INFORMATICA, 2014, Vol. 25, No. 3, 361–384
© 2014 *Vilnius University*
DOI: http://dx.doi.org/10.15388/Informatica.2014.18

Learning Inductive Riemannian Manifold in Abstract Form by Modeling Embedded Dynamical System

Elham BAVAFAYE HAGHIGHI[1]*, Mohamad RAHMATI[1],
Guenther PALM[2], Saeed SHIRY GHIDARY[2]

[1] *Computer Engineering and Information Technology Department*
Amirkabir University of Technology, Tehran, Iran
[2] *Institute of Neural Information Processing, University Ulm, Ulm, Germany*
e-mail: e.bavafa@aut.ac.ir, rahmati@aut.ac.ir, guenther.palm@uni-ulm.de, shiry@aut.ac.ir

Received: December 2012; accepted: June 2013

Abstract. Manifold learning algorithms do not extract the structure of datasets in an abstract form or they do not have high performance for complex data.In this paper, a method for Learning an Inductive Riemannian Manifold in Abstract form (LIRMA) is presented in which the structure of patterns is determined by solving the embedded dynamical system of the patterns. In order to model corresponding system, the true sequence of patterns is estimated using a topology preserving method.LIRMA has the advantage of being an inductive method with low complexity. Additionally, it is a topology preserving method with respect to quantitative measures.

Key words: learning inductive Riemannian manifold in abstract form (LIRMA), embedded dynamical system, underlying structure of dataset, continues-invertible-smooth mapping.

1. Introduction

Dimensionality reduction is an important step in many machine learning problems such as solving the curse of dimensionality, removing the uninformative variables of a dataset or visualization problem. The dimensionality reduction methods are categorized into linear and non-linear approaches. The linear methods such as Principal Component Analysis (PCA) are successful when the assumption of patterns lying on a linear subspace is true (Jolliffe, 2002; Theodoridis and Koutroumbas, 2003; Izenman, 2008; Lee and Verleysen, 2007; Burges, 2009). However, this assumption is not satisfied for more complex datasets. Non-linear approaches such as principal curve and surface, polynomial PCA, kernel PCA, Self Organizing Map (SOM) and manifold learning methods are useful for dimensionality reduction of these datasets (Izenman, 2008; Lee and Verleysen, 2007; Burges, 2009; Masahiro, 2004; Law, 2006; Belkin, 2003). However, they do not extract the underlying structure of the dataset in an abstract form or they are not applicable for

*Corresponding author.

high dimensional datasets. By learning the underlying structure of data, inductive low dimensional representation of test patterns is possible.

Principal curve and principal surfaces (Izenman, 2008; Lee and Verleysen, 2007; Masahiro, 2004) present dataset in an abstract form and are applied inductively for non-linear dimensionality reduction. However, the general structure of the principal surface should manually be determined in advance. Local minimum solutions make these methods not a proper candidate for high dimensional datasets. Polynomial PCA is also not applicable in high dimensional feature space. The reason is that the size of extended matrix analyzed by PCA increases quickly (Izenman, 2008; Lee and Verleysen, 2007; Masahiro, 2004).

Kernel PCA is a dimensionality reduction approach which does not extract the underlying structure of the dataset. Determining the correct structure of kernels plays an important role in successful results of this method (Izenman, 2008; Lee and Verleysen, 2007; Masahiro, 2004). Similarly SOM does not extract the underlying structure of dataset. However, it has an acceptable performance for dimensionality reduction of datasets which are compatible with clustering assumption (Izenman, 2008; Burges, 2009; Law, 2006). Furthermore, there are local methods which propose better generalization ability by applying simple structures on local intervals. However, determining the local intervals is a challenging task and the success of the results depends on it. Additionally, these methods are not able to extract the global structure of dataset. As a result, they cannot be applied beyond the range of training samples (Ozertem and Erdogmus, 2011; Vapnik, 1998).

Manifold learning methods are developed rapidly by representing manifold topology using graphs and by applying Geodesic distance as a new metric (Lee and Verleysen, 2007). It is expected that the non-linear underlying structure of patterns and their low dimensional representation are discovered by using a manifold learning method. Theoretically, it is desirable that the underlying structure is determined as an invertible function (Izenman, 2008). As explained before, the advantage of such function is that mapping from low dimensional space to the original one and an inverse mapping are possible. However, in the discrete-graphical representation of a graph-based method, the underlying structure implicitly embeds in the connections between graph nodes (Izenman, 2008; Lee and Verleysen, 2007; Burges, 2009; Masahiro, 2004; Law, 2006; Belkin, 2003). Therefore, it is not explicitly estimated as an invertible mapping. Additionally, an adjacency graph is constructed with respect to the K nearest neighbors of samples in these methods. Low value of K partitions the constructed graph into separated connected components. Since the method is applied on the largest connected component, there might be missed training patterns during the dimensionality reduction process (Izenman, 2008; Lee and Verleysen, 2007).

Since there is no direct mapping for low dimensional representation of test patterns in graph based approaches, mediator methods are applied process (Izenman, 2008; Lee and Verleysen, 2007). Neural Networks such as Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF), K Nearest Neighbors (KNN) and interpolation function are some of mediators to estimate the low dimensional representation of the test patterns (Lee and

Verleysen, 2007). The memory and computational complexities of corresponding methods might be considerable in comparison to a direct mapping.

Although enhancements in the line of performance and applicability of non-linear dimensionality reduction methods are presented in recent papers, the main concept behind these methods are not changed (Tuzel *et al.*, 2008; Raducanu and Dornaika, 2012; Li *et al.*, 2011; Zhang *et al.*, 2011). In this paper, we present a method which models the embedded linear dynamical system of a dataset and applies it to estimate the underlying structure of data. By solving the embedded dynamical system, the parameterized mapping of Riemannian manifold of the dataset is obtained in abstract form (Ogata, 2002). As a result, this function can be applied inductively for test patterns without requirement to mediator methods such as KNN. In order to determine the embedded dynamical system, the order of patterns is estimated using a topology preserving dimensionality reduction method (Izenman, 2008; Lee and Verleysen, 2007; Burges, 2009; Masahiro, 2004; Law, 2006; Belkin, 2003; Zhang and Zha, 2004).

The important advantage of Learning an Inductive Riemannian Manifold in Abstract form (LIRMA) is that this method extracts the structure of patterns by using a weighted summation of rich set of functions (i.e. families of $t^{\alpha}\exp(\lambda t)$, $t^{\alpha}\exp(at)\sin(bt)$ and $t^{\alpha}\sin(bt)$ where $\alpha \in \mathbb{Z}^{+}$ and $\lambda, a, b \in \mathbb{R}$). By applying this abstract form, LIRMA is used inductively. Consequently, it has low computational and memory complexities and also acceptable performance to preserve the topology of data.

The arrangement of remaining sections is as follows: In Section 2, a review on principal surfaces and manifolds is made. Parametric modeling of linear dynamical systems is the subject of Section 3. Learning an Inductive Riemannian Manifold in Abstract form (LIRMA) using its embedded dynamical system is presented in Section 4. In Section 5, experimental results are given and finally, Section 6 includes our conclusions and future works.

2. Principal Surfaces and Manifold Learning Methods

Manifold learning is one of the effective methods for non-linear dimensionality reduction which develops rapidly by applying graph theory and metrics such as Geodesic distance (Izenman, 2008; Lee and Verleysen, 2007; Burges, 2009; Masahiro, 2004; Law, 2006; Belkin, 2003). The goal of manifold learning is to discover the non-linear underlying structure of dataset and the low dimensional representation of patterns (Izenman, 2008). According to the definition of Georg Friedrich Bernhard Riemann (1854), when a topological manifold is continuous and smooth (i.e. differentiable in any order or $\mathcal{C}^{\infty}$) and a metric such as geodesic distance is defined for it, the manifold is called Riemannian Manifold (Izenman, 2008; Lee and Verleysen, 2007). In case of a Riemannian manifold, the structure of the manifold is estimated using various tangent spaces. These tangent spaces discretize the manifold to flatten sub-manifolds which are defined in a same dimensionality (Tuzel *et al.*, 2008).

In Lin and Zha (2008), a frame work is presented for Riemannian Manifold Learning (RML) for dimensionality reduction which maps patterns on only one tangent space.

The low dimensional tangent space is estimated by using PCA which is applied on the center of patterns and its nearest neighbors.However, applying only one tangent space causes that the topological structure of Riemannian manifold and the theoretical expectations from Riemannian manifold are not satisfied. Additionally, the manifold assumption for applying RML is that its subject is considered as a unit sphere. Face recognition is a compatible task for this assumption. In Local Tangent Space Alignment (LTSA) (Zhang and Zha, 2004), the tangent spaces of all of training samples are estimated using their nearest neighbors. The tangent spaces are represented in low dimensional feature space using PCA. The global coordinate system is then learned to represent training samples in a single coordinate. Uniform distribution of patterns and lying samples and their close neighbors on linear subspaces are necessary conditions for best performance of LTSA (Zhang and Zha, 2004).

Principal curve and principal surfaces (Izenman, 2008; Lee and Verleysen, 2007; Masahiro, 2004) are methods which present dataset in an abstract form and are used inductively for non-linear dimensionality reduction. Before applying the learning algorithm for a principal curve/surface, its general structure should be determined in advance. The learning process includes projection and expectation step. In one iteration of learning, the low dimensional representations of samples are updated according to the previous structure of the curve (i.e. projection). In the expectation step, the parameters of the curve are expected by considering the updated representations of the samples. Sensitivity to the initial parameters of the curve/surface and consequently local minimum problem are the main disadvantages of this learning algorithm. These problems are more challenging for high dimensional datasets.

