Unsupervised Learning Algorithms

Unsupervised learning algorithms automatically discover hidden patterns and structures within unlabeled data without requiring pre-existing labels or targets.

1，K-Means Clustering

K-Means Clustering is a common unsupervised learning algorithm used to partition a dataset into K distinct clusters, ensuring that each data point belongs to the cluster with the nearest centroid.

The basic principle of K-Means clustering involves an iterative optimization process. It allocates data points to clusters and updates the cluster centroids to minimize the sum of squared distances (referred to as "inertia" or "within-cluster sum of squares") between each data point and its assigned cluster centroid.

Basic Principles:

1. Random Initialization: Initially, K data points are randomly chosen to serve as the initial cluster centroids.
2. Assign Data Points: For each data point, calculate its distance to each cluster centroid and assign it to the cluster with the nearest centroid.
3. Update Cluster Centroids: For each cluster, compute the average of all data points within the cluster to define the new centroid.
4. Iterative Process: Repeat steps 2 and 3 until the cluster centroids no longer significantly change or until reaching a predetermined number of iterations.

The result of K-Means clustering is the partitioning of data points into K clusters, each represented by a central point. This algorithm typically requires multiple runs and iterations with different initial cluster centroid selections to find the global optimum solution. K-Means clustering is a simple yet effective clustering algorithm, but it is sensitive to the initial selection of cluster centroids and requires the specification of the value for K.

2，Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a widely used dimensionality reduction technique and data analysis method employed to reduce the dimensions of a dataset while retaining its essential information. The fundamental principle of PCA involves projecting high-dimensional data into a lower-dimensional subspace through linear transformation to find the directions of maximum variance, known as principal components.

Basic Principles:

Data Centering: Initially, the original data is standardized by centering it—subtracting the mean of each feature from the corresponding feature values in each data point—to ensure a mean of zero for the data.

Covariance Matrix: Subsequently, the covariance matrix of the data is calculated. This matrix represents the relationships between different features.

Eigendecomposition: The covariance matrix undergoes eigendecomposition to find its eigenvalues and eigenvectors.

Selection of Principal Components: The eigenvectors with the highest eigenvalues are chosen as principal components. These vectors form the new coordinate axes in the lower-dimensional subspace where the data is represented.

Projection: The original data is projected onto the selected principal components, achieving the data's dimensionality reduction.

Through PCA, high-dimensional data can be mapped to a lower-dimensional space, reducing the dimensionality of the data. This aids in data visualization, eliminating redundant features, accelerating the training of machine learning models, and enhancing model generalization performance. Selecting an appropriate dimension for dimensionality reduction (number of principal components) is a key parameter in PCA, often requiring adjustments based on the problem and performance requirements.

3，Gaussian Mixture Model

The Gaussian Mixture Model (GMM) is a probabilistic model used to model data distributions by combining multiple Gaussian distributions. It finds wide applications across various domains, including clustering, density estimation, anomaly detection, and more. The basic principle of GMM assumes that the data is composed of a mixture of several Gaussian distributions, each Gaussian distribution being referred to as a component. Here are the basic principles of GMM:

Basic Principles:

1. Linear Combination of Gaussian Distributions: GMM assumes that data is a combination of K Gaussian distributions, where K is the user-defined number of components. Each Gaussian distribution is referred to as a component, representing the probability density of data in different regions.
2. Latent Variables: Each data point is associated with a latent variable, representing which component the data point belongs to. Typically, this latent variable is a discrete variable with values ranging from 1 to K.
3. Data Generation Process: The process of generating a data point is as follows:

- Firstly, select a component (value of the latent variable) to determine which Gaussian distribution the data point belongs to.
- Then, generate the specific data point according to the chosen component's Gaussian distribution.

4. Parameter Estimation: Techniques such as maximum likelihood estimation are used to estimate the mean, variance, and mixing coefficients of each component to fit the data.

The GMM is commonly used in cluster analysis, where each component can represent a cluster, and the latent variable of each data point indicates its belonging to a specific cluster. Additionally, GMM is applicable in various domains such as density estimation, generative modeling, and anomaly detection.

4. **Hierarchical Clustering** is a clustering algorithm based on data similarity, creating a hierarchical cluster structure by iteratively merging or dividing data points.

Hierarchical clustering comprises two main types: Agglomerative and Divisive. Agglomerative clustering begins with each data point as an individual cluster, progressively merging similar clusters until all data points belong to one large cluster. Divisive clustering starts with all data points belonging to a single large cluster and gradually separates clusters until each data point forms its own individual cluster.

Basic Principles:

Initial State: Each data point is initially considered as a separate cluster, creating an initial set of N clusters, where N is the number of data points.

Similarity Measurement: Calculate the similarity between each pair of clusters, often using distance metrics (such as Euclidean distance, Manhattan distance, correlation, etc.) to measure the dissimilarities between data points within clusters.

Merge Most Similar Clusters: Identify the two most similar clusters and merge them into a new cluster. Rules for merging commonly include single linkage, complete linkage, average linkage, and others.

Repeat Merging: Repeat step 3, continuing to merge the most similar clusters until only one cluster remains, or until a predetermined stopping condition is met (such as a specific number of clusters).

Merge Rules: The choice of rules for merging two clusters significantly impacts the clustering results. For instance, under single linkage, the distance between two clusters is typically defined by the distance between the closest points from each cluster. In contrast, complete linkage computes the distance between the farthest points of two clusters.

Dendrogram: The result of hierarchical clustering is often illustrated as a dendrogram, where each node in the tree represents a cluster or a merged cluster, and the leaf nodes represent individual data points. Dendrograms can assist in determining the appropriate number of clusters.

Advantages of hierarchical clustering include the absence of a need to predefine the number of clusters, enabling the representation of data similarities hierarchically. However, it tends to have relatively higher computational complexity. The critical issues in hierarchical clustering involve selecting suitable similarity measures, merge rules, and determining the number of clusters.

5，Autoencoders

Autoencoders are a type of neural network architecture used for unsupervised learning and feature learning. The fundamental principle involves learning a compact representation (encoding) of data, then reconstructing the original data from that encoding, allowing for key feature extraction, dimensionality reduction, denoising, and new data generation. Autoencoders typically consist of two parts: an Encoder and a Decoder.

Basic Principles:

- Encoder: The Encoder maps input data into a lower-dimensional encoding space, typically achieved through a series of neural network layers. The objective of the Encoder is to compress the input data into a compact representation, capturing the key features of the data.

- Decoder: The Decoder maps the encoded representation back to the original data space, usually realized through a series of neural network layers. The goal of the Decoder is to reconstruct an output that closely resembles the original data from the encoding.
- Reconstruction Loss: The training process of an autoencoder involves minimizing the Reconstruction Loss between the input data and the output from the Decoder. This is typically done using Mean Squared Error (MSE) as the loss function. The Reconstruction Loss measures the difference between the reconstructed data and the original data, driving the autoencoder to learn significant data features.
- Dimensionality Reduction and Feature Learning: Autoencoders can be used for data dimensionality reduction by decreasing the dimensionality of the encoding, thereby extracting the crucial features of the data. The hidden layer representation of the Encoder can be considered as a compressed representation of the data.

The training objective of an autoencoder is to minimize reconstruction loss by adjusting the parameters of both the Encoder and the Decoder using backpropagation. This process aims to generate better encoding representations and reconstructed data. Post-training, the hidden layer representation of the Encoder typically serves as a feature representation of the data.