



Feature dimensionality reduction: a review

Weikuan Jia¹ · Meili Sun¹ · Jian Lian² · Sujuan Hou¹

Received: 5 August 2021 / Accepted: 23 December 2021 / Published online: 21 January 2022
© The Author(s) 2022

Abstract

As basic research, it has also received increasing attention from people that the “curse of dimensionality” will lead to increase the cost of data storage and computing; it also influences the efficiency and accuracy of dealing with problems. Feature dimensionality reduction as a key link in the process of pattern recognition has become one hot and difficulty spot in the field of pattern recognition, machine learning and data mining. It is one of the most challenging research fields, which has been favored by most of the scholars’ attention. How to implement “low loss” in the process of feature dimension reduction, keep the nature of the original data, find out the best mapping and get the optimal low dimensional data are the keys aims of the research. In this paper, two-dimensionality reduction methods, feature selection and feature extraction, are introduced; the current mainstream dimensionality reduction algorithms are analyzed, including the method for small sample and method based on deep learning. For each algorithm, examples of their application are given and the advantages and disadvantages of these methods are evaluated.

Keywords Curse of dimensionality · Dimension reduction · Feature selection · Feature extraction

Introduction

As a basic research, datasets with many characteristics are called high-dimensional data, it has also received increasing attention from people. The growth and update speed of data sets are accelerating, and the data is developing in a high-dimensional and unstructured direction. Massive and complex data contains a lot of useful information, but it also increases the difficulty to use the data effectively. For example, the problem called the “curse of dimensionality” appears due to the rapid and large-scale expansion of dimensions [1–3]. Quite a lot of computing time and storage space are spent on the processing of the data. Effective information is submerged in complex data, making it difficult to discover the essential characteristics of the data. It takes lots of time and manpower to process the data. And this problem also has a bad influence on the accuracy of the

recognition. In Fig. 1, taking the performance of the classifier as an example, when the data dimension increases, the performance of the classifier becomes better; when the data dimension continues to increase, the performance of the classifier becomes worse. How to analyze the huge amount of information and extract useful information features from high-dimensionality data, as well as eliminate the influence of related or repetitive factors. In other words, the problems need to be solved by dimension reduction. The basic principle of feature dimensionality reduction is to map a data sample from a high-dimensional space to a relatively low-dimensional space. Its basic task is to find the mapping and obtain an effective low-dimensional structure hidden in high-dimensional observable data [4–7].

The process of mapping high-dimensional data to low-dimensional space through projections will inevitably lead to the loss of some original information. The problem that needs to be resolved at present is to obtain useful reduction data from the high-dimensional data set to meet the recognition accuracy and storage requirements under the premise of maintaining the essential characteristics of the original data optimally. However, in many practical situations, the identification and acquisition of effective features are often not so easy. It makes dimension reduction become one of the most important and difficult tasks in the field of pattern recognition, data mining, and machine learning. It has transferred to

✉ Weikuan Jia
jwk_1982@163.com

✉ Sujuan Hou
hsj1985@126.com

¹ School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China

² School of Intelligent Engineering, Shandong Management University, Jinan 250357, China

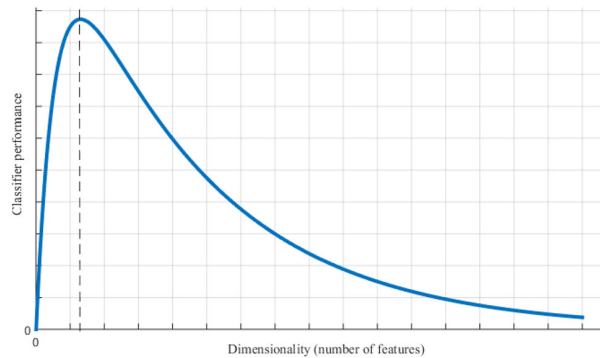


Fig. 1 The trend of classifier performance with the change of dimension

some important tasks in sugar content prediction [8], DNA microarray [9], and other tasks. As basic research, dimension reduction has also received increasing attention from people. A large number of domestic and foreign researchers have devoted themselves to these fields. The various algorithms proposed by them have solved the problem of information dimension reduction to some extent, but these methods also have deficiencies. Many scholars have proposed new insights which make the research of pattern feature dimension reduction take a big step forward. The following is a discussion of the research progress of dimension reduction in recent years.

Principle of feature dimensionality reduction

Datasets with many characteristics are called high-dimensional data. There are often lots of redundant information in it, including related or duplicated factors. The dimension reduction is to eliminate these interferences. Feature dimensionality reduction uses existing feature parameters to form a low-dimensional feature space and overcomes the effects of redundant or irrelevant information, so to map the effective information contained in the original features to fewer features.

In the mathematical sense, suppose there is a n -dimensional vector

$$X = [x_1, x_2, \dots, x_n]^T, \quad (1)$$

X are mapped to a m -dimensional vector Y through a map f , where

$$Y = [y_1, y_2, \dots, y_m]^T, \quad (2)$$

and

$$m \ll n. \quad (3)$$

Vector Y should contain the main features of vector X . Mathematically, the mapping function can be expressed as

$$Y = f(X). \quad (4)$$

This is the process of feature extraction and selection [10, 11]. It can also be called the "low loss reduction dimension" process of the original data. A low-dimensional vector as a result of dimension reduction can be applied to the fields of pattern recognition, data mining, and machine learning.

This mapping f is the algorithm that we want to find for feature reduction. The choice of mapping f differs depending on the pending problem.

Feature selection

Feature selection can also be called variable selection or feature subset selection, and it is a process of selecting feature subsets that are applied to model construction [12]. There are four reasons for the use of feature selection techniques: simplifying the model to make it easier for researchers (users) to interpret; shortening run time; avoiding curses of dimensionality; enhancing generalization by reducing excessive fitting (formally reducing variance).

The most important prerequisite for using feature selection techniques is that the data contains many redundant or related features that can be deleted without losing a lot of information. Redundant features or related features are two different concepts, because a related feature may be redundant in the presence of other related features that are closely related to it. Feature selection is generally used in areas where there are many features and relatively few samples (or data points), as shown in Fig. 2 [13, 14].

Feature extraction

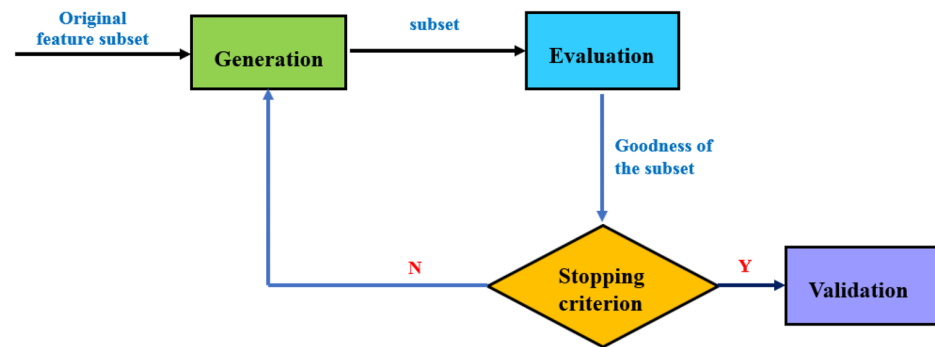
Feature extraction implements two functions:

1. Separate effective information from redundant data.
2. Reduce the operations performed by the classifier through reducing the dimension.

Feature extraction generates new features from the original features, which means that the new feature after feature extraction is a mapping of the original features. Its advantage is that the compression of new features is more efficient. The disadvantage is that when the original feature set has an obvious physical meaning, the new features may lose meaning [15, 16].

Feature selection algorithms

In general, feature selection can be viewed as a search optimization problem. The search of the minimum feature subset

Fig. 2 Process of feature selection

is proved to be an NP problem. Only the exhaustive search can find the optimal solution. The exhaustive search process usually has a relatively large computational cost. To find the optimal feature subset, the M -feature combinations of all possible N original features must be searched. This combination explosion leads to an exponential increase in the amount of calculation with the increase of the total number of features. So in most practical situations, the search-optimized exhaustive search cannot be achieved [17].

For this reason, people want to search for suboptimal solutions. Usually the feature selection algorithm should define the following elements: (1) search starting point and search direction (2) search strategy; (3) feature evaluation criterion; (4) stop criterion.

Current research focuses on two aspects: search strategy and evaluation criteria.

Based on the search strategy

Complete search algorithm

1. Branch and bound search (BBS)

The BBS algorithm is a method for finding solutions on the solution space tree of the problem [18]. It usually uses the method of minimum cost or breadth first to search on the solution space tree. Its main idea is “pruning”. BBS is an exhaustive optimization algorithm. Adding branch boundaries on the basis of exhaustive algorithm development can greatly reduce the number of scenarios that need to be calculated. When using the BBS algorithm, the upper and lower bounds of the target value should be determined at first. When the node becomes an expansion node, all its child nodes will be found for one time. Those child nodes that cannot be used or result in non-optimal solutions are discarded. The remaining nodes are added to the active node table. After that, take the next node in the active node table and repeat the above process. Until the feasible solution to the problem is found or the live node table is empty.

Wang provided us with an example of a BBS algorithm application for the analysis of unbalanced meteorological

data based on the branch-and-bound algorithm [19]. To meet the real-time requirements required for short-term weather forecasting and optimize the data, a logical paradigm is constructed using the BBS algorithm, and the potential correlation between meteorological data is discovered from a more detailed point of view. In addition, in Liquid Composite Molding (LCM) [20], the BBS is used for filling the LCM process. It realizes the optimal solution by dividing the solution set into smaller sets and eliminating sets that cannot contain the optimal solution. The results of using BBS to find the gate position that can produce the shortest fill time, and find the auxiliary gate position that can cancel the interference in the filling process, besides reduce the size of the dry point are more efficient and accurate than the results of exhaustive search and genetic algorithms. In addition to those above examples, the BBS algorithm is also used in circuit layout and loading problems.

In the existing precise method, the application of the BBS algorithm is very extensive, especially in terms of a machine scheduling problem. Although the BBS algorithm is an exhaustive algorithm optimization and reduces the computational solution, it still brings a huge computational cost.

2. Beam search (BS)

The BS algorithm searches for all promising nodes layer by layer and does not backtrack [21]. The number of nodes explored for each level is called the beam width. The evaluation function is used to give a branch that continues searching. This algorithm is mainly used to search in the decision tree, especially where the search space is large. It is not an iterative process, but a constructive one, therefore less computing time is needed. In addition, the BS algorithm has additional ideal properties. For example, it can generate job sequences of different lengths, which makes it easier to schedule.

Kumar introduced the application of the BS algorithm in multi-label learning [22]. BS algorithm is used for reasoning processing, and BS is combined with data training to determine the appropriate tag order, which provides a state-of-the-art method for multi-label learning. In addition, the BS algorithm has also been applied to the solution of prac-

tical problems, such as the container loading problem [23]. The use of the BS algorithm in this problem is superior to all other methods and can maximize the use of container space.

However, the BS algorithm also faces the problem of balancing the relationship between fast but poor evaluation functions and higher computational requirements. Based on the development of the BS algorithm, a filtered directional search FBS was proposed. The FBS algorithm can take care of both the amount of computation and the efficiency of the search. In general, a successful FBS for a specific problem should specify the following four elements [24]: (1) Find the search space representation defined by the solution space; (2) Determination of beam width and filter width; (3) The formation process of branches; (4) Selection of evaluation functions.

Heuristic search algorithm

Heuristic search is under the guidance of "experience" "wise choice" and "expectation". A good suboptimal value or a global optimal value is found before the other invalid subsets. The actual performance of heuristic search depends on the design of the heuristic algorithm and the current problems. When time is sufficient, the algorithm can find the optimal feature subset. Common heuristic searches include sequence search, bidirectional search, sequence float selection, and so on. Here we introduce sequence search and bidirectional search.

1. Sequence search algorithm

Classical sequence search algorithms include Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) [25].

The SFS algorithm [26] is a top-down heuristic search. First, it initializes a subset of features to be an empty set and selects one feature which makes the evaluation function get the optimal value at a time. In fact, it is a greedy algorithm that can select important classification features and is widely used in feature selection. However, using this method can only increase features and cannot delete features, and it does not take the redundancy and correlation between features into account.

Therefore, a number of the improved algorithms based on SFS were proposed. For example, the literature [27] proposed an improved SFS algorithm that aims at the problems of conventional SFS methods: adding features sequentially to the previously evaluated optimal subset until a stop criterion is reached (probably no performance improvement), only consider getting the optimal subset from the previous steps to move on to the next step. The improved algorithm proposes to add a standard, through which the collection can be evaluated in the next step to limit the search. In medical

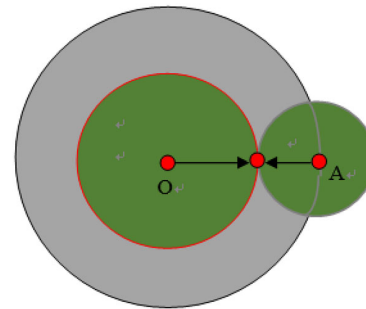


Fig. 3 Schematic diagram of Bi-directional search

data, there is usually no unique combination of features to provide the best interpretation of the results. This algorithm solves the problem of selecting physiological variables in patients with septic shock and obtains the best performance combination currently. This allows the SFS algorithm to be further enriched and developed.

The sequential backward search algorithm SBS is the opposite of SFS: starting from a feature full set, deleting one feature that makes the feature function optimal at a time. When this algorithm is used to select features, features can be deleted and cannot be added.

2. Bi-directional search (BDS)

The bi-directional search contains two separate searches [28]. The search from the starting node to the target node is called forward search, and the corresponding search is called reverse search. When the two directions search generate the same child node, the search process ends. As shown in Fig. 3, the starting point of the BDS is shown as follows. Point O represents the search starting point and point A represents the search target. The gray circle represents the possible search range of the unidirectional search. The two green circles represent the search range of a two-way search, and it is easy to prove that the green area is smaller than the gray one.

The bi-directional search BDS is used both in artificial intelligence and path planning [29]. Its key step is to design a mechanism for merging two partial searches that need to be customized for a given search problem. Without a proper design, the bidirectional search may be worse than the unidirectional search, because it must try to prevent the confusion of the two search boundaries. Kandl proposed the first successful bidirectional search method and proved that it was more efficient than one-way searching [30].

For different issues, the bidirectional search strategy is also different. For example, in the case of the peak adjustment of cascaded hydropower stations, a forward-backward heuristic search method was proposed creatively. Using this method to solve the model, the goal of coordinated optimization of power plants was achieved. Zhou proposed an incremental algorithm for differential constraint systems [31]

and the bi-directional search was used in it. Experimental results show that this method is much faster than a one-way search.

Random search algorithm

The random search algorithm uses a random function. Common search algorithms include genetic algorithms, ant colony algorithms, simulated annealing, and tabu search. This paper focuses on genetic algorithms and ant colony algorithms.

1. Genetic algorithm (GA)

GA is a search strategy based on the analogy natural selection theory directly. It is an evolutionary algorithm consisting of four parts: a group of individuals (or chromosomes) that can represent a possible solution; an appropriate function to evaluate individual fitness; a selection function to select the individual which is suitable to produce the next generation; there is also a genetic operator, such as crossover and mutation, to explore the new search space.

Figure 4 clearly shows the GA algorithm [32]. Where Gen is the number of genetic generations, i is the total number of individuals which have been processed. When $i = M$, go to the next generation. M is the population size of current generation.

Each alternative has a series of attributes (chromosomes or genotypes) that are normally represented by a string of 0 and 1 binary characters. In each generation, individuals with better adaptability are selected in the current population, and genomes of all individuals have been modified to produce a new generation through random mutations and crossovers. The algorithm terminates when the generated number reaches the maximum value supported or reaches a satisfactory level of health.

The advantages of genetic algorithms are as follows: (1) good parallelism; (2) a wide range of applicability; (3) better robustness and global optimization performance; (4) simple and effective operation.

GA is widely used in practical problems because of its advantages. In the dynamic airspace configuration [33], GA is used to find a highly reliable BES model (building energy simulation model) that captures the thermal behavior of buildings widely and ensures the advanced nature of the verification plan. Pezzella showed a GA for Flexible Job Shop Scheduling (FJSP) that integrates different initial population generation strategies and selects individuals for breeding [34]. The calculation result shows that integrating more strategies in the genetic framework will bring better results. This result is combined with the flexibility of the genetic paradigm, which proves that GA is an effective method to solve FJSP.

