BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 01/06 pp235–258
Volume 3, Number 4, December 2020
DOI: 10.26599/BDMA.2020.9020011

Survey on Lie Group Machine Learning

Mei Lu and Fanzhang Li*

Abstract: Lie group machine learning is recognized as the theoretical basis of brain intelligence, brain learning, higher machine learning, and higher artificial intelligence. Sample sets of Lie group matrices are widely available in practical applications. Lie group learning is a vibrant field of increasing importance and extraordinary potential and thus needs to be developed further. This study aims to provide a comprehensive survey on recent advances in Lie group machine learning. We introduce Lie group machine learning techniques in three major categories: supervised Lie group machine learning, semisupervised Lie group machine learning, and unsupervised Lie group machine learning. In addition, we introduce the special application of Lie group machine learning in image processing. This work covers the following techniques: Lie group machine learning model, Lie group subspace orbit generation learning, symplectic group learning, quantum group learning, Lie group fiber bundle learning, Lie group cover learning, Lie group deep structure learning, Lie group semisupervised learning, Lie group kernel learning, tensor learning, frame bundle connection learning, spectral estimation learning, Finsler geometric learning, homology boundary learning, category representation learning, and neuromorphic synergy learning. Overall, this survey aims to provide an insightful overview of state-of-the-art development in the field of Lie group machine learning. It will enable researchers to comprehensively understand the state of the field, identify the most appropriate tools for particular applications, and identify directions for future research.

Key words: Lie group machine learning; Lie group subspace orbit generation learning; quantum group learning; symplectic group learning; Lie group fiber bundle learning

1 Introduction

Machine learning, as a branch of artificial intelligence, has been playing an increasingly important role in scientific research in recent years[1–4]. The development of many disciplines, such as bioinformatics, physics, chemistry, material analysis, and so on, requires intelligent methods to enrich the content of related disciplines. These intelligent methods rely on machine learning as the most basic and key core technology. As a new learning method in the field of machine learning, Lie group Machine Learning (LML) not only inherits the advantages of machine learning but also integrates the ideas of Lie groups to form an innovative learning paradigm. Since LML was put forward in 2004, it has attracted wide attention in the world. Relative to traditional machine learning methods, LML exhibits the characteristics of differential manifolds and groups. It provides not only a geometric representation of data but also specific algebraic solutions. A group preserves system completeness, i.e., the differential provides a specific algebraic calculation method, while a manifold provides a geometric representation method,

• Mei Lu is with the School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China and is also with the School of Computer Science and Technology, Jiangsu Normal University, Xuzhou 221000, China. E-mail: louisazhaoxiaolu@sina.com.
• Fanzhang Li is with the School of Computer Science and Technology, Soochow University, Suzhou 215006, China. E-mail: lfzh@suda.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2020-05-14; accepted: 2020-06-09

which is in accordance with the cognitive model combining qualitative representation and quantitative representation in cognitive theory. LML follows the research mechanism of using "cognitive science as its foundation, mathematics for its methods, theoretical calculation as its standard, analysing rules of data as its goal, and computer technology as its way to construct machine learning theory, technology, methods, and applications". LML realizes the cognitive view of "solving complex problems with simple models". It has a unique advantage with regard to the use of model continuity theory to solve realistic discrete data. It is also advantageous in modeling voluminous data with minimal data constructs and solving unstructured data by using structured methods; nonlinear data by using linear methods; and the two-way mechanism of perception and cognition, the correspondence between individuals and the environment, and the dynamic permission relationship by using qualitative and quantitative methods.

Especially in the rapid development of big data, internet of things, cloud computing, cognitive science, space science, materials science, and so on, LML breaks the old thought that group theory only exists in the mind and cannot be well associated with real problems; it also truly embodies the scientific concept that "Mathematics is not everything, but without mathematics, nothing can be done".[3]

This paper is organized as follows. Section 2 introduces the concepts of Lie algebras and Lie groups. Section 3 introduces the LML model. Section 4 introduces LML algorithms, including unsupervised LML, supervised LML, and semisupervised LML. Section 5 introduces the application of LML in 3D image processing. Section 6 introduces neuromorphic synergy learning. Section 7 provides the conclusion.

2 Introduction to Lie Algebras and Lie Groups

In this section, we collect several basic concepts of Lie algebras and Lie groups[5–7].

Definition 1 **A k-Lie algebra or Lie algebra over k** ($k \subseteq \mathbb{R}$ or $k \subseteq \mathbb{C}$) consists of a vector space a over a field k, together with a k-bilinear map $[,]$: $a_v \times a_v \to a_v$ called the Lie bracket, such that for $x, y, z \in a_v$,

$[x, y] = -[y, x]$, (Skew symmetry)

$[x, [y, z]] + [y, [x, z]] + [z, [x, y]] = 0$. (Jacobi identity)

Here k-bilinear means that for $x_1, x_2, x, y_1, y_2, y \in a_v$ and $r_1, r_2, r, s_1, s_2, s \in k$,

$$[r_1 x_1 + r_2 x_2, y] = r_1[x_1, y] + r_2[x_2, y],$$

$$[x, s_1 y_1 + s_2 y_2] = s_1[x, y_1] + s_2[x, y_2].$$

A simple conclusion exists about the general form of Lie brackets for Lie algebra, including closure, bilinearity, alternating, and Jacobi identity.

Closure: $[x, y] \in a_v$;

Bilinearity: $[ax + by, z] = a[x, z] + b[y, z]$;

Alternating: $[x, x] = 0$;

Jacobi identity: $[x, [y, z]] = [z, [y, x]] = [y, [z, x]]$.

To establish the definition of a Lie group, we first define a smooth map.

Definition 2 A continuous map g: $V_1 \to V_2$, where each $V_k \subseteq \mathbb{R}^{m_k}$ is open, is *smooth* if it is infinitely differentiable. A smooth bijection g is a *diffeomorphism* if its inverse g^{-1}: $V_2 \to V_1$ is also smooth.

Definition 3 Let G be a smooth manifold which is also a topological group with multiplication map mult: $G \times G \to G$ and inverse map inv: $G \to G$ and view $G \times G$ as the product manifold. Then G is a *Lie group* if mult and inv are smooth maps.

Lie group theory is the fundamental representation of a space of transformations. The three central elements of the Lie group framework are Lie distance, intrinsic mean, and principal geodesics. Lie distance is a measure of the similarity of two transformations. The intrinsic mean represents the "average" of a set of transformations, i.e., the transformation that minimizes the Lie distance to all transformations in the set. A geodesic is a $1-d$ subspace of transformations that is the shortest path between two transformations. A principal geodesic is one that accounts for the maximum variation in a set of transformations along a path; it is analogous to the principal component of a covariance matrix.

The relationship between Lie groups and Lie algebras is that the tangent space of group G at identity e, T_e, is called the Lie algebra. The exponential map exp is a mapping from the Lie algebra elements to the Lie group elements. The logarithmic map log takes group elements onto a tangent plane. The Lie group distance between two points is defined as $d(x_1, x_2) = \left\|\log(x_1^{-1} x_2)\right\|$, where $\|\cdot\|$ is the Frobenius norm of the resulting algebra element[8, 9]. The commonly parameter of Lie group and Lie algebras are listed in Table 1.

A set of *orthogonal matrices* denoted as $O(n)$ is an example of a matrix Lie group[10]. Orthogonal matrices are known to have a determinant of either

Table 1 Parameter of Lie groups and Lie algebras.

Name	Symbol	Lie algebra
Euclidean space	i^n	i^n
General linear group	GL(n, F)	gl(n, F)
Special linear group	SL(n, F)	sl(n, F)
Orthogonal group	$O(n, F)$	$o(n, F)$
Special orthogonal group	SO(n, F)	so(n, F)
Symplectic group	SP(n,F)	sp(n, F)
Unitary group	$U(n)$	$u(n)$
Special Unitary group	SU(n)	su(n)
(p, q) type Unitary group	$U(p, q)$	$u(p, q)$
Special (p, q) type Unitary	SU(p, q)	su(p, q)

$+1$ or -1; however, they are not connected on a manifold. Thus, a special subgroup of orthogonal matrices whose determinant is equal to $+1$ is usually specified. Formally, let SO(n) be a set of $n \times n$ orthogonal matrices defined as

$$\mathrm{SO}(n) = \{Y \in \mathbb{R}^{n\times n}:\ Y^{\mathrm{T}}Y = I,\ \det(Y) = 1\}.$$

The additional determinant constraint $(\det = 1)$ ensures that all matrices in this group are rotation matrices. This set of matrices is called a special orthogonal group, which is also an example of a Lie group.

A pictorial summary of a Lie group, Lie algebra, and exponential and logarithmic mappings is presented in Fig. 1. In Fig. 1, Y is an $n \times n$ matrix parametrized by a curve $f(t)$ in SO(n) such that $f(0) = I$, and $f'(0) = \Delta$.

3 Lie Group Machine Learning Model

Xu and Li[11–14] discussed the general LML model and established a number of common classifiers, including the LML model, Lie group algebra machine learning model, Lie group geometric machine learning model, geometric learning algorithm for Dynkin graphs in LML, linear classifier design of LML, and LML SO(3) classifier.

As a Lie group is an analytic manifold with a group structure and as group operations are analytic, the LML norm can be used to analyze the dimensions, compactness, connectivity, nilpotency, subgroups, cosets, quotient groups, quantum groups, and so on. These basic structures serve as the foundation for the design of the LML algorithm.

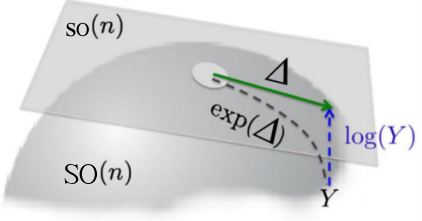

Fig. 1 Example of a special orthogonal group: mappings between a Lie group and a Lie algebra using the exponential and logarithmic maps.

Lie group machine learning[15,16]. Let $G \subseteq \mathbb{R}^D$ denote the input space, $M \subseteq \mathbb{R}^d$ denote the output field, and $D > d$. The definition of mapping φ of *the left effect of G on M* is as follows:

$$\varphi : G \times M \to M,$$
$$g, x \to \varphi(g, x).$$

The following should hold

$$\varphi(g_1, \varphi(g_2, x)) = \varphi(g_1 g_2, x), \varphi(e, x) = x.$$

The left effect of LML is shown in Fig. 2.

Let $\varphi(g, \cdot) = \varphi_g(\cdot)$. For any $g \in G$, M is a diffeomorphism exchange:

$$\varphi_g : M \to M,$$
$$x \to \varphi(g, x).$$

Mapping set $\{\varphi_g, g \in G\}$ satisfies $\varphi_{g_1} \cdot \varphi_{g_2} = \varphi_{g_1 g_2}$, $\varphi_e = \mathrm{id}_M$, where id_M is an identity function $\mathrm{id}_M : M \to M$.

Similarly, we can define the right effect of group G on M:

$$\psi : M \times G \to M,$$
$$x, g \to \psi(x, g).$$

satisfying $\psi(\psi(x, g_1), g_2) = \psi(x, g_1 g_2), \psi(x, e) = x$.

The right effect of LML is shown in Fig. 3.

The diffeomorphism exchange of the right effect from G to M is easy to use:

$$\psi_g : M \to M,$$
$$x \to \psi(x, g).$$

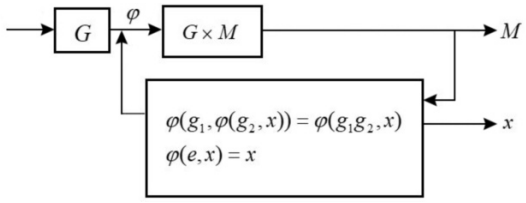

Fig. 2 Left effect model of Lie group machine learning.

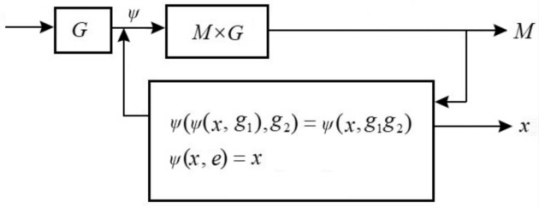

Fig. 3 Right effect model of Lie group machine learning.

Algebra model of LML[4, 15, 16]. According to the concept of LML, four different mathematical structures, namely, Lie algebra, one-parameter subgroup, left invariant vector field, and left invariant manifold; they are defined as a natural correspondence, i.e., Lie algebras $\tilde{=}$ [left invariant vector field] $\tilde{=}$ [left invariant flaw] $\tilde{=}$ [one-parameter subgroup]. These mathematical structures serve as the foundation for the study of the Lie algebra root calculation model, decomposition model, and classification model of LML. The algebraic model of LML is shown in Fig. 4.

