K-Nearest Neighbor Classification

KNN:K最近邻分类算法

yyy

分类

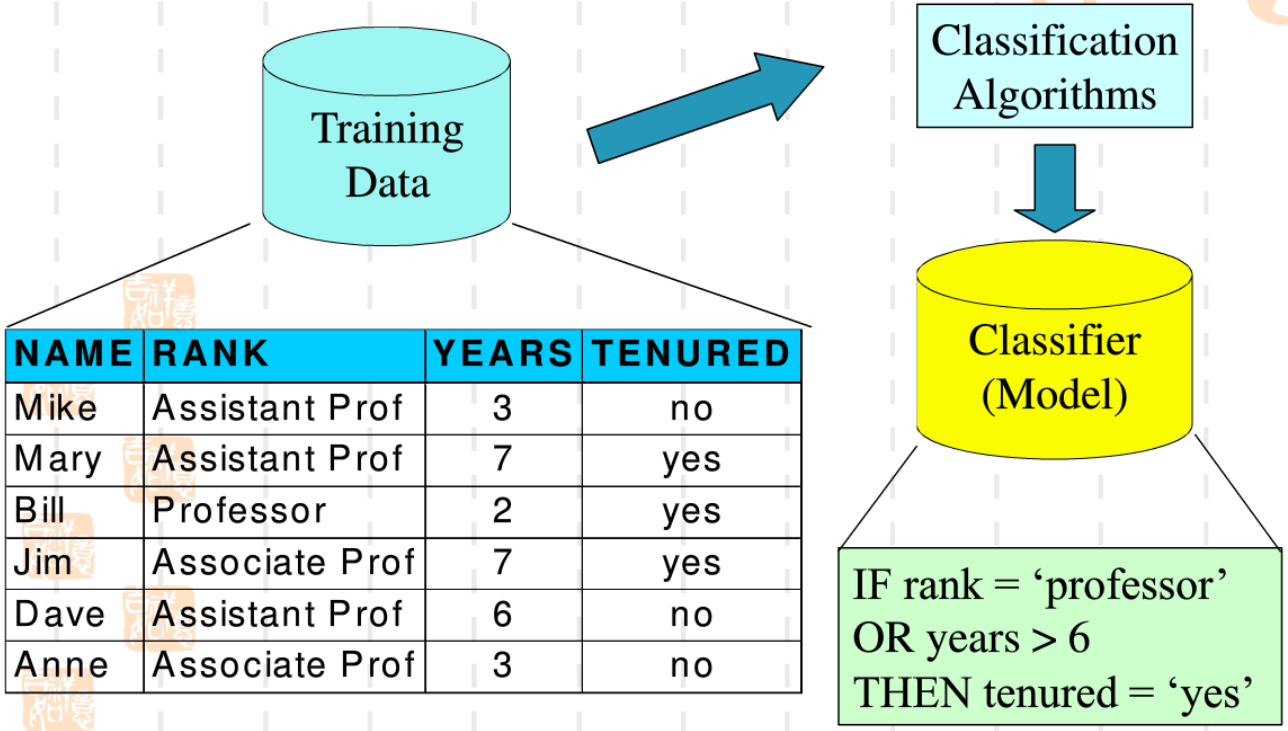
- 输入数据是记录的集合。每条记录也称为样本或样例，用元组（x，y）表示。x是属性集合，y是类标号（分类属性或目标属性）。类标号是离散的。（回归的目标属性y是连续的）
- 分类：通过学习得到一个目标函数（分类函数）f，把每个属性集x映射到一个预先定义的类标号y。
- 分类任务：确定对象属于哪个**预定义**的目标类。

脊椎动物的数据表

名字	体温	冬眠	有腿	胎生	类标号
人类	恒温	否	是	是	哺乳类
蝙蝠	恒温	是	是	是	哺乳类
青蛙	冷血	是	是	否	两栖类
蟒蛇	冷血	是	否	否	爬行类