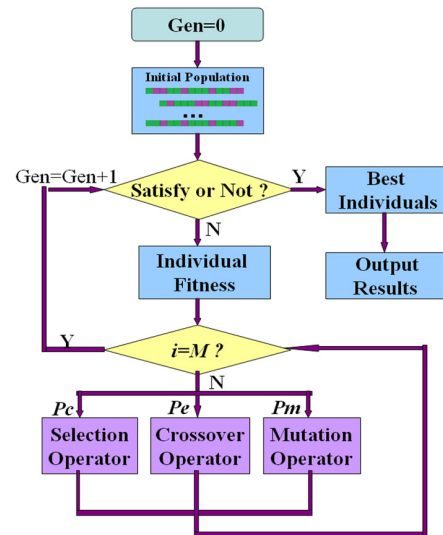


Fig. 4 Flow chart of genetic algorithm

GA has less computing time and better convergence and can be well applied in practical problems. However, GA can only solve small-scale problems and cannot be applied to large-scale calculations. The optimization and improvement of genetic algorithms are the keys to solve this problem in the future.

2. Ant colony algorithm (ACO)

As a classical algorithm in swarm intelligence search, ACO is an algorithm applied to discrete optimization [35]. It simulates the actual ant colony behavior of ants during foraging [36]. The algorithm steps are as follows:

1. The ant explores the region randomly for food.
2. The ants moved the food back to the cave and leave traces of chemical pheromone.
3. The amount of pheromone increased with the increase of the amount of food.
4. Other ants find food sources based on the pheromone trace.

The first step involves the initialization of pheromone traces. Then according to the probabilistic state transition rules, each ant creates a solution which depends on the state of the pheromone. Finally, the number of pheromones changes in two phases: one is the evaporation phase, during which a small portion of pheromones is evaporated; the other is the intensive phase, in which each ant has a large number of pheromones, and the number of pheromones is proportional to the adaptability of the solution. This process is iterative until the criteria are stopped. ACO algorithm is an iterative process, the flow chart as shown in Fig. 5.

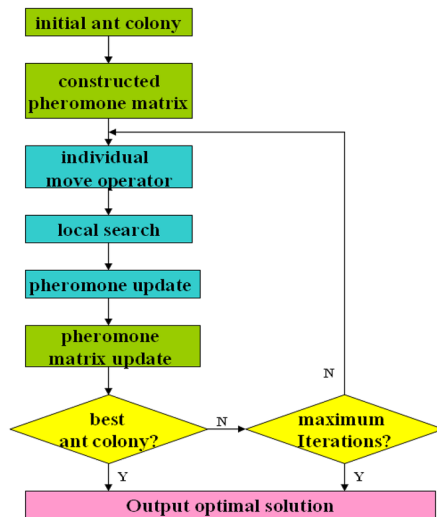


Fig. 5 flow chart of ACO algorithm

Wind energy can be used as a clean power generation method. People want to make the best use of wind energy. Eroğlu proposed an algorithm based on the ant colony algorithm for maximizing the expected energy output [37], it considering the wake loss based on wind turbine position and wind direction fully. The results show that in the appropriate solution time, using the ant colony algorithm can find a better layout of the wind farm and not just local optimal solution, and the performance of the algorithm is better than the existing continuous problem algorithm. In addition, the ACO algorithm has also been applied in traffic signal timing optimization [38] and multi-level vehicle routing problem [39].

The ACO algorithm uses the collective search feature of the ant colony to find the shortest path from the cave to the food. It is widely used in data analysis, robot collaborative problem solving, power, communications, water conservancy, traffic, etc. It shows its advantages in exploring unknown solutions and proves its excellent ability to search the global. However, the ACO algorithm also has some disadvantages such as long search time and convergence to non-optimal solutions. When the size of the problem is relatively large, it is not suitable for using the ant colony algorithm.

Based on evaluation criterion

The feature selection method differentiates from the feature set evaluation strategy and is divided into Filter and Wrapper [40].

Filter uses the statistical performance of the data it trains for feature evaluation and has no relation to the subsequent learning algorithm. It has a faster speed, but the evaluation

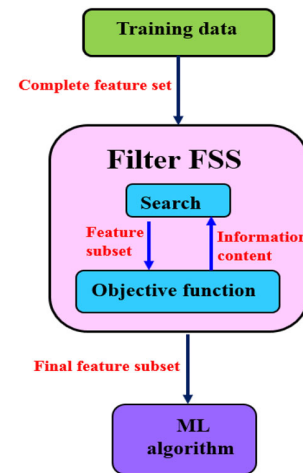


Fig. 6 Principle of the filter method

results differ greatly from the performance of subsequent learning algorithms. Wrapper evaluates feature subsets using the training accuracy of follow-up learning algorithms, so that small deviations can be achieved, but the amount of calculation is large and it is not suitable for large data sets.

Filter method

Filter regards evaluation criteria as the criteria for variable selection and uses a more appropriate criterion to evaluate the selected features quickly. The filter usually finds a suitable standard to evaluate variables and use a threshold to remove variables below this threshold to filter out less relevant variables, to reduce the degree of correlation between features and increase the degree of correlation between features and classes. The filter is simple and practical, and it is widely used. The principle of filter is shown in Fig. 6.

The evaluation criteria are divided into the following four categories: distance-based (Euclidean distance, Mahalanobis distance, Bhattacharyya distance, etc.), information-based (Shannon entropy, conditional entropy, information gain, mutual information, etc.), independence-based (relevance, Similarity) and consistency based.

Wrapper method

Wrapper incorporates the process of feature selection into algorithm learning. The predictor is viewed as a black box. The prediction performance is used as an objective function to evaluate the subset of variables. Some search algorithms can be used to find a subset of objective function variables that maximize classification performance. The principle of the wrapper is shown in Fig. 7.

Unlike the filter method, the wrapper method is based on three component methods: search strategy, predictor, and

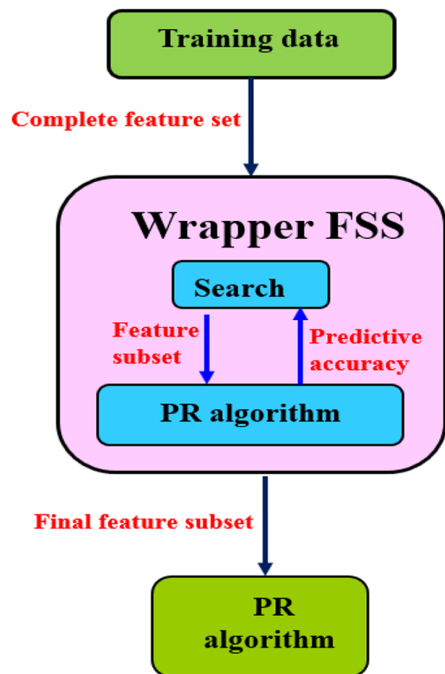


Fig. 7 Principle of the wrapper method

evaluation function. The subset of features that are evaluated is determined by the search strategy. The predictor can be any classification method. Its performance is used as an objective function to evaluate the feature subset determined by the search strategy to find the optimal subset.

The wrapper is better than the Filter, but it takes more time and requires more computing resources.

Based on the optimized algorithm

The best way to search for the optimal feature subset is the exhaustive method, but it is only suitable for certain situations; in addition to BBS [41], its precondition for use is that the discriminant function is a monotonically increasing function of the number of features, but it's usually difficult to reach. Then, based on heuristic rules, people proposed some optimized search algorithms, such as feature selection based on Tabu search [42], feature selection based on mathematical programming [43] and so on. When the heuristic rules are reasonable, a well-designed optimal search algorithm can generally be obtained. It does not check all the feature combinations, but it can estimate a set of implicit and effective feature combinations and even sort all features according to heuristic rules. If the rules are set properly, this kind of algorithm can get almost the same effect as the previous two search strategies in the application, and its calculation speed is faster. Such algorithms can achieve good results when the dimension is not very large; however, in the thousands of

dimensions, even in the feature space of the upper million, it seems powerless. In this case, a single feature optimization algorithm such as mutual information [44] and other feature selection algorithms are used.

The method of heuristic search strategy based on Support Vector Machine (SVM) is the hot spot of research nowadays. It needs to select a very reliable classification algorithm as the basis for feature selection. This is also a huge challenge for the use of this method. There are many feature selection strategies based on SVM [45], the improved methods such as based on the Recursive Feature Replacement (SVM)-based Recursive Feature Replacement method [46]. Its evaluation criterion is the cross-validation error rate of the feature set. It sets the feature subset to be empty at first, and then adds new features with the smallest cross-validation error rate until all the features are sorted one by one. These methods have all been well verified.

Feature extraction algorithm

Feature extraction algorithms are divided into two categories: linear and nonlinear. Linear methods are easier to calculate than nonlinear methods and they can be parsed, so early data dimensionality reduction uses mostly linear methods. However, since many of the problems we meet are nonlinear and time-varying systems, the current research on nonlinear feature reduction methods is relatively more [47].

Linear feature extraction algorithm

Based on variance total contribution ratio

Suppose there are n variables in the original sample, denoted by $X = x_1, x_2, \dots, x_n$, through the orthogonal transformation, integrated into n comprehensive variables, namely:

$$\begin{cases} y_1 = c_{11}x_1 + c_{12}x_2 + \dots + c_{1p}x_n \\ y_2 = c_{21}x_1 + c_{22}x_2 + \dots + c_{2p}x_n \\ \dots \\ y_n = c_{n1}x_1 + c_{n2}x_2 + \dots + c_{nn}x_n \end{cases} \quad (5)$$

and they meet the following equation:

$$c_{k1}^2 + c_{k2}^2 + \dots + c_{kn}^2 = 1, k = 1, 2, \dots, n, \quad (6)$$

In which y_i and y_j ($i \neq j, i, j = 1, 2, \dots, p$) are independent; thus, X variance is transferred to the comprehensive variables y_1, y_2, \dots, y_n .

By the correlation coefficient matrix R of sample X , using Jacobi method, solution of the roots of the characteristic equation

$$|\lambda I - R| = 0. \quad (7)$$

We can get n non-negative eigenvalues λ_i ($i = 1, 2, \dots, n$) of the correlation coefficient matrix of the sample, carries on the sorting, there are $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$. From n comprehensive variables extract front m features, the proportion of the variance of the former m principal components taking up all of the variances can be defined as VTCR, denote by α :

$$\alpha = \frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^n \lambda_i}. \quad (8)$$

In practice applications, m is determined by the value of α , which plays the role of dimension reduction, and α represents the original data information included in feature extraction.

The statistical analysis theory can analyze the statistical laws in the context of several objects and indicators. It is a comprehensive analysis method and is one of the commonly used feature extraction methods. The theoretical basis of this method is relatively strong, and it contains many well-developed algorithms that can effectively analyze and process the data. To statistically analyze the characteristics of the data or to classify the data subsets, it is necessary to make the data set satisfy statistically unrelated assumptions. For example, more representative is Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Factor analysis (FA).

1. Principal component analysis

PCA is a kind of statistical analysis method that transforms several feature indicators into a few comprehensive indicators from the perspective of feature validity. PCA allows the original complex variable to be represented by several integrated factors that reflect the information contained in the original variable as much as possible, and these factors do not relate to each other, to achieve the purpose of simplification [48, 49]. If there are n samples, the number of indicators measured by a single sample is p , so there are a total of np data, but the indicators usually interact with each other, and the PCA is to study how to find the principal components from the indicators.

Here, the principal components are required to reflect as much as possible the information contained in the original data, and these principal components should be independent of each other. In the sense of global minimum reconstruction error, the high-dimensional observation data is transformed into sub-spaces with lower dimensions through projection. The sub-space generated

by the Eigen vectors corresponding to the largest eigenvalues of the data covariance matrix is exactly satisfied. Based on the above condition, PCA has perfect theoretical and practical feasibility, but its feasibility is based on the premise that the data is embedded in the global linear or approximately linear low-dimensional space; it largely retains the second-order matrix information in the original data, which is the best and simplest of the original data, but the variance does not fully reflect the amount of information, and the classification information in the original data is not well used, and even the compressed data is not conducive to pattern classification.

PCA uses a linear combination of variables to represent the principal components. The only prerequisite for the existence of principal components is that the eigenvalues of the given covariance matrix or correlation matrix are unique. The total variance of the explanatory variables is the focus of the PCA, but under normal circumstances, variance does not fully reflect the amount of information. Some scholars proposed a two-dimensional principal component analysis [50] based on PCA to extract the statistical features of palmprint images and demonstrated that its generalization ability is better than that of traditional PCA. Thus, the improved two-dimensional principal component analysis also emerged. It cannot only ensure the overall divergence of the training sample image, but also can be better used for feature acquisition to improve the recognition ability. It effectively reduces the feature dimension of the original algorithm, and improves the complexity of the recognition calculation, and further improves system availability. In addition, PCA-based multiple principal component analysis [51] discusses methods for solid solution phases.

2. Linear discriminant analysis

As a typical representative of the linear method, the main task of LDA is to convert the original sample through the projection to the best discriminant vector space to play a role in extracting the classification information and reducing the dimension, so that the data samples after projection have the largest interclass distance and the smallest intraclass distance (maximum inter-class scatter matrix and smallest intraclass scatter matrix).

In addition, there are incremental principal component analysis (Incremental PCA, IPCA) and incremental discriminant analysis. And a new incremental facial feature extraction method—incremental weighted average sample analysis for real-time face recognition is proposed [52]. Semi-supervised linear discriminant analysis (SLDA) is proposed, which can use the limited number of labeled data and a quantity of the unlabeled ones for training so that LDA can accommodate the situation of a few labeled data available [53]. Statistically uncorrelated identification analysis method [54] and other methods

solve the shortcomings of classical mathematical methods well.

The above statistical analysis methods analyze the characteristics of the data or classify data subsets in a statistical sense when analyzing the characteristics of the data, and they do not from the perspective of the information to examine the characteristics of the data, analyzes the information content of the data subset, and evaluate model reliability and effectiveness. Moreover, the statistical analysis method usually assumes that the data set statistics are irrelevant, and this assumption cannot be satisfied in many cases of high-dimensional data.

3. Factor analysis (FA)

The basic philosophy of FA is to divide the observation variables into several classes, make the ones which are related close in the same class, the relativity between the variables of different classes is lower, then each class of variables represents a basic structure in fact, that is the public factor. Then we can discover each variable's best subset from numerous factors, describe the multivariable systems results and the influence on the system of the various factor from the information included in the subsets.

The FA algorithm and the PCA algorithm are slightly different, supposes the observable random vector $X_i = x_1, x_2, \dots, x_n$ but the unobservable vector $F_j = F_1, F_2, \dots, F_m$

$$X_i = \sum_{j=1}^m a_{ij} F_j + c_i \varepsilon_i, i = 1, 2, \dots, n; j = 1, 2, \dots, m. \quad (9)$$

In this formula, $n > m$, a_{ij} is the factor loading that represents the correlation coefficient of the i th variable and the j th factor and reflects the importance of the i th variable to the j th factor. F is called a public factor, they are the factors which appear in the expression of each original observation variable, and are mutually independent unobservable theoretical variables. c_j represents the load of the unique factor, ε_i affects the unique factor of X_i . The basic question of FA is to decide the factor loading by the correlation coefficient between variables. Supposes A is the factor loading matrix, namely,

$$A = (a_{ij})_{n \times m}. \quad (10)$$

Determine the number of factors extracted according to the value of α , further get A , calculate the synthesis score of the factors to achieve dimensionality reduction.

Dehak proposed a new speaker representation for speaker verification. In this modeling [55], a new low-dimensional speaker- and channel-dependent space is defined using simple factor analysis. This space is named the total variability

space because it models both speaker and channel variabilities.

Independent component analysis

Independent Component Analysis (ICA) [56, 57] is a new statistical method developed recently. The purpose of this method is to linearly decompose the observed data into statistically independent components.

The main principle of ICA is to adopt an implicit statistical variable model:

$$x = As. \quad (11)$$

This statistical model is called the ICA model. Its meaning is that the independent components are mixed with each other to get the data that can be observed. The independent component s is a potentially variable quantity, the mixing matrix A is assumed to be unknown, and only the random vector x can be directly observed, and it is necessary to estimate A and s under a small number of conditions.

The ICA assumes that the components are statistically independent, non-Gaussian distributed, and the unknown mixed matrix is a square matrix. If the inverse W of A can be calculated, the independent components can be calculated from

$$s = Wx. \quad (12)$$

It follows that the ICA model cannot determine the variance and order of independent components.

