Human Action Recognition Using Tensor Dynamical System Modeling

Chan-Su Lee
Dept. of Electronic Engineering, Yeungnam University
280 Daehak-ro, Gyeongsan-si, Gyeongbuk-do, 38541, Rep. of Korea
chansu@ynu.ac.kr

Abstract

This paper presents a new framework for human action classification using a tensor dynamical model of human action from 3-dimensional (3D) volume sequences and distance measurement on Grassmann manifold. The tensor dynamical model is an extension of linear dynamical models for multi-dimensional sequence analysis. Each subdimensional linear dynamic model is estimated from tensor sequences using an iterative expectation-maximization (EM) algorithm after projection of tensor sequence to each dimensional axis. The combination of distances on Grassmann manifold of linear dynamic systems in each dimension of the tensor dynamic model provides similarity measurement between two tensor dynamical systems. The proposed approach can be applied to 3D depth or convex hull data as well as 2D video image sequences. Experimental results show good performance in human action recognition from INRIA multiview human action database.

1. Introduction

Analysis of temporal sequence data has many potential applications such as human motion analysis, video action recognition, dynamic texture analysis and biological sequence analysis and so on. In temporal sequence analysis, one of the key characteristics is modeling dynamics of the sequence. Hidden Markov models [19] and its extensions [10] are frequently used for the analysis of temporal sequence and human motion analysis. Dynamic Bayesian network may provides general framework to model sequence data interpretation. Recently, nonlinear manifold based approaches are also applied for the analysis of high dimensional sequence data in low dimensional space [11].

Many sequence data are multi-dimensional. Video sequence data for gestures and action recognition are two dimensional in addition to one temporal dimension. Multichannel EEG signals may have more than 20 channels. Recently multi-dimensional tensor data are being generated in a wide range of emerging applications. However, most of the current solutions are based on one dimensional vector analysis. Usually, a high-dimensional vector must estimate a large number of parameters, and also destroy the natural structure and correlation in the original data. Multidimensional tensor subspace analysis can learn more compact and useful representations than conventional linear subspace learning. Then, how can we approach multidimensional dynamic sequence data?

In this paper, we present an extension of normal distribution to the multivariate normal distribution, which can provide a foundation to extend conventional linear dynamical systems (LDSs) into tensor dynamical systems (TDS). For the metrics of the LDSs, recently, geometric manifold spaces [3] such as Grassmann manifold are used [20]. This paper extends the model to the tensor dynamical system. Preliminary experimental results of human action recognition using tensor dynamical system show the potential of the proposed approach in multidimensional dynamic sequence data analysis.

2. Frameworks: Tensor Dynamical Models

We present tensor dynamical models as an extension of linear and multivariate dynamic model to multi-dimensional sequence data using tensor normal distribution and parameter estimation using Expectation maximization (EM) algorithms. We first define basic tensor algebra for the simplification of the notation. Then we explain normal distribution and multivariate normal distribution to introduce tensor normal distribution. The tensor normal distribution is used for the basis for tensor dynamic model and its parameter estimation.

2.1. Tensor representation and operator

Let $\mathbb{N}$ be the set of all positive integers and $\mathbb{R}$ be the set of all real numbers. Given $I \in \mathbb{N}^M$, where $M \in \mathbb{N}$, we generate a tensor-product space $\mathbb{R}^{I_1 \times \cdots \times I_M}$. Then a tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ is an element of a tensor product space. A tensor $\boldsymbol{\mathcal{Y}}$ may be referenced by either a full vector $(i_1, \cdots, i_M)$ or a by subvector, using the $\bullet$ symbol to indicate coordinate that are not fixed [18].

The factorization of a tensor $\mathcal{Z} \in \mathbb{R}^{IJ}$ is given by $A_{i_1\cdots i_M j_1\cdots j_M} = \prod_{m=1}^{M} A^{(m)}_{i_m j_m}$, where $A^{(m)} \in \mathbb{R}^{I_m \times J_m}$ for all m. In matrix form, we have $mat(\mathcal{A}) = A^{(M)} \otimes A^{(M-1)} \otimes \cdots \otimes A^{(1)}$, where $\otimes$ is the Kronecker matrix product [14, 15]. Vectorization $vec(\mathcal{X})$ is obtained by shaping the tensor into a vector. In case of matrix $\boldsymbol{A} = (\boldsymbol{a}_1, \cdots, \boldsymbol{a}_q)$ be $p \times q$ matrix, $vec(A)$ forms a column vector whose size is $p \times q$ by column stacking.

