Sparse Coding and Dictionary Learning with Linear Dynamical Systems*

Wenbing Huang[1], Fuchun Sun[1], Lele Cao[1], Deli Zhao[2], Huaping Liu[1] and Mehrtash Harandi[3]
[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);
[3] Australian National University & NICTA, Australia;
[1]{huangwb12@mails, fcsun@mail, caoll12@mails, hpliu@mail}.tsinghua.edu.cn,
[2]zhaodeli@gmail.com, [3]Mehrtash.Harandi@nicta.com.au,

Abstract

Linear Dynamical Systems (LDSs) are the fundamental tools for encoding spatio-temporal data in various disciplines. To enhance the performance of LDSs, in this paper, we address the challenging issue of performing sparse coding on the space of LDSs, where both data and dictionary atoms are LDSs. Rather than approximate the extended observability with a finite-order matrix, we represent the space of LDSs by an infinite Grassmannian consisting of the orthonormalized extended observability subspaces. Via a homeomorphic mapping, such Grassmannian is embedded into the space of symmetric matrices, where a tractable objective function can be derived for sparse coding. Then, we propose an efficient method to learn the system parameters of the dictionary atoms explicitly, by imposing the symmetric constraint to the transition matrices of the data and dictionary systems. Moreover, we combine the state covariance into the algorithm formulation, thus further promoting the performance of the models with symmetric transition matrices. Comparative experimental evaluations reveal the superior performance of proposed methods on various tasks including video classification and tactile recognition.

1. Introduction

Object recognition based on spatio-temporal data is an active research area across several domains such as machine learning [3, 18], computer vision [34, 16, 17] and robotics [28]. The coupling of the spatial texture and the temporal dynamics makes spatio-temporal data analysis more challenging than static data. A popular method of representing spatio-temporal data is to model them by Linear Dynamical Systems (LDSs) [9]. To allow the comparison between dynamical processes, a distance metric or kernel function needs to be defined first. Once the distance or kernel has been defined, classifiers such as Nearest Neighbors (NNs) and Support Vector Machines (SVMs) can be used to recognize spatio-temporal sequences. For this purpose, various kinds of distances or kernels have been proposed, such as Martin Distance [34, 6], Kullback-Leibler divergence [5], and Binet-Cauchy kernel [40]. Several recent studies have been carried out to integrate learning techniques into LDSs; for instance, Vidal *et al.* [39] proposed a LDS-based boosting method for time series modeling; and Ravichandran *et al.* [33] designed bag-of-systems for video analysis.

Despite the wide applications of LDSs, little attention has been paid to combining sparse coding with LDS modeling to deliver robust techniques. In the past decade, sparse coding has been successfully adopted in various tasks such as image restoration [30], face recognition [43], and texture classification [31] to name a few. For sparse coding, natural signals such as images are represented as a combination of a few atoms in a dictionary that is usually over-complete. Using sparsity as a prior leads to state-of-the-art results in many fields [43]. In this paper, we generalize sparse coding from Euclidean space to the space of LDSs. Specifically, we attempt to reconstruct a given LDS by using a superposition of LDS atoms, where the coefficients of the superposition are enforced to be sparse. Both the codes and the dictionary atoms are learned to minimize the coding objective function. Sparse coding with the LDS dictionary can then be seamlessly used for categorizing spatio-temporal data.

However, the space of LDSs, which is non-Euclidean, has a complicated manifold structure [1, 33]. Carrying out sparse coding and dictionary learning on this kind of space is challenging. Recent studies such as [37] proposed to embed LDSs into a finite-dimensional Grassmann manifold. With this embedding, sparse coding and dictionary learning with LDSs can then be performed on the finite Grassmannian [23, 21]. The first cornerstone of these models [23, 21] is to represent each LDS with its finite observability sub-

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

space by taking a fixed-order approximation of the extended observability matrix. Nevertheless, as we will discuss in this paper, this may result in several drawbacks. Firstly, such finite approximation is computationally expensive if the observability order is large; but it is insufficient to model the changes along the rows of the extended observability otherwise. Secondly, if we want to learn the dictionary atoms with the finite method, we can only learn the embedding points of the finite observability but not the parameters of the dictionary LDSs (*e.g.* the measurement matrix and the transition matrix). It is believed that these parameters are important for further analysis of the learned dictionary. Moreover, various methods have been developed for defining the distance metric [32, 7, 34] and performing classification tasks [34, 6, 33] on the space of infinite LDSs, indicating that deriving sparse coding and dictionary learning with infinite LDSs could be theoretically interesting.

Hence, in this paper, we attempt to make the following contributions. (1) We perform sparse coding and dictionary learning with the original form of LDSs that is represented by the extended observability subspace. As a more general framework of [21], learning the codes and dictionary atoms on infinite Grassmannian maintains the full changes along the sequences. More importantly, in our models, the calculations related to the infinite observability subspaces can be efficiently derived by the representation of the system parameters, which enables us to learn the system parameters of the dictionary explicitly and reduce the computational cost significantly compared to the finite method. (2)To overcome the limitation caused by the the symmetry constraint to the state transition matrix in dictionary learning, we additionally consider the state covariance as a complementary feature of the symmetric transition matrix to describe the state process, thus further promoting the modeling performance. (3) We employ proposed models to categorize spatio-temporal sequences on diversified benchmark datasets including videos and tactile series. Compared to state-of-the-art methods, our models achieve considerable improvements in discrimination accuracy on most tasks.

The rest of the paper is organized as follows. Section 2 reviews the LDS preliminaries. Sparse coding is derived in Section 3 and dictionary learning is developed in Section 4. Then, Section 5 combines the state covariance into the algorithm framework and Section 6 analyzes the computational complexities of proposed models. Finally, Section 7 conducts the experiments; and Section 8 concludes this paper.

2. Briefs of Fundamental Concepts

2.1. Linear dynamical systems

LDSs represent time series by assuming them to be the output of the following 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 $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_T] \in \mathbb{R}^{n\times T}$ is a sequence of n-dimensional hidden state vectors, and $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_T] \in \mathbb{R}^{m\times T}$ is a sequence of m-dimensional observed variables. The model is parameterized by $\Theta = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{R}, \overline{\mathbf{y}}\}$, where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the transition matrix; $\mathbf{C} \in \mathbb{R}^{m\times n}$ is the measurement matrix; $\mathbf{B} \in \mathbb{R}^{n\times n_v}(n_v \leq n)$ is the noise transformation matrix; $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{I}_{n_v\times n_v})$ and $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{R})$ denote the process and measurement noise components, respectively; $\overline{\mathbf{y}} \in \mathbb{R}^m$ represents the mean of $\mathbf{Y}$. Given the observed sequence, several methods [38, 36] have been proposed to learn the optimal system parameters, while the method in [9] is widely used.

Since $\mathbf{C}$ describes the spatial appearance and $\mathbf{A}$ represents the dynamics, the tuple $(\mathbf{A}, \mathbf{C})$ can be adopted as the feature descriptor for an LDS . Unfortunately, ($\mathbf{A}$,$\mathbf{C}$) does not lie in a vector space as it needs to satisfy several constraints [37]. The transition matrix $\mathbf{A}$ needs to be stable with eigenvectors inside the unit circle. The columns of $\mathbf{C}$ are constrained to be orthonormal. Furthermore, any Riemannian metric for the space of LDS needs to be invariant to the changes of the state space basis. All these constraints make it hard to determine the Riemannian geometry of the LDS space [33]. To circumvent the difficulties associated to utilizing the tuple $(\mathbf{A}, \mathbf{C})$, a family of approaches apply the extended observability subspace to represent an LDS [34, 6, 33, 37], which is the topic of the next section.