The improved independent component analysis algorithm based on the sparsity of the basis function is used for image feature extraction [58]. This algorithm does not require sophisticated optimization of high-order nonlinear comparison functions, so it has good sparsity and fast convergence speed. The literature [59] used ICA for face recognition studies and demonstrated that ICA has a very broad prospect of development fully. However, since ICA has only appeared and developed in recent years, its theory and algorithm are still not mature, and there are some contents that need to be supplemented and improved. Wang presented an independent component analysis approach to dimensionality reduction [60], to be called ICA-DR which uses mutual information as a criterion to measure data statistical independency that exceeds second-order statistics. As a result, the ICA-DR can capture information that cannot be retained or preserved by second-order statistics-based dimensionality reduction techniques. Independent component analysis (ICA) also has been used for the feature extraction of microarray gene expression data in numerous works [61]. For microarray data, Musheer et al. proposed a novel (artificial bee colony) ABC-based

feature selection approach [62], including two stages: ICA-based extraction method and ABC-based wrapper approach, respectively. Therefore, the merit of ICA is that the number of extracted features is always equal to the number of samples.

Multi-dimensional scaling analysis

Multi-dimensional scaling (MDS) is a method of dimension reduction that aims to visualize the differences and is widely used in multidimensional data analysis in some scientific fields such as image retrieval [63], common-sense reasoning [64], and so on. Its goal is to find multidimensional data projections in the lower-dimensional space (R2 or R3) to maintain the similarity or inconsistency of the data. It optimally maps the object's proximity index to the distance between the multidimensional spatial points and visualizes the data so that users can test structured assumptions or discover hidden patterns in the data. MDS compresses large data containing several variables into a lower-dimensional space to obtain an intuitive spatial graph, and uses points within the space to represent implicit relationships between variables. After reduction the distance between two points in the low-dimensional space is the same as that in the original high-dimensional space, and when visualizing similar objects more similarly, the difference between different objects will be greater.

Steps of the classical multidimensional scaling analysis algorithm:

Step 1 Construct matrix $A = [a_{ij}] = -\frac{1}{2}d_{ij}^2$ according to the distance matrix $D = [d_{ij}] \in R^{n \times n}$.

Step 2 Calculate the inner product matrix $B = [b_{ij}] = [a_{ij} - \bar{a}_i - \bar{a}_j + \bar{a}..]$.

Step 3 Calculate the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ and eigenvectors of B . The distance matrix here is the Euclidean matrix, so the eigenvalues are negative. If there is a negative eigenvalue, the matrix must not be Euclidean. Make

$$S_k = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^n |\lambda_i|}, \quad (13)$$

and S_k is equivalent to the cumulative contribution rate in principal component analysis. Of course, we hope that the value of k is not too large, and the cumulative contribution rate is large. In the example of the predicted city coordinates proposed above, since x_i is two-dimensional coordinate data, when the k value is 2, the cumulative contribution rate is almost equal to 100%. However, for general data, the cumulative contribution rate of two-dimensional data may not be maintained at 100%. This needs to be judged according to the situation.

Step 4 After determining a k value, refactoring $\tilde{X} = E_k \Lambda_k^{1/2}$. E_k is a matrix composed of the first k eigenvectors

retained by matrix B , and Λ_k is a diagonal matrix composed of k eigenvalues.

It can be seen from the above derivation that the classical multidimensional scaling and principal component analysis are essentially the same, the difference is that the multidimensional scaling is based on the sample, and the principal component analysis is based on variables. Moreover, it can be proved that the k -dimensional principal coordinate of \tilde{X} after classical multidimensional scaling is just the value of the first k principal components obtained by principal component analysis after X -centering.

The original data of the MDS is a quadratic symmetric matrix, which is the similarity (dissimilarity) of the analysis objects. The simplest example is the Euclidean distance matrix. Usually, they are not necessarily mathematical distances [65].

Singular-value decomposition

In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix. It is the generalization of the Eigen decomposition of a positive semidefinite normal matrix (for example, symmetric matrix with positive eigenvalues) to any $m \times n$ matrix via an extension of polar decomposition [66].

For singular values, they are similar to the eigenvalues in the Eigen decomposition and are also arranged in order from large to small in the singular value matrix. And the reduction of singular values is especially fast. In many cases, the sum of the singular values of the top 10% or even the first 1% accounts for more than 99% of the sum of all singular values. That is,

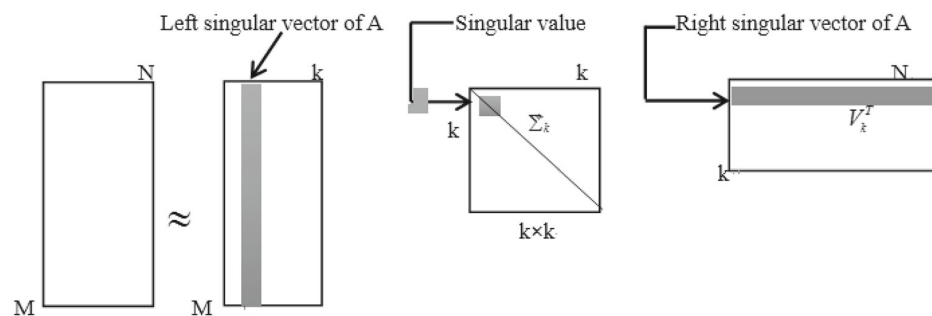
$$A_{m \times n} = U_{m \times m} \Sigma_{n \times n} V_{n \times n}^T \approx U_{m \times k} \Sigma_{k \times k} V_{k \times n}^T, \quad (14)$$

where k is much smaller than n . That is, a large matrix can be represented by three small matrices $U_{m \times k}$, $\Sigma_{k \times k}$, and $V_{k \times n}^T$. As shown in Fig. 8, matrix A can be approximated by three small matrices in the gray part now.

SVD can be seen as a concrete implementation of principal component analysis. It is a linear algebra technique and one of the most basic methods of processing complex data. A singular value decomposition of an $m \times n$ real matrix A is: Convert A to a diagonal matrix by constructing $P^T A Q$, where P is an $m \times m$ orthogonal matrix and Q is an $n \times n$ orthogonal matrix.

SVD is suitable for a wide range of eigenvector analysis problems. Chen proposed a dimension reduction clustering method based on the daily load curve of SVD [67], which solved the problems brought about by large amounts of historical load curves to data storage and calculation and shortened the running time besides increased accuracy of load curve clustering. Kang proposed an SVD-based feature

Fig. 8 Flow chart of SVD



extraction method [68], which is used to identify faults of induction motors and classify faults. The classification accuracy of the SVD-based method is proved well in both noise and non-noise environments.

SVD as a relatively simple algorithm is used in some machine learning algorithms. Its principle is simple and easy to implement. Its shortcoming is that the matrix interpretability is relatively weak, but this shortcoming does not affect its actual use.

Non-linear feature extraction algorithm

Based on kernel function optimized

The main idea of kernelization is to refer to the kernel function in other algorithms and to realize the transformation from the nonlinear problem in the original space to the linear problem in the feature space. However, the calculation is still performed in the original space. The use of the nucleation method is based on selecting a conditional function $K(x_i, x_j)$ with continuity and symmetry, and satisfying Mercer. Where x_i and x_j are sample points in the input space, which implements the mapping of the input space d_L to the d_H -dimensional feature space $\Phi : R^{d_L} \rightarrow H$, and there is

$$K(x_i, x_j) = \sum_{n=1}^{d_H} \Phi_n(x_i) \Phi_n(x_j). \tag{15}$$

The real purpose of mapping is that mapping the problems which are difficult in the input space to the feature space and solving them. The most frequently used kernel functions are linear polynomial functions, p -order polynomial functions, Gaussian radial basis function kernel functions, etc.

Such as kernel-based principal component analysis (KPCA) [69], its basic idea is to transform the input data x into the feature space F through a non-linear mapping $\Phi(x)$, and then use the linear PCA on F . The projection calculation of KPCA's characteristic values and vectors in F does not require the display form of the mapping $\Phi(x)$ for the operation of, and only need to calculate the mapping point product. In practical situations, the dot product can be calculated by the following kernel function.

$$K_{ij} = k(x_i, x_j) = (\Phi(x_i) \cdot \Phi(x_j)). \tag{16}$$

The nonlinearity of KPCA is achieved using the kernel transform to transform the input space into the Hilbert feature space. Therefore, it can be considered that the PCA calculation is performed in the input space and the Kernel PCA is completed in the feature space.

Kernel FDA [70] and kernel canonical correlation discriminant analysis [71] refer to the kernel function well and overcome the weaknesses that can only solve the linear problem. Although it is complicated in form, it is easy to solve the problem. The idea of nuclearization can be called a bridge between linear and nonlinear transformations so that some methods that can only solve linear problems can be applied to solve nonlinear problems.

Nonnegative matrix factorization

Nonnegative matrix factorization (NMF) [72], also known as non-negative matrix approximation or positive matrix factorization, is an unsupervised learning method. It decomposes a matrix into the result of the multiplication of two matrices, all of which are non-negative. In NMF, non-negative constraints prevent the basic functions from canceling each other out and produce a partial-based representation. The expression of NMF is: for a given non-negative matrix V , two non-negative matrices W and H satisfying $V = WH$ can be found, so a non-negative matrix can be decomposed into two non-negative matrix multiplication. The NMF algorithm presents a class of uncomplicated iterative methods for solving U and V . This method has a fast convergence rate and both sides' non-negative matrix storage space is small. NMF is faster than traditional processing algorithms.

Starting from non-negative initial conditions for W and H , iteration of these update rules for non-negative V finds an approximate factorization $V \approx WH$ by converging to a local maximum of the objective function:

$$F = \sum_{i=1}^n \sum_{\mu=1}^{\omega} [V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu}]. \tag{17}$$

The fidelity of the approximation enters the updates through the quotient $V_{i\mu}/(WH)_{i\mu}$.

NMF directly treats non-negative decomposition problems as constrained nonlinear programming problems. The NMF subspace requires that the base of the subspace and the projection coefficients of the sample on the subspace are non-negative, this constraint limits the data projected into the subspace to be the additive combination of the subspace base, and without the subtraction. Therefore, the obtained subspace formed by the non-negative basis of the data representation is non-orthogonal and partially unbounded, which makes its representation of the data more compact and less redundant, that is, has better squeeze, and is more conducive to the representation of data.

NMF has the following characteristics: (1) the result of decomposition does not contain negative values, has clear physical meaning and interpretability, and is very suitable for non-negative data processing; (2) being able to discover the underlying structural features of the data, and also reduce the dimensionality of the data features, saving storage and computing resources, which has obvious advantages in dealing with high-dimensional data; (3) the psychological and physiological structure of NMF is based on the fact that the human eye's perception of the whole is composed of the perception of the part, that is, the whole is a partial non-negative linear combination, so it has the characteristics of intelligent data description; (4) the non-negative limitation also results in a certain sparsity of the decomposition results. The relatively sparse representation can suppress the adverse effects of external changes (partial occlusion, illumination changes, image rotation) on feature extraction to a certain extent.

In face learning, the image database is viewed as a $n \times m$ matrix V , each column containing n non-negative pixel values of the m face images, then constructing an approximate decomposition form

$$V \approx WH \text{ or } V_{i\mu} = (WH)_{i\mu} = \sum_{a=1}^r W_{ia}H_{a\mu}. \quad (18)$$

The r column of W is called the base image. Each column of H is called a code and has a one-to-one correspondence with the faces in V . The coding consists of coefficients that represent the face with a linear combination of the base images. The dimensions of the matrix factors W and H are $n \times r$ and $r \times m$. The rank r of the factor classification is generally chosen such that $(n+m)r < nm$, and WH can be considered as a compressed form of the data in V . The NMF does not allow negative terms in the matrix factors W and H , these non-negativity constraints allow the combination of multiple base images to represent the face. However, only additive combinations are allowed because the non-zero elements of W and H are positive numbers and no subtraction occurs. For these reasons, non-negative constraints are

consistent with intuitive concepts that combine parts into a whole, which is how NMF learns a part-based representation.

In the text recognition, the NMF algorithm combines semantically related words to form semantic features, and uses the context to distinguish multiple meanings of the same word to handle the ambiguity of the "protagonist."

NMF has shown great success in the face and text recognition [73, 74], however, it is impossible to learn objects from different viewpoints or learn from highly expressed objects, because learning these complicated parts may require a fully layered model with multiple layers of hidden variables, which is not satisfied by single-layer NMF. Although non-negative constraints may contribute to the model-based learning of these models, we do not think that they are sufficient. In addition, NMF did not learn anything about the "grammar" relationship between the parts. The NMF assumes that hidden variables are non-negative but does not make further assumptions about their statistical dependencies.

In addition, the use of astronomical spectrometers is to analyze data [75]. The spectral data it provides are essentially non-negative. Therefore, an effective non-negative matrix factorization algorithm is presented in this paper. The algorithm has a novel smoothing constraint on the purpose of spatial target recognition and classification for separating spectral reflectance data. Liu describes the use of NMF as data analysis and interpretation tool in computational biology in various applications of computational biology [76], examples include molecular pattern discovery, class comparisons and predictions, cross-platform and cross-species analysis, gene function characterization, and biomedical informatics.

Based on information theory

In 1948, Shannon [77] proposed the concept of information entropy for the first time,

$$H(X) = \sum_{i=1}^n p(x_i)I(x_i) = - \sum_{i=1}^n p(x_i) \log(x_i), \quad (19)$$

where $p(x_i) \geq 0$, $\sum_{i=1}^n p(x_i) = 1$, $i = 1, 2, \dots, n$, $I(x_i)$ indicates self-information of x_i .

Entropy is the "uncertainty" of a random event or the measure of the amount of information [78]. It exists in the information field as the scientific theoretical foundation of modern information theory. The significance of introducing information theory into feature extraction is that it can more easily solve problems encountered in feature extraction. The improvement of methods based on information theory has also become a hot topic in recent research.

The entropy analysis uses the uncertainty of entropy to obtain useful features [79]. It does not need to know the specific size of the feature and its distribution details when

using it. Because the size of the entropy is inversely proportional to the degree of separation between its corresponding classes, so the classification using the feature with the smallest entropy (that is, the feature with the least certainty) is the best. The sorting of the characteristic parameters obtained by the entropy value analysis method can relatively accurately reflect the degree of separation of the recognition object on each characteristic parameter.

The information feature compression algorithm based on Symmetric Cross Entropy Criteria (SCEC) [80] proposed a new concept of Symmetric Cross Entropy (SCE) based on the definition of mutual entropy, and based on this, a SCEC for measuring two random variables which can measure the degree of difference between the two probability distributions is established. It can also be called a symmetric interaction entropy criterion. The smaller the SCEC is, the smaller the difference between the two sets of data is. When the SCEC is zero, the two sets of data are exactly the same. Therefore, for feature compression, given the number of compression dimensions d , we can find those d features which make the SCEC tends to have a maximum.

In addition, the feature extraction of stereo image points based on the information entropy of images, the use of fuzzy information entropy to determine fault feature parameters, etc. they are enough to illustrate the remarkable achievements of information entropy in feature extraction.

Based on wavelet transform and its improvement

Wavelet has become one of the most widely used mathematical tools in signal and numerical processing analysis [81]. Although the time of wavelet technology as a basic theory is still relatively short, wavelet analysis has shown great potential and applicability in many scientific and engineering fields, especially those phenomena where classical Fourier methods have been proved to be ineffective.

Wavelet Transform (WT) solves the contradictions between time resolution and frequency resolution well [82]. The window of the WT is an adjustable time–frequency window. The short window and the long window are applied to high frequency and low frequency. The observation and analysis of the signal use different scales and resolutions. After moderately discretization, WT can construct a normalized orthogonal system, which makes WT play a pivotal role in both theory and practice.

The time function $f(t)$ is expressed as the following wavelet progression:

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{j,k} \psi_{j,k}(t), \quad \psi_{j,k} = \psi(2^j t - k), \quad (20)$$

where $\psi_{j,k}$ is the wavelet function, j and k the wavelet coefficient, $d_{j,k} = \langle f, \tilde{\psi}_{j,k} \rangle$.

From the above formula, we can see that the indicators of wavelet coefficients include frequency index j and time index k . This means that the wavelet coefficients change according to the frequency change, and when the frequency index j is the same, the wavelet coefficients are different for different time k . When solving the wavelet coefficients at different frequency levels and at different times, only part of the information around the time is needed. This is because the compact supporting nature of the wavelet function makes it zero outside a certain interval. After the data signal is decomposed by wavelet, WT coefficients are obtained at multiple scales. These coefficients comprehensively describe the features contained in the signal. These coefficients can be used as feature subsets of the classification to achieve the purpose of dimensionality reduction.

Zhang demonstrated the feature extraction method based on WT from four aspects: WT-based modulus maxima feature, energy feature based on wavelet decomposition, entropy feature based on wavelet packet decomposition, and feature extraction method based on adaptive wavelet network, and use it for feature extraction of underwater acoustic signals [83]; He used adaptive harmonic transform to analyze an unstable voltage or current signal as a tool for feature extraction of power quality [84]. WT is used to analyze EKG and EEG signals and extract effective features to simplify classifiers are helpful for clinical diagnosis [85]. There have been many achievements in this field. Compared with traditional feature extraction methods, the classification accuracy of wavelet transform methods is significantly improved, and it indicates that wavelet analysis will play a wide and huge role in the future of feature extraction.

