Unsupervised Learning Algorithms in Big Data: An Overview

Mohan Zhang

Beijing 21st century international school, Beijing, China

Email: 2695422746@qq.com

Abstract. With the progression of technology, there are more ways to produce complex and spiral data without signs. For the development of Artificial intelligence, machine learning is generated to help humans with human training or without. In this paper, based on the characteristics and properties of unsupervised algorithms, first, we are going to identify and classify methods of unsupervised dig data analysis into clustering and dimensionality reduction, and then systematically conclude the clustering algorithms (K-means, Hierarchical clustering, GMM, and DBSCAN) and dimensionality reduction algorithm (PCA, LLE, and MDS). Then, we will discuss some of those applications. Eventually, we will conclude and imagine the future development of big data analysis.

Keywords: Unsupervised Machine Learning, Big Data, Clustering algorithms, Dimensionality reduction algorithms, Applications

1 Introduction

Progression of world Internet technology does change our world, and the change also occurs in information formation and complexity of data. Due to the sheer amount of data which is various in type and size, some new big data analysis approaches appeared. One is machine learning which can help humans to deal with those data by using computers. In this paper, we will focus on an unsupervised machine learning algorithm that is the subpart of machine learning.

Unsupervised learning [1] has been widely used in big data analysis. No specific objective, training data without labels, and hard to determine goodness of outcome after learning are three common differences from supervised learning [2]. Thus, for mass and unlabeled data, unsupervised learning can be used to find potential and hidden patterns of structures within data, and it categorizes data into different clusters or low dimensions.

There are several examples of unsupervised learning algorithms. Clustering algorithms include K-means [3], Hierarchical Clustering [4], GMM (Gaussian Mixture Models) [5], and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [6]. Dimensionality reduction [7] algorithms include PCA (Principle compo-

A. Holl et al. (Eds.): ICHESS 2022, ASSEHR 720, pp. 910–931, 2022.
https://doi.org/10.2991/978-2-494069-89-3_107

nent analysis) [8], LEE (Locally Linear Embedding) [9], and MDS (Multidimensional Scaling) [10]. These two classifications are common categories.

Clustering algorithms and Dimension reduction algorithms are an important part of unsupervised big data analysis. The content in this paper includes the analysis of algorithms in the context of big data. Then, we will conclude those methods, and we will look ahead about its applications.

2 Clustering

Clustering [11] has been used for probing data analysis, and it is used for unsupervised machine learning. Clustering algorithms commonly divide a clump of data into different categories or clumps with the same features within and different features between.

2.1 K-means

K-means [3] is the most common way for clustering due to its properties: easy to understanding and short coding.

2.1.1 K-means algorithm.

K-means algorithm [12] is easy to understand. The objection of K-means is to divide a given group of data into K groups, and then it will give each sample the corresponding center. There are 4 steps.

- Pre-process data (To standardize the data and remove then outliers).
- Randomly select K centers.

$$\mu_1, \mu_2, \ldots, \mu_k \tag{1}$$

- Defining the loss function.

$$J(c, \mu) = \min \sum_{i=1}^{n} \|x_i - \mu_{ci}\|^2 \tag{2}$$

- For t=0, 1, 2, ... is the number of iterative steps, repeat the following process to know the convergence of the function: in Equation (2). Here are two steps.

For each sample x_i, it is assigned to the nearest center.

$$c_i^{\mathrm{t}} < -\arg \min_k \|x_i - \mu_k^i\|^2 \tag{3}$$

For each class center K, the center of the class is recalculated.

$$\mu_k^{(t+1)} < -\arg \min_{\mu} \sum_{i:c_i^t=k}^{b} \|x_i - \mu\|^2 \tag{4}$$

2.1.2 Advantages and disadvantages of K-means Clustering.

- Advantages

— K-means is easy and high efficiency.
— This algorithm is easy to understand and implement.

- Disadvantages

— It is necessary to determine the number of clusters manually in advance.
— It is sensitive to the setting of the initial value, and the result of the algorithm is related to the selection of the initial value.
— It is not resistant to noise and abnormal data. If an outlier has a large value, it can seriously affect the data distribution.
— The problem of data distribution clustering with non-convex shapes cannot be solved.
— Non-spherical clusters cannot be identified.

2.1.3 Applications of K-means.

K-means had been used to analyze the regional energy consumption of different industries to improve regional energy efficiency [13]. Another application of K-means was in Amazon Web Services Lambada Function. In this paper, they presented a novel application in cloud computing which finds that, if K-means was used in unsupervised machine learning in cloud computing, there was a negligible latency within mobile applications and Lambada function [14].

Additionally, K-means was applied in the classification of personality. For that purpose, the accuracy rate was justified by using this algorithm, and, finally, they found 16 types of personality [15].

2.2 Hierarchical Clustering

Hierarchical clustering has been used [4]. We do not have to set the fixed K value which is the number of clusters. It can bend closed points or clusters into a new cluster. Finally, it will form a tree diagram that shows relations. There are two types of Hierarchical Clustering algorithms [4].

2.2.1 Hierarchical Clustering algorithms.

- Divisive method

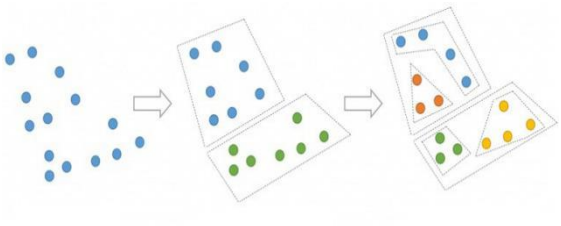


Fig. 1. The Hierarchy(down)

This algorithm is to divide a cluster into some points or smaller clusters just as Figure.1. Here are 4 steps.

