**Slide 1: title**

Hi everyone, my name is tingting, I want to talk about clustering algorithms.

**Slide 2: Outlines**

Here is the Outlines today.

**Slide 3: Research goals**

I want to use the DBSCAN algorithm to classify earthquake-prone areas and create a seismic [/ˈsaɪzmɪk/](cmd://Speak/_us_/seismic) hazard index map, providing vital [/ˈvaɪtl/](cmd://Speak/_us_/vital) information to raise public awareness of seismic[/ˈsaɪzmɪk/](cmd://Speak/_us_/seismic) risks.

**Slide 4: What is clustering?**

Clustering is an unsupervised machine learning technique that groups similar data points into clusters based on shared characteristics, aiming to uncover underlying patterns in the data without prior [/ˈpraɪ ər/](cmd://Speak/_us_/prior) knowledge of labels or categories.

**Slide5: Common clustering algorithms**

Common clustering algorithms include K-means, which partitions data into clusters by minimizing within-cluster variance [/ˈveriəns/](cmd://Speak/_us_/variance), and hierarchical clustering, which builds a dendrogram of clusters Gaussian [/'ɡausiən/](cmd://Speak/_us_/gaussian) Mixture Models assume that data originates [/əˈrɪdʒɪneɪt/](cmd://Speak/_us_/originate) from multiple Gaussian/'ɡausiən/ distributions and assign probabilities [/ˌprɑːbəˈbɪləti/](cmd://Speak/_us_/probability) for each data point belonging to a cluster

**Slide6: Other clustering algorithms**

Here are other common algorithms, such as density-based, Grid-based, and model-based algorithm and so on.

**Slide7: Applications of clustering**

Clustering algorithms are widely used across various fields, such as marketing for customer segmentation based on behavior or demographics, and in natural language processing for organizing text data. They also play a role in image segmentation for object recognition and anomaly detection in finance and cybersecurity to identify unusual patterns that may indicate fraud [/frɔːd/](cmd://Speak/_us_/fraud) or security threats.

**Slide8: Density-based algorithm**

In this presentation, I will discuss density-based algorithms, specifically DBSCAN, which groups data points based on density. It identifies clusters by finding neighborhoods where the data point density exceeds a specified threshold [/ˈθreʃhoʊld/](cmd://Speak/_us_/threshold). Key concepts include core points, border points and noise points.

**Slide9: Key Characteristics**

Density-based algorithm has two key characteristics:

**Epsilon** [/ˈɛpsəˌlɑn/](cmd://Speak/_us_/epsilon) defines the range within which the algorithm searches for neighboring points to form clusters. For example, if Epsilon is set to 0.5, points within a 0.5 unit distance from each other are considered neighbors.

**Min Points** represents the minimum number of points that must exist within the Epsilon range to form a cluster, including the core point. For example, if min points is set to 5, a core point must have at least 4 neighbor points within the Epsilon distance to be part of a cluster.

**Slide10: Application-Earthquake spread location**

Earthquakes pose a continuous global threat, leading to economic losses and casualties [/'kæ ʒjʊə lis/](cmd://Speak/_us_/CASUALTIES). Over 20 earthquakes were recorded worldwide in the past 24 hours, and in 2024, the US experienced more than 70,000 earthquakes. Creating earthquake hazard index maps can help the public and government better understand high-risk areas, allowing for proactive measures.

**Slide11: Dataset**

BMKG online database providing data and information about earthquakes. I downloaded all earthquake information that occurred in 2024, including earthquake time, longitude, latitude, depth, and magnitude [/ˈmæɡnɪtuːd/](cmd://Speak/_us_/magnitude). I analysis these data to see the characteristics of each set of earthquakes.

**Slide12: Challenges and solutions**

Earthquake distribution is complex and irregular, often impacted by false alarms and poor monitoring. To address these issues, the DBSCAN algorithm is used for its efficiency with large spatial datasets and ability to handle noise and outliers. Density-reachable analysis helps cluster points while identifying noise. Additionally, the silhouette [/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) coefficient [/ˌkolɪ'fɪʃnt/](cmd://Speak/_us_/coefficient) evaluates model performance under varying parameters, ensuring effective clustering.

