第 7 章　机器学习（流程）

教材： 王万良《人工智能导论》（第4版）

https://www.icourse163.org/course/ZJUT-1002694018

社区资源： https://github.com/Microsoft/ai-edu

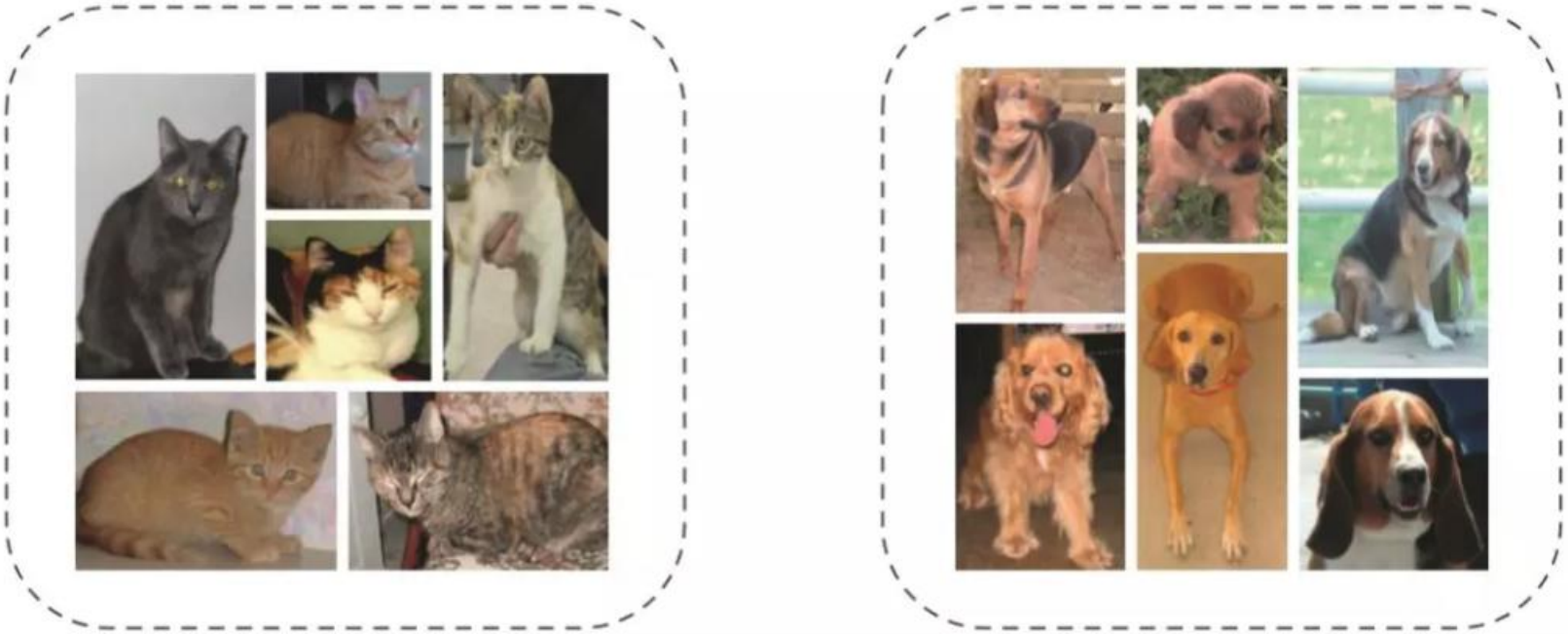
参考： 海豚大数据及人工智能实验室

机器学习流程：以猫狗识别为例

- 在机器学习领域，这种以教计算机区分不同类别事物为目标的任务被称为分类。
- 机器学习流程
  - 收集数据
  - 设计特征
  - 训练模型
  - 测试模型
  - 改进模型