- First, put raw data into the first clump C, and it will form the topmost part of the hierarchies.
- Second, using K-means to divide clump C into K clumps(groups)

  Ci, i = 1, 2, ..., k, and to form a new layer.

- Repeating the second step until every clump cannot be divided or end standards are satisfied.

- Agglomerative method

This algorithm is to combine small data fragments into a big cluster. To be more specific. The merging algorithm of hierarchical clustering determines the similarity between data points of each category and all data points by calculating the distance between them. The smaller the distance between them, the higher the similarity. Thus, the two nearest data points or categories are combined to generate a tree diagram. Here are two steps.

- Finding two points that have the shortest distance, and forming a new clump.
- Repeating the step above until there is only one clump remaining.

2.2.2 Distance calculation algorithms.

(Assuming clusters C_i, and C_j)

- Single-link

$$\mathrm{D}(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \qquad (5)$$

- Complete-link

$$\mathrm{D}(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \qquad (6)$$

- UPGMA

$$D(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum \sum_{x \in C_i, y \in C_j} d(x, y) \quad (7)$$

2.2.3 Advantages and disadvantages of Hierarchical Clustering.

- Advantages

— The similarity of distance and rules is easy to define with few restrictions.
— There is no need to specify the clustering number in advance.
— Hierarchical relationships of classes can be found
— Odd shapes can be clustered.

- Disadvantages

— High computational complexity.
— Singular values can also have a great influence.
— Algorithms are likely to cluster into chains.

2.2.4 Applications of Hierarchical Clustering.

Hierarchical Clustering had been used to determine meaningful tourism by detecting geo-localized data from 1505 users in the Zeeland app [16]. Another application of Hierarchical Clustering was to evaluate students' academic performance in different subjects and compared those students' scores data with a big data set [17].

In addition, Hierarchical clustering had been used to multi-parametric for prostate cancer to differentiate tumor and normal tissue and the result showed that the accuracy of differentiating reaches clinical standards [18].

2.3 GMM

The Gaussian Mixed Model [5] means to the linear combination of several Gaussian distribution functions like Figure.2. In GMM, the learning process is to train several probability distributions. The so-called mixed Gaussian model is to estimate the probability density distribution of samples, and the estimated model is the weighted sum of several Gaussian models (The number of models is established before training). Each Gaussian model represents a class (a Cluster) of data. By projecting the data in the sample onto several Gaussian models, the probabilities of each class can be obtained. Then we can choose the class with the highest probability to decide the result.

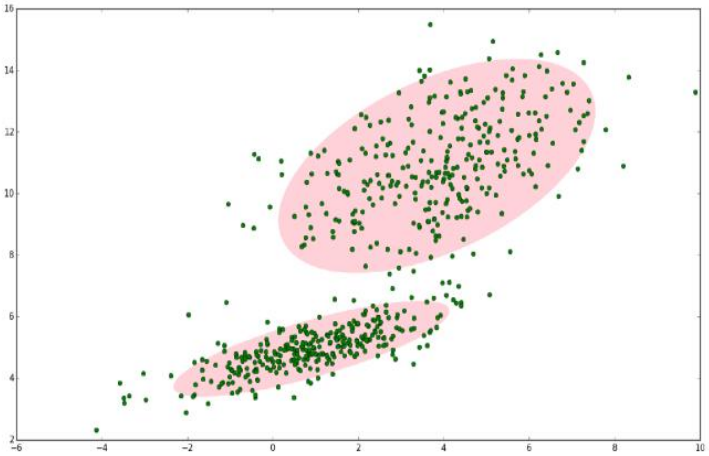


Fig. 2. Combination of two Gaussian Distributions

2.3.1 GMM algorithm: GMM probability density function.

$$\mathrm{p}(x)=\sum_{k=1}^{\mathrm{K}} p(k)\, p(x|\ k)=\sum_{k=1}^{K} \pi_k \mathrm{N}\left(\mathrm{x}|\ \mu_k, \sum k_k\right) \tag{8}$$

$N(x \mid \mu_k, \sum_k)$ is called the k^{th} component.

$$\sum_{i=1}^{K} \pi_i = 1 \tag{9}$$

and μ_k is the mixture coefficient actually. The μ_k is the weight of each component:

$$N(x \mid \mu_k, \sum\ \ k) \tag{10}$$

This function (9) can be explained: assuming there is a set of data X = *{*X_1, X_2, ..., X_n*}*, and X_i are formed by Gaussian distribution (there are K Gaussian distribution generators) with unknown generator and unknown proportion π_k of each generator in a mixture model.

Because we do not know $\pi_{\mathrm{k}}, \mu_k, \sum_k$, so we need to estimate these parameters at first. The maximum likelihood method is to maximize the probability value of the sample point on the estimated probability density function. In order to prevent the overflow phenomenon in the calculation process, we can take the logarithm of the objective function to calculate:

$$\max \sum_{i=1}^{N} \log p\left(x_i\right) \tag{11}$$

so the maximum logarithmic likelihood function is:

$$\max \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k N\left(x_i | \ \mu_k, \sigma_k\right) \right) \quad (12)$$

The most common algorithm that we use to estimate parameters in the Gaussian mixed model is EM.

EM algorithm. EM algorithm [19] has two steps. The first step, assuming we know the value of each parameter in each Gaussian model (initialize it or use the previous iteration result), is to estimate the weight of each Gaussian model. The second step is to ensure the parameters in the Gaussian model based on the estimated weight. Then, the algorithm will repeat two steps until the parameters reach stable and reach the "end value" (the optimal value).

