華東師範大學
EAST CHINA NORMAL UNIVERSITY

第4讲 机器学习实例——鸢尾花分类

刘小平

2022.9

鸢尾花分类

内容

- 鸢尾花分类简介
- 构建人工智能系统实现分类
- 特征提取
- 分类器

鸢尾花分类简介

- 全世界有300多种鸢尾花，常见的有山鸢尾、变色鸢尾等。
- 假设要构建一个人工智能系统，使它能自动识别区分山鸢尾和变色鸢尾两个品种，这样的系统，可以称它为分类器。
- 分类——根据信息数据的不同特点，判断它属于哪个类别。

Setosa
山鸢尾

Versicolor
变色鸢尾

构建人工智能系统实现分类

特征提取

分类器

变色鸢尾？
Or
山鸢尾？

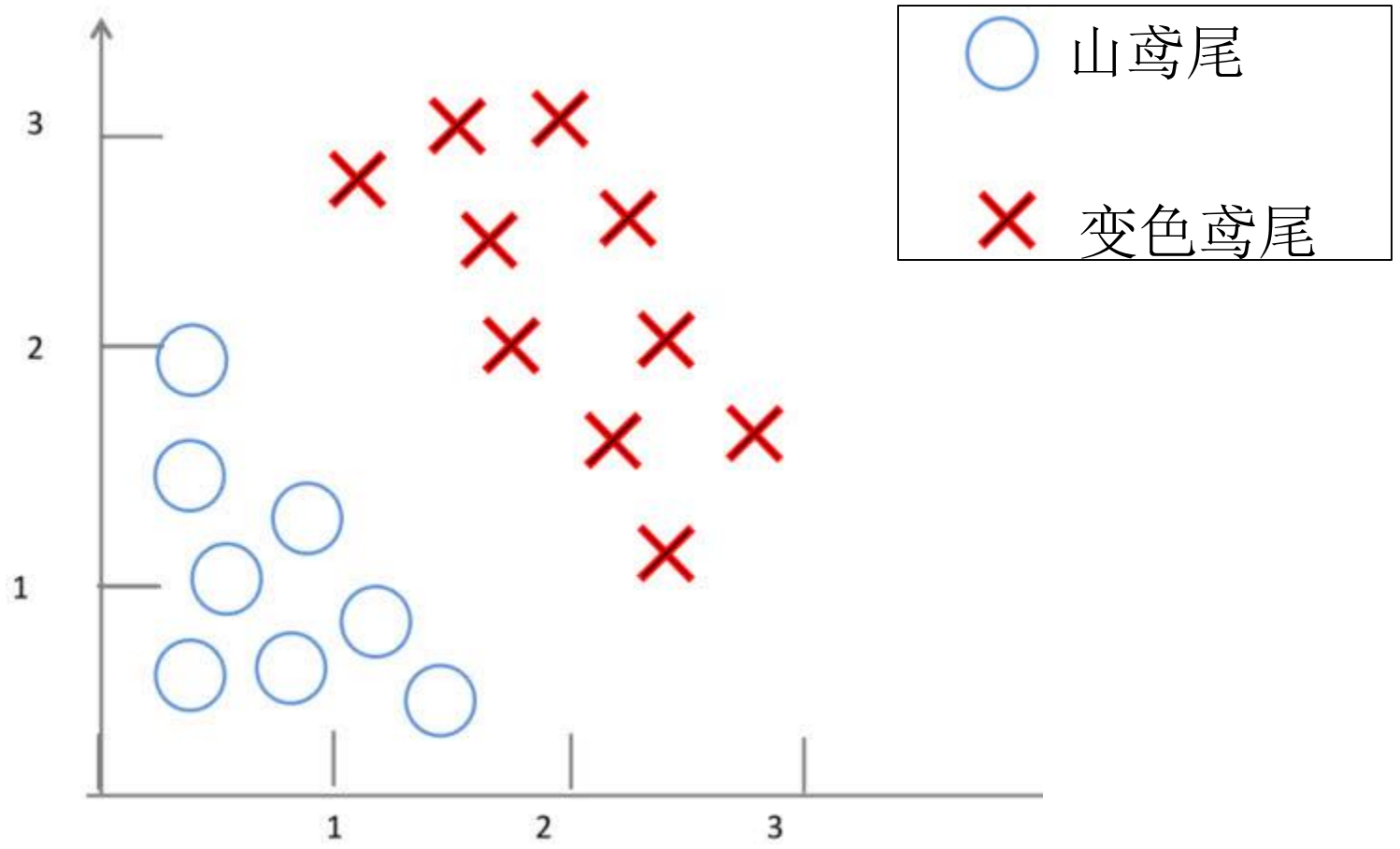
特征提取

- 鸢尾花的特征：花瓣长度、花瓣宽度、花瓣颜色、植株高度
- 人类识别鸢尾花的经验：鸢尾花的大小
- 选取的特征，决定了分类器最终效果
- 这里选择花瓣长度和宽度作为鸢尾花的特征

