

K-Nearest Neighbor

Data Mining

Lê Công Minh Khôi – 519H0181



Introduction

Simple algorithm

Classification/R
egression

Based on a
similarity
measure



1970s

Statistical
estimation

Pattern
recognition



1990s

Popular in machine learning

The background features a dark purple gradient with several large, organic, fluid shapes in lighter shades of purple and blue. A large, semi-transparent circle is centered behind the text.

**Why do we need a kNN
algorithm?**



kNN

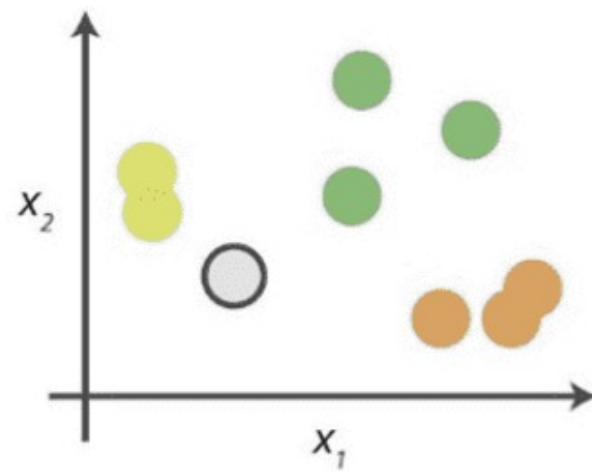
Simple To
Implement

Versatile

Effective

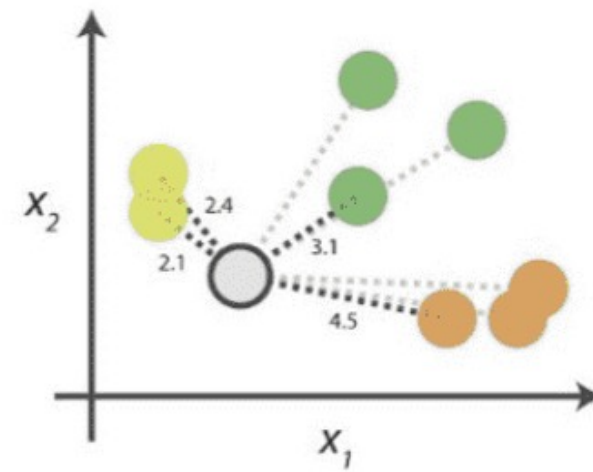
How does the kNN algorithm work?

0. Look at the data




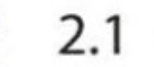



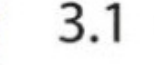


Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances









Start by calculating the distances between the grey point and all other points.

2. Find neighbours

Point Distance			
		2.1	→ 1st NN
		2.4	→ 2nd NN
		3.1	→ 3rd NN
		4.5	→ 4th NN

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

Class	# of votes	
	2	→ Class  wins the vote! Point  is therefore predicted to be of class  .
	1	
	1	

Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

The kNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

Step by step explanation

1. Load the data
2. Choose k
3. For each point in the data
 1. Calculate the distance between current point and the point we try to predict
 2. Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first k entries from the sorted collection
6. Get the labels of the selected k entries
7. Return prediction
 1. If regression, return the mean of the k labels
 2. If classification, return the mode of the k labels



How to choose the value of k ?