Comparing the advantage of applying LIRMA with projection-expectation algorithm demonstrates the challenge of learning for principal curve/surface. In Section 4, it is shown that the underlying structure of a dataset is estimated using the summation of the families $t^{\alpha}\exp(\lambda t)$, $t^{\alpha}\exp(at)\sin(bt)$ and $t^{\alpha}\sin(bt)$ where $\alpha \in \mathbb{Z}^{+}$ and $\lambda, a, b \in \mathbb{R}$. As a result, for n dimensional feature space, $n^2 + 3n$ parameters is needed (e.g. when $n = 50$, number of parameters $= 2650$). Therefore, the search space grows quickly and the projection-expectation algorithm returns a local minimum solution. Applying local learning methods might bean efficient strategy to solve corresponding challenge.

In order to reduce the order of complexity of a classifier, regressor and dimension reduction method, local methods are applied. Decreasing Vapnik–Chervonenkis (VC) dimension and better controlling the generalization ability are the main advantages of applying local learning approaches. However, determining the local intervals is a challenging task and the success of the results depends on it (Vapnik, 1998). Gaussian kernels and tangent spaces are usual options to estimate local intervals (Ozertem and Erdogmus, 2011). In addition to determing the local intervals, these methods are not able to extract the global structure of a dataset. As a result, they cannot be applied beyond the range of patterns (e.g. red points in Fig. 2, Section 4). Furthermore, acceptable performance in comparison to the other non-linear dimensionality reduction methods is not reported for local learning manifold (Ozertem and Erdogmus, 2011).

By applying graph based manifold learning such as (Isometric feature Mapping) ISOMAP, (Locally Linear Embedding) LLE and Laplacian Eigenmap (LE), the adjacency graph applies local information of dataset without the challenges of determining local intervals (Izenman, 2008; Lee and Verleysen, 2007; Burges, 2009; Masahiro, 2004; Law, 2006; Belkin, 2003). Although the underlying topological structure of dataset is not defined explicitly, it is embedded implicitly in the connections between graph nodes. Methods such as ISOMAP try to keep a global-local structure of patterns; however, LLE and LE preserve local curvatures better than a global shape (Izenman, 2008; Lee and Verleysen, 2007). Without considering different concepts behind graph based manifold learning methods for non-linear dimensionality reduction, they are similar in three main steps (Izenman, 2008). These steps are summarized as follows:

1. Extracting the weighted adjacency graph according to the neighborhood relations between patterns. Determining the proper number of neighbors in this step plays an important role in the success of the algorithm. Low values for the number of neighbors makes the adjacency graph spars. In contrast, applying large number of neighbors disturbs the local information and curvatures.
2. The second step includes the special characteristics of each algorithm. It prepares patterns for representation in low dimensional sub-space. According to the transforming process of the adjacency matrix of the graph, patterns will be prepared for low dimensional representation in the 3rd step.
3. The final step to represent patterns in low dimensional space involves solving an eigen-equation.

More details on learning graph based manifolds are given in Izenman (2008), Lee and Verleysen (2007), Burges (2009), Masahiro (2004), Law (2006), Belkin (2003). Since graph based methods are applied on the largest connected component of the adjacency graph, the dimensionality reduction process may ignore some of training samples (missed samples). With respect to the discrete-graph based representation of samples, the underlying structure of a dataset is defined implicitly between the nodes of the graph. However, corresponding structure does not extracted explicitly as an invertible mapping. As a result, mediator methods should be applied in order to represent a test pattern in low dimensional space. Recommended mediators to embed test patterns are neural networks such as Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF), K Nearest Neighbors (KNN) and local linear interpolation (Lee and Verleysen, 2007).

Local linear interpolation is one of the usual methods for representing a test pattern. The low dimensional representation is computed using linear interpolation on the low dimensional images of its K nearest neighbors. The computational complexities of finding K nearest neighbors (in the best case by applying Tournament sort Chen, 2002) and interpolation are $\mathcal{O}(nl + l + K \log(l))$ and $\mathcal{O}(Km)$ respectively (important notations are summarized in Table 1). As a result, the overall computational complexity of the algorithm is in the order of $\mathcal{O}(ln)$. All of training patterns in high dimensional space are needed to find the K nearest neighbors. Therefore, the memory complexity is in the order of $\mathcal{O}(ln)$. By increasing the number of training samples, the computational and memory complexities of interpolation increases accordingly.

The computational and memory complexities of neural networks depend on the complexity of dataset (i.e. topology of patterns in high dimensional space and their low di-

Table 1
Important notations.

$\mathbb{R}^n$	n-Dimensional Euclidian space where high dimensional patterns lie.
$\mathbb{R}^m$	m-Dimensional Euclidian space where low dimensional representation of patterns lie ($m \ll n$).
K	Number of nearest neighbors in KNN and LLE.
x_i	$x_i \in \mathbb{R}^n$ is ith training pattern ($1 \leqslant i \leqslant l$) or test pattern ($l+1 \leqslant i \leqslant L$).
$x(t)$	$x(t) \in \mathbb{R}^n$ is the state vector or response time of a continuous state equation.
x_r or $x_{r(i,k)}$	x_r is rth or $r(i,k)$th state vector in a discrete state space.
δ_k	Determines the selected features in kth set of correlated variables.
n_k	The number of variables in kth set of correlated variables. Therefore, it is the order of corresponding dynamical system for kth set.
x_{i,δ_k}	Representation of x_i using features determined by δ_k.
$\gamma_{i,k}$	One-dimensional representation of ith pattern for kth set of correlated variables using a non-linear dimension reduction method (e.g. LLE).
A^C or A_k^C	$A^C \in \mathbb{R}^{n_k \times n_k}$ is the state matrix of a continuous state space for kth set of correlated variables.
A^D or A_k^D	$A^D \in \mathbb{R}^{n_k \times n_k}$ is the state matrix of a discrete state space for kth set of correlated variables.
T	Time interval for sampling and discretizing state vectors.
$z(t)$ or $z_k(t)$	$z(t), z_k(t) \in \mathbb{C}^{n_k}$; the state vector of Jordan canonical state space for kth set of correlated variables.
U or U_k	$U, U_k \in \mathbb{C}^{n_k \times n_k}$; the orthonormal modal matrix to represent the Jordan canonical state space for kth set of correlated variables.
J^C, J^D	$J^C, J^D \in \mathbb{C}^{n_k \times n_k}$, the Jordan canonical state matrix for continuous and discrete state spaces for a set of correlated variables.
λ_p^C, λ_p^D	pth different eigenvalue of A^C or A^D for a set of correlated variables.
n_p	The power of $(\lambda - \lambda_p^C)$ in the characteristic polynomial of dynamical system for a set of correlated variables. For kth set of correlated variables: $\sum_p n_p = n_k$.
$\tau_{j,i,k}$	Scaled time or low dimensional representation for jth variable of ith sample for kth set of correlated variables.

mensional image). For example the number of basis in RBF are increased up to l in the worst case. As a result, computational and memory complexities are in the order of $\mathcal{O}(ln)$. By applying MLP, the computational complexity might be lower than KNN. However, the underlying structure of a dataset which is implicitly defined between the graph nodes of a manifold, is not defined by MLP. Additionally, local minimum solution is another disadvantage of applying neural networks.

By learning an inductive Riemannian manifold in abstract form, the underlying structure of patterns is estimated using a continuous invertible smooth function. As a result, low dimensional representation is accomplished inductively and directly for test patterns with respect to the structure of the dataset.

3. Parametric Modeling of Linear Dynamical Systems

By mathematical modeling of a linear dynamical system, the state space equation of the system is estimated. Representation of this system using state space equation is given by (1) (Ogata, 1995, 2002).

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t), \\ y(t) = C(t)x(t) + D(t)u(t). \end{cases} \tag{1}$$

In (1), $x(.)$, $u(.)$ and $y(.)$ are state vector, input (or control)vector and output vector of the continuous system respectively. $A(.)$, $B(.)$, $C(.)$ and $D(.)$ are the state matrix, input matrix, output matrix and feed through (feed forward) matrix of a time-variant system. In a time-invariant continuous system, these matrices are constants (i.e. A, B, C and D).

Modeling linear dynamical systems are divided to white box and black box approaches (Ogata, 2002; Ljung and Glad, 1994; Sishwick, 2007; Haefner, 2005). In a white box approach, the linear dynamical components of the system (or their linear approximations) and their interactions are observable. As a result, the order of the equation and its parameters are determined according to the specifications of the components. For more complex systems, black box approaches are applied. Non-parametric and parametric methods are the two main categories of black box approaches. In case of non-parametric methods, the model is estimated according to impulse response or frequency response of the system (Ljung and Glad, 1994). How to excite the system using an impulse or defining a sufficiently rich set of input to stimulate the major frequencies of the system are some of the main challenges of non-parametric modeling of linear dynamical systems. Additionally, these methods are not proportional for an autonomous system in which the matrix B is zero and the system is not affected by $u(.)$. In these systems, the dynamicity is determined according to the sequence of observed state vectors beginning from an initial state.