Projection pursuit

Projection Pursuit (PP) is a kind of algorithm for processing high-dimensional observation data [78, 79, 86, 87]. It converts high-dimensional data into low-dimensional subspace through projection, then finds the projection that can reflect the original features and analyzes the data. After data is projected into a lower dimensional space, robust variance can be maximized. This method is not constrained by the assumption of normal distribution, data visibility is improved, and variable disturbances which are not related to the data structure and characteristics or have little relationship with them can be eliminated.

Suppose x is a p -dimensional random vector, Y is a one-dimensional random variable (If Y is a m -dimensional random variable, it can be transformed into a one-dimensional random variable by appropriate methods such as principal component extraction). Based on the n observations (X_i, X_i)

($i = 1, 2, 3, \dots, n$) of (X, Y) , the estimated regression function is

$$f(x) = E(Y|X = x). \quad (21)$$

In the formula, $E(\cdot)$ is the expectation factor. The specific steps of the PP method are given below:

1. Choose an initial regression model, such as $f_0(x) = C$ or a unary nonlinear function.
2. Find a direction a , so that the current residual $r_i = y_i - f(x_i)$ ($i = 1, 2, \dots, n$) and projection $Z = a^\tau x$ have the largest possible regression dependence and obtain the smoothing function $g_a(Z)$.
3. Update the model to $f(x) = f_0(x) + g_a(a^\tau x)$. After m steps of iteration, the regression function can be expressed as $f(x) = f_0(x) + \sum g_j(a_j^\tau x)$ ($j = 1, 2, \dots, m$).

In the formula, $g_j(a_j^\tau x) = g_{a_j}(a_j^\tau x)$ is called the Ridge function. Here is mainly the choice of projection direction a and projection index. Given $a_j, g_j, j > m$ and look for a_m, g_m to make the objective function

$$Q = \sum_{j=1}^m r^2 m_j = \sum_{j=1}^m [r_{m-1,j} - g_m(a_m^\tau x_j)]^2 = \min \quad (22)$$

If a is given, the conditional expectation of g_m is $g(a^\tau x) = E(y|a^\tau x)$.

Obviously, the key here is the determination of a . The specific method is as follows:

Step 1 Give the projection direction a randomly.

Step 2 Sort $Z_i = a^\tau x_i$ ($i = 1, 2, \dots, n$) from small to large and record the sequence as $\{Z_i\}$. The corresponding residual sequence is recorded as $\{r_i\}$.

Step 3 For steady consideration, do a sliding median smoothing. Thereby isolated outliers $r_i = \text{Med}\{r_{i-1}, r_i, r_{i+1}\}$ can be moved.

Step 4 Perform a local linear fit on a total of $2k$ pairs of (Z_i, r_i) (k left neighbors and k right neighbors of $\{Z_i\}$), and find the residual equation sum, and count $\sigma_i^2 = [W]/2k$.

Step 5 Do a local linear fit to $\{r_i\}$, the bandwidth is determined by $\sigma_i'^2, \sigma_i''^2 = (\sigma_{i-k}^2 + \dots + \sigma_{i-1}^2 + \sigma_{i+1}^2 + \dots + \sigma_{i+k}^2)/2k$ and get $g_a(Z_i)$. This step uses a larger neighborhood bandwidth to overcome the larger local variation. (4) to (6) do not include the i data itself in the calculation to prevent over fitting.

Step 6 Use the sum of squared residuals of r_i and $g_a(Z_i)$ as the projection index $Q(a)$ of the a direction until the projection direction a which makes the projection index reaches a very small value is found. It should be pointed out here that the m in the formula is determined by the iterative method.

The principle is the absolute value of the relative increment of the projection index of the m step and the projection index of the $m - 1$ step is small than or equal to a certain threshold or stop iteration when the number of iterations times is greater than a predetermined maximum.

MT Gao uses a genetic algorithm to find the best projection direction [88]. It projects the original data structures and features and can be represented by the projection weight of the optimal projection direction. This method is more effective compared with K-means clustering when it is used in text clustering.

Manifold learning

Manifold is a term for general geometric objects, including curves and surfaces of various dimensions [89, 90]. Like general dimensionality reduction analysis, manifold learning is to re-set a set of data of high-dimensional space in low-dimensional space. The difference is that in manifold learning it is assumed that the processed data is sampled on a potential manifold, or there is a potential manifold for this set of data. The points on the manifold have no coordinates, so to represent these data points, we put the manifold into the ambient space and use the coordinates on the outer space to represent the points on the manifold.

Manifold learning aims to discover the inherent regularity of the distribution of high-dimensional data sets. The basic idea is that the points in the high-dimensional observation space is transformed into a manifold in the observation space by a collective effect of a few independent variables. If the manifold of the observation space is effectively expanded or the intrinsic main variables are found, the data set can be dimension-reduced.

We can use mathematical language to describe it: manifold learning (ML) is derived from differential geometry. Its definition is: Let M be a Hausdorff topological space, if every point P on M has an open neighborhood $N \subset M$ and can make an open subset of N and n -dimensional Euclidean spaces homeomorphic R^n , then M can be called an n -dimensional manifold. It maps the raw data to a new coordinate system, making the classification problem simpler and improving performance by learning better background models. However, due to the need to calculate the distance metric between all data points, the computational cost is too high [91].

Now the process of dimension reduction for manifold learning can be summarized formally: assuming that the data is a low-dimensional manifold uniformly sampled in a high-dimensional Euclidean space, manifold learning is to recover low-dimensional manifold structures from high-dimensional sampled data, that is, to find low-dimensional manifolds in high-dimensional space, and find the corresponding embedding map to achieve dimension reduction or data visualization. It searches the essence of things from

the observed phenomena and finds the inherent law of generating data. Representative algorithms for manifold learning include Isometric map (ISOMAP), Locally Linear Embedding (LLE), Locality Preserving Projection (LPP).

Isometric map

ISOMAP believes that when the data set has an embedded manifold structure [92, 93], the corresponding description of the observation space data set in the low-dimensional structure can be obtained according to the distance map. ISOMAP describes the relationship between points by geodesic distance. In the global sense, the distance between the points is obtained by finding the shortest path of each point in the sense of the figure, and then the low-dimensional embedded coordinates are obtained by the classical MDS algorithm. Therefore, ISOMAP can be considered as a variant of the MDS algorithm.

Prerequisites: The low-dimensional manifold where the high-dimensional data are located is equidistant from a subset of the Euclidean space; the subset of the Euclidean space that is equidistant from the manifold in which the data are located is a convex set, such as in Fig. 9.

The main steps of ISOMAP are as follows:

Step 1 Construct local neighborhoods. First for the data set $X = \{x_1, x_2, \dots, x_n\}$, calculating the Euclidean distance $d_x(x_i, x_j)$ of any two sample vectors x_i and x_j . Comparing each point with all of the others. We think they are adjacent when the distance between two points is less than the fixed radius ε (or i is the K -neighbor of j), then connect them. The length of the side is $d_x(x_i, x_j)$. And the neighborhood graph G is obtained.

Step 2 Calculate the shortest distance. In graph G , the shortest distance between any two sample vectors x_i and x_j is $d_G(x_i, x_j)$. If there is a connection between x_i and x_j , then the initial value of $d_G(x_i, x_j)$ is $d_x(x_i, x_j)$, otherwise $d_G(x_i, x_j) = \infty$. For $k = 1, 2, \dots, n$, there is

$$d_G(x_i, x_j) = \min\{d_G(x_i, x_j), d_G(x_i, x_k) + d_G(x_k, x_j)\} \tag{23}$$

so we can get the matrix $D_G = \{d_G(x_i, x_j)\}$. It is composed of the shortest path of all pairs of points in graph G .

Step 3 Construct a d -dimensional embedding. Constructing a d -dimensional which maintains the feature geometry embedded in the space Y with the MDS method.

$$\tau(D_G) = -\frac{H \times (D_G)^2 \times H}{2} \tag{24}$$

H is the unit matrix and is in the same order with D_G . Perform eigen decomposition on $\tau(D_G)$, taking the largest first d eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_d$ and corresponding eigen-

vectors V_1, V_2, \dots, V_d . Let V_p^i be the i th component of the p th feature vector, and the corresponding low-dimensional data represents as $y_i = \lambda_p^{1/2} V_p^i$.

Overall, when ISOMAP algorithm is used to reduce dimension, the advantages are as follows: suitable for learning internal flat low-dimensional manifolds. ISOMAP combines the main features of linear algorithms (such as PCA and MDS)—computational effectiveness, global optimization, and progressive convergence. This method of replacing the traditional Euclidean distance with the geodesic distance can more effectively express the data of the high-dimensional space in the low-dimensional space and reduce the data information lost after the dimension reduction.

However, there are some shortcomings of the ISOMAP algorithm: not suitable for learning manifolds with large intrinsic curvature. Under noise interference, ISOMAP will be unstable for visualization, and a large neighborhood will cause short-circuit phenomenon, that is, there will be obvious confounding after point projection of different neighborhoods in low-dimensional manifolds. Selecting a smaller neighborhood ensures the stability of the overall structure but it results in a large number of "holes" in the low-dimensional projection results, or makes the graph reconstructed by the shortest path algorithm not connected. The determination of the dimensionality reduction dimension is usually carried out under the condition that the essential dimension is unknown, and the residual curve is observed through multiple experiments. The ISOMAP algorithm calculates the shortest distance between two points on the graph using the Dijkstra algorithm, but it is still slow to implement.

According to the characteristics of the ISOMAP algorithm, we should pay attention to the following problems in application: when the subset of Euclidean space equidistant from the high-dimensional manifold is not convex, that is, when there is a "void" in the high-dimensional space, it is very likely to occur a bending anomaly when calculating the distance between any sample points on a high-dimensional observation space, and this would affect the representation of low-dimensional embedding results; The isometric feature mapping algorithm may be unstable in the data topology space. Because if the selected neighborhood is too small, the neighbor graph will not be connected. If the selected neighborhood is too large, it may cause a short circuit; When using the ISOMAP algorithm to recover the geometry of a nonlinear manifold, the calculation time required is relatively large, which is mainly spent calculating the shortest path between sample points.

Locally linear embedding

Local linear embedding (LLE) creates a reconstructive relationship between all the beam axis sample points and its neighboring sample points, so that some key features in

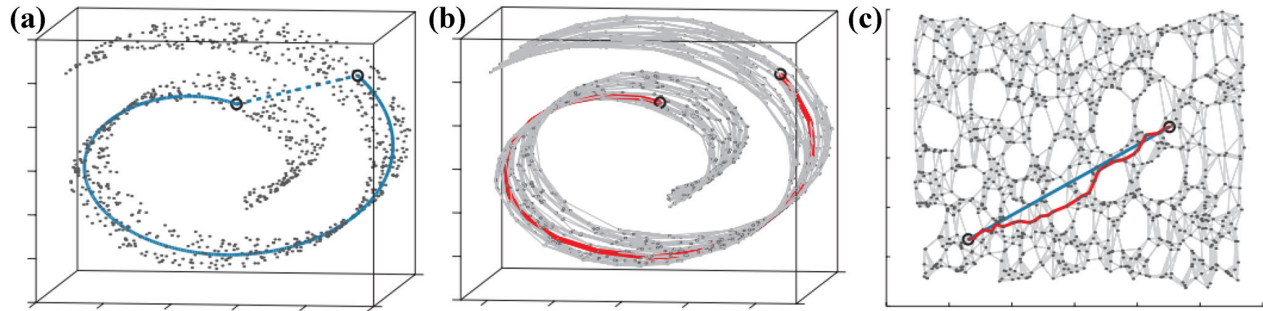


Fig. 9 Schematic diagram of ISOMAP algorithm [94]. Note: **A**: Euclidean distance and Geodesic distance; **B**: diagrams of adjacent points; **C**: reduced to two-dimensional data. **A** Euclidean distance (length of dash line) of the high-dimensional input space and Geodesic distance from low-dimensional manifold (length of solid curve); **B** the true geodesic path in the geodesic map that is effectively calculated through neighborhood graph G and this path is taken as the shortest path. **C** the two-dimensional embedding recovered by ISOMAP, where blue lines in the embedding are simpler than the corresponding graph paths (red)

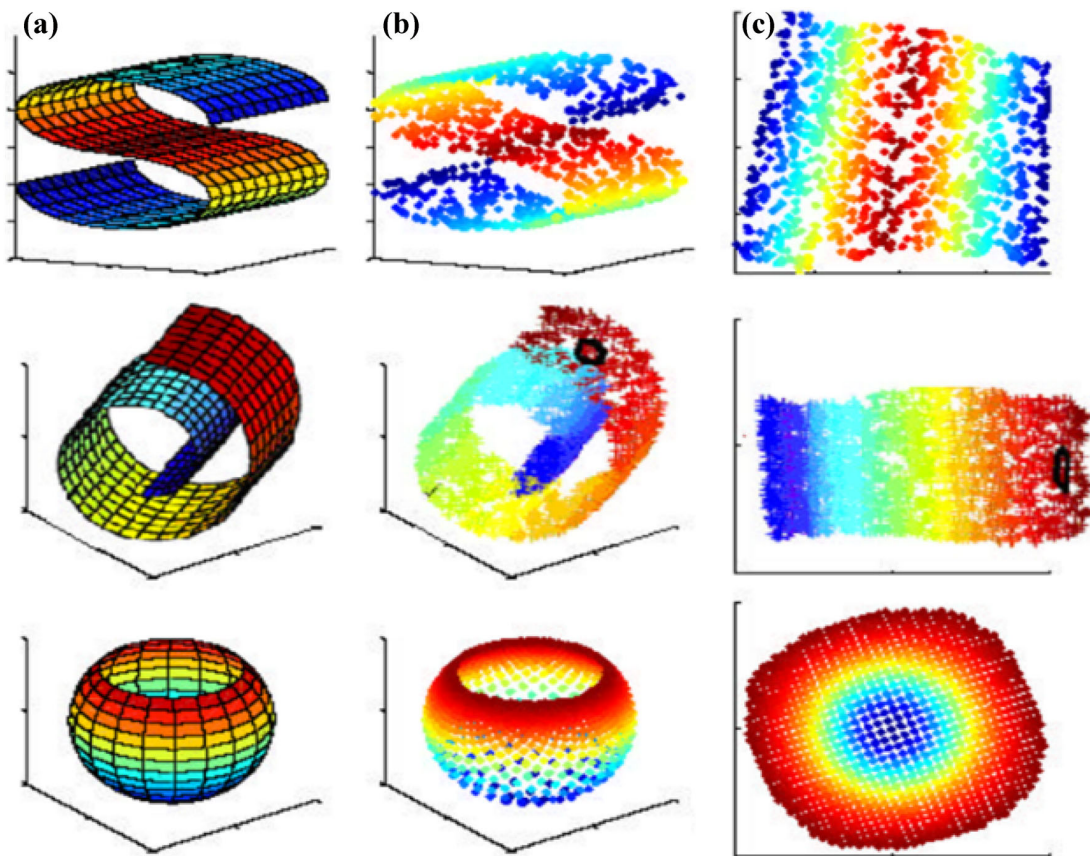


Fig. 10 Schematic diagram of LLE algorithm [99]. **b** The sample point (3D) extracted from **a**, where red and blue can be considered as two types of data. The data (**b**) are mapped to the 2D space (**c**) by the nonlinear dimensionality reduction algorithm LLE. The original data manifold pattern is still retained in 2D space. It can be seen from the color in (**c**) that the data processed by the LLE algorithm can maintain the neighborhood characteristics of the original data well

the high-dimensional measurement space can be preserved during the data reduction [95–98]. The basic idea is to approximate all data points by their adjacent weighted linear

combinations and the linear approximation of all data points is best preserved (Fig. 10).

The LLE algorithm can be attributed to three steps:

Step 1 Find the k nearest neighbors for each sample point and specify the k nearest sample points to the nearest k sample points. k is a predetermined value.

