4th-week-1

# kNN

Wang Jianfang
王建芳(in Chinese)

# Agenda

- Concept
- Example
- Code

# kNN

- **kNN**
- namely **k-NearestNeighbor** algorithm
- 

# Classification of machine learning algorithms

- Supervised learning
- Unsupervised learning
- Semi-supervised learning

# Supervised learning

- A function (model parameter) is learned from the given trainin g data set. When new data comes, the result can be predicted a ccording to this function. The training set of supervised learnin g requires input and output, which can also be said to be chara cteristics and objectives. The objectives of the training set are l abeled manually.
  - **Regression** :The prediction results are continuous, such as p redicting tomorrow's temperature, 23,24,25 degrees.
  - Classification:The results of the forecast are discrete, such a s predicting the weather for tomorrow - cloudy, sunny, rainy.

# Unsupervised learning

- The input data is not marked, and there is no d efinitive result.
- The category of sample data is unknown, so it i s necessary to classify the sample set (clusterin g) according to the similarity among samples t o minimize the gap within the class and maxim ize the gap between the classes.

# Semi-supervised learning

- The training datasets contain both marked sam ple data and unmarked sample data

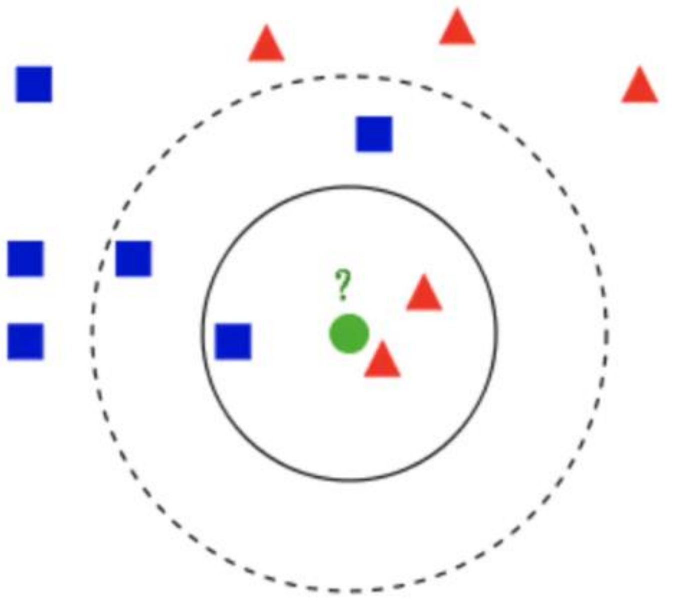
# Intro.

- K-Nearest Neighbor (KNN) is one of the simplest ma chine learning classification algorithms.
- In 1968, it was proposed by Cover and Hart, and its a pplication scenarios include character recognition, tex t classification, image recognition and other fields.
- Its core idea is that a sample is most similar to the K s amples in the data set, and if the majority of the K sa mples belong to a certain category, the sample also be longs to that category.

# Principle

- The core idea of the KNN algorithm is to represent the classification of the target data by using the nearest K samples.
- Specifically, there are training sample sets, and each sample contains data features and classification values.
- Input new data, comparing the data with each sample in the training sample sets, and find the nearest K. The classification with more occurrences can be used as the classification of new data in the k data,.

# Example




- We need to figure out what green looks like.
- When k is equal to 3, it's in the triangle.
- When k is equal to 5, it's a square.

# Example

- The principle of KNN is that when predicting a new value of *x*, it will judge which category *x* belongs to according to the category of K point s closest to it. It sounds a little convoluted, but let's look at the picture.

# KNN Characteristics

- Therefore, this method has the following characteristics:
  - Supervised learning: the training sample set contains classifi cation information
  - The algorithm is simple and easy to understand and implem ent
  - The results are influenced by K value, K generally does not exceed 20
  - The distance from each sample in the sample set needs to be calculated due to the large amount of calculation.
  - The imbalance of training sample set leads to inaccurate res ults

# KNN Method

- K-NN is used to divide each group of data into a certain class. Its pseudo code is as follows:
  ① Calculate the distance between the point in the known clas s dataset and the current point;
  ② According to the increasing order of distance;
  ③ Select the K points with the minimum distance from the cu rrent;
  ④ The frequency of the first K points was determined;
  ⑤ The most frequent category of the first K points is determi ned as the prediction classification of the current point.