机器学习流程：以猫狗识别为例

- 收集数据
  - 更大和更多样化的训练集可使计算机（或人）更好地完成学习任务



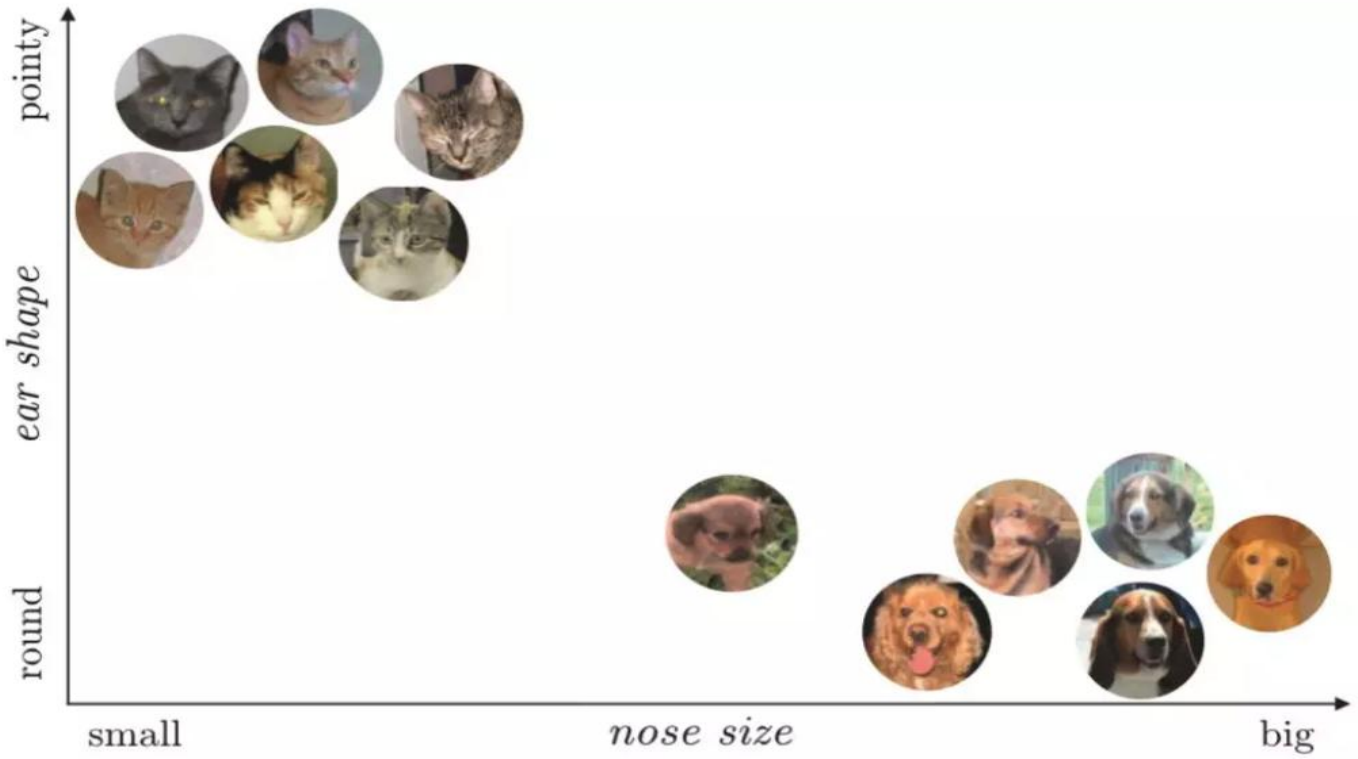
▲图1-1 包含六只猫（左图）和六条狗（右图）的训练集，这个数据集用来训练区分猫和狗图片的机器学习模型

机器学习流程：以猫狗识别为例

- 设计特征
  - 需要提供设计合理的特征，或者更理想情况下，让计算机自己找到这样的特征。
  - 设计高质量的特征非常依赖于应用（“腿的数量”）
  - 从训练集中提取特征也非常具有挑战性（图像的清晰度）