Step 2 The local reconstruction weight matrix for this sample point is calculated from the neighbors of each sample point. Here an error function is defined as follows:

$$\min \varepsilon(W) = \sum_{i=1}^N \left| x_i - \sum_{j=1}^k w_j^i x_{ij} \right|^2, \tag{25}$$

where $x_{ij} (j = 1, 2, \dots, k)$ are the k nearest neighbors of x_i , w_j^i is the weight between x_i and x_{ij} , and need to meet the conditions: $\sum_{i=1}^k w_j^i = 1$. Finding the W matrix here requires constructing a local covariance matrix Q^i :

$$Q_{jm}^i = (x_i - x_{ij})^T (x_i - x_{im}). \tag{26}$$

And then, the local optimized reconstruction weight matrix can be obtained:

$$w_j^i = \frac{\sum_{m=1}^k (Q^i)_{jm}^{-1}}{\sum_{p=1}^k \sum_{q=1}^k (Q^i)_{pq}^{-1}}. \tag{27}$$

Step 3 The output value of the sample point is calculated from the local reconstruction weight matrix of the sample point and its neighbors, and all the sample points are mapped into the low-dimensional space. The mapping conditions are satisfied as follows:

$$\min \varepsilon(Y) = \sum_{i=1}^N \left| y_i - \sum_{j=1}^k w_j^i y_{ij} \right|^2, \tag{28}$$

where $\varepsilon(Y)$ is the loss function value, y_i is the output vector of x_i , $y_{ij} (j = 1, 2, \dots, k)$ are the k nearest neighbors of y_i , and two conditions must be satisfied $\sum_{i=1}^N y_i = 0$, $\frac{1}{N} \sum_{i=1}^N y_i y_i^T = I$, where I is a $m \times m$ identity matrix, here $w_j^i (i = 1, 2, \dots, N)$ can be stored in $N \times N$ sparse matrix W . While x_j is the nearest neighbor of x_i , $W_{i,j} = w_j^i$, otherwise, $W_{i,j} = 0$. Then the loss function can be rewritten as:

$$\min \varepsilon(Y) = \sum_{i=1}^N \sum_{j=1}^N M_{i,j} y_i^T y_j \tag{29}$$

where M is a $N \times N$ symmetric matrix and its expression is $M = (I - W)^T (I - W)$.

To minimize the loss function value, taking the eigenvector corresponding to the smallest m non-zero eigenvalues with Y as M . In the process, the eigenvalues of M are arranged from small to large, and the first eigenvalue is almost close to zero, then the first eigenvalue is discarded. Usually

take the feature vector corresponding to the feature value in $2 \sim m + 1$ as the output result.

As a classic nonlinear dimensionality reduction method, LLE is becoming more and more attractive to researchers because of its ability to process large amounts of high-dimensional data and non-iterative embedding methods. It is more effective than PCA to reduce the data dimension and is also simpler to calculate: (1) there are only two parameters to set; (2) optimization does not involve local minima; (3) maintaining the local geometry of high-dimensional data in the embedded space; (4) the embedded space has a single global coordinate system.

However, some inherent deficiencies, such as its sensitivity to noise, unavoidable conditional features, lack of processing methods, etc., limit its application. Therefore, some LLE extension methods have been proposed, such as the supervised LLE algorithm, SLLE [100, 101]. The traditional LLE algorithm finds k nearest neighbors based on the Euclidean distance between sample points. While SLLE is processing this step, it increases the category information of sample points. The remaining steps of SLLE are the same as the LLE algorithm. LLE hopes to maintain local geometric properties when it comes to dimensionality reduction. There is a certain relationship with the number of sample points when finding neighboring points and reducing values according to neighboring points. If we want to increase the speed of operation, we must reduce the number of sample points. According to this disadvantage, LLE based on clustering and improved distance are proposed [102]. The clustering-based method greatly reduces the time needed for operations; the improved distance LLE method can achieve satisfactory results when the number of neighbors is small. At the same time, it can also select the number of fuzzy neighbors. Compared with the unmodified LLE method, the time for dimension reduction has been greatly reduced, and the selection range of parameter K has also been expanded. The LLE method and its extension method have been widely used in machine learning and play an increasingly crucial role in the data processing.

Locality preserving projection

Locally preserved projection (LPP) is a recently proposed method of dimensionality reduction that attempts to capture the manifold structure of data [103, 104]. In LPP, neighborhood information is stored in a graph and the base vector is found using the Laplacian concept. A weighing function is used to assign weights for the edges of the graph. If the data points are mapped very far, then this function will be severely penalized, therefore it will pay more attention to the nearest neighbors [105–107].

Given a set X , the general problem of linear dimensionality reduction methods is to find a transformation matrix A to

map the original data set X to the low-dimensional Y . $G(V, E)$ is a weighted graph that connects adjacent vertices. The LLP maps weighted graphs G to a lower dimension. In this graph, the connection points are consistent with the previous positions. If Y is such a map, one criterion for getting a good mapping is to minimize the following objective function:

$$\sum_{ij} (y_i - y_j)^2 W_{ij}. \quad (30)$$

W_{ij} is a similarity matrix. If the adjacent vertices x_i and y_i are mapped far, this is obviously not reasonable. Therefore, minimizing the above function is equivalent to maintaining the similarity between x_i and x_j , and y_i and y_j . Assuming that a is a transformation vector for A , we can get:

$$\frac{1}{2} \sum_{ij} (y_i - y_j)^2 W_{ij} = \frac{1}{2} \sum_{ij} (a^T x_i - a^T x_j)^2 W_{ij}, \quad (31)$$

limitation factor:

$$a^T XDX^T a = 1. \quad (32)$$

In the end, it can be transformed into a generalized eigenvalue problem:

$$XLX^T a = \lambda XDX^T a, \quad (33)$$

where λ is the smallest eigenvalue solution of the above equation and a is the corresponding eigenvector.

LPP can maintain the local structure of the original data comparatively well, it wants to bring two points which are near in the original space closer to each other after dimension reduction. This is good for neighboring points of a class. But for neighboring points of different classes, if two classes are relatively close to each other or partially overlapped, due to the characteristics of local retention, and without considering category information, the two different classes will be projected to one place and will lead to unwanted results [108].

Therefore, some LPP variants [109] were proposed. For example, the extended version of LPP (ELPP) improves resilience and resolves ambiguity in overlapping areas; in the supervised variant ESLPP-MD, the concept of internal and interclass distances is used to obtain better type discrimination. These variants make the projection of data more robust and achieve better dimensionality reduction.

Brief summary

In a general sense, a manifold in the ML method is a topological space, which can detect inherent low-dimensional structures in nonlinear high-dimensional data. The research contents of ML include preserving or highlighting the dimensionality reduction of the limited data set of the specified

features in the original high-dimensional data; the density estimation problem of high-dimensional finite sample points that meets certain distributions; the latent variable model creation of high-dimensional observation data with few partial implicit interferences. Although ML is a fundamental research direction, it has a large amount of applicable space. In recent years, ML has become a hot issue [110], and it has been widely used in data analysis, pattern recognition and machine learning, attracting more and more attention.

How to automatically judge whether the nonlinearity of the data set is caused by the intrinsic curvature or the mapping model? How to deal with the intermittent manifold and the non-intermittent manifold or the variable dimension manifold? There is no good solution at present. This is also a question that needs further study in manifold learning.

As an intermediate process of data processing, manifold learning still has the following problems: (1) manifold learning algorithm has high computational complexity; (2) manifold learning algorithm adopts local neighborhood idea when restoring intrinsic invariant, the algorithm's stability is related to the neighborhood selection, resulting in weak classification ability; (3) how to adaptively determine the parameters needed in the manifold learning algorithm, rather than empirical or artificial settings; (4) incremental learning, how to modify the mapping relationship according to the newly input samples without recalculation; (5) in the actual high-dimensional sampling data, noise is often present due to various factors, so that the distortion and deformation of the original data structure appear after mapping to the low-dimensional space; (6) how to find the mapping relationship between the two spaces, including linear and nonlinear mapping relationships to reconstruct manifolds is a key issue that needs to be addressed for both supervised and semi-supervised learning.

Method of algorithm fusion

For each method and the improved algorithm, the dimensionality reduction problem is solved in some senses, but each method has its own disadvantages while exerting its own advantages. The combination of various methods can compress features better and provide better feature information for recognition, thereby improving the accuracy of recognition.

Information theory based PCA feature compression algorithm [111]. According to the definition of the information function, combined with the inherently unique nature of the eigenvalues, the generalized information function is defined and used in the feature compression of PCA, and then the information rate and the cumulative information rate appear, thus creating an information-based PCA feature compression algorithm. The algorithm can better describe the degree of information compression. Compared with the principal com-

ponents obtained by PCA, it covers more information of the original features. This algorithm combines the advantages of both information theory and PCA.

Information feature analyzed by orthogonal transform [112]. Under the minimum mean square error criterion, an optimization theory model is created, and the results are calculated and given, which proves that the K–L transform is the optimal orthogonal transform of feature selection. According to the definition and connotation of entropy in information theory, the second representation entropy is proposed. The new entropy function is used to analyze the information features under the transformation and the information payload of the eigenvalues after transformation.

Zuo combined three methods of constrained global deformation (CGD), component texture fitting (CTF) and component feature refinement (CFR) for facial expression recognition [113]. It has been verified by practice that the feature extraction of the three methods uplink is better than the independent extraction. Bidirectional feature data compression method based on principal component analysis and immune clustering eliminates the correlation between the characteristic parameters and has more efficient execution efficiency and wider adaptability. Combining iris technology with multi-dimensional scale analysis for feature extraction can improve the accuracy of iris recognition [114].

These are all good examples of algorithm fusion. Different theories merge with each other to make up for the defects of the single algorithm and provide an effective method for feature extraction.

Dimensionality reduction for small sample

The existing dimension reduction algorithms, mainly for large sample data, mostly consider the characteristic variables of the sample directly, such as PCA, FA, nonnegative matrix factorization (NMF), and so on. Their dimension reduction effect is relatively ideal. However, if encountering small samples, the traditional dimension reduction algorithm is difficult to obtain the ideal low-dimensional data, because the feature dimension is larger than the number of samples and the covariance matrix is singular.

Characteristic of small sample

When the sample size is smaller than the data dimensionality, which is the case for many high-dimensional and low sample size data, this sample is named the small sample [32, 115]. In the general case, the small sample is a relative concept including two cases as follows.

One class refers to that the number of training samples is less than its pattern feature dimension. The weakness of this type of small sample is that the parameters of feature

extraction and classifier algorithm cannot be estimated. For example, the sample covariance matrix within the class is singular, and the optimal recognition feature is difficult to be extracted; in addition, each class covariance matrix is singular, so the quadratic discriminate analysis method cannot be used directly.

Another class refers to that the number of training samples is more than its pattern feature dimension, but the difference is not big. Although the number of training samples can meet the non-singularity requirements of the covariance matrix, the fewer training samples will cause the instability of the inverse matrix of the covariance matrix.

Even so, the small sample has its own advantages. For example, due to the fewer training samples, the learning time of feature extraction and classifier algorithm is short, so the running time of the algorithm appropriately will be saved to improve the classification precision of samples. In the practical application field, small sample size problems are widespread. The main characteristic of a small sample is a higher feature dimension and fewer numbers of samples, and even the feature dimension is larger than the number of the samples which will cause the traditional learning methods too difficult to deal with small samples and easily lead to the following problems.

(1) It is easy to produce lots of redundant information and irrelevant information due to the rapid increase of the feature dimension. (2) Due to the lack of sample size, it easily leads to inadequate training or over-fitting phenomenon and drops the classification ability of the classifier. (3) When the training sample set has been changed, it is easy to cause that feature extraction and classification learning algorithm will be unstable, that is, the generalization ability of the model will be low.

Partial least squares algorithm

The partial least squares algorithm is the characteristic development of ordinary least squares (OLS). Its basic idea is that while the independent variable matrix X is compressed, the correlation of the dependent variable matrix Y is given consideration too. Suppose there are n independent variables $\{x_1, x_2, \dots, x_n\}$, p dependent variables $\{y_1, y_2, \dots, y_p\}$. After preprocessing, matrix X is decomposed into

$$X = TP^T + E. \quad (34)$$

In the formula, T is the score matrix, P is the load matrix, and E is the residual error matrix. Matrix multiplication of TP^T can be expressed as the sum products of score vector t_i (the i th column of matrix T) and load vector p_i (the i th

column of matrix P), and then the above formula can be written as

$$X = \sum_{i=1}^n t_i p_i^T + E_i = 1, 2, \dots, n. \quad (35)$$

Similarly,

$$Y = \sum_{j=1}^p u_j q_j^T + F_j = 1, 2, \dots, r. \quad (36)$$

PLS analysis is the score t and u separately extracted from corresponding X and Y , and they are the linear combination of independent variables and dependent variables. And both scores satisfy the maximum load of variation information of independent variables and dependent variables; the covariance between them is the largest. The establishment of the regression equation is:

$$u_j = b_k t_i. \quad (37)$$

PLS does iterative calculations using each information of other for each dimension, and each iteration adjusts t_i , u_j for the second-round extraction according to residual information of X , Y . Until the absolute value of the residual matrix element is approximate to zero, the precision satisfies the requirements, and then the algorithm stops. In the process of iteration, t_i , u_j can maximize the expression of the variance of X and Y at the same time.

PLS regression does not need to use all the components to establish the regression equation. It can get better regression equation only by selecting the front m components ($0 \leq m \leq n$). Generally, K -fold cross-validation method is used to calculate the prediction residual sum of squares, further to determine the number of components extracted, reaching the purpose of dimension reduction.

In the complicated multivariable system, PLS regression adopts a new method on the process of information recognizing and selecting. It does not judge that the variable is retained or abandoned one by one, but is obtained using the thought of information decomposition. PLS recombines the information of the independent variable system, it generally extracts the comprehensive variables, which has the best explanatory for system and eliminates the interference of overlapping information or non-explanatory information. So in the system modeling, PLS can overcome the adverse effects of variables' correlation and gets more accurate and reliable results.

From steps of PLS regression, the modeling strategy is built on the basis of information decomposition and extraction. PLS successively extract the comprehensive components t_1, t_2, \dots, t_m ($0 \leq m \leq n$) form the multivariable

system x_1, x_2, \dots, x_n . This is equivalent to the information recombination and extraction for x_1, x_2, \dots, x_n . The obtained comprehensive components have the strongest ability of explanation for Y and the generalization for X . Meanwhile, the non-explanatory information for Y is eliminated naturally. The convergence rate of the PLS regression algorithm is fast, which will be easy to get a fully satisfactory result.

Under normal circumstances, when analyzing the small sample, sometimes the phenomenon of overfitting might happen in the modeling process. When using PLS to reduce the feature dimension of small sample data, it mainly from the following two aspects to prevent the overfitting phenomenon. First, some correlated variables were transformed into uncorrelated variables by orthogonal transformation, namely, it can eliminate the correlation between variables and ensure the uncorrelation between principal components. Secondly, in the PLS regression modeling process, the cross-validation method is used to evaluate the precision of the model.

Least absolute shrinkage and selection operator

The least absolute shrinkage and selection operator (Lasso) is also a compression estimation method with reducing feature quantity as the core [116]. Its idea is to reduce the feature dimension by constructing a penalty function to compress the feature coefficients and make some regression coefficients become 0. It is a regularization method based on the L1 penalty term. Mathematically, the multiple linear regression model can be expressed as

$$Y = \alpha_0 + \alpha_1 x_1 + \dots + \alpha_i x_i + \alpha_v x_v + \varepsilon,$$

where α_i represents the regression coefficient, ε denotes the bias. The penalty function penalizes the L1-norm of the regression coefficient and requires the sum of the absolute values of all regression coefficients to be less than or equal to the penalty parameter λ ($\lambda \geq 0$). Assuming that there are v variables and s samples, the penalty function can be described as

$$\sum_{k=1}^v |\alpha_k| \leq \lambda. \quad (38)$$

Therefore, the lasso criterion can be described as

$$\text{lasso} = \sum_{i=1}^s (y_i - \alpha X)^2 + n\lambda \sum_{k=1}^v |\alpha_k|. \quad (39)$$

The optimization objective function of Lasso can be described as

$$(\hat{\alpha})^{\text{LASSO}} = \arg \min \sum_{i=1}^s \left(y_i - \sum_{k=1}^v \alpha_k x_{ik} \right)^2, \quad (40)$$

$$\text{s.t. } \sum_{k=1}^v |\alpha_k| \leq \lambda. \quad (41)$$

Based on the above objective function, the regression coefficient is compressed by adjusting the penalty parameter λ . It should be noted that the size of the parameter λ is related to the convergence of the model.

If the parameter of λ is too small, there will be too many noise variables. On the contrary, if the value of λ is too large, the regression coefficients of multiple independent variables are compressed to 0 and it will contain too few arguments. K -fold cross-validation method [117] is applied to select the parameters to select a reasonable parameter.

Lasso is a convex optimization method based on L1-norm to feature dimensionality reduction [118]. This method has the advantages of sparse features, preserving subset shrinkage and simplifying data, so it has a good effect on feature extraction of small samples with high-dimensional data. At present, it has been applied in the fields of gene analysis and image recognition. Lasso was adopted to assess gene effects in genome-wide association studies (GWAS) of brain images [119].

