Learning Stable Linear Dynamical Systems with the Weighted Least Square Method

Wenbing Huang[1], Lele Cao[1], Fuchun Sun[1], Deli Zhao, Huaping Liu[1] and Shanshan Yu
[1] Department of Computer Science and Technology, Tsinghua University, State Key Lab. of Intelligent Technology and Systems, Tsinghua National Lab. for Information Science and Technology (TNList);
[1]{huangwb12@mails, caoll12@mails, fcsun@mail, hpliu@mail}.tsinghua.edu.cn,
zhaodeli@gmail.com, yushanshan_live@126.com

Abstract

Standard subspace algorithms learn Linear Dynamical Systems (LDSs) from time series with the least-square method, where the stability of the system is not naturally guaranteed. In this paper, we propose a novel approach for learning stable systems by enforcing stability directly on the least-square solutions. To this end, we first explore the spectral-radius property of the least-square transition matrix and then determine the key component that incurs the instability of the transition matrix. By multiplying the unstable component with a weight matrix on the right side, we obtain a weighted-least-square transition matrix that is further optimized to minimize the reconstruction error of the state sequence while still maintaining the stable constraint. Comparative experimental evaluations demonstrate that our proposed methods outperform the state-of-the-art methods regarding the reconstruction accuracy and the learning efficiency.

1 Introduction

Recently, Linear Dynamical Systems (LDSs) have been applied widely for time series modeling in various disciplines. They model the spatial appearance of an input sequence by linearly correlating each observation variable with an underlying state, and then discover dynamical patterns by encoding the evolution of the hidden states with an ARMA model [Doretto *et al.*, 2003]. This simple but flexible framework has promoted a variety of explorations on enhancing the modeling ability of LDSs. For instance, the works in [Saisan *et al.*, 2001; Chan and Vasconcelos, 2005; Woolfe and Fitzgibbon, 2006; Vishwanathan *et al.*, 2007] proposed to define kernel or distance metrics to allow comparisons between LDSs; [Ravichandran *et al.*, 2013] extended the idea of bag-of-features to bag-of-systems for video analysis; and [Huang *et al.*, 2016] combined sparse coding with LDS modeling to deliver robust techniques. The LDS-based models have been successfully applied for various video tasks including synthesis [Doretto *et al.*, 2003; Siddiqi *et al.*, 2007], segmentation [Vidal and Ravichandran, 2005; Chan and Vasconcelos, 2009], classification [Mumtaz *et al.*, 2015] and abnormal detection [Huang *et al.*, 2016].

An LDS is regarded to be stable if all eigenvalues of the transition matrix have a magnitude of 1 at most. While standard methods [Shumway and Stoffer, 1982; Van Overschee and De Moor, 1994; Doretto *et al.*, 2003] have been proposed to learn the system parameters of a given LDS, none of them enforced the stability constraint to the dynamics of the LDS. As verified in both [Siddiqi *et al.*, 2007] and our experiments, the transition matrix learned from the finite sequence may be unstable even if the system is stable. Ignoring the stable criterion will be harmful in some specific applications; in sequences simulation, for example, it will cause significant distortion if an unstable LDS is applied to generate the synthesized sequence. In addition, many LDS-based models [Saisan *et al.*, 2001; Ravichandran *et al.*, 2013; Afsari *et al.*, 2012] take the stable constraint as the mathematically-indispensable condition in the algorithmic formulation, but they usually neglect this constraint in real applications.

Standard subspace algorithm [Doretto *et al.*, 2003] learns transition matrices with the least-square approach. An ideal way to enforce stability in subspace methods is combining the least-square objective with the stable constraint to formulate a new optimization problem. However, the added constraint makes the new problem intractable to solve, as the set of stable matrices is proved to be non-convex [Siddiqi *et al.*, 2007]. Several convex approximations of the stable constraint have been proposed in [Lacy and Bernstein, 2002; 2003; Siddiqi *et al.*, 2007]. Particularly, LB-1 solved a semi-definite program (SDP) via bounding the largest singular value of the transition matrix by 1. Since such constraint may be too conservative for guaranteeing stability, a follow-up work by the same authors in [Lacy and Bernstein, 2003] replaced the Lyapunov inequalities in LB-1 with new inequalities that were proved to be equivalent to the stable constraint. As claimed by the authors in [Siddiqi *et al.*, 2007], the equivalent transformation can help LB-2 to obtain a feasible region close to the right one; but it can also cause certain distortion in the objective value. Hence, in [Siddiqi *et al.*, 2007], the authors proposed a Constraint Generation (CG) method, which solved the Quadratic Program (QP) at each step by incrementally adding constraints to improve stability, and finally stopped the iteration until a stable solution is obtained. The CG method was shown to outperform both LB-1 and LB-2 by the experiments in [Siddiqi *et al.*, 2007].

In this paper, we propose to find a stable solution by directly performing adjustment on the least-square solution. As the least-square solution is derived by minimizing the reconstruction error regardless of the stable constraint, we are interested in analyzing its spectral radius and thereafter determining the factors of incurring the instability of this solution. Such insight studies are interesting and crucial, since they can provide us with a direct clue for deriving stable LDSs. Compared to the above mentioned methods, straightly fixing the unstable term of the least-square solution is more flexible and efficient, as will be demonstrated by our experiments. A related work similar to our idea can be dated back to [Maciejowski, 1995] where the authors computed a stable solution by augmenting extended observability matrix with zeros. Nevertheless, in that paper, the spectral-radius property of the least-square solution has not been fully explored; and the resulting algorithm had not taken the reconstruction objective into consideration.