机器学习流程：以猫狗识别为例

- 提取特征
  - 鼻子的大小，相对于头的大小（从小到大）
  - 耳朵的形状（从圆到尖）



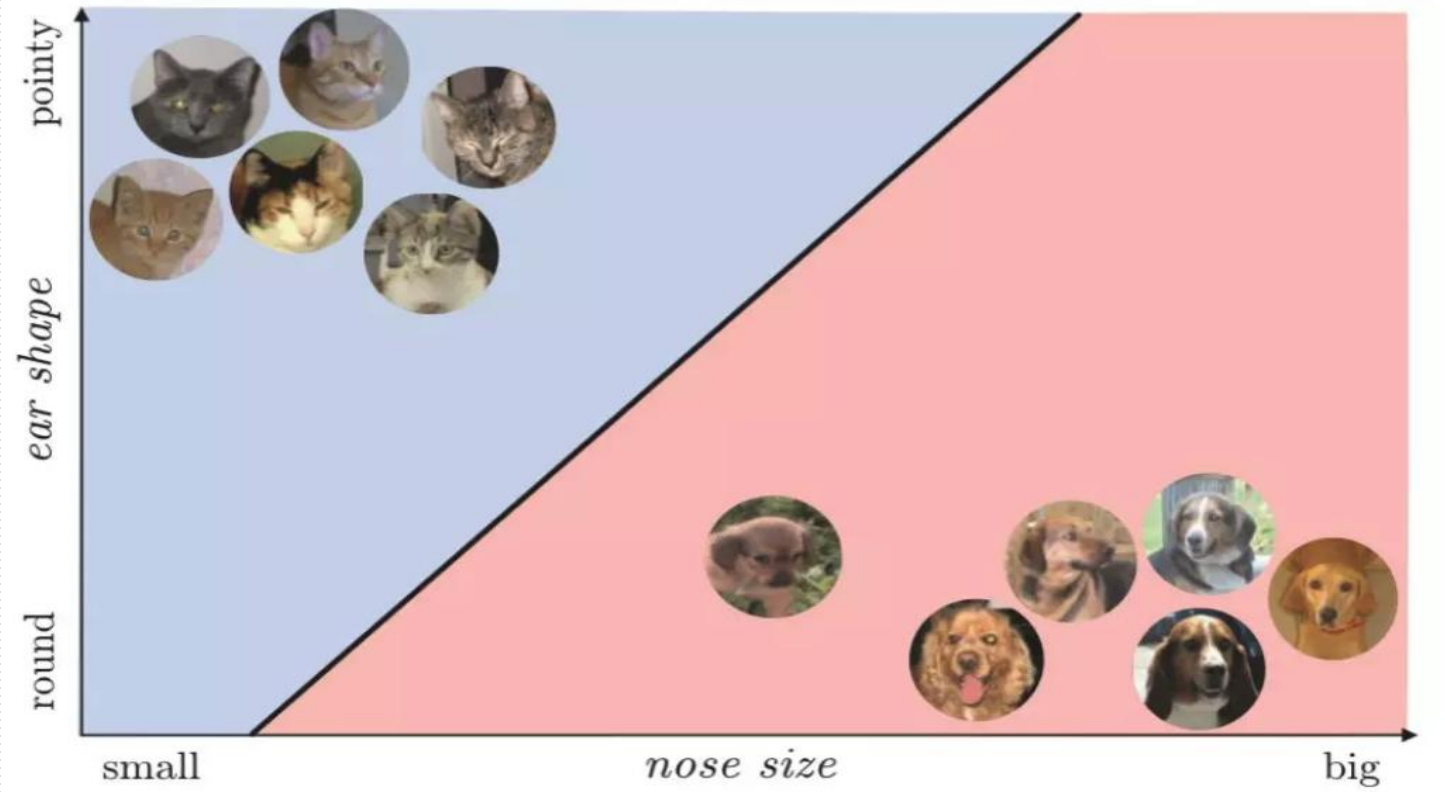
▲图1-2 训练集的特征空间表示，其中水平和垂直坐标分别表示“鼻子大小”特征和“耳朵形状”特征，训练集中的猫和狗分别处于特征空间的不同位置，这说明特征选得很合适

机器学习流程：以猫狗识别为例

- 训练模型
  - 让计算机在我们精心设计的特征空间中找到能够区分猫和狗的一条直线或者一个线性模型
  - 直线（在二维空间中）有斜率和截距两个参数，这意味着要为这两个参数找到正确的值。直线的参数必须根据训练数据（的特征表示）来确定。
  - 确定参数的过程依赖于一组名为数值优化的工具，此过程被称作模型的训练。

机器学习流程：以猫狗识别为例

- 一个已训练好的线性模型，它将特征空间分成猫和狗两个区域。



▲图1-3 一个已训练好的线性模型完美地将训练集中的两类动物区分开来：如果将来任何新图片的特征表示落入这条线之上（左上区域），该图片就会被归类为猫；如果落入这条线之下（右下区域），该图片就会被归类为狗