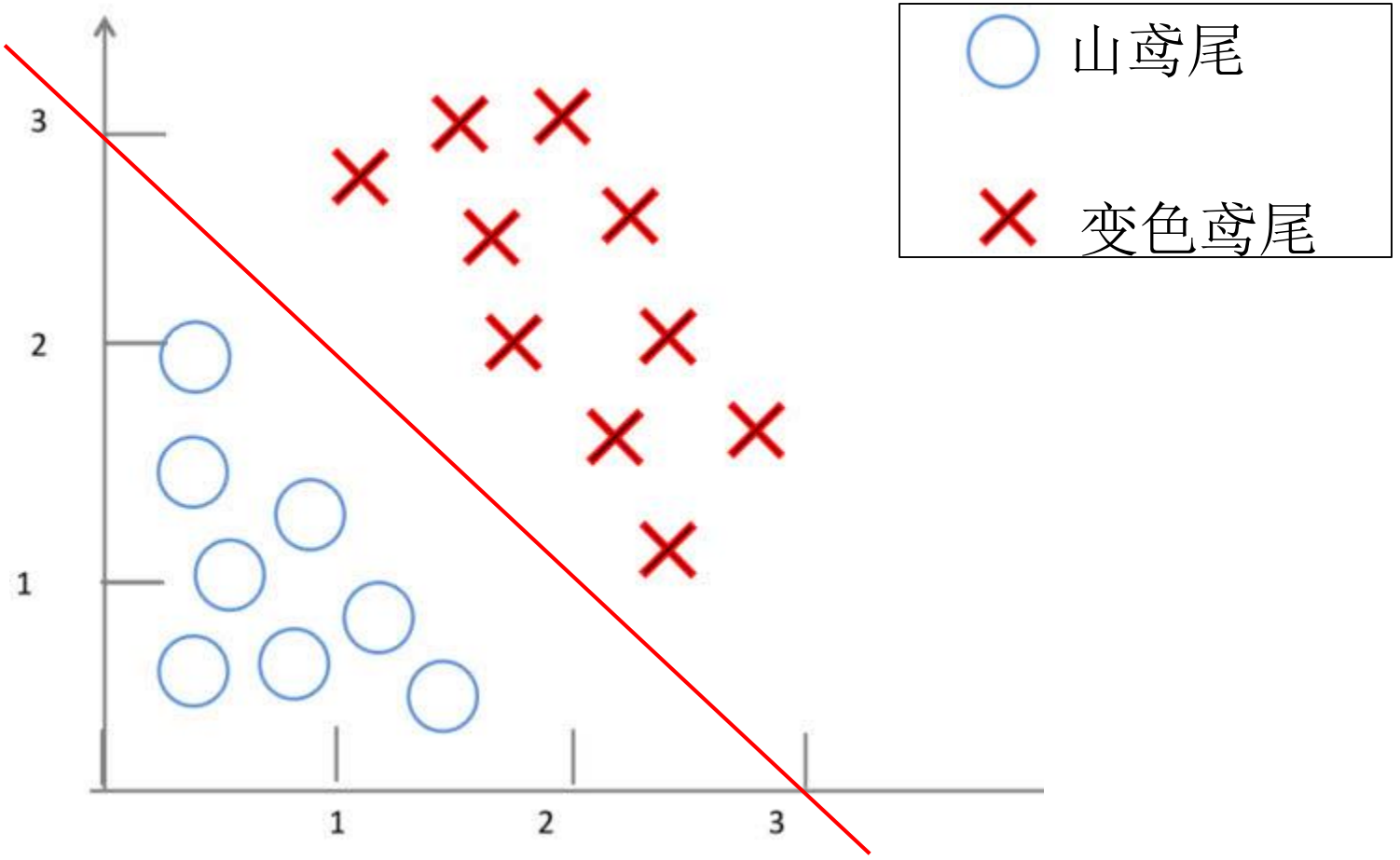
描述特征的术语

- 特征向量(feature vector)：x=(长度，宽度)
- 特征点(feature point)：表征特征向量的点成为特征点
- 特征空间(feature space)：所有这些特征点构成的空间成为特征空间。

鸢尾花特征图示



分类器



分类器

- 直线方程：$x_1 + x_2 - 3 = 0$
  - 假设山鸢尾花落在直线左下区域，特征点输出−1；
  - 变色鸢尾花落在直线右上区域，特征点输出+1；
  - 用如下函数表示分类器：
- 分类函数：

$$g(x1, x2) = \begin{cases} +1, x_1 + x_2 - 3 \geq 0 \\ -1, x_1 + x_2 - 3 < 0 \end{cases}$$

- 记 $f(x) = f(x_1, x_2) = x_1 + x_2 - 3$ ，它是分类函数的核心。
- 线性分类器：$f(x_1, x_2, ..., x_n) = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + b$
- 分类器参数：$a_1, a_2 \cdots, a_n, b$

