人工智能：机器学习 I

饶洋辉
计算机学院，
中山大学
raoyangh@mail.sysu.edu.cn
http://cse.sysu.edu.cn/node/2471

课件来源：中山大学陈川副教授；海军工程大学贲可荣教授等

机器学习

- “Learning is any process by which a system improves performance from experience.” - Herbert Simon

- Tom Mitchell (1997)：**机器学习是指一个电脑程序从经验 E（如带标签数据）中学习如何提高某些任务 T（如分类）的性能指标 P（如精度和召回率），即以P作为指标，通过E来提高T的表现。**

机器学习

- 机器学习就是让计算机能够像人那样自动获取新知识，并在实践中不断地完善自我和增强能力，使得系统在下一次执行同样任务或类似的任务时，会比现在做得更好或效率更高。

- 机器学习的研究一方面可以使机器能自动获取知识，赋予机器更多的智能；另一方面可以进一步揭示人类思维规律和学习奥秘，帮助人们提高学习效率。机器学习的研究还会对记忆存储模式、信息输入方式及计算机体系结构产生重大影响。

机器学习的类型

- 有监督学习
  - 分类、回归
  - 给定：输入特征 + 输出（标签）
- 无监督学习
  - 聚类、关联分析
  - 给定：输入特征
- 混合学习
  - 对有监督学习和无监督学习的融合
- 强化学习
  - 只给出对当前输出的一个评价，而非具体的标签

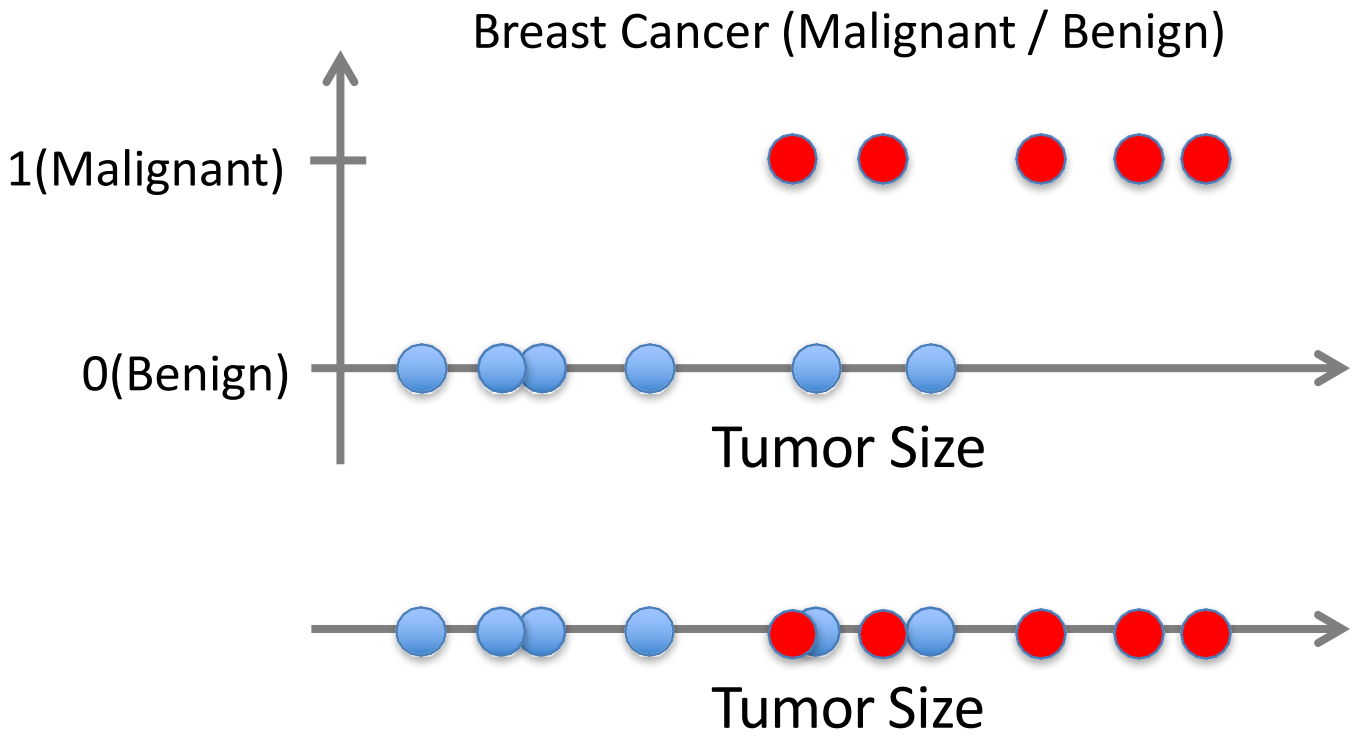
Based on slide by Pedro Domingos

有监督学习：分类

- 预测离散型变量
  - 首先基于一个包含x值（输入特征），以及离散型的真实y值（标签）的训练数据集构建分类模型；然后将该模型用来预测新的只包含x值的测试数据集的y值。