Geometric model of LML[4, 15, 16]. Using some of the geometric properties of a Lie group, we construct the geometric model of LML to facilitate the learning of system representation and measurement. From the left invariance of the orthogonal vector field group, all left translations are isometric transformations of the Riemannian space. From the invariance of the inner product, all right translations are isometric transformations. Let $T_e(G)$ be a tangent space of a unit point, i.e., let the Lie algebra g of G, $T_a(G)$, be the tangent space of any sample point a, and let dl_a and dr_a be the linear mappings induced by the left transformation l_a and the right transformation r_a respectively in the tangent space. Thus, geometric model structure shown in Fig. 5 is obtained.

Lie group metrics. A geodesic is important in studying manifolds in a high-dimensional space as a means of measurement. It is the local shortest path in a Riemann manifold. As manifolds are locally Euclidean, the distance of a geodesic cannot be easily calculated. The commonly used technique is to replace the small part of a manifold with a tangent plane. For example, in Ref. [17], the 3D nose-shaped net for a human gender and ethnicity classification algorithm is based on a 3D nose-shaped organizational structure. A nose measurement method for determining the distances between different noses is designed to construct the 3D nose-shaped net. The shape space of the nose curves can be regarded as M/G, $M/G = \{[p] | p \in M\}$, and G is a Lie group that is acting smoothly on a manifold, which is used to build the nose curves' shape space for nose similarity measurements. Gender and ethnicity classification results are achieved in the 3D nose-shaped net simultaneously. Boutellaa et al.[18] presented a covariance matrix-based fall detection method that uses multiple wearable sensors. This method employs the covariance of the raw signals and the nearest neighbor classifier. Instead of Euclidean metrics, Boutellaa et al.[18] used geodesic metrics, which provide higher fall detection accuracy. Heider et al.[19] presented the SO(3) invariance of an informed graph-based deep neural network for an anisotropic elastoplastic material algorithm. In this work, the spectral form is applied to represent tensors, and three metrics, namely, the Euclidean distance between the Euler angles, the distance from the identity matrix, and the geodesic on the unit sphere in Lie algebra, are employed to constitute the objective functions for the supervised machine learning. Experimental results reveal that using a loss function based on a geodesic on the unit sphere in Lie algebra, together with an informed and directed graph, yields more accurate rotation predictions than other tested approaches.

The Geodesic Distance algorithm of Lie group Machine Learning (GDoLML) is described in

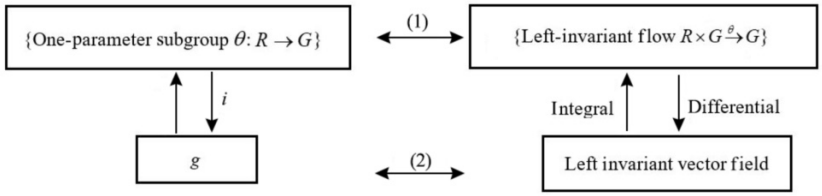

Fig. 4 Algebraic model of Lie group machine learning.

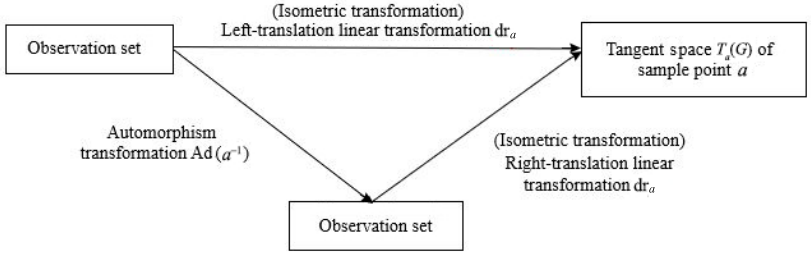

Fig. 5 Geometric model of Lie group machine learning.

Algorithm 1[4,15].

In fact, Algorithm 1 is a general algorithm for calculating geodesic distance. In practical applications, the common way to calculate the geodesic distance between two points x_1 and x_2 is as follows:

$$d(x_1, x_2) = \left\| \log(x_1^{-1} x_2) \right\|,$$

Geometric learning algorithm for Dynkin graphs in LML[4,15,16]. This algorithm shows that if the learning problem is solvable, then the Dynkin diagram can be drawn corresponding to the learning space. According to the Dynkin diagram formed by the data, the Lie group corresponding to the special unit group can be found, and the corresponding solution can be established. Therefore, the Lie group as the main basis for machine learning theory can be used.

4 Lie Group Machine Learning

Similar to traditional machine learning algorithms, LML algorithms can be divided into three broad categories: supervised learning, semisupervised learning, and unsupervised learning. Supervised learning is useful in cases where a label is available for a training set but is missing and needs to be predicted for other instances. Semisupervised learning uses labeled and unlabeled data to improve supervised learning. The goal is to learn a predictor that predicts future test data better than the predictor learned from the labeled training data alone. The reason for this improvement is that a large amount of unlabeled data enable the system to model the inherent structure of the data accurately. Unsupervised learning is useful in cases where the challenge is to discover implicit relationships in a given unlabeled dataset.

Algorithm 1 GDoLML

Input: Sample point a

Output: The distance between all points in the coordinate neighborhood of a and a

1: Initialization: (X, τ) is a topological space, $x \in X$.
2: Generate the neighborhood U_a of the sample point a.
3: Analyze the neighborhood information of a in the sample set of points to take an automorphism under the invariant inner product, and calculate the value of $g_{i,j}(a)$.
4: Substitute the resulting value $g_{i,j}(a)$ into $D(a,b) = \sqrt{\sum_{i,j=1}^{n} g_{i,j}(a)(a_i - b_i)(a_j - b_j)}$ to calculate the distance between each point in U_a and a.
5: Output all values of distance (the smaller the distance, the closer the result to be learned)

4.1 Supervised Lie group machine learning

In this section, we analyze and summarize the algorithms of supervised LML.

4.1.1 Classification of text based on Lie group machine learning

In the traditional classification algorithm, the similarity between two documents often needs to be measured; commonly used metrics include the Euclidean distance, vector angle cosine, and so on. However, because manifolds are locally Euclidean, calculating the distance of a geodesic directly in a manifold is difficult. To address this issue, Li et al.[4] and Xu and Li[14] proposed Lie Group Classification (LGC) for text classification. In this method, the vector space of a document is embedded in a differential manifold, and each document corresponds to a point on the manifold. Moreover, the geodesic distance is used instead of the Euclidean distance to measure the similarity between two documents. Experimental results show that the recall and precision of text categorization with the LGC algorithm are higher than those with the Recurrent Neural Network (RNN) and Support Vector Machine (SVM). The LGC algorithm is described in Algorithm 2.

With regard to the metrics for text documents, Lebanon[20] considered learning a Riemannian metric

Algorithm 2 LGC

1: The training document vector is described according to the set of feature words. Suppose that the feature set of document i is $t_i = (t_{i1}, t_{i2}, \ldots, t_{in})$ and the corresponding weight of document i is $w_i = (w_{i1}, w_{i2}, \ldots, w_{in})$. Then, the vector representation of document i is $d_i = (t_{i1}, w_{i1}; t_{i2}, w_{i2}, \ldots, t_{in}, w_{in})$. Thus, all document vectors form a vector space, which is embedded in a differential manifold. Each document is a point on the manifold.
2: For a new document, determine the vector representation of the document according to the feature set.
3: Generate the coordinate neighbourhood U_a of a new document a, i.e., the document set S, establishing its tangent space local coordinate system, computing the geometric distance of every document S_k and new document a in set S. $D(a, s_k) = \sqrt{\sum_{i,j=1}^{n} g_{i,j}(a)(a_i - s_{k,i})(a_j - s_{k,j})}$.
4: Compute the weight of each point in S for the classes of document c_x, $W(d, c_x) = \sum_{s_k \in S} D(a, s_k) \times f(s_k, c_x)$, where $f(s_k, c_x)$ is the class function. Thus, if s_k belongs to class c_x, its function value is 1; otherwise, it is 0.
5: Comparing the weights of the classes, the document a belongs to the largest weight of the class.

associated with a given differentiable manifold and a set of points. The author chose a metric from a parametric family that is based on maximizing the inverse volume of a given dataset of points. This metric is related to maximum likelihood under a model that assigns probabilities that are inversely pro-portional to the Riemannian volume element. A metric in a multinomial simplex is learned as follows: the metric candidates are pullback metrics of the Fisher information under a Lie group of transformations. When applied to text document classification, the resulting geodesic distance outperforms the tf-idf cosine similarity measure.

4.1.2 Lie group subspace orbit generation learning

Chen[15] and Chen and Li[16] expanded the content of LML and studied the learning problem of Lie groups from a graph structure. According to the independent division hypothesis axiom, consistency hypothesis axioms and the basic facts of each element in the observation space set can be described by a set of real independent parameters, in addition to the characteristics of the geometric model of LML. On the basis of the LML space, the concept of the LML subspace orbit generating a lattice is established. Generating lattice theory and the learning algorithm related to lattice theory, which are derived from multiple identical dimensions or orbits and rank subspaces, are given under the action of various typical groups in the observation set. These groups include the orbit generating lattice learning algorithm of the general linear group, orbit generating lattice breadth-first and depth-first learning algorithms, heuristic learning algorithm, and other typical groups of learning algorithms.

In LML, an orbit is defined as the learning path in the learning subspace that encompasses a series of learning operations (operators) in the learning process. All orbits form the entire learning subspace; a state subset and the operator used constitute an orbit, and an orbit constitutes a partial chain.

The purpose of Lie group subspace orbit generation learning is to find the optimal path from the initial state to the target state. If each state is treated as a node and each arc represents an operator, then the learning subspace can be represented by a directed graph. The node relationship is embodied in the specific problem and can be understood as the sample to be learned in the node; here, the specified operator is the best, i.e., conducive to obtaining the optimal orbit. Figure 6 shows the relationship between learning space, learning subspace, orbit, and target orbit.

The depth-first orbit generation algorithm is described as Algorithm 3.

In the process of generating the orbit, the key step is how to determine the nodes to be extended. Chen[15] and Chen and Li[16] proposed two methods, namely, the breadth-first orbit generation algorithm and the depth-first orbit generation algorithm. For the breadth-first orbit generation algorithm, the given weight known as the first-level weight can be used for a simple calculation to create a new weight, i.e., a list of all the weights in the second-level orbit. The process is repeated, and the weight list of any level can be obtained using the weights and simple calculations of the next lower layer. For the depth-first orbit generation algorithm, the orbit generation can be processed in a different way. The orbit can be seen as a tree, and according to the depth of priority for calculation, the deeper levels of the weight of the orbit can be reached as far as possible. A weight is stored on the basis of certain specifications before being traced back to the uncomputed "branch".

To identify a learning orbit effectively, Chen and Li[21] presented the LML subspace orbit generation learning algorithm with heuristic information. Using special information related to the specific field of learning, the current study is guided in the most promising direction. Chen and Li[16] further discussed the LML learning subspace orbit generation lattice

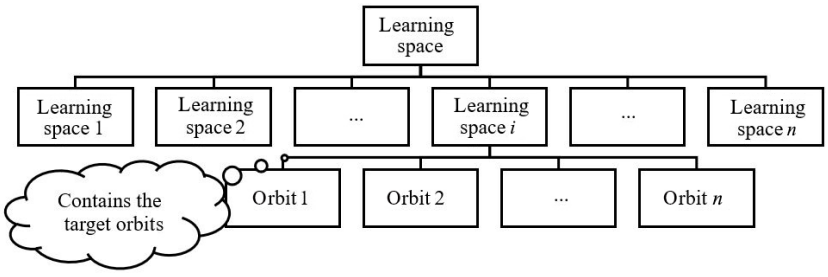

Fig. 6 Diagram of learning space, learning subspace, orbit, and target orbit.

Algorithm 3 Orbit Generation Depth-First algorithm (OGDF)

Data structure: Levelcount: An integer array that counts for each layer; weight-structure stack.
Each weight-structure contains:
$\omega = (\omega_1, \ldots, \omega_l)$: Weight vector;
level: Rank of ω;
from: An integer that labels the parent value;
i: Next index to be tested.
Input: $m = (m_1, \ldots, m_l)$: Control weight
D: Dynkin matrix
l: rank of G
Max level: Calculate the maximum level of the track
Output: Weight vector list and four related integers.
The first integer indicates the position of ω in the list;
The second integer is the hierarchy of ω;
The third and fourth integers indicate the weight vector and the calculation of ω;
Process: Make level = 0, levelcount[level] = 1, i = 1;
Make from = 1; // Location of paternity values
Make to = 1; //Position of the new weight, i.e., the total number of weights computed
Output m; been learned
Put m, level, from, i into the stack;
while (The stack is nonempty) **do** {
  Out ω, level, from, i of stack;
  while ($i \leqslant l$) **do**
    if ($\omega_i > 0$) **then** {
      if ($v_j \geqslant 0$ to $i < j \leqslant l$) **then** {
        //Store v?
        if ($i = 1 \leqslant j$) **then**
          Out ω, level, from, i+1 of stack;
        level + +;
        levelcount[level] + +;
        Output v, level, from, i, to;
        if (level $\geqslant$ max level) **then**
          jump out while loop;
        Copy v to ω;
        Let i be equal to the first element in D_i; }
      Otherwise i + +;
    }
  Otherwise i + +; }

learning algorithm under the action of the general linear group GLn(Fn), where Fn is a finite of n elements. The group elements formed by n-order regular matrices can be characterized by γ- parameters under certain conditions. Then, a continuous change of γ- parameters may produce the entire manifold. For the General Linear group on the sample dataset GLn(Fn), the n^2 elements of the matrix calibrate the points in the n^2-dimensional Euclidean space. This change in the n^2 parameters reflects the change relation from one point in the Euclidean space to another. Thus, GLn(Fn)'s subgroups of various γ parameters can be represented by some γ-dimensional subspaces in the n^2-dimensional Euclidean space. The change of data state in the learning system is proved by the transformation of the parameter. Starting from any element a, a continuous change through a parameter produces an arc that links any pair of spaces in a group, thereby forming the graph structure.