Other methods

Linear discriminant analysis (LDA) has been used extensively in dimension reduction algorithms [120], but it is constrained by the small sample problem. For this reason, a boundary-based feature extraction algorithm suitable for small samples is proposed [121]. The algorithm redefines the classification limits, and also takes into account the intra-class and interclass dispersion degrees defined by LDA, and the difference in variances of various classes. It can not only maximize the boundary to get the optimal projection vector, but also can prevent the small sample problem caused by the singularity of the intra-class dispersion matrix.

Embedding learning means that each sample is embedded into a lower-dimensional space, which makes it easier to distinguish different classes. Embedding learning essentially reduces the feature dimension by reducing the search space of the model. Metric learning [122] and learn-to-measure [123] from meta-learning reduce the feature dimension by reducing the search space. As an example of embedding learning, KernelBoost [124] is a method of metric learning that learns the pairwise distance in a kernel function manner with the help of a boosting algorithm. Assuming that (x_{i1}, x_{i2}, y_i) is an equivalence constraint, y_i is defined as 1 if (x_{i1}, x_{i2})

belong to the same categories, while y_i is defined as -1 if (x_{i1}, x_{i2}) belong to the different categories. The kernel function composed of several weak kernel functions is learned, $K(x_1, x_2) = \sum_{t=1}^T \alpha_t K_t(x_1, x_2)$. A Gaussian Mixture Model (GMM) of the data is learned by each weak kernel $K_t(\cdot, \cdot)$, where $K_t(x_i, x_j)$ refers to the probability that both two point x_i and x_j belong to the same Gaussian component in the t th GMM. The loss function of this method can be optimized, $l = \sum_{i,j} \exp(-y_{ij} K(x_i, x_j))$.

In addition, hybrid methods for dimension reduction are also applied to feature dimensionality reduction of the small sample. For microarrays, as a typical application of the small sample, ICA is used to reduce the size of microarray data and ABC is utilized to optimize the reduced feature vectors. To select informative genes based on a Naïve Bayes (NB) algorithm, a new hybrid search technique is proposed by ICA and ABC [125]. The other new combination of feature selection/extraction approach is designed for Artificial Neural Networks (ANNs) classification of high-dimensional microarray data with the help of ICA and ABC by Aziz et al. [126].

Dimensionality reduction by deep learning

The traditional shallow learning algorithm plays an important role in machine learning. The rapid development of Internet technology puts forward higher requirements for the intelligent analysis and prediction of big data. In the era of big data, a more powerful and complex deep learning model structure can more effectively and accurately capture the characteristics of massive data, and better restore the nature of the characterization data, so as to make accurate predictions for the future. The deep learning approach seeks to find the internal structure of the data and discover the true relationship between variables. However, the depth learning network obtained by deep learning is relatively difficult to train, and this has become the embarrassment of its development. In 2006, Hinton pointed out that the difficulty of deep learning network training can be solved by “layer-by-layer initialization” [127, 128]. This makes deep learning rejuvenate in the field of machine learning, setting off an upsurge of deep learning in academia and industry. The superior features are reflected in speech recognition and medical research.

Principle

The concept of deep learning originated from the study of artificial neural networks. Hinton and his student pointed out that deep learning is a learning process in which a sample gets a deep network structure with multiple levels after training and learning. It stimulates the human brain’s nervous system to build a multi-level complex model and extracts the features

in the original data from the bottom layer to the high layer, thereby establishes the mapping from the underlying signal to the high-level semantics well. The goal of deep learning is to establish a multi-layer connection structure. In the processing of image texts, the data features are processed hierarchically by several transformation stages, finally, the feature representation of the sample in the original space is transformed into a new feature space. Then combining low-level features to form more abstract high-level representations and attribute categories or features, hierarchical feature representation is obtained, which is more conducive to the classification or visualization of features [129–131].

Deep learning is relative to shallow-learning methods such as support vector machines and maximum entropy. Shallow learning relies on artificial experience to extract sample features. After learning, it is a single-layer feature without hierarchy. Compared with shallow learning, deep learning differs in (1) the depth of the model structure is emphasized, and the deep learning model generally has more than five or even ten layers of hidden layer nodes; (2) emphasize the importance of feature learning, that is, change the feature representation of the original sample data layer by layer to a new feature space, making the prediction or classification less difficult [132]. At present, deep learning has become a new research direction in the field of machine learning. It can be widely used because it can find more valuable information in big data with less manpower and time consumption. The common deep learning methods of feature dimensionality reduction are mainly divided into the supervised-based, unsupervised-based method and semi-supervised methods according to the number of labeled samples.

Supervised-based methods

Supervised feature extraction methods can be divided into two categories: based local region and based global region. The two main methods of local region feature dimensionality reduction are convolution and pooling. The typical methods of global feature extraction are recurrent neural networks and Transformer.

Based local region

Convolutional neural networks (CNNs) are typically deep learning methods based on local region feature extraction. A convolution neural network is composed of stacked convolution layers and pooling layers [133–135], where both convolution layers and pooling layers can be used to reduce the feature dimension.

For a convolution layer, suppose that there are a set of kernels $K = \{k_1, k_2, \dots, k_m\}$ and additional biases $B = \{b_1, b_2, \dots, b_m\}$, a set of the new feature map $X_l = \{X_1, X_2, \dots, X_m\}$ can be calculated and a non-line trans-

form $\sigma(\cdot)$ is applied to all features in an element-wise. This process is repeated for each convolution layer l . Mathematically, a spatial convolution can be described as

$$X_m^l = \sigma \left(W_m^{l-1} \cdot X^{l-1} + b_m^{l-1} \right). \quad (42)$$

The size of the new feature maps can be expressed as

$$H^l = \frac{H^{l-1} - h^{l-1} + 2 \cdot P^{l-1}}{S^{l-1}} + 1, \quad (43)$$

$$W^l = \frac{W^{l-1} - w^{l-1} + 2 \cdot P^{l-1}}{S^{l-1}} + 1, \quad (44)$$

where the size of the new feature map is $H^l \times W^l$, P^{l-1} represents padding size of $(l-1)$ th layer; S^{l-1} is set to stride size of $(l-1)$ th layer, h^{l-1} and w^{l-1} are height and width of the convolution kernel.

For pooling layers, its main task is to reduce spatially dimensions of feature maps. Pooling operations can extract features with more high representation ability and speed up the convergence. Three types are applied to pooling operation: max pooling, L2-norm pooling, and average pooling. This typically takes a pooling kernel with a stride of the same length. A new feature map can be obtained via a pooling layer. The size of the new feature map can be expressed as follows:

$$H^l = \frac{H^{l-1} - h_2^{l-1}}{S^{l-1}} + 1, \quad (45)$$

$$W^l = \frac{W^{l-1} - w_2^{l-1}}{S^{l-1}} + 1, \quad (46)$$

where h_2^{l-1} and w_2^{l-1} represent height and width of $(l-1)$ th pooling kernel. Taking max-pooling as an example, the feature map outputs the maximum value in each sub-region after max pooling.

1. ResNet

ResNet is a common network that uses stacking convolution and pooling layers to extract high-level semantic information [136]. When the samples are input into the network, the output of the previous layer will be used as the input of the next layer, and the feature map with higher representation ability has been extracted. However, with the increase of the depth of the neural network, there will be the problem of gradient explosion or gradient disappearance, which makes the network training slow or the loss decline is not obvious. To alleviate the above problems, skip structure is introduced into the convolutional neural network. Skip structure is to skip one or more layers and transfer feature information to the deeper layer of the neural network. This architecture enables us to train deeper networks while maintaining good performance.

As an important backbone of feature extraction, this network has been widely used in many fields, such as computer vision, EEG detection, and so on. Based on the target detection algorithm of two-stage Faster R-CNN [137], the ResNet network with a depth of 50 is adopted to reduce the feature dimension, extract high-level semantic information and generate pyramid features using a feature pyramid network (FPN), and then input the pyramid features into the detector for target detection. Mask R-CNN [138] also takes ResNet as a basic backbone to extract features and uses the FPN to generate multi-scale feature maps for object detection and image segmentation. The feature extraction networks represented by ResNet can represent high-level semantic information with strong representative ability, especially the information mapped to large-scale targets. Although the feature extraction network based on spatial convolution and pooling can extract high-level semantic information, only the key information is extracted from the convolution kernel or pooling kernel. Therefore, the context information surrounding the convolution and pooling kernel context information is ignored or discarded, which will lead to the loss of some feature information.

2. Dilated convolution

Dilated convolution uses a convolution kernel with a dilation convolution factor to enlarge receptive fields and extract features [139]. Suppose that $F : Z^2 \rightarrow R$ is a discrete function, $\Omega_r = [-r, r]^2 \cap Z^2$ and $k : \Omega_r \rightarrow R$ discrete filter of size $(2r + 1)^2$. The convolution operator $*$ can be calculated as

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t). \quad (47)$$

Suppose that l is a dilated factor, the dilated convolution $*_l$ can be described as:

$$(F *_l k)(p) = \sum_{s+l=t=p} F(s)k(t). \quad (48)$$

Further, the receptive field corresponding to dilated convolution increases exponentially without losing the feature resolution.

Dilated convolution can supplement the feature information extracted only from local regions, and fully consider the relationship between foreground and background. For optimizing small objects, Cui et al. proposed a context-aware block to merge multi-level contextual information with the help of stacked dilated convolution [140]. Zhang et al. introduced multi-scale dilated convolution into the field of image super-pixels, and the dilated convolution and spatial convolution are fused in each layer [141]. The dilated

convolutions can expand the receptive field without increasing computational complexity. Compared with the standard spatial convolution, the feature extraction network based on dilated convolution can extract more abundant feature information. Although the above operations can extract richer features, the mapped features of small targets are relatively few.

Based global region

Feature extraction from a global perspective can obtain richer information. The global feature extraction method was first applied in the fields of natural language processing with times or sequences such as text and speech. Recurrent neural networks and Transformer are two classical methods that extract features from a global perspective.

1. Recurrent neural networks and its variants

The biggest difference between RNN and convolutional neural networks is that RNN can realize some "memory function", which is the best choice for time series analysis [142]. Just as human beings can better understand the world with their past memories. RNN also implements a mechanism similar to the human brain, which retains a certain memory of the processed information, unlike other types of neural networks, which cannot retain the processed information [143, 144]. A typical RNN network consists of an input X , an output H and a neural network unit A . Different from ordinary neural networks, neural network unit A of RNN network is not only related to input and output, but also has a loop with itself. In other words, the essence of RNN is that the output information of the previous time will act on the network feature information of the next time. Due to the gradient disappearing, RNN can only have short-term memory. To alleviate this problem, a variant of the RNN network of long short-term memory (LSTM) is proposed, which uses the cell state with gate structure to make information flow continuously [145]. The gate structure is a way of information selection to remove or add information to the cell state. There are many other LSTM variants, such as clockwork RNN [146], which are applied for feature extraction of sequential tasks.

Recurrent neural networks and its variants have been applied in many tasks according to the characteristics of processing problems. Cho et al. designed a new RNN Encoder-Decoder, containing two recurrent neural networks [147]. One network is applied to encode a sequence of symbols into a vector representation of fixed-length and another network is used to decode the representation into another sequence of symbols. To maximize the conditional probability of a target sequence, encoder and decoder are jointly trained. Ergen proposed

an online training algorithm build on long short-term memory (LSTM) networks with the help of stochastic gradient descent and exponentiated gradient algorithms [148]. RNN and its variants can take temporal context into account and can achieve better performance in text processing and other fields.

2. Transformer

The Transformer is also a way to realize feature extraction from a global perspective [149]. It is first used in natural language processing. The biggest difference between Transformer and LSTM is that Transformer can operate in parallel and LSTM is sequential. The Transformer adopts an encoder–decoder structure, as shown in Fig. 11. The core of the encoder and decoder is a multi-head self-attention mechanism. Attention mechanism was first proposed in 2017. In essence, the attention mechanism is to weighted sum the value vectors of the elements. The query and key are used to calculate the weight coefficient of the corresponding value, where the weight coefficient is expressed by the similarity between the query vector and key vector. Suppose that the input is set to X , the key vector K , value vector V and query vector Q can be represented. The similarity of query and key can be described as

$$\text{Similarity}(Q, K_i) = Q \cdot K_i. \tag{49}$$

The attention can be presented as follows:

$$\text{Attention}(Q, K, V) = \sum_{i=1}^{L_x} \text{softmax}(\text{Similarity}(Q, K_i)) \cdot V_i. \tag{50}$$

Finally, attention can be calculated:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V, \tag{51}$$

where $\frac{1}{\sqrt{d_k}}$ is set to scaling factor. For multi-head attention, it allows the model to represent information from different locations and subspaces. Multi-head attention can be expressed as

$$\begin{aligned} \text{Multihead}(Q, K, V) \\ = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) W^O, \end{aligned} \tag{52}$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, the projections are parameter matrices $W_i^Q \in R^{d_{\text{model}} \times d_k}$, $W_i^K \in R^{d_{\text{model}} \times d_k}$, $W_i^V \in R^{d_{\text{model}} \times d_v}$, $W_i^O \in R^{hd_v \times d_{\text{model}}}$.

Taking an image as an example, the image is divided into patches as input of the Transformer. Assume that the size of the input image is $H \times W \times 3$, an image is divided into patches

size of $4 \times 4 \times 3$, containing $\frac{HW}{4^2}$ patches. These patches are flattened and are fed into a linear projection. Embedded patches and a position embedding are fed into a Transformer encoder. The output is reshaped to a new feature map F_1 of size $\frac{H}{4} \times \frac{W}{4} \times C_1$. In the same way, the output of the previous stage is the input of the next stage and the feature maps F_2 , F_3 and F_4 are obtained [150].

The Transformer-based feature extraction method is a hot topic in current research. Pyramid Vision Transformer is proposed to obtain a high-resolution output for dense prediction tasks and reduce computations of large feature maps in a progressive shrinking manner. Although Transformer can extract feature mapping from the global view, which is conducive to the detection of small targets, only coarse-grained global information is extracted and fine-grained local information is ignored. There are many optimized strategies to relieve this problem. The focal mechanism as a typical strategy is introduced to integrate global information and local information to improve the precision of target detection [151]. Other methods are designed such as Transformer in Transformer (TNT) [152], bottleneck Transformer (BoTNet) [153].

Unsupervised-based

Unsupervised deep learning methods for feature dimensionality reduction mainly include deep auto encoders, restricted Boltzmann machines, and deep belief networks. Taking deep auto encoders and deep belief networks as examples, the following describes in detail how to reduce the feature dimension.

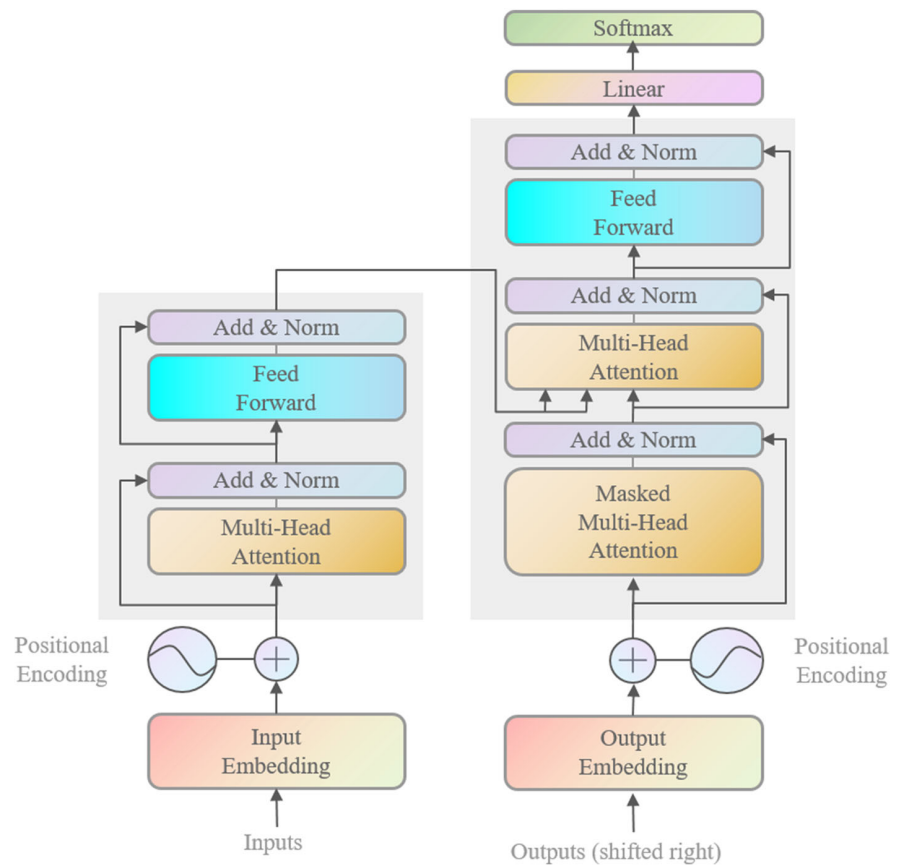
Deep auto encoder

A deep auto encoder is a generation model with more than hidden layers. When the architecture of the network has one hidden layer, the deep auto-encoder degenerate into an auto encoder. An auto encoder was introduced first by Rumelhart et al. [154] as a dimensionality reduction algorithm. It makes the target value equal to the input value with the help of the back-propagation algorithm. A simple auto encoder is composed of an input layer, hidden layer and output layer. This auto encoder reconstructs the input X from the output X' via the hidden layer h . A weight matrix $W_{X,h}$ and bias matrix $b_{X,h}$ are defined as parameters from the input layer to the hidden layer. $W_{h,X'}$ and $b_{h,X'}$ correspond to the parameters from the hidden layer to the output layer. The hidden layer can be presented as

$$h = \sigma(W_{X,h}X + b_{X,h}), \tag{53}$$

$$X' = \sigma(W_{h,X'}h + b_{h,X'}), \tag{54}$$

Fig. 11 The Transformer architecture [149]. Left is an encoder and right is a decoder



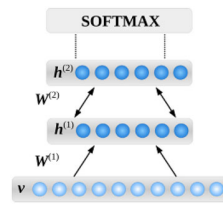
where $\sigma(\cdot)$ represents an activation function. It is worth noting that the dimension of hidden layer $|h|$ is smaller than that of input layer $|X|$ and output layer $|X'|$. From the input layer to the hidden layer is considered as a compression process, and from the hidden layer to the output layer is viewed as a reconstruction process. The goal of an auto encoder is to find a function $X' \approx X$ while $|h|$ is less than $|X|$. Although the neural network with a hidden layer can express all kinds of data, it is still difficult to achieve good generalization ability. An auto encoder has the problems of over-fitting and poor generalization ability, so it is difficult to effectively extract the internal abstract features of data [155]. To relieve these problems, the deep auto encoder is introduced.