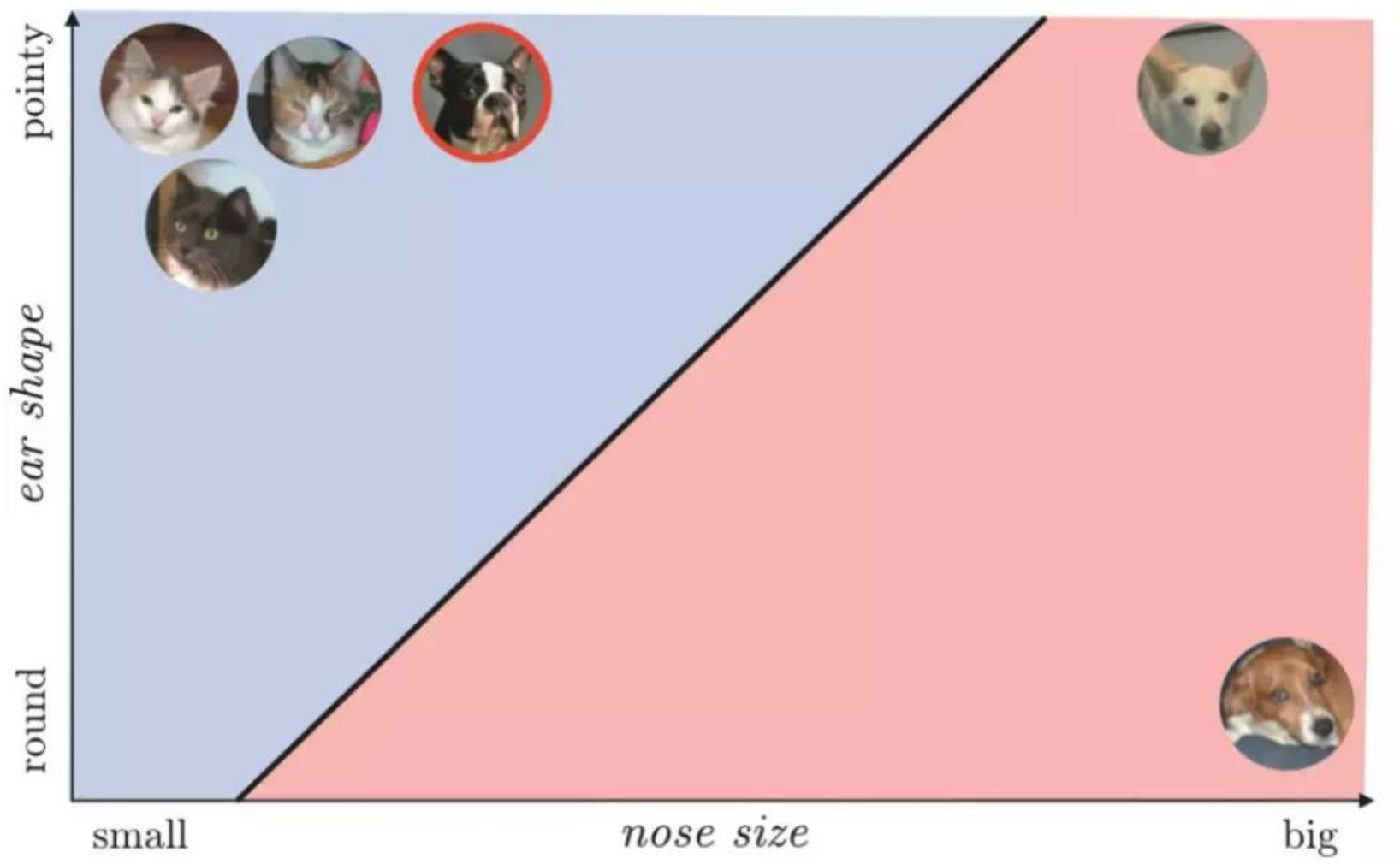
机器学习流程：以猫狗识别为例

测试模型

- 给计算机提供一些以前没有见过的猫和狗的图片（一般称为数据的测试集），然后看看它对每幅图片中动物的识别能力

▲图1-4 猫和狗图片的测试集，注意，其中的一条狗，也就是右上方的波士顿，有小鼻子和尖耳朵，根据我们选择的特征表示，计算机会认为这是一只猫

机器学习流程：以猫狗识别为例



▲图1-5 用我们已训练好的线性模型来识别测试图片（的特征表示），注意，由于波士顿像训练集中的猫一样拥有尖耳朵和小鼻子，因此它被误分类成一只猫

机器学习流程：以猫狗识别为例

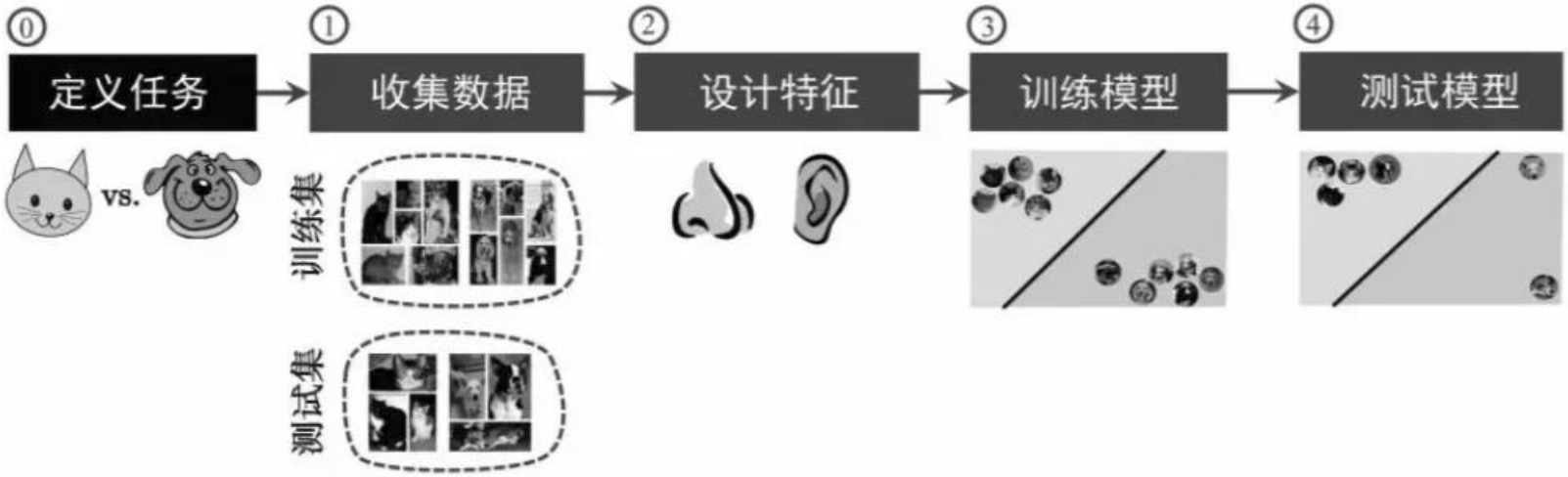
- 测试模型
  - 误分类完全是因为选择的特征
    - 这些特征是根据图1-1中的训练集设计的。这只狗的特征是小鼻子和尖耳朵，恰好和训练集中的猫相匹配，所以被误分类
    - 由于训练集太小且不够多样化，基于训练集选择出来的特征是不能完全有效的。