In the parametric modeling approach (Ljung and Glad, 1994; Sishwick, 2007; Haefner, 2005) the parameters of a dynamical system are estimated according to the initial state vector of it. Least Square Error (LSE) and its iterative versions are usual candidates to estimate/learn the state matrix of a discrete-autonomous dynamical system. The state space equation of a discrete-autonomous dynamical system in $\mathbb{R}^n$ is given in (2) (Ogata, 1995).

$$x_{r+1} = A^D x_r, \quad x_r \in \mathbb{R}^n, \ 1 \leqslant r \leqslant l. \tag{2}$$

The subscript r in (2), defines the order of the state vectors during sampling process. l is the number of observations. Equation (2) is also called Auto Regressive (AR) model (Ljung and Glad, 1994). From this point forward, the notations A^D and A^C are applied to present state matrices of Discrete and Continuous state spaces respectively (see (2) and (3)).

$$\dot{x}(t) = A^C x(t). \tag{3}$$

Equation (3) presents the continuous form of an autonomous time-invariant dynamical system. By considering T as the time interval of sampling and discretizing state vectors, the relation between A^D and A^C is defined by (4) (Ogata, 1995).

$$(x_{r+1} - x_r)/T \cong A^C x_r \quad \Rightarrow \quad x_{r+1} = \big(TA^C + I\big)x_r \quad \Rightarrow \quad A^D \cong TA^C + I. \tag{4}$$

In (4), I is $n \times n$ identity matrix. In Section 4, it is assumed that approximation (4) is equality.

In order to estimate A^D using LSE, the matrices $X_{1,l-1} \in \mathbb{R}^{n\times(l-1)}$ and $X_{2,l} \in \mathbb{R}^{n\times(l-1)}$ are constructed in such way that rth $(1 \leqslant r \leqslant l-1)$ column of these matrices are x_r and x_{r+1} respectively. At this step, (2) is rewritten by using $X_{1,l-1}$, $X_{2,l}$ which is given by (5) (Ljung and Glad, 1994; Sishwick, 2007; Haefner, 2005).

$$X_{2,l} = A^D X_{1,l-1}. \tag{5}$$

According to (5), A_D is estimated as follows:

$$A^D \cong X_{2,l} X_{1,l-1}{}^T \left(X_{1,l-1} X_{1,l-1}{}^T\right)^{-1} = X_{2,l} X^+_{1,l-1}. \tag{6}$$

In (6), ^{T} is transpose operator and $^+$ is Moore–Penrose pseudo inverse (Strang, 1988; Sheldon, 1997). Estimating A^D with respect to all of training samples in a batch form, makes modeling process more robust to the noise and distortion (Ljung and Glad, 1994; Sishwick, 2007; Haefner, 2005).

4. Learning an Inductive Riemannian Manifold in Abstract Form

The major expectation from a non-linear dimensionality reduction method is that it preserves the topological information of samples. As a result, similar patterns remain close together in their one dimensional representation which is learned by a topology preserving method. In this regard, deformable sample shapes on one dimensional manifold demonstrates existence of an embedded dynamical system in dataset (Fig. 1 and Eq. (2)). Consequently, a manifold can be estimated by modeling its embedded dynamical system.

The first step of estimating the underlying structure includes finding the most correlated features. It is expected that correlated features belong to the same structure. As a result, in order to represent dataset in m dimensional feature space, m sets of correlated features are determined. For each set of features, its embedded dynamical system is modeled. However, an ordered sequence of observations from the state vectors is required for such modeling. Therefore, patterns (which are presented by the selected correlated features) are ordered with respect to their one dimensional representation. The one dimensional representation is learned by a topology preserving non-linear dimensionality reduction method.At this step, the AR model of the system is estimated. By modeling the embedded linear dynamical system, corresponding Riemannian manifold is determined in abstract form by solving its canonical Jordan equivalent representation. With respect to the underlying structure of the dataset, samples are represented in low dimensional feature space inductively and directly.

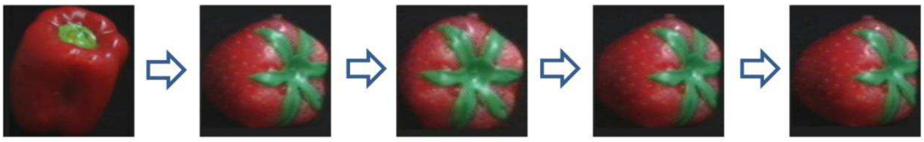


Fig. 1. Deformable sample shapes on a manifold confirms existence of an embedded dynamical system in the dataset. These figures are taken from COIL100 (Nene *et al.*, 1996, Table 2) and the order of images are determined by LLE.

4.1. *Partitioning to Correlated Features*

The probability of lying on a surface increases for most correlated features. In the other words, correlated features belong to the same structure. As a result, in order to represent n dimensional patterns in $\mathbb{R}^m$ using LIRMA, m subsets of correlated features are determined using PCA (Jolliffe, 2002; Theodoridis and Koutroumbas, 2003; Izenman, 2008; Lee and Verleysen, 2007). The selected features for kth dimension $(1 \leqslant k \leqslant m)$ are presented using δ_k which is defined by (7).

$$\begin{aligned}
&\delta_k = (\delta_{k1}, \delta_{k2}, \dots, \delta_{kn}), \quad \delta_{kj} \in \{0, 1\}, \\
&\sum_{k=1}^{m} \delta_{kj} = 1, \qquad \sum_{j=1}^{n} \delta_{kj} > 1, \quad 1 \leqslant j \leqslant n,\ 1 \leqslant k \leqslant m.
\end{aligned} \tag{7}$$

$\delta_{kj} = 1$ means that the jth element of x is selected to be applied for kth set. The condition $\sum_{k=1}^{m} \delta_{kj} = 1$ ensures that all of the variables are applied in LIRMA. Additionally, each variable is appeared in only in one of δ_ks. This condition makes a unique representation for each variable (see Section 4.5). The condition $\sum_{j=1}^{n} \delta_{kj} > 1$ guarantees that the number of applied variables for each dynamical system is more than one. Consequently, $n_k = \sum_{j=1}^{n} \delta_{kj}$ is the number of applied variables in each correlated set or it is the order of dynamical system of each set.

PCA is used to determine the set of correlated variables with respect to the importance of variables in each principal component. By sorting the elements of a principal component in increasing order, the effectiveness of corresponding variables are determined. With respect to the effect of variable scales on the covariance matrix and consequently on the principal components, applying PCA on the centered-normalized version of variables (Izenman, 2008) is more effective to reduce the effect of variable scales.

Since PCA results in a set of orthonormal basis of principal components, the importance of each variable in an eigenvector is different from the other eigenvectors. As a result, it is not probable that the most important variables of a major principal component have a same degree of importance in the other ones. Therefore, the first $\lfloor n/m \rfloor$ non-selected variables of kth major eigenvectorare considered for kth set where $\lfloor . \rfloor$ is the floor operator. According to the explanations of this subsection, the algorithm of partitioning to correlated features is summarized as follows:

1. Apply PCA on the centered-normalized dataset.
2. Select the first m major principal components.
3. For $1 \leqslant k \leqslant m$:
   3.1 Select the first $\lfloor n/m \rfloor$ non-selected variables of kth eigenvector for kth set of correlated features.
   3.2 Set $\delta_{kj} = 1$ $(1 \leqslant j \leqslant n)$ for the selected variables accordingly.

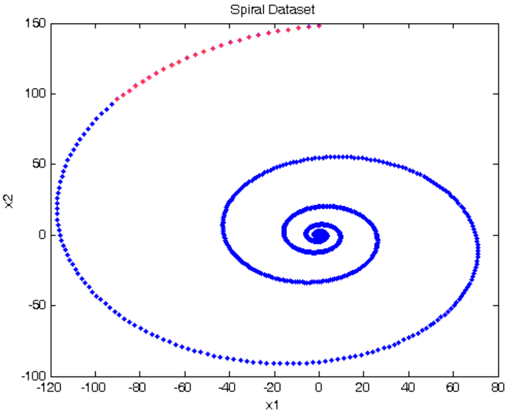


Fig. 2. The spiral dataset. The traditional methods of manifold learning do not estimate the underlying structure of such datasets in abstract form. As a result, they are not able to predict the shape of the dataset beyond the range of training samples (e.g. red points).

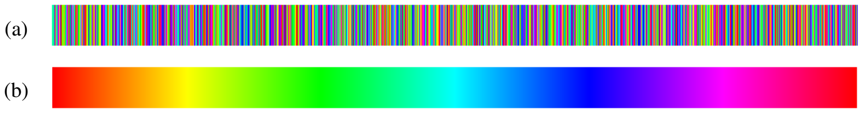


Fig. 3. In this figure, $t \in [1, 1000]$ is illustrated using the correspondence between [1, 1000] and the color range [orange, red]. (a) The unordered sequence of t by which the dataset of spiral is constructed. (b) By applying LLE to represent dataset in one dimensional space, the order of t and consequently the order of samples are estimated.

4.2. *Order Reconstruction*

Estimating the order of patterns for each set of correlated variables is required to determine corresponding AR model. The ith sample ($1 \leqslant i \leqslant l$) with δ_k features is represented using $x_{i,\delta_k} \in \mathbb{R}^{n_k}$. By applying a topology preserving non-linear dimensionality reduction method (Izenman, 2008; Lee and Verleysen, 2007; Burges, 2009; Masahiro, 2004; Law, 2006; Belkin, 2003; Zhang and Zha, 2004) on x_{i,δ_k}s to represent them in one dimensional feature space, $\gamma_{i,k} \in \mathbb{R}$ is determined for each pattern. Since the applied method is topology preserving, it is expected that similar $\gamma_{i,k}$s belong to similar x_{i,δ_k}s. As a result by sorting $\gamma_{i,k}$s, corresponding x_{i,δ_k}s in the sequence are similar and deformable (Fig. 1). Therefore, the order of x_{i,δ_k}s is estimated by sorting the values of $\gamma_{i,k}$ in ascending order.

