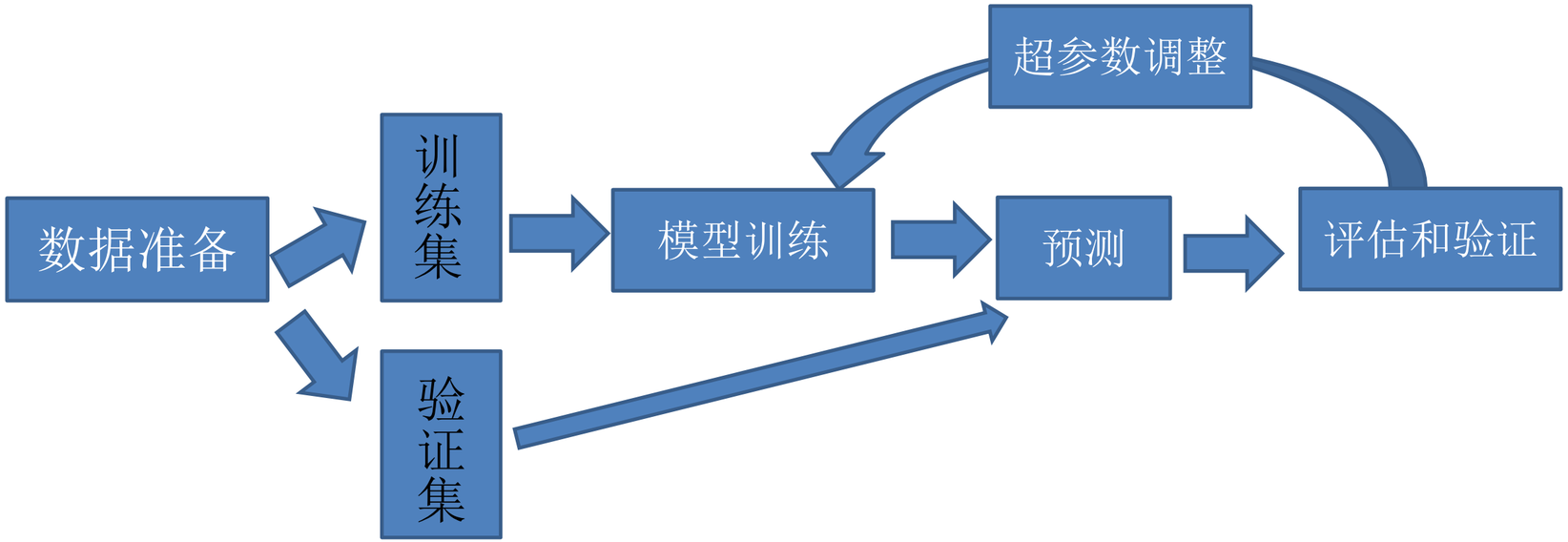
典型的分类过程



所有分类方法都是先建模，后预测的吗？

“惰性”学习法(lazy learning)：存储训练集，等测试样本来了才建模和预测，没有独立的模型训练过程！

第五章 分类

5.1 模型评估和性能度量
5.2 决策树
5.3 贝叶斯分类
5.4 k最近邻分类
5.5 组合分类

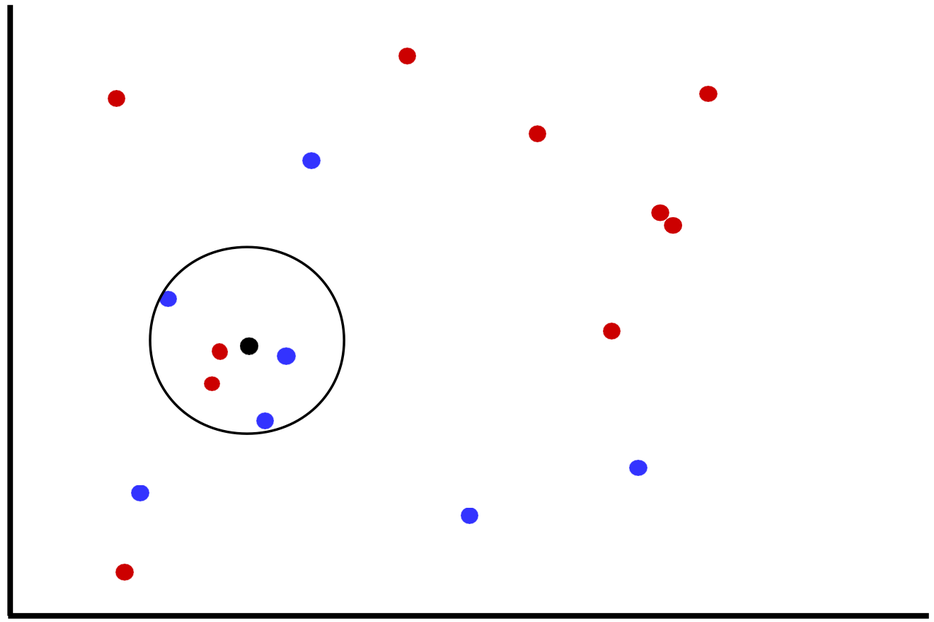
k最近邻分类

k-nearest neighbor(k-NN)