The product $\mathcal{A} \circledast \mathcal{X}$ of two tensors $\mathcal{A} \in \mathbb{R}^{IJ}$ and $\mathcal{X} \in \mathbb{R}^{J}$, where $I \in \mathbb{N}^M$, $J \in \mathbb{N}^M$, and $M \in \mathbb{N}$, is given by $(\mathcal{A} \circledast \mathcal{X})_{i_1\cdots i_M} = \sum_{j_1\cdots j_M} \mathcal{A}_{i_1\cdots i_M j_1\cdots j_M} \mathcal{X}_{j_1\cdots j_M}$. The product is only defined if the dimensionalities of the last M modes of $\mathcal{A}$ match the dimensionalities of $\mathcal{X}$. The vectorization of the tensor product can be represented by generalization of the standard matrix-vector product as follows [18]:

$$vec(\mathcal{A} \circledast \mathcal{X}) = mat(\mathcal{A}) vec(\mathcal{X}). \tag{1}$$

2.2. Tensor normal distribution

The univariate standard normal distribution U is defined by

$$f_U(u) = (2\phi)^{-1/2} e^{-\frac{1}{2}u^2}, -\infty < u < \infty, \tag{2}$$

and denoted $U \sim N(0,1)$. Its expectation is $E[U] = 0$ and its variance is $D[U] = 1$. General univariate normal distribution with mean μ and variance $\sigma > 0$ can be defined using univariate standard normal distribution U as follows:

$$\mu + \sigma U, \sigma > 0, -\infty < u < \infty, \tag{3}$$

It can be explicitly defined with a density function

$$f_X(x) = (2\phi\sigma^2)^{-1/2} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}, -\infty < u, x < \infty, \sigma > 0 \tag{4}$$

and dented $X \sim N(\mu, \sigma^2)$.

Multivariate standard normal distribution can be described using an extension of normal distribution in vector form. p dimensional multivariate standard normal vector $\boldsymbol{u} = (U_1, U_2, \cdots, U_p)$ consists of p independent identically distributed (i.i.d.) $N(0,1)$ elements. Then p dimensional multivariate standard normal density function can be defined from Eq. 2

$$f_{\boldsymbol{u}}(\boldsymbol{u}) = (2\pi)^{-\frac{1}{2}p} e^{-\frac{1}{2}tr(\boldsymbol{u}\boldsymbol{u}')}, \tag{5}$$

and we say that $\boldsymbol{u} \sim N_p(\boldsymbol{0}, \boldsymbol{I})$, where $tr(\boldsymbol{u}\boldsymbol{u}') = \boldsymbol{u}'\boldsymbol{u}$. General p dimensional multivariate normal distribution with mean $E[\boldsymbol{x}] = \boldsymbol{\mu}$ and dispersion $D[\boldsymbol{x}] = \boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is non-negative definite and can be represented by $\boldsymbol{\Sigma} = \boldsymbol{\tau}\boldsymbol{\tau}'$ with full rank, $r(\boldsymbol{\tau}) = p$ [12]

$$\boldsymbol{\mu} + \boldsymbol{\tau}\boldsymbol{u}, \tag{6}$$

where $\boldsymbol{u} \sim N_p(\boldsymbol{0}, \boldsymbol{I})$ and the distribution of $\boldsymbol{x}$ is denoted by $\boldsymbol{x} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

$$f_x(\boldsymbol{x}) = (2\phi)^{-\frac{1}{2}p} |\boldsymbol{\Sigma}|^{-1/2} e^{-\frac{1}{2}tr\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{u})(\boldsymbol{x}-\boldsymbol{u})'\}}. \tag{7}$$