分类

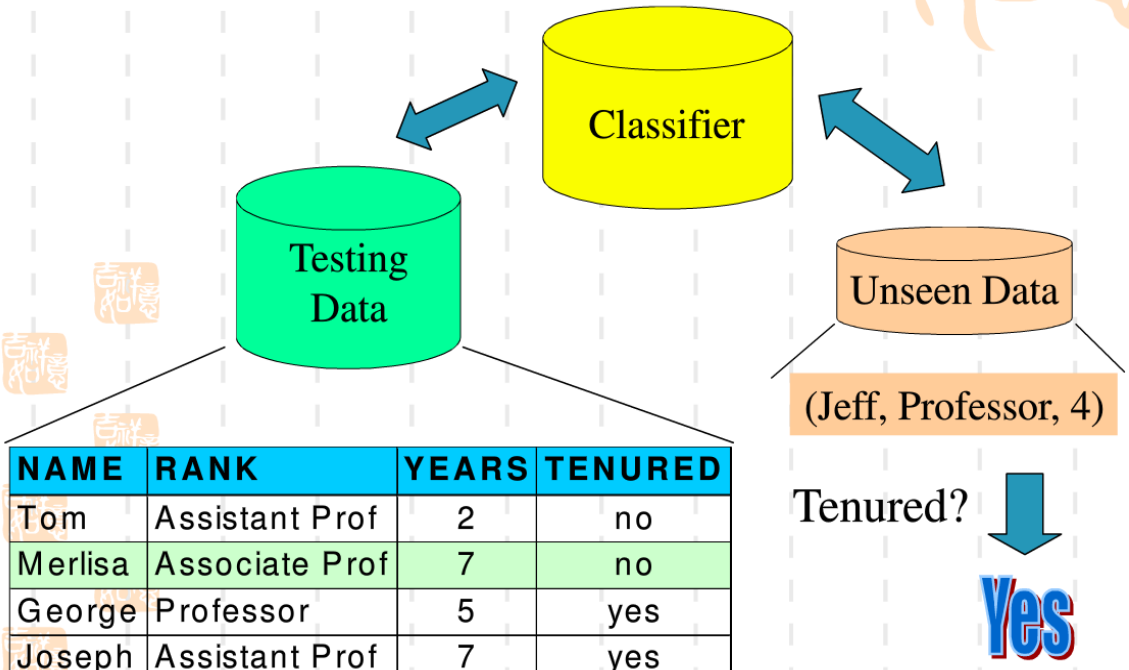
分类的实现—模型构建



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

分类

分类的实现—利用模型预测



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

分类性能

表1 二类问题的混淆矩阵

		预测的类	
		类=1	类=0
实际的类	类=1	f_{11}	f_{10}
	类=0	f_{01}	f_{00}

使用性能度量来衡量分类模型性能的信息，如**准确率**和**错误率**。

准确率=正确预测数/预测总数= （f_{11} +f_{00} ）/（f_{11}+f_{10}+f_{01}+f_{00} ）

错误率=错误预测数/预测总数= （f_{10} +f_{01} ）/（f_{11}+f_{10}+f_{01}+f_{00} ）

精准率=正确预测为1的数/正确预测为1的数+错误预测为1的数
= (f11/f11+f01)(查准率)

召回率 = 正确预测为1的数/正确预测为1的数+错误预测为0的数
= （f11/f11+f10）(查全率)

F1值是精确率和召回率的调和均值，即F1=2PR/(P+R)，相当于精确率和召回率的综合评价指标。

最近邻算法

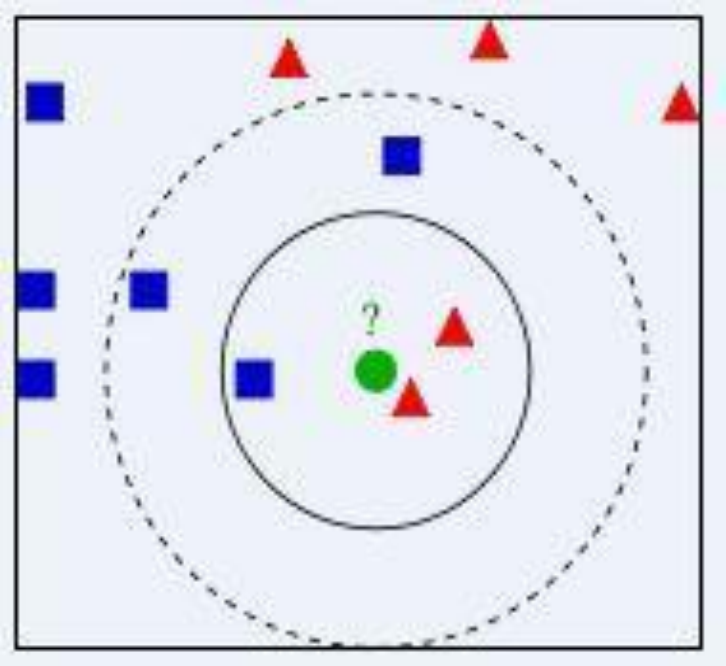
我们引出**最近邻算法**的定义：为了判定未知样本的类别，以全部训练样本作为代表点，计算未知样本与所有训练样本的距离，并以最近邻者的类别作为决策未知样本类别的唯一依据。（P38 特征空间划分图）