4.1.3 Quantum group learning

A noncommutative and non-cocommutative Hopf algebra is called a quantum group[22]. A quantum group is the generalized concept of the classical symmetry of Lie groups and Lie algebras. Quantum group theory has been widely used in many fields. For example, Nasios and Bors[22] applied these concepts to nuclear classification and achieved satisfactory results. Such achievement reflects the excellent opportunities for machine learning and provides useful examples for data analysis based on machine learning. In Ref. [23], He and Li proposed quantum group learning algorithms in the Lie group machine learning framework that apply quantum group theory to process data. They constructed a quantum group classifier with the operations of quantum symmetric transformation and covariant differentiation in a nonexchange space. They used these algorithms to deal with nonexchange and asymmetric data, such as the coding sequence in a gene, the position between atomic groups, and the pharmacophore in drug molecular design. These algorithms achieved better reality test performance than state-of-the-art methods.

The classification algorithm of quantum group classifiers in machine learning is described as Algorithm 4. The data of the sample set can be transformed into the classical quantum group that we need[23]:

SLq(n): quantum special linear groups;

Algorithm 4 Classification algorithm of quantum group classifier in machine learning: QGClassifier (Gq)

Input: Gq
Output: $\{1, -1\}$
1: Initialisation: Gq is a quantum group whose group elements are $\{X_1, X_2, \ldots, X_n\}$
2: **for** $i = 1$ to n **do**
3: **if** $\det_q X_i == 1$ **then**
4: return 1; //Gq is SLq(n) group//
5: **else**
6: return -1; //Gq is GLq(n) group//

GLq(n): quantum general linear groups.

The construction of a specific quantum group learning classifier is described below:

Step 1: Map the sample set to this nonempty set of Gq.

Step 2: Construct the corresponding quantum group structure according to Gq and test whether it is a quantum group.

Step 3: Calculate the classical quantum group with the established quantum group structure.

Step 4: Use the classification algorithm of quantum group classifier (Algorithm 4) to classify and train the appropriate classifier.

Step 5: Perform the instance test.

4.1.4 Symplectic group learning

Symplectic geometry and symplectic groups have been successfully applied to many fields, including the field of dynamics. For example, Wu et al.[24] applied the symplectic geometry algorithm to the implementation process of a Hamiltonian system, which is an important type of dynamical system. Feng and Qin[25] employed the symplectic algorithm in the dynamics of a multibody system and successfully solved the multibody system dynamics equation. Xu et al.[26] proposed the numerical solution of a differential strategy based on a Hamiltonian system and the symplectic algorithm.

In their further study of LML, Li et al.[4, 27] and Fu[28] chose a typical Lie group, i.e., the symplectic group, to design a symplectic group classifier. They developed three symplectic group classification methods for different modes of data organization. For the first one, they used the symplectic group matrix to construct the singular value of a corresponding image in the symplectic matrix space and then constructed the feature subspace of the image in the symplectic matrix space. Singular value decomposition is used to decompose an image into a linear combination of a base image with rank 1. They subsequently used geodesic distance to calculate the projection length of the identified samples in each feature subspace and classified the samples into classes of characteristic subspaces corresponding to the maximum projection length. For the second one, the nearest feature line method is extended to the matrix space, the feature lines are constructed directly in the symplectic matrix space by using two different face image matrices of each category, and the ability of two samples to characterize the respective categories is expanded. For the last one, the nearest neighbor characteristic line method is extended to the symplectic matrix space. When the symplectic matrix is transformed, the transformed symplectic matrix and several special symplectic matrices can be compared by referring to the transformation properties of the conformal symmetric standard matrix. If the matrix of the other image to be examined and the matrix obtained by the transcoding of the source image have the same form as the symplectic matrix, then the image to be examined and the source image are consistent, i.e., the purpose of recognition is achieved. This method can directly deal with the original face image matrix and does not require the matrix to be straightened into a high-dimension vector. Therefore, the computational complexity of high-dimensional vector extraction is avoided, and the problem of face recognition is solved using the symplectic group classifier.

The main recognition process of the symplectic group classifier in the process of image classification is as follows:

(1) In the training phase, perform singular value decomposition on the i-th training sample X_{ji} of the j-th class. Take the base image $A_{ji}(m) = u_{ji}^{m}(v_{ji}^{m})^{H}(m = 1, 2, \ldots, k)$ corresponding to the k largest singular values. A set of base images obtained from training samples X_{ji} constitute a discriminant function $g_{ji}(X)$ of the j-th class:

$$g(X) = \sqrt{\sum_{m=1}^{k} \alpha_{ji}^{2}(m)} = \sqrt{\sum_{m=1}^{k} \left|\langle A_{ji}(m)\rangle\right|^{2}},$$
$$j = 1, 2, \ldots, C, i = 1, 2, \ldots, N,$$

where $\alpha_{ji}(m) = \langle A_{ji}(m), X\rangle$ is the projection value of the m-th base image $A_{ji}(m)$ of the i-th training sample X_{ji} to be recognized by the sample image X for the j-th class.

(2) The classifier recognition process is to input the image X to be recognized into each discriminant function $g_{ji}(X)$ and to find the output of the discriminant function. Put X in the corresponding category of $g_{ji}(X)$ of the maximum output. If each class has multiple training samples, then calculate first the maximum length for each class: $h_j = \max_i(g_{ji}(X))$, $(j = 1, 2, \ldots, C; i = 1, 2, \ldots, N)$.

In this way, X can be classified into the c-th category.

4.1.5 Lie group covering learning

The technique of covering is an important tool for studying algebraic geometry, which includes coverage in research-based transformations, model maps, singularities, and so on. A single connected Lie group is a differential manifold. The idea of covering has been applied to many areas, such as data mining and machine learning[29–35]. The concept of covering in Lie groups can be introduced to deal with simply connected Lie groups, which can be covered with a differential manifold[33]. The transformation from a simply connected Lie group to its covering group is shown in Fig. 7.

For the simply connected Lie group G, a set of simply connected Lie groups (such as G_1' and G_2' in the graph) can be constructed to cover the unknown simply connected Lie group G obtained after mapping the samples in the same class. Then, space G comprising the simply connected Lie group G_1', G_2', etc. is a cover of the original simply connected Lie group G. As the cover groups G' and G are local isomorphic, the properties of covering group G' reflect the properties of the original Lie group. The samples in the same class belong to the same composite covering group. If the distance between sample points is measured by geodesic distance, then the distance of the different composite coverage groups can be used to determine the class of sample points.

In dealing with the multiple connectivity problem, many papers have presented different methods[30–35]. In Ref. [32], the exterior and interior of the domain were conformally mapped to unit disks and circle domains (unit disks with several inner disks removed), and a shape distance between shape signatures was defined to measure dissimilarities between shapes. In Ref. [30], the neural network approach for simply connected domains was extended to the case of multiconnected domains, and the self-organizing map technique was applied alternatively to the boundary and interior mesh nodes. In the case of multiconnected domains, this algorithm is applied to specify automatically the holes in a fixed mesh.

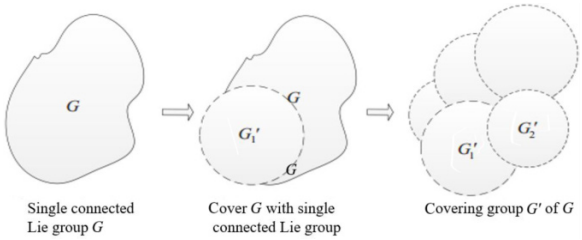

Fig. 7 Transformation from simply connected Lie group G to its covering group G', here G' is the covering group of Lie group G.

Guan[31] proposed the single connected covering algorithm of LML. The purpose of learning is to find the optimal path from the initial state to the target state. Each state is called a node, each arc represents an operator, and one state can be directed to another state. According to different problems, a specific weight can be set for the operation operator. A path from the initial state to the target state forms a cover. As the two Lie groups have local isomorphism, the topological properties of the original Lie group are maintained (both have the same Lie algebra). Therefore, Guan[31] proposed a simply connected covering algorithm for LML by solving the covering map problem. However, in many cases, the learning problem is not a one-to-one correspondence. It has multiple input spaces and output fields, as well as multiple inputs and outputs. To address this issue, Guan[31] proposed a multiconnected covering algorithm of LML by transforming the Lie group problem into its coverage group and using the isomorphic relation between the Lie group and the covering group[31].

Chen et al.[33] presented a multiconnected Lie group covering learning algorithm for image classification. According to the connectivity of Lie groups, they attempted to map the research objects with different category characteristics into the space of the multiconnected Lie group. On the basis of the homotropy equivalence of attachments on each simply connected Lie group, the study explored the equivalent representation of the optimal path for each category by covering ideas so as to present the category information of images by employing its multivalued representation.

The multiconnected Lie group covering learning algorithm for image classification is described as Algorithm 5[33].

Yan and Li[34] further proposed path optimization algorithms for covering learning. They discussed a geodesic curve for the optimal mapping of roads to minimize the correlation of roads from different connected spaces and maximize the correlation of roads within the same connected space.

Wu and Li[35] proposed a multi-Lie group kernel covering learning algorithm. The algorithm uses the algebraic and manifold structures of Lie groups. Data are mapped to multiple Lie groups, and the homotopic tracks can be covered according to the track relation

Algorithm 5 Multiply Connected Lie Group Covering Learning algorithm (MCLGCL)

Input: Sample set $U = \{u_1, u_2, \ldots, u_n \in \mathbb{R}^D\}$, the number of categories s.

Output: Covering center set C_i $(i = 1, 2, \ldots, s)$ of each simply connected domain, and the geodesic set R_i $(i = 1, 2, \ldots, s)$ of each simply connected covering.

1: Construct point clouds based on each sample of sample set U, and map these point clouds to SO(2) group which realizes the mapping from sample points to multiple connected Lie group space, then get the sample set $X = \{x_1, x_2, \ldots, x_n \in \mathbb{R}^D\}$ which is divided into S categories, namely $X = \{X^1, X^2, \ldots, X^S\}$.

2: Construct a simple connected cover in the sample set X^i $(i = 1, 2, \ldots, s)$. Take each sample x_j^i as the center to construct simple connected cover and record the geodesic radius r_j^i with each sample x_j^i as the center and the number of samples covered N_j^i.

3: Calculate the evaluation $f_j^i = N_j^i / r_j^i$ of each sample x_j^i, take the best evaluation sample x_j^i as the coverage center, add x_j^i to C_i, and add r_j^i to R_i.

4: Delete the samples covered by the selected center in X^i, and then repeat the above steps to find the best evaluation samples in the remaining samples of X^i until all the samples in X^i are covered. If all samples of all categories in X have been covered by the covering groups, the algorithm ends.

of Lie group samples on the manifold. Thus, these covering areas can present category information.

Wu and Li[35] also proposed an optimization algorithm for the multi-Lie group covering learning algorithm. Their previous multi-Lie group kernel covering learning algorithm reduces the intersection of roads and improves the correctness of classification for multiconnected spaces. However, the performance of the kernel learning algorithm depends on the choice of kernel function. To address this issue, they introduced Lie group homomorphic mapping into this algorithm. Through this Lie group homomorphic mapping, the original Lie group samples are mapped to the target Lie group space. Thus, the degree of road association is minimized in different single connected spaces in the target Lie group space, and the degree of road correlation in the same single connected space is maximized to reduce road cross problems.

4.1.6 Lie group deep structure learning

Deep structure learning refers to a learning model that involves multiple levels of nonlinear operations, such as neural networks containing multiple hidden layers; this learning model consists of many nonlinear layers that are nonlinear and transformable functions[36, 37].