Slide 13: Advantages of DBSCAN

Slide 14: Disadvantage of DBSCAN

**Slide 15: Implementation: eps-neighborhood of a point**

Epsilon[/ˈɛpsəˌlɑn/](cmd://Speak/_us_/epsilon) is a very important parameter. Finding the core point and its neighbor points is the basis for us to get a cluster.

From the figure we can see that the yellow point p is a core point. Assuming the min point parameter is set to 3, we can see that all the blue points p1p2p3 are border points, that is, neighbors of point p.

This formula is that Epsilon neighborhood of a point defines the neighborhood around a data point.

When we calculate the distance from point p to point P1, we use Euclidean distance. It is a commonly used metric for calculating the straight-line distance between two points in Euclidean space. It is

The Euclidean distance d is given by this formula.

**Slide 16: Implementation – density-reachable**

After identifying core points and their neighbors, we classify which points belong to the same cluster while excluding noise. For example, points m, p, o, and r are core points, and q is density-reachable from m and indirectly from p. However, p cannot reach q since q is not a core point. Density-connected points, like r and s, have a mutual [/ˈmjuːtʃuəl/](cmd://Speak/_us_/mutual) relationship, meaning if one is reachable from the other through core points, the connection is symmetric [/si'metrik/](cmd://Speak/_us_/symmetric).

**Slide 17: Implementation - psecode**

Here is a flowchart and pseudocode/sudocode/.

The logic is that the algorithm identifies clusters by selecting random points, checking if they are core points, border points , or noise points, and repeat grouping core points into clusters based on Epsilon and Min points until all points are visited.

**Slide 18: Evaluation**

We use the silhouette [/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) coefficient [/ˌkolɪ'fɪʃnt/](cmd://Speak/_us_/coefficient) to evaluate the performance. Here is the formula.

a(i) is the average distance to all other points in the same cluster

b(i) is the average distance of all points in the nearest [/'niərist/](cmd://Speak/_us_/nearest) different clusters.

The score ranges from negative 1 to positive 1, where negative 1 indicates incorrect clustering and positive1 represents highly dense and well-separated clustering.

Let’s look at an example:

a(x1) is obtained by taking the average distance between two points in the same cluster C1, where the distance is calculated using Euclidean Distance.

b(x1) is obtained by calculating the average distance of all near clusters, C2 and C3, and then using the smallest value.

Next, we can use the formula to get the silhouette [/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) score.

**Slide 19: Optimization**

Adjust different parameters and calculate the corresponding silhouette[/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) coefficient [/ˌkolɪ'fɪʃnt/](cmd://Speak/_us_/coefficient), and determine the best parameters based on the calculation results.

There are two evaluation criteria [/kraɪˈtɪrɪə/](cmd://Speak/_us_/criteria) for best clustering results.

It is because the silhouette[/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) coefficient[/ˌkolɪ'fɪʃnt/](cmd://Speak/_us_/coefficient) is close to 1, which means the density of clustering is higher. And larger number of clusters that are formed, the more areas that are prone to earthquakes are detected.

**Slide 20: Conclusion**

DBSCAN effectively handles complex clusters and noise in earthquake distribution analysis, but challenges like uneven density and the need to incorporate various attributes beyond spatial data require further exploration for improved clustering results.

**Slide 20: References**

Here are my reference materials.

Any questions?

**-----------------------------------------**

**Slide 1: title**

Hi everyone, my name is tingting, I want to talk about clustering algorithms.

**Slide 2: Outlines**

Here is the Outlines today. First I will show the research goals. And then I will show some concepts of clustering. And then I will demonstrate a real-word application, challenges, solutions. Finally, my conclusion and reference materials.

**Slide 3: Research goals**

Through the density-based spatial [/ˈspeɪʃl/](cmd://Speak/_us_/spatial) clustering of noise (DBSCAN) algorithm , the regions were classified based on the density of earthquake-prone areas, hoping that the earthquake-prone area grouping information could serve as a form of earthquake hazard [/ˈhæzərd/](cmd://Speak/_us_/hazard) mitigation [/ˌmɪtɪ'ɡeʃən/](cmd://Speak/_us_/mitigation) and minimize the losses [/lɔːses/](cmd://Speak/_us_/loss) caused by earthquakes.