最近邻算法的缺陷——对噪声数据过于敏感，为了解决这个问题，我们可以把未知样本周边的多个最近样本计算在内，扩大参与决策的样本量，以避免个别数据直接决定决策结果。由此，引进K-近邻（k nearest neighbor）算法。

KNN算法是用来干什么的

K-最近邻算法是最近邻算法的一个延伸。基本思路是：选择距离未知样本最近的K个样本，该K个样本大多数属于某一类型，则未知样本判定为该类型。

问题：有一个未知形状X(图中绿色的圆点)，如何判断X是什么形状?



右图中，绿色圆要被决定赋予哪个类，是红色三角形还是蓝色四方形？如果K=3，由于红色三角形所占比例为2/3，绿色圆将被赋予红色三角形那个类，如果K=5，由于蓝色四方形比例为3/5，因此绿色圆被赋予蓝色四方形类。

KNN算法基本描述

- k-近邻法：基本规则是，在所有N个样本中,找到测试样本的k(k<=N)个最近邻者，当k=1时，knn问题就变成了最近邻问题。其中各类别所占个数表示成k_i，i＝1，…，c。定义判别函数为：
  $g_i(x)=k_i$，i=1，2,…,c。
- k-近邻一般采用k为奇数，跟投票表决一样，避免因两种票数相等而难以决策。(适用于二分类)
- 多数表决决策规则为：

$$j = \arg\max_i g_i(\mathbf{x}), \quad i = 1,...,c$$

计算步骤如下：

1）算距离：给定测试对象，计算它与训练集中的每个对象的距离

2）找邻居：圈定距离最近的k个训练对象，作为测试对象的近邻

3）做分类：根据这k个近邻归属的主要类别，来对测试对象分类

距离度量

- **KNN算法中，距离如何定义？**就是如何度量邻居之间的相识度，也就是如何选取邻居的问题，我们知道相似性的度量方式在很大程度上决定了选取邻居的准确性，也决定了分类的效果，因为判定一个样本点的类别是要利用到它的邻居的，如果邻居都没选好，准确性就无从谈起。因此我们需要用一个量来定量的描述邻居之间的距离，也可以形象的表述为邻居之间的相似度，具体的距离度量方式有很多，不同的场合使用哪种需要根据不同问题具体探讨，如文本类型，一般用余弦相似度。

距离度量

曼哈顿距离(Manhattan Distance)

从名字就可以猜出这种距离的计算方法了。想象你在曼哈顿要从一个十字路口开车到另外一个十字路口，驾驶距离是两点间的直线距离吗？显然不是，除非你能穿越大楼。实际驾驶距离就是这个“曼哈顿距离”。而这也是曼哈顿距离名称的来源， 曼哈顿距离也称为**城市街区距离(City Block distance)**。

两个n维向量a(x11,x12,…,x1n)与 b(x21,x22,…,x2n)间的曼哈顿距离

$$d_{12} = \sum_{k=1}^{n} |x_{1k} - x_{2k}|$$

距离度量

欧氏距离(Euclidean Distance)

欧氏距离是最易于理解的一种距离计算方法，源自欧氏空间中两点间的距离公式。

两个n维向量a(x11,x12,…,x1n)与 b(x21,x22,…,x2n)间的欧氏距离：

$$d_{12} = \sqrt{\sum_{k=1}^{n} (x_{1k} - x_{2k})^2}$$

距离度量

切比雪夫距离 (Chebyshev Distance)