Further, many variants were proposed [156]. For example, Liu provided us with a reference example of a deep auto-encoder to extract features for classification tasks [157]. To optimize the performance of feature extraction through obtaining the required network parameters based on proper learning rate, Song proposed a variable learning speed DAE (VLSAE) that adaptively adjusts its learning rate with the help of multi-scale reconstruction error (MRE) and weight update correlation (WUC) [158]. These variants make dimensionality reduction more robust and more applicable to multiple tasks.

Deep belief network

A deep belief network (DBN) is also a common unsupervised deep learning method to reduce the dimension of the feature vectors and extract features from high-dimensional data [159]. A deep belief network is a probabilistic generative model. The generation model establishes a joint distribution between observation data and labels, and evaluates both $p(\text{Observation}|\text{Label})$ and $p(\text{Label}|\text{Observation})$, while the discrimination model only evaluates $p(\text{Label}|\text{Observation})$. Compared with the supervised deep learning method, this is an effective method to solve the phenomena of slow learning speed and over-fitting of deep neural networks. The basic structural unit of a deep belief network is a Restricted Boltzmann Machine (RBM), as shown in Fig. 12. The hidden layer of the previous RBM is the visible layer of the next RBM, and the output of the previous RBM is the input of the next RBM. Restricted Boltzmann machines extract the abstract features by learning the probability density distribution of the data. The deep confidence network, which is to extract a variety of probability features through the superposition of RBM, learns the probability distribution of the data layer by layer. The deep belief network is highly flexible and easy to

Fig. 12 The DBN architecture [160]



expand, and it is the structural basis for building a new type of deep learning network.

A deep belief network has been applied in many fields, such as image recognition, information retrieval, natural language understanding, fault prediction and so on. To adapt to different problem environments and data types, many variants based on deep belief networks have been proposed. To optimize the initialization and local optimization in time series prediction, a deep belief network was applied to predict time series, which was applied to the approximation and short-term prediction of chaotic time series such as a logistic map and Lorenz chaos [161]. Unsupervised depth belief network (DBN) is also adopted to extract features and reduce data dimensions from the fusion observation of Electro-Dermal activity (EDA), Photoplethysmography (PPG) and Zygomaticus Electromyography (zEMG) sensor signals. The prepared feature fusion vector is generated by combing DBN with the statistical features of EDA, PPG and zEMG for emotion classification [162].

Based on semi-supervised

Semi-supervised feature dimensionality reduction mainly uses supervised or unsupervised feature dimensionality reduction. Specifically, there are two common deep learning methods based on semi-supervised: unsupervised pre-training [163] and pseudo-supervised pre-training [164]. Unsupervised pre-training dimension reduction refers to training all data with unsupervised deep learning methods, such as deep auto encoder, and then training the classifier with labeled data. The unlabeled data determined to be correctly classified are selected and added to the training set, and then the network is trained. Repeat the above operation continuously. The pseudo-supervised pre-training dimensionality reduction method attaches pseudo label information to the unlabeled data utilizing semi-supervised algorithm or clustering algorithm and uses the pseudo label information to pretrain the network, and then uses the labeled data to fine-tune the network. It is worth noting that pseudo labels refer to tags that predict unlabeled data with the help of the network and take the prediction results as unlabeled data. Its idea is simple self-training. Semi-supervised feature dimensionality reduction helps to extract feature maps with generalization ability, improve the generalization ability of the model, and help the model reduce the interdomain difference between different tasks.

Semi-supervised feature dimensionality reduction method has been widely used in different types of tasks. Due to a few labeled data, it is conducive to transfer learning between different data and enhances the applicability of the model to different tasks. For the phrase-based translation model, Lu et al. learned some new and effective features with the help of the deep auto encoder (DAE) paradigm [165]. To make full use of large-scale unlabeled data, Li et al. proposed an effective method for semi-supervised deep learning that optimized the performance of the model, where a classifier was trained and generated pseudo-labels on unlabeled data [166]. Wu et al. proposed a semi-supervised method for classification using limited labeled data and abundant unlabeled data to train a deep neural network, where unlabeled data are utilized with their pseudo labels (cluster labels) to train the network. The limited labeled data are applied to fine-tune parameters [167].

Applications

Deep learning is widely used in the fields of speech recognition, image classification, and video recognition. To adapt the demand of image features e-commerce to the mobile terminal to develop, a multi-scale deep neural network is used to extract image features that are more robust to complex image background and target object scale changes. At the same time, the similarity between the pictures is learned according to the information of the category of the product category, to better perform the search of the products whose image contents match, and improve the accuracy of the query. The performance of the speech recognizer has a great influence on the performance of keyword detection, while the traditional keyword detection is an acoustic model of the LVCSR using the GMM (Gaussian Mixture Model) and the HMM (Hidden Markov Model) combined GMM-HMM model [168, 169], its recognition rate is not high. In recent years, deep learning technology has had a huge impact on speech recognition. People have conducted deep research on DNN (Deep Neural Network) instead of GMM (Gaussian Mixture Model) to compose the DNN-HMM acoustic model [169, 170]. The DNN-HMM acoustic model was used to replace the GMM-HMM acoustic model in keyword detection, and a keyword detection system was established based on the DNN-HMM acoustic model. The experimental results show that the speech recognizer based on the DNN-HMM model has a higher recognition rate than the speech recognizer based on the GMM-HMM model, and the performance of the keyword detection system is better.

Deep learning, as a powerful tool for mankind to solve intelligent problems by mimicking the brain structure, provides new techniques and ideas for research and practice in other related fields and has broad research prospects. In particular, with the in-depth study of deep learning, the

introduction of various more efficient deep learning models and the improvement of learning algorithms, the use of deep learning to solve a variety of intelligent problems has gradually evolved into the mainstream of machine learning research. It also makes deep learning gradually occupy the core position in the field of machine learning. However, in the study of deep learning related theories and learning algorithms, there are still some problems that need further discussion. For example, how to improve the discriminating ability and interpreting ability of each layer feature extracted by deep learning in unsupervised learning scenarios? How to balance the relationship between the number of deep learning network layers and the number of single-layer neural units, making deep learning face different application scenarios to improve its modeling and promotion performance? Can deep learning be used in more extensive practical applications such as image segmentation, multi-modality sensing, and missing data recovery? The research and discussion of these issues will have very important theoretical and practical significance. It will also promote the further development of deep learning [171–173].

Conclusion

The dimensionality reduction process of high-dimensional data sets can be decomposed into three phases that are independent and related to each other: (1) description of data set structure; (2) metrics of data set structure; (3) structure-based dimensionality reduction criterion. The proposed and formed dimension reduction method also includes three aspects of work: (1) establishing the corresponding mathematical model of the research problem and the data set structure model; (2) proposing corresponding metrics or selection rules for the model; (3) establishing data-based dimension reduction criterion or loss rule of the set structure.

Pattern feature extraction and selection is the basis for recognizing training learning and also the first step in recognition, besides it plays a key role in the accuracy of recognition. Simultaneous feature extraction is also an important part of data mining. The use of reasonable, reliable and feasible feature compression methods for the resolution of different problems and the processing of data features will enable the recognition to take a big step toward higher accuracy. The main significance of feature extraction is "low loss and dimensionality reduction", which tends to simplify the problem to be easy to calculate or to increase the speed of operation and make learning and training of the system easy.

The rapid development of computer software and hardware technology has promoted the large-scale application of pattern recognition technology, for example, it is used widely in agricultural, biological, medical, meteorological, industrial product testing, and other fields. Pattern recognition

technology develops rapidly and plays a very important role in the field of computer vision and hearing, such as handwriting recognition, speech recognition, and biometrics. Feature extraction and selection can be regarded as the premise and key of pattern recognition. As the first step of pattern recognition, pattern feature extraction and selection must analyze and process this information. Feature extraction and selection have also been applied in many fields.

From the perspective of applications, most of the contacts, in reality, are nonlinear and time-varying systems. Therefore, the current research hotspots are on the feature extraction and selection of high-dimensional nonlinear modes. From a methodological point of view, cutting-edge research is the organic integration of multiple theories. For example, information theory, neural networks, and other theories are introduced into feature extraction; organic integration of different methods gives full play to their advantages and avoids weaknesses; the current manifold learning and independent component analysis as emerging theoretical methods will also arouse the attention of many scholars. In addition, with the increase of data explosion, how to extract and select the feature with strong representation ability from small sample data is an important research direction in the future.

Funding This work is supported by Natural Science Foundation of Shandong Province in China (ZR2020MF076, ZR2020MF133); National Nature Science Foundation of China (No. 62072289); Focus on Research and Development Plan in Shandong Province (No.: 2019GNC106115); China Postdoctoral Science Foundation (No.: 2018M630797); Shandong Province Higher Educational Science and Technology Program (No.: J18KA308); Taishan Scholar Program of Shandong Province of China (No.: TSHW201502038).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Hughes G (1968) On the mean accuracy of statistical pattern recognizers. *IEEE Trans Inf Theory* 14(1):55–63

2. Keogh E, Mueen A (2010) Curse of dimensionality. In: Sammut C, Webb GI (eds) *Encyclopedia of machine learning*. Springer, US
3. Ye H, Sugihara G (2016) Information leverage in interconnected ecosystems: overcoming the curse of dimensionality. *Science* 353(6302):922–925
4. Ding SF et al (2012) A survey on feature extraction for pattern recognition. *Artif Intell Rev* 37(3):169–180
5. Cunningham JP, Yu BM (2014) Dimensionality reduction for large-scale neural recordings. *Nat Neurosci* 17(11):1500–1509
6. Cunningham JP, Ghahramani Z (2015) Linear dimensionality reduction: survey, insights, and generalizations. *J Mach Learn Res* 16:2859–2900
7. Ornek C, Vural E (2019) Nonlinear supervised dimensionality reduction via smooth regular embeddings. *Pattern Recogn* 87:55–66
8. Silva R, Melo-Pinto P (2021) A review of different dimensionality reduction methods for the prediction of sugar content from hyperspectral images of wine grape berries. *Appl Soft Comput* 113:107889
9. Aziz R, Verma CK, Srivastava N (2017) Dimension reduction methods for microarray data: a review. *AIMS Bioeng* 4(2):179–197
10. Zebari R et al (2020) A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *J Appl Sci Technol Trends* 1(2):56–70
11. Aziz R, Verma CK, Srivastava N (2018) Artificial neural network classification of high dimensional data with novel optimization approach of dimension reduction. *Ann Data Sci* 5(4):615–635
12. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. *Comput Electr Eng* 40(1):16–28
13. Dash M, Liu H (1997) Feature selection for classification. *Intell Data Anal* 1(1–4):131–156
14. Dash M, Liu H, Yao J (1997) Dimensionality reduction of unsupervised data. In: *Proceedings ninth IEEE international conference on tools with artificial intelligence*. IEEE
15. Hu J (2008) Survey on feature dimension reduction for high-dimensional data. *Appl Res Comput* 25(9):2601–2606
16. Hira ZM, Gillies DF (2015) A review of feature selection and feature extraction methods applied on microarray data. *Adv Bioinformatics* 2015:198363
17. Hua JP, Tembe WD, Dougherty ER (2009) Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recogn* 42(3):409–424
18. Welch WJ (1982) Branch-and-bound search for experimental designs based on D optimality and other criteria. *Technometrics* 24(1):41–48
19. Wang JH et al (2016) Analysis of imbalanced weather data based on branch-and-bound approach. *Appl Res Comput* 33(6):1648–1652
20. Gokce A, Hsiao K-T, Advani SG (2002) Branch and bound search to optimize injection gate locations in liquid composite molding processes. *Compos A Appl Sci Manuf* 33(9):1263–1272
21. Ow PS, Morton TE (1988) Filtered beam search in scheduling. *Int J Prod Res* 26(1):35–62
22. Kumar A et al (2013) Beam search algorithms for multilabel learning. *Mach Learn* 92(1):65–89
23. Araya I, Riff M-C (2014) A beam search approach to the container loading problem. *Comput Oper Res* 43:100–107
24. Wang SJ, Xi LF, Zhou BH (2007) Filtered-beam-search-based algorithm for dynamic rescheduling in FMS. *Robot Comput Integr Manuf* 23(4):457–468
25. Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 27(8):1226–1238
26. Cotter SF et al (1999) Forward sequential algorithms for best basis selection. *IEE Proc Vis Image Signal Process* 146(5):235–244
27. Pereira RD et al (2013) Modified sequential forward selection applied to predicting septic shock outcome in the intensive care unit. *Synergies of soft computing and statistics for intelligent data analysis*. Springer, pp 469–477
28. Korff RE (1985) Depth-first iterative-deepening: an optimal admissible tree search. *Artif Intell* 27(1):97–109
29. Lo WY, Zwicker M (2010) Bidirectional search for interactive motion synthesis. *Comput Graph Forum* 29(2):563–573
30. Kaindl H, Kainz G (1997) Bidirectional heuristic search reconsidered. *J Artif Intell Res* 7:283–317
31. Zhou J, Müller M (2004) Solving systems of difference constraints incrementally with bidirectional search. *Algorithmica* 39(3):255–274
32. Jia W, Zhao D, Ding L (2016) An optimized RBF neural network algorithm based on partial least squares and genetic algorithm for classification of small sample. *Appl Soft Comput* 48:373–384
33. Ruiz GR et al (2016) Genetic algorithm for building envelope calibration. *Appl Energy* 168:691–705
34. Pezzella F, Morganti G, Ciaschetti G (2008) A genetic algorithm for the flexible job-shop scheduling problem. *Comput Oper Res* 35(10):3202–3212
35. Rostami M et al (2021) Review of swarm intelligence-based feature selection methods. *Eng Appl Artif Intell* 100:104210
36. Jordehi AR, Jasni J (2015) Particle swarm optimisation for discrete optimisation problems: a review. *Artif Intell Rev* 43(2):243–258
37. Eroğlu Y, Seçkiner SU (2012) Design of wind farm layout using ant colony algorithm. *Renew Energy* 44:53–62
38. He J, Hou Z (2012) Ant colony algorithm for traffic signal timing optimization. *Adv Eng Softw* 43(1):14–18
39. Reed M, Yiannakou A, Evering R (2014) An ant colony algorithm for the multi-compartment vehicle routing problem. *Appl Soft Comput* 15:169–176
40. Mao Y et al (2007) Survey for study of feature selection algorithms. Moshi Shibie yu Rengong Zhineng/Pattern Recogn Artif Intell 20(2):211–218
41. Narendra PM, Fukunaga K (1977) A branch and bound algorithm for feature subset selection. *IEEE Trans Comput* 26(09):917–922
42. Zhang HB, Sun GY (1999) Tabu search algorithm for feature selection. *Acta Autom Sin* 25(4):457–466
43. Zhang X (1998) Dynamic programming method for feature selection. *Acta Autom Sin* 24:680–684
44. Lewis DD (1992) An evaluation of phrasal and clustered representations on a text categorization task. In: *Proceedings of the 15th annual international ACM SIGIR conference on research and development in information retrieval*, pp 37–50
45. Simek K et al (2004) Using SVD and SVM methods for selection, classification, clustering and modeling of DNA microarray data. *Eng Appl Artif Intell* 17(4):417–427
46. Fajarewicz K, Wiench M (2003) Selecting differentially expressed genes for colon tumor classification. *Int J Appl Math Comput Sci* 13:327–335
47. Mishra D, Sharma S (2021) Performance analysis of dimensionality reduction techniques: a comprehensive review. *Adv Mech Eng*:639–651
48. Moore B (1981) Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans Autom Control* 26(1):17–32
49. Abdi H, Williams LJ (2010) *Principal component analysis*. Wiley Interdiscip Rev Comput Stat 2(4):433–459
50. Qiang LI et al (2005) Online palmprint identification based on improved 2D PCA. *Acta Electron Sin* 33(10):1886–1889
51. Senkov O et al (2015) Accelerated exploration of multi-principal element alloys with solid solution phases. *Nat Commun* 6(1):1–10