Normal distribution for tensor-valued random variables can be described by an extension of vector valued multivariate random variables into multi-dimensional random variables similar to an extension of vector valued data into tensor analysis using multilinear algebra. A matrix normal distribution, which is an extension of one dimensional vector-based multivariate normal distribution to two dimensional one, can be described by an extension of the vector version using a *bilinear* extension. When a matrix $\boldsymbol{X} : p \times n$ is matrix normally distributed with parameter $\boldsymbol{\mu}, \boldsymbol{\Sigma} = \boldsymbol{\tau}\boldsymbol{\tau}'$ and $\boldsymbol{\Psi} = \boldsymbol{\gamma}\boldsymbol{\gamma}'$, its distribution can be described as

$$\boldsymbol{\mu} + \boldsymbol{\tau}\boldsymbol{U}\boldsymbol{\gamma}', \tag{8}$$

where $\boldsymbol{\mu} : p \times n$ is non-random and $\boldsymbol{U} : r \times s$ consists of s i.i.d $N_r(\boldsymbol{0}, \boldsymbol{I})$ vectors $\boldsymbol{U}_i, i = 1, 2, \cdots, s$, $\boldsymbol{\tau} : p \times r$ and $\boldsymbol{\gamma} : n \times s$. The matrix normally distributed $\boldsymbol{X}$ will be denoted $\boldsymbol{X} \sim N_{p,n}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Psi})$. Since $vec\boldsymbol{X}$ and $\boldsymbol{X}$ have the same distribution, $\boldsymbol{X}$ has the same distribution as

$$vec(\boldsymbol{\mu}) + (\boldsymbol{\gamma} \otimes \boldsymbol{\tau}) vec(\boldsymbol{U}). \tag{9}$$

The matrix normally distributed $\boldsymbol{X} \sim N_{p,n}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Psi})$ means the same as $vec(\boldsymbol{X}) \sim N_{pn}(vec(\boldsymbol{\mu}), \boldsymbol{\Psi} \otimes \boldsymbol{\Sigma})$. Since the expectation of $\boldsymbol{U}$ equals zero, $E[\boldsymbol{X}] = \boldsymbol{\mu}$, and since by definition of the dispersion matrix $D[\boldsymbol{X}] = D[vec(\boldsymbol{X})]$, we obtain that $D[\boldsymbol{X}] = \boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}$. The density of $N_{p,n}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Psi})$ is given by

$$f_{\boldsymbol{X}}(X) = (2\phi)^{-\frac{1}{2}pn} |\boldsymbol{\Sigma}|^{-n/2} |\boldsymbol{\Psi}|^{-p/2} e^{-\frac{1}{2}tr\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\boldsymbol{u})\boldsymbol{\Psi}^{-1}(\boldsymbol{X}-\boldsymbol{u})'\}}, \tag{10}$$

where $vec(\boldsymbol{X})'(\boldsymbol{\Psi} \otimes \boldsymbol{\Sigma})^{-1} vec(\boldsymbol{X}) = tr(\boldsymbol{\Sigma}^{-1}\boldsymbol{X}\boldsymbol{\Psi}^{-1}\boldsymbol{X}')$ and $|\boldsymbol{\Psi} \otimes \boldsymbol{\Sigma}| = |\boldsymbol{\Psi}|^p |\boldsymbol{\Sigma}|^n$.

The bilinear normal distribution can be expressed using unit basis vectors

$$\begin{aligned} \textstyle\sum_{ij} \quad & X_{ij}\boldsymbol{e}^1_i(\boldsymbol{e}^2_j)' \\ = \quad & \sum_{ij} \mu_{ij}\boldsymbol{e}^1_i(\boldsymbol{e}^2_j)' + \sum_{ij}\sum_{km} \tau_{ik}\gamma_{mj}U_{km}\boldsymbol{e}^1_i(\boldsymbol{e}^2_j)' \\ = \quad & \sum_{ij} \mu_{ij}\boldsymbol{e}^2_j \otimes \boldsymbol{e}^1_i + \sum_{ij}\sum_{km} \tau_{ik}\gamma_{mj}U_{km}\boldsymbol{e}^2_j \otimes \boldsymbol{e}^1_i, \end{aligned} \tag{11}$$

where $\boldsymbol{e}^1_i : p \times 1$, $\boldsymbol{e}^2_j : n \times 1$ are the unit basis vectors, $U_{km} \sim N(0,1)$, and $\boldsymbol{e}^1_i(\boldsymbol{e}^2_j)' \rightarrow \boldsymbol{e}^2_j \otimes \boldsymbol{e}^1_i$.