In sum, we attempt to make the following contributions in this paper: i) We perform reduction on the original form of the least-square solution so as to study the spectral-radius property of the transition matrix. The developed upper-bound of the spectral radius enables us to propose two algorithms including Zero-Padding (ZP) and Bound-Normalization (BN) for stabilizing the least-square solution. ii) To further enhance the performance of ZP and BN, we develop a more flexible algorithm that is dubbed as the Weighted-Least-Square (WLS) method. Particularly, WLS right-multiplies the unstable term of the lest-square solution with a weight matrix that is learned by minimizing the reconstruction objective. iii) We compare the performance of proposed models with previous methods including LB-1, LB-2 and CG on various datasets. The WLS methods are demonstrated to be superior to other compared methods in terms of accuracy and efficiency.

The rest of the paper is organized as follows. Section 2 introduces the preliminaries of LDS modeling. The analysis on the spectral-radius of the least-square solution is performed in Section 3; and the formulation of the WLS method is provided in Section 4. Then Section 5 conducts the experiments; and finally Section 6 concludes this paper.

2 Preliminaries

2.1 Linear Dynamical Systems

LDSs [Doretto *et al.*, 2003] encode time series with the model

$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{v}_t, \\ \mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{w}_t + \overline{\mathbf{y}}, \end{cases} \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_\tau] \in \mathbb{R}^{n\times\tau}$ is the state sequence; $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_\tau] \in \mathbb{R}^{m\times\tau}$ is the observed sequence; $\overline{\mathbf{y}} \in \mathbb{R}^m$ computes the mean of $\mathbf{Y}$; $\mathbf{A} \in \mathbb{R}^{n\times n}$ denotes the transition matrix; $\mathbf{C} \in \mathbb{R}^{m\times n}$ is the measurement matrix; $\mathbf{B} \in \mathbb{R}^{n\times n_v} (n_v \le n)$ represents the noise transformation matrix; $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{I}_{n_v})$ and $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{R})$ denote the process and measurement noise components, respectively, with $\mathbf{I}_{n_v}$ being the $n_v \times n_v$ identity matrix and $\mathbf{R} \in \mathbb{R}^{m\times m}$.

Given the observed sequence, the optimal system parameters $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{R}\}$ can be found with the subspace method presented in [Doretto *et al.*, 2003]. This approach first estimates the state sequence by performing PCA on the observations, and then learns the dynamics in the state space via the Least Square (LS) method. We denote the centered observation matrix as $\mathbf{Y}' = [\mathbf{y}_1 - \overline{\mathbf{y}}, \cdots, \mathbf{y}_\tau - \overline{\mathbf{y}}]$. By performing the Singular Value Decomposition (SVD) of $\mathbf{Y}'$ as $\mathbf{Y}' = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$ where $\mathbf{U} \in \mathbb{R}^{m\times n}$, $\mathbf{S} \in \mathbb{R}^{n\times n}$ and $\mathbf{V} \in \mathbb{R}^{n\times\tau}$, the measurement matrix and the hidden states are estimated as $\mathbf{C} = \mathbf{U}$ and $\mathbf{X} = \mathbf{S}\mathbf{V}^{\mathrm{T}}$, respectively.[1] The transition matrix $\mathbf{A}$ is learned to minimize the state reconstruction error

$$J^2(\mathbf{A}) = \|\mathbf{X}_1 - \mathbf{A}\mathbf{X}_0\|_F^2, \tag{2}$$

where $\mathbf{X}_0 = [\mathbf{x}_1, \cdots, \mathbf{x}_{\tau-1}]$, $\mathbf{X}_1 = [\mathbf{x}_2, \cdots, \mathbf{x}_\tau]$ and $\|\cdot\|_F$ denotes the Frobenius norm. The optimal $\mathbf{A}$ is given by

$$\mathbf{A} = \mathbf{S}\mathbf{V}^{\mathrm{T}}\mathbf{D}_1\mathbf{V}(\mathbf{V}^{\mathrm{T}}\mathbf{D}_2\mathbf{V})^{-1}\mathbf{S}^{-1}, \tag{3}$$

where $\mathbf{D}_1 = \begin{pmatrix} 0 & 0 \\ \mathbf{I}_{\tau-1} & 0 \end{pmatrix}$ and $\mathbf{D}_2 = \begin{pmatrix} \mathbf{I}_{\tau-1} & 0 \\ 0 & 0 \end{pmatrix}$. In case that $\mathbf{V}^{\mathrm{T}}\mathbf{D}_2\mathbf{V}$ is not invertible (actually, we will prove that $\mathbf{V}^{\mathrm{T}}\mathbf{D}_2\mathbf{V}$ is always invertible in Section 3), most literatures compute the transition as $\mathbf{A} = \mathbf{X}_1\mathbf{X}_0^{\dagger}$ instead, where $\dagger$ computes the pseudo-inverse. Other parameters of LDSs like $\mathbf{B}$ and $\mathbf{R}$ can be estimated given $\mathbf{A}$ and $\mathbf{C}$. Interested readers can refer to [Doretto *et al.*, 2003] for details. We hereafter denote the LS solution obtained from Eq. (3) as $\mathbf{A}_l$ in order to distinguish it from those derived by other methods.

2.2 Learning Stable LDSs

Let $\{\lambda_i\}_i^n$ denote the eigenvalues of an $n \times n$ matrix M in a decreasing order of magnitudes, and define M's *spectral radius* as $\rho(M) = |\lambda_1|$. An LDS with the transition dynamics $\mathbf{A}$ is regarded to be stable if and only if all $\mathbf{A}$'s eigenvalues have a magnitude no more than 1, *i.e.* $\rho(\mathbf{A}) \le 1$.

To guarantee stability, the LS problem in Eq. (2) is reformulated as

$$\begin{aligned} \min_{\mathbf{A}} \quad & J^2(\mathbf{A}) \\ \text{s.t.} \quad & \rho(\mathbf{A}) \le 1. \end{aligned} \tag{4}$$

However, the feasible region constrained by $\rho(\mathbf{A}) \le 1$ is non-convex [Siddiqi *et al.*, 2007], making Eq. (4) intractable to solve. Several methods including LB-1 [Lacy and Bernstein, 2002], LB-2 [Lacy and Bernstein, 2003] and CG [Siddiqi *et al.*, 2007] solve Eq. (4) by replacing the stable constraint with devised convex approximations. Although CG is shown to be much efficient than both LB-1 and LB-2, it has the disadvantage of being time-consuming to converge to a stable solution if the optimization update gets stuck in the local minima, which will be further demonstrated in our experiments.