Usually odd

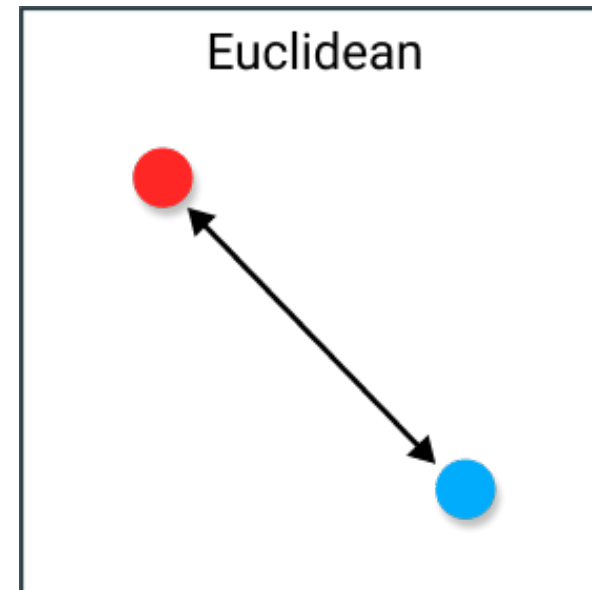
k 
sensitive to
noise 

Try different k
value

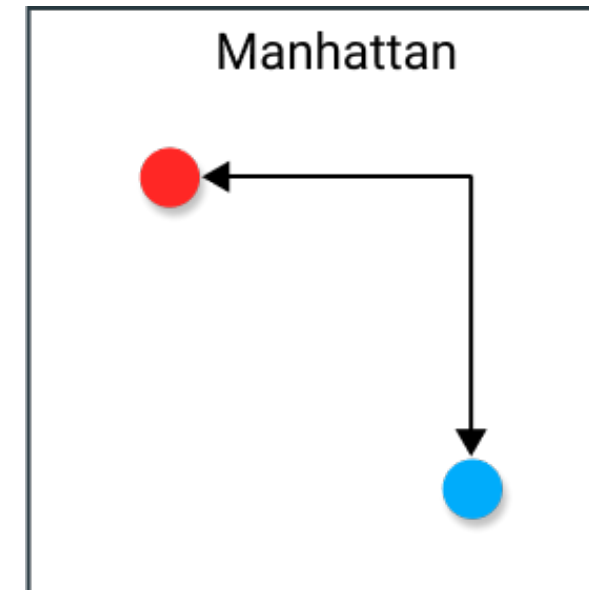
Pick the best result

Calculating the distance

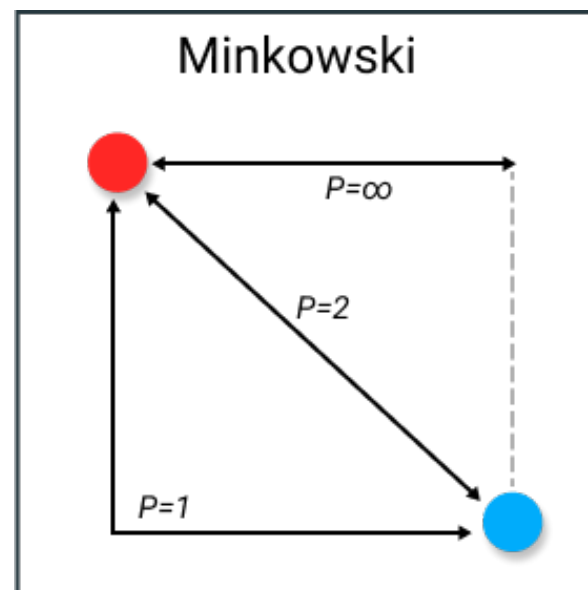
Euclidean



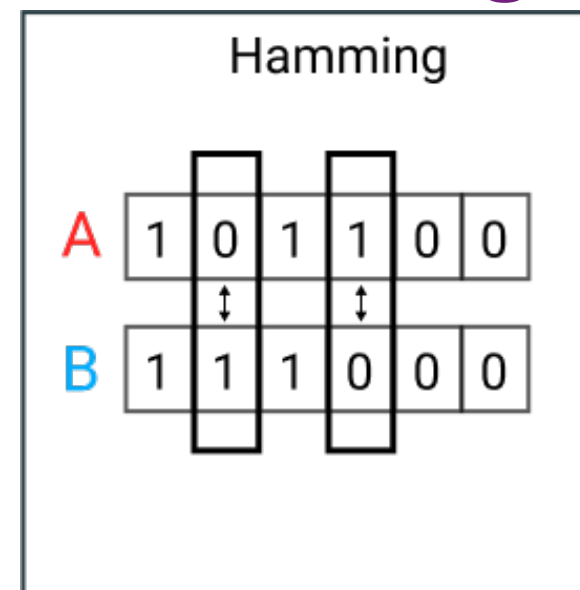
Manhattan



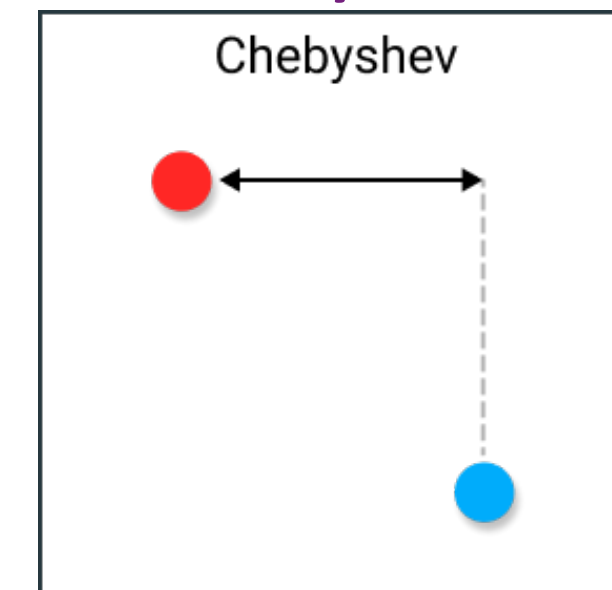
Minkowski



Hamming



Chebyshev



The image features several abstract, fluid shapes in shades of purple and blue. One large circular shape is in the top left. A horizontal, elongated shape with a central constriction is in the top right. A curved, teardrop-like shape is in the bottom left. The word "Advantages" is centered in the top half of the image.

Advantages

**Simple to
implement**

Versatile

**Training is
trivial**

**Works with any
number of
classes**

**Effective if the
training data is
large**



Disadvantages

**Computationally
expensive**

**High memory
requirement**

**Prediction stage
might be slow
with a large
testing dataset**



**Does not work
well with high
dimensional
data**

**One-hot
encoding is
required for
categorical
features**

Thank you