Tensor normal distribution can be expressed using the basis vectors similar to the bilinear normal distribution. A matrix $\boldsymbol{X}$ is tensor normal distribution of with order k, $\boldsymbol{X} \sim N_{p_1,p_2,\cdots,p_k}(\boldsymbol{\mu}, \sum_k, \sum_1, \sum_2, \cdots, \sum_{k-1})$, if

$$\begin{aligned} \textstyle\sum_{i_1,i_2,\cdots,i_k} \quad & X_{i_1,\cdots,i_k}\boldsymbol{e}^1_{i_1} \otimes \boldsymbol{e}^2_{i_2} \otimes \cdots \otimes \boldsymbol{e}^k_{i_k} \\ = \textstyle\sum_{i_1,i_2,\cdots,i_k} \quad & u_{i_1,\cdots,i_k}\boldsymbol{e}^1_{i_1} \otimes \boldsymbol{e}^2_{i_2} \otimes \cdots \otimes \boldsymbol{e}^k_{i_k} \\ + \textstyle\sum_{i_1}^{p_1}\sum_{j_1}^{p_1} \quad & \sum_{j_2}^{p_2} \cdots \sum_{j_k}^{p_k} \tau^1_{i_1 j_1}\tau^2_{i_2 j_2} \\ \cdots \tau^k_{i_k j_k} \quad & U_{j_1 j_2 \cdots j_k}\boldsymbol{e}^1_{i_1} \otimes \boldsymbol{e}^2_{i_2} \otimes \cdots \boldsymbol{e}^k_{i_k}, \end{aligned} \tag{12}$$

where $\sum_i = \boldsymbol{\tau}^i(\boldsymbol{\tau}^1)'$, $\boldsymbol{e}^r_{i_r} : p_r \times 1$ and $U_{j_1 j_2 \cdots j_k} \sim N(0,1)$. The tensor normal distribution can be represented in coordinate free form as follows:

$$X_{i_1,\cdots,{}_{1_k}} = \mu_{i_1,\cdots,i_k} + \sum_{j_1 j_2 \cdots j_k} \tau^1_{i_1 j_1} \tau^2_{i_2 j_2} \cdots \tau^k_{i_k j_k} U_{j_1 j_2 \cdots j_k}. \tag{13}$$

2.3. Tensor dynamical models

Tensor dynamical systems, or multilinear dynamical systems, are an extension of linear dynamical systems for multi-dimensional data using tensor normal distributions. Linear dynamical systems (LDS) can be represented using state-space models as follows:

$$\boldsymbol{x}_{n+1} = \boldsymbol{A}\boldsymbol{x}_n + \boldsymbol{w}, \boldsymbol{y}_n = \boldsymbol{C}\boldsymbol{x}_n + \boldsymbol{v}, \tag{14}$$

where $\boldsymbol{x}_n$ is state space at frame time n, output $\boldsymbol{y}_n$ is linear function of the state $\boldsymbol{x}_n$, $\boldsymbol{A}$ is a state transition matrix, $\boldsymbol{C}$ is an observation matrix, $\boldsymbol{w}_n$ is a normally distributed zero-mean random state noise with variance $\boldsymbol{Q}$, and $\boldsymbol{v}_n$ is a normally distributed zero-mean output noise with variance $\boldsymbol{R}$. The conditional densities for the state and output can be represented from the normally distributed models as follows:

$$\boldsymbol{x}_n | \boldsymbol{x}_{n-1} \sim N_k(\boldsymbol{x}_n | A\boldsymbol{x}_{n-1}, \boldsymbol{Q}), \tag{15}$$

$$\boldsymbol{y}_n | \boldsymbol{x}_n \sim N_p(\boldsymbol{y}_n | C\boldsymbol{x}_n, \boldsymbol{R}), \tag{16}$$

where p is the dimensional of output $\boldsymbol{y}_n$ and k is the dimension of the state space $\boldsymbol{x}_n$. Expectation Maximization (EM) algorithm can be used to estimate parameters of LDS from observation sequences [4, 9]. The E step computes the expected log likelihood using Kalman filter forward and backward recursions. The M step updates parameters by taking the corresponding partial derivative of the expected log likelihood [9].