机器学习流程：以猫狗识别为例

- 改进模型——提升学习器性能
  - 首先，应该收集更多的数据，组成一个庞大且多样的训练集。
  - 其次，需要考虑设计更具有辨识性的特征（比如，眼睛的颜色、尾巴的形状等）来进一步帮助我们区分猫和狗。
  - 最后，还要用设计的特征训练一个新的模型，并用同样的方式来测试，看它与原来的模型相比是否有所改进

经典机器学习问题的流程

- 解决机器学习问题的一般流程

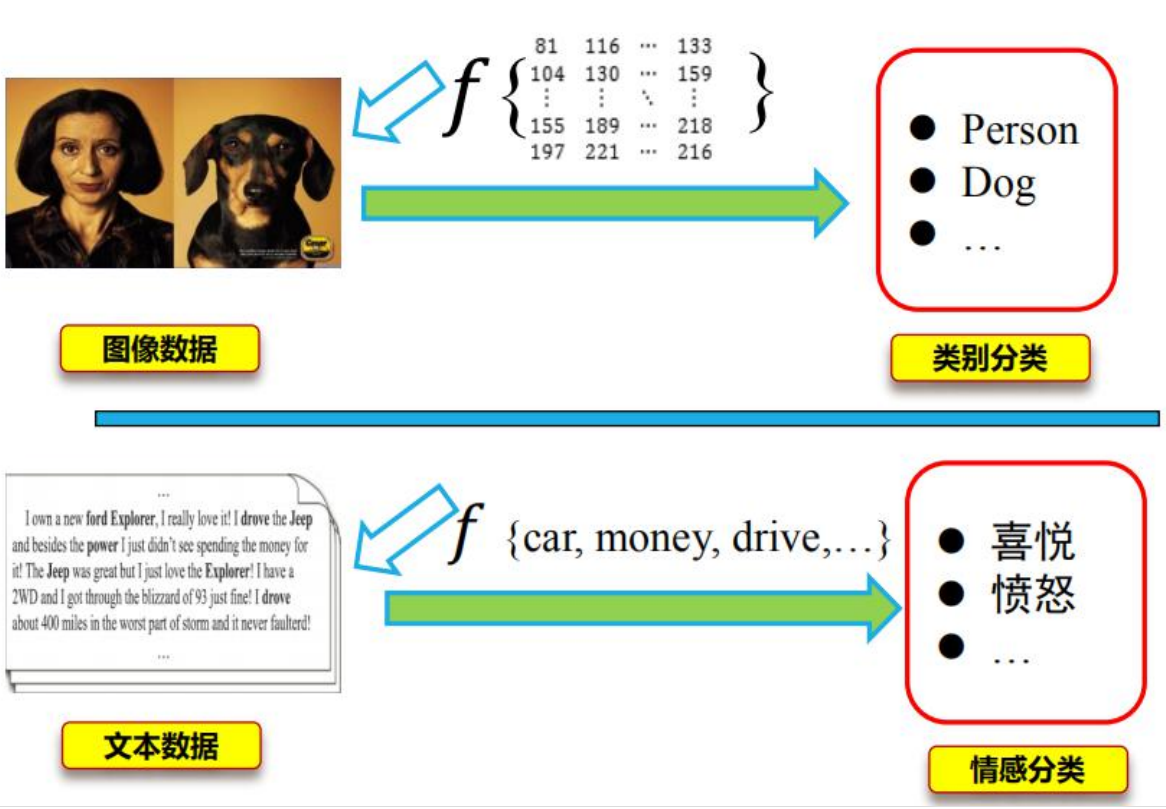


▲图1-6 猫和狗分类问题的学习流程，相同的一般化流程基本上可用于所有的机器学习问题

经典机器学习问题的流程

- 解决机器学习问题的一般流程
  - 定义问题：我们想教计算机做什么任务？
  - 收集数据：为训练集和测试集收集数据。数据越大、越多样越好。
  - 设计特征：什么样的特征最能描述数据？
  - 训练模型：用数值优化技术在训练集上调整恰当模型的参数。
  - 测试模型：评估训练模型在测试数据上的性能。如果评估结果不佳，则重新考虑所使用的特征，并尽可能收集更多的数据。