He and Li[38] pointed out that a complex problem can be broken down into a number of simple questions. Each simple problem corresponds to a layer, which constitutes a deep structure. In a certain problem, the deep structures of layers present specific relationships, which are not mutually independent but involve numerous nonlinear layers. Hong[39] proposed a face recognition method based on singular value vector features and proved that the singular value vector exhibits the properties of stability, translation, rotation, and mirror invariance. However, the singular value of a face contains only a small amount of effective information, thereby leading to a low face recognition rate[40]. To effectively detect superficial structures, such as protrusions and recesses in the face space, He and Li[38] proposed a Lie group layered learning algorithm. Firstly, the main features of the source image are extracted by the singular value decomposition method, and the deep structure based on the image eigenvalue is constructed. Secondly, the training and test samples are decomposed by singular value decomposition to obtain the left and right orthogonal matrices. Thirdly, each layer of this deep structure corresponds to the Lie group generator of the training and test samples. Learning from the top down, the first layer is learned for the measured and training samples of the geodesic distance d to retain the geodesic distance $d <$ min for the distance test samples. Finally, according to the $n-1$ layer, the test sample image is retained by finding the n-th layer of training and test samples between the geodesic distance d_n. The result of the last layer is the result of the required test, and the output of layer i is the input sample of layer $i + 1$.

To maximize the use of the data feature to guide the learning process, He and Li[38] put forward Lie group deep structure heuristic learning by involving a new selection criterion in Lie group layered learning. The difference degree of each layer is calculated, and the sample with the smallest difference is retained and used as the output. Stratification can be layered through the screening of layers to reduce the amount of data to be tested and thereby improve the accuracy of classification.

Yang et al.[41] proposed an algorithm for capturing the Lie group manifold structure of visual impression. They developed single-layer Lie group models and stacked them to yield a deep architecture; they also solved the learning problem of network weight by designing a Lie group-based gradient descent algorithm.

4.1.7 Lie group means learning

Mean calculation is one of the necessary steps in many machine learning algorithms[42]. Given the great application value of Lie groups, especially affine groups, special orthogonal groups, and positive definite symmetric groups, in solving problems, the mean calculation of Lie groups has become an important research topic.

In Ref. [43], Govindu defined the mean of Lie groups with the Lie algebras of special orthogonal and special Euclidean groups. The mean provides efficient and accurate algorithms for fusing motion information. In Refs. [44, 45], the linear representation of Lie group elements in a tangent space was added to the manifold to improve the per-formance of the mean shift algorithm for nonlinear spaces. These existing algorithms are used for sample clustering and for dealing with motion segmentation in the field of computer vision. In Ref. [46], Moakher presented precise definitions of the properly invariant notions of mean rotation. The mean rotation associated with the intrinsic metric in SO(3) is the Riemannian center of mass of the given rotation matrices; it shares many common features with the geometric mean of positive numbers and the geometric mean of positive Hermitian operators.

Gao and Li[47] proposed a simple classifier of Lie groups that is based on the inner mean value, i.e., the Internal Mean algorithm for the N-class Lie Group Sample (IMfNcLGS), which is described as Algorithm 6. As the inner mean is highly representative of the commonality of a class of things, if an unknown sample is closer to the inner mean of one category than to those of other categories, then this unknown sample is most likely to belong to that category. Therefore, after the calculation of the geodesic distance of each test sample to various inner mean values corresponding to its category, the category of the sample is determined as the category of the inner mean within the shortest distance; that is

$$i^* = \arg\min_{i=1,\ldots,C} \left\| \log(\mu_i^{-1} x) \right\|,$$

where C is the number of clusters, and μ_i is the internal mean of each category. To calculate the inner mean value, Gao et al.[42,48] proposed two methods. One method is based on iteration and is suitable for a Lie group with a complex geometric structure; this method is described as Algorithm 6. The other method is used to solve the inner mean value by directly using mathematical methods; it is written as

$$E_\mu = \exp\left(\frac{1}{n} \sum_{i=1}^{n} \log_2 x_i \right),$$

where x_i is the i-th data point, and n is the number of data points.

On the basis of the Lie mean algorithm, Gao et al.[42,48] proposed the first form of the Lie–Fisher algorithm and the second form of the Lie–Fisher algorithm. They deduced the Fisher projection direction of the samples in the nonlinear space of the Lie group on the basis of the Fisher projection theory and derived the calculation formula. For the first form of the Lie–Fisher algorithm, they mapped the sample points on the Lie group to the corresponding Lie algebra space, thus forming a new sample set. They then calculated the mean of each new sample, the overall mean of the new sample set, and the mean of the various classes. The standard Fisher method was used to solve the intraclass divergence and interclass divergence, and the projection direction that can generate the Lie–Fisher geodesic was obtained. The difference between the two forms of the Lie–Fisher algorithm is as follows. The process of Lie–Fisher discriminant analysis is to find a classification geodesic on the Lie group manifold. After all the samples are projected onto the geodesic line, the set of projection points of the sample have the maximum ratio between intraclass divergence and interclass divergence. From the derivation of the two forms of the Lie–Fisher algorithm and regardless of the distribution characteristics of the Lie group samples, the first form of the Lie–Fisher algorithm first maps all of them via logarithm mapping, which involves mapping

Algorithm 6 IMfNcLGS

Input: $\{x_{ij}\}_{i=1,\ldots,c}^{j=1,\ldots,n_i} \in G$, x_{ij} represents the j-th sample of the i-th category, and n_i represents the number of training samples in the i-th category.

Output: $\mu_i, i = 1, \ldots, c$, i.e., the internal mean of each category.

Procedure:

Do
  $i = 1$
  $k = 0$
  $\mu = x_{i1}$
  Do
    $\mathrm{Del}\mu = \tau / n_i \sum_{j=1}^{n_i} \log(\mu^{-1} x_{ij})$
    $\mu = \exp(\mathrm{Del}\mu)$
    $k = k + 1$
  While $\|\mathrm{Del}\mu\| > \epsilon$ and $k <$ Maxmum_Iterations
  $\mu_i = \mu$
  $i = i + 1$
While $i \leqslant c$

the Lie group samples to the Lie algebra linear space. Then, it uses the linearity of the Lie algebra space to calculate the mean and the direction of the projection. The main difference between the second and first forms of the Lie–Fisher algorithm is that the calculation of the mean is not carried out in the Lie algebra space but in the nonlinear space of the Lie group.

In Refs. [44, 45], the linear representation of Lie group elements in the tangent space was added to the manifold to improve the performance of the mean shift algorithm of a nonlinear space. The algorithms are used for sample clustering and for dealing with the motion segmentation in the field of computer vision.

4.1.8 Lie group kernel learning

The kernel trick is a useful computational theory that was originally used in SVM[49]. It is used to map nonlinear data from low dimensions into high dimensions in an intelligent manner[50]. With the kernel trick, the data become linearly separable. It is widely used in existing linear algorithms, such as the kernel-based principal component analysis algorithm[49], kernel-based independent component analysis algorithm[51], and Kernel Fisher Discriminant Analysis (KFDA) algorithm[52].

However, the traditional vector-based kernel function cannot be used for samples of a Lie group matrix in LML. To address this issue, Gao et al.[42,47] proposed the Kernel-based Fisher Linear Discriminant Analysis (KLieDA) algorithm, which is rooted in the Lie group kernel function. They designed a number of new kernel functions, such as the Lie group polynomial kernel, Radial Basis Function (RBF) kernel of the Lie group, Lie group linear kernel, and Lie group perceptron kernel for Lie group kernel machine learning algorithms. In KLieDA, nonlinearly separable samples are mapped to a high-dimensional space so that they become linearly separable. During the classification period, the Lie group mean accelerates the classification discrimination of unknown samples. Relative to the traditional Kernel Fisher Discriminant Analysis (KFDA) algorithm, KLieDA has two innovations in maintaining the same time and space complexity. Firstly, it uses the kernel functions of matrix Lie groups for handling matrix samples. Secondly, it calculates the mean of the original sample and the projection of the mean in the F space. When an unknown sample is mapped to the F space, KLieDA only requires the distance from each mean point; hence, the computational complexity is reduced.

A two-class KLieDA algorithm is described as Algorithm 7. In this algorithm, the log-Euclidean mean is the global mean. From a geometric viewpoint, the log-Euclidean mean calculates the geometrical average, which is calculated when all the group elements are mapped to the Lie algebraic linear space. The calculation of the log-Euclidean mean is performed once, and the time complexity is $O(n)$.

To efficiently deal with the complex nonlinear variations of face images, Xu et al.[53] proposed a Lie group kernel to address facial analysis problems. In this algorithm, a Linear Dynamic Model (LDM)-based face representation method captures the appearance and spatial information of a face image. Then, the derived LDM can be parameterized as a specially structured upper triangular matrix. The similarity between LDMs for any two face images can be characterized by a Lie group kernel.

4.1.9 Tensor learning

Tensor is a term in multiple linear algebras. In practical applications, many types of data are high-dimensional and involve multiple inputs while most traditional machine learning algorithms process vectorized data.

Algorithm 7 Two-class Lie group KLieDA classification algorithm

Input: $\{x_{ij}\}_{i=1,2}^{j=1,\ldots,n_i} \in G$, x_{ij} represents the j-th sample of the i-th category, and n_i represents the number of training samples in the i-th category.

Output: Category attribution for each unknown sample.

Procedure:

(1) For each Lie group sample x_{ij}, find the means $\mu_i (i = 1, 2)$ of the two classes of Lie group samples use Algorithm 6 for the general Lie group, and find the log-Euclidean mean of the covariance sample using $\mu_i = \exp\left(\frac{1}{n_i}\sum_{j=1}^{n_i} \log_2 x_{ij}\right), i = 1, 2.$

(2) Choose the specific Lie group kernel functions, and then obtain $M_i = 1/n_i \sum_{k=1}^{n_i} k(x_j, x_k^i)$ of two types Lie group samples. Then, set $M = (M_1 - M_2)(M_1 - M_2)^{\mathrm{T}}$.

(3) Calculate $N = \sum_{i=1,2} K_i(l - 1_{n_i})K_i^{\mathrm{T}}$, where K_i is an $n \times n_i$ matrix and $(K_i)_{nm} = K(X_n, X_m^i)$.

(4) Calculate $\alpha : \alpha = N^{-1}(M_1 - M_2)$

(5) Calculate the projection of the two classes of means μ_1 and μ_2 on the vector v in space F: $\tilde{\mu}_1 = (v \cdot \phi(\mu_1)) = \sum_{i=1}^{c}\sum_{j=1}^{n_i} \alpha_i k(x_{ij}, \mu_1)$, $\tilde{\mu}_2 = (v \cdot \phi(\mu_2)) = \sum_{i=1}^{c}\sum_{j=1}^{n_i} \alpha_i k(x_{ij}, \mu_2)$.

(6) For unknown x, project it on the vector v in space F: $\tilde{x} = (v \cdot \phi(x)) = \sum_{i=1}^{c}\sum_{j=1}^{n_i} \alpha_i k(x_{ij}, x)$. Then, the category attribution is obtained by $i^* = \arg\min_{i=1,2} |\tilde{x} - \tilde{\mu}_i|$

If these tensor data are forced to be vectorized, then the "curse of dimensionality" and the "small sample size problem" occur, and the intrinsic structure of the data is destroyed. Therefore, the novel machine learning methods based on multiple linear algebras (tensors) have attracted much interest. Tensor methods provide an effective approach to dealing with multiple input data[54–59]. Existing tensor methods have been applied to pattern recognition, web data mining, textual analysis, signal processing, and so on. Popular methods include the Generalized Low-Rank Approximation Matrix (GLRAM)[56], Higher Order Orthogonal Iteration (HOOI)[54], 2DPCA[55], CubeSVD[58], Tensor Subspace Analysis (TSA)[59], and neighborhood-embedded tensor learning.

Li[60] presented a data reduction algorithm that is based on tensor fields. This method deals with the reduction of data in tensor fields from the perspective of affine transformation of tensor fields based on the tensor decomposition technology Higher-Order Singular Value Decomposition (HOSVD). An iterative technique is adopted to solve the optimal problem and thereby obtain the optimal value. As tensor data are not confined to a single tensor and tensor order is not confined to a second or third order, the algorithm is more universal than the common tensor algorithms, such as GLRAM and HOOI. However, such property makes the algorithm increasingly complex. Li[60] also introduced tensor fields and tensor bundle theory into machine learning and discussed the basic concepts of tensor clusters and tensor fields on manifolds.

To deal with noisy and corrupted data, Lu[61] proposed a tensor low-rank subspace dictionary learning algorithm. They jointly used low-rank representation theory and sparse representation theory and considered the structures and features of data. The proposed model is robust and capable of handling abnormal samples and is capable of improving the power of representation and classification. Lu[61] further proposed a novel semisupervised Support Tensor Machine (STM). This proposed method can achieve the highly efficient optimization of STMs, and it can incorporate the transudative method to solve the problem of semisupervised classification. This method thus improves the power of classification and minimizes the time complexity.