国际象棋的玩法。国王走一步能够移动到相邻的8个方格中的任意一个。那么国王从格子(x1,y1)走到格子(x2,y2)最少需要多少步？自己走走试试。你会发现最少步数总是max(| x2-x1 | , | y2-y1 |) 步 。有一种类似的一种距离度量方法叫切比雪夫距离。

两个n维向量a(x11,x12,…,x1n)与 b(x21,x22,…,x2n)间的切比雪夫距离

$$d_{12} = \max_i(|x_{1i} - x_{2i}|)$$

这个公式的另一种等价形式是

$$d_{12} = \lim_{k\to\infty}\left(\sum_{i=1}^{n}|x_{1i} - x_{2i}|^k\right)^{1/k}$$

距离度量

闵可夫斯基距离(Minkowski Distance)

闵氏距离不是一种距离，而是一组距离的定义。

闵氏距离的定义

两个n维变量a(x11,x12,…,x1n)与 b(x21,x22,…,x2n)间的闵可夫斯基距离定义为：

$$d_{12} = \sqrt[p]{\sum_{k=1}^{n} |x_{1k} - x_{2k}|^p}$$

其中p是一个变参数。

当p=1时，就是曼哈顿距离

当p=2时，就是欧氏距离

当p→∞时，就是切比雪夫距离

根据变参数的不同，闵氏距离可以表示一类的距离。

KNN算法的实现步骤

算法步骤：

1：令k是最近邻数目，D是训练样例的集合

2：for 每个测试样例z do

3：　　计算z和每个训练样例之间的距离d

4：　　对d进行升序排序

5：　　取前k个训练样例的集合

6：　　统计K个最近邻样本中每个类别出现的次数

7：　　选择出现频率最大的类别作为未知样本的类别

8：　end for

Kd树

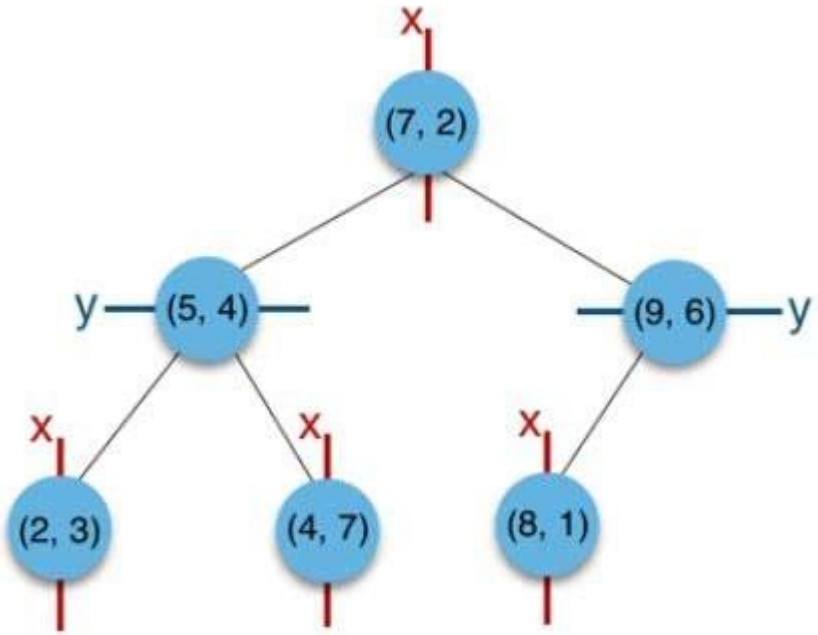
- kd树定义：
- kd树（k-dimensional树的简称），是一种对k维空间中的实例点进行存储以便对其进行快速搜索的二叉树结构。利用kd树可以省去对大部分数据点的搜索，从而减少搜索的计算量。

Kd树

- Kd树构造方法：
- kd 树是每个节点均为k维数值点的二叉树，其上的每个节点代表一个超平面，该超平面垂直于当前划分维度的坐标轴，并在该维度上将空间划分为两部分，一部分在其左子树，另一部分在其右子树。即若当前节点的划分维度为d，其左子树上所有点在d维的坐标值均小于当前值，右子树上所有点在d维的坐标值均大于等于当前值，本定义对其任意子节点均成立。