机器学习: 从数据中学习映射函数



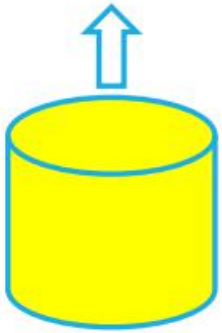
- 原始数据中提取特征
- 学习映射函数f
- 通过映射函数f将原始数据映射到语义空间，即寻找数据和任务目标之间的关系

监督学习的重要元素

监督学习：损失函数

训练映射函数f

使得$f(x_i)$预测结果尽量等于y_i



训练数据集

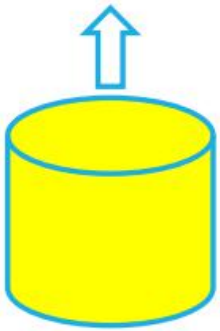
$(x_i, y_i), i = 1, \ldots, n$

- 训练集中一共有n个标注数据，第i个标注数据记为(x_i, y_i),其中第i个样本数据为x_i，y_i是x_i的标注信息。
- 从训练数据中学习得到的映射函数记为f，f对x_i的预测结果记为$f(x_i)$。损失函数就是用来计算x_i真实值y_i与预测值$f(x_i)$之间差值的函数。
- 很显然，在训练过程中希望映射函数在训练数据集上得到“损失”之和最小，即$\min \sum_{i=1}^{n} Loss(f(x_i), y_i)$。

监督学习：损失函数

训练映射函数f

使得$f(x_i)$预测结果尽量等于y_i



训练数据集

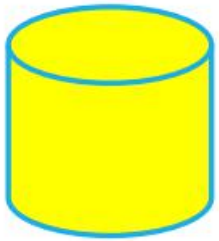
$(x_i, y_i), i = 1, \ldots, n$

损失函数名称	损失函数定义
0-1损失函数	$Loss(y_i, f(x_i)) = \begin{cases} 1, f(x_i) \neq y_i \\ 0, f(x_i) = y_i \end{cases}$
平方损失函数	$Loss(y_i, f(x_i)) = (y_i - f(x_i))^2$
绝对损失函数	$Loss(y_i, f(x_i)) = \lvert y_i - f(x_i) \rvert$
对数损失函数/对数似然损失函数	$Loss(y_i, P(y_i \lvert x_i)) = -logP((y_i \lvert x_i)$

典型的损失函数

监督学习：训练数据与测试数据

从训练数据集学习
得到映射函数f

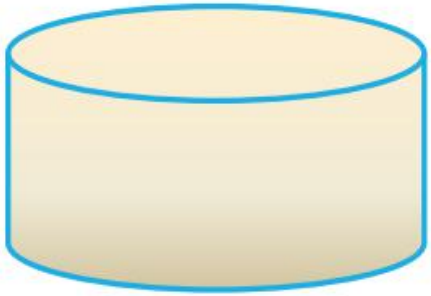


训练数据集
$(x_i, y_i), i = 1, \ldots, n$

在测试数据集
测试映射函数f

测试数据集
$(x_i', y_i'), i = 1, \ldots, m$

未知数据集
上测试映射函数f



监督学习：经验风险与期望风险

- 经验风险(empirical risk)
  - 训练集中数据产生的损失。经验风险越小说明学习模型对训练数据拟合程度越好。

- 期望风险(expected risk):
  - 当测试集中存在无穷多数据时产生的损失。期望风险越小，学习所得模型越好

监督学习：经验风险与期望风险

- 映射函数训练目标：经验风险最小化(empirical risk minimization, ERM)

$$\min_{f\in\Phi}\frac{1}{n}\sum_{i=1}^{n}Loss(y_i,f(x_i))$$

  - 选取一个使得训练集所有数据损失平均值最小的映射函数。这样的考虑是否够？

- 映射函数训练目标：期望风险最小化(expected risk minimization)

$$\min_{f\in\Phi}\frac{1}{n}\int_{x\times y}Loss\big(y,f(x)\big)P(x,y)dxdy$$

  - 期望风险是模型关于联合分布期望损失，经验风险是模型关于训练样本集平均损失。
  - 根据大数定律，当样本容量趋于无穷时，经验风险趋于期望风险。所以在实践中很自然用经验风险来估计期望风险。
  - 由于现实中训练样本数目有限，用经验风险估计期望风险并不理想，要对经验风险进行一定的约束

监督学习:“过学习(over-fitting)”与“欠学习(under-fitting)

- 经验风险最小化

$$\min_{f\in\Phi}\frac{1}{n}\sum_{i=1}^{n}Loss(y_i,f(x_i))$$

- 期望风险最小化

$$\min_{f\in\Phi}\frac{1}{n}\int_{x\times y}Loss\big(y,f(x)\big)P(x,y)dxdy$$

经验风险小（训练集上表现好）	期望风险小（测试集上表现好）	泛化能力强
经验风险小（训练集上表现好）	期望风险大（测试集上表现不好）	过学习（模型过于复杂）
经验风险大（训练集上表现不好）	期望风险大（测试集上表现不好）	欠学习
经验风险大（训练集上表现不好）	期望风险小（测试集上表现好）	“神仙算法”或“黄粱美梦”

