www.geeksforgeeks.org /how-to-choose-the-right-distance-metric-in…

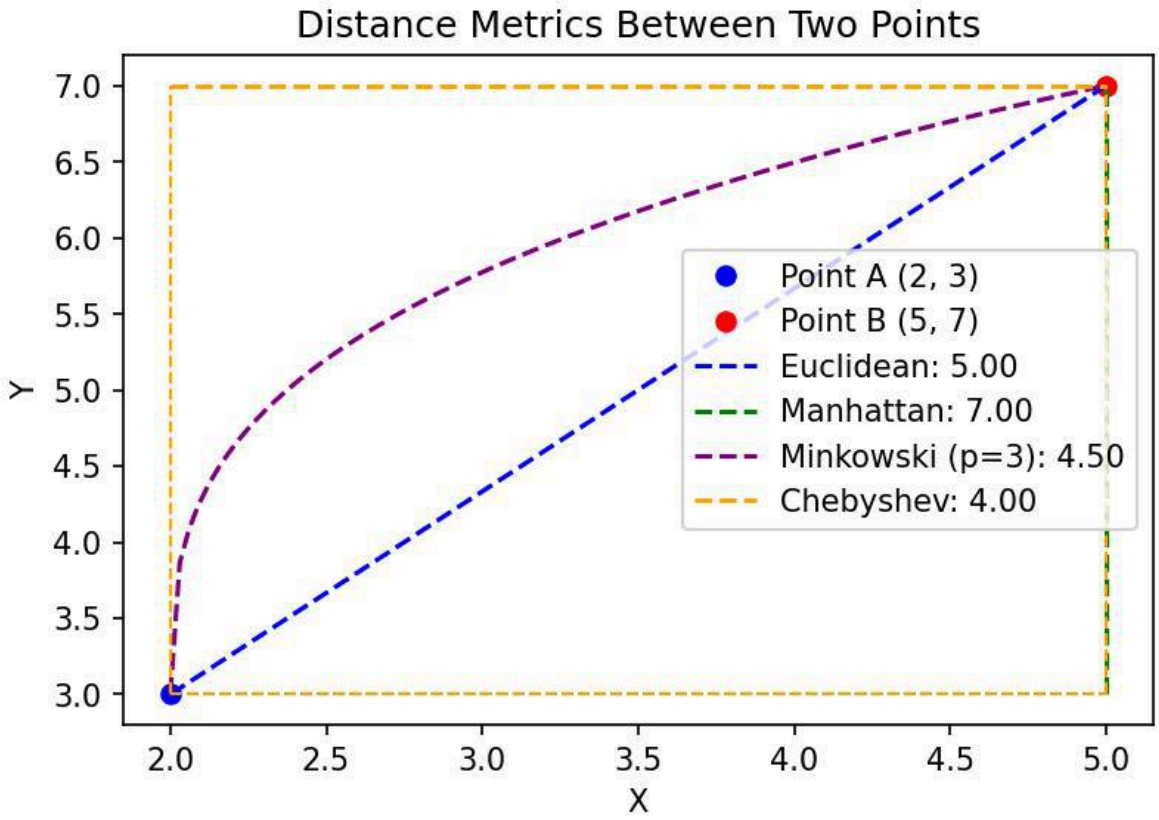
How to choose the right distance metric in KNN?

GeeksforGeeks ⋮ 6-8 minutes ⋮ 11/25/2024

When using the K-Nearest Neighbors (KNN) algorithm for classification or regression tasks, one of the most crucial decisions you'll make is choosing the right distance metric as it is an algorithm that works by identifying the 'k' nearest neighbors to a new data point and then predicting its class or value based on these neighbors. In simple terms, the distance metric determines how the algorithm measures the proximity between data points, and the right choice can significantly impact the accuracy and performance of your model.