Image retagging aims to improve the tag quality of social images by completing missing tags, rectifying noise-corrupted tags, and assigning new high-quality tags. Tang et al.[62] proposed the social anchor-unit graph regularized tensor completion method. They constructed an anchor-unit graph across multiple domains instead of using the traditional anchor graph in a single domain. The method can significantly accelerate large-scale graph-based learning by exploring only a small number of anchor points. Then, a tensor completion method based on social anchor-unit graph regularization is implemented to refine the tags of anchor images. Tags are finally efficiently assigned to non-anchor images by leveraging the relationship between the non-anchor units and the anchor units.

4.1.10 Frame bundle connection learning

Connection is a core concept that is widely used in modern geometry. By means of connection, the elements between two points in a geometric entity can be compared with each other.

The manifold learning method based on tangent bundles usually constructs a single frame for data processing[4]. The original algorithm might lose its efficacy for dealing with the multimanifold data. To address this issue, Li et al.[63] proposed a connection learning algorithm that is based on frame bundles, that is, the multimanifold data processing model. According to the structure of a manifold, the problem can be easily solved using a connection operator. The frame bundle is introduced into the connection learning framework by constructing coordinate frames for manifolds and combining them to form the coordinate coverage. To decrease the influence of noise on manifold learning, Li et al.[63] added a regularization item to a new method called the longitudinal space connection learning model based on frame bundles. The method separates the noise from the original sample data by regarding it as a new type of manifold data structure and handling it accordingly. By analyzing the longitudinal space connection learning model based on frame bundles, Li et al.[63] found that the performance of the algorithm is improved by sacrificing the computation time. To address this issue, they proposed a horizontal space connection learning model that is based on frame bundles by employing the horizontal space field of a tangent space to the algorithm. As the horizontal space field is the tangent subspace with m dimensions while the longitudinal space has m^2, the time complexity can be reduced by reducing the dimension of the original dataset.

Li et al.[63] proposed an image segmentation algorithm that is based on invariant image features by applying manifold affine connection theory. In this algorithm, an image is divided into small superpixel areas by combining local distance mapping and the watershed algorithm. This division is performed in a tensor space on the Riemannian manifold. The algorithm then finds the contact factor of the inside and outside regions between different tangent spaces by changing the size of the closed area determined by the level set function. The final target area is then obtained. The specific application to medical images shows that the proposed algorithm is promising.

4.1.11 Finsler geometric learning

The geometry with Finsler metrics is called Finsler geometry. Finsler metrics are simply Riemannian metrics without quadratic restrictions; it was first introduced by Riemann in 1854[64].

Chen et al.[65] proposed the Finsler Metric-based KNN (FMKNN) algorithm. As the KNN algorithm is an inert learning method, it has many shortcomings, such as low classification speed, influence of the weight of attributes and other factors on accuracy, and strong dependence on the capacity of the sample library. To address this issue, Chen et al.[65] adopted a class of computable Finsler metrics, i.e., (α, β) metrics in the KNN algorithm, which play a crucial role in Finsler geometry. Experimental results show that the classification performance of the algorithm can be improved after introducing the Finsler metric. FMKNN has a number of shortcomings. For example, the projection vector of LDA is used when choosing the Finsler metric function. Hence, the FMKNN algorithm can only achieve satisfactory results when used on datasets with a good LDA effect in the calculation of the Finsler metric. To address this issue, Chen et al.[65] introduced the Finsler metric and proposed a geometric learning algorithm that is based on Finsler metrics by using the class label of samples. The algorithm is a supervised learning algorithm that can handle multimanifold structured data by reducing the difference within classes and increasing the difference between classes by using the label information of samples. Specifically, a dataset is divided according to the classification information of the training data. Assuming that each partition is a single manifold structure, the existing classical manifold learning algorithms can be used to reduce dimensionality. Then, the Finsler distance between any two centers is calculated to form the distance matrix, and the low-dimensional representation of the center point is obtained by applying the Multidimensional Scaling (MDS) algorithm to the distance matrix. Finally, the low-dimensional representation of each partition set and the low-dimensional representation of the center points are rotated and translated, and the geometric structure relation between the partitioned datasets is maintained. In this way, the low-dimensional representation of the whole dataset is obtained.

To solve the problems with the k-means algorithm, including the optimization effect of similarity measures and criterion functions being insufficient and the analysis performance of multidimensional manifold data being ineffective, Xu et al.[64] proposed a k-means algorithm that is based on Finsler geometry by introducing a Finsler metric. Experimental results show the feasibility and effectiveness of the proposed algorithm.

4.1.12 Homology boundary learning

Boundary division can be split into data partition and image division[66]. The main boundary learning algorithms are the tangent vector quantization algorithm[67] and Canny algorithm[68].

Xian and Li[66] analyzed homology theory and pointed out that it is a method for edge division. On the basis of homology theory, they proposed the homology edge division algorithm. According to the relation of the homotopy equivalence of topological spaces, these spaces can be classified with homotopy (same topological space with the same homotopy type, different types of topological spaces with different homotopy types, etc.). Homotopy mapping can be divided into equivalence classes, and each equivalence class is called a homotopy class. In classification, the classification object can be described by judging whether the space is homeomorphic. As the homeomorphism space is homotopy equivalent, the homotopy equivalence classification of topological spaces is a generalization of topological classification. In their classification algorithm, the classification object is described by judging whether the space is homeomorphic. The algorithm then judges whether the two connected edges are homeomorphic according to the properties of the homology groups. If homology groups of the same dimensions in a subdivision space are homeomorphic,

then they are isomorphic. The homology group is subsequently computed to determine whether the two subdivision spaces can be homeomorphic. Finally, whether the relationship between the two-dimensional chains is equivalent is determined according to whether the homological relation between q-dimensional chains is an equivalent relation.

At present, a number of existing margin learning algorithms cannot effectively retain the feature structure invariability of data when solving the edge partition problem. To address this issue, Zhao and Li[69] presented the neighborhood homology learning algorithm that is based on monomorphic division theory in homology algebra. They presented a new method for constructing the neighborhood complex of a graph and designed a criterion for judging the similarity between two given graphs.

4.2 Unsupervised Lie group machine learning

Fiber bundle theory is an important part of the study of geometry[4, 70]. It describes the relationship between the global and local properties of a differential geometry object. A fiber bundle is a generalized product of a manifold. In sum, let E and M be two smooth manifolds. The mapping of E to M is smooth, the affine space at each point on M is composed of an n-dimensional vector set, and (E, M) is the vector bundle on manifold M. Intuitively, vector bundle E is the result of the product manifold and fiber bonding. Adhesion requires the linear relationship on the fiber to remain unchanged. Obtaining the intrinsic information or knowledge hidden in data, such data should be subject to in-depth analysis. The data objects that are actually processed can be grouped or categorized according to their similarities. After such processing, the data objects comprise the structure of the class or cluster. This cluster structure is consistent with the fiber bundle structure. Different types of high-dimensional data in clustering analysis may correspond to different subspaces. In fiber bundle language, every subspace generated by higher-dimensional clustering is a fiber, and the set of all subspaces is a bundle space.

Fiber bundle as a mathematical tool has been successfully applied to computer vision, pattern recognition, neural networks, and cybernetics. In computer vision, Sochen et al.[71] used fiber bundles to establish a framework for nonlinear diffusion. In neural networks, Pearson[72] developed a fiber bundle for feedforward neural network representation. In pattern recognition, Chao and Kim[73] established a fiber bundle model of surface shapes and realized the rapid generation of surface shapes.

Most existing manifold learning algorithms are based on small local neighborhoods, and obtaining their global coordinates is sometimes unrealistic. As many low-dimensional embedded manifold dimensions sampled from high-dimensional data are determined by several hidden variables, the dimensions of a dataset can be reduced if the curl manifold in the observing space can be effectively expanded or the main internal hidden structures can be found. Therefore, Zhou and Li[74] proposed a Lie group fiber bundle learning model. The local properties of the manifold are approximated by the tangent space of each point on the manifold and each tangent space composed of the tangent bundle. Hence, the fiber bundle can be used as a tool for the further analytical processing of the manifold structure and its tangent bundle. The vector field dimensionality reduction algorithm based on Tangent Local Principal Direction (VTLPD) (i.e., tangent bundle field reduction) is described as Algorithm 8.

Gao[75] proposed the diffusion geometry of fiber bundles, that is, Horizontal Diffusion Map (HDM) algorithm. This algorithm is a graph-based framework for analyzing complex datasets with nonscalar or functional pairwise relations, particularly datasets in which the similarity scores between samples can be obtained from correspondence relations between sophisticated individual structures carried within each sample. In this algorithm, a data object is viewed as being approximately sampled from a smooth base manifold, and data points are viewed as samples on the fibers of a fiber bundle over the base manifold.

Algorithm 8 VTLPD

Input: sample set X, dimension d of the low-dimensional embedded space;

Step 1 Find the t centers of X with the k-mean; divide X into t blocks, each expressed as $X_i = [x_{i1}, \ldots, x_{ik}]$;

Step 2 Local principal component analysis. Calculate the covariance matrix $(X_i - \overline{x_i}e^{\mathrm{T}})^{\mathrm{T}}(X_i - \overline{x_i}e^{\mathrm{T}})$ to obtain the first principal eigenvector θ_i. The first principal component is $z_i = (\theta_i^{\mathrm{T}} \cdot X)^{\mathrm{T}}$;

Step 3 Find the coordinates $z_i = z_i - \mathrm{ctr}_i$ of the first principal component vector z_1 on each block, where ctr_i is the center of each cluster, and the matrix $Z = [z_1, z_2, \ldots, z_t]$

Step 4 Call manifold learning algorithm dimension reduction vector field Z.

The data points on the same data object are assumed to come from the same fiber. A dataset with pairwise structural correspondences is modeled as a fiber bundle equipped with a connection. Computing the pairwise similarity between data objects typically requires optimizing a certain function over the space of admissible pairwise structural correspondences. The optimal correspondence is then used to assign a distance or similarity score between the two data objects under comparison. The HDM framework aims to mine this hidden information from pairwise structural correspondences. The HDM algorithm provides a two-level data representation. As the second-level embedding for data objects leverages the rich structural information at the level of data points, it is expected to be semantically more meaningful than the spectral representation obtained from standard diffusion maps, which cannot take advantage of individual structural information.

4.3 Category representation learning

In 1991, Asperti and Longo[76] proposed the relation between category theory and computer science. Morphism, as a symbol of category theory, generalizes a function generally and provides a unified explanation for all aspects of programming theory. This categorization of mathematical forms is suitable for many aspects of computer science. In 2018, Muhiuddin[77] pointed out that category theory now occupies a central position not only in contemporary mathematics but also in theoretical computer science and even in mathematical physics. It can roughly be described as a general mathematical theory of structures and systems of structures. It is, at the very least, a very powerful language or conceptual framework that allows us to see, among other things, how structures of different kinds are related to one another as well as the universal components of a family of structures of a given kind. A category is an abstract structure: a collection of objects, together with a collection of the morphisms between them. In 2010, Zhou and Li[74] proved that machine learning systems can be represented by categories and that various learning algorithms can be represented in terms of category theory. For example, decision tree learning and the Bayes learning algorithm can be defined as machine learning categories. They also adopted functor category theory to study the expression and a mapping mechanism to study the dimensionality reduction principle. In the data reduction category, they obtained the relation between image data objects and the objects in the low-dimensional space. Converting high-dimensional data objects into relatively tractable low-dimensional data objects can reduce the complexity of morphism between various data objects.

A linear dimensionality reduction algorithm for the dimensionality reduction of data is shown in Fig. 8.

In Fig. 8, primitive high-dimensional data object X is the input value and low-dimensional coordinate data object Y is the output value.

(1) Construct the associated objects M. Morphism g is the process of constructing associated objects, $g: X \to M$.

(2) Compute the eigenvalue Z and eigenvector W of the associated objects $M, h: M \to Z, \varphi: Z \to W$, and obtain the compound operation law for the data reduction category φh: $M \to W$.

(3) By analyzing the characteristics and relations of known objects in the data-reduced category C, the data object Y of low-dimensional coordinates is obtained. A custom operator is needed to determine the intermediate transition object T, satisfying ϕ: $T \to Y$.

Xu et al.[78] proposed several basic concepts of the category representation of machine learning methods on the basis of category theory. By analyzing the decision tree, SVM, principal component analysis, and deep neural network methods with category representation, they presented the corresponding category representation for each algorithm. They also put forward the corresponding theoretical proof and feasibility analysis and confirmed the feasibility of adopting the category representation method by using simulation experiments.

4.4 Semisupervised Lie group machine learning

The use of a large number of unlabeled examples not only leads to the construction of a reliable classifier but also reduces the manpower and material

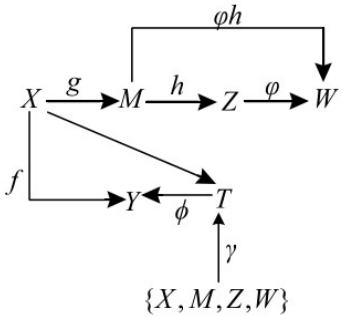

Fig. 8 Linear dimensionality reduction algorithm for the dimensionality reduction of data.

resources required, thereby improving the performance of semisupervised machine learning; hence, this approach has attracted considerable research interest in theory and practice and is one of the most important topics in current machine learning research[38, 61, 79–81].