监督学习:结构风险最小

- 经验风险最小化，仅反映了局部数据

$$\min_{f\in\Phi}\frac{1}{n}\sum_{i=1}^{n}Loss(y_i,f(x_i))$$

- 期望风险最小化，无法得到全量数据

$$\min_{f\in\Phi}\frac{1}{n}\int_{x\times y}Loss\big(y,f(x)\big)P(x,y)dxdy$$

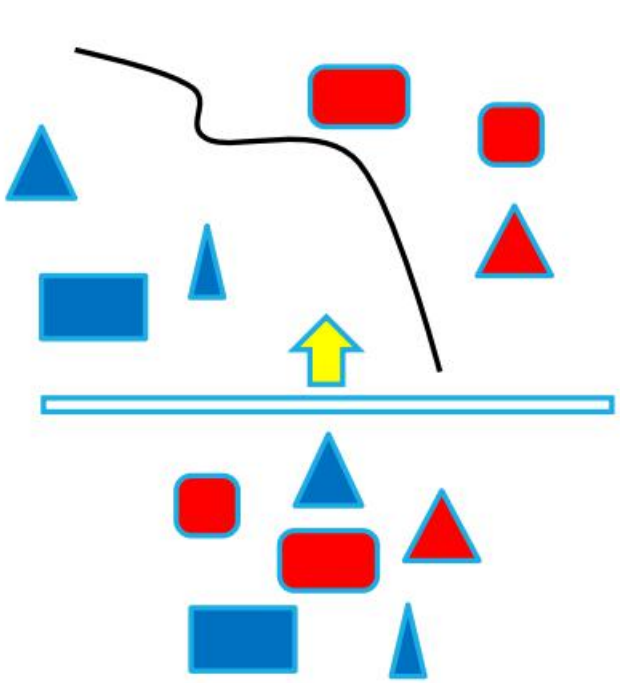
结构风险最小化(structural risk minimization)：为了防止过拟合，在经验风险上加上表示模型复杂度的正则化项(regulatizer)或惩罚项(penalty term)：