3 Spectral Radius Analysis on the LS Solution

In this section, we will further reveal the spectral-radius property of the LS solution and analyze what factors make $\mathbf{A}_l$ unstable. Several valuable conclusions have been derived elaborately including the reduced form of $\mathbf{A}_l$ and the upper-bound of its spectral radius. To the best of our knowledge, the majority of the intriguing results presented in this section have not yet been revealed previously.

[1] One can perform SVD on the multiple time-steps *Hankel* matrix instead of $\mathbf{Y}$' to estimate $\mathbf{C}$ and $\mathbf{X}$, as discussed in [Siddiqi *et al.*, 2007]. Following most LDS literatures, *e.g.* [Doretto *et al.*, 2003], we only apply the single time-step observation matrix in this paper.

3.1 The Reduced Form of $\mathbf{A}_l$

At the first glance, it seems hard to unfold the inverse part $(\mathbf{V}^{\mathrm{T}}\mathbf{D}_2\mathbf{V})^{-1}$ in Eq. (3), thus making the analysis of the spectral radius of $\mathbf{A}_l$ inflexible. Indeed, we are able to reduce the inverse of $\mathbf{V}^{\mathrm{T}}\mathbf{D}_2\mathbf{V}$ with the help of two usually-ignored properties: i) $\mathbf{V}$ is orthogonal, *i.e* $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$; and ii) the summation of the rows of $\mathbf{V}$ is equal to a zero vector. The first property is obvious as $\mathbf{V}$ is the right orthogonal matrix generated by SVD on $\mathbf{Y}'$. The second one holds basically because $\mathbf{V}^{\mathrm{T}} = \mathbf{S}^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{Y}'$ and $\mathbf{Y}'$ is columnwise centered. Despite their simplicity, these properties are the keys to derive the conclusions in this section.

For convenience, we denote the i-th row of $\mathbf{V}$ as $\mathbf{v}_i^{\mathrm{T}}$. The orthogonal and the rowwise centered properties of $\mathbf{V}$ can therefore be formulated as

$$\sum\nolimits_{i=1}^{\tau} \mathbf{v}_i\mathbf{v}_i^{\mathrm{T}} = \mathbf{I}_n \quad \text{and} \quad \sum\nolimits_{i=1}^{\tau} \mathbf{v}_i = \mathbf{0}. \tag{5}$$

We immediately have the following conclusion:

Theorem 1. *The arbitrary row of* $\mathbf{V}$ *has an inner-product strictly less than* 1. *In particular,* $\mathbf{v}_i^{\mathrm{T}}\mathbf{v}_i \le \frac{\tau-1}{\tau} < 1$ *for* $1 \le i \le \tau$.

Proof. Without loss of generality, we just prove that $\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau \le \frac{\tau-1}{\tau} < 1$. Since the summation $\sum_{i=1}^{\tau-1}\mathbf{v}_i\mathbf{v}_i^{\mathrm{T}}$ is symmetry, there exits an orthogonal matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ and a diagonal matrix $\mathbf{\Lambda} \in \mathbb{R}^{n\times n}$ satisfying $\sum_{i=1}^{\tau-1}\mathbf{v}_i\mathbf{v}_i^{\mathrm{T}} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{\mathrm{T}}$. Let $\boldsymbol{\alpha} = \mathbf{P}^{\mathrm{T}}\mathbf{v}_\tau$; and denote $\boldsymbol{\beta}_i = \mathbf{P}^{\mathrm{T}}\mathbf{v}_i$, $1 \le i \le \tau - 1$. The orthogonal constraint in Eq. (5) can be rewritten as $\mathbf{\Lambda} + \boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}} = \mathbf{I}_n$, which implies that

$$\mathbf{\Lambda} = diag([1,\cdots,1,\sum\nolimits_{i=1}^{\tau-1}\beta_{i,n}^2]), \tag{6}$$

$$\boldsymbol{\alpha} = [0,\cdots,0,\sqrt{\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau}]^{\mathrm{T}}, \tag{7}$$

and

$$\sum\nolimits_{i=1}^{\tau-1}\beta_{i,n}^2 + \mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau = 1, \tag{8}$$

where $\beta_{i,n}$ is the n-th element of $\boldsymbol{\beta}_i$, and $diag(\mathbf{v})$ represents a diagonal matrix whose diagonal elements are assigned by the vector $\mathbf{v}$. Additionally considering the centered property of $\mathbf{V}$ in Eq. (5), we have

$$\sum\nolimits_{i=1}^{\tau-1}\beta_{i,n} + \sqrt{\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau} = 0. \tag{9}$$

Equations (8- 9) enable $1 - \mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau = \sum_{i=1}^{\tau-1}\beta_{i,n}^2 \ge \frac{1}{\tau-1}(\sum_{i=1}^{\tau-1}\beta_{i,n})^2 = \frac{1}{\tau-1}\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau$, thus $\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau \le \frac{\tau-1}{\tau} < 1$. □

According to the proof of Theorem (1), $\mathbf{V}^{\mathrm{T}}\mathbf{D}_2\mathbf{V} = \sum_{i=1}^{\tau-1}\mathbf{v}_i\mathbf{v}_i^{\mathrm{T}} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{\mathrm{T}} = \mathbf{P}diag([1,\cdots,1,1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau])\mathbf{P}^{\mathrm{T}}$. Since $1 - \mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau > 0$ holds strictly, then

Corollary 1. $\mathbf{V}^{\mathrm{T}}\mathbf{D}_2\mathbf{V}$ *is always invertible.*

Corollary 2. *The solution in Eq. (3) is reduced to the form as* $\mathbf{A}_l = \mathbf{S}\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{S}^{-1}$, *where* $\mathbf{K} = \mathbf{V}\mathbf{P} \in \mathbb{R}^{\tau\times n}$.