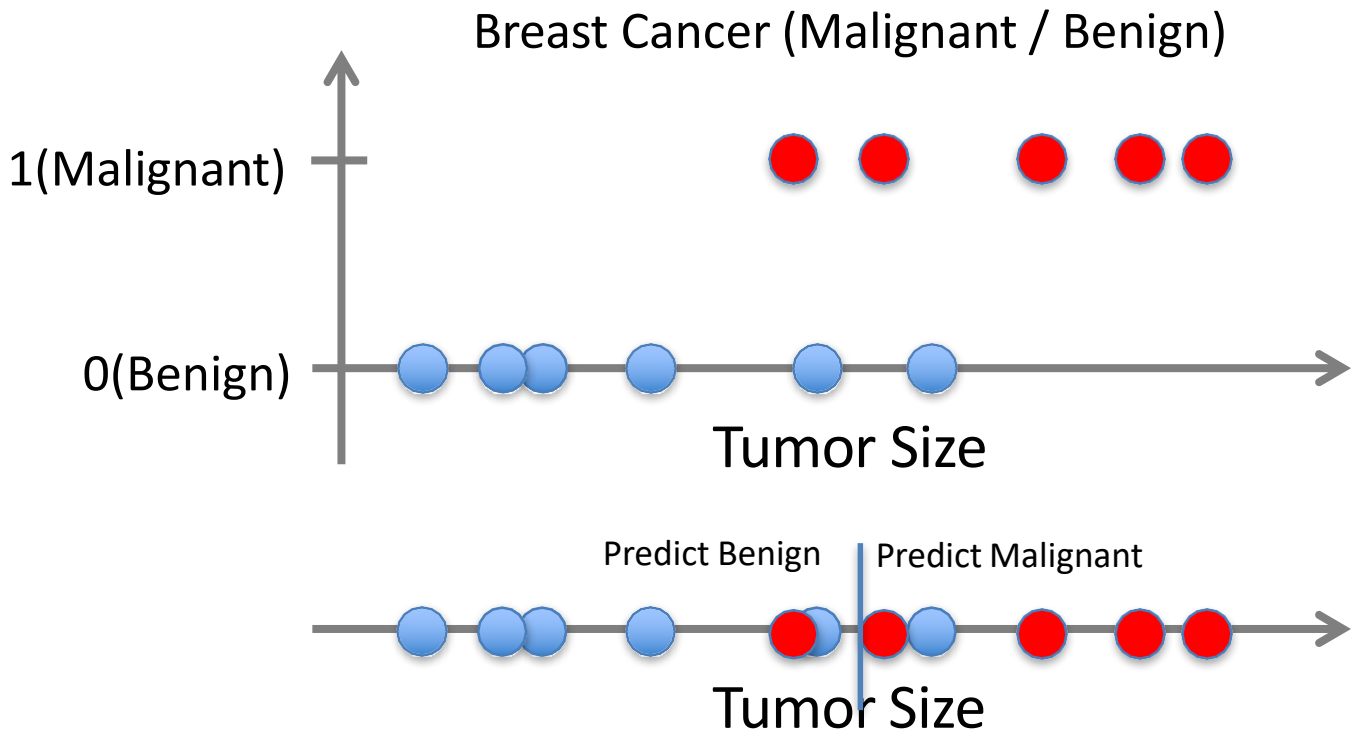
- 给定(x_1, y_1), (x_2, y_2), …, (x_n, y_n)，学习一个输入x、预测y的函数$f(x)$
  - y为离散型 == 分类