I want to use the DBSCAN algorithm to classify earthquake-prone areas and create a seismic [/ˈsaɪzmɪk/](cmd://Speak/_us_/seismic) hazard index map, providing vital [/ˈvaɪtl/](cmd://Speak/_us_/vital) information to raise public awareness of seismic[/ˈsaɪzmɪk/](cmd://Speak/_us_/seismic) risks.

**Slide 4: What is clustering?**

I think everyone of us has known or heard of clustering algorithms.

Clustering is an unsupervised [/ˌʌn'sju:pəvaizd/](cmd://Speak/_us_/unsupervised) machine learning technique for grouping similar data points into clusters, where data points in the same cluster share more characteristics than those in other clusters.

The meaning of unsupervised and supervised…

The goal of clustering is to identify underlying[/ˌʌndərˈlaɪɪŋ/](cmd://Speak/_us_/underlying) patterns or groupings in data without prior [/ˈpraɪər/](cmd://Speak/_us_/prior) knowledge of labels or categories.

Clustering is an unsupervised machine learning technique that groups similar data points into clusters based on shared characteristics, aiming to uncover underlying patterns in the data without prior [/ˈpraɪ ər/](cmd://Speak/_us_/prior) knowledge of labels or categories.

**Slide5: Common clustering algorithms**

Here are some common clustering algorithms:

* I believe everyone must have heard of the **K-means** algorithm, because it is so widely used. The principle of k-means algorithm is that partitions the data into k clusters based on minimizing within-cluster variance [/ˈveriəns/](cmd://Speak/_us_/variance)
* **Hierarchical clustering** is also a very useful algorithm that builds a tree of clusters, term as dendrogram [/'dendrəgræm/](cmd://Speak/_us_/dendrogram), and merges or splits them at each step. There are mainly two types of Hierarchical clustering: Agglomerative [/ə'ɡlɔmərətiv/](cmd://Speak/_us_/agglomerative) hierarchical clustering and Divisive Hierarchical clustering.

**Agglomerative** [/ə'ɡlɔmərətiv/](cmd://Speak/_us_/agglomerative) is a bottom to up approach, it merges the closest pair of clusters and repeat this step until only a single cluster is left .

**Divisive** is the opposite, it is a top to down approach, it split the farthest point in the cluster and repeat this process until each cluster only contains a single point.

* **Gaussian** [**/'ɡausiən/**](cmd://Speak/_us_/gaussian) **Mixture Models** Assumes that the data is generated from a mixture of several Gaussian distributions and assigns probabilities for each data point belonging to a cluster.

Common clustering algorithms include K-means, which partitions data into clusters by minimizing within-cluster variance [/ˈveriəns/](cmd://Speak/_us_/variance), and hierarchical clustering, which builds a dendrogram of clusters through two approaches: agglomerative (merging clusters) and divisive (splitting clusters). Gaussian [/'ɡausiən/](cmd://Speak/_us_/gaussian) Mixture Models assume that data originates [/əˈrɪdʒɪneɪt/](cmd://Speak/_us_/originate) from multiple Gaussian/'ɡausiən/ distributions and assign probabilities [/ˌprɑːbəˈbɪləti/](cmd://Speak/_us_/probability) for each data point belonging to a cluster

**Slide6: Other clustering algorithms**

Here are other common algorithms, such as density-based, Grid-based, and model-based algorithm and so on.

**Slide7: Applications of clustering**

Currently, many fields use clustering algorithms, such as marketing, healthcare, finance, biology, education, social science, image and video processing, manufacturing and so on.

In **marketing**, customer segmentation involves grouping customers based on common characteristics, such as purchasing behavior or demographics [/ˌdɛməˈɡræfɪks/](cmd://Speak/_us_/demographics), allowing businesses to tailor [/ˈteɪlər/](cmd://Speak/_us_/tailor) their products, services, and communication strategies to better meet the needs of each segment.