The core idea here is that the algorithm relies on a distance metric to find these nearest neighbors. The most common distance metrics include:

- Euclidean
- Manhattan
- Minkowski
- Chebyshev distances



Visualization of each of the metrics individually

Here's a brief overview of each of them:

1. Euclidean Distance : Distance Metric in KNN

Euclidean distance is the most commonly used metric and is set as the default in many libraries, including Python's Scikit-learn. It measures the straight-line distance between two points in a multi-dimensional space.

Euclidean Distance: $d_{\text{Euclidean}}(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$

where $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$ are two points in n-dimensional space.

2. Manhattan Distance (L1 Norm)

Manhattan distance, also known as the taxicab or city block distance, measures the distance traveled along the grid-like streets of a city. It is the sum of the absolute differences between the corresponding coordinates of two points.

Manhattan Distance (L1 Norm):$d_{\text{Manhattan}}(p, q) = \sum_{i=1}^{n} |p_i - q_i|$

where $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$.

3. Minkowski Distance

Minkowski distance is a generalized form that can be adjusted to give different distances based on the value of 'p'. When p=1, it becomes Manhattan distance, and when p=2, it becomes Euclidean distance.

Minkowski Distance:$d_{\text{Minkowski}}(p, q, p) = \left(\sum_{i=1}^{n} |p_i - q_i|^p \right)^{1/p}$

where p is a parameter, and $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$.

4. Chebyshev Distance (Maximum Norm)

Chebyshev distance calculates the maximum absolute difference along any dimension. It is useful in scenarios where the maximum difference is critical.

Chebyshev Distance (Maximum Norm):$d_{\text{Chebyshev}}(p, q) = \max_i |p_i - q_i|$

where $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$.

Code Example:

Output:

```
Accuracy using Euclidean distance: 1.0
Accuracy using Manhattan distance: 1.0
Accuracy using Minkowski distance (p=3): 1.0
Accuracy using Chebyshev distance: 1.0
```

This code uses the K-Nearest Neighbors (KNN) algorithm with different distance metrics (Euclidean, Manhattan, Minkowski, and Chebyshev) to classify the Iris dataset. The dataset is split into training and test sets, and each KNN model is trained and tested. Finally, it calculates and prints the accuracy for each metric, showing how different distance metrics impact model performance.

Choosing the Right Distance Metric in KNN

Distance Metric	When to Use	Use Case Scenario
Euclidean Distance	- Continuous numerical data.	- Predicting house prices based on square footage and number of bedrooms.

Distance Metric	When to Use	Use Case Scenario
	- When the data is well-scaled.	- Image recognition where pixel values are continuous features.
	- Data with features on a grid (e.g., city streets).	- Delivery routing for trucks following city grids.
Manhattan Distance	- When data is less sensitive to outliers.	- Robot navigation through a grid with restricted movement (i.e., only vertical or horizontal).
	- When you need a flexible metric that can represent different distances.	- Analyzing weather data like temperature, humidity, and wind speed to predict likelihood of rain.
Minkowski Distance	- When you want to tune the parameter 'p' for customization (e.g., p=1 for Manhattan, p=2 for Euclidean).	- Choosing between Euclidean or Manhattan depending on the problem's spatial relationship.

Distance Metric	When to Use	Use Case Scenario
Chebyshev Distance	- When the maximum difference between coordinates is important.	- In a board game, measuring the maximum number of moves a piece can make in any direction.
	- When features represent movements along a grid with equal importance.	- Robot movement where diagonal and straight moves are equally important (e.g., chess, checkers).

- **Data Nature**: Choose a distance metric that aligns with the nature of your data. For example, Euclidean distance is suitable for continuous numerical data, while Hamming distance is used for binary data.
- **Feature Scaling**: Ensure that your data is scaled properly to avoid features with larger scales dominating the distance calculation.
- **Experimentation**: Use cross-validation to test different distance metrics and values of 'k' to find the best combination for your dataset.
- **Practical Use**: The right distance metric can significantly improve the accuracy and performance of your KNN model. For instance, Manhattan distance is less sensitive to outliers compared to Euclidean distance, making it more suitable for datasets with noisy data.

Previous Chapter Next Chapter