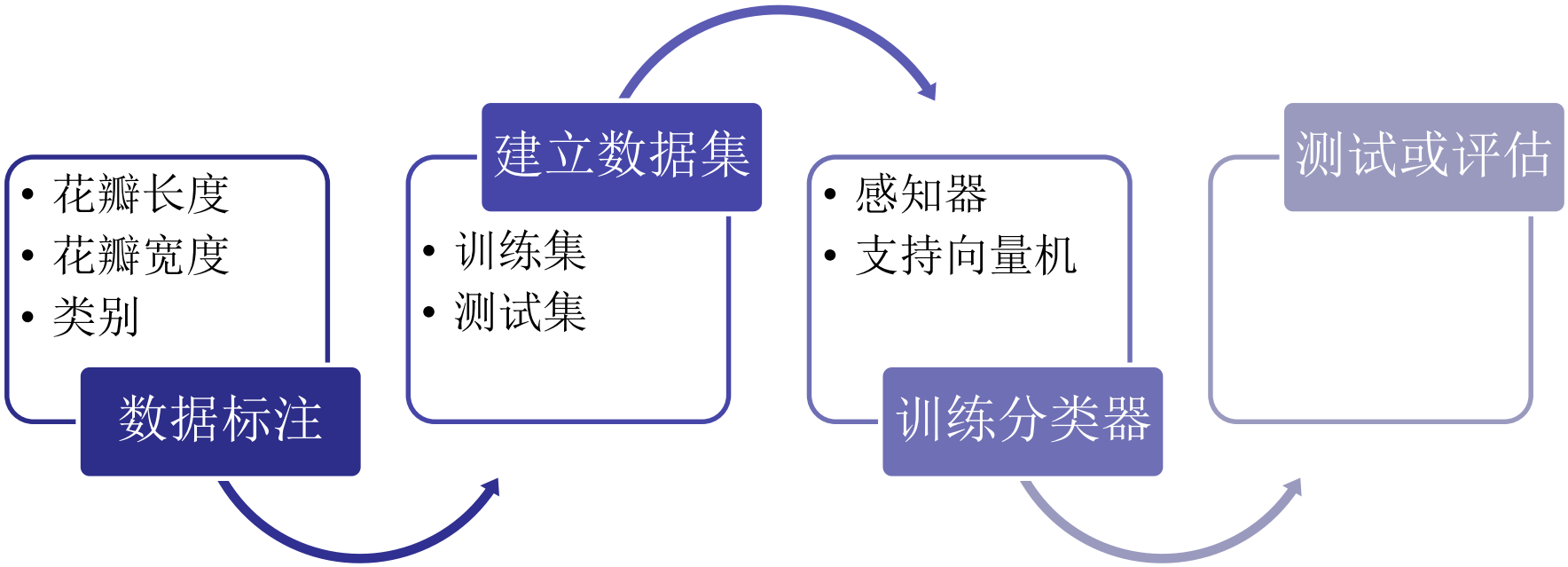
分类器

- 训练分类器：使分类器经过学习得到合适参数的过程。例如：找到一条好的分类直线。
  - 训练：训练样本、训练数据、训练集
- 分类器的测试或评估：计算分类准确率

$$\text{分类准确率}=\frac{\text{分类正确的样本数}}{\text{测试样本总数}}\times 100\%$$

  - 测试：测试样本、测试数据、测试集
- 分类器的应用：将一朵鸢尾花的花瓣长度和宽度数据，输入到训练好的分类器中，分类器就会输出它的预测结果。

分类器

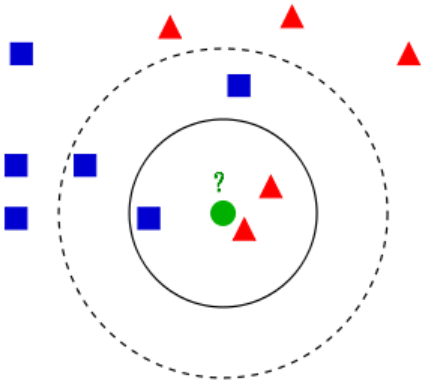


KNN算法

- 何谓K近邻算法，即K—Nearest Neighbor Algorithm，简称KNN算法，单从名字来猜想，可以简单粗暴的认为是：分析一个人时，我们不妨观察和他最亲密的几个人。同理的，在判定一个未知事物时，可以观察离它最近的几个样本，这就是 KNN（K最近邻）的方法。

KNN算法思想

- 如上图所示，有两类不同的样本数据，分别用蓝色的小正方形和红色的小三角形表示，而图正中间的那个绿色的圆所标示的数据则是待分类的数据。
- 问题：图中的绿色的圆属于哪一类？
- 如果K=3，绿色圆点的最近的3个邻居是2个红色小三角形和1个蓝色小正方形，少数从属于多数，基于统计的方法，判定绿色的这个待分类点属于红色的三角形一类。
- 如果K=5，绿色圆点的最近的5个邻居是2个红色三角形和3个蓝色的正方形，还是少数从属于多数，基于统计的方法，判定绿色的这个待分类点属于蓝色的正方形一类。



判断电影类型实例+

电影唐人街探案属于什么类型？

序号	电影名称	拥抱镜头	打斗镜头	电影类型
1	叶问3	2	65	动作片
2	伦敦陷落	3	55	动作片
3	代理情人	38	2	爱情片
4	新步步惊心	34	10	爱情片
5	海王	2	50	动作片
6	无双	4	40	动作片
7	人间中毒	28	1	爱情片
8	傲慢与偏见	25	0	爱情片
9	唐人街探案	3	30	？