In natural language processing, **document or text clustering** helps in organizing large sets of textual data by grouping similar documents together, enabling efficient information retrieval [/rɪ'trivl/](cmd://Speak/_us_/retrieval), topic discovery, and content recommendation.

**Image segmentation** in computer vision is the process of partitioning an image into meaningful regions or objects, facilitating [/fəˈsɪlɪteɪting/](cmd://Speak/_us_/facilitate) applications such as object recognition, scene understanding, and medical imaging analysis.

**Anomaly** [**/əˈnɑːməli/**](cmd://Speak/_us_/anomaly) **detection** is a technique used across different domains, such as finance, healthcare, and cybersecurity, to identify unusual patterns or outliers in data that could indicate potential fraud [/frɔːd/](cmd://Speak/_us_/fraud), system failures, or security threats [/θret/](cmd://Speak/_us_/threats).

Clustering algorithms are widely used across various fields, such as marketing for customer segmentation based on behavior or demographics, and in natural language processing for organizing text data. They also play a role in image segmentation for object recognition and anomaly detection in finance and cybersecurity to identify unusual patterns that may indicate fraud or security threats.

**Slide8: Density-based algorithm**

In this presentation, I will describe more density based algorithm.

DBSCAN is a popular **density-based clustering algorithm** used to group data points based on density. It works by finding neighborhoods where the density of data points exceeds [/ɪkˈsiːd/](cmd://Speak/_us_/exceed) a specified [/ˈspɛsəˌfaɪd/](cmd://Speak/_us_/specified) threshold [/ˈθreʃhoʊld/](cmd://Speak/_us_/threshold). This allows it to automatically detect clusters in data, even with varying shapes and sizes.

we need to understand three basic concepts: core point, border point and noise point.

Core Point is a point that has at least min points within its epsilon [/ˈɛpsəˌlɑn/](cmd://Speak/_us_/epsilon) neighborhood.

Border Point is a point that is within the epsilon neighborhood of a core point but is not a core point itself.

Noise Point is neither a core point nor a border point.

In this presentation, I will discuss density-based algorithms, specifically DBSCAN, which groups data points based on density. It identifies clusters by finding neighborhoods where the data point density exceeds a specified threshold [/ˈθreʃhoʊld/](cmd://Speak/_us_/threshold). Key concepts include core points, border points and noise points.

* **Core Point**: Has at least a minimum number of points within its epsilon neighborhood.
* **Border Point**: Located within the epsilon neighborhood of a core point but not a core point itself.
* **Noise Point**: Neither a core nor border point.

**Slide9: Key Characteristics**

Density-based algorithm has two key characteristics:

**Epsilon** [/ˈɛpsəˌlɑn/](cmd://Speak/_us_/epsilon) defines the range within which the algorithm searches for neighboring points to form clusters. For example, if Epsilon is set to 0.5, points within a 0.5 unit distance from each other are considered neighbors.

**Min Points** represents the minimum number of points that must exist within the Epsilon range to form a cluster, including the core point. For example, if min points is set to 5, a core point must have at least 4 neighbor points within the Epsilon distance to be part of a cluster.

**Slide10: Application-Earthquake spread location**

Next I want to describe a real-world application, earthquake spread location. Earthquakes occur constantly, with over 20 recorded worldwide in the past 24 hours. In 2024, the number of earthquakes in the US over 70 thousands, causing significant economic losses and potential casualties [/'kæʒjʊəlti/](cmd://Speak/_us_/CASUALTIES). Thus [/ðʌs/](cmd://Speak/_us_/thus), creating earthquake hazard index maps can help the public and government better understand high-risk areas, enabling proactive measures. These maps provide visual data to identify risks and prioritize disaster preparedness [/prɪ'pɛrdnəs/](cmd://Speak/_us_/preparedness), such as constructing earthquake-resistant buildings, developing emergency plans, and promoting public education. They also assist in scientific research and policy-making, optimizing resource allocation [/ˌælə'keʃən/](cmd://Speak/_us_/allocation) and response strategies.