2.2. The extended observability subspaces

Starting from the initial state $\mathbf{x}_1$, the expected observation sequence is obtained as $\mathrm{E}[\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \cdots] = [\mathbf{C}^{\mathrm{T}}, (\mathbf{CA})^{\mathrm{T}}, (\mathbf{CA}^2)^{\mathrm{T}}, \cdots]^{\mathrm{T}}\mathbf{x}_1$, meaning that it lies in the column space of the extended observability matrix given by $\mathbf{O} = [\mathbf{C}^{\mathrm{T}}, (\mathbf{CA})^{\mathrm{T}}, (\mathbf{CA}^2)^{\mathrm{T}}, \cdots]^{\mathrm{T}} \in \mathbb{R}^{\infty\times n}$. Since the column space of $\mathbf{O}$, *i.e.* the extended observability subspace, is invariant to the choice of the basis of the state space, it can be applied as the descriptor of an LDS. Therefore, the distance between two LDSs is considered as the distance between the respective extended observability subspaces, which can be derived by computing the subspace angles [7]. The subspace angles between two extended observability matrices $\mathbf{O}_1$ and $\mathbf{O}_2$, associated with parameters ($\mathbf{A}_1$,$\mathbf{C}_1$) and ($\mathbf{A}_2$,$\mathbf{C}_2$) respectively, can be calculated by solving the following Lyapunov equation

$$\mathbf{A}_i^{\mathrm{T}}\mathbf{O}_{ij}\mathbf{A}_j - \mathbf{O}_{ij} = -\mathbf{C}_i^{\mathrm{T}}\mathbf{C}_j, \tag{2}$$

where $\mathbf{O}_{ij} = \mathbf{O}_i^{\mathrm{T}}\mathbf{O}_j = \sum_{t=0}^{\infty}(\mathbf{A}_i^t)^{\mathrm{T}}\mathbf{C}_i^{\mathrm{T}}\mathbf{C}_j\mathbf{A}_j^t,\ i,j \in \{1,2\}$. The squared cosine of the subspace angle α_k is equal to the k-th principal eigenvalue of $\mathbf{O}_{11}^{-1}\mathbf{O}_{12}\mathbf{O}_{22}^{-1}\mathbf{O}_{21}$. The LDS distance (such as geodesic distance [44] and Martin distance [32]) can then be defined with the subspace angles.

2.3. Sparse coding on finite Grassmannian

As proposed by [37, 21], one can approximate the extended observability by taking the L-order observability

matrix, *i.e.* $\mathbf{O}(n,L) = [\mathbf{C}^{\mathrm{T}}, (\mathbf{CA})^{\mathrm{T}}, \cdots, (\mathbf{CA}^{L-1})^{\mathrm{T}}]^{\mathrm{T}}$. In this way, an LDS can be alternately identified as an n-dimensional subspace of $\mathbb{R}^{Lm}$. Sparse coding with LDSs is then performed on finite Grassmannian. Because it is hard to define tractable arithmetical calculations and distance metric on Grassmannian, Harandi *et al.* [21] homeomorphically embeds the Grassmannian into the space of symmetric matrices, thus leading to the coding objective:

$$\min_{\mathbf{Z}} \sum_{i=1}^{N} \parallel \mathbf{X}_i\mathbf{X}_i^{\mathrm{T}} - \sum_{j=1}^{K} \mathbf{Z}_{j,i}\mathbf{D}_j\mathbf{D}_j^{\mathrm{T}} \parallel_F^2 + \lambda \parallel [\mathbf{Z}]_i \parallel_1 . \quad (3)$$

Here, $\mathbf{X}_i \in \mathbb{R}^{Lm\times n}$ and $\mathbf{D}_j \in \mathbb{R}^{Lm\times n}$ are the L-order orthonormalized observability matrices of the i-th data LDS and the j-th dictionary atom, respectively, while the coefficient matrix is $\mathbf{Z} \in \mathbb{R}^{K\times N}$ and $[\mathbf{Z}]_i$ denotes the i-th column of $\mathbf{Z}$. The learning task aims to represent each data in the set $\{\mathbf{X}_i\}_{i=1}^N$ of size N as a sparse linear combination of the dictionary atoms $\{\mathbf{D}_j\}_{j=1}^K$ of size K, where $\mathbf{Z}_{j,i}$ is the representation coefficient of $\mathbf{X}_i$ with respect to $\mathbf{D}_j$. The ℓ_1-norm regularization is employed to the coefficients $\{[\mathbf{Z}]_i\}_{i=1}^N$ for sparsity assurance; and λ is the sparsity penalty factor.

3. Sparse Coding with Infinite LDSs

Approximating the observability with a finite matrix results in an unavoidable issue about how to choose the value of the order L: if L is small, it is insufficient to model the asymptotical behavior of the extended observability; increasing the value of L could make the finite observability contain rich information but also increase the computational complexity. In this section, we perform sparse coding directly on the space of extended observability subspaces, *i.e.* infinite Grassmannian. To this end, the space formulation, the distance metric and arithmetical calculations on infinite Grassmannian should be discussed.

3.1. Formulation of the infinite Grassmannian

The group of the extended observability matrices together with the stability and orthonormality constraints for $\mathbf{A}$ and $\mathbf{C}$ respectively, can be written as $\mathcal{O}(n,\infty) = \{\mathbf{O} \mid \mathbf{O} = [\mathbf{C}^{\mathrm{T}}, (\mathbf{CA})^{\mathrm{T}}, (\mathbf{CA}^2)^{\mathrm{T}}, \cdots]^{\mathrm{T}}, \mathbf{C}^{\mathrm{T}}\mathbf{C} = \mathbf{I}_n, \mid \mu(\mathbf{A}) \mid < 1\}$, where $\mathbf{I}_n$ is a $n \times n$ identity matrix, and $\mu(\mathbf{A})$ denotes an arbitrary eigenvalue of $\mathbf{A}$. Prior to further derivation, we need to perform orthonormalization on $\mathcal{O}(n,\infty)$ by virtue of the Cholesky decomposition. For any $\mathbf{O} \in \mathcal{O}(n,\infty)$, we derive the Cholesky decomposition $\mathbf{L} = Chol(\mathbf{O}^{\mathrm{T}}\mathbf{O})$, *i.e.* $\mathbf{L}\mathbf{L}^{\mathrm{T}} = \mathbf{O}^{\mathrm{T}}\mathbf{O}$, where $\mathbf{L}$ is a lower triangular matrix. According to Equation (2), $\mathbf{O}^{\mathrm{T}}\mathbf{O}$ is positive definite as $\mathbf{A}$ is stable. Thus, the Cholesky decomposition of $\mathbf{O}^{\mathrm{T}}\mathbf{O}$ always exists, and $\mathbf{L}$ is guaranteed to be inventible. The columns of the matrix $\mathbf{V} = \mathbf{O}\mathbf{L}^{-\mathrm{T}}$ are orthonormal and span the same subspace as the columns of $\mathbf{O}$. We denote the orthonormalization of $\mathcal{O}(n,\infty)$ as $\mathcal{V}(n,\infty) = \{\mathbf{V} \mid \mathbf{V} = \mathbf{O}\mathbf{L}^{-\mathrm{T}}, \mathbf{L} = Chol(\mathbf{O}^{\mathrm{T}}\mathbf{O}), \mathbf{O} \in \mathcal{O}(n,\infty)\}$. The quotient space of $\mathcal{V}(n,\infty)$ is defined as $\mathcal{S}(n,\infty)$ based on the equivalence relation $\sim$ which is given by: for any $\mathbf{V}_1, \mathbf{V}_2 \in \mathcal{V}(n,\infty)$, $\mathbf{V}_1 \sim \mathbf{V}_2$ if and only if $Span(\mathbf{V}_1) = Span(\mathbf{V}_2)$, where $Span(\mathbf{V})$ denotes the subspace spanned by columns of $\mathbf{V}$. The infinite Grassmannian that is embedded in the infinite-dimensional vector space, *i.e.* $\mathcal{G}(n,\infty)$, has already been defined in [44]. The definition of $\mathcal{S}(n,\infty)$ indicates that $\mathcal{S}(n,\infty)$ is actually a special $\mathcal{G}(n,\infty)$ with an extra intrinsic structure due to the stability and orthonormality constraints to $\mathbf{A}$ and $\mathbf{C}$, respectively. We represent LDSs with points in $\mathcal{S}(n,\infty)$.