Figure 2 illustrates the spiral dataset which is produced synthetically for $x_1 = \sin(0.01\pi t)\exp(0.005t)$ and $x_2 = \cos(0.01\pi t)\exp(0.005t)$. The values of t belong to the random permutation of the integers in [1, 1000]. As a result, the sequence of samples in dataset is not preserved. Figure 2 illustrates the shape of the dataset according to the values of x_1 and x_2. In Fig. 3(a), the unordered sequence of t is presented. In this illustration, the values of t is represented schematically with respect to the correspondence between [1, 1000] and the color range [orange, red] where orange is the left side of Fig. 3(b) and red is right side of Fig. 3(b). It is shown in Fig. 3(b) that the order of patterns is estimated by applying LLE by representing dataset in one dimensional space. In order to apply LLE, the adjacency graph is constructed using 15 nearest neighbors for each sample.

The sequence γ_k estimates the order of applied t in Riemannian manifold. However, the true values of t by which the underlying structure is determined in abstract form is not determined.

4.3. *Model Estimation*

Once the $\gamma_{i,k}$s, are determined, these values are ordered increasingly. In this way, $r(i,k)$ is the rank of ith sample in the kth correlated set. Consequently, $x_{r(i,k)}$ is $r(i,k)$th sample in the ordered sequence of kth correlated set. By applying a topology preserving method, it is expected that $\gamma_{i,k}$s of adjacency samples in an ordered sequence have minor differences (Fig. 1). The gradual deforming shape of patterns in an ordered sequence is mathematically represented by (8) using a discrete linear dynamical system.

$$x_{r(i,k)+1} = A_k^D x_{r(i,k)}, \tag{8}$$

where $A_k^D \in \mathbb{R}^{n_k \times n_k}$. By considering $x_{r(i,k)=1}, x_{r(i,k)=2}, \dots, x_{r(i,k)=l}$, the matrices $X_{1,l-1} \in \mathbb{R}^{n_k \times (l-1)}$ and $X_{2,l} \in \mathbb{R}^{n_k \times (l-1)}$ are constructed by applying the first and the last $l-1$ samples of the sequence respectively. A_k^D is estimated by applying (6) with respect to all of training samples in a batch form. As a result, the estimated AR model is robust of to the noise (Ljung and Glad, 1994; Sishwick, 2007; Haefner, 2005). Since the sequence of patterns is estimated using a non-linear dimensionality reduction method, there are minor dis-orderings in the sequence which affects as noise and distortion. Applying (6), makes our modeling robust to the dis-ordering noise.

According to A_k^D and (4), A_k^C is given by (9).

$$A_k^C \cong \left(A_k^D - I_{n_k \times n_k}\right)/T. \tag{9}$$

As a result, the embedded dynamical system for kth correlated set of variables is estimated in the continuous form of (3). For simplicity the index k is omitted in most of formulas from this point forward (i.e. $z_k \leftrightarrow z$, $U_k \leftrightarrow U$, $A_k^D \leftrightarrow A^D$ and $A_k^C \leftrightarrow A^C$). By representing (3) using equivalent Jordan canonical state equation, it is known that (Strang, 1988; Sheldon, 1997):

$$\dot{z}(t) = J^C z(t),$$
$$\text{s.t.} \begin{cases} A^C = U^T J^C U, \\ U^T U = I_{n \times n}, \\ z(t) = Ux(t). \end{cases} \tag{10}$$

The factorization of $A^C = U^T J^C U$ is computed using Singular Value Decomposition (SVD). In (10), $z(t) \in \mathbb{C}^{n_k}$ ($\mathbb{C}$ is the set of complex numbers) and the modal matrix $U \in \mathbb{C}^{n_k \times n_k}$ includes orthonormal eigenvectors of A^C. It rotates the underlying Riemannian surface without modifying topology of the dataset in a complex space.The importance of representing by equivalent Jordan canonical state equation is that it is possible to analyze variables related to each Jordan block independent from the other blocks.

4.4. *Response Time Analysis*

The response time of (10) is a parameterized curve on Riemannian surface (Ogata, 2002). In order to analyze the response time of the system, the roots of characteristic polynomial of state space should be analyzed. Since these roots are the main frequencies of the system, they are not changed in the equivalent state space. Additionally, there is a relationship between J^C and J^D.[2] By considering $A^C = (A^D - I)/T$, it is proved in linear algebra (Strang, 1988; Sheldon, 1997) that the modal matrices of A^C and A^D are the same (i.e. U). Additionally, the relationship between eigenvalues of A^C and A^D is as follows:

$$\lambda^C = \left(\lambda^D - 1\right)/T, \tag{11}$$

where λ^C and λ^D are eigenvalues of A^C and A^D respectively. In order to solve (10), λ^Cs are analyzed. The characteristic polynomial of equivalent state space ($P(\lambda)$) (Ogata, 2002), is presented in (12) according to its eigenvalues;

$$P(\lambda) = \prod_{p=1}^{\acute{n}} \left(\lambda - \lambda_p^C\right)^{n_p}. \tag{12}$$

In (12), λ_p^C is pth different eigenvalue of A^C, $\acute{n}$ is number of different λ^Cs and n_p is the power of $(\lambda - \lambda_p^C)$ where $\sum_p n_p = n_k$. For $n_p = 1$ and $\lambda_p^C \in \mathbb{R}$, the response time of jth variable of $z(t)$ (i.e. $z_j(t)$) which is in correspondence with λ_p^C, is presented as (Ogata, 1995, 2002):

$$z_j(t) = \exp\left(\lambda_p^C t\right) z_j(0). \tag{13}$$

The value of $z_j(0)$ is determined using $Ux_{r(i,k)=1}$ where $x_{r(i,k)=1}$ is the first pattern (state) in the ordered sequence (see (10)). With respect to (13) and (11), the value of t for ith sample in kth set is given by (14).

$$\tau_{j,i,k} = t_{j,i,k}/T = \operatorname{Re}\left(\log\left(z_j(t)/z_j(0)\right)/\left(\lambda_p^D - 1\right)\right). \tag{14}$$

The general definition of $\log(y)$ for $y = a + b\underline{i} \in \mathbb{C}$ is $\log(r) + \theta\underline{i}$ where $r = \sqrt{a^2 + b^2}$ and $\theta = \operatorname{arctg}(b/a)$.[3] It can be shown that $\log(z_j(t)/z_j(0))/(\lambda_p^D - 1)$ is a real number for a noise free dataset for $\lambda_p^D, \lambda_p^C \in \mathbb{C}$. In practice, because of the noise and disturbance of

[2] J^D is computed by factorizing $A^D = U^T J^D U$ using SVD.

[3] In order to make difference between i as ith sample and i in imaginary part of a complex number, $\sqrt{-1}$ is shown by $\underline{i}$.

data, complex values might be appeared at right side of (14). By applying Re(.), the real part of the right side of (14) is extracted.

In (14), note that T is only a scaling factor. As a result, τ is applied for low dimensional representation of patterns. Additionally, it is expected that $\tau_{j,i,k}$s are the same for variables of x_{i,δ_k}. However, such expectation is not possible in a real world noisy dataset. As a result, the mean of $\tau_{j,i,k}$s is computed for variables of x_{i,δ_k}. Computing the mean of $\tau_{j,i,k}$s makes the result of LIRMA less sensitive to the noise of the dataset.

In case of $\lambda_p^C = a_p + b_p \underline{i} \in \mathbb{C}$, the response time is given by:

$$z_j(t) = \exp\big(\lambda_p^C t\big) z_j(0) = \big(\exp(a_p t)\cos(b_p t) + \exp(a_p t)\sin(b_p t)\underline{i}\big) z_j(0). \tag{15}$$

The solution of (15) is the same as (14). Complex conjugate root theorem states that when $\lambda_p^C = a_p + b_p \underline{i}$ is one of the roots of $P(\lambda)$, $\lambda_p^C = a_p - b_p \underline{i}$ is the other root of the equation (McGuire *et al.*, 2002). The response time of such systems includes a spiral form similar to Fig. 2. For the example given in Fig. 2, the complex roots of state equation are estimated as $0.0055 \pm 0.0313\underline{i}$. By applying the rotation matrix, x_1 and x_2 are estimated in abstract form as $\sin(0.0313t)\exp(0.0055t)$ and $\cos(0.0313t)\exp(0.0055t)$ which are close estimations of their original structures. Additionally, according to the abstract form of the spiral dataset, it is possible to predict its shape beyond the range of the training patterns (e.g. red points in Fig. 2).