Earthquakes pose a continuous global threat, leading to economic losses and casualties [/'kæ ʒjʊə lis/](cmd://Speak/_us_/CASUALTIES). Over 20 earthquakes were recorded worldwide in the past 24 hours, and in 2024, the US experienced more than 70,000 earthquakes. Creating earthquake hazard index maps can help the public and government better understand high-risk areas, allowing for proactive measures.

**Slide11: Dataset**

BMKG online database providing data and information about earthquakes. I downloaded all earthquake information that occurred in 2024, including earthquake time, longitude, latitude, depth, and magnitude [/ˈmæɡnɪtuːd/](cmd://Speak/_us_/magnitude). I analysis these data to see the characteristics of each set of earthquakes.

**Slide12: Challenges and solutions**

The distribution of earthquakes is very complex and irregular [/ɪˈreɡjələr/](cmd://Speak/_us_/irregular), and some wrong data may be generated due to false alarms and improper monitoring methods. Therefore, how to deal with such a complex spatial distribution problem, how to set parameters, and how to eliminate data noise and outliers are the problems that should be solved first when analyzing earthquake spread location.

I use the DBSCAN algorithm to address the spatial complexity problem and the impact of noisy data because this algorithm can efficiently handle large-scale data sets, especially those with spatial information. Its time complexity is usually O(nlogn), and its performance is better than K-Means in many cases. Another advantage is that it can easily handle noise data and is not affected by outliers. Secondly, density-reachable is analyzed [/'ænl,aɪzd/](cmd://Speak/_us_/analyzed) to determine which points belong to the same cluster, and those points are identified as noise.

For parameter sensitive problem, this experiment also uses the silhouette [/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) coefficient [/ˌkolɪ'fɪʃnt/](cmd://Speak/_us_/coefficient) to evaluate the model performance under different parameters.

Earthquake distribution is complex and irregular, often impacted by false alarms and poor monitoring. To address these issues, the DBSCAN algorithm is used for its efficiency with large spatial datasets and ability to handle noise and outliers. Density-reachable analysis helps cluster points while identifying noise. Additionally, the silhouette [/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) coefficient [/ˌkolɪ'fɪʃnt/](cmd://Speak/_us_/coefficient) evaluates model performance under varying parameters, ensuring effective clustering.

Slide 13: Advantages of DBSCAN

Slide 14: Disadvantage of DBSCAN

**Slide 15: Implementation: eps-neighborhood of a point**

Epsilon[/ˈɛpsəˌlɑn/](cmd://Speak/_us_/epsilon) is a very important parameter. Finding the core point and its neighbor points is the basis for us to get a cluster.

From the figure we can see that the yellow point p is a core point. Assuming the min point parameter is set to 3, we can see that all the blue points p1p2p3 are border points, that is, neighbors of point p.

This formula is that Epsilon neighborhood of a point defines the neighborhood around a data point.

**NEps(P):** represents the Epsilon neighborhood of point P, which includes all points whose distance to point P does not exceed Epsilon.

**q ∈ D** : represents the point q in the dataset D

**dist(p, p1):** represents the distance between P and P1

**Eps is the** parameter represents the maximum neighborhood range between calculation points.

When we calculate the distance from point p to point P1, we use Euclidean distance. It is a commonly used metric for calculating the straight-line distance between two points in Euclidean space. It is widely used in various fields such as machine learning, computer vision, and clustering algorithms.

Let's understand **Euclidean Distance** through a simple example. There are two points in a two-dimensional [/dɪ'mɛnʃənl/](cmd://Speak/_us_/dimensional) space. The coordinates [/kəu'ɔ:dineits/](cmd://Speak/_us_/coordinates) of point P1 are X1 and Y1, and the coordinates of point P2 are X2 and Y2. The Euclidean distance d is given by this formula.

**Slide 16: Implementation – density-reachable**

After identifying a core point and its neighbors, we need to determine which points are in the same cluster. A cluster may contain many core points and border points, but we need to be careful to exclude noise data.