The formulation of the LS solution in Corollary (2) is more compact than its original form in Eq. (3) as the inverse of $\mathbf{V}^{\mathrm{T}}\mathbf{D}_2\mathbf{V}$ has been reduced to the inverse of a diagonal matrix that can be closely computed. More importantly, we can further derive the spectral radius of $\mathbf{A}_l$ as

$$\begin{aligned} \rho(\mathbf{A}_l) &= \rho(\mathbf{S}\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{S}^{-1}) \\ &= \rho(\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}^{-1}). \end{aligned} \tag{10}$$

3.2 The Upper-bound of the Spectral Radius

Applying ℓ_2 norm to Eq. (10), we obtain

$$\begin{aligned} \rho(\mathbf{A}_l) &\le \|\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}^{-1}\|_2 \\ &\le \|\mathbf{K}^{\mathrm{T}}\|_2\|\mathbf{D}_1\|\|\mathbf{K}\|_2\|\mathbf{\Lambda}^{-1}\|_2 \\ &\le \frac{1}{1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau}. \end{aligned} \tag{11}$$

Eq. (11) shows that $\mathbf{A}_l$ is guaranteed to be stable if the value of $\mathbf{v}_\tau$ is enforced to be zero. Such result is quite surprising, as the stability of $\mathbf{A}_l$ is irrelevant to the values of other rows once the last row of $\mathbf{V}$ is set to be zero. We are also aware that a similar result has been reported in [Maciejowski, 1995] where the authors performed zero-padding at the end of the observability of LDSs to guarantee stability.

However, the gap between $\rho(\mathbf{A}_l)$ and the upper-bound given by Eq. (11) would be large if the value of $\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau$ approaches to 1. A more rigid upper-bound is given by the following theorem

Theorem 2. *For any* $\mathbf{A}_l$, $\rho(\mathbf{A}_l) \le \min(b_1, b_2)$ *holds, where* $b_1 = \frac{1}{1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau}$ *and* $b_2 = 1 + \sqrt{\frac{n}{1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau}}$.

Proof. We only need to prove $\rho(\mathbf{A}_l) \le b_2$. Let $\mathbf{\Lambda}_1 = diag([1,\cdots,1,0])$ and $\mathbf{\Lambda}_2 = diag([0,\cdots,0,\frac{1}{1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau}])$. Then, Eq. (10) enables $\rho(\mathbf{A}_l) = \rho\left(\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}(\mathbf{\Lambda}_1+\mathbf{\Lambda}_2)\right)$. Specifically, $\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}_2 = [\mathbf{0},\cdots,\mathbf{0},\mathbf{a}]$, where $\mathbf{a} \in \mathbb{R}^{n\times 1}$. By denoting $K_{i,j}$ as the element at the i-th row and the j-th column of $\mathbf{K}$, $a_k = \frac{1}{1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau}\sum_{i=2}^{\tau}K_{i,k}K_{i-1,n}$ computing the k-th element of $\mathbf{a}$. We can further derive $a_k^2 \le \frac{1}{(1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau)^2}\sum_{i=2}^{\tau}K_{i,k}^2\sum_{i=2}^{\tau}K_{i-1,n}^2 \le \frac{1}{(1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau)^2}\sum_{i=2}^{\tau}K_{i-1,n}^2 = \frac{1}{(1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau)^2}(1-K_{\tau,n}^2)$ by virtue of the Cauchy inequality and the orthogonality of $\mathbf{K}$. The proof of Theorem (1) indicates $K_{\tau,n} = \alpha_n = \sqrt{\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau}$. Thus, we have $a_k^2 \le \frac{1}{1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau}$. Now, we focus back on $\rho(\mathbf{A}_l)$. Clearly, $\rho(\mathbf{A}_l) \le \|\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}(\mathbf{\Lambda}_1+\mathbf{\Lambda}_2)\|_2 \le \|\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}_1\|_2 + \|\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}_2\|_2 \le 1 + \sqrt{\mathbf{a}^{\mathrm{T}}\mathbf{a}} = 1 + \sqrt{\sum_{k=1}^{n}a_k^2} \le 1 + \sqrt{\frac{n}{1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau}}$, thus concluding the proof. □

As $\lim\limits_{\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau\to 1}\frac{b_2}{b_1} = 0$ holds, the upper-bound given by Theorem (2) will switch to b_2 when $\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau$ is approaching to 1.

Theorem (2) demonstrates that $\mathbf{A}_l$ can be adjusted to be stable in two simple ways: i) setting $\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau = 0$ as have been demonstrated in Eq. (11); and ii) dividing $\mathbf{A}_l$ with the upper-bound, *i.e.* computing $\mathbf{A}' = \frac{1}{\min(b_1,b_2)}\mathbf{A}_l$. We denote such two methods as Zero-Padding (ZP) and Bound-Normalizing (BN). Both ZP and BN will certainly cause some (sometimes large) distortion on LDS modeling since they are formulated regardless of the state reconstruction error. Anyway, they have provided efficient strategies for stabilizing the transition matrix $\mathbf{A}_l$ with hardly any extra computation overhead. More importantly, the solutions of both ZP and BN can be combined with any intermediate result of the Weighted-Least-Square (WLS) method to generate a stable transition matrix via the binary interpolation, which will be discussed in the next section.

4 Weighted-Least-Square Methods

The WLS method locally stabilizes the LS solution $\mathbf{A}_l$ by multiplying the unstable component with a weight matrix. In this section, we first provide the details of the model formulation for the WLS method; and then develop an efficient algorithm to solve the corresponding optimization problem.