Tensor time series consist of a sequence tensor $\boldsymbol{\mathcal{Y}}_{1,\cdots,N} = [\boldsymbol{\mathcal{Y}}_1, \cdots, \boldsymbol{\mathcal{Y}}_N]$, where $\boldsymbol{\mathcal{Y}}_n \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ for all n. Tensor dynamical models seek sequence of latent tensors $\boldsymbol{\mathcal{X}}_{1,\cdots,N} = [\boldsymbol{\mathcal{X}}_1, \cdots, \boldsymbol{\mathcal{X}}_N]$, where $\boldsymbol{\mathcal{X}}_n \in \mathbb{R}^{J_1 \times \cdots \times J_M}$ for all t. Each latent tensor $\boldsymbol{\mathcal{X}}_n$ emits an observation $\boldsymbol{\mathcal{Y}}_n$ with tensor transition $\boldsymbol{\mathcal{A}}$, and projection tensor $\boldsymbol{\mathcal{C}}$. For a given $\boldsymbol{\mathcal{X}}_n$, $1 \leq n \leq N-1$, $\boldsymbol{\mathcal{X}}_{n+1}$ can be generated according to the conditional distribution

$$\boldsymbol{\mathcal{X}}_{n+1} | \boldsymbol{\mathcal{X}}_n \sim \mathcal{N}(\boldsymbol{\mathcal{A}} \circledast \boldsymbol{\mathcal{X}}_n, \boldsymbol{Q}), \tag{17}$$

where $\boldsymbol{Q}$ is the conditional covariance shared by all $\boldsymbol{\mathcal{X}}_n$ and $\boldsymbol{\mathcal{A}}$ is the *transition tensor* which describes the dynamics of the evolving sequence $\boldsymbol{\mathcal{X}}_n$. When the transition tensor $\boldsymbol{\mathcal{A}}$ is factorized into M matrices $\boldsymbol{A}^{(m)}$, each of which acts on a mode of $\boldsymbol{\mathcal{X}}_n$. To each $\boldsymbol{\mathcal{X}}_n$ there corresponds an observation $\boldsymbol{\mathcal{Y}}_n$ generated by

$$\boldsymbol{\mathcal{Y}}_n | \boldsymbol{\mathcal{X}}_n \sim \mathcal{N}(\boldsymbol{\mathcal{C}} \circledast \boldsymbol{\mathcal{X}}_n, \boldsymbol{\mathcal{R}}), \tag{18}$$

where $\boldsymbol{\mathcal{R}}$ is the covariance shared by all $\boldsymbol{\mathcal{Y}}_n$.

2.4. Parameter estimation

Given sequence of observations $\boldsymbol{\mathcal{Y}}_{1,\cdots,N}$, we need to fit the tensor dynamical models by estimating parameters $\boldsymbol{\theta} = (\boldsymbol{\mathcal{U}}_0, \boldsymbol{\mathcal{Q}}_0, \boldsymbol{\mathcal{Q}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{C}})$, where $\boldsymbol{\mathcal{U}}_0, \boldsymbol{\mathcal{Q}}_0$ is initial parameters satisfies

$$\boldsymbol{\mathcal{X}}_1 \sim \mathcal{N}(\boldsymbol{\mathcal{U}}_0, \boldsymbol{\mathcal{Q}}_0). \tag{19}$$