$$\min_{f\in\Phi}\frac{1}{n}\sum_{i=1}^{n}Loss(y_i,f(x_i)) + \lambda J(f)$$

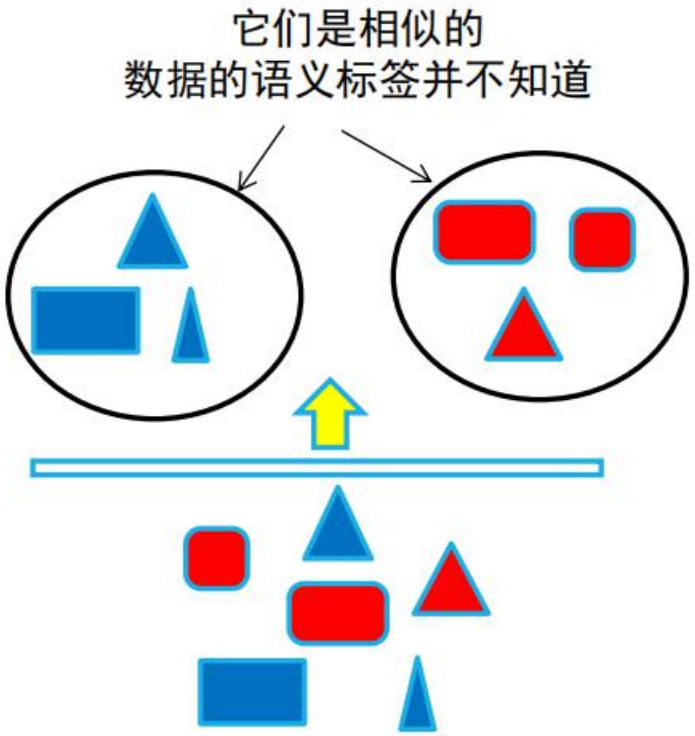
经验风险　　模型复杂度

在最小化经验风险与降低模型复杂度之间寻找平衡

监督学习 versus 无监督学习



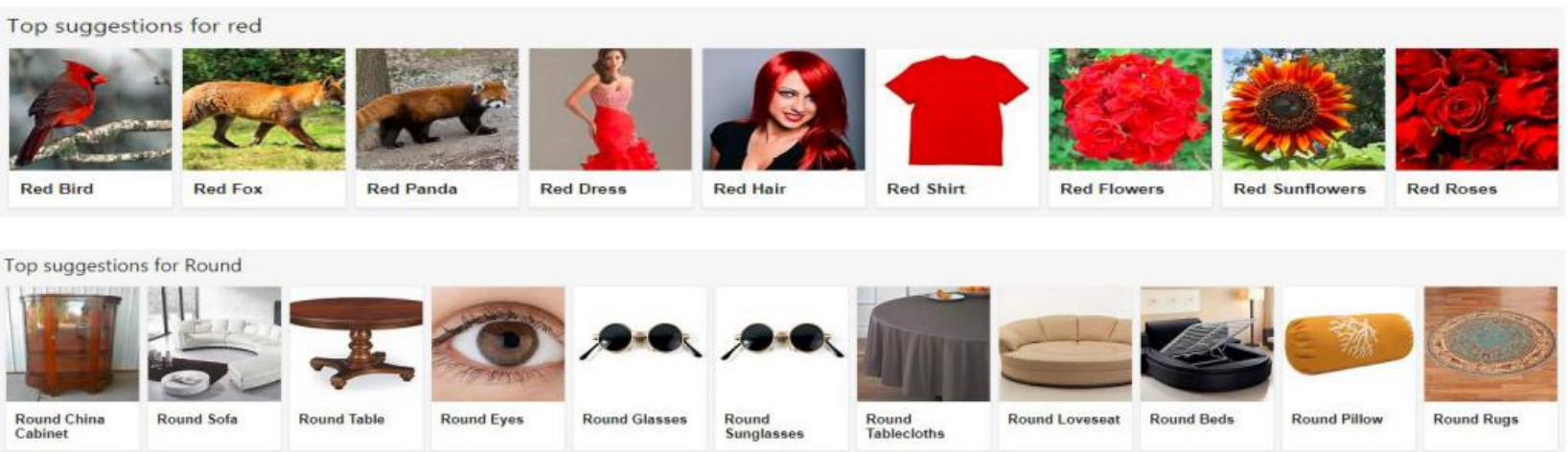
左：监督学习



右：无监督学习

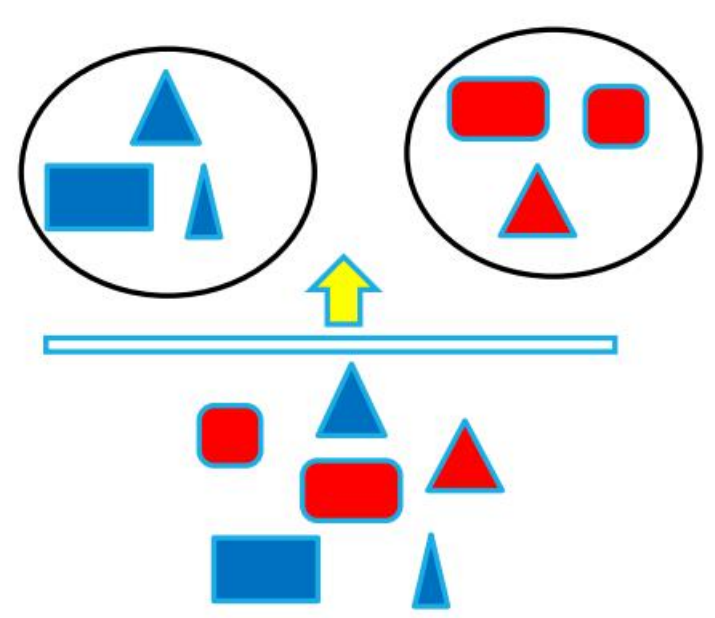
无监督学习的重要因素

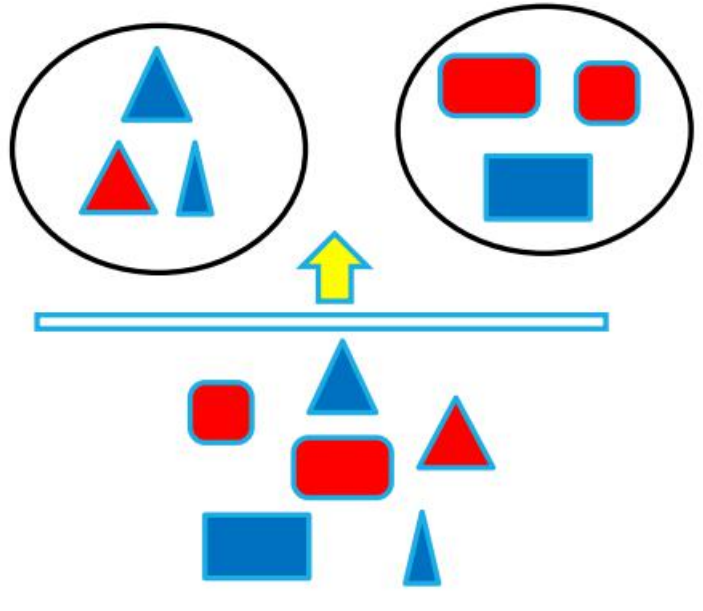
数据特征	图像中颜色、纹理或形状等特征	听觉信息中旋律和音高等特征	文本中单词出现频率等特征
相似度函数	定义一个相似度计算函数，基于所提取的特征来计算数据之间的相似性		



无监督学习：数据特征和相似度函数都很重要

- 相似度函数：颜色相似　　相似度函数：形状相似





THE END