4.1 Formulation

Equations (10-11) show that the instability of $\mathbf{A}_l$ is induced by the term $\mathbf{\Lambda}^{-1}$ (more precisely, the last diagonal element of $\mathbf{\Lambda}^{-1}$). One can fine-tune $\mathbf{\Lambda}^{-1}$ by multiplying it with a weight matrix on the right side, the new transition matrix can therefore be redefined as

$$\mathbf{A}_{wls} = \mathbf{S}\mathbf{P}\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}^{-1}\mathbf{W}\mathbf{P}^{\mathrm{T}}\mathbf{S}^{-1}, \tag{12}$$

where the weight matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$. Since $\rho(\mathbf{A}_{wls}) = \rho(\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}^{-1}\mathbf{W})$, the value of $\mathbf{W}$ is determined by setting $\rho(\mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}^{-1}\mathbf{W}) \leq 1$ for ensuring the stability condition.

Moreover, a good LDS model is expected to fit the original data well. Thus, we need to find an optimized $\mathbf{W}$ to minimize the reconstruction error given by Eq. (2). Particularly, we substitute the new transition matrix defined in Eq. (12) into Eq. (2), thereby formulating the objective function as

$$f(\mathbf{W}) = \sum_{i=1}^{n} \lambda_i \mathbf{W}_i^{\mathrm{T}}\mathbf{H}\mathbf{W}_i - 2\lambda_i\mathbf{H}_i^{\mathrm{T}}\mathbf{W}_i + \mathrm{tr}(\mathbf{X}_1\mathbf{X}_1^{\mathrm{T}}), \tag{13}$$

where $\mathbf{W}_i$ and $\mathbf{H}_i$ are the i-th column vectors of $\mathbf{W}$ and $\mathbf{H}$, respectively; λ_i is the i-th diagonal element of $\mathbf{\Lambda}$; and $\mathbf{H} = \mathbf{M}^{\mathrm{T}}\mathbf{M}$ with $\mathbf{M} = \mathbf{S}\mathbf{V}^{\mathrm{T}}\mathbf{D}_1\mathbf{V}\mathbf{P}\mathbf{\Lambda}^{-1}$. Note that $\mathbf{H}$ is a symmetric nonnegative-definite matrix and $f(\mathbf{W})$ is a quadratic function of $\mathbf{W}$.

Combining the reconstruction objective with the stable constraint, we attain the optimization problem in the form of

$$\begin{aligned} \min_{\mathbf{W}} \quad & f(\mathbf{W}) \\ \text{s.t.} \quad & \rho(\mathbf{L}\mathbf{W}) \leq 1, \end{aligned} \tag{14}$$

where $\mathbf{L} = \mathbf{K}^{\mathrm{T}}\mathbf{D}_1\mathbf{K}\mathbf{\Lambda}^{-1} \in \mathbb{R}^{n\times n}$.

Prior to developing the algorithm to solve the problem in Eq. (14), we first present several remarks as follows.

Remark 1. *The resulting transition matrices of ZP and BN are the feasible solutions of Eq. (14) regardless of the optimization objective, while the LS solution is the optimized solution of Eq. (14) regardless of the stable constraint. To show this, by setting* $\mathbf{W} = \mathrm{diag}([1,\cdots,1,1-\mathbf{v}_\tau^{\mathrm{T}}\mathbf{v}_\tau])$ *for ZP and* $\mathbf{W} = \mathrm{diag}([\frac{1}{\min(b_1,b_2)},\cdots,\frac{1}{\min(b_1,b_2)}])$ *for BN, we obtain the same transition matrices of ZP and BN as those defined in Section 3.2. Besides, the optimized weight of the objective in Eq. (14) ignoring the constraint is found to be the identity matrix* $\mathbf{I}_n$*, of which the corresponding transition matrix is equal to the LS solution.*

Remark 2. *Different from Eq. (4), Eq. (14) searches for the optimized transition matrix only within the weight space surrounding the LS solution. The optimized solution searched in the weight space is of course no better than that searched in the whole matrix space, theoretically. However, in practice, restricting the solution space to surround the LS solution helps improve the convergence efficiency and thus can derive a high-quality solution, especially when the LS matrix is near the bound of the feasible region. We will further demonstrate this property in our experiments.*

Remark 3. *Eq. (14) can be reduced to a more simple form by constraining the weight matrix* $\mathbf{W}$ *to be diagonal, which is able to significantly reduce the computational complexity, as the dimensionality of the solution space decreases from* n^2 *to* n*. Despite a cost of enlarging the reconstruction error, such benefit is important for LDS modeling on real-time applications. For consistency, we denote the this algorithm as the Diagonal-Weighted-Least-Square (DWLS) method below.*

4.2 Algorithm

We develop a CG-like optimization algorithm to solve Eq. (14) based on three basic operations: convex approximation of the stable constraint, iterative check of the spectral radius, and binary interpolation between the temporary solution and the stable candidate. Such procedures are inspired from the CG method [Siddiqi *et al.*, 2007] but have been improved for efficiency purposes here. The details of the implementation for WLS is provided in Algorithm 1. Note that Algorithm 1 is also applicable for DWLS.

Convex approximation of the stable constraint. The feasible region under the stable constraint in Eq. (14) is nonconvex. We perform the SVD of $\mathbf{L}\mathbf{W}$ as $\mathbf{L}\mathbf{W} = \mathbf{U}'\mathbf{S}'\mathbf{V}'$. The generated constraint is given by $\mathbf{g} = \mathrm{vec}(\mathbf{v}_1'\mathbf{u'}_1^{\mathrm{T}}\mathbf{L})$, where $\mathrm{vec}(\cdot)$ returns a vector taken column-wise from the input matrix; $\mathbf{u'}_1$ and $\mathbf{v'}_1$ are the first columns of $\mathbf{U}'$ and $\mathbf{V}'$, respectively. Since $\mathbf{g}^{\mathrm{T}}\mathrm{vec}(\mathbf{W}) \leq 1 \Rightarrow \rho(\mathbf{L}\mathbf{W}) \leq 1$, we can formulate a QP problem [2] by replacing the constraint $\rho(\mathbf{L}\mathbf{W}) \leq 1$ with $\mathbf{g}^{\mathrm{T}}\mathrm{vec}(\mathbf{W})$ to enforce stability in Eq. (14). If the optimal result of this QP is still unstable, the above process will be repeated and the generated constraint will be added.