4.4.1 Lie group deep structure semisupervised learning algorithm

He and Li[38, 81] proposed the Lie group deep structure semisupervised learning algorithm. In this algorithm, the deep structure of a process is constructed to break down a complex problem into a number of simple questions and to establish an effective link between each simple question. Thus, with the deep structure analysis of complex data, each layer corresponds to a complex problem decomposition of a simple problem, and each layer has a valid index. In the output layer, the deep structures of the hidden layer and auxiliary layer are embedded in the supervised learning algorithm that is based on a semisupervised Lie group. The experimental results reveal the effect of this algorithm.

4.4.2 Semisupervised learning algorithm based on Lie group

Xu[80] put forward a series of semisupervised algorithms that are based on LML. Firstly, the author proposed a semisupervised learning algorithm that is based on a linear Lie group. This semisupervised learning model is based on a Lie group's algebraic and geometric structures. It can find the special structure of a linear Lie group, target the special structure of the linear Lie group, and identify its corresponding infinitesimal left and right operator generation elements. The latter step is equivalent to finding the base vector of the Lie algebra corresponding to the Lie group.

Given a small amount of marked data, the model can judge whether the other elements of the learning system can be represented by the operator generator of the linear Lie group according to the closure of the group algorithm and the law of the joint. The algorithm mainly considers the Lie group relationship $[X, Y]$ between learning object X and learning object Y. It also makes full use of the algebraic group structure and the geometric manifold structure of the Lie group in considering the unlabeled example from the perspective of the operator generating element and the translational dimension of the manifold. Its accuracy is higher than that of a single algorithm with algebraic or geometric methods.

Further, Xu[80] proposed a semisupervised learning algorithm based on parameter Lie groups. In this algorithm, the sample data are mapped to the parameter Lie group according to the representation of parameters. This algorithm performs well due to the advantage of parameter Lie group learning and semisupervised learning.

The Semisupervised Learning Algorithm based on Parameter Lie Groups (SSLA-PLG) is described as Algorithm 9.

Algorithm 9 SSLA-PLG algorithm flow

Algorithm function: Use Lie group method for semisupervised learning

Input: Sample set $X = (x_1, x_2, \ldots, x_l, x_{l+1}, \ldots, x_{l+k} \in \mathbb{R}^D)$,
where $x_1, x_2, \ldots, x_l$ is the labelled data, i.e., there exits the set $L = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$, where $x_{l+1}, x_{l+2}, \ldots, x_{l+k}$ are unlabeled data ($l \ll k$).

Output: Label y of data x

Step 1 According to the input sample set $X = (x_1, x_2, \ldots, x_l, x_{l+1}, \ldots, x_{l+k} \in \mathbb{R}^D)$, represent it with a parameter separately. Determine the corresponding learning system isomorphism (state) parameter Lie group and make full use of the marked data, representing the object to be learned with the parameters of Lie group. The object is required to meet the structure of the parameter Lie group.

Step 2 According to the algebraic structure of the parameter Lie group, the group method is used for marking. Based on the marked data as the base point, the definition of the element generated by the parameter Lie group is

$$X_\mu = \lim_{\alpha^\mu \to 0} \frac{g(0, \ldots, \alpha^\mu, \ldots, 0) - g(0, \ldots, 0)}{i\alpha^\mu}$$

which calculates the generated meta. Determine other unlabeled data according to the group operation rule, and if the unlabeled data can be represented by the matrix generator, the category is marked.

Step 3 According to the geometric structure of the parameter Lie group, the analytic manifold is used for marking. Generate the field $U_i (1 < i < l)$ of all labelled sample points x_i. Analyze the neighborhood information $U_i (1 < i < l)$ of each labelled sample point x_i. Select each constant inner product $g_{i,j}(x_i)$ under the function of self-isomorphism on the unit point in the sample set. Calculate the value of $g_{i,j}(x_i)$. Put the value of $g_{i,j}(x_i)$ into the following formula:

$$d(x_i, x_j) = \sqrt{\sum_{i,j=1}^{n} g_{i,j}(x_i)(x_{ii} - x_{ji})(x_{ij} - x_{jj})}.$$

Calculate the distance between x_i and x_j in U_i $(1 < i < l)$. Determine the smallest value d of each point x_i corresponding to x_j, and mark x_j as the marker that is the same as x_i.

Step 4 Check whether the data are marked; if not, return to Step 2 and repeat.

4.4.3 Spectral estimation learning

As a classical mathematical analysis and algebraic method, the spectral analysis method already has a complete theoretical foundation. It has a wide range of applications in machine learning. A number of spectral estimation clustering methods based on similarity matrix eigenvalue decomposition have also been proposed. The spectral method defines a relation matrix describing the similarity between pairs of data points and calculates their eigenvalues and eigenvectors. Then, suitable feature vectors are selected to project the low-dimensional embedding of data. The key of the spectral estimation method is to choose the proper kernel function as the criterion for choosing the similarity matrix.

The learning algorithm of spectral manifolds is aimed at discovering a low-dimensional representation in a high-dimensional vector space. Hence, it has attracted great attention in recent years. Dong[82] and Yang et al.[83] studied the spectrum learning algorithm and proposed a series of algorithms on spectral estimation learning. Firstly, they proposed a Manifold Dimensionality Reduction algorithm with Geodesic distance as the proximity measurement (MDRG). On the basis of the spectrum technique, they extended the MDRG method and proposed a synchronous spectrum estimation learning algorithm. The learning problem of the image feature manifold can be given as a mathematical model by defining the criterion function, i.e., the optimal division among the subclasses of sample points is found by solving the optimal solution of the criterion function. To take advantage of local information, they proposed a spectral estimation learning algorithm for the topological invariance of image feature manifolds. In this method, local curves are used to preserve the nonlinear structure characteristics of manifolds. The local tangent space is selected to reconstruct the sample points, the local reconstruction error is minimized, and the reconstruction error does not increase in the global adjustment process. Thus, only the eigenvectors corresponding to the maximal two eigenvalues of the Laplacian operator in the manifold are used as features according to the spectral characteristics of the manifold. Moreover, pattern recognition and data dimension reduction can be achieved.

Huang and Li[84] proposed the Isospectral Manifold Learning Algorithm (IMLA). Isospectral manifold learning is one of the main contents of spectrum methods. Isospectral manifold learning stems from the conclusion that if the spectra of manifolds are the same, so are their internal structures. However, the difficult task in the calculation of spectra is the selection of the optimal neighborhood size and construction of reasonable neighboring weights. To address this issue, they proposed the isospectral manifold learning algorithm. By modifying directly the sparse reconstruction weight, the IMLA considers the within-neighboring information and between-neighboring information. This method preserves the sparse reconstructive relationship and sufficiently utilizes the discriminant information.

Spectral manifold learning algorithms have undergone considerable development. However, their computational complexity remains high. To address this issue, Huang and Li[85] proposed a Fast learning algorithm of Spectral Manifold (FSM). FSM applies two technologies to reduce the high computational complexity. Firstly, it selects p anchor points from n data points through random selection or k-means selection and represents the data points as the linear combinations of these anchor points. Secondly, linear manifold learning can be used to compute the low-dimensional parameterizations of high-dimensional data effectively and thereby reduce the computational complexity of the optimal eigenvalue.

Ren et al.[86] investigated the possibility of applying spectral methods to recover the parameters of supervised latent Dirichlet allocation. Supervised topic models simultaneously model the latent topic structure of large collections of documents and the response variable associated with each document. Existing inference methods are based on variational approximation or Monte Carlo sampling, which often suffers from the local minimum defect. To address this issue, they presented a two-stage spectral method, which recovers the parameters of LDA followed by a power update to recover the regression model parameters. They further presented a single-phase spectral algorithm to jointly recover the topic distribution matrix and regression weights. Experimental results demonstrate the practical effectiveness of the spectral algorithms.

4.4.4 Semisupervised tensor learning

In the fields of machine learning, pattern recognition, image processing, and computer vision, data are usually

represented by tensors. The STM is an improved model based on SVM[49,87]. It inherits the advantages of the solid theory of SVM in wide application fields while maintaining the original spatial structure of data. The STM can effectively alleviate the overfitting problem caused by small-scale datasets and avoid the curse of dimensionality caused by high-dimensional data. However, the traditional STM is a supervised learning model that cannot deal with unlabeled samples. To address this issue, a large number of researchers have made great improvements.

Fei et al.[88] proposed a Transductive Support Tensor Machine (TSTM) algorithm to train effective classifiers by using a large amount of unlabeled data and labeled data. In the TSTM, the image frame, audio, and text in video shots as data points are represented by a third-order tensor. The algorithm also considers the manifold structure of the tensor space from the contextual temporal associated cooccurring multimodal media data. The TSTM inherently preserves the intrinsic structure of the submanifold where tensor shots are sampled. It can also map out-of-sample data points directly. However, the TSTM needs to resort to iterative techniques, which are time consuming. Hence, Liu et al.[89] proposed a low-rank approximation-based TSTM, in which the tensor rank-one decomposition is used to compute the inner product of the tensors. The Concave-Convex Procedure-based (CCCP)-TSTM provides significant performance gains in terms of test accuracy and training speed.

Hu et al.[90] proposed a semisupervised tensor-based graph embedding learning algorithm. In this algorithm, two graphs are designed to characterize the intrinsic local geometrical structure of the tensor samples of the object and the background, and two propositions are proved for finding the transformation matrices, which are used to map the original tensor samples to the tensor-based graph embedding space. Through a transfer learning-based semisupervised strategy for iteratively adjusting the embedding space, extensive discriminant information in the embedding space is encoded. They applied this algorithm to visual tracking. Experimental results demonstrate the effectiveness of the proposed tracking algorithm.

5 Application of Lie Group Machine Learning to Image Processing

The human actions of understanding and analyzing have received much research attention for multiple areas of applications, including human–robot interaction, surveillance, daily living, and video-based monitoring[91]. Lie groups have played an important role due to their properties. Vemulapalli et al.[92] proposed human action recognition by representing 3D human skeletons as points in a Lie group. In this algorithm, a skeletal representation is created in the Lie group $\mathrm{SE}(3) \times \mathrm{SE}(3) \times \cdots \times \mathrm{SE}(3)$, which is a curved manifold, on the basis of the observation that 3D rigid body motions are members of the space. In this way, human actions can be modeled as curves in a Lie group, and these action curves are mapped as a vector space. Public software packages for Lie group manifolds[92] are widely available, and they are capable of implementing 3D skeletal representations of people. Liu et al.[93] proposed a 3D-based deep convolutional neural network for action recognition with a depth sequence algorithm. In this method, a 3D-based deep convolutional neural network is constructed, and a joint-based feature vector (JointVector) is computed for each sequence by fusing the SVM classification results and the JointVector results. This method can learn feature representation, which is time invariant and viewpoint invariant, from depth sequences. Cai et al.[94] presented attribute mining for scalable 3D human action recognition. The scalable skeletal human action recognition from 3D videos can identify novel actions without rebuilding in most applications. A potential solution is to identify the intrinsic attributes, which are semantic-aware and shared among known and novel actions. Zhang et al.[95] presented a combination of depth-skeleton feature with sparse coding for action recognition. RGB-D human action recognition is an active research topic in computer vision and robotics. They also proposed an action recognition method that combines gradient information and sparse coding. Núñez et al.[96] presented convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Demisse et al.[97] proposed a deformation-based representation for analyzing expressions from 3D faces. In their approach, a point cloud of a 3D face is decomposed into an ordered deformable set of curves that start from a fixed point. Subsequently, a mapping function is defined to identify the set of curves with an element of a high-dimensional matrix Lie group, specifically the direct product of SE(3).