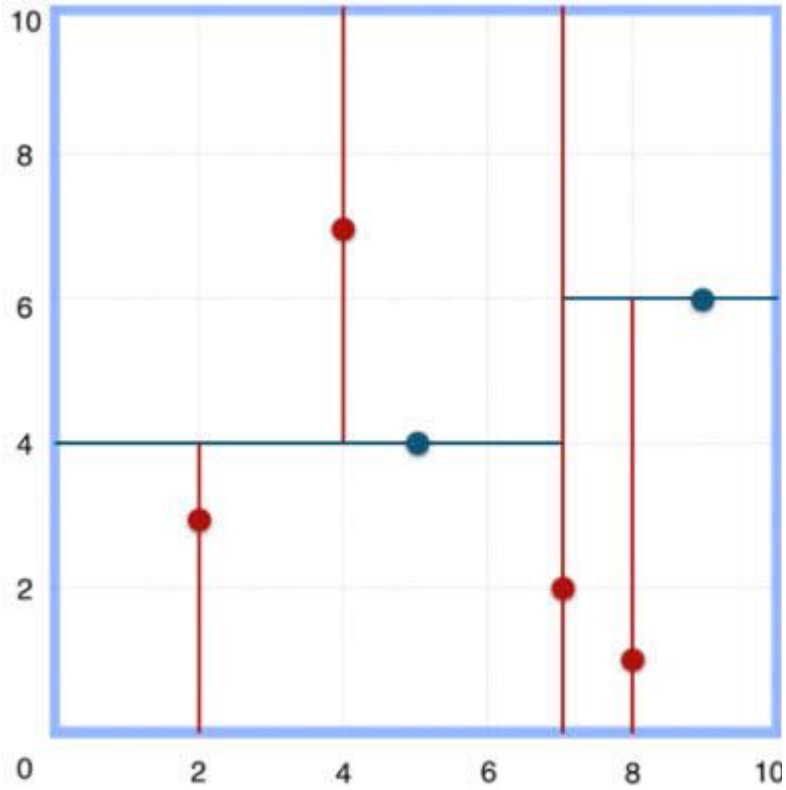
Kd树

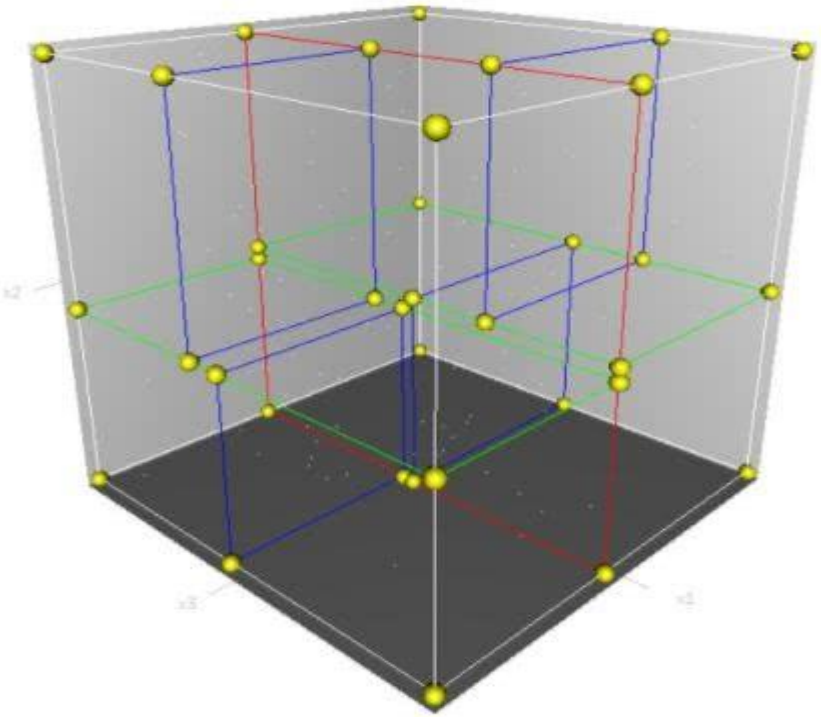
- 图3.4
- 集合：{(2,3)，(5,4)，(9,6)，(4,7)，(8,1)，(7,2)}。
- 升序排序：(2,3)，(4,7)，(5,4)，(7,2)，(8,1)，(9,6)



