium, 2010, pp. 286–290.

[9] A. Hoerl, *Application of ridge analysis to regression problems*, Chemical Engineering Progress, 58 (1962), pp. 54–59.

[10] M. Isard and A. Blake, *Condensationconditional density propagation for visual tracking*, IJCV, 29 (1998), pp. 5–28.

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

[12] T. Katayama, *Subspace methods for system identification*, Springer, 2005.

[13] Y. Koren, R. Bell, and C. Volinsky, *Matrix factorization techniques for recommender systems*, Computer, (2009), pp. 30–37.

[14] A. Krishnan and R. S. Kushwaha, *A multiple regression analysis of evaporation during the growing season of vegetation in the arid zone of india*, Agricultural Meteorology, 12 (1973), pp. 297–307.

[15] D. Lee and S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[16] L. Li, J. McCann, N. S. Pollard, and C. Faloutsos, *Dynammo: Mining and summarization of coevolving sequences with missing values*, in KDD, ACM, 2009, pp. 507–516.

[17] P. Li, T. Zhang, and B. Ma, *Unscented kalman filter for visual curve tracking*, Image and vision computing, 22 (2004), pp. 157–164.

[18] Z. Liu and M. Hauskrecht, *Clinical time series prediction with a hierarchical dynamical system*, in Artificial Intelligence in Medicine, 2013, pp. 227–237.

[19] Z. Liu and M. Hauskrecht, *Clinical time series prediction: Toward a hierarchical dynamical system framework*, Artificial intelligence in medicine, 65 (2015), pp. 5–18.

[20] Z. Liu and M. Hauskrecht, *A regularized linear dynamical system framework for multivariate time series analysis*, in AAAI, 2015, pp. 1798–1804.

[21] Z. Liu and M. Hauskrecht, *Learning adaptive forecasting models from irregularly sampled multivariate clinical data*, in AAAI, 2016.

[22] Z. Liu, L. Wu, and M. Hauskrecht, *Modeling clinical time series using gaussian process sequences*, in SDM, 2013, pp. 623–631.

[23] J. Martens, *Learning the linear dynamical system with asos*, in ICML, 2010, pp. 743–750.

[24] P. V. Overschee and B. D. Moor, *N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems*, Automatica, 30 (1994), pp. 75–93.

[25] P. V. Overschee and B. D. Moor, *Subspace identification for linear systems: Theory Implementation Applications*, 2012.

[26] H. Rauch, *Solutions to the linear smoothing problem*, IEEE Transactions on Automatic Control, 8 (1963), pp. 371–372.

[27] G. C. Reinsel, *Elements of multivariate time series analysis*, Springer, 2003.

[28] M. Rogers, L. Li, and S. Russell, *Multilinear dynamical systems for tensor time series*, in NIPS, 2013, pp. 2634–2642.