Scientists have made great efforts in developing advanced Lie Group Classification (LGC) approaches

for improving image classification accuracy. Von Tycowicz et al.[98] proposed an efficient Riemannian statistical shape model using differential coordinates and applied it to the classification of data from the Osteoarthritis Initiative. In this method, a differential representation is introduced to put the local geometric variability into focus. The differential coordinates are modeled as elements of a Lie group, thereby endowing the shape space with a non-Euclidean structure. A key advantage of this method is that statistics in a manifold shape space become numerically tractable, thus improving performance by several orders of magnitude over other state-of-the-art methods. Von Tycowicz et al.[98] presented a new approach toward the use of various types of geometry as artificial intelligence tools in robot control: the idea of minimum operation transformations. This approach is a new branch of soft computing for the adaptive control of a special class of nonlinear coupled multivariable systems. Its uniform structures are obtained from certain abstract geometry-related Lie groups. The advantages are as follows: a priori known and reduced structure size; increased lucidity; and simple, short, and explicit algebraic procedure instead of intricate learning. Hayat et al.[99] presented an RGB-D-based image set classification for robust face recognition from Kinect data. In this method, the raw Kinect data are used for pose estimation and automatic cropping of the face region. On the basis of the estimated poses, the face images of a set are divided into multiple image subsets. An efficient block-based covariance matrix representation is applied to model images in an image subset on a Lie group. Then, SVM models are separately learned for each image subset on a Lie group, and a fusion strategy for classification is introduced to combine the results from all image subsets. It incurs low computational cost and achieves a high identification rate. Yin et al.[100] presented a locally adaptive sparse representation on Riemannian manifolds to achieve a robust classification algorithm. In this method, the log-Euclidean kernel is used to embed Symmetric Positive Definite (SPD) matrices into a reproducing kernel Hilbert space, where the meaningful linear reconstruction of SPD matrices can be implemented. The SPD matrix manifold, which is a Riemannian manifold, belongs to a Lie group. By exploiting the geodesic distance between SPD matrices, the proposed method can effectively characterize the intrinsic local Riemannian geometry within data so as to effectively uncover the underlying submanifold structure. As SPD matrix manifolds, which are Riemannian manifolds, belong to Lie groups, they cripple many methods that rely on linear reconstruction. Wang et al.[101] provided a video feature descriptor that combines motion and appearance cues with length-invariant characteristics. In this method, one video feature descriptor that combines motion and appearance cues is designed. This feature descriptor is of length-invariant characteristics and is adopted to represent a video sequence for an abnormal event detection problem. Experiments on benchmark datasets validate the advantages of this proposed feature descriptor.

Image registration is an important problem in image processing, and it has attracted much attention. Dong et al.[102] presented LieTrICP: an improvement of the trimmed iterative closest point algorithm. This algorithm combines the trimmed iterative closest point algorithm and Lie group representation for registration between two-point sets. Given two low overlapped point sets, the method uses Lie group representation to estimate the geometric transformation from the selected point pairs. The experimental results demonstrate that LieTrICP is more accurate and robust than several other algorithms in a variety of situations, including missing points, perturbations, and outliers. Ying et al.[103] presented a nonlinear 2D shape registration method via thin-plate spline and Lie group representation algorithm. This method comprises two steps. In the affine registration step, the Lie group parameterization method is applied to globally align two shapes to assume a global similarity. In the locally nonlinear deformation step, the thin-plate spline approach is used. By alternatively iterating these two steps, the proposed method not only preserves the advantages of spline methods but also overcomes the overmatching phenomenon in shape registration.

6 Neuromorphic Synergy Learning

Neuromorphic synergy learning is an interdisciplinary frontier of cognitive science, artificial intelligence, brain-like intelligence, machine learning, robotics, pattern recognition, and image processing. Moreover, it is a key technology for big data, the internet of things, and cloud computing[4].

Related research on human-inspired planning has been determined as an important national strategic

development direction in many countries, such as the USA, Japan, the European Union, and China. Examples of related activities include the USA's brain project called the brain activity map project or the Brain Activity Map project (BAM), the European Union's Human Brain Project or Human Brain Project (HBP) for the European Commission's future and emerging technologies, China's brain project, and so on.

Currently, brain-like intelligence is the representative of 21st century frontier technology, and it plays an important role in human brain science plans launched by many countries around the world, especially in the fields of big data, the internet of things, and cloud computing. Thus, brain-like intelligence has become one of the most challenging technologies. Academic journals, such as *Science* and *Trends in Neuroscience*, have reported on human brain projects and neural informatics, and they have claimed that the human brain project, which includes an extensive context, is bigger than the Human Genome Project. As the core of brain-like intelligence, the neuromorphic synergy learning algorithm is expected to play a decisive role in the field of brain computing.

Under this background, Li et al.[4] focused on the three core scientific problems in neuromorphic synergy learning, i.e., the symbol grounding relation, bidirectional mechanism, and affordance learning, and studied neuromorphic synergy learning from four levels, namely, proposing a new theory; exploring a new method; initiating a new technique; and developing a new platform that includes the Lie group cognitive theory framework, neuromorphic synergy learning framework, symbol grounding learning, bidirectional synergy learning, and affordance learning.

7 Conclusion and Future Work

LML has attracted increasing attention in recent years. This work introduces LML models and the three major categories of LML: supervised LML, semisupervised machine learning, and unsupervised machine learning. In addition, neuromorphic synergy learning is introduced. This work covers the following techniques: LML model, Lie group subspace orbit generation learning, symplectic group learning, quantum group learning, Lie group fiber bundle learning, Lie group cover learning, Lie group deep structure learning, Lie group semisupervised learning, Lie group kernel learning, tensor learning, frame bundle connection learning, spectral estimation learning, Finsler geometric learning, homology boundary learning, category representation learning, and neuromorphic synergy learning.

LML plays an important role in the field of machine learning as it covers extensive research contents. In addition to the basic contents described above, LML covers statistical learning, deep learning, and meta learning, all of which deserve further investigation. We hope that this survey not only provides an enhanced understanding of LML for researchers but also facilitates future research activities and application developments in this field.

Acknowledgment

This work was supported by the National Key Research and Development Program (Nos. 2018YFA0701700 and 2018YFA0701701) and Scientific Research Foundation for Advanced Talents (No. jit-b-202045). The authors thank all of the reviewers for their valuable comments.

References

[1] F. Z. Li, S. P. He, and X. P. Qian, Survey on lie group machine learning, (in Chinese), *Chin. J. Comput.*, vol. 33, no. 7, pp. 1115–1126, 2010

[2] F. Z. Li, X. P. Qian, L. Xie, and S. P. He, *Machine Learning Theory and Its Applications*, (in Chinese). Hefei, China: China University of Science and Technology Press, 2009.

[3] F. Z. Li, L. Zhang, J. W. Yang, X. P. Qian, B. J. Wang, and S. P. He, *Lie Group Machine Learning*, (in Chinese). Hefei, China: China University of Science and Technology Press, 2013.

[4] F. Z. Li, L. Zhang, and Z. Zhang, Lie Group Machine Learning, https://doi.org/10.1515/9783110499506, 2019.

[5] T. A. Haill, An application of the lie group theory of continuous point transformations to the Vlasov-Maxwell equations (Plasma Physics), PhD dissertation, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1985.

[6] A. Baker, *Matrix Groups*: *An Introduction to Lie Group Theory*. Berlin, Germany: Springer, 2002.

[7] W. Fulton and J. Harris, Representation theory: A first course, https://doi.org/10.2307/3618117, 1991.

[8] M. Hunacek, Lie groups: An introduction through linear groups, https://doi.org/10.1017/S0025557200183561, 2008.

[9] P. Yarlagadda, O. Ozcanli, and J. Mundy, Lie group distance based generic 3-d vehicle classification, in *Proc. 2008 19th Int. Conf. on Pattern Recognition*, Tampa, FL, USA, 2008.

[10] Y. M. Lui, Advances in matrix manifolds for computer vision, *Image Vision Comput.*, vol. 30, nos. 6&7, pp. 380–388, 2012.

[11] H. Xu and F. Z. Li, The design of Su(n) classifier of lie group machine learning (LML), *J. Comput. Informat. Syst.*, vol. 1, no. 4, pp. 835–841, 2005.

[12] H. Xu and F. Z. Li, Algorithms of dynkin diagrams in lie group machine learning, *J. Commun. Comput.*, vol. 4, no. 3, pp. 13–17, 2007.

[13] H. Xu and F. Z. Li, Geometry algorisms of dynkin diagrams in lie group machine learning, *J. Nanchang Inst. Technol.*, vol. 25, no. 2, pp. 74–78, 2006.

[14] H. Xu and F. Z. Li, Lie group machine learning's axiom hypothesizes, in *Proc. 2006 IEEE Int. Conf. on Granular Computing*, Atlanta, GA, USA, 2006, pp. 401–404.

[15] F. Chen, Research and application on orbits generated algorithm of learning subspace in Lie-Group Machine Learning (LML), (in Chinese), master dissertation, Soochow University, Suzhou, China, 2007.

[16] F. Chen and F. Z. Li, Orbits generated theory of learning subspace and its algorithm in Lie-Group machine learning (LML), *J. Suzhou Univ.* (*Nat. Sci. Ed.*), vol. 23, no. 1, pp. 61–66, 2007.

[17] C. L. Lv, Z. K. Wu, D. Zhang, X. C. Wang, and M. Q. Zhou, 3D nose shape net for human gender and ethnicity classification, *Pattern Recognit. Lett.*, vol. 126, pp. 51–57, 2019.

[18] E. Boutellaa, O. Kerdjidj, and K. Ghanem, Covariance matrix based fall detection from multiple wearable sensors, *J. Biomed. Informat.*, vol. 94, p. 103 189, 2019.

[19] Y. Heider, K. Wang, and W. C. Sun, SO(3)-invariance of informed-graph-based deep neural network for anisotropic elastoplastic materials, *Comput. Methods Appl. Mech. Eng.*, vol. 363, p. 112 875, 2020.

[20] G. Lebanon, Metric learning for text documents, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 497–508, 2006.

[21] F. Chen and F. Z. Li, Orbits generated lattice algorithm of learning subspace in Lie-group Machine Learning (LML), (in Chinese), *Comput. Eng. Appl.*, vol. 43, no. 15, pp. 184–187, 2007.

[22] N. Nasios and A. G. Bors, Kernel-based classification using quantum mechanics, *Pattern Recognit.*, vol. 40, no. 3, pp. 875–889, 2007.

[23] S. P. He and F. Z. Li, A molecular docking drug design algorithm based on quantum group, (in Chinese), *J. Nanjing Univ.* (*Nat. Sci.*), vol. 44, no. 5, pp. 512–519, 2008.

[24] Y. M. Wu, J. Y. Hu, and X. G. Yin, Symplectic integrators of the equations of multibody system dynamics on manifolds, (in Chinese), *Adv. Mech.*, vol. 32, no. 2, pp. 189–195, 2002.

[25] K. Feng and M. Z. Qin, *Symplectic Geometric Algorithms for Hamiltonian Systems*. Berlin, Germany: Springer, 2010.

[26] Z. X. Xu, D. Y. Zhou, and Z. C, Deng, Numerical method based on hamilton system and symplectic algorithm to differential games, *Appl. Math. Mech.*, vol. 27, no. 3, pp. 341–346, 2006.

[27] H. X. Fu and F. Z. Li, Research of the symplectic group classifier based on lie group machine learning, in *Proc. 4^{th} Int. Conf. on Fuzzy Systems and Knowledge Discovery* (*FSKD 2007*), Haikou, China, 2007, pp. 649–655.

[28] H. X. Fu, Research on symplectic group classifier in Lie group machine learning, (in Chinese), master dissertation, Soochow University, Suzhou, China, 2008.

[29] W. W. Guan and F. Z. Li, Drug molecular design using lie group machine learning (LML), in *Proc. Int. Conf. on Advanced Intelligence*, Beijing, China, 2008, pp. 411–414.

[30] O. Nechaeva, The neural network approach to automatic construction of adaptive meshes on multiply-connected domains, in *Proc. Int. Joint Conf. on Neural Network*, Orlando, FL, USA, 2007, pp. 1912–1917.

[31] W. W. Guan, Research of covering algorithm in Lie Group machine learning, (in Chinese), master dissertation, Soochow University, Suzhou, China, 2009.

[32] L. M. Lui, W. Zeng, S. T. Yau, and X. F. Gu, Shape analysis of planar objects with arbitrary topologies using conformal geometry, in *Proc. 11^{th} European Conf. on Computer Vision*, Crete, Greece, 2010, pp. 672–686.

[33] Y. Chen, F. Z. Li, and P. Zou, Multiply connected lie group covering learning algorithm for image classifi-cation, (in Chinese), *J. Front. Comput. Sci. Technol.*, vol. 8, no. 9, pp. 1101–1112, 2014.

[34] C. Yan and F. Z. Li, Path optimization algorithms for covering learning, (in Chinese), *J. Software*, vol. 26, no. 11, pp. 2781–2794, 2015.

[35] L. H. Wu and F. Z. Li, Multiply Lie Group kernel covering learning algorithm for image classification, (in Chinese), *J. Front. Comput. Sci. Technol.*, vol. 10, no. 12, pp. 1737–1743, 2016.

[36] Y. S. Bengio, Learning deep architectures for AI, *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[37] G. E. Hinton, S. Osindero, and Y. W. Teh, A fast learning algorithm for deep belief nets, *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[38] W. H. He and F. Z. Li, Research on Lie Group deep structure learning algorithm, (in Chinese), *J. Front. Comput. Sci. Technol.*, vol. 4, no. 7, pp. 646–653, 2010.

[39] Z. Q. Hong, Algebraic feature extraction of image for recognition, *Pattern Recognit.*, vol. 24, no. 3, pp. 211–219, 1991.