- The first step, for the i^{th} sample X_i, the probability generated by the k^{th} model is:

$$w_i(k) = \frac{\pi_k N\left(x_i | \ \mu_k, \sigma_k\right)}{\sum_{j=1}^{K} \pi_j N\left(x_i | \ \mu_j, \sigma_j\right)} \quad (13)$$

We use the maximum likelihood estimation (MLE) to estimate the parameters in Kth Gaussian model.

$$\mu_k = \frac{1}{N} \sum_{i=1}^{N} w_i(k) x_i \quad (14)$$

$$\sigma_k = \frac{1}{N} \sum_{i=1}^{N} w_i(k) \left(x_i - \mu_k\right) \left(x_i - \mu_k\right)^T \quad (15)$$

$$N_k = \sum_{i=1}^{N} w_i(k) \quad (16)$$

Repeating the above two steps until the function converges. Here is the general converging process Figure.3.

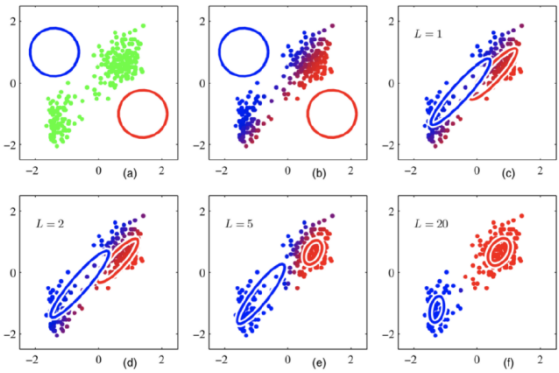


Fig. 3. GMM data processing

2.3.2 Advantages and disadvantages of GMM.

- Advantages

— GMM is the fastest mixture model algorithm.
— Mathematical properties and computation performance is excellent.
— GMM can simulate the distribution of any variable in the model and it is easy to extend to unsupervised learning.

- Disadvantages

— EM converges well but doesn't promise to find the global maximum value, even though it may reach the local maximum value. Solution: do iteration with different initialized parameters and keep the best performance.
— GMM performs unsatisfactorily at high-dimensional data. Especially, it is hard to estimate covariance by insufficient samples.
— GMM performs unsatisfactorily at high-dimensional data. Especially, it is hard to estimate covariance by insufficient samples.

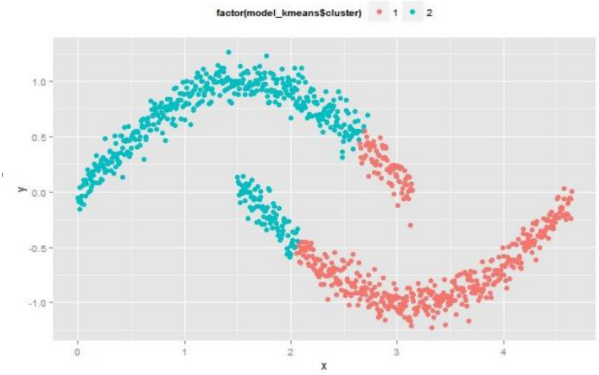
2.3.3 Applications of GMM.

GMM was used for data analysis. It was used in the unsupervised adaptation of the Brain-Computer interface [20], and the result showed that there was a lower error rate compared with the other two unsupervised methods. Another application [21] was to identify hydrological characteristics and features in the Southern Ocean such as temperature and salinity of the ocean, and, finally, get temperature profiles classification and current circulation without geographical data: latitude or longitude.

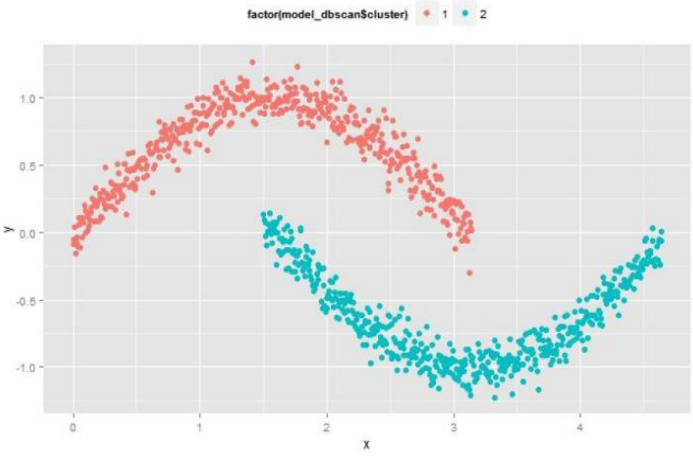
2.4 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is one of the earliest (1996) algorithms. The main thought of DBSCAN is to measure the density of

the space. As long as the density of points in a region is greater than a certain threshold value, it is added to a similar cluster. It can be used to find out oddly-shaped clusters as Figure.4, and we do not have to ensure the number of clusters previously. Thus, it is an efficient way to deal with large scale of data sets.



(a) K-means clustering



(b) DBSCAN clustering

Fig. 4. Comparing of 2 types of clustering

2.4.1 DBSCAN model.

We should know the following definitions and mathematics symbols: Neighborhood, density, and three different kinds of points in this model (assuming our sample set is $D = (x_1, x_2, \ldots, x_m)$)

- $\in$ (neighborhood)

For $x_j \in D, N_\varepsilon(x_j) = \{x_i \in D \mid \text{distance}(x_i, x_j) \leq\}$.

The number of subsets is recorded as $\left|N_\varepsilon\left(x_j\right)\right|$

- Core objects

For a randomly sample in $x_j \in D$, if $|N_\varepsilon(x_i)| \geq$ MinPts (MinPts is the minimum number of samples in subset Nε(xj)), xj is a core object.