# Distance Functions

- Let X be an n-dimensional real vector space $R_n$ ,

$$x_i, x_j \in \mathcal{X}, \quad x_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})^T, x_j = (x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(n)})^T,$$

- The $L_P$ distance of $x_i$, $x_j$ is defined as: $p > 1$

$$L_p(x_i, x_j) = (\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p)^{\frac{1}{p}}$$

- When p = 1, it is called Manhattan distance:

$$L_1(x_i, x_j) = \sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|$$

- When p = 2, it is called Euclidean distance:

$$L_2(x_i, x_j) = (\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^2)^{\frac{1}{2}}$$

- When p = ∞ , it is the maximum value of each coordinate distance:

$$L_\infty(x_i, x_j) = max_l |x_i^{(l)} - x_j^{(l)}|$$

# Distance Functions Example

- Given the three points $x_1 = (1,1)^T$, $x_2 = (5,1)^T$, $x_3 = (4,4)^T$, in two-dimensional space, we try to find the nearest neighbor of $x_1$ under the $L_p$ distance when $p$ takes different values.
- Vectors

# Distance Functions Example

- The distance between $x_1$ and $x_2$:
- The Manhattan distance between $x_1$ and $x_3$ is:
- Then the European distance between $x_1$ and $x_3$ is:
- The $L_3$ distance between $x_1$ and $x_3$ is:

# Distance Functions Example

- Given the three points $x_1 = (1,1)^T$, $x_2 = (5,1)^T$, $x_3 = (4,4)^T$, in two-dimensional space, we try to find the nearest neighbor of $x_1$ under the $L_P$ distance when $p$ takes different values.
- solution:For $x_1$ and $x_2$, because the number of $x_1$ and $x_2$ in the f irst dimension is 1 and 5, and the number in the second dimens ion is 1. Therefore, when calculating the distance between $x_1$ a nd $x_2$, we only need to calculate $x_1^{(1)}$ and $x_2^{(1)}$, $L_P(x_1, x_2) = 4$.

# Distance Functions Example

- For $x_1$ and $x_3$, because the numbers of $x_1$ and $x_3$ in the first dim ension are not the same, and the numbers in the second dimens ion are not the same, the Manhattan distance between $x_1$ and $x_3$ is:

$$L_1(x_1, x_3) = \sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}| = \sum_{l=1}^{2} |x_i^{(l)} - x_j^{(l)}| = 3 + 3 = 6$$

- Then the European distance between $x_1$ and $x_3$ is:

$$L_2(x_i, x_j) = (\sum_{l=1}^{n} \ |x_i^{(l)} - x_j^{(l)}|^2)^{\frac{1}{2}} = (\sum_{l=1}^{2} \ |x_i^{(l)} - x_j^{(l)}|^2)^{\frac{1}{2}} = 3\sqrt{2} = 42.4$$

- The $L_3$ distance between $x_1$ and $x_3$ is:

$$L_3(x_i, x_j) = (\sum_{l=1}^{n} \ |x_i^{(l)} - x_j^{(l)}|^3)^{\frac{1}{3}} = 3.78$$

# Distance Functions

- From Step 3, we can see the distance between the position poi nt we need to first ask for and each known point we choose, so there are three formulas for calculating the distance between t wo points:

Euclidean $$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan $$\sum_{i=1}^{k}\left|x_i - y_i\right|$$