3.2. Constructing the coding objective

Inspired by the method proposed in [21], we attempt to embed $\mathcal{S}(n,\infty)$ into the space of symmetric matrices via mapping $\Pi : \mathcal{S}(n,\infty) \to \mathrm{Sym}(\infty), \Pi(\mathbf{V}) = \mathbf{V}\mathbf{V}^{\mathrm{T}}$. The metric on $\mathrm{Sym}(\infty)$ is naturally induced by the Frobenius norm: $\parallel \mathbf{W} \parallel_F^2 = \mathrm{Tr}(\mathbf{W}^{\mathrm{T}}\mathbf{W})$, $\mathbf{W} \in \mathrm{Sym}(\infty)$. However, it will encounter the difficulty that the Frobenius norm of a point on $\mathrm{Sym}(\infty)$ is usually infinite due to the infinite dimensionality. Fortunately, the Frobenius norm of the point in the embedding $\Pi(\mathcal{S}(n,\infty))$ is guaranteed to be finite, which can be derived by Corollary 1. More generally, the Frobenius norm of the linear combination of the points in $\Pi(\mathcal{S}(n,\infty))$ is finite, as proven in the following theorem.

Theorem 1. *Suppose* $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_M \in \mathcal{S}(n,\infty)$, *and* $y_1, y_2, \cdots, y_M \in \mathbb{R}$, *we have*

$$\parallel \sum_{i=1}^{M} y_i \Pi(\mathbf{V}_i) \parallel_F^2 \quad = \quad \sum_{i,j=1}^{M} y_i y_j \parallel \mathbf{V}_i^{\mathrm{T}}\mathbf{V}_j \parallel_F^2,$$

where $\mathbf{V}_i^{\mathrm{T}}\mathbf{V}_j = \mathbf{L}_i^{-1}\mathbf{O}_i^{\mathrm{T}}\mathbf{O}_j\mathbf{L}_j^{-\mathrm{T}}$. $\mathbf{O}_i^{\mathrm{T}}\mathbf{O}_j$ *is computed with the Lyapunov equation defined in Equation (2),* $\mathbf{L}_i$ *and* $\mathbf{L}_j$ *are Cholesky decomposition matrices for* $\mathbf{O}_i^{\mathrm{T}}\mathbf{O}_i$ *and* $\mathbf{O}_j^{\mathrm{T}}\mathbf{O}_j$, *respectively.*[1]

Based on Theorem (1), we have two corollaries:

Corollary 1. *For any* $\mathbf{V}_1, \mathbf{V}_2 \in \mathcal{S}(n,\infty)$, *we have*

$$\parallel \Pi(\mathbf{V}_1) - \Pi(\mathbf{V}_2) \parallel_F^2 \quad = \quad 2\left(n - \parallel \mathbf{V}_1^{\mathrm{T}}\mathbf{V}_2 \parallel_F^2\right).$$

Furthermore, $\parallel \Pi(\mathbf{V}_1) - \Pi(\mathbf{V}_2) \parallel_F^2 = 2\sum_{k=1}^{n} \sin^2 \alpha_k$, *where* $\{\alpha_k\}_{k=1}^n$ *are subspace angles between* $\mathbf{V}_1$ *and* $\mathbf{V}_2$.

Corollary 2. *The embedding map* $\Pi(\mathbf{V})$ *is diffeomorphism (a one-to-one, continuous, and differentiable mapping with a continuous and differentiable inverse), meaning that* $\mathcal{S}(n,\infty)$ *is topologically isomorphic to the embedding* $\Pi(\mathcal{S}(n,\infty))$, *i.e.* $\mathcal{S}(n,\infty) \cong \Pi(\mathcal{S}(n,\infty))$.

Hence, the sparse coding objective function on infinite Grassmannian is formulated as $\min_{\mathbf{Z}} L(\mathbf{Z}, \mathbb{D})$, where

$$L(\mathbf{Z}, \mathbb{D}) = \sum_{i=1}^{N} \mathrm{dist}^2(\mathbf{V}_i, \mathbb{D}) + \lambda \parallel [\mathbf{Z}]_i \parallel_1, \quad (4)$$

[1] The proofs of all theorems are given in the supplementary material.

Algorithm 1 Dictionary learning with infinite LDSs

Input: $\mathbf{X}$
Extract the data system parameters $\{(\mathbf{\Theta}_i, \mathbf{C}_i)\}_{i=1}^N$ with Algorithm 2 (proposed in the supplementary material);
Assign the values of the dictionary system parameters $\{(\bar{\mathbf{\Theta}}_r, \bar{\mathbf{C}}_r)\}_{r=1}^K$ by random;
for $t = 1$ **to** $MaxNumIters$ **do**
  Learn the sparse codes $\mathbf{Z}$ by solving Equation (5);
  for $r = 1$ **to** K **do**
    for $k = 1$ **to** n **do**
      Compute $\mathbf{S}(r, k)$ as defined in Equation (8);
      Update $[\bar{\mathbf{C}}_r]_k$ according to Theorem 3;
      Update $\bar{\boldsymbol{\theta}}_{r,k}$ according to Equation (10);
    end for
  end for
end for
Output: $\{(\bar{\mathbf{\Theta}}_r, \bar{\mathbf{C}}_r)\}_{r=1}^K$

and, $\mathrm{dist}(\mathbf{V}_i, \mathbb{D}) = \| \mathbf{V}_i\mathbf{V}_i^{\mathrm{T}} - \sum_{j=1}^K \mathbf{Z}_{j,i}\mathbf{D}_j\mathbf{D}_j^{\mathrm{T}} \|_F$; $\mathbf{V}_i$ and $\mathbf{D}_j$ are points in $\mathcal{S}(n, \infty)$.

According to Theorem (1), we can ignore the terms that are irrelevant to $\mathbf{Z}$ and rewrite $L(\mathbf{Z}, \mathbb{D})$ as

$$\sum_{i=1}^N [\mathbf{Z}]_i^{\mathrm{T}}\mathbf{K}(\mathbb{D})[\mathbf{Z}]_i - 2[\mathbf{Z}]_i^{\mathrm{T}}\mathbf{k}(\mathbf{V}_i, \mathbb{D}) + \lambda \| [\mathbf{Z}]_i \|_1, \quad (5)$$

where $\mathbf{K}(\mathbb{D})_{i,j} = \| \mathbf{D}_i^{\mathrm{T}}\mathbf{D}_j \|_F^2$ and $[\mathbf{k}(\mathbf{V}_i, \mathbb{D})]_j = \| \mathbf{V}_i^{\mathrm{T}}\mathbf{D}_j \|_F^2$. This problem is convex as $\mathbf{K}(\mathbb{D})$ is positive semi-definite. It can be solved efficiently by using methods like homotopy-LARS algorithm [8]. We are aware that Equation (5) is similar to the kernel sparse coding for static images which is recently proposed in [19]. However, our goal is to obtain sparse coding of LDSs using LDSs as dictionary atoms. Moreover, the dedicated algorithm for dictionary learning should be devised, which will be discussed in the next section.

4. Dictionary Learning with Infinite LDSs

The dictionary learning problem is finding the good dictionary that has a small reconstruction error over all observations while preserving the sparsity penalty. Based on Equation (4), dictionary learning on LDSs can be defined as $\min_{\mathbf{Z},\mathbb{D}} L(\mathbf{Z}, \mathbb{D})$. A common approach for solving this problem is to update $\mathbf{Z}$ and $\mathbb{D}$ alternately. When the dictionary $\mathbb{D}$ is fixed, optimizing the codes $\mathbf{Z}$ is exactly the sparse coding problem raised in Equation (5). In reverse, to update dictionary atoms with the codes fixed, we break the minimization problem into K sub-minimization problems by updating each atom independently. As we have denoted in Section 3.1, each dictionary atom or data sequence is associated with a parameter tuple consisting of a transition matrix and a measurement matrix. The tuples of the atom $\mathbf{D}_r$ and the data $\mathbf{V}_i$ are $(\bar{\mathbf{A}}_r, \bar{\mathbf{C}}_r)$ and $(\mathbf{A}_i, \mathbf{C}_i)$, respectively. By substituting the tuples into $L(\mathbf{Z}, \mathbb{D})$ and ignoring the terms that are irrelevant to dictionary atoms, dictionary learning can be seen as solving $\min_{\mathbb{D}} \sum_{r=1}^K 2\Gamma(r)$, where

$$\Gamma(r) = \sum_{i=1}^N \sum_{\substack{j=1 \\ j\neq r}}^K \mathbf{Z}_{r,i}\mathbf{Z}_{j,i} \| \bar{\mathbf{L}}_r^{-1} \sum_{t=0}^{\infty} (\bar{\mathbf{A}}_r^t)^{\mathrm{T}}\bar{\mathbf{C}}_r^{\mathrm{T}}\bar{\mathbf{C}}_j\bar{\mathbf{A}}_j^t\bar{\mathbf{L}}_j^{-\mathrm{T}} \|_F^2 - \sum_{i=1}^N \mathbf{Z}_{r,i} \| \bar{\mathbf{L}}_r^{-1} \sum_{t=0}^{\infty} (\bar{\mathbf{A}}_r^t)^{\mathrm{T}}\bar{\mathbf{C}}_r^{\mathrm{T}}\mathbf{C}_i\mathbf{A}_i^t\mathbf{L}_i^{-\mathrm{T}} \|_F^2 . \quad (6)$$