- Directly density-reachable

If x_i is in the neighborhood $\in$ of x_j and x_j is the core object, we call x_i directly density-reachable from x_j .

- Density-reachable

For x_i and x_j, if there is sample sequence $p_1, p_2, ..., p_T$, and it satisfies $p_1 = x_i$, $p_T = x_j$, and $p_t + 1$ is directly density-reachable from p_t, x_j is density-reachable from xi. Sequence $p_1, p_2, ..., p_T - 1$ are all core objects.

- Density-connected

For x_i and x_j, if there is a core object x_k that lets x_i and x_j density-reachable from x_k, so we call x_i and x_j are density-connected.

- Core point

Assuming $x \in X$, if $\rho(x) \geq M$, (Minimum number of points required to form a cluster), we call x the core point of X. The set of all the core points is called X_c, and $X_{nc} = X \backslash X_c$ represents the set of all the non-core points as Figure.5.

- Border point

If$x \in X_{nc}$, and $\exists y \in X$ satisfy $y \in N_\epsilon(x) \cap X_c$, it means there is a core point in x neighborhood so x is called the boundary point of X. We call the set of all border points X_{bd} as Figure5.

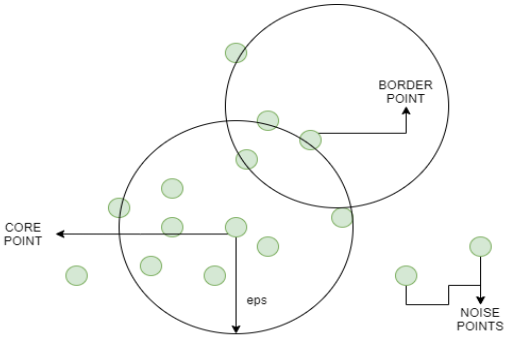


Fig. 5. DBSCAN Points

- Noise point

If $x \in X_{\text{noi}}$, $X_{\text{noi}} = X \setminus (X_c \cup X_{bd})$.Thus, x points are noise points as Figure5.

2.4.2 DBSCAN algorithm.

The main thought of the DBSCAN algorithm is: Starting from a selected core point, the region expands to the region of reachable density continuously, so as to obtain a maximized region containing core points and boundary points. The objection of this algorithm is to divide a set of data into K clusters and noise points. (Assuming that a set of data is $X = x^{(1)}, x^{(2)}, ..., x^{(N)}$) Thus, we introduce a cluster to sign a set of data.

$$m_i = \begin{cases} j(j > 0)[1] \\ -1[2] \end{cases} \quad (17)$$

For the equation [1], it will be used when $x(i)$ belongs to the J^{th} cluster.

For the equation [2], it will be used when $x(i)$ is noise point.

Thus, the outcome will be a signed array, m_i, $i = 1, 2, ..., N$.

Process of DBSCAN algorithm

Here is the conclusion of the DBSCAN algorithm.

Input: Sample set $x_1, x_2, ..., x_m$, neighborhood parameters $(\in, MinPts)$ and sample distance measurement method.

Output: A clump C partition

- 1) Initializing the core object set $\Omega = \phi$, clump numbers k=0, and cluster partition $C = \Phi$. Also, we have to initialize the collection of unaccessed samples $\Gamma = D$.
- 2) For $j = 1, 2, ..., m$, finding all core objects by following steps

— By using distance measurement method, we can find subsample set $\left|N_\varepsilon\left(x_j\right)\right|$ of x_j.

— If the number of samples in subset satisfies $\left|N_\varepsilon\left(x_j\right)\right| \geq MinPts$, we combine x_j into the core sample set: $\Omega = \Omega \cup \{x_j\}$.

- 3) If the core sample set $\Omega = \phi$, algorithm ending. Else, the algorithm continues.
- 4) In core object set Ω, we randomly chose a core object 'o' and initialize the array of clump core objects $\Omega_{cur} = \{o\}$. In addition, we have to initialize serial numbers k = k + 1 and the current clump sample sets $C_k = \{o\}$.Then, we upload the unassessed sample sets $\Gamma = \Gamma - (o)$.
- 5) If $\Omega_{cur} = \{o\}$, we upload Clump C partition $C = \{C_1, C_2, \ldots, C_k\}$. And the core sample set $\Omega = \Omega - C_k$. Then, the algorithm goes step3. Else, we only have to upload core object set $C = \{C_1, C_2, \ldots, C_k\}$.

- 6) We pick up an object o from the array Ω_{cur} of core object and find out all subsets of objects $N_{\varepsilon}(o')$ by using neighborhood distance threshold ε. We let $\Delta = N_{\varepsilon}(o') \cap \Gamma$ and upload current sample set $C_k = C_k \cup \Delta$, unacessed sample sets $\Gamma = \Gamma - \Delta$, and $\Omega_{cur} = \Omega_{cur} \cup (\Delta \cap \Omega) - o'$ .Then, the algorithm goes to step5.

  Output: A clump C partition C = C1, C2, . . . , Ck.

2.4.3 Advantages and disadvantages of DBSCAN.

- Advantages

– Dense data sets of any shape can be clustered
– Outliers can be found while clustering, and are not sensitive to outliers in the data set. It is resistant to the outliers
– There is no bias in the clustering results. In contrast, the initial value of k-means clustering algorithm has a great influence on the clustering results.

- Disadvantages

– If the density of sample sets is not uniform and the clustering spacing difference is large, the clustering quality is poor.
– If the sample set is large, the clustering convergence time is long.
– It is very complicated to adjust the parameters, and the parameters have a great influence on the results.