Kd树

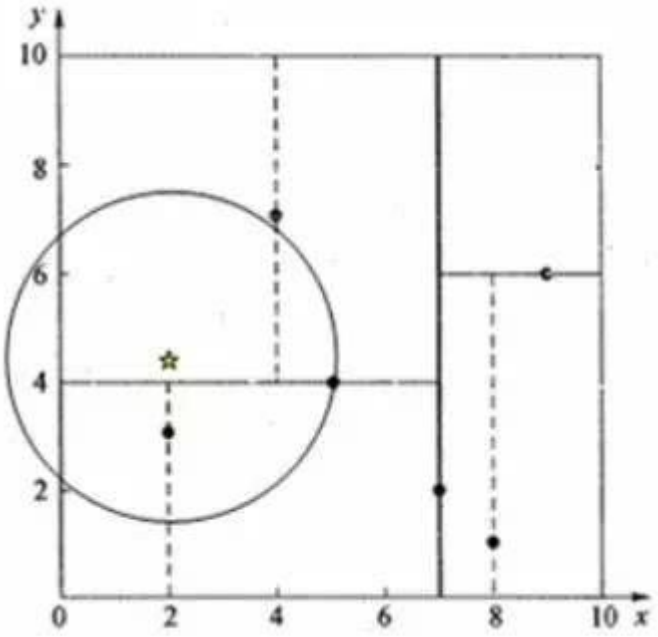
- 特征空间划分示例

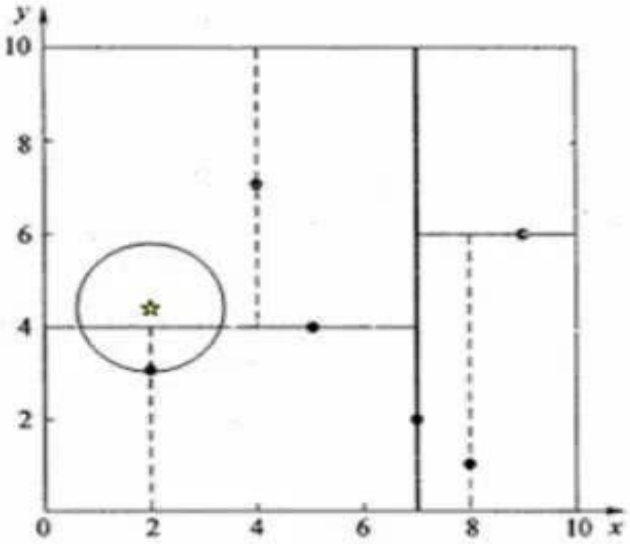




Kd树

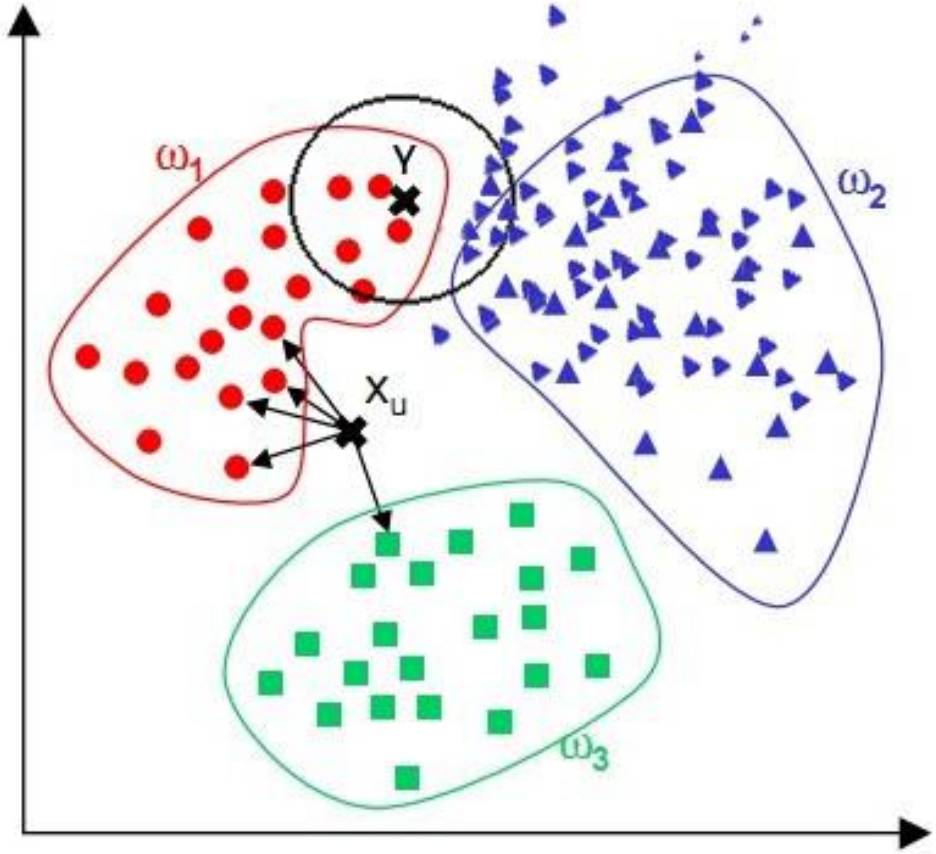
- 首先假设（4,7）为当前最近邻点，计算其与目标查找点的距离为3.202。回溯到（5,4），计算其与查找点之间的距离为3.041，小于3.202，所以“当前最近邻点”变成（5,4）。
- 以目标点（2,4.5）为圆心，以目标点（2,4.5）到“当前最近邻点”（5,4）的距离（即3.041）为半径作圆，如上图所示。可见该圆和y = 4超平面相交，所以需要进入（5,4）左子空间进行查找，即回溯至（2,3）叶子节点
- （2,3）距离（2,4.5）比（5,4）要近，所以“当前最近邻点”更新为（2,3），最近距离更新为1.5。
- 回溯至（7,2），以（2,4.5）为圆心1.5为半径作圆，并不和x = 7分割超平面交割，如下图所示。至此，搜索路径回溯完。返回最近邻点（2,3），最近距离1.5。





KNN算法的缺陷

观察下面的例子，我们看到，对于未知样本X，通过KNN算法，我们显然可以得到X应属于红点，但对于未知样本Y，通过KNN算法我们似乎得到了Y应属于蓝点的结论，而这个结论直观来看并没有说服力。



KNN算法的具体实现

由上面的例子可见：该算法在分类时有个重要的不足是，当样本不平衡时，即：一个类的样本容量很大，而其他类样本数量很小时，很有可能导致当输入一个未知样本z=(**x**',y')时，该样本的K个邻居中大数量类的样本占多数。 但是这类样本并不接近目标样本，而数量小的这类样本很靠近目标样本。