Here, $\bar{\mathbf{L}}_j$ and $\mathbf{L}_i$ are the Cholesky decomposition matrices for orthonormalizing the extended observability matrices associated with the dictionary atom $\mathbf{D}_j$ and the data $\mathbf{V}_i$, respectively. By imposing the stability constraint to $\bar{\mathbf{A}}_r$ and the orthonormality constraint to $\bar{\mathbf{C}}_r$, the sub-problem can be written as

$$\min_{\bar{\mathbf{A}}_r, \bar{\mathbf{C}}_r} \Gamma(r), \quad \text{s.t. } \bar{\mathbf{C}}_r^{\mathrm{T}}\bar{\mathbf{C}}_r = \mathbf{I}_n;\ |\mu(\bar{\mathbf{A}}_r)| < 1. \quad (7)$$

There are mainly two challenges in solving this sub-problem: (1) The infinite summations involved in Equation (6) make the transition matrix and the measurement matrix coupled together, hence impeding separate update of $\bar{\mathbf{A}}_r$ and $\bar{\mathbf{C}}_r$. (2) For any orthonormal square matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$, the tuple $(\mathbf{P}^{-1}\bar{\mathbf{A}}_r\mathbf{P}, \bar{\mathbf{C}}_r\mathbf{P})$ derives the same objective $\Gamma(r)$ as $(\bar{\mathbf{A}}_r, \bar{\mathbf{C}}_r)$, implying that $(\bar{\mathbf{A}}_r, \bar{\mathbf{C}}_r)$ does not lie in a Euclidean space. The traditional optimization methods adopted in Euclidean space such as gradient decent method and Newton method may be inapplicable to this problem.

Fortunately, this minimization sub-problem can be efficiently addressed if assuming the transition matrices of the dictionary and the data to be symmetric. As presented in the supplement material, if $\bar{\mathbf{A}}_r$ is symmetric, $(\bar{\mathbf{A}}_r, \bar{\mathbf{C}}_r)$ can be equivalently transformed to $(\bar{\mathbf{\Theta}}_r, \hat{\mathbf{C}}_r)$, where the diagonal matrix $\bar{\mathbf{\Theta}}_r$ consists of the eigenvalues of $\bar{\mathbf{A}}_r$; $\hat{\mathbf{C}}_r = \bar{\mathbf{C}}_r\mathbf{P}_r^{-1}$ and $\mathbf{P}_r$ is an orthonormal square matrix. For consistency, we denote $\hat{\mathbf{C}}_r$ as $\bar{\mathbf{C}}_r$ by ignoring the difference between them in the following context. We can derive:

Theorem 2. *If the transition matrices of dictionary atoms and the data systems are all symmetric, then Equation (7) is equivalent to*

$$\min_{\bar{\mathbf{C}}_r, \bar{\boldsymbol{\theta}}_r} \sum_{k=1}^n [\bar{\mathbf{C}}_r]_k^{\mathrm{T}}\mathbf{S}(r, k)[\bar{\mathbf{C}}_r]_k \quad \text{s.t.} \quad \bar{\mathbf{C}}_r^{\mathrm{T}}\bar{\mathbf{C}}_r = \mathbf{I}_n; \quad -1 < \bar{\boldsymbol{\theta}}_{r,k} < 1,\ 1 \le k \le n. \quad (8)$$

Here, $\mathbf{S}(r, k) = \sum_{i=1}^N \sum_{j=1, j\neq r}^K \mathbf{Z}_{r,i}\mathbf{Z}_{j,i}\bar{\mathbf{C}}_j\mathbf{E}(r, j, k)\bar{\mathbf{C}}_j^{\mathrm{T}} - \sum_{i=1}^N \mathbf{Z}_{r,i}\mathbf{C}_i\mathbf{F}(r, i, k)\mathbf{C}_i^{\mathrm{T}}$; *Both* $\mathbf{E}(r, j, k)$ *and* $\mathbf{F}(r, i, k)$ *are diagonal matrices:*

$$\mathbf{E}(r, j, k) = \mathrm{diag}([\tfrac{(1-\bar{\boldsymbol{\theta}}_{r,k}^2)(1-\bar{\boldsymbol{\theta}}_{j,1}^2)}{(1-\bar{\boldsymbol{\theta}}_{r,k}\bar{\boldsymbol{\theta}}_{j,1})^2}, \cdots, \tfrac{(1-\bar{\boldsymbol{\theta}}_{r,k}^2)(1-\bar{\boldsymbol{\theta}}_{j,n}^2)}{(1-\bar{\boldsymbol{\theta}}_{r,k}\bar{\boldsymbol{\theta}}_{j,n})^2}]);$$

$$\mathbf{F}(r, i, k) = \mathrm{diag}([\tfrac{(1-\bar{\boldsymbol{\theta}}_{r,k}^2)(1-\boldsymbol{\theta}_{i,1}^2)}{(1-\bar{\boldsymbol{\theta}}_{r,k}\boldsymbol{\theta}_{i,1})^2}, \cdots, \tfrac{(1-\bar{\boldsymbol{\theta}}_{r,k}^2)(1-\boldsymbol{\theta}_{i,n}^2)}{(1-\bar{\boldsymbol{\theta}}_{r,k}\boldsymbol{\theta}_{i,n})^2}]);$$

where $[\bar{\boldsymbol{\theta}}_{j,1}, \cdots, \bar{\boldsymbol{\theta}}_{j,n}]$ *and* $[\boldsymbol{\theta}_{i,1}, \cdots, \boldsymbol{\theta}_{i,n}]$ *denote the eigenvalues of the matrix* $\bar{\mathbf{A}}_j$ *and* $\mathbf{A}_i$*, respectively.*

We further break the optimization in Equation (8) into n sub-minimization problems. Precisely speaking, we find the optimal pair $([\bar{\mathbf{C}}_r]_k, \bar{\boldsymbol{\theta}}_{r,k})$ by fixing other pairs $\{([\bar{\mathbf{C}}_r]_o, \bar{\boldsymbol{\theta}}_{r,o})\}_{o=1,o\neq k}^{n}$, thereby leading to the following sub-minimization problem,

$$\begin{aligned} \min_{[\bar{\mathbf{C}}_r]_k, \bar{\boldsymbol{\theta}}_{r,k}} \quad & [\bar{\mathbf{C}}_r]_k^{\mathrm{T}} \mathbf{S}(r,k) [\bar{\mathbf{C}}_r]_k \\ \text{s.t.} \quad & [\bar{\mathbf{C}}_r]_k^{\mathrm{T}} [\bar{\mathbf{C}}_r]_k = 1,\ [\bar{\mathbf{C}}_r]_k^{\mathrm{T}} [\bar{\mathbf{C}}_r]_o = 0, \\ & 1 \leq o \leq n,\ o \neq k, -1 < \bar{\boldsymbol{\theta}}_{r,k} < 1. \end{aligned} \tag{9}$$

We are able to obtain the solution of $[\bar{\mathbf{C}}_r]_k$ for Equation (9) by the following theorem.