2.4.4nApplications of DBSCAN.

DBSCA has been used for classification of Internet traffic, and DBSCAN algorithm demonstrates a better effectiveness and efficiency in processing large data set [22]. Another application is to use DBSCAN with Noise algorithm to differentiate normal and anomalous weather data though utilizing weather variables [23].

In addition, DBSCAN has been used for agriculture development. In this paper [24], DBSCAN is one of algorithms to obtain the optimal environmental conditions for growing of wheat to reach the highest productivity.

3 Dimensionalit reduction

Dimensionality reduction [7] has been used in various field of research or studying in dig data. Dimensionality reduction in machine learning refers to the use of a mapping method to map data points from a high-dimensional space to a low-dimensional space. At present, most dimensionality reduction algorithms deal with data expressed by vectors, and some algorithms deal with data expressed by higher- order tensors. The reason why the data representation after dimensionality reduction is used is that

the original high- dimensional space contains redundant information and noise information, which causes errors in practical applications such as image recognition and reduces accuracy. By reducing the dimension, we hope to reduce the error caused by redundant information and improve the accuracy of identification (or other applications). Or we hope to find the intrinsic structural features of data by dimensionality reduction algorithm. I will discuss PCA, LLE, and MDS and conclude their algorithms and applications on big data.

- Functions of dimensionality reduction

— Reduce the complexity of time and space
— It saves the cost of extracting unnecessary features
— Removing the noise from the data sets.
— Simpler models have stronger robustness on small data sets.
— When the data can be interpreted with fewer features, we can better interpret the data.
— Data visualization

- The purpose of dimensionality reduction

It is used for feature selection and feature extraction.

— Feature selection: select important feature subsets and delete other features;
— Feature extraction: fewer new features formed from the original features.

- Methods of dimensionality reduction as shown in Figure.6.

3.1 PCA

PCA (principle component analysis) [8] is widely used in data dimensionality reduction algorithms and unsupervised machine learning. The main idea of PCA is to map the N-dimensional features to the K-dimension, which is reconstructed based on the original N-dimensional features. PCA is to find a set of mutually orthogonal coordinate axes sequentially from the original space. The selection of the new coordinate axes is closely related to the data itself. Among them, the first new coordinate axis is selected in the direction of the largest variance in the original data, the second new coordinate axis is selected in the plane orthogonal to the first coordinate axis to make the largest variance, and the third axis is selected in the plane orthogonal to the first and second axes to make the largest variance. And so on, we get n of these axes. With the new axes obtained in this way, we find that most of the variance is contained in the first k axes, and the variance contained in the latter axis is almost zero. Thus, we can ignore the rest of the axes, leaving only the first k axes that contain most of the variance. This is equivalent to retaining only the dimension features containing most of the variance while ignoring the feature dimensions containing almost zero variance. Thus, dimension reduction is carried out on the data features.

3.1.1 PCA algorithms.

There are two main types of PCA algorithms.

- First, the PCA algorithm is based on the eigenvalue to decompose the covariance matrixes.

Input: data set $X = \{x_1, x_2, x_3, \ldots, x_n\}$ which needs to be reduced to k dimensions.

- De-averaging (i.e., decentralization), i.e. subtracting the average value of each feature.
- Calculate the covariance matrix.

Note: Dividing or not dividing the sample number n or n-1 here actually has no effect on the feature vector obtained.

- Find the eigenvalues and eigenvects of the covariance matrix by the eigenvalue decomposition method.
- Sort the eigenvalues from large to small, and select the largest k among them. Then the corresponding K feature vectors are used as row vectors respectively to form the feature vector matrix P.
- Transform the data into a new space constructed by K feature vectors, that is, Y=PX.

- Secondly, PCA is based on SVD to decompose the covariance matrixes

Input: data set $X = \{x_1, x_2, x_3, \ldots, x_n\}$. which needs to be reduced to k dimensions.

- De-averaging, that is, subtracting the average value of each feature.
- Calculate the covariance matrix.
- Calculate the eigenvalues and eigenvectors of the covariance matrix by SVD.
- Sort the eigenvalues from large to small, and select the largest k among them. Then the corresponding K eigenvectors are used as column vectors respectively to form the eigenvector matrix.
- Transform the data into a new space constructed by K feature vectors.

In PCA dimensionality reduction, we need to find the maximum k eigenvectors of the sample covariance matrix $\frac{1}{n}XX^T$, and then use the matrix composed of the maximum K eigenvectors to do low-dimensional projection dimensionality reduction. It can be seen that in this process, the covariance matrix $\frac{1}{n}XX^T$ needs to be worked out first. When the sample number is large and the sample feature number is large, the calculation is still large.