[40] Y. Tan, T. N. Tan, Y. H. Wang, and Y. C. Fang, Do singular values contain adequate information for face recognition? *Pattern Recognit.*, vol. 36, no. 3, pp. 649–655, 2003.

[41] M. D. Yang, F. Z. Li, L. Zhang, and Z. Zhang, Lie group impression for deep learning, *Informat. Process. Lett.*, vol. 136, pp. 12–16, 2018.

[42] C. Gao, F. Z. Li, and C. Shen, Research on Lie Group kernel learning algorithm, (in Chinese), *J. Front. Comput. Sci. Technol.*, vol. 6, no. 11, pp. 1026–1038, 2012.

[43] V. M. Govindu, Lie-algebraic averaging for globally consistent motion estimation, in *Proc. 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2004, p. 8 161 455.

[44] R. Subbarao and P. Meer, Nonlinear mean shift for clustering over analytic manifolds, in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, New York, NY, USA, 2006.

[45] O. Tuzel, R. Subbarao, and P. Meer, Simultaneous multiple 3D motion estimation via mode finding on lie groups, in *Proc. 10^{th} IEEE Int. Conf. on Computer Vision*, Beijing, China, 2005, pp. 18–25.

[46] M. Moakher, Means and averaging in the group of

rotations, *SIAM J. Matrix Anal. Appl.*, vol. 24, no. 1, pp. 1–16, 2002.

[47] C. Gao and F. Z. Li, Lie group means learning algorithm, (in Chinese), *Pattern Recognit. Artif. Intell.*, vol. 25, no. 6, pp. 900–908, 2012.

[48] C. Gao, Research on Lie Group mean learning algorithm and its application, (in Chinese), master dissertation, Soochow University, Suzhou, China, 2012.

[49] B. Schölkopf, A. Smola, and K. R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation*, vol. 10, no. 5, pp, 1299–1319, 1998.

[50] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, Theoretical foundations of the potential function method in pattern recognition learning, *Automat. Remote Control*, vol. 25, no. 6, pp. 821–837, 1964.

[51] J. Cheng, Q. S. Liu, and H. Q. Lu, Texture classification using kernel independent component analysis, in *Proc. 17^{th} Int. Conf. on Pattern Recognition*, Cambridge, UK, 2004, p. 8 213 183.

[52] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, Fisher discriminant analysis with kernels, in *Neural Networks for Signal Processing IX: Proc. 1999 IEEE Signal Processing Society Workshop*, Madison, WI, USA, 1999, p. 6 497 095.

[53] C. Y. Xu, C. Y. Lu, J. B. Gao, T. J. Wang, and S. C. Yan, Facial analysis with a lie group kernel, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 7, pp. 1140–1150, 2015.

[54] B. N. Sheehan and Y. Saad, Higher order orthogonal iteration of tensors (HOOI) and its relation to PCA and GLRAM, in *Proc. SIAM Int. Conf. on Data Mining*, Minneapolis, MN, USA, 2007.

[55] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, Two-dimensional PCA: A new approach to appearance-based face representation and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, 2004.

[56] J. P. Ye, Generalized low rank approximations of matrices, *Mach. Learn.*, vol. 61, no. 1, pp. 167–191, 2005.

[57] M. Lu and F. Z. Li, Neighborhood-embedded tensor learning, (in Chinese), *J. Front. Comput. Sci. Technol.*, vol. 11, no. 7, pp. 1102–1113, 2017.

[58] J. T. Sun, H. J. Zeng, H. Liu, Y. C. Lu, and Y. Lu, CubeSVD: A novel approach to personalized web search, in *Proc. 14^{th} Int. Conf. on World Wide Web*, Chiba, Japan, 2005.

[59] X. F. He, D. Cai, and P. Niyogi, Tensor subspace analysis, in *Proc. 18^{th} Int. Conf. on Neural Information Processing Systems*, British Columbia, Canada, 2005, pp. 499–506.

[60] X. L. Li, Research and application on a data reduction method based on tensor field, (in Chinese), master dissertation, Soochow University, Suzhou, China, 2009.

[61] C. T. Lu, Research on tensor learning algorithm and its application on disease prediction, (in Chinese), master dissertation, Soochow University, Suzhou, China, 2017.

[62] J. H. Tang, X. B. Shu, Z. C. Li, Y. G. Jiang, and Q. Tian, Social anchor-unit graph regularized tensor completion for large-scale image retagging, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 2027–2034, 2019.

[63] J. Li, F. Z. Li, and L. Zhang, Image segmentation algorithm based on affine connection, (in Chinese), *J. Front. Comput. Sci. Technol.*, vol. 6, no. 3, pp. 267–274, 2012.

[64] Q. Xu, F. Z. Li, and P. Zou, The k-means algorithm based on finsler geometry, (in Chinese), *J. Univ. Sci. Technol. China*, vol. 7, pp. 570–575, 2014.

[65] M. Chen, S. P. He, and F. Z. Li, A lie group machine learning algorithm for linear classification, (in Chinese), *Microelectron. Comput.*, vol. 26, no. 10, pp. 170–173, 2009.

[66] M. Xian and F. Z. Li, Learning algorithm based on homology boundary, (in Chinese), *Comput. Eng. Appl.*, vol. 44, no. 21, pp. 192–194&204, 2008.

[67] F. Aiolli and A. Sperduti, A re-weighting strategy for improving margins, *Artif. Intell.*, vol. 137, nos. 1&2, pp. 197–216, 2002.

[68] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI–8, no. 6, pp. 679–698, 1986.

[69] M. M. Zhao and F. Z. Li, Neighborhood homology learning algorithm, (in Chinese), *CAAI Trans. Intell. Syst.*, vol. 9, no. 3, pp. 336–342, 2014.

[70] A. Perperidis, K. Dhaliwal, S. McLaughlin, and T. Vercauteren, Image computing for fibre-bundle endomicroscopy: A review, *Med. Image Anal.*, vol. 62, p. 101 620, 2020.

[71] N. Sochen, R. Kimmel, and R. Malladi, A general framework for low level vision, *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 310–318, 1998.

[72] D. W. Pearson, Approximating vertical vector fields for feedforward neural networks, *Appl. Math. Lett.*, vol. 9, no. 2, pp. 61–64, 1996.

[73] J. H. Chao and J. Kim, A fibre bundle model of surfaces and its generalization, in *Proc. 17^{th} Int. Conf. on Pattern Recognition*, Cambridge, UK, 2004, p. 8 221 516.

[74] L. L. Zhou and F. Z. Li, Research on mapping mechanism of learning expression, in *Proc. 5^{th} Int. Conf. on Rough Set and Knowledge Technology*, Beijing, China, 2010, pp. 298–303.

[75] T. R. Gao, The diffusion geometry of fibre bundles: Horizontal diffusion maps, *Appl. Comput. Harmonic Anal.*, doi: 10.1016/j.acha.2019.08.001.

[76] A. Asperti and G. Longo, *Categories Types and Structures: An Introduction to Category Theory for the Working Computer Scientist*. Cambridge, MA, USA: MIT Press, 1991.

[77] G. Muhiuddin, *Basic Concepts of Category Theory and Its Applications*. LAP LAMBERT Academic Publishing, 2018.

[78] X. X. Xu, F. Z. Li, L. Zhang, and Z. Zhang, The category representation of machine learning algorithm, (in Chinese), *J. Comput. Res. Dev.*, vol. 54, no. 11, pp. 2567–2575, 2017.

[79] M. Lu, L. Zhang, X. J. Zhao, and F. Z. Li, Constrained neighborhood preserving concept factorization for data representation, *Know. Based Syst.*, vol. 102, pp. 127–139, 2016.

[80] H. X. Xu, A semi-supervised learning algorithm based on Lie Group and application, (in Chinese), master dissertation, Soochow University, Suzhou, China, 2009.

[81] W. He, Research on the covering algorithm of machine

learning and its application, (in Chinese), master dissertation, Soochow University, Suzhou, China, 2011.
[82] M. X. Dong, Spectral estimation of image features manifold learning method, (in Chinese), master dissertation, Soochow University, Suzhou, China, 2012.
[83] M. D. Yang, F. Z. Li, and L. Zhang, Advances in the study of lie group machine learning in recent ten years, (in Chinese), *Chin. J. Comput.*, vol. 38, no. 7, pp. 1337–1356, 2015.
[84] Y. J. Huang and F. Z. Li, Isospectral manifold learning algorithm, (in Chinese), *J. Softw.*, vol. 24, no. 11, pp. 2656–2666, 2013.
[85] Y. J. Huang and F. Z. Li, Fast learning algorithm of spectral manifold, (in Chinese), *J. Front. Comput. Sci. Technol.*, vol. 8, no. 6, pp. 735–742, 2014.
[86] Y. Ren, Y. N. Wang, and J. Zhu, Spectral learning for supervised topic models, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 726–739, 2018.
[87] D. C. Tao, X. L. Li, X. D. Wu, W. M. Hu, and S. J. Maybank, Supervised tensor learning, *Knowl. Inf. Syst.*, vol. 13, no. 1, pp. 1–42, 2007.
[88] W. Fei, Y. N. Liu, and Y. T. Zhuang, Tensor-based transductive learning for multimodality video semantic concept detection, *IEEE Trans. Multimed.*, vol. 11, no. 5, pp. 868–878, 2009.
[89] X. L. Liu, T. J. Guo, L. F. He, and X. W. Yang, A low-rank approximation-based transductive support tensor machine for semisupervised classification, *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1825–1838, 2015.
[90] W. M. Hu, J. Gao, J. L. Xing, C. Zhang, and S. Maybank, Semi-supervised tensor-based graph embedding learning and its application to visual discriminant tracking, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 172–188, 2017.
[91] T. Huynh-The, C. H. Hua, T. T. Ngo, and D. S. Kim, Image representation of pose-transition feature for 3D skeleton-based action recognition, *Inf. Sci.*, vol. 513, pp. 112–126, 2020.
[92] R. Vemulapalli, F. Arrate, and R. Chellappa, Human action recognition by representing 3D skeletons as points in a lie group, in *Proc. 2014 IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, p. 14 632 342.
[93] Z. Liu, C. Y. Zhang, and Y. L. Tian, 3D-based deep convolutional neural network for action recognition with depth sequences, *Image Vision Comput.*, vol. 55, pp. 93–100, 2016.
[94] X. Y. Cai, W. G. Zhou, and H. Q. Li, Attribute mining for scalable 3D human action recognition, in *Proc. 23rd ACM Int. Conf. on Multimedia*, Brisbane, Australia, 2015, pp. 1075–1078.
[95] H. L. Zhang, P. Zhong, J. L. He, and C. X. Xia, Combining depth-skeleton feature with sparse coding for action recognition, *Neurocomputing*, vol. 230, pp. 417–426, 2017.
[96] J. C. Núñez, R. Cabido, J. J. Pantrigo, A. S. Montemayor, and J. F. Vélez, Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition, *Pattern Recognit.*, vol. 76, pp. 80–94, 2018.
[97] G. G. Demisse, D. Aouada, and B. Ottersten, Deformation based curved shape representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1338–1351, 2018.
[98] C. Von Tycowicz, F. Ambellan, A. Mukhopadhyay, and S. Zachow, An efficient riemannian statistical shape model using differential coordinates: With application to the classification of data from the osteoarthritis initiative, *Med. Image Anal.*, vol. 43, pp. 1–9, 2018.
[99] M. Hayat, M. Bennamoun, and A. A. El-Sallam, An RGB-D based image set classification for robust face recognition from Kinect data, *Neurocomputing*, vol. 171, pp. 889–900, 2016.
[100] M. Yin, Z. Z. Wu, D. M. Shi, J. B. Gao, and S. L. Xie, Locally adaptive sparse representation on riemannian manifolds for robust classification, *Neurocomputing*, vol. 310, pp. 69–76, 2018.
[101] T. Wang, M. N. Qiao, Y. Chen, J. Chen, A. C. Zhu, and H. Snoussi, Video feature descriptor combining motion and appearance cues with length-invariant characteristics, *Optik*, vol. 157, pp. 1143–1154, 2018.
[102] J. M. Dong, Y. X. Peng, S. H. Ying, and Z. Y. Hu, Lietricp: An improvement of trimmed iterative closest point algorithm, *Neurocomputing*, vol. 140, pp. 67–76, 2014.
[103] S. H. Ying, Y. W. Wang, Z. J. Wen, and Y. P. Lin, Nonlinear 2D shape registration via thin-plate spline and lie group representation, *Neurocomputing*, vol. 195, pp. 129–136, 2016.

Mei Lu received the PhD degree in computer science and technology from Soochow University in 2016. Now she is an associate professor and master supervisor at Jiangsu Normal University. Her main research interests include machine learning, data mining, big data science and engineering, etc.

Fanzhang Li received the MS degree in computer science and technology from University of Science and Technology of China in 1995. Now he is a professor and PhD supervisor at Soochow University. His main research interests include Lie group machine learning, big data science and engineering, dynamic fuzzy logic, etc.