52. Song F et al (2008) A highly scalable incremental facial feature extraction method. *Neurocomputing* 71(10–12):1883–1888
53. Wang S et al (2016) Semi-supervised linear discriminant analysis for dimension reduction and classification. *Pattern Recogn* 57:179–189
54. Jin Z et al (2001) Face recognition based on the uncorrelated discriminant transformation. *Pattern Recogn* 34(7):1405–1416
55. Dehak N et al (2010) Front-end factor analysis for speaker verification. *IEEE Trans Audio Speech Lang Process* 19(4):788–798
56. Comon P (1994) Independent component analysis, a new concept? *Signal Process* 36(3):287–314
57. Hyvärinen A, Oja E (2000) Independent component analysis: algorithms and applications. *Neural Netw* 13(4–5):411–430
58. Huang QH, Wang S, Liu Z (2007) Improved algorithm of image feature extraction based on independent component analysis. *Opto-Electron Eng* 1:121–125
59. Yuen PC, Lai J-H (2002) Face representation using independent component analysis. *Pattern Recogn* 35(6):1247–1257
60. Wang J, Chang C-I (2006) Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE Trans Geosci Remote Sens* 44(6):1586–1600
61. Kong W et al (2008) A review of independent component analysis application to microarray gene expression data. *Biotechniques* 45(5):501–520
62. Musheer RA, Verma C, Srivastava N (2019) Novel machine learning approach for classification of high-dimensional microarray data. *Soft Comput* 23(24):13409–13421
63. Beatty M, Manjunath B (1997) Dimensionality reduction using multi-dimensional scaling for content-based retrieval. In: *Proceedings of international conference on image processing*. IEEE
64. Cambria E, Mazzocco T, Hussain A (2013) Application of multi-dimensional scaling and artificial neural networks for biologically inspired opinion mining. *Biol Inspir Cognit Architectures* 4:41–53
65. Dzidolikaite A (2015) Genetic algorithms for multidimensional scaling. *Mokslas Lietuvos ateitis/Sci Future Lithuania* 7(3):275–279
66. Wall ME, Rechtsteiner A, Rocha LM (2003) Singular value decomposition and principal component analysis. A practical approach to microarray data analysis. Springer, pp 91–109
67. Chen Y et al (2018) Application of singular value decomposition algorithm to dimension-reduced clustering analysis of daily load profiles. *Automat Electr Power Syst* 42(3):105–111
68. Kang M, Kim JM (2013) Singular value decomposition based feature extraction approaches for classifying faults of induction motors. *Mech Syst Signal Process* 41(1–2):348–356
69. Yang J (2005) KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition. *IEEE Trans Pattern Anal Mach Intell* 27:230
70. Mika S et al (1999) Fisher discriminant analysis with kernels. In: *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop* (cat. no. 98th8468). IEEE
71. Zhu X et al (2012) Dimensionality reduction by mixed kernel canonical correlation analysis. *Pattern Recogn* 45(8):3003–3016
72. Lee DD et al (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* 401:788
73. Wang Y et al (2005) Non-negative matrix factorization framework for face recognition. *Int J Pattern Recognit Artif Intell* 19(04):495–511
74. Lee CW et al (2003) Font classification using NMF. In: *Computer analysis of images & patterns, international conference, Caip, Groningen, the Netherlands, August*, pp 470–477
75. Pauca VP, Piper J, Plemmons RJ (2006) Nonnegative matrix factorization for spectral data analysis. *Linear Algebra Appl* 416(1):29–47
76. Liu W, Yuan K, Ye D (2008) Reducing microarray data via nonnegative matrix factorization for visualization and clustering analysis. *J Biomed Inform* 41(4):602–606
77. Shannon CE (1948) A mathematical theory of communication. *Bell Syst Tech J* 27(3):379–423
78. Quiroga RQ, Panzeri S (2009) Extracting information from neuronal populations: information theory and decoding approaches. *Nat Rev Neurosci* 10:173
79. ShiZhong L, XiaoJun J, SuLei Z (1999) The application study of entropy analysis method in feature extraction. *J North China Inst Technol* 20(3):278–281
80. Ding S, Jin F, Wang X (2005) Information feature compression algorithm based on SCEC. *Mini-micro Syst* 26(7):1202–1205
81. Daubechies I (2009) *The wavelet transform, time-frequency localization and signal analysis*. Princeton University Press, Princeton
82. Bruce LM, Koger CH, Li J (2002) Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. *IEEE Trans Geosci Remote Sens* 40(10):2331–2338
83. Zhang J, Zhang B, Jiang X-Z (2000) Analyses of feature extraction methods based on wavelet transform. *Signal Process* 16(2):156–162
84. He H, Starzyk JA (2005) A self-organizing learning array system for power quality classification based on wavelet transform. *IEEE Trans Power Deliv* 21(1):286–295
85. Hamaneh MB et al (2013) Automated removal of EKG artifact from EEG data using independent component analysis and continuous wavelet transformation. *IEEE Trans Biomed Eng* 61(6):1634–1641
86. Jones MC, Sibson R (1987) What is projection pursuit? *J R Stat Soc Ser A (General)* 150(1):1–18
87. Durvaux F et al (2015) Efficient selection of time samples for higher-order DPA with projection pursuits. In: *International workshop on constructive side-channel analysis and secure design*. Springer, pp 34–50
88. Gao MT, Wang ZO (2007) A new algorithm for text clustering based on projection pursuit. In: *2007 International conference on machine learning and cybernetics*. IEEE 6: 3401–3405
89. Law MH, Jain AK (2006) Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Trans Pattern Anal Mach Intell* 28(3):377–391
90. Raducanu B, Dornaika F (2012) A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recogn* 45(6):2432–2444
91. Olson CC, Judd KP, Nichols JM (2018) Manifold learning techniques for unsupervised anomaly detection. *Expert Syst Appl* 91:374–385
92. Balasubramanian M et al (2002) The isomap algorithm and topological stability. *Science* 295(5552):7–7
93. Zhang Z, Chow TW, Zhao M (2012) M-Isomap: Orthogonal constrained marginal isomap for nonlinear dimensionality reduction. *IEEE Trans Cybern* 43(1):180–191
94. Tenenbaum JB, De Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323
95. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326
96. Pan Y, Ge SS, Al Mamun A (2009) Weighted locally linear embedding for dimension reduction. *Pattern Recogn* 42(5):798–811
97. Chen J, Liu Y (2011) Locally linear embedding: a survey. *Artif Intell Rev* 36(1):29–48
98. Ge SS, He H, Shen C (2012) Geometrically local embedding in manifolds for dimension reduction. *Pattern Recogn* 45(4):1455–1470
99. Yin FP (2012) Based on improved LLE and FSVM method in face recognition of application. *Sci Technol Eng* 12(34):9390–9395

100. De Ridder D et al (2003) Supervised locally linear embedding. Artificial neural networks and neural information processing—I-CANN/ICONIP 2003. Springer, pp 333–341
101. Zhang S-Q (2009) Enhanced supervised locally linear embedding. *Pattern Recogn Lett* 30(13):1208–1218
102. Wang H et al (2006) Application of dimension reduction on using improved LLE based on clustering. *J Comput Res Dev* 43(8):1485
103. He X, Niyogi P (2003) Locality preserving projections. *Adv Neural Inf Process Syst* 16:153–160
104. Kokiopoulou E, Saad Y (2007) Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Trans Pattern Anal Mach Intell* 29(12):2143–2156
105. Wong WK, Zhao H (2012) Supervised optimal locality preserving projection. *Pattern Recogn* 45(1):186–197
106. Li W et al (2011) Locality-preserving dimensionality reduction and classification for hyperspectral image analysis. *IEEE Trans Geosci Remote Sens* 50(4):1185–1198
107. Jiang R et al (2016) Dimensionality reduction on anchorgraph with an efficient locality preserving projection. *Neurocomputing* 187:109–118
108. Shen Z-H, Pan Y-H, Wang S (2008) A supervised locality preserving projection algorithm for dimensionality reduction. *Pattern Recogn Artif Intell* 21(2):233–239
109. Shikkenawis G, Mitra SK (2016) On some variants of locality preserving projection. *Neurocomputing* 173:196–211
110. Zhan YB (2011) Research on manifold learning theories, methods and applications. Ph.D. thesis of University of Defence Technology
111. Ding S et al (2004) New PCA feature compression algorithm based on information theory. *Mini-micro Syst* 25(4):694–697
112. Ding S, Jin F, Wang J (2003) Information feature analysis and selection of orthogonal transformation. *Acta Geodaetica Et Cartogr Sin* 32(1):73–77
113. Fei Z, With P (2005) Facial feature extraction using a cascade of model-based algorithms. In: *Advanced video and signal based surveillance, 2005. AVSS 2005. IEEE conference on* pp 348–353
114. Nabti M, Bouridane A (2008) An effective and fast iris recognition system based on a combined multiscale feature extraction technique. *Pattern Recogn* 41(3):868–879
115. Ji S, Ye J (2008) Generalized linear discriminant analysis: a unified framework and efficient model selection. *IEEE Trans Neural Netw* 19(10):1768–1782
116. Tibshirani R (1996) Regression shrinkage and selection via the Lasso. *J R Stat Soc Ser B (Methodological)* 58:267–288
117. Rodriguez JD, Perez A, Lozano JA (2009) Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Trans Pattern Anal Mach Intell* 32(3):569–575
118. Bhadra A et al (2019) Lasso meets horseshoe: a survey. *Stat Sci* 34(3):405–427
119. Kohannim O et al (2012) Discovery and replication of gene influences on brain structure using LASSO regression. *Front Neurosci* 6:115
120. Chen LF et al (2000) A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recogn* 33(10):1713–1726
121. Huang R (2007) A margin based feature extraction algorithm for the small sample size problem. *Chin J Comput* 30(7):1173–1178
122. Lu J et al (2020) Learning from very few samples: a survey. *arXiv preprint arXiv:2009.02653*
123. Wang Y et al (2020) Generalizing from a few examples: A survey on few-shot learning. *ACM Comput Sur (CSUR)* 53(3):1–34
124. Hertz T, Hillel AB, Weinshall D (2006) Learning a kernel function for classification with small training samples. In: *Proceedings of the 23rd international conference on Machine learning*, pp 401–408
125. Aziz R, Verma CK, Srivastava N (2017) A novel approach for dimension reduction of microarray. *Comput Biol Chem* 71:161–169
126. Aziz R et al (2017) Artificial neural network classification of microarray data using new hybrid gene selection method. *Int J Data Min Bioinform* 17(1):42
127. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313:504
128. Hinton GE, Osindero S, Teh YW (2014) A fast learning algorithm for deep belief nets. *Neural Comput* 18(7):1527–1554
129. Arel I, Rose DC, Karnowski TP (2010) Deep machine learning—a new frontier in artificial intelligence research [research frontier]. *IEEE Comput Intell Mag* 5(4):13–18
130. Yi S, Wang X, Tang X (2014) Deep learning face representation by joint identification-verification. *Adv Neural Inf Process Syst* 27
131. Wen W et al (2016) Learning structured sparsity in deep neural networks. *Adv Neural Inf Process Syst* 29:2074–2082
132. Yu K et al (2013) Deep learning: yesterday, today, and tomorrow. *J Comput Res Dev* 50(9):1799–1804
133. Traore BB, Kamsu-Foguem B, Tangara F (2018) Deep convolution neural network for image recognition. *Eco Inform* 48:257–268
134. Al-Saffar AAM, Tao H, Talab MA (2017) Review of deep convolution neural network in image classification. In: *2017 International conference on radar, antenna, microwave, electronics, and telecommunications (ICRAMET). IEEE*, pp 26–31
135. Yang J, Li J (2017) Application of deep convolution neural network. In: *2017 14th International computer conference on wavelet active media technology and information processing (ICCWAMTIP). IEEE*, pp 229–232
136. He K et al (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
137. Ren S et al (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv Neural Inf Process Syst* 28:91–99
138. He K et al (2017) Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision*, pp 2961–2969
139. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv: 1511.07122*
140. Cui L et al (2020) Context-aware block net for small object detection. *IEEE Trans Cybern (Early access)*, pp 1–14
141. Zhang Z, Wang X, Jung C (2018) DCSR: Dilated convolutions for single image super-resolution. *IEEE Trans Image Process* 28(4):1625–1635
142. Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. *arXiv preprint arXiv: 1409.2329*
143. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv: 1506.00019*
144. Jozefowicz R, Zaremba W, Sutskever I (2015) An empirical exploration of recurrent network architectures. In: *International conference on machine learning. PMLR*, pp 2342–2350
145. Yu Y et al (2019) A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput* 31(7):1235–1270
146. Koutnik J et al (2014) A clockwork rnn. In: *International conference on machine learning. PMLR*, pp 1863–1871
147. Cho K et al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*
148. Ergen T, Mirza AH, Kozat SS (2019) Energy-efficient LSTM networks for online learning. *IEEE Trans Neural Netw Learn Syst* 99:1–13
149. Vaswani A et al (2017) Attention is all you need. In: *Advances in neural information processing systems*, pp 5998–6008

150. Wang W et al (2021) Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. arXiv preprint arXiv: 2102.12122
151. Yang J et al (2021) Focal self-attention for local-global interactions in vision transformers. arXiv preprint arXiv: 2107.00641
152. Han K et al (2021) Transformer in transformer. arXiv preprint arXiv: 2103.00112
153. Srinivas A et al (2021) Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 16519–16529
154. Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science
155. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507
156. Dong G et al (2018) A review of the autoencoder and its variants: a comparative perspective from target recognition in synthetic-aperture radar images. *IEEE Geosci Remote Sens Mag* 6(3):44–68
157. Liu T et al (2017) NIRS feature extraction based on deep auto-encoder neural network. *Infrared Phys Technol* 87:124–128
158. Song W et al (2021) A new deep auto-encoder using multi-scale reconstruction errors and weight update correlation. *Inf Sci* 559:130–152
159. Hinton GE (2009) Deep belief networks. *Scholarpedia* 4(5):5947
160. Roder M et al (2021) Reinforcing learning in Deep Belief Networks through nature-inspired optimization. *App Soft Comput* 108:107466
161. Kuremoto T et al (2014) Time series forecasting using a deep belief network with restricted Boltzmann machines. *Neurocomputing* 137:47–56
162. Hassan MM et al (2018) Human emotion recognition using deep belief network architecture. *Inf Fusion* 51:10–18
163. Gogna A, Majumdar A (2016) Semi supervised autoencoder. In: International conference on neural information processing. Springer
164. Lee DH (2013) Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Workshop on challenges in representation learning, ICML 3(2):896
165. Lu S, Chen Z, Xu B (2014) Learning new semi-supervised deep auto-encoder features for statistical machine translation. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, pp 122–132
166. Li Z, Ko B, Choi H-J (2019) Naive semi-supervised deep learning using pseudo-label. *Peer-to-peer Netw Appl* 12(5):1358–1368
167. Wu H, Prasad S (2017) Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Trans Image Process* 27(3):1259–1270
168. Sainath TN et al (2015) Convolutional, long short-term memory, fully connected deep neural networks. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
169. Seide F et al (2011) Feature engineering in context-dependent deep neural networks for conversational speech transcription. In: 2011 IEEE workshop on automatic speech recognition & understanding. IEEE, pp 24–29
170. Li J et al (2012) Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM. In: 2012 IEEE Spoken Language Technology Workshop (SLT). IEEE, pp 131–136
171. Schmidhuber J (2015) Deep learning in neural networks: an overview. *Neural Netw* 61:85–117
172. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436
173. Lngkvist M, Karlsson L, Loutfi A (2014) A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recogn Lett* 42:11–24

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.