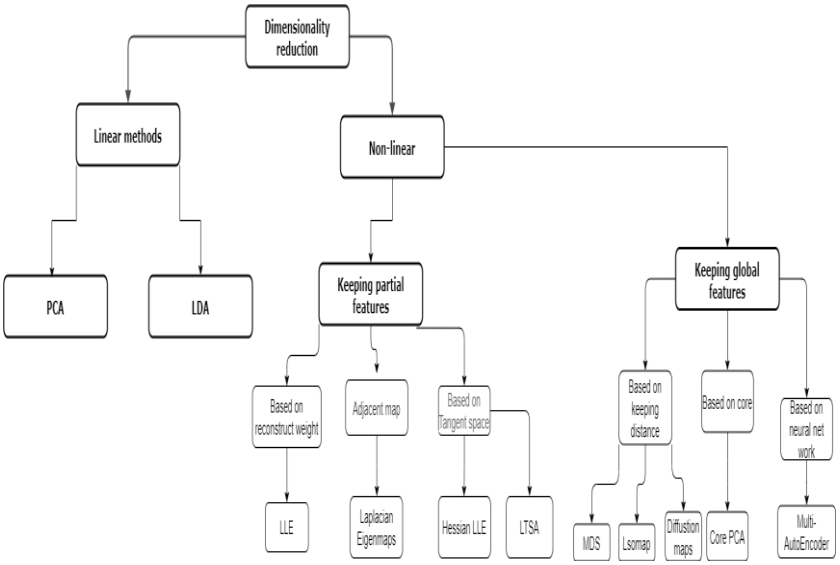


Fig. 6. Dimensionality reduction methods

3.1.2 Advantages and disadvantages of PCA.

- Advantages

— Data sets are easy to use.
— Algorithm calculation process is time-saving.
— It can remove noise points.
— It is easy to understand the outcome by visualization of outcomes.
— There is not any limitations on parameters.

- Disadvantages

— There are some limitations on eigenvalue decomposition.
— In the case of non-Gaussian distribution, the principal element obtained by the PCA method may not be optimal.

3.1.3 Applications of PC.

PCA was used to determine 123 imperative genes for COVID-19 progression including immune-related genes. The result was from comparing RNA expression profiles of 16 COVID-19 patients and 18 healthy control subjects from 60683 candidate probes [25]. Another application was that, in this paper [26], PCA was applied to consider and divide agricultural data for the purpose of determining optimal parameters to augment crop yield.

In addition, PCA was employed to develop the desired recognition system by using images of 10000 people from assorted racial origins and age range from 18 to 60 years old. The accuracies of face recognition were found very precise.

3.2 LLE

LLE [9] has been widely used in unsupervised machine learning in big data analysis. Its feature is non-linear, and it is one of the classical algorithms in manifold learning [27]. It tends to keep the partial characters in a sample so it is an important algorithm for image recognition and high-dimension data visualization and etc.

3.2.1 LEE conclusive algorithm.

As is shown in Figure 7, there are three main steps.

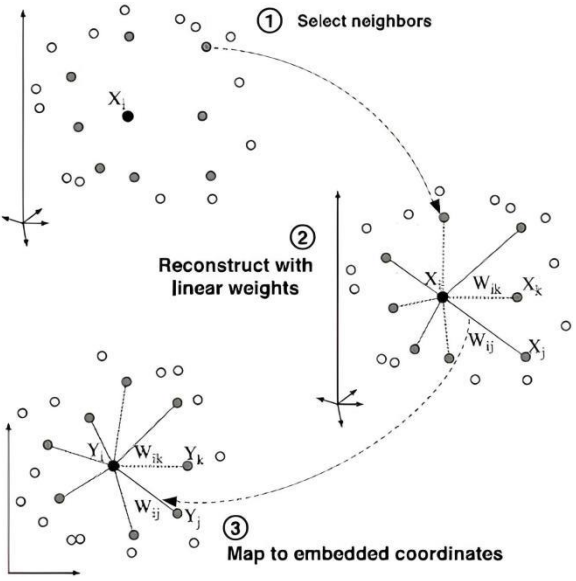


Fig. 7. LLE algorithm process

The first step is the process of k-nearest neighbor, this process uses the same method as the KNN algorithm [28] to find the nearest neighbor.

- The second step is to find the linear relationship of K neighbors of each sample in the neighborhood and get the weight coefficient W of the linear relationship assuming we have m numbers of samples with n dimension $x1, x2, ..., x_m$.
- The third step is to use weight coefficients to reconstruct sample data in low dimensions.

The specific processes are as follows:

Input: sample set $D = \{x_1, x_2, \ldots, x_m\}$ the nearest neighbor k, dimension number d being reduced Output: low dimensional sample set matrix D'

– For i 1 to m, calculating k nearest neighbors with x_i, ($x_{i1}, x_{i2}, ..., x_{ik}$), by measuring euclidean metric.

— For i 1 to m, obtaining the local covariance matrix $Z_i = \left(x_i - x_j\right)\left(x_i - x_j\right)^T$, and obtaining the corresponding weight coefficient vector:

$$W_i = \frac{Z_i^{-1}1_k}{1_k^T Z_i^{-1}1_k} \tag{18}$$

— The weight coefficient matrix W is composed of the weight coefficient vector-W_i and the calculation matrix $M = (I - W)(I - W)^T$
— Calculate the first d+1 eigenvalues of the matrix M, and calculate the eigenvectors corresponding to the d+1 eigenvalues $\{y_1, y_2, \ldots y_{d+1}\}$.
— The matrix spanned from the second feature vector to the d+1 feature vector is the output low-dimensional sample set matrix $D' = \{y_2, y_3, \ldots y_{d+1}\}$.

3.2.2 Advantages and disadvantages.

- Advantages

— You can learn locally linear low-dimensional manifolds of any dimension
— The algorithm means sparse matrix eigen decomposition, with relatively small computational complexity so it is easy to implement.

- Disadvantages

— The manifold learned by the algorithm can only be unclosed and the sample set is dense and uniform.
— The algorithm is sensitive to the selection of the nearest neighbor sample number, and different nearest-neighbor numbers have a great influence on the final dimensionality reduction result.

3.2.3 Applications of LLE.

LLE implemented dimensionality in hyperspectral images which was favorable for hyperspectral data classification [29] The image contained a spectrum with hundreds of dimensions that embodied many data. Also, for increasing feature selection efficiency and effectiveness [30], LLE was employed to strengthen teh relationship between UFS (unsupervised feature selection) [31] and the feature sub-space.