When λ_p^C has only the imaginary part (i.e. $\pm b_p \underline{i}$), the dataset includes a circle shape. The response time of $z_j(t)$ is given by,

$$z_j(t) = \big(\cos(b_p t) + \sin(b_p t)\underline{i}\big) z_j(0). \tag{16}$$

When $n_p > 1$, there is Jordan block on the main diagonal of J^C. An example for J^C with $\acute{n} = 2$, $n_1 = 1$ and $n_2 = 2$ is $J^C = [\lambda_1, 0, 0;\ 0, \lambda_2, 1;\ 0, 0, \lambda_2]$ in which rows are separated with ';'. The main diagonal of J^C includes a single λ_1 and a Jordan block for λ_2 (i.e. $[\lambda_2, 1;\ 0, \lambda_2]$). The Jordan block corresponding to λ_p^C is a $n_p \times n_p$ matrix which is presented by (17).

$$J_p^C = \begin{bmatrix} \lambda_p^C & 1 & 0 & \dots & 0 \\ 0 & \lambda_p^C & 1 & \dots & 0 \\ 0 & 0 & \lambda_p^C & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & \lambda_p^C \end{bmatrix}_{n_p \times n_p}. \tag{17}$$

More details on construction of Jordan blocks, its generalized eigenvectors and response time can be found in Ogata (1995), Strang (1988), Sheldon (1997). The response time

$[z_j(t), \dots, z_{j+n_p-1}(t)]^T$ associated with J_p^C, is presented in (18).

$$\begin{bmatrix} z_j(t) \\ z_{j+1}(t) \\ \vdots \\ z_{j+n_p-2}(t) \\ z_{j+n_p-1}(t) \end{bmatrix} = \begin{bmatrix} \exp(\lambda_p^C t) & t\exp(\lambda_p^C t) & t^2\exp(\lambda_p^C t)/2! & \cdots & t^{n_p-1}\exp(\lambda_p^C t)/(n_p-1)! \\ 0 & \exp(\lambda_p^C t) & t\exp(\lambda_p^C t) & \dots & t^{n_p-2}\exp(\lambda_p^C t)/(n_p-2)! \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & t\exp(\lambda_p^C t) \\ 0 & 0 & 0 & \dots & \exp(\lambda_p^C t) \end{bmatrix}_{n_p \times n_p} \times \begin{bmatrix} z_j(0) \\ z_{j+1}(0) \\ \vdots \\ z_{j+n_p-2}(0) \\ z_{j+n_p-1}(0) \end{bmatrix}. \tag{18}$$

According to (18), $\tau_{j+n_p-1,i,k}$ is estimated using (14). The response time of the other variables includes weighted summation of $t^\alpha \exp(\lambda_p^C t)/\alpha!$ $(1 \leqslant \alpha < n_p)$. As a result, for $n_p > 1$, the response time is invertible for z_{j+n_p-1} which is not associated with the generalized eigenvectors of Jordan block. In (18), the part of underlying structure of dataset is estimated using a rich set of functions includes the family of $t^\alpha \exp(\lambda t)$, $t^\alpha \exp(at)\sin(bt)$ and $t^\alpha \sin(bt)$.

4.5. *Underlying Structure Estimation Using LIRMA*

Figure 4 illustrates the algorithm of learning inductive Riemannian manifold in abstract form. The first step is partitioning the set of features to m subsets. From this point, LIRMA can take the advantages of parallel computing (El-Rewini and Abd-El-Barr, 2005). For each set of correlated features, patterns are ordered using a topology preserving non-linear dimensionality reduction method (e.g. ISOMAP, LLE, LE or LTSA).

After determining the order of patterns, the AR model which presents the dynamicity between deformable ordered patterns is estimated. The dynamical system is represented using equivalent Jordan canonical form. In order to determine the underlying structure for kth set of correlated variables $(1 \leqslant k \leqslant m)$, corresponding response time $z_k(t)$ and modal matrix U_k are required. It is worth reminding that for simplicity the index k is omitted in most formulas of Sections 4.3 and 4.4 (i.e. $z_k \leftrightarrow z$, $U_k \leftrightarrow U$, $A_k^D \leftrightarrow A^D$ and $A_k^C \leftrightarrow A^C$). With respect to the roots of characteristic equation presented by (12), the response time of each variable of z_k is determined (using (13), (15), (16) and (18)). The response time $x_{\delta_k}(t)$ is given by (19).

$$x_{\delta_k}(t) = U_k^T z_k(t). \tag{19}$$

By modeling the embedded dynamical system of a dataset, LIRMA estimates the underlying structure in the form of Riemannian surface using weighted summation of the families $t^\alpha \exp(\lambda t)$, $t^\alpha \exp(at)\sin(bt)$ and $t^\alpha \sin(bt)$ where $\alpha \in \mathbb{Z}^+$, $\lambda, a, b \in \mathbb{R}$. The condition $\sum_{k=1}^m \delta_{kj} = 1$ of (7) ensures that each variable is appeared in only in one of δ_ks. As a result, there is a unique representation for each variable.

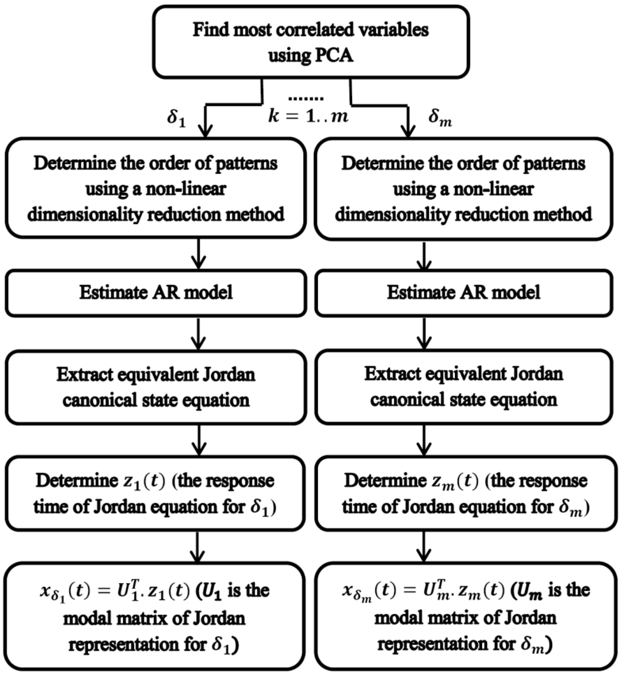

Find most correlated variables using PCA
....
δ_1
$k = 1..m$
δ_m
Determine the order of patterns using a non-linear dimensionality reduction method
Determine the order of patterns using a non-linear dimensionality reduction method
Estimate AR model
Estimate AR model
Extract equivalent Jordan canonical state equation
Extract equivalent Jordan canonical state equation
Determine $z_1(t)$ (the response time of Jordan equation for δ_1)
Determine $z_m(t)$ (the response time of Jordan equation for δ_m)
$x_{\delta_1}(t) = U_1^T . z_1(t)$ (U_1 is the modal matrix of Jordan representation for δ_1)
$x_{\delta_m}(t) = U_m^T . z_m(t)$ (U_m is the modal matrix of Jordan representation for δ_m)

Fig. 4. The algorithm of learning inductive Riemannian manifold in abstract form. This algorithm can take the advantage of parallel computing.

The computational complexity of LIRMA depends on applying PCA for partitioning into correlated features ($\mathcal{O}(n^3 + n^2 l)$) (Sharma and Paliwal, 2007), LLE ($\mathcal{O}(nl^2 + nlK^3 + ml^2)$) (Kayo, 2006), modeling dynamical system using Moore–Penrose pseudo inverse operator ($\mathcal{O}(n^3)$) (Stewart, 1998) and representing the equivalent Jordan canonical state equation ($\mathcal{O}(2n^3)$) (Sharma and Paliwal, 2007). As a result, the computational complexity of the algorithm is polynomial with respect to its parameters. In order to apply LIRMA, several matrices with $n \times n$ and $n \times l$ dimensionalities are required. Since in general $n < l$, the memory complexity of training LIRMA is in the order of $\mathcal{O}(n\ l)$.

4.6. *Inductive Dimension Reduction Using LIRMA*

The main advantage of applying LIRMA is estimating the underlying structure of dataset in abstract form. It results in applying LIRMA inductively for non-linear dimensionality reduction. The process of dimensionality reduction using LIRMA is illustrated in Fig. 5. After learning inductive Riemannian manifold in abstract form (Fig. 4), the pattern x is partitioned to x_{δ_k}s ($1 \leqslant k \leqslant m$). From this point, the algorithm can take the advantage of parallel computing (El-Rewini and Abd-El-Barr, 2005). Each x_{δ_k} is represented in corresponding equivalent Jordan canonical state space using (20)

$$z_k = U_k x_{\delta_k}. \tag{20}$$

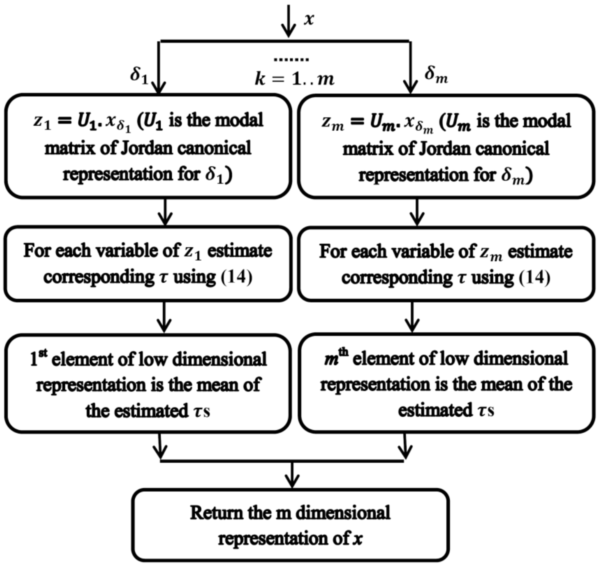

x
δ_1
....
$k = 1..m$
δ_m
$z_1 = U_1.x_{\delta_1}$ (U_1 is the modal matrix of Jordan canonical representation for δ_1)
$z_m = U_m.x_{\delta_m}$ (U_m is the modal matrix of Jordan canonical representation for δ_m)
For each variable of z_1 estimate corresponding τ using (14)
For each variable of z_m estimate corresponding τ using (14)
1st element of low dimensional representation is the mean of the estimated τs
mth element of low dimensional representation is the mean of the estimated τs
Return the m dimensional representation of x

Fig. 5. The algorithm of non-linear inductive dimensionality reduction using LIRMA. This algorithm can take the advantage of parallel computing.