k-NN分类算法的两个基本步骤：

1. 对某个测试样本，在训练集中找到k个离其最近的训练样本，即k个近邻；

2. 找出这k个近邻中出现次数最多的那个类标签作为该测试样本标签的预测结果。



给定测试点（黑色的点），当k=5时，k-NN对该测试点的预测结果为：蓝色。

k-NN算法的特点

- k-NN算法仍然为有监督的学习算法；
- 它属于“惰性”学习算法，即不会预训练一个分类器或预测模型，而是将模型的构建与未知数据类别的预测同时进行。

K-NN算法不仅可以对离散因变量（y对应类别）预测来进行分类，也可以对连续因变量做预测（y是连续值，为回归问题），怎么做？

决定k-NN分类算法的关键

关键1：哪些才是近邻，即如何衡量相似度？

常用的相似度/距离度量包括：欧式距离、 余弦相似度等。

 计算相似度之前可能需要进行特征规范化。

算法实现需要考虑的问题

一般需要考虑算法的训练和测试阶段需要的计算量（时间复杂度）、内存需求等。

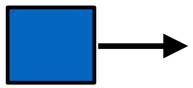
- k-NN不需要训练，所以没有训练时间；
- 测试阶段，需要计算测试样本与n个训练样本之间的相似度，当n很大时这可能会很慢；

总结：当应用于规模大、响应快、以及计算资源有限的应用时，需要寻求近邻的快速实现算法，比如对稀疏数据可以先建立索引。

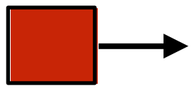
对稀疏数据建立索引

假设测试样本只有两个特征的值大于0，左图中的红色和蓝色。

要从训练集中找到与该测试样本相近的训练样本，只要从这两个特征至少有一个的值是大于0的训练样本中找。



$[x_1, x_3, x_9]$



$[x_2, x_4, x_5]$

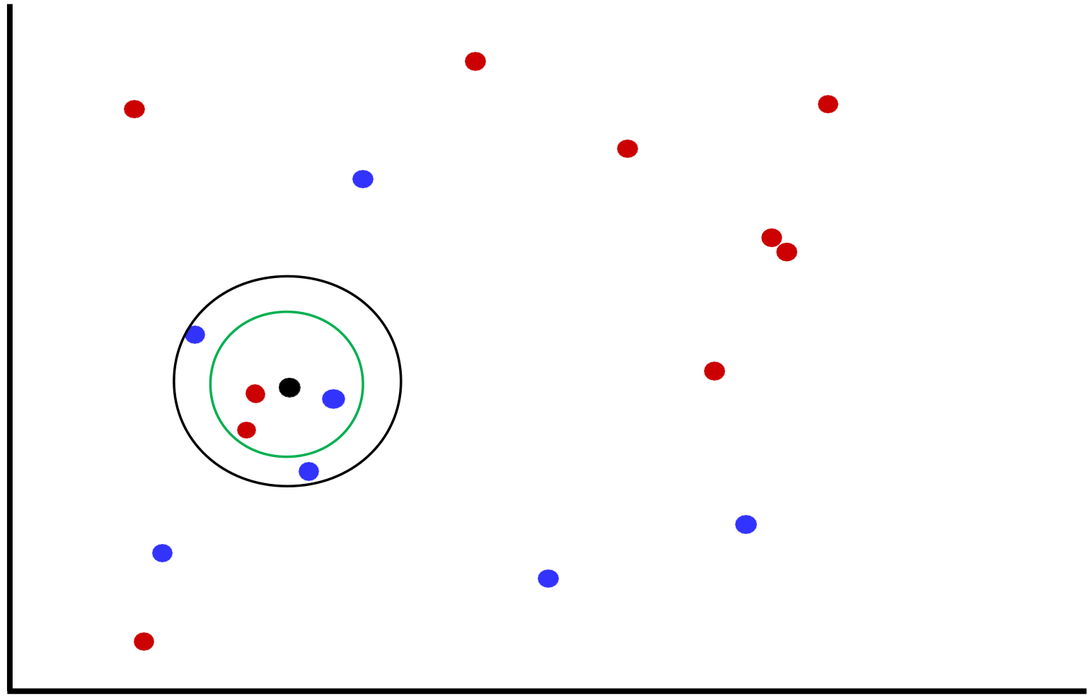
我们对每个特征建立一个对应训练样本的索引。如左图表示蓝色和红色特征分别不为0的训练样本。

通过以上特征-样本的索引，只要计算这6个训练样本与测试样本之间的相似度，而不是整个训练集中的样本。

以上方法利用了数据的稀疏性，比如文本数据

决定k-NN分类算法的关键

关键2：怎么设置近邻的个数，即k取多少？



给定测试点（黑色的点），

当k=5时，预测结果为：蓝色。

当k=3时，预测结果为：红色。

k取值对k-NN算法的影响

- **k太小**：容易出现过拟合，使得结果对训练集中的噪声很敏感；

思考：如果k=1时，测试样本的标签预测值取决于什么？

- **k太大**：模型太简单（欠拟合），过于平滑，不能有效反映数据集中类别之间的差别。

思考：如果k=n（训练样本个数），测试样本的标签被预测成什么？

k的取值

一般规则：

- 为了避免（二分类）出现类别一样多的情况，k一般取奇数；
- 对于数据规模大、结构复杂的情况，k一般取得大一点；对较小的数据集，k的取值得相对小。

具体操作：

从k=1开始，逐渐增大k的值，一般不超过20。最后采用在验证集上得到准确率最高的那个k值。

课外拓展

1. 近似近邻搜索问题：

给定集合χ，对$q \in \chi$，寻找另外一个点$\mathrm{p} \in \chi$，使其满足

$$d(q,p) \leq (1+c)d(q,h)$$

其中h是χ中距离q最近的点。

2. 基于哈希的方法：
局部敏感哈希-Locality Sensitivity Hashing