Iterative check of the spectral radius. The convex approximation for enforcing stability is sufficient but not necessary. Thus, we compute the spectral radius of the current solution at each step. Once the spectral radius is decreased to be no larger than 1, we will stop the optimization process before we finally obtain a feasible solution with respect to the approximate constraints. Such setting helps in obtaining a high-quality solution.

Binary interpolation between the temporary solution and the stable candidate. In the CG method, for increasing the objective value of the final output, the binary interpolation is employed between the last solution and the LS solution to locate a boundary of the stable region. Apart from this purpose, in our WLS algorithm, the binary interpolation is also applied between the current solution and a selected stable candidate to obtain a feasible solution if the objective gain between two neighbouring updates is less than a certain threshold. Such setting can prevent the optimization process from getting stuck in a local minima. As demonstrated in Section 3.2, the stable candidate can be determined by ZP or BN. We chose the transition matrix of ZP, *i.e.* $\mathbf{A}_{zp}$, as the stable candidate in our experiments.

[2] As a convex problem, QP can be solved efficiently with the M function embedded in the Matlab software, *i.e.* *quadprog*( ).

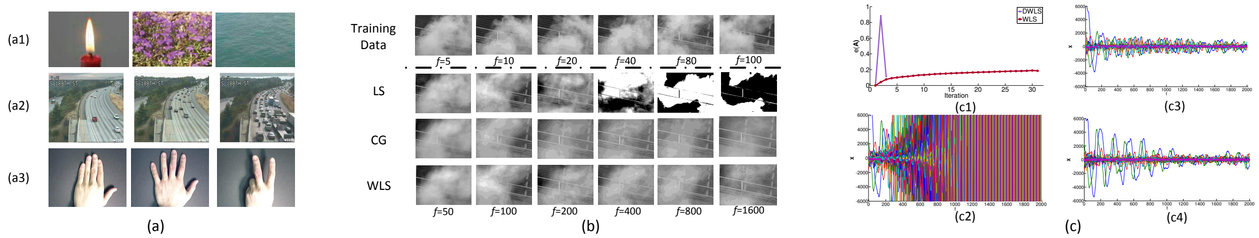

Figure 1: (a) Samples of the benchmark datasets: (a1) *UCLA*; (a2) *UCSD*; (a3) *Cambridge*. (b) Synthesized sequences generated by LS, CG and WLS. The symbol f denotes the frame index. (c) Plots on the *steam* sequence: (c1) displays the convergent curves of DWLS and WLS; (c2-c4) plot the evolutions of the state vectors in LS, CG and WLS, respectively.

Algorithm 1 Learning the stable transition matrix by WLS

```
Input: $\mathbf{S}, \mathbf{P}, \mathbf{V}, \mathbf{L}, \mathbf{A}_{zp}$
Initialize the weight matrix: $\mathbf{W} = \mathbf{I}_n$;
Initialize the constraints to be empty: $\mathbf{G} = [\,]$ and $\mathbf{b} = [\,]$;
Compute the initial point: $\mathbf{A}'_0 = \mathbf{LW}$ (Eq. (14));
Compute the initial objective: $\mathbf{f}_0 = \mathbf{f}(\mathbf{W})$ (Eq. (13));
for $i = 1$ to $nIters$ do
  Compute $\mathbf{A}' = \mathbf{LW}$ (Eq. (14));
  if $\mathbf{A}'$ is stable then
    break;
  end if
  Perform $\mathbf{A}' = \mathbf{U}'\mathbf{S}'\mathbf{V}'^{\mathrm{T}}$;
  Compute $\mathbf{g} = \mathrm{vec}(\mathbf{v}_1\mathbf{u}_1^{\mathrm{T}}\mathbf{L})$;
  Update the constraints: $\mathbf{G} = [\mathbf{G}; \mathbf{g}]$ and $\mathbf{b} = [\mathbf{b}; 1]$;
  Solve the QP problem: $\mathbf{W} = \mathrm{quadprog}(\mathbf{f}, \mathbf{G}, \mathbf{b})$;
  Return the current objective: $\mathbf{f}_i = \mathbf{f}(\mathbf{W})$ (Eq. (13));
  if $\mathbf{f}_i - \mathbf{f}_{i-1} < f_0 * $ threshold then
    Apply the binary interpolation between $\mathbf{A}'$ and $\mathbf{A}'_{zp}$:
    $\mathbf{A}' = \mathrm{interpolation}(\mathbf{A}', \mathbf{A}'_{zp})$;
    break;
  end if
end for
Apply the binary interpolation between $\mathbf{A}'$ and $\mathbf{A}'_0$:
$\mathbf{A}'_{best} = \mathrm{interpolation}(\mathbf{A}', \mathbf{A}'_0)$;
Output the transition matrix: $\mathbf{A}_{wls} = \mathbf{SPA}'_{best}\mathbf{P}^{\mathrm{T}}\mathbf{S}^{-1}$;
```

Table 1: Performance on the *steam* sequence. $n = 40$.