向量化

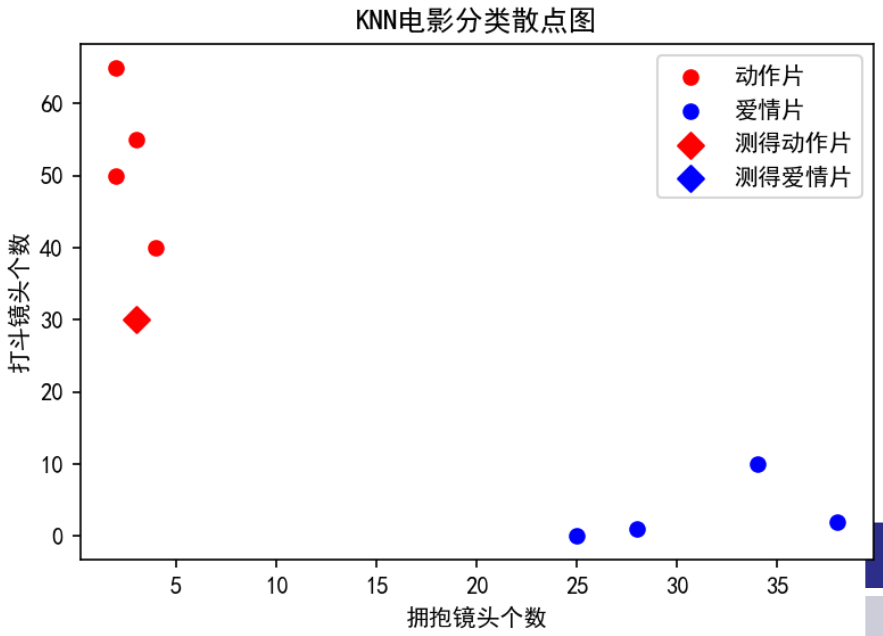
动作片：0

爱情片：1

点	X1	X2	y
A点	2	65	0
B点	3	55	0
C点	38	2	1
D点	34	17	1
E点	2	50	0
F点	4	40	0
G点	28	1	1
H点	25	0	1
I点	3	30	？

二维特征样本，一维标签（目标）

相似度——欧氏距离+



计算所有各点与 1 点的距离

电影	点	X1	X2	电影类型(y)	距离	K=1	K=3
叶问3	A点	2	65	动作片	35		
伦敦陷落	B点	3	55	动作片	25		✔
代理情人	C点	38	2	爱情片	44.8		
新步步惊心	D点	34	17	爱情片	36.9		
海王	E点	2	50	动作片	20		✔
无双	F点	4	40	动作片	10	✔	✔
人间中毒	G点	28	1	爱情片	38		
傲慢与偏见	H点	25	0	爱情片	37		
唐人街探案	1点	3	30	?			