Minkowski $$\left(\sum_{i=1}^{k}(\left|x_i - y_i\right|)^q\right)^{1/q}$$

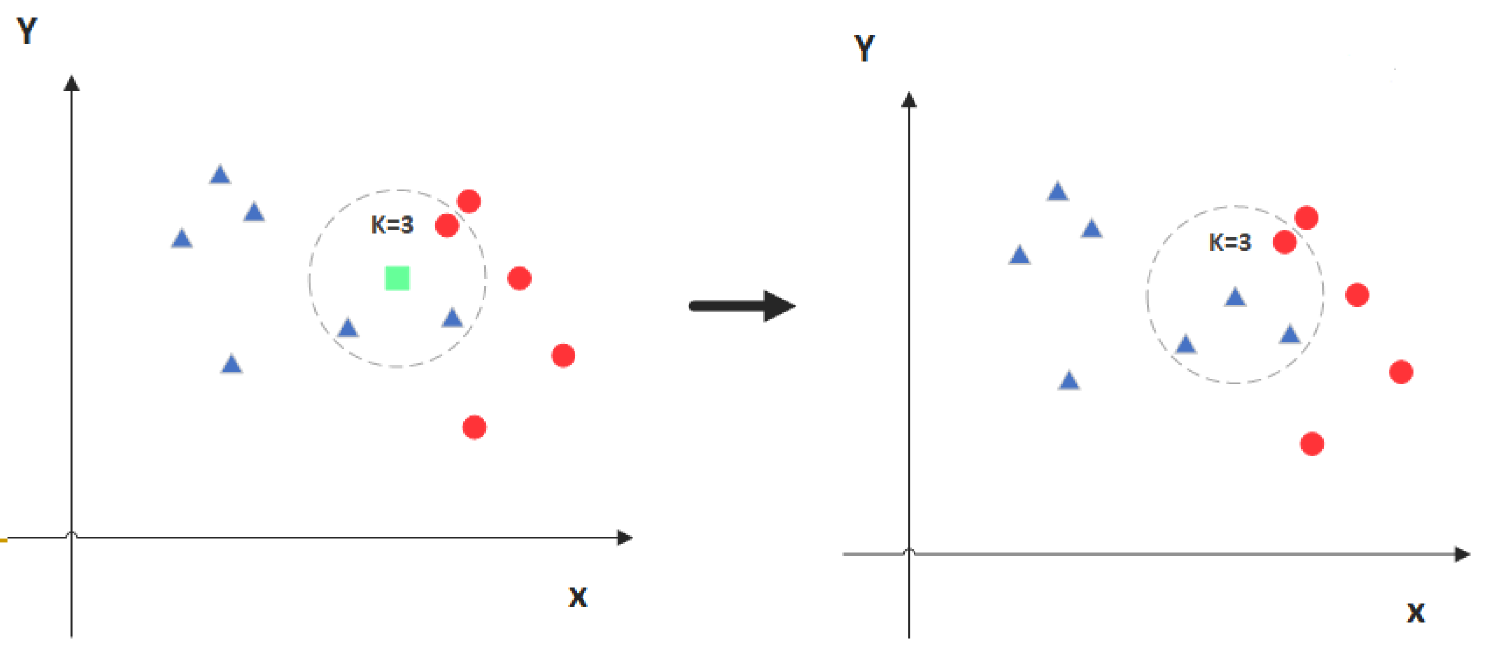
# Code

```
import numpy
import operator


def createDataSet():
    group = numpy.array([[1.0, 1.1], [1.0, 1.0], [0, 0], [0, 0.1]])
    labels = ['A', 'A', 'B', 'B']
    return group, labels


def classify0(inX, dataSet, labels, k):
    dataSetSize = dataSet.shape[0]
    diffMat = numpy.tile(inX, (dataSetSize, 1)) - dataSet
    sqDiffMat = diffMat ** 2
    sqDistances = sqDiffMat.sum(axis=1)
    distances = sqDistances ** 0.5
    sortedDistIndicies = distances.argsort()
    classCount={}
    for i in range(k):
        voteIlabel = labels[sortedDistIndicies[i]]
        classCount[voteIlabel] = classCount.get(voteIlabel, 0) + 1
    sortedClassCount = sorted(classCount.items(), key=operator.itemgetter(1), reverse=True)
    return sortedClassCount[0][0]


data_set, labels = createDataSet()
result = classify0([0, 0], data_set, labels, 3)
print(result)
```