Methods	LB-1*	LB-2	CG	BN	ZP	DWLS	WLS
$\rho(\mathbf{A})$	0.989	0.997	1	0.466	0.986	0.999	1
$e(\mathbf{A})$	0.45	1.04	0.22	308	2.98	0.11	0.19
Time (s)	2474	35.51	5.78	0.002	0.002	0.18	6.35

5 Experiments

In this section, we compare the performance of proposed algorithms with state-of-the-art methods including LB-1, LB-2 and CG. Since LB-1 usually fails to converge in practice, we implement its simulated version LB-1* instead, as encouraged by [Siddiqi *et al.*, 2007]. The experimental performance is evaluated in terms of two criteria: the reconstruction error and the training efficiency. Following [Siddiqi *et al.*, 2007], we apply the normalized-error metric given by $e(\mathbf{A}) = (J^2(\mathbf{A}) - J^2(\mathbf{A}_l))/J^2(\mathbf{A}_l)$ where $J(\mathbf{A})$ is defined in Eq. (2). We first carry out experiments on a particular sequence to allow detailed comparisons among different methods. The sequence we choose is the *steam* dynamic texture that has been adopted to evaluate the CG method in [Siddiqi *et al.*, 2007]. Then we perform further evaluations with various sequences selected from three benchmark datasets: the dynamic texture dataset *UCLA*, the dynamic scene dataset *UCSD*, and the hand gesture dataset *Cambridge*. These datasets are applied widely in the LDS literatures; and their details are provided in the next section. All experiments are carried out with Matlab 8.1.0.604 (R2013a) on Intel Core i5, 2.20-GHz CPU with 12-GB RAM.

5.1 Benchmark Datasets

The *UCLA* dataset [Saisan *et al.*, 2001] contains 50 categories of dynamic textures. Each category consists of 4 gray-scale video sequences captured from different viewpoints. Every video sequence includes 75 frames with the original size of 160×110 pixels and has been clipped to a 48×48 window that keeps representative motion. The traffic dataset *UCSD* [Chan and Vasconcelos, 2005] consists of 254 video sequences of highway traffic with a variety of traffic patterns in various weather conditions. Each video is recoded with a resolution of 320 × 240 pixels for a duration between 42 and 52 frames. We utilize the clipped version that has been resized to the scale of 48 × 48 in this experiment. *Cambridge* [Kim and Cipolla, 2009] consists of 900 images sequences of 9 gesture classes generated by 3 primitive hand shapes and 3 primitive motions. Each class contains 100 image sequences performed by 2 subjects, with 10 arbitrary camera motions and under 5 illumination conditions. Each image has a original spatial size of 320 × 240 and has been resized to the scale of 20 × 20 as suggested by [Kim and Cipolla, 2009]. The examples of the datasets are illustrated in Figure 1 (a).

5.2 Performance on the *steam* Sequence

The *steam* sequence contains 149 frames of images with the scale of 120 × 170 pixels, as illustrated in Figure 1 (b). In this experiment, the hidden dimension of the LDS model n is set to be 40. As for WLS and DWLS, the parameter *threshold* in Algorithm 1 is fixed to be 0.0001. Table 1 reports the performance of all compared methods trained on the *steam* sequence. It is observed that both BN and ZP perform much

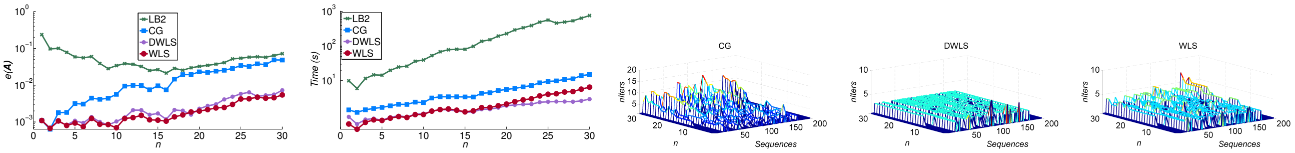

Figure 2: Performance on *UCLA*. The left two figures plot $e(\mathbf{A})$ and the training time with varying n. The right three figures display the numbers of iterations, *i.e.* $nIters$, for CG, DWLS and WLS, respectively.

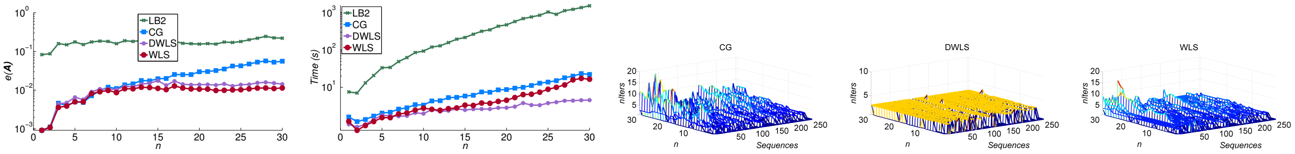

Figure 3: Performance on *UCSD*. The left two figures plot $e(\mathbf{A})$ and the training time with varying n. The right three figures display the numbers of iterations, *i.e.* $nIters$, for CG, DWLS and WLS, respectively.

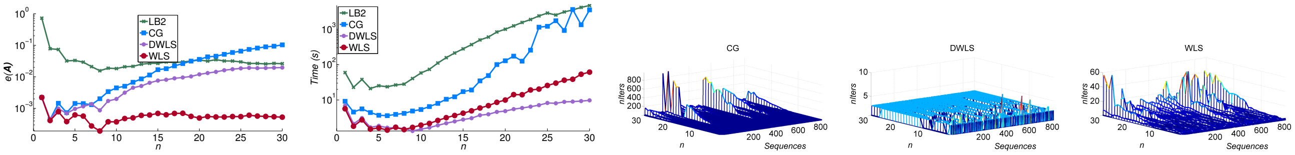

Figure 4: Performance on *Cambridge*. The left two figures plot $e(\mathbf{A})$ and the training time with varying n. The right three figures display the numbers of iterations, *i.e.* $nIters$, for CG, DWLS and WLS, respectively.

faster than other methods, while their reconstruction performance is the worst, which is due to the fact that the transition matrices of these two methods are computed regardless of the reconstruction objective. WLS achieves the lower error than previous algorithms including LB-1*, LB-2 and CG, although its execution time is a little more than CG. DWLS obtains the lowest error as well as the fastest training time among all compared methods except BN and ZP. It is surprising that DWLS can outperform WLS on the reconstruction error as it is a specific case of WLS (Remark 3). To explain why this happens, we display the convergence paths of WLS and DWLS in Figure 1 (c1). It shows that DWLS converges to the stable region within only one iteration and then obtains a high-quality solution after the binary-interpolation. Different from DWLS, WLS spends more iterations to converge, indicating that restricting the weight space to be diagonal helps speed up the convergence rate in this case.