有监督学习：分类



Based on example by Andrew Ng

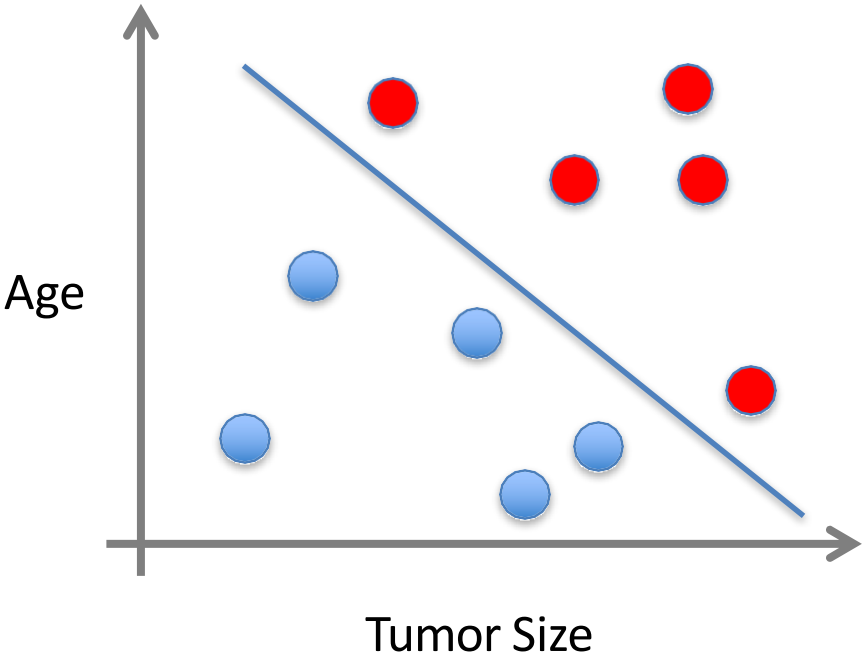
有监督学习：分类



Based on example by Andrew Ng

有监督学习：分类

- x可以是多维的，即每个样本的输入特征都是一个向量。



- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

…

Based on example by Andrew Ng

评测指标

- 准确率
- 速度
  - 构建分类模型的时间（训练速度）
  - 使用分类模型的时间（预测速度）
- 鲁棒性
  - 模型在处理噪音和缺失值方面的能力
- 可扩展性
  - 模型用在更大规模的数据集上的能力
- 可解释性

评测指标

评测指标

距离

- 向量 **x** 与 **y** 之间的欧式距离用如下公式计算:

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$$

其中

- n 是向量的维度
- x_k 和 y_k 是 **x** 与 **y** 的第 k 项元素

距离

- 欧式距离的衡量公式可以一般化成闵可夫斯基距离（*Minkowski* distance metric）

$$d(\mathbf{x},\mathbf{y}) = \left(\sum_{k=1}^{n} | x_k - y_k |^r \right)^{\frac{1}{r}}$$

- 闵可夫斯基距离的三个常见的例子：
  - r=1: 街区距离（City block distance, L_1 norm）
  - r=2: 欧式距离（Euclidean distance, L_2 norm）
  - r=∞: 极大距离（Supremum distance, L_{max} or L_∞ norm），该距离是两向量对应元素之间差距最大的距离

距离

$$x_i = (x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(n)})^{\mathrm{T}}$$

- *Lp*距离：

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

  - 欧式距离：

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

  - 曼哈顿距离：

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

  - *L*∞距离：

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

距离

- 四个向量的 x 与 y 坐标值如下：
  - p1 = <0, 2>
  - p2 = <2, 0>
  - p3 = <3, 1>
  - p4 = <5, 1>

距离

L_1	p1	p2	p3	p4
p1	0.0	4.0	4.0	6.0
p2	4.0	0.0	2.0	4.0
p3	4.0	2.0	0.0	2.0
p4	6.0	4.0	2.0	0.0

L_2	p1	p2	p3	p4
p1	0.0	2.8	3.2	5.1
p2	2.8	0.0	1.4	3.2
p3	3.2	1.4	0.0	2.0
p4	5.1	3.2	2.0	0.0

L_∞	p1	p2	p3	p4
p1	0.0	2.0	3.0	5.0
p2	2.0	0.0	1.0	3.0
p3	3.0	1.0	0.0	2.0
p4	5.0	3.0	2.0	0.0

k-近邻分类

- 随着人们生活水平不断的提高,红酒越来越受到人们的喜爱。红酒的产量越来越大，然而红酒品质鉴定的手段还是仅靠品酒师的人工品尝打分来判定红酒质量的好坏，显然这种鉴定方式难以满足当今市场的需求。现在有不少学者运用一些机器学习的算法来对红酒质量进行预测研究，使得红酒品质鉴定的速度得到大幅提升并且有着较高的准确率。

k-近邻分类

- 对于红酒品质的分类，可以基于红酒的理化指标(例如：酒精的浓度、pH值、糖的含量、非挥发性酸含量、 挥发性酸含量、柠檬酸含量等)作为特征，建立分类模型，然后对红酒品质进行预测。本案例中，我们将使用UCI数据库中的 Wine Quality Data Set 数据集，利用k-近邻分类算法来进行红酒品质的分类。