In addition, LLE improved the use of brain MRI to predict Alzheimer's disease (AD). They used LLe to decrease dimensions of multiple MRI data of regional brain volume and cortical thickness, and the LLE showed that it was an efficacious way through testing 413 individuals who had AD [32].

3.3 MDS

MDS (Multidimensional Scaling) [10] is one of the classical method of manifold learning in unsupervised machine learn- ing. MDS is a visualization method to display high dimensional multivariate data in low-dimensional space. The method looks similar to plotting with principal component scores or plotting with scores of two linear discriminants. The basic goal of multidimensional scaling is to "fit" the original data into a low-dimensional coordinate system so it can minimize any deformation caused by dimensionality reduction. In addition, there are commonly three kinds of MDS including Classical MDS, Metric MDS, and Non-metric MDS as shown in Figure 8.

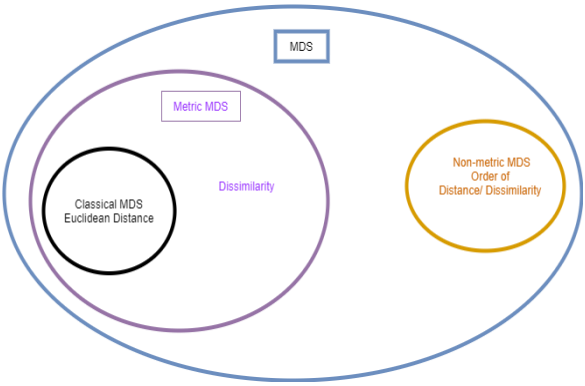


Fig. 8. Types of MDS

3.3.1 MDS conclusive algorithm.

Here are the algorithm steps below [10].

- Based on original data and Euclidean Distance to calculate distance matrix $D = \{d_{ij}\}_{m\times m}$.
- Obtaining four intermediate variables.

$$T_1 = d_{ij}{}^2, T_2 = \sum_{i=1}^{m} d^2{}_{ij}, T_3 = \sum_{j=1}^{m} d^2{}_{ij}, T_4 = \sum_{i=1}^{m}\sum_{j=1}^{m} d^2{}_{ij} \tag{19}$$

- Getting elements from matrix B by using function.

$$b_{ij} = -\frac{1}{2}T_1 + \frac{1}{m}T_2 + \frac{1}{m}T_3 - \frac{1}{2m^2}T_4 \tag{20}$$

- Obtaining B by using Eigendecomposition.

$$B = U\Lambda U^T = \left(\Lambda^{\frac{1}{2}}U^T\right)^T \left(\Lambda^{\frac{1}{2}}U^T\right) \tag{21}$$

- By the magnitude of the eigenvalues, obtaining Λ_d, U_d.

- The final MDS solution is $Z = \Lambda_{d'}^{\frac{1}{2}} U_{d'}^{T}$.

3.3.2 Advantages and disadvantages of MDS.

- Advantages

— No prior knowledge is required and the calculation is simple.
— The relative relationship of data in the original space is retained, and the visualization effect is better.

- Disadvantages

— If the user has some prior knowledge of the observed object and has mastered some characteristics of the data, but cannot intervene in the processing process through parameterization or other methods, the expected effect may not be achieved.
— It is believed that all dimensions have the same contribution to the goal, but in fact, some dimensions have little impact on the goal, while others have a relatively large impact on the goal.

3.3.3 Applications of MDS:

MDS was used to visualize geological features of differences and similarities in 16 fluvial and 5 aeolian sand samples. MDS successfully gained geological insights from big data [33].

In addition, an algorithm based on MDS was employed and it was called nMDS [34]. It measured the dissimilarity of the gene activities in the transcriptional response of cell-cycle-synchronized human fibroblasts to serum. They produced a circular comparative pattern of genes that was clear-cut. The large-scale data were from Microarray experiments [35].

4 Conclusion

Unsupervised machine learning techniques have drawn re- markable attention from data science to gain imperative infor- mation from large-scale data. Many algorithms were applied in different fields of study and research, and these algorithms have had varying degrees of success. This paper provides an overview of unsupervised machine learning algorithms with those advantages and disadvantages. The applications of each algorithm are discussed.

References

1. H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, “Unsupervised learning based on artificial neural network: A review,” in *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. IEEE, 2018, pp. 322–327.