According to the main frequencies in the response time of $z_k(t)$, $\tau_{j,k}$s ($1\leqslant j \leqslant n_p$) are determined for each variable of z_k (using (14)). For $n_p > 1$, corresponding response time is invertible for the variable which is not associated with the generalized eigenvectors of Jordan block. Since log(.) is not defined for zero input, an epsilon number should be added to x to solve such issues. The kth element of low dimensional representation of x is the mean of $\tau_{j,k}$s ($1 \leqslant j \leqslant \acute{n}$). Applying the mean operator, makes our proposed method more robust to noise and disturbances.

The computational complexity of dimension reduction algorithm is related to rotation using U_k ($\mathcal{O}(n^2)$) and the number of applying log(.) ($\mathcal{O}(n)$). As a result, the computation complexity of the algorithm is from the order of $\mathcal{O}(n^2)$. In order to apply the result of LIRMA for non-linear dimension reduction, n frequencies (eigenvalues) of the system and rotation matrix are required. Therefore, the memory complexity is from the order of $\mathcal{O}(n^2)$. Since $n < l$ in most of datasets, the computational and memory complexity of the proposed method is better than graph based approaches.

5. Experimental Results

In Section 4, LIRMA is applied for spiral dataset. Additionally, its ability is tested successfully for other lost order linear dynamical systems with dimensionality up to 30. In these tests, LLEis applied with $K = 15$. The number of nearest neighbors to find the order of patterns plays an important role for successful results of LIRMA. For the synthetic example of Fig. 2, $K = 30$ is not a correct option. In this case, x_1 and x_2 are estimated as

Table 2
Specifications of the applied datasets.

	n	l	L
MNIST	784	1000	11 000
Forest	54	2000	7000
COIL100	292	1800	7200
Face	10 304	160	400
Robot	24	4910	5457
Segment	19	210	2310
MFeat	649	1800	2000

$\exp(0.1725t)$ and $\exp(1.1019t)$. However, by applying $K = 15$, x_1 and x_2 are estimated as $\sin(0.0313t)\exp(0.0055t)$ and $\cos(0.0313t)\exp(0.0055t)$ which are close to their original versions. $K = 15$ is confirmed by the experimental results of Karbauskaitė and Dzemyda (2009). In addition to the lost order dynamical systems, our proposed method is applied for real world datasets which are more challenging tasks. In this case, the ability of LIRMA for inductive dimensionality reduction is tested with respect to a quantitative measure.

5.1. *Datasets Specifications*

Table 2 presents the specification of datasets which are applied in this paper. MNIST (MNIST dataset) is the set of handwritten digits. Each digit has been size-normalized and centred in a fixed size (28 × 28) image. The first 1000 samples in original training set of MNIST are used for training in this paper. COIL100 (Nene *et al.*, 1996) contains colour images of 100 different objects which are turned by 5°. As a result, there are 72 images from different views for each object. In COIL100, 18 images from each object (which are turned 20°) are applied to train and the 54 remaining images are used to test.

Similar to Kietzmann *et al.* (2008), from each image of COIL100, 292 dimensional features are extracted. Each extracted feature contains 64 × 3 dimension for the histograms of Lab channels, 64 dimensional histogram of Discrete Cosine Transformation (DCT), 8 dimension for Hu moments in addition to the logarithm form of their absolute values, 10 dimensional shape information which contains centroid, compactness, perimeter, eccentricity, circularity, aspect ratio, elongation, maximum and minimum diameters in addition to the logarithm of their absolute values.

For Face dataset (Face Dataset), 10 different images have been taken from 40 distinct human subjects. The images are presented using 92 × 112 pixels in 256 grey levels. The facial expressions (open/closed eyes, smiling/not smiling), facial details (glass/no glass) or lightening are not the same for some subjects. Other datasets which are Forest Cover Type (Forest), Wall Following Robot (Robot), Segmentation (Segment) and Multiple Feature Digit (MFeat), are downloaded from UCI repository (UCI Repository). Train and test samples of Forest are selected randomly from its original dataset. No feature extraction is applied on the UCI, Face and also on MNIST datasets.

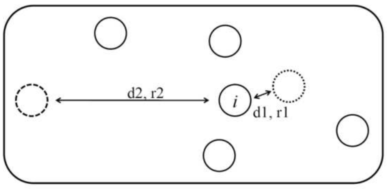


Fig. 6. The Euclidian distance holds more topological information of the dataset in comparison to a rank measure. For two neighbors of ith sample, Euclidian distances (d1 and d2) and the ranks of the neighbors (r1 and r2) are given. It might be possible that d1 $\ll$ d2 while r1 $<$ r2.

5.2. *Topology Preserving Quantitative Measures*

Recently, quantitative measures are introduced to verify how much a manifold preserves the topology of a dataset. These quantitative measures are more reliable than a visual-subjective qualitative one. In Karbauskaitė and Dzemyda (2009), it is shown that Mean Relative Rank Errors (MRRE) is one of the acceptable measures to ensure that the local topological information of the dataset is preserved. It ensures that the nearest neighbors of samples in the new space have the same rank of closeness in the original one and wise versa. By considering X as the set of training samples in high dimensional space, MRRE defines the rank $\overline{\overline{r}}_X(i,j)$ as follows:

1. By taking x_i as the reference point, compute Euclidian distances between x_i and x_j $(1 \leqslant i, j \leqslant l)$.
2. For each x_i, sort the distances in ascending order.
3. The output of $\overline{\overline{r}}_X(i,j)$ is the rank of x_j in the sorted list of distances such that for $j = \min_{1\leqslant s\leqslant l,\, s\neq i} \|x_i - x_s\|$, $\overline{\overline{r}}_X(i,j) = 1$.

Similarly, $\overline{\overline{r}}_Y(i,j)$ is defined for Y which includes low dimensional representation of the samples. In our experiments, $y_i \in Y$ is in correspondence with γ (in case of comparison methods, i.e. LLE, LE, etc.) or τ (for LIRMA). $MRRE_{train}$ (from Y to X) is defined by (21).

$$MRRE_{train} = \frac{1}{C}\sum_{i=1}^{l}\sum_{j\in N_K(y_i)} \frac{|\overline{\overline{r}}_X(i,j) - \overline{\overline{r}}_Y(i,j)|}{\overline{\overline{r}}_Y(i,j)}, \qquad C = l\sum_{k=1}^{K}\frac{|2k-l-1|}{k}. \tag{21}$$

In (21), $N_K(y_i)$ is a set which includes the indices of K nearest neighbors of y_i and C is the normalization constant which scales the result between [0, 1]. For a topology preserving method, $MRRE_{train}$ converges to zero. In other words, the rank of the nearest neighbors of the new representation is similar to the original space. Although MRRE is one of the reliable measures to ensure topological information is preserved, it is based on the rank of the neighbors. The Euclidian distance holds more topological information of the dataset in comparison to a rank measurement. Figure 6 illustrates a schematic example in which for two neighbors of ith sample, Euclidian distances (d1 and d2) and the ranks of the neighbors (r1 and r2) are given. It might be possible that d1 $\ll$ d2 while r1 $<$ r2.

Additionally, missed samples in graph based methods make *False Ranking* in *MRRE*. Since the given rank to the patterns ignores the missed samples, the assigned rank is not equal to the true one (i.e. by considering all of the training samples). As a result, *MRRE* is not a reliable measure when there are missed samples.

With respect to the problems of *MRRE*, we propose Mean Relative Euclidian Distances (MRED). Corresponding measure is defined by (22) and (23) for training and test patterns respectively.

$$MRED_{train} = \frac{1}{l}\sum_{i=1}^{l}\sum_{j\in N_K(y_i)} \|x_i - x_j\|, \tag{22}$$

$$MRED_{test} = \frac{1}{(L-l)}\sum_{i=l+1}^{L}\sum_{j\in N_K(y_i)} \|x_i - x_j\|. \tag{23}$$

In (22) and (23), $\|x_i - x_j\|$ returns Euclidian distance between two patterns. The missed training samples of graph based method are penalized using mean of $MRED_{train}$ evaluated for LIRMA. Applying Euclidian distance in MRED is more reliable option to determine how much a method preserves local topological information of a dataset.

5.3. *Performance Evaluation*

In this section, the performance of LIRMA is evaluated in comparison to kernel PCA, ISOMAP, LLE, LE and LTSA (MTDR v0.8b, 2012) by using MRED measure. In order to select comparison methods, different approaches are analysed. As explained in Sections 1 and 2, principal surfaces and polynomial PCA estimate the underlying structure of dataset; however, they are not applicable for high dimensional or complex real world datasets (Izenman, 2008). For example, in order to apply principal surface for Forest, MNIST and Face datasets using a structure similar to LIRMA, 3078, 617 008 and 106 203 328 parameters are needed to learn respectively.