k-近邻分类

- 我们使用一份包含1599个样本的关于葡萄牙的Vinho Verde葡萄酒数据集。 每个样本包含12个变量，其中最后一个变量quality为预测变量。

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

- k-近邻分类算法：所有属性（变量）同等重要。
- k指的是选取的和待预测样本距离最近的训练样本数。

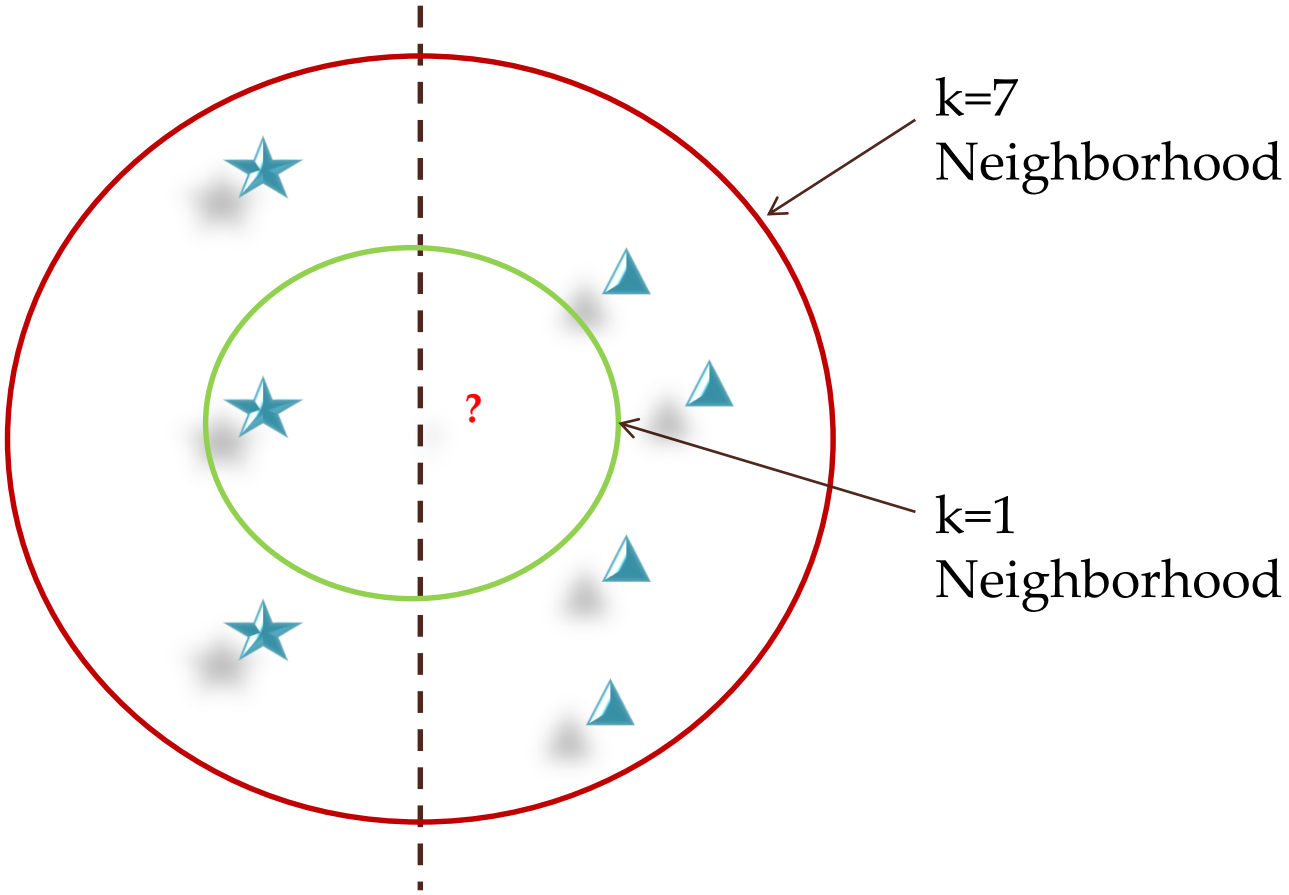
k-近邻分类

- 0、1、2、3为训练集
- 4为测试集
- k = 1, 2, 3, 4
- 采用街区距离

- d(4, 0) = 0.0
- d(4, 1) = 49.1
- d(4, 2) = 25.7
- d(4, 3) = 37.6

变量名称	含义说明
fixed acidity	非挥发性酸含量
volatile acidity	挥发性酸含量
citric acid	柠檬酸
residual sugar	糖含量
chlorides	氯化物
free sulfur dioxide	游离二氧化硫
total sulfur dioxide	总二氧化硫
density	密度
pH	酸碱度
sulphates	硫酸盐
alcohol	酒精浓度
quality	品质， 为预测变量