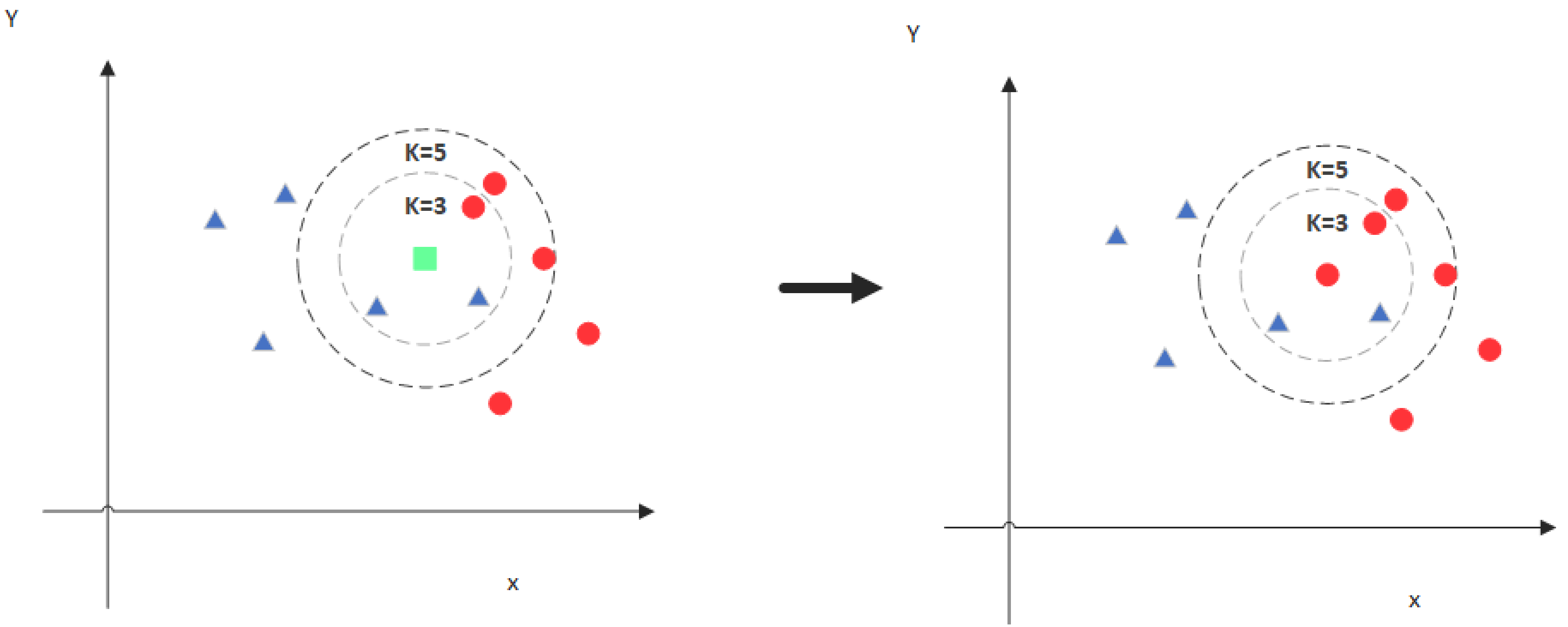
# Distance formula

- Where createdataset() is used to generate a sim ple dataset with tags.
- Classify0 () uses the Euclidean distance formul a to calculate the distance between the new dat a and the data in the dataset.

- The green point on the graph is the point we're trying to predic t, let's say K is equal to 3. The KNN algorithm then finds the t hree closest points (circled here) to see which have more categ ories. For example, in this case, there are more blue triangles, a nd the new green dots are classified as blue triangles.



- But when K is equal to 5, the decision is different. This time there are more red circles, so the new green dots are classified as red circles. From this example, we can see that K is important.



# Homework

- Implementation of KNN algorithm classification in Python (Iris Dataset)

# Questions and Comments?

**Thank you!!**