2. R. Choudhary and H. K. Gianey, "Comprehensive review on supervised machine learning algorithms," in *2017 International Conference on Machine Learning and Data Science (MLDS)*. IEEE, 2017, pp. 37–43.
3. Y. Li and H. Wu, "A clustering method based on k-means algorithm," *Physics Procedia*, vol. 25, pp. 1104–1109, 2012.
4. F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.
5. D. A. Reynolds, "Gaussian mixture models." *Encyclopedia of biomet- rics*, vol. 741, no. 659-663, 2009.
6. K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, "Dbscan: Past, present and future," in *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*. IEEE, 2014, pp. 232–238.
7. X. Huang, L. Wu, and Y. Ye, "A review on dimensionality reduction techniques," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 10, p. 1950017, 2019.
8. T. Zhang and B. Yang, "Big data dimension reduction using pca," in *2016 IEEE international conference on smart cloud (SmartCloud)*. IEEE, 2016, pp. 152–157.
9. L. K. Saul and S. T. Roweis, "An introduction to locally linear embedding," *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html*, 2000.
10. M. C. Hout, M. H. Papesh, and S. D. Goldinger, "Multidimensional scaling," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 4, no. 1, pp. 93–103, 2013.
11. O. Nasraoui and C.-E. B. N'Cir, "Clustering methods for big data analytics," *Techniques, Toolboxes and Applications*, vol. 1, pp. 91–113, 2019.
12. K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algo- rithm," *IEEE access*, vol. 8, pp. 80 716–80 727, 2020.
13. G. Liu, J. Yang, Y. Hao, and Y. Zhang, "Big data-informed energy efficiency assessment of china industry sectors based on k-means clustering," *Journal of cleaner production*, vol. 183, pp. 304–314, 2018.
14. A. Deese, "Implementation of unsupervised k-means clustering algo- rithm within amazon web services lambda," in *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CC- GRID)*. IEEE, 2018, pp. 626–632.
15. A. Talasbek, A. Serek, M. Zhaparov, S.-M. Yoo, Y.-K. Kim, and G.-H. Jeong, "Personality classification by applying k-means clustering," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2020, pp. 421–426.
16. J. Rodr´ıguez, I. Semanjski, S. Gautama, N. Van de Weghe, and D. Ochoa, "Unsupervised hierarchical clustering approach for tourism market segmentation based on crowdsourced mobile phone data," *Sen- sors*, vol. 18, no. 9, p. 2972, 2018.
17. S. Rana and R. Garg, "Application of hierarchical clustering algorithm to evaluate students performance of an institute," in *2016 second in- ternational conference on computational intelligence & communication technology (CICT)*. IEEE, 2016, pp. 692–697.
18. Y. Akamine, Y. Ueda, Y. Ueno, K. Sofue, T. Murakami, M. Yoneyama, M. Obara, and M. Van Cauteren, "Application of hierarchical clustering to multi-parametric mr in prostate: Differentiation of tumor and normal tissue with high accuracy," *Magnetic Resonance Imaging*, vol. 74, pp. 90–95, 2020.
19. H. Watanabe, S. Muramatsu, and H. Kikuchi, "Interval calculation of em algorithm for gmm parameter estimation," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 2686–2689.

20. G. Liu, G. Huang, J. Meng, D. Zhang, and X. Zhu, “Improved gmm with parameter initialization for unsupervised adaptation of brain–computer interface,” *International Journal for Numerical Methods in Biomedical Engineering*, vol. 26, no. 6, pp. 681–691, 2010.
21. D. C. Jones, H. J. Holt, A. J. Meijers, and E. Shuckburgh, “Unsupervised clustering of southern ocean argo float temperature profiles,” *Journal of Geophysical Research: Oceans*, vol. 124, no. 1, pp. 390–402, 2019.
22. C. Yang, F. Wang, and B. Huang, “Internet traffic classification using dbscan,” in *2009 WASE International Conference on Information Engi- neering*, vol. 2. IEEE, 2009, pp. 163–166.
23. S. Wibisono, M. Anwar, A. Supriyanto, and I. Amin, “Multivariate weather anomaly detection using dbscan clustering algorithm,” in *Jour- nal of Physics: Conference Series*, vol. 1869, no. 1. IOP Publishing, 2021, p. 012077.
24. J. Majumdar, S. Naraseeyappa, and S. Ankalaki, “Analysis of agriculture data using data mining techniques: application of big data,” *Journal of Big data*, vol. 4, no. 1, pp. 1–15, 2017.
25. K. Fujisawa, M. Shimo, Y.-H. Taguchi, S. Ikematsu, and R. Miyata, “Pca-based unsupervised feature extraction for gene expression analysis of covid-19 patients,” *Scientific reports*, vol. 11, no. 1, pp. 1–11, 2021.
26. K. Badapanda, D. P. Mishra, and S. R. Salkuti, “Agriculture data visualization and analysis using data mining techniques: application of unsupervised machine learning,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 1, pp. 98–108, 2022.
27. A. J. Izenman, “Introduction to manifold learning,” *Wiley Interdisci- plinary Reviews: Computational Statistics*, vol. 4, no. 5, pp. 439–446, 2012.
28. G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “Knn model-based approach in classification,” in *OTM Confederated International Confer- ences” On the Move to Meaningful Internet Systems”*. Springer, 2003, pp. 986–996.
29. A. Ramirez and M. Rahnemoonfar, “Improved locally linear embedding for big-data classification,” in *Proceedings of the 6th ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data*, 2017, pp. 37–41.
30. J. Miao, T. Yang, L. Sun, X. Fei, L. Niu, and Y. Shi, “Graph regularized locally linear embedding for unsupervised feature selection,” *Pattern Recognition*, vol. 122, p. 108299, 2022.
31. S. Solorio-Ferna´ndez, J. A. Carrasco-Ochoa, and J. F. Mart´ınez-Trinidad, “A review of unsupervised feature selection methods,” *Artificial Intelli- gence Review*, vol. 53, no. 2, pp. 907–948, 2020.
32. X. Liu, D. Tosun, M. W. Weiner, N. Schuff, A. D. N. Initiative *et al.*, “Locally linear embedding (lle) for mri based alzheimer’s disease classification,” *Neuroimage*, vol. 83, pp. 148–157, 2013.
33. A. Jakaitiene, M. Sangiovanni, M. R. Guarracino, and P. M. Pardalos, “Multidimensional scaling for genomic data,” in *Advances in Stochastic and Deterministic Global Optimization*. Springer, 2016, pp. 129–139.
34. S. M. Holland, “Non-metric multidimensional scaling (mds),” *Depart- ment of Geology, University of Georgia, Athens, Tech. Rep. GA*, pp. 30 602–2501, 2008.
35. Y.-H. Taguchi and Y. Oono, “Relational patterns of gene expression via non-metric multidimensional scaling analysis,” *Bioinformatics*, vol. 21, no. 6, pp. 730–740, 2005.