We cannot directly maximize the likelihood of the data with respect to $\boldsymbol{\theta}$ due to latent variable $\boldsymbol{\mathcal{X}}_n$. The EM algorithm provides iterative updating $E[\boldsymbol{\mathcal{X}}_n]$ and $\boldsymbol{\theta}$ in an alternating manner [7]. For the tensor dynamical models, instead of working directly to the tensor normal distribution, metricized and vectorized representation of likelihood can be used as follows [18]:

$$L(\boldsymbol{\theta} | \boldsymbol{\mathcal{X}}_{1,\cdots,N}, \boldsymbol{\mathcal{Y}}_{1,\cdots,N}) = L(vec(\boldsymbol{\theta}) | vec(\boldsymbol{\mathcal{X}}_{1,\cdots,N}), vec(\boldsymbol{\mathcal{Y}}_{1,\cdots,N}), \tag{20}$$

where $vec(\boldsymbol{\theta}) = (vec(\boldsymbol{\mathcal{U}}_0)), mat(\boldsymbol{\mathcal{Q}}_0), mat(\boldsymbol{\mathcal{Q}}), mat(\boldsymbol{\mathcal{A}}), mat(\boldsymbol{\mathcal{R}}), mat(\boldsymbol{\mathcal{C}}))$. The parameters of vectorized tensor dynamical models can be estimated. The factorizable $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{C}}$ can also be locally maximized by computing the gradient with respect to the vectorized projection matrices $\boldsymbol{v} = [vec(\boldsymbol{C}^{(1)})^T \cdots vec(\boldsymbol{C}^{(M)})^T]^T$ [18]. As a result of parameter estimation using EM, tensor sequence data $\boldsymbol{\mathcal{Y}}_{1,\cdots,N}$ is characterized by a transition tensor $\boldsymbol{\mathcal{A}} = \boldsymbol{A}^{(M)} \otimes \cdots \otimes \boldsymbol{A}^{(1)}$ and a projection tensor $\boldsymbol{\mathcal{C}} = \boldsymbol{C}^{(M)} \otimes \cdots \otimes \boldsymbol{C}^{(1)}$. The action recognition from tensor dynamical models is to analyze and estimate similarity from the collection of transition tensors and projection tensors for each class.

3. Statistical Analysis on Grassmann Manifold for Tensor Dynamic Models

There has been a study of control theory to model perturbation of linear dynamical systems like deformation [13] and scaling [5], and identifying linear dynamic systems [16]. Recently, there has been active research on modeling dynamics of human motion and dynamic textures [17] using system identification techniques. Dynamic textures are analyzed and synthesized using auto-regressive moving average process (ARMA) with linear dimensionality reduction [8]. They proposed a closed form solutions for learning the ARMA model parameters. The K-L divergence between two models is used for distance measurement. The space of linear dynamic system has a Riemannian structure and an inner product in the space of model should be defined on Stiefel manifolds [20].

3.1. Modeling dynamics on Stiefel manifold

In the linear dynamical system, the transition matrix $\boldsymbol{A}$ is constrained to be stable with eignevalues inside the unit circle and the observation matrix $\boldsymbol{C}$ is constrained to be an orthogonal matrix when the temporal dynamics is learned in the low-dimensional space via PCA [20]. Subspace angle between column-spaces of observability matrices are frequently used for comparison of the LDS models [6]. Recently several other approaches to measure similarity of linear dynamic systems are proposed [1, 2].

3.2. Statistical analysis on the Grassmann manifold

For the linear dynamical systems, the expected observation sequence can be given by

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA^2} \\ \cdot \\ \cdot \end{bmatrix} \boldsymbol{y}_0 \tag{21}$$

Thus the expected observation lies in the column space of the observation matrix given by

$$\boldsymbol{O}^T = [\boldsymbol{C}^T (\boldsymbol{CA})^T (\boldsymbol{CA^2})^T \cdots (\boldsymbol{CA^{(m-1)}})]. \tag{22}$$

In the case of tensor dynamical models, the observation matrix can be computed for each latent tensor:

$$\boldsymbol{O^{(i)}}^T = [\boldsymbol{C}^T (\boldsymbol{CA^{(i)}})^T (\boldsymbol{CA^{2(i)}}))^T \cdots (\boldsymbol{CA^{(m-1)(i)}})^T]. \tag{23}$$

Each observation subspace spanned by the columns of the matrix can be represented by an orthonormal basis after orthonormalization of each latent observation matrix. The orthonormal subspace of each latent tensor is a point on a Grassmann manifold and the N latent tensor dynamical systems are represented by N points on the Grassmann manifold. When each tensor latent state space $\boldsymbol{X}^{(i)}$ is d dimensional subspace, the latent observation matrix can be represented by d-dimensional orthonormal bases, which is represented by a quotient space of $SO(n)$.