The manifold assumption for applying RML is that its subject is considered as a unit sphere (Lin and Zha, 2008) which is not satisfied in most of real applications. As a result, RML is not selected in our comparisons. Determining local intervals is a challenging task in case of local methods and acceptable performance in comparison to the other non-linear dimensionality reduction methods is not reported for them (Ozertem and Erdogmus, 2011). However, in case of LTSA, the tangent spaces of all of training samples are estimated using their nearest neighbors (Zhang and Zha, 2004). As a result, LTSA is an acceptable option to show the performance of a local approach by considering all tangent spaces.

As Graph based methods are the most successful non-linear approaches (Lee and Verleysen, 2007), ISOMAP, LLE and LE areselected for comparisons.ISOMAP is a qualified graph based method to preserve the global-local structure of dataset.LLE and LE are locally topology preserving approaches (Izenman, 2008; Lee and Verleysen, 2007;

Burges, 2009; Masahiro, 2004; Law, 2006; Belkin, 2003). Therefore, the performance of LIRMA is evaluated in comparison to ISOMAP, LLE, LE and LTSA. These methods are applied with 15 nearest neighbours. Additionally, kernel PCA is applied in our comparisons as a kernel based approach. All of these methods are applied individually for dimensionality reduction as well as to determine the order of patterns in LIRMA (i.e. LIRMA + ISOMAP, LIRMA + LLE, etc.). Figures 7(a)–7(g) illustrate the performance of these methods to train and test of different datasets. In case of Kernel PCA, ISOMAP, LLE, LE and LTSA, interpolation functions are applied forlow dimensional representation of test samples. In Table 3, the mean and standard deviation of MRED measures computed for LIRMA (on LIRMA + ISOMAP, LIRMA + LLE, etc.) and the other comparison methods (ISOMAP, LLE, etc.) are given.

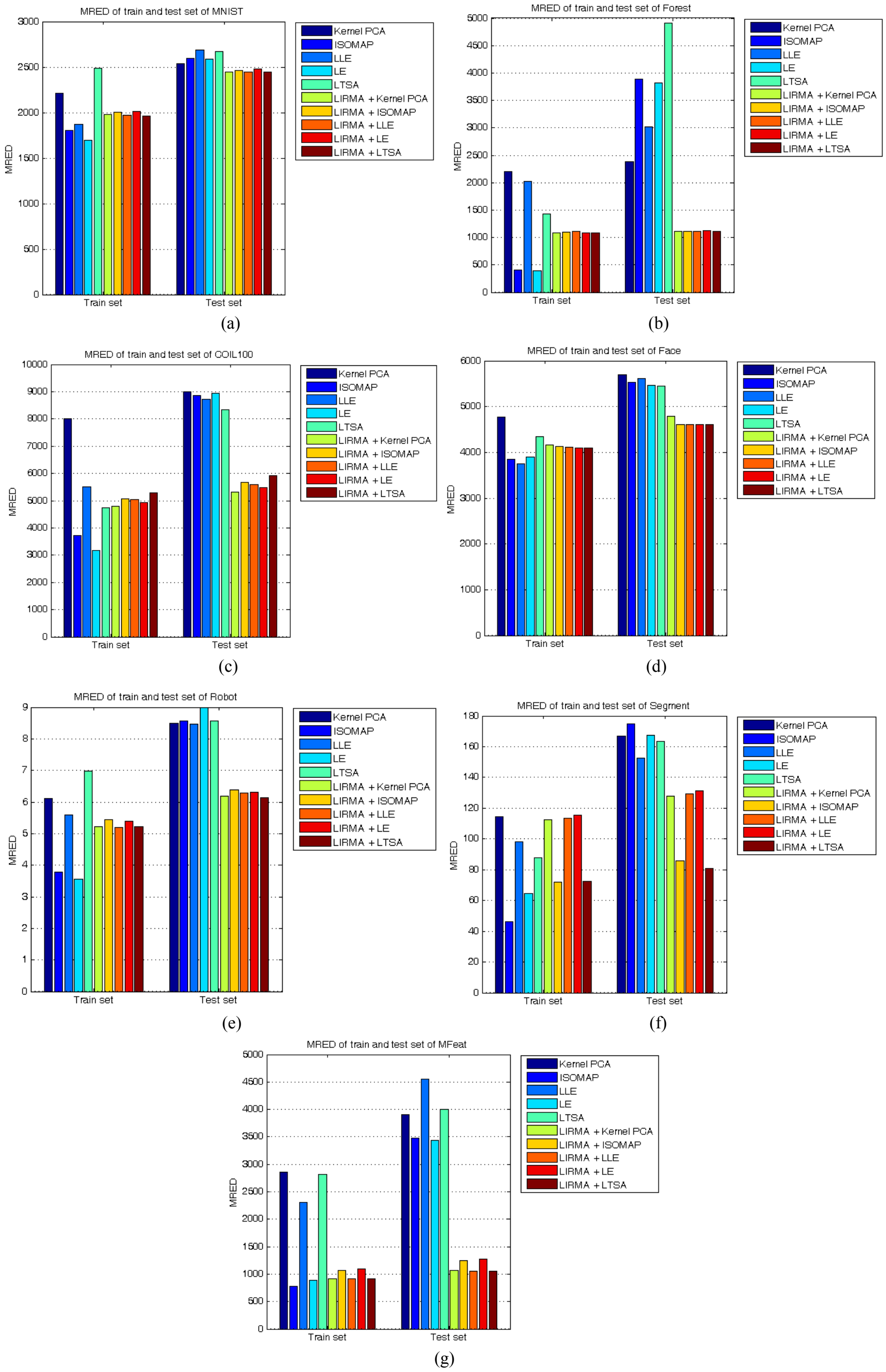
It is explained in Sections 2 and 4.3 that the embedded dynamical system of LIRMA is estimated by applying all training samples in a batch form. Therefore, the result of our proposed method is robust to the noise and minor disordering. Additionally, LIRMA presents an acceptable tradeoff between train and test sets with respect to the experiments which are presented in Fig. 7 and Table 3. Another advantage is that LRMA has low variance when different methods are applied to find the order of patterns. However, in case of Segment dataset, low number of samples increases the sensitivity to dis-ordering (Fig. 7(f) and Table 3). Similar to Segment, Face includes low number of samples. However, by considering the ratio between variance to mean, its variance ratio for comparison methods is lower than Segment (Table 3). As a result, LIRMA confronts less dis-ordering in case of Face dataset.

As explained in Sections 1 and 2, one of the disadvantages of applying graph based methods is missed training samples. The number of missed training samples for LLE and ISOMAP are 3 (MNIST), 2 (Forest), 426 (CIOL100), 8 (Face), 742 (Robot), 4 (Segment) and 6 (MFeat). In case of LE, the number of missed training samples are 126 (CIOL100) and 742 (Robot). The other methods have no missed samples.

Although experimental results confirm that LIRMA is a topology preserving method according to quantitative measures, we do not claim that our proposed method is the optimal candidate for dimensionality reduction. For example, it would be better to apply graph based methods for datasets which are not compatible with Riemannian manifold assumption (e.g. open cube). Additionally, when inductivity of a manifold is not important in a special task, an over-fitted graph based approach is a better option.

6. Conclusion and Future Works

In this paper, we presented a method for Learning Inductive Riemannian Manifold in Abstract form (LIRMA) by modeling embedded dynamical system. By taking the advantage of LIRMA, the global underlying structure of dataset is determined using a weighted summation of rich set of functions (i.e. $t^{\alpha}\exp(\lambda t)$, $t^{\alpha}\exp(at)\sin(bt)$ and $t^{\alpha}\sin(bt)$ ($\alpha \in \mathbb{Z}^{+}$ and $\lambda, a, b \in \mathbb{R}$)). LIRMA is applied for high dimensional, complex, real world datasets which is a challenging task for methods such as principal surfaces and polynomial PCA.



(a) (b) (c) (d) (e) (f) (g)

Fig. 7. (a)–(g) illustrate the performance of Kernel PCA, ISOMAP, LLE, LE, LTSA and LIRMA to train and test of MNIST, Forest, CIOL100, Face, Robot, Segment and MFeat respectively. The order of patterns in LIRMA is estimated using the comparison methods (i.e. LIRMA + ISOMAP, LIRMA + LLE, etc.).

Table 3
Mean and standard deviation of MRED measures computed for LIRMA and the other comparison methods.

MRED	MNIST	Forest	COIL100	Face	Robot	Segment	MFeat
Tr-Other	2013.6 ± 325.6	1290.4 ± 862.1	5032.6 ± 1891.1	4119.4 ± 427.9	5.20 ± 1.4	82.32 ± 27.0	1930.5 ± 1020.8
Tr-LIRMA	1986.6 ± 22.4	1095.1 ± 14.0	5025.0 ± 185.3	4116.6 ± 27.9	5.29 ± 0.1	97.14 ± 22.8	981.4 ± 92.2
Ts-Other	2615.5 ± 60.0	3607.3 ± 955.8	8770.4 ± 262.4	5544.8 ± 104.4	8.61 ± 0.2	165.0 ± 8.0	3872.9 ± 457.1
Ts-LIRMA	2456.0 ± 17.6	1114.8 ± 4.8	5599.0 ± 224.7	4645.0 ± 81.5	6.25 ± 0.1	111.02 ± 25.2	1138.4 ± 112.9

Tr: Train set. Ts: Test set. Other: mean and standard deviation is computed on MRED of Kernel PCA, ISOMAP, LE, LLE and LTSA. LIRMA: mean and standard deviation is computed on MRED of LIRMA + Kernel PCA, LIRMA + ISOMAP, LIRMA + LE, LIRMA + LLE and LIRMA + LTSA.