Let us understand this example:

Points 𝑚, 𝑝, 𝑜, 𝑟 are core objects because each one is in an Eps-neighborhood that contains at least three points. Object 𝑞 directly density-reachable from m. Object m is directly density-reachable from 𝑝 and vice versa. Object 𝑞 is indirectly density-reachable of 𝑝 because 𝑞 is directly density-reachable of m and m is directly density-reachable of 𝑝. However, 𝑝 is not density-reachable from 𝑞 because 𝑞 is not a core object. Similarly, *r* and 𝑠 are density-reachable of 𝑜 and 𝑜 is density-reachable of 𝑟. Therefore, 𝑜, 𝑟, and 𝑠 are all density-connected*.*

Directly and indirectly density-reachable is a directed and asymmetric[/ˌeɪsɪˈmetrɪk/](cmd://Speak/_us_/asymmetric) relationship which means one point can be reachable from another without the reverse being true. For instance, m can reach a neighboring point 1, but the q might not have enough neighbors to reach back to m.

density-connected is an undirected and symmetric relationship, means if two points are density-connected through a chain of core points, they are reachable. If point s is density-connected to point r, then point r is also density-connected to point s.

After identifying core points and their neighbors, we classify which points belong to the same cluster while excluding noise. For example, points m, p, o, and r are core points, and q is density-reachable from m and indirectly from p. However, p cannot reach q since q is not a core point. Density-connected points, like r and s, have a mutual [/ˈmjuːtʃuəl/](cmd://Speak/_us_/mutual) relationship, meaning if one is reachable from the other through core points, the connection is symmetric [/si'metrik/](cmd://Speak/_us_/symmetric).

**Slide 17: Implementation - psecode**

Here is a flowchart and pseudocode/sudocode/.

The input contains the earthquake dataset D and two parameters.

The logic is that the algorithm identifies clusters by selecting random points, checking if they are core points, border points , or noise points, and repeat grouping core points into clusters based on Epsilon and Min points until all points are visited.

**Slide 18: Evaluation**

We use the silhouette [/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) coefficient [/ˌkolɪ'fɪʃnt/](cmd://Speak/_us_/coefficient) to evaluate the performance. It is the maximum value of the average silhouette width for the entire data set.

Here is the formula.

a(i) is the average distance to all other points in the same cluster

b(i) is the average distance of all points in the nearest [/'niərist/](cmd://Speak/_us_/nearest) different clusters.

The score ranges from negative 1 to positive 1, where negative 1 indicates incorrect clustering and positive1 represents highly dense and well-separated clustering.

Let’s look at an example:

a(x1) is obtained by taking the average distance between two points in the same cluster C1, where the distance is calculated using Euclidean Distance.

b(x1) is obtained by calculating the average distance of all near clusters, C2 and C3, and then using the smallest value.

Next, we can use the formula to get the silhouette [/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) score.

**Slide 19: Optimization**

Adjust different parameters and calculate the corresponding silhouette[/ˌsɪlu'ɛt/](cmd://Speak/_us_/silhouette) coefficient [/ˌkolɪ'fɪʃnt/](cmd://Speak/_us_/coefficient), and determine the best parameters based on the calculation results.

There are two evaluation criteria [/kraɪˈtɪrɪə/](cmd://Speak/_us_/criteria) for best clustering results: one is to meet a higher silhouette coefficient, and the other is that more clusters are selected.

It is because the silhouette coefficient[/ˌkolɪ'fɪʃnt/](cmd://Speak/_us_/coefficient) is close to 1, which means the density of clustering is higher. And larger number of clusters that are formed, the more areas that are prone to earthquakes are detected.

**Slide 20: Conclusion**

We know that clustering algorithm is widely used. And DBSCAN is very effective in handling clusters with complex shapes and noise data.

In the application of earthquake distribution, other factors need to be considered, such as the impact of uneven density distribution on clustering effectiveness. Additionally, it is essential to cluster not only spatial [/ˈspeɪʃl/](cmd://Speak/_us_/spatial) attributes but also other properties like time, color, and temperature. Forcing normalization of non-spatial and spatial data can reduce the clustering characteristics of these attributes. These are areas that require further exploration in the future.

DBSCAN effectively handles complex clusters and noise in earthquake distribution analysis, but challenges like uneven density and the need to incorporate various attributes beyond spatial data require further exploration for improved clustering results.