Theorem 3. *We denote $[\bar{\mathbf{C}}_r]_{-k} \in \mathbb{R}^{m\times(n-1)}$ as the sub-matrix of $\bar{\mathbf{C}}_r$ by removing the column $[\bar{\mathbf{C}}_r]_k$, i.e. $[\bar{\mathbf{C}}_r]_{-k} = [[\bar{\mathbf{C}}_r]_1, \cdots, [\bar{\mathbf{C}}_r]_{k-1}, [\bar{\mathbf{C}}_r]_{k+1}, \cdots, [\bar{\mathbf{C}}_r]_n]$, and define $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_{m-n+1}] \in \mathbb{R}^{m\times(m-n+1)}$ as the orthonormal basis of the orthonormal complement of $[\bar{\mathbf{C}}_r]_{-k}$. If $\mathbf{u} \in \mathbb{R}^{(m-n+1)\times 1}$ is the eigenvector of $\mathbf{W}^{\mathrm{T}}\mathbf{S}(r,k)\mathbf{W}$ corresponding to the smallest eigenvalue, then $\mathbf{W}\mathbf{u}$ is the optimal solution of $[\bar{\mathbf{C}}_r]_k$ for Equation (9).*

We apply gradient-based method to update $\bar{\boldsymbol{\theta}}_{r,k}$. Since the value of $\bar{\boldsymbol{\theta}}_{r,k}$ is constrained within $(-1, 1)$, an auxiliary variable $\boldsymbol{\rho}_{r,k}$ is used to replace $\bar{\boldsymbol{\theta}}_{r,k}$ by setting

$$\bar{\boldsymbol{\theta}}_{r,k} = 2Sig(\boldsymbol{\rho}_{r,k}) - 1, \tag{10}$$

where $Sig(\cdot)$ is a sigmoid function. The gradient of the objective function in Equation (9) with respect to $\boldsymbol{\rho}_{r,k}$ is given by $\frac{\partial\Phi(r,k)}{\partial\boldsymbol{\rho}_{r,k}} = 2\frac{\partial\Phi(r,k)}{\partial\bar{\boldsymbol{\theta}}_{r,k}}\frac{\partial Sig(\boldsymbol{\rho}_{r,k})}{\partial\boldsymbol{\rho}_{r,k}}$, where $\Phi(r,k) = [\bar{\mathbf{C}}_r]_k^{\mathrm{T}}\mathbf{S}(r,k)[\bar{\mathbf{C}}_r]_k$.

In our dictionary learning algorithm, we use LDS with Symmetric Transition matrix (LDSST) to model the spatio-temporal data. Given the observed sequences, learning the transition matrix in LDSST is different from that in LDS. The details are presented in the supplementary material. For reader's convenience, we provide the algorithmic procedures for dictionary learning in Algorithm 1.

5. Models Considering the State Covariance

We have derived sparse coding and dictionary learning by parameterizing each LDS with the tuple $(\mathbf{A}, \mathbf{C})$. As shown in Equation (1), the matrix $\mathbf{B}$ determines the covariance of the state process. Applying $\mathbf{B}$ as an additional descriptor is able to re-discover the dynamical patterns contained in the covariance component when $\mathbf{A}$ can not model the dynamics well. In our dictionary learning algorithm, we constrain $\mathbf{A}$ to be symmetric, which could somewhat limit the modeling ability of LDSs . Combining the matrix $\mathbf{B}$ into the model formulation helps to overcome this limitation.

The covariance matrix of the whole sequence derived by [5] is hard to be combined in our models. In this paper, we consider the one-step covariance. Equation (1) demonstrates that the conditional probability of frame $\mathbf{y}_{t+1}$ given $\mathbf{x}_t$ is expressed as $p(\mathbf{y}_{t+1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{y}_{t+1}; \mathbf{C}\mathbf{A}\mathbf{x}_t + \bar{\mathbf{y}}, \mathbf{C}\mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}} + \mathbf{R})$, with the one-step covariance of $\mathbf{C}\mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}+\mathbf{R}$. We neglect the measurement covariance $\mathbf{R}$ as we only focus on the covariance of the state dynamic. As presented in the supplementary material, $\mathbf{B} = \mathbf{U}'\mathbf{S}'^{1/2}$. For more stable performance, we normalize $\mathbf{B}$ by eliminating the scale effect and only reserving the directions term. Then the final one-step covariance we obtain is $\mathbf{C}\mathbf{U}'\mathbf{U}'^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}$. Since the covariance locates in the space of symmetric matrices, the distance metric can be induced by Frobenius norm.

Adding the covariance terms to the sparse coding objective in Equation (4) with a linear combination, we obtain

$$L(\mathbf{Z}, \mathbb{D}) = \beta L_{mean} + (1-\beta)L_{cov} + \lambda \parallel [\mathbf{Z}]_i \parallel_1, \tag{11}$$

where $L_{mean} = \sum_{i=1}^{N} \parallel \mathbf{V}_i\mathbf{V}_i^{\mathrm{T}} - \sum_{j=1}^{K}\mathbf{Z}_{j,i}\mathbf{D}_j\mathbf{D}_j^{\mathrm{T}} \parallel_F^2$; $L_{cov} = \sum_{i=1}^{N} \parallel \boldsymbol{\Omega}_i - \sum_{j=1}^{K}\mathbf{Z}_{j,i}\bar{\boldsymbol{\Omega}}_j \parallel_F^2$; $\boldsymbol{\Omega}_i$ and $\bar{\boldsymbol{\Omega}}_j$ denote the one-step covariances of the i-th data and the j-th dictionary, respectively; β determines the weights of the trade-off between L_{mean} and L_{cov}. Equation (11) can be reduced to the form similar to Equation 5 for learning the codes.

The dictionary learning problem is reformulated as solving $\min_{\mathbb{D}} \sum_{r=1}^{K}\Gamma(r)$, where

$$\Gamma(r) = \beta\Gamma_{mean}(r) + (1-\beta)\Gamma_{cov}(r). \tag{12}$$

Here, $\Gamma_{mean}(r)$ has been defined in Equation (6); $\Gamma_{cov}(r) = \sum_{i=1}^{N}\sum_{j=1,j\neq r}^{K}\mathbf{Z}_{r,i}\mathbf{Z}_{j,i}\mathrm{Tr}(\bar{\boldsymbol{\Omega}}_r\bar{\boldsymbol{\Omega}}_j) - \sum_{i=1}^{N}\mathbf{Z}_{r,i}\mathrm{Tr}(\bar{\boldsymbol{\Omega}}_r\boldsymbol{\Omega}_i)$. Since $\bar{\boldsymbol{\Omega}}_r = \bar{\mathbf{C}}_r\bar{\mathbf{U}}'_r\bar{\mathbf{U}}'^{\mathrm{T}}_r\bar{\mathbf{C}}_r^{\mathrm{T}}$, $\Gamma_{mean}(r)$ and $\Gamma_{cov}(r)$ are relevant due to the common factor $\bar{\mathbf{C}}_r$. For simplicity and practicability, we get rid of this relevance by reassigning the covariance as $\bar{\boldsymbol{\Omega}}_r = \bar{\mathbf{H}}_r\bar{\mathbf{H}}_r^{\mathrm{T}}$, where $\bar{\mathbf{H}}_r \in \mathbb{R}^{m\times n_v}$ is orthonormal. For data LDS, $\mathbf{H}_i = \mathbf{C}_i\mathbf{U}'_i$; while for dictionary atoms, $\bar{\mathbf{H}}_r$ is independent of $\bar{\mathbf{C}}_r$. In this way, we update $\bar{\mathbf{C}}_r$ and $\bar{\mathbf{A}}_r$ by minimizing $\Gamma_{mean}(r)$ and update $\bar{\mathbf{H}}_r$ by minimizing $\Gamma_{cov}(r)$, separately. With derivations similar to Theorem 3, the optimized $\bar{\mathbf{H}}_r$ is given as the eigenvectors of the matrix $\mathbf{S}_H$ corresponding to the n_v smallest eigenvalues, where $\mathbf{S}_H = \sum_{i=1}^{N}\sum_{j=1,j\neq r}^{K}\mathbf{Z}_{r,i}\mathbf{Z}_{j,i}\bar{\mathbf{H}}_j\bar{\mathbf{H}}_j^{\mathrm{T}} - \sum_{i=1}^{N}\mathbf{Z}_{r,i}\mathbf{H}_i\mathbf{H}_i^{\mathrm{T}}$. It is easy to develop the algorithm for learning the covariance-involved dictionary. We only need to revise Algorithm 1 by computing the sparse codes with Equation (11) instead and adding the update of $\bar{\mathbf{H}}_r$ for each atom.