KNN算法描述

K近邻算法的一般步骤如下：

（1）选择参数K；

（2）计算未知样本与所有已知样本的相似度；

（3）根据相似度对样本进行排序，并选择最近K个已知样本；

（4）根据少数服从多数的投票法则，预测未知样本为K个最邻近样本中最多数的类别。

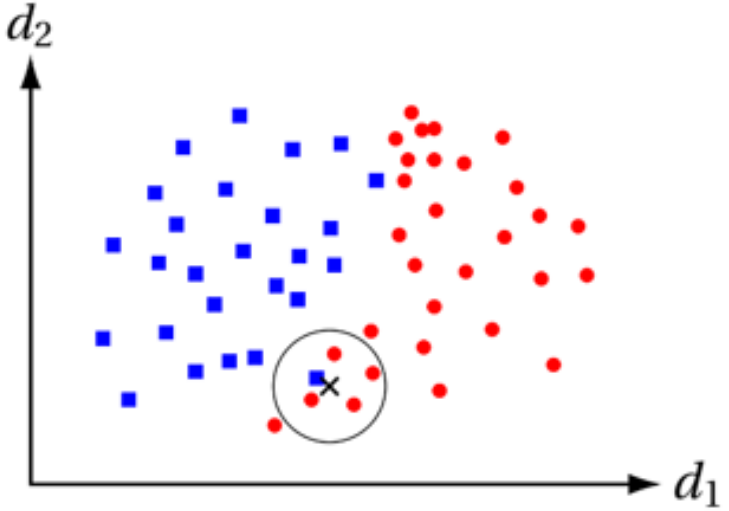


图 5-8 分类问题的 K 近邻算法

$$E(x_1, x_2) = \sqrt{\sum_{j=1}^{m}(x_{1j} - x_{2j})^2} \text{ 。}$$

- 分类器不需要使用训练集进行训练，训练时间复杂度为0。
- KNN分类的计算复杂度和训练集中的样本数目成正比，也就是说，如果训练集中样本总数为n，那么KNN的计算复杂度为O(n)。

使用sklearn搭建KNN模型示例

- 1、电影类型识别。
- 2、鸢尾花种类识别。
- 思考：KNN算法是监督学习、非监督学习、半监督学习？

使用sklearn搭建KNN模型

核心操作步骤：

1. 创建 KNeighborsClassifier **分类器对象**，并进行初始化。
2. 调用分类器对象 fit 方法，对数据集进行训练。
3. 调用 predict 方法，对测试集进行预测。

创建 KNeighborsClassifier 分类器对象

sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, weights='uniform',algorithm='auto',leaf_size=30,p=2,metric='minkowski', metric_params=None, n_jobs=None, **kwargs)

返回： KNeighborsClassifier 对象实例。

常用参数：

- n_neighbors：int 型，可选，缺省值为 5，就是 KNN 中的近邻数量 k 值。
- weights：计算距离时使用的权重。
  - 缺省值是“uniform”，均匀的权重。每个邻域中的所有点的权重均相等。
  - 也可以取值“distance”，权重点按其距离的倒数表示，查询点的近邻比远邻具有更大的影响力。
  - 用户定义的函数，它接受距离数组，并返回包含权重的相同形状的数组。
- metric：距离的计算。
  - 缺省值是“minkowski”。当 p=2，metric='minkowski'时，使用的是欧式距离。
  - p=1,metric='minkowski'时为曼哈顿距离。

使用sklearn搭建KNN模型常用方法

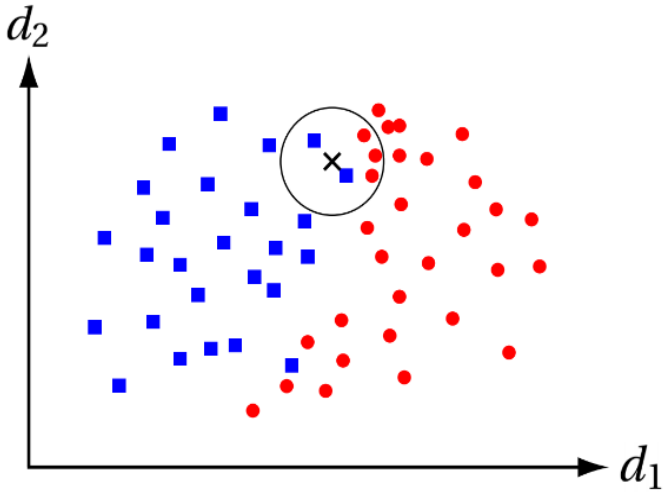
- fit(X,y)：训练/拟合（自动调整模型参数得到最终模型对象）。
  - X：训练集特征数据样本。（二维数据）
  - y：训练集的目标值（标签）数据样本。（一维或二维数据）
  - 返回：KNeighborsClassifier 对象实例。
- predict(X)：对测试集进行预测。
  - X：测试集特征数据样本。
  - 返回：根据给定的特征数据预测得出其所属的类别标签（测试集目标数据）。
- score(X，y，……)：得分指标，全部类别平均正确率。
  - X：测试集特征数据样本。
  - y：测试集中的目标真实值。
  - 返回：各类别平均正确率(mean accuracy)。
- kneighbors(X=None，……)：返回指定的邻居信息。
  - X：测试集特征数据样本。
  - n_neighbors：最近邻居数。
  - return_distance：逻辑值。返回邻居的距离和邻居的索引号，或只返回邻居的索引号。
  - 返回：由 return_distance 决定。