The distance measurement on this Grassmann manifold can be estimated using the Riemannian structure. The distance of two points on the manifold can be represented by the shortest path among all the smooth paths on the manifold [20]. Exponential map and skew-symmetric matrix are utilized in the computation [20].

4. Experimental Results

We evaluated the performance of human action recognition on INRIA data set [21]. The data set consists of 10 actors performing 11 actions with multiple execution of each action. View-invariant representation and features proposed in [21] were used in the experiment. Evaluation results using one person for testing and the rest of them for training shows better performance using tensor dynamical models in action recognition.

5. Conclusion and Future Works

In this paper, we presented a new tensor dynamical system and its application to human action recognition from multiple views. Our action recognition experiment shows potential advantage of tensor dynamical systems compared with linear dynamical system approaches in human action recognition. The model may be applicable for other applications such as motion capture data analysis, texture analysis, and bio-informatics data analysis.

Statistical analysis of these distance distributions may improve classification performance of different activities of human motion sequence based on class mean of each activity.

Acknowledgement: This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2016R1D1A1B03931351).

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 *Proc. of CVPR*, pages 2208–2215, 2012. 4

[2] B. Afsari and R. Vidal. The alignment distance on spaces of linear dynamical systems. In *CDC*, pages 1162–1167, 2013. 4

[3] B. Afsari and R. Vidal. *Distances on Spaces of High-Dimensional Linear Stochastic Processes: A Survey*. Springer International Publishing, 2014. 1

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 3

[5] R. W. Brockett and P. Krishnaprasad. A scaling theory for linear systems. *IEEE Transactions on Automatic Control*, 25(2):197–207, 1980. 3

[6] K. D. Cock and B. D. Moor. Subspace angles between arma models. *Systems & Control Letters*, pages 265–270, 2002. 4

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(4):1–38, 1977. 3

[8] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *IJCV*, 51(2):91–109, 2003. 3

[9] Z. Ghahramani and G. E. Hinton. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, Department of Computer Science, University of Toronto, 1996. 3

[10] Z. Ghahramani and M. I. Jordan. Factorial hidden markov models. *Machine Learning*, 29(2):245–273, 1997. 1

[11] S. S. Ho, P. Dai, and F. Rudzicz. Manifold learning for multivariate variable-length sequences with an application to similarity search. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1333–1344, 2016. 1

[12] T. Kollo and D. von Rosen. *Advanced Multivariate Statistics with Matrices*. Springer, 2005. 2

Table 1. Action recognition performance in INRIA database

Methods	Motion History+Mahal. [21]	Grassmann+Linear DS [20]	Proposed Tensor DS
Recognition rate	93.33	93.93	**95.46**

[13] P. S. Krishanprasad and C. F. Martin. On families of systems and deformations. *Int. J. Control*, 38(5):1055–1079, 1983. 3

[14] A. N. Langville and W. J. Stuart. The kronecker product and stochastic automata networks. *Journal of Computational and Applied Matehmatics*, 167(2), 2004. 2

[15] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, 1988. 2

[16] P. V. Overschee and B. D. Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29(3):649–660, 1993. 3

[17] A. Ravichandran, R. Chaudhry, and R. Vidal. Categorizing dynamic textures using a bag of dynamical systems. *IEEE Trans. PAMI*, 35(2):342–353, 2013. 3

[18] M. Rogers, L. Li, and S. Russsell. Multiliinear dynamical systems for tensor time series. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2634–2642, 2013. 1, 2, 3

[19] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *Proceedings of the International Symposium on Computer Vision*, pages 265–, 1995. 1

[20] P. K. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Trans. PAMI*, 33(11):2273–2286, 2011. 1, 3, 4, 5

[21] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2):249–257, 2006. 4, 5