6. Computational Complexity

For sparse coding (Equation (11)), the key is the kernel-matrix computation. For each kernel, we need to perform Cholesky decomposition, solve the Lyapunov Equation and calculate the matrix multiplication, which scale $O(n^3)$, $O(n^3)$ and $O(mn^2)$, respectively. Recalling that $n \ll m$, all these computations scale $O(mn^2)$. The number of kernels between dictionary atoms and that between dictionary and data are K^2 and NK, respectively. Thus, the total complexity of sparse coding is $O((NK + K^2)mn^2)$.

For each subproblem of dictionary learning (Equation (12)), we primarily need to calculate the matrix $\mathbf{S}(r,k)$ and

find the smallest eigenvector of $\mathbf{W}^{\mathrm{T}}\mathbf{S}(r,k)\mathbf{W}$ for minimizing $\Gamma_{mean}(r)$; calculate the matrix $\mathbf{S}_H$ and find its n_v smallest eigenvectors for minimizing $\Gamma_{cov}(r)$. Computing $\mathbf{S}(r,k)$ and $\mathbf{S}_H$ scales $O(K(N+nm^2)+\gamma nm^2)$, where γ denotes the number of non-zero members in the r-th row of $\mathbf{Z}$. We apply the Grassmannian-based Conjugate Gradient Method [12] to find the smallest eigenvector of $\mathbf{W}^{\mathrm{T}}\mathbf{S}(r,k)\mathbf{W}$, which has a computational cost of $O(m^2)$. This operation needs to be repeated for n times until we have all the columns of $\bar{\mathbf{C}}_r$ updated. Thus solving the eigenvector problem costs $O(nm^2)$ in total. Similarly, finding the the n_v smallest eigenvectors of $\mathbf{S}_H$ scales $O(n_v m^2)$. To sum up, the computation complexity of updating one dictionary atom adds up to $O(K(N+nm^2)+\gamma nm^2)$.

As shown in Equation (3), the finite-approximation method [21] employs the L-order observability $\mathbf{O}(n,L) \in \mathbb{R}^{Lm\times n}$ as the representations of the data and dictionary LDSs. With the analysis similar to our models, the computation complexities of the finite method are found to be $O(L(NK+K^2)mn^2)$ for sparse coding and $O(K(N+nL^2m^2)+\gamma nL^2m^2)$ for updating one dictionary atom, respectively. Compared to our infinite models, the finite method scales poorly specially when L is large; we will further demonstrate this in the experimental section.

7. Experiments

We evaluate our proposed models on two groups of experiments in this section. For the first group, we compare the performance of our sparse coding algorithms with state-of-the-art methods on several benchmark datasets. For the second group, we evaluate the effectiveness of the dictionary learning method. For sake of consistency, we hereafter denote sparse coding on LDSs with arbitrary transition matrices (Section 3) as LDS-SC, sparse coding on LDSs with symmetric transition matrices (Section 4) as LDSST-SC, the LDSST-SC model combining the state covariance (Section 5) as covLDSST-SC, the dictionary learning algorithm (Section 4) as LDSST-DL, LDSST-DL considering the the state covariance (Section 5) as covLDSST-DL. For the compared models, the basic LDS model [34, 6] where the Martin distance is applied is denoted as LDS-Martin; sparse coding and dictionary learning on finite Grassmannian [21] are denoted as gLDS-SC and gLDS-DL, respectively. All experiments are carried out with Matlab 8.1.0.604 (R2013a) on Intel Core i7, 2.90-GHz CPU with 8-GB RAM.

7.1. Benchmark datasets

A variety of datasets are applied in our experiments, including the hand gesture dataset *Cambridge* [25], the traffic scene analysis dataset *UCSD* [5], the face emotion recognition dataset *CK+* [27], the dynamic texture recognition dataset *DynTex++* [20], and three tactile recognition datasets *SD* [28], *SPR* [28] and *BDH* [42]. For *Cambridge* and *UCSD*, the image sequences are treated as the input. The images in *Cambridge* are resized to 20×20 pixels as suggested by [25]. For *DynTex++*, we utilize the histogram of LBP from Three Orthogonal Planes (LBP-TOP) [45] by splitting each video into sub-videos of length 8, with a 6-frame overlap. For *CK+*, the input are the extracted 68-landmark of face images. For *SD*, *SPR*, and *BDH*, the tactile series obtained from the array sensors on the robot hands are used to recognize the objects that the robot hands are grasping. Thus, the input are the force values recorded in the sensor arrays along the time axis. We apply the suggested divisions of the training set and testing set by previous works on all datasets except *CK+*. Specifically, on *Cambridge*, the first 80 videos of each class are used for testing while the remaining 20 for training [22]. On *UCSD*, four random divisions have been performed by the authors in [5]. In each division, 75% of the sequences are utilized for training and the rest 25% for testing. On *DynTex++*, the training and testing data are generated with a random fifty-fifty division of the dataset over 20 trials [20]. The three tactile datasets, *i.e.* *SD*, *SPR*, and *BDH*, are split randomly into the training and testing sets with a ratio of $9:1$ over 10 trials [28, 42]. For *CK+*, the authors in [15] employed the leave-one-out cross-validation scheme. Here, we perform a more challenging division by applying half of the dataset for training while the remaining for testing. For reader's convenience, we illustrate some samples in Figure 1. The details of the datasets are presented in the supplement material.

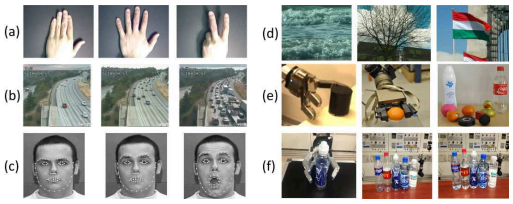

Figure 1. Samples of the benchmark datasets: (a) *Cambridge*; (b) *UCSD*; (c) *CK+*; (d) *DynTex++*; (e) *SD* and *SPR*; (f) *BDH*.

7.2. Sparse coding

In this section, the training samples are considered to be the dictionary atoms without dictionary learning; and the reconstruction error approach presented in the the supplementary material is adopted for classification.

Comparison with the state-of-the-arts. We compare the proposed sparse coding methods, *i.e.* LDS-SC, LDSST-SC and covLDSST-SC, with models that achieved competitive results on *Cambridge*, *UCSD*, *CK+*, *SD*, *SPR*, and *BDH*. We also implement the LDS-Martin model as a referenced baseline, where the Nearest-Neighbor (NN) method is utilized as the classifier. For proposed models and LDS-Martin, we vary the value of n and report the best results. Additionally for covLDSST-SC, the parameter n_v is fixed to be 4 and the weight β is selected from $\{0.8, 0.6, 0.2\}$. Table 1 reports the classification results. We first note that the best

Table 1. Averaged classification accuracies of the proposed sparse coding methods compared with the state-of-the-arts.