不同的模型，得分算法不同

几乎所有的模型皆有此三种方法，用法统一

KNN的缺点

- 1 需要大量的空间去存储所有已知样本，特别当样本是图形、视频，数据量大时，会导致内存不足；
- 2由于每次预测需要计算未知样本和所有已知样本的相似度(因此算法的计算复杂度和样本集中的样本数量成正比；
- 3 在样本分布密度不平衡的区域（密度大的这一类样本容易占据主导）导致未知样本被错分到该类别。



被错分的未知样本

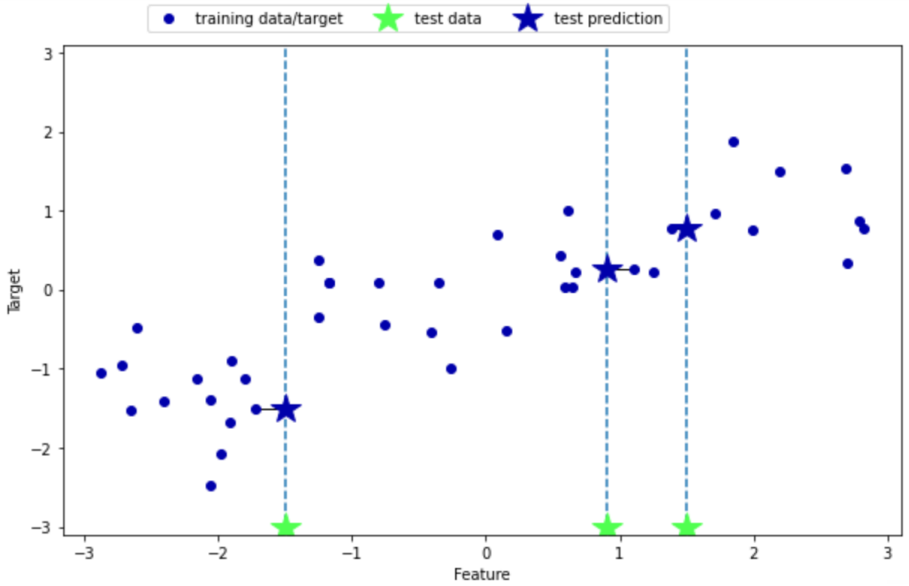
改进——权重加权的KNN算法

假设得到k个近邻点(按它们到未知样本的距离从小到大排列，即$d_1, d_2, \cdots, d_k$，第i个靠近的近邻点的权重为：

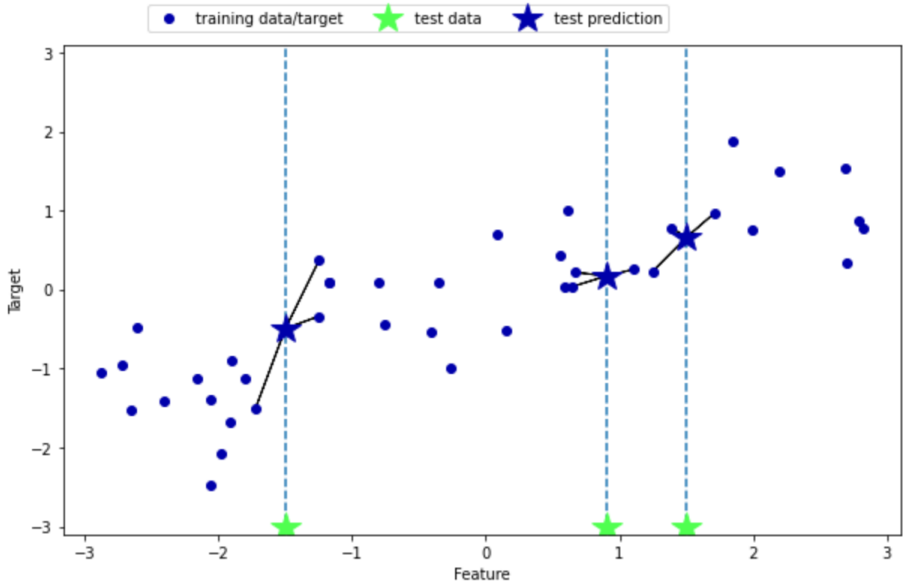
$$w_i = \begin{cases} \dfrac{d_k - d_i}{d_k - d_1}, & d_k \neq d_1 \\ 1, & d_k = d_1 \end{cases}$$