With respect to the abstract form of the underlying structure, LIRMA is used inductively with low computational and memory complexities. Unlike non-linear graph based approaches, LIRMA represents all of patterns in low dimensional space without missed samples. The proposed method is able to predict the structure of a dataset even beyond the range of training samples which is a unique ability in comparison to the state of the art non-linear dimensionality reduction methods.

In this paper also Mean Relative Euclidian Distances (MRED) is introduced as a local quantitative measure to evaluate the performance of a dimension reduction method. MRED is defined based on Euclidian distance which holds more topological information in comparison to the rank measurement. Additionally, MRED does not have the effect of false ranking of Mean Relative Rank Errors (MRRE). As a result, it is a proper option to evaluate how much a dimension reduction method preserves local topological information of a dataset.

With respect to the MRED measurement, LIRMA presents an acceptable tradeoff between train and test sets in real world datasets. Its performance as an inductive manifold is considerably better than the other methods for test sets. Since the embedded dynamical system is estimated with respect to all of the training samples in a batch form, the estimated AR model is robust to the noise and dis-orderings in the sequence of patterns. Therefore, LIRMA results lower variance when it is applied with different methods for finding the order of patterns. Applying LIRMA in classification task is one of the proposed future works. Additionally, using non-linear assumptions for modeling the embedded dynamical system can increase the performance of LIRMA.

References

Belkin, M. (2003). *Problems of learning on manifolds*. PhD thesis, Department of Mathematics, University of Chicago.

Burges, Ch.J.C. (2009). *Dimension reduction: a guided tour*. Microsoft research technical report, MSR-TR-2009-2013.

Chen, J. (2002). An improvement on tree selection sort. In: Mount, D.M., Stein, C. (Eds.), *Algorithm Engineering and Experiments*. Springer, Berlin.

El-Rewini, H., Abd-El-Barr, M. (2005). *Advanced Computer Architechture and Parallel Processing*. Willey, New York.

Face Dataset. AT&T Laboratories, Cambridge (2002).
`http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`.

Haefner, J.W. (2005). *Modeling Biological Syatems, Principles and Applications*, 2nd ed. Springer, Berlin.

Izenman, A.J. (2008). *Modern Multivariate Statistical Technics*. Springer, Berlin.

Jolliffe, I.T. (2002). *Principal Component Analysis*. 2nd ed. Springer, Berlin.

Karbauskaitė, R., Dzemyda G. (2009). Topology preservation measures in the visualization of manifold-type multidimensional data. *Informatica*, 20(2), 235–254.

Kayo, O. (2006). *Locally Linear Embedding Algorithm Extensions and Applications*. Faculty of Technology, Department of Electrical and Information Engineering, University of Oulu.

Kietzmann, T. C., Lange, S., Riedmiller, M. (2008). Incremental GRLVQ: learning relevant features for 3D object recognition. *Neurocomputing*, 71(13–15), 2868–2879.

Law, H.Ch. (2006). *Clustering, dimensionality reduction and side information*. PhD thesis, Department of Computer Science and Engineering, Michigan State University.

Lee, J.A., Verleysen, M. (2007). *Nonlinear Dimensionality Reduction*. Springer, New York.

Li, H., Jiang, H., Barrio, R., Liao, X., Cheng, L., Su, F. (2011). Incremental manifold learning by spectral embedding methods. *Pattern Recognition Letters,* 32(10), 1447–1455.

Lin, T., Zha, H. (2008). Riemannian manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5), 796–809.

Ljung, L., Glad, T. (1994). *Modeling of Dynamic Systems*. Prentice Hall, New York.

Masahiro, M. (2004). *Dimension reduction methods*. Humboldt-Universität Berlin, Center for Applied Statistics and Economics (CASE) 2004(15). http://hdl.handle.net/10419/22189.

McGuire, G., O'Farrell, A.G., Redmond, D.B., Watson, R.O., Wraith, D.J. (2002). *Maynooth Mathematical Olympiad Manual*. Logic Press.

MNIST dataset. http://yann.lecun.com/exdb/mnist/.

MTDR v0.8b (2012). *Matlab Toolbox for Dimensionality Reduction (v0.8b)*. By Maaten, L.V.D., Delft University of Technology.

Nene, S.A., Nayar, Sh.K., Murase, H. (1996). *Columbia Object Image Library (COIL 100)*. Technical report No. CUCS-006-96, Department of Computer Science, Columbia University.

Ogata, K. (1995). *Discrete Time Control Systems*, 2nd ed. Prentice Hall, New York.

Ogata, K. (2002). *Modern Control Engineering*, 4th ed. Prentice Hall, New York.

Ozertem, U., Erdogmus, D. (2011). Locally defined principal curves and surfaces. *Journal of Machine Learning Research*, 12, 1249–1286.

Raducanu, B., Dornaika, F., (2012). A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45(6), 2432–2444.

Sharma, A., Paliwal, K.K. (2007). Fast principal component analysis using fixed-point algorithm. *Pattern Recognition Letters*, 28(10), 1151–1155.

Sheldon, A., (1997). *Linear Algebra Done Right*. 2nd ed. Springer, Berlin.

Sishwick, P.A., (2007). *Hand Book of Dynamic System Modeling*, Taylor & Freancis, London.

Stewart, G. W. (1998). *Matrix Algorithms*, Vol. 1: *Basic Decompositions*. SIAM, Philadelphia.

Strang, G. (1988). *Linear Algebra and Its Applications*, 3rd ed. Harcourt Brace Jovanovich College Publishers.

Theodoridis, S., Koutroumbas, K. (2003). *Pattern Recognition*, 2nd ed. Elsevier Academic, Amsterdam.

Tuzel, O., Porikli, F., Meer, P. (2008). Pedestrian detection via classification on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10), 1713–1727.

UCI repository (2013). http://archive.ics.uci.edu/ml/.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, New York.

Zhang, P., Qiao, H., Zhang, B. (2011). An improved local tangent space alignment method for manifold learning. *Pattern Recognition Letters*, 32(2), 181–189.

Zhang, Zh., Zha, H. (2004). Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Journal Scientific Computing*, 26(1), 313–338.

E. Bavafaye Haghighi is PhD student in Artificial Intelligence at Computer Engineering department of Amirkabir University of Technology. Her MSc (in artificial intelligence) and BSc (in software engineering) degrees are from Tehran University and Amirkabir University of Technology respectively. She is semifinalist in Iranian National Olympiad in Mathematics (INOM-1996) and Iranian National Olympiad in Informatics (INOI-1997).

She received bronze medal in INOM-1997. In addition to Artificial Intelligence, she passed several courses in the field of system identification, modeling and control of linear and non-linear dynamical systems. Her research interests include pattern recognition and machine learning (supervised and non-supervised learning, dimensionality reduction, theoretical aspects of machine learning and the advantages of applying dynamical systems in machine learning), image processing, bioinformatics, modeling non-linear dynamical systems and chaos.

M. Rahmati received the MSc in Electrical Engineering from the University of New Orleans, USA in 1997 and the PhD degree in Electrical and Computer Engineering from University of Kentucky, Lexington, KY USA in 2003. He is currently an associate professor at the Computer Engineering Department, Amirkabir University of Technology (Tehran Polytechnic). His research interests are in the fields of pattern recognition, image processsing, bioinformatics, video processing, and data mining. He is the research coordinator of the department and he is a member of IEEE Signal Processing Society.

G. Palm studied mathematics at the Universities of Hamburg and Tübingen. After his graduation in mathematics he worked at the Max-Planck-Institute for Biological Cybernetics in Tübingen on the topics of nonlinear systems, associative memory and brain theory. In 1983/1984, he was a fellow at the Wissenschaftskolleg in Berlin. From 1988 to 1991 he was professor for Theoretical Brain Research at the University of Düsseldorf. Since then he is professor for computer science and director of the Institute of Neural Information Processing at the University of Ulm. His research topics in computer science include information theory and applications of artificial neural networks in speech, vision, robotics, sensor-fusion and pattern recognition.

S. Shiry Ghidary is an assistant prof. at Amirkabir University of Technology. He received his BSc degree in Electronic engineering from Amirkabir University of Technology in 1990, his MSc degree in computer architecture from same university in 1994 and his PhD in Artificial Intelligent Systems from Kobe University in 2002. He is in charge of Amirkabir Robotic Research Center. His research interests include Machine Learning, AI, Robotics, Mechatronics, Machine Vision, and Cognitive science.

Induktyvios abstrakčios formos Rymano daugdaros mokymasis, modeliuojant įterptąsias dinamines sistemas

Elham BAVAFAYE HAGHIGHI, Mohamad RAHMATI, Guenther PALM,
Saeed SHIRY GHIDARY

Daugdaros mokymosi algoritmai negali išgauti duomenų aibių struktūrų abstrakčios formos. Be to, algoritmai nėra našūs analizuojant sudėtingus duomenis. Šiame straipsnyje pasiūlytas induktyvios abstrakčios formos Rymano daugdaros mokymosi metodas, kuriame šablonų struktūra nustatoma, sprendžiant įterptąją dinaminę sistemą. Siekiant sumodeliuoti atitinkamą sistemą, šablonų sekų tinkamumas įvertinamas topologijos išlaikymo metodu. Pasiūlytas metodo privalumas yra tas, kad tai nesudėtingas induktyvus metodas. Be to, eksperimentais patvirtinta, kad pagal vertintą kokybinį matą, metodas išlaiko topologiją.