Datasets	References	LDS-Martin	Proposed models					
			Our best	LDS-SC	LDSST-SC	covLDSST-SC ($n_v = 4$)		
						$\beta = 0.8$	$\beta = 0.6$	$\beta = 0.2$
Cambridge	90.7 [22], 83.05 [29]	88.3	**91.7**	91.7	85.7	85.7	86.8	90.3
UCSD	**95.0** [5], 87.8 [35]	92.9	93.3	93.3	89.4	89.8	90.2	93.3
CK+	83.7 [15], 76.0 [13]	77.3	**86.7**	84.5	85.4	86.7	86.5	86.3
SD	97 [28], 92 [10]	95	**100**	98	98	98	98	100
SPR	91 [28], 89 [10]	95	**97**	96	95	97	97	97
BDH	87 [42]	98	**100**	100	99	100	100	98

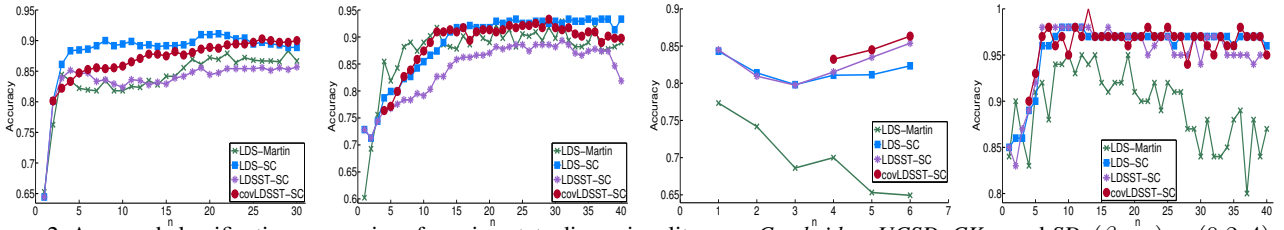

Figure 2. Averaged classification accuracies of varying state dimensionality n on *Cambridge*, *UCSD*, *CK+*, and *SD*. $(\beta, n_v) = (0.2, 4)$.

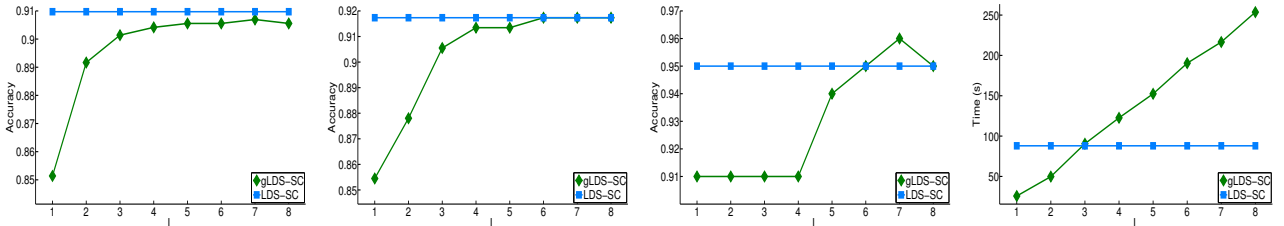

Figure 3. Comparisons between LDS-SC and gLDSs. The first three figures display the averaged classification accuracies on *Cambridge*, *UCSD* and *SPR*. The fourth figure demonstrates the training time of the compared models on *UCSD*. $n = 10$.

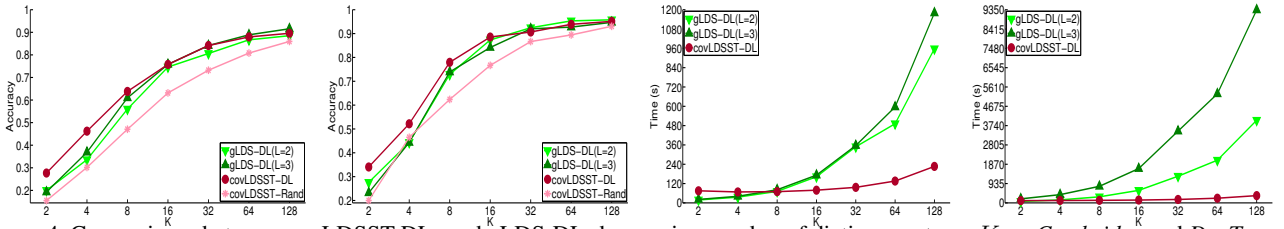

Figure 4. Comparisons between covLDSST-DLs and gLDS-DLs by varying number of dictionary atoms K on *Cambridge* and *DynTex++*. The left two figures show the averaged accuracies, while the right two ones display the training time. $(n, \beta, n_v) = (10, 0.2, 4)$.

results of proposed models outperform all compared models on all datasets except *UCSD*. On *UCSD*, the method proposed in [5] achieves the best performance due to its highly-complicated distance; LDS-SC obtains a comparable accuracy while its complexity to calculate the distance is much simpler. LDSST-SC is found to be worse than LDS-SC as a whole, because the symmetric constraint to transition matrices could limit the modeling ability. With an appropriate β, covLDSST-SC can promote the performance of LDSST-SC significantly, and even outperform LDS-SC in some cases, thus verifying the effectiveness of the state covariance on enhancing the modeling ability of LDSST.

Varying n. To evaluate the sensitivity of the hidden dimensionality n to the eventual performance, we vary the value of n and report the classification results of LDS-Martin, LDS-ST, LDSST-SC, and covLDSST-SC on *Cambridge*, *UCSD*, *CK+*, and *SD*. Figure 2 demonstrates that, the LDS-Martin model performs consistently on *Cambridge* and *UCSD* but much worse on *CK+* and *SD*. Our models perform consistently on all datasets after n grows beyond a certain value. The model covLDSST-SC can constantly improve the performance of LDSST-SC, which once again validates the importance of the state covariance to the performance of covLDSST-SC.

Comparison with the finite method. As clarified in Section 3, the model LDS-SC is an infinite generalization of the finite-approximation method gLDS-SC. Thus, we are interested in the asymptotical behavior of gLDS-SC when the observability order L increases. For this purpose, we carry out experiments on *Cambridge*, *UCSD* and *SPR*. As expected, the classification accuracy of gLDS-SC finally converges to that of LDS-SC when L increases, which is

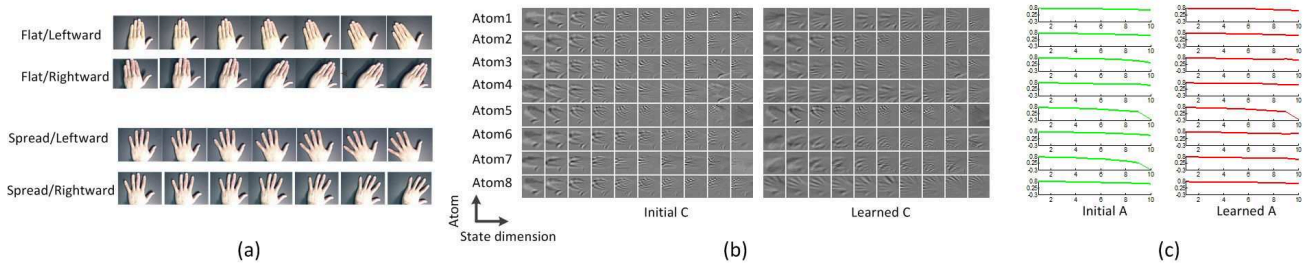

Figure 5. The visualization of the initial and the learned dictionaries on *Cambridge*. $(n, \beta, n_v) = (10, 0.2, 4)$. (a) Samples of the 4 sub-categories in *Cambridge*. (b) Visualization of $\mathbf{C}$: rows corresponds to atoms and columns to the state dimensions. (c) Plots of $\mathbf{A}$: different plots display the values of the transition eigenvalues of different atoms.

illustrated in Figure 3. In Section 6, we have shown that the computational complexity of gLDS-SC ($O(L(NK + K^2)mn^2)$) is L times of LDS-SC ($O((NK + K^2)mn^2)$). Larger L will cause more computational cost of gLDS-SC. On the dataset *UCSD*, for example, gLDS-SC needs more training time than LDS-SC when $L > 3$, as demonstrated in Figure 3. Since LDS-SC additionally requires Cholesky decomposition and Lyapunov equation derivation, it performs more slowly than gLDS-SC when $L < 3$.

7.3. Dictionary learning

As demonstrated by the experimental results in the last section, taking the state covariance term into the algorithm formulation can further improve the performance. Thus, in this section, we implement covLDSST-DL instead of LDSST-DL to perform comparison with other methods. The dictionary atoms are initialized randomly. The codes of training and testing systems with respect to the learned dictionary are fed to a linear SVM [14] for classification.