距离最近的近邻点的权重为1，距离越远权重越小，最远的为0。这样，在预测未知样本应该属于哪一类的时候，不再是按照邻近点的个数来决定类别，而是要加权每个点的权重，最后按照加权求和值的大小来决定类别。

KNN——回归示例



K近邻算法也可以用来解决回归问题。找到最近邻的K个点（这里是1或3个），然后根据这K个近邻点，计算它们对应的标签值的平均值，即为最后的预测值。



KNN算法三个基本要素

- K 值的选择
- 距离度量
- 分类决策规则

KNN算法的优点

- 简单，易于理解，易于实现，无需估计参数，无需训练
- 适合对稀有事件进行分类
- 特别适合于多分类问题(multi-modal,对象具有多个类别标签)

KNN算法的缺点

- 当样本不平衡时（如一个类的样本容量很大，而其他类样本容量很小时）有可能导致当输入一个新样本时，该样本的K个邻居中大容量类的样本占多数。
- 需要存储全部训练样本，计算量较大
- 可解释性较差，无法给出决策树那样的规则。

相似度量

- 以分类为例，在特征空间中，如果一个样本的K个最相似的样本中的大多数属于某一个类别，那么该样本也应该属于这个类别。
- 注意：最相似是如何判定的，它是通过相似度量来给出的
- 相似度：做分类时常常需要估算不同样本之间的相似性度量，即相似度，通常是通过距离来衡量的，如欧氏距离、余弦相似度、Pearson相关系数等。

如何确定K值？

- 遍历搜索结合交叉验证选择最佳参数

```
from sklearn import neighbors
from sklearn import datasets
from sklearn.model_selection import cross_val_score
# 获取鸢尾花数据集
iris = datasets.load_iris()
#观察 KNN 模型的不同 K 超参数效果
for k in range(3, 16, 2):
    # 生成 KNN 近邻算法模型对象
    knn = neighbors.KNeighborsClassifier(
                                n_neighbors=k)
    # 5折交叉验证得分
    scores = cross_val_score(knn,
                             iris.data,
                             iris.target, cv=5)
    print(scores)
    print(f'K = {k}: 平均正确率 {100 * scores.mean():.2f}%.')
```

运行结果：

```
[0.96666667 0.96666667 0.93333333 0.96666667 1.
K = 3: 平均正确率 96.67%.
[0.96666667 1.         0.93333333 0.96666667 1.
K = 5: 平均正确率 97.33%.
[0.96666667 1.         0.96666667 0.96666667 1.
K = 7: 平均正确率 98.00%.
[0.96666667 1.         0.96666667 0.93333333 1.
K = 9: 平均正确率 97.33%.
[0.93333333 1.         1.         0.96666667 1.
K = 11: 平均正确率 98.00%.
[0.93333333 1.         0.96666667 0.96666667 1.
K = 13: 平均正确率 97.33%.
[0.93333333 1.         0.93333333 0.96666667 1.
K = 15: 平均正确率 96.67%.
```

谢谢！