这个时候，我们有理由认为该未知样本属于数量小的样本所属的一类，但是，KNN却不关心这个问题，它只关心哪类样本的数量最多，而不去把距离远近考虑在内，**因此，我们可以采用权值的方法来改进。和该样本距离小的邻居权值大，和该样本距离大的邻居权值则相对较小，由此，将距离远近的因素也考虑在内，避免因一个样本过大导致误判的情况。**

●距离加权表决： $w_i = 1/d(\mathbf{x}', \mathbf{x}_i)^2$

$$j = \arg\max_i w_i \times g_i(\mathbf{x}), \quad i = 1, \ldots, c$$

KNN算法几大问题

1、k值设定为多大？

k太小，分类结果易受噪声点影响；k太大，近邻中又可能包含太多的其它类别的点。k值通常是采用交叉检验来确定。

经验规则：k一般低于训练样本数的平方根

(a) Neighborhood too small.　(b) Neighborhood just right.　(c) Neighborhood too large.

图 5

KNN算法几大问题

2、类别如何判定最合适？

投票法没有考虑近邻的距离的远近，距离更近的近邻也许更应该决定最终的分类，所以加权投票法更恰当一些。

3、如何选择合适的距离衡量？

高维度对距离衡量的影响：众所周知当变量数越多，欧式距离的区分能力就越差。变量值域对距离的影响：值域越大的变量常常会在距离计算中占据主导作用，因此应先对变量进行标准化。

4、训练样本是否要一视同仁？

在训练集中，有些样本可能是更值得依赖的。可以给不同的样本施加不同的权重，加强依赖样本的权重，降低不可信赖样本的影响。

谢谢大家！