Learning effectiveness analysis. To verify the effectiveness of covLDSST-DL, we also test the baseline model, *i.e* covLDSST-Rand, in which the dictionary atoms are chosen from the training set randomly and no dictionary learning is involved. Besides, we implement the finite-approximation method gLDS-DL with $L = 2, 3$. For fair comparison, we use the same classifier (linear SVM) and the same value of n ($n = 10$), for covLDSST-DL, covLDSST-Rand and gLDS-DL. Experiments are carried out on *Cambridge* and *DynTex++*. On *Cambridge*, we apply the first half sequences of each class for learning the dictionary while the rest are for testing. On *DynTex++*, We evaluate the performance of the compared models on a 9-classes subset. In particular, we select the videos of the first 9 classes from the original dataset, thus constructing a smaller dataset with 900 videos in total. Half of the videos are used for learning and the others for testing. Figure 4 shows that covLDSST-DL consistently outperforms covLDSST-Rand under the varying number of the dictionary atoms. Compared to gLDS-DLs, covLDSST-DL achieves higher accuracies when the dictionary size K is small (*e.g.* $K < 16$), and obtains equivalent performance when K is large. As discussed in Section 5, the computational complexity of gLDS-DL is higher than covLDSST-DL. We also display the training time of gLDS-DLs and covLDSST-DL in Figure 4. Obviously, gLDS-DLs become much computationally expensive as K increases. Our covLDSST-DL performs scalably even with a large K. In addition to the 9-classes subset, we also evaluate covLDSST-DL on original *DynTex++*. The model covLDSST-DL reaches a recognition rate of 92.0% when $K = 516$, which is comparable to that of the Grassmannian-kernel-based dictionary learning method [23], *i.e.* 92.8%.

Dictionary visualization. The model covLDSST-DL is capable of learning the dictionary measurement matrix $\mathbf{C}$ and the transition matrix $\mathbf{A}$, explicitly and separately. Thus, we can visualize the learned pairs $(\mathbf{A}, \mathbf{C})$ to demonstrate what patterns they have discovered. For simplicity, we perform covLDSST-DL on the 4-class subset of *Cambridge*, *i.e.* Flat\Leftward, Flat\Rightward, Spread\Leftward, and Spread\Rightward. Dictionary atoms are initialized randomly by choosing 8 videos from one single class: Flat\Leftward. Figure 5 (a) visualizes both the initial and the learned pairs. Clearly, more spatial patterns such as the spread-hand shape and the hand-rightward state, have been discovered by the learned $\mathbf{C}$. There are also slight changes in $\mathbf{A}$ after the learning. The transition matrices of different atoms have a small difference, indicating that the dynamic within each dictionary is similar to each other, presumably because the movement speed of the hand and the sampling frequency of the camera keep almost consistent.

8. Conclusion

In this paper, we address the challenging issue about performing sparse coding and dictionary learning on the true space of LDSs that is formulated as an infinite Grassmannian. Compared to the finite-approximation methods, the proposed models are not only theoretically beneficial but also computationally efficient. In addition, we combine the state covariance into the model formulation, thus further improving the performance significantly. The effectiveness of our models is verified by various experiments on different tasks including hand gesture recognition, dynamical scene classification, face emotion recognition, dynamic texture categorization and tactile recognition.

References

[1] B. Afsari, R. Chaudhry, A. Ravichandran, and R. 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. 1

[2] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[3] C. M. Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006. 1

[4] L. Bo, X. Ren, and D. Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2115–2123, 2011.

[5] A. B. Chan and N. 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. 1, 5, 6, 7

[6] A. B. Chan and N. Vasconcelos. Classifying video with kernel dynamic textures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6. IEEE, 2007. 1, 2, 6

[7] K. De Cock and B. De Moor. Subspace angles between ARMA models. *Systems & Control Letters*, 46(4):265–270, 2002. 2

[8] D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008. 4

[9] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision (IJCV)*, 51(2):91–109, 2003. 1, 2

[10] A. Drimus, G. Kootstra, A. Bilberg, and D. Kragic. Design of a flexible tactile sensor for classification of rigid and deformable objects. *Robotics and Autonomous Systems*, 62(1):3–15, 2014. 7

[11] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[12] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. 6

[13] M. T. Eskil and K. S. Benli. Facial expression recognition based on anatomy. *Computer Vision and Image Understanding*, 119:1–14, 2014. 7

[14] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research (JMLR)*, 9:1871–1874, 2008. 8

[15] X. Fan and T. Tjahjadi. A spatial-temporal framework based on histogram of gradients and optical flow for facial expression recognition in video sequences. *Pattern Recognition*, 2015. 6, 7

[16] C. Gan, M. Lin, Y. Yang, Y. Zhuang, and A. G. Hauptmann. Exploring semantic inter-class relationships (sir) for zero-shot action recognition. In *Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, 2015. 1

[17] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. 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 (CVPR)*, pages 2568–2577, 2015. 1

[18] C. Gan, Y. Yang, L. Zhu, D. Zhao, and Y. Zhuang. Recognizing an action using its name: A knowledge-based approach. *International Journal of Computer Vision (IJCV)*, pages 1–17, 2016. 1

[19] S. Gao, I. W.-H. Tsang, and L.-T. Chia. Kernel sparse representation for image classification and face recognition. In *European Conference on Computer Vision (ECCV)*, pages 1–14. Springer, 2010. 4

[20] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In *European Conference on Computer Vision (ECCV)*, pages 223–236. Springer, 2010. 6

[21] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson. Extrinsic methods for coding and dictionary learning on Grassmann manifolds. *International Journal of Computer Vision (IJCV)*, 114(2):113–136, 2015. 1, 2, 3, 6

[22] M. Harandi and M. Salzmann. Riemannian coding and dictionary learning: Kernels to the rescue. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 6, 7

[23] M. Harandi, C. Sanderson, C. Shen, and B. Lovell. Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3120–3127. IEEE, 2013. 1, 8

[24] J. Ho, Y. Xie, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *International Conference on Machine Learning (ICML)*, pages 1480–1488, 2013.

[25] T.-K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 31(8):1415–1428, 2009. 6

[26] H. Liu and F. Sun. Hierarchical orthogonal matching pursuit for face recognition. In *Asian Conference on Pattern Recognition (ACPR)*, pages 278–282. IEEE, 2011.

[27] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 94–101. IEEE, 2010. 6

[28] M. Madry, L. Bo, D. Kragic, and D. Fox. St-hmp: Unsupervised spatio-temporal feature learning for tactile data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2262–2269. IEEE, 2014. 1, 6, 7

[29] A. Mahmood, A. Mian, and R. Owens. Semi-supervised spectral clustering for image set classification. In *IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 121–128. IEEE, 2014. 7

[30] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008. 1

[31] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *Advances in neural information processing systems (NIPS)*, pages 1033–1040, 2009. 1

[32] R. J. Martin. A metric for ARMA processes. *IEEE Transactions on Signal Processing*, 48(4):1164–1170, 2000. 2

[33] A. Ravichandran, R. Chaudhry, and R. Vidal. Categorizing dynamic textures using a bag of dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(2):342–353, 2013. 1, 2

[34] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto. Dynamic texture recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–58. IEEE, 2001. 1, 2, 6

[35] A. C. Sankaranarayanan, P. K. Turaga, R. Chellappa, and R. G. Baraniuk. Compressive acquisition of linear dynamical systems. *SIAM Journal on Imaging Sciences*, 6(4):2109–2133, 2013. 7

[36] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *Journal of time series analysis*, 3(4):253–264, 1982. 2

[37] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(11):2273–2286, 2011. 1, 2

[38] P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994. 2

[39] R. Vidal and P. Favaro. Dynamicboost: Boosting time series generated by dynamical systems. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–6. IEEE, 2007. 1

[40] S. Vishwanathan, A. J. Smola, and R. 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. 1

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

[42] J. Yang, H. Liu, F. Sun, and M. Gao. Tactile sequence classification using joint kernel sparse coding. In *International Joint Conference on Neural Networks (IJCNN)*, 2015. 6, 7

[43] M. Yang, D. Zhang, and J. Yang. Robust sparse coding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 625–632. IEEE, 2011. 1

[44] K. Ye and L.-H. Lim. Distance between subspaces of different dimensions. *arXiv preprint*, 2014. 2, 3

[45] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007. 6