Besides, we synthesized the simulated sequences of the *steam* dynamic texture with the systems learned by LS, CG and WLS in Figure 1 (b). Clearly, the sequence produced by the unstable LS model demonstrates a dramatic increase in image contrast over time. Both WLS and CG continue to generate qualitatively reasonable images, while those yielded by WLS look denser and more natural than CG. To reveal the evolution of the state vector, we display the resulting state values of LS, CG and WLS over time in Figure 1 (c2-c4).

5.3 Comparisons on Benchmark Datasets

In this experiment, we further evaluate the performance of the methods including LB-2, CG, DWLS, and WLS, with various sequences from the benchmark datasets introduced in Section 5.1. Given an LDS model, we first compute the normalized-errors for different series in a selected dataset, and then apply the average of the normalized-errors as the evaluation measurement for this dataset. Since the hidden dimension n influences the eventual performance dramatically, we vary its value from 1 to 30 for each compared model on all datasets.

Figures 2, 3 and 4 report the performance of the compared methods on *UCLA*, *UCSD* and *Cambridge*, respectively. As expected, WLS consistently outperforms all other methods on all datasets in terms of the reconstruction performance. DWLS performs closely to WLS on *UCLA* and *UCSD*, indicating that a desired solution can be obtained within the region of the diagonal weight matrices. Regarding the training time, both WLS and DWLS perform much more efficiently than LB-2 and CG, thus verifying the efficiency of Algorithm 1. The training time of the compared methods depends on the value of $nIters$, *i.e.* the number of iterations needed to reach a stable region. Hence, we further display $nIters$ of CG, DWLS and WLS in Figures 2-4. Obviously, DWLS and WLS spend much fewer iterations than CG, even if the implementation of Algorithm 1 is inspired from CG. The experimental results here can validate our conjecture in Remark 2 that restricting the solution space within the weight space can speed up the convergence rate and improve the eventual accuracy.

6 Conclusion

In this paper, several theoretical results have been derived to analyze the stability of the least-square solution, including the reduced form of the least-square transition matrix and the upper bound of its spectral radius. Based on these analyses, we propose the Weighted-Least-Square method and its variants to learn stable systems from time series. Experiments on various datasets demonstrate that our methods outperform previous algorithms including LB-1, LB-2 and CG, in terms of the accuracy and the efficiency.

Acknowledgments

This work is jointly supported by National Natural Science Foundation of China under Grant No. 61327809, 61210013, 91420302 and 91520201.

References

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

[Byravan *et al.*, 2015] Arunkumar Byravan, Mathew Monfort, Brian Ziebart, Byron Boots, and Dieter Fox. Graph-based inverse optimal control for robot manipulation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2015.

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

[Chan and Vasconcelos, 2009] Antoni B Chan and Nuno Vasconcelos. Layered dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(10):1862–1879, 2009.

[Doretto *et al.*, 2003] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *International Journal of Computer Vision (IJCV)*, 51(2):91–109, 2003.

[Gan *et al.*, 2015] Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alex G Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2568–2577, 2015.

[Gan *et al.*, 2016] Chuang Gan, Yi Yang, Linchao Zhu, Deli Zhao, and Yueting Zhuang. Recognizing an action using its name: A knowledge-based approach. *International Journal of Computer Vision (IJCV)*, pages 1–17, 2016.

[Huang *et al.*, 2016] Wenbing Huang, Fuchun Sun, Lele Cao, Deli Zhao, Huaping Liu, and Mehrtash Harandi. Sparse coding and dictionary learning with linear dynamical systems. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[Kazakov and Lempert, 2015] Alexander L Kazakov and Anna A Lempert. On mathematical models for optimization problem of logistics infrastructure. *International Journal of Artificial Intelligence*, 13(1):200–210, 2015.

[Kim and Cipolla, 2009] Tae-Kyun Kim and Roberto Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(8):1415–1428, 2009.

[Lacy and Bernstein, 2002] Seth L Lacy and Dennis S Bernstein. Subspace identification with guaranteed stability using constrained optimization. In *American Control Conference*, 2002.

[Lacy and Bernstein, 2003] Seth L Lacy and Dennis S Bernstein. Subspace identification with guaranteed stability using constrained optimization. *IEEE Transactions on Automatic Control*, 48(7):1259–1263, 2003.

[Maciejowski, 1995] Jan M Maciejowski. Guaranteed stability with subspace methods. *Systems & Control Letters*, 26(2):153–156, 1995.

[Mumtaz *et al.*, 2015] Adeel Mumtaz, Emanuele Coviello, Gert RG Lanckriet, and Antoni B Chan. A scalable and accurate descriptor for dynamic textures using bag of system trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(4):697–712, 2015.

[Ravichandran *et al.*, 2013] Avinash Ravichandran, Rizwan Chaudhry, and René Vidal. Categorizing dynamic textures using a bag of dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(2):342–353, 2013.

[Saisan *et al.*, 2001] Payam Saisan, Gianfranco Doretto, Ying Nian Wu, and Stefano Soatto. Dynamic texture recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–58. IEEE, 2001.

[Shumway and Stoffer, 1982] Robert H Shumway and David S Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of time series analysis*, 3(4):253–264, 1982.

[Siddiqi *et al.*, 2007] Sajid M Siddiqi, Byron Boots, and Geoffrey J Gordon. A constraint generation approach to learning stable linear dynamical systems. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

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

[Vidal and Ravichandran, 2005] René Vidal and Avinash Ravichandran. Optical flow estimation & segmentation of multiple moving dynamic textures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 516–521. IEEE, 2005.

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

[Woolfe and Fitzgibbon, 2006] Franco Woolfe and Andrew Fitzgibbon. Shift-invariant dynamic texture recognition. In *European Conference on Computer Vision (ECCV)*, pages 549–562. Springer, 2006.