k-近邻分类



k-近邻分类

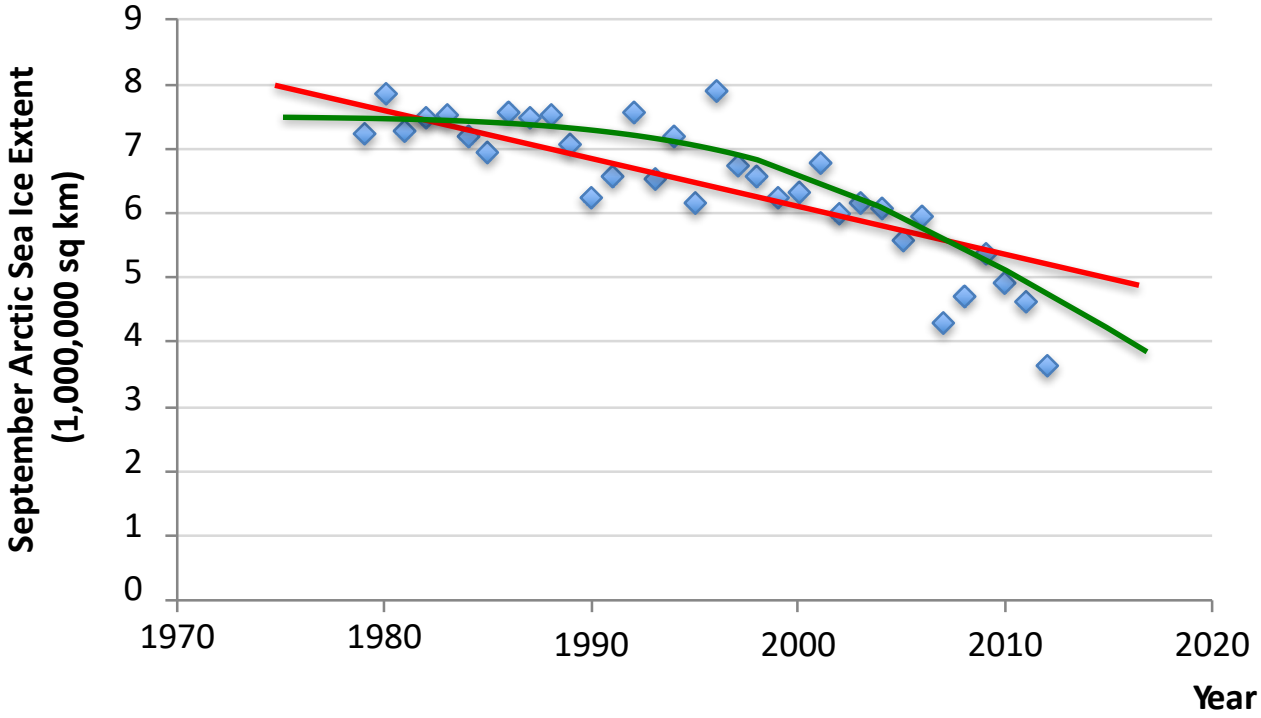
- 工作原理
  - 存在一个样本数据集合，也称作训练样本集，并且样本集中每个数据都存在标签，即我们知道样本集中每个数据与所属分类的对应关系。
  - 输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似数据（最近邻）的分类标签。
  - 一般来说，只选择样本数据集中前k个最相似的数据。k一般不大于20，最后，选择k个最近邻中出现次数最多的分类，作为新数据的分类。

k-近邻分类

- 优点
  - 精度高
  - 对异常值不敏感
  - 无数据输入假定
- 缺点
  - 时间复杂度高
  - 空间复杂度高
- 适用数据范围
  - 离散型和连续型

有监督学习：回归

- 给定(x_1, y_1), (x_2, y_2), …, (x_n, y_n)，学习一个输入x、预测y的函数$f(x)$
  - y为连续型 == 回归



Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

有监督学习：回归

- 线性回归：最小二乘法

$$n^{-1}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i) = 0$$

$$n^{-1}\sum_{i=1}^{n} x_i(y_i - w_0 - w_1 x_i) = 0$$

$$Q(w_0, w_1) = \min_{w_0, w_1} \sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2$$

$$\partial Q(w_0, w_1) / \partial w_0 = 0 \qquad \partial Q(w_0, w_1) / \partial w_1 = 0$$

$$-2\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i) = 0 \qquad -2\sum_{i=1}^{n} x_i(y_i - w_0 - w_1 x_i) = 0$$

有监督学习：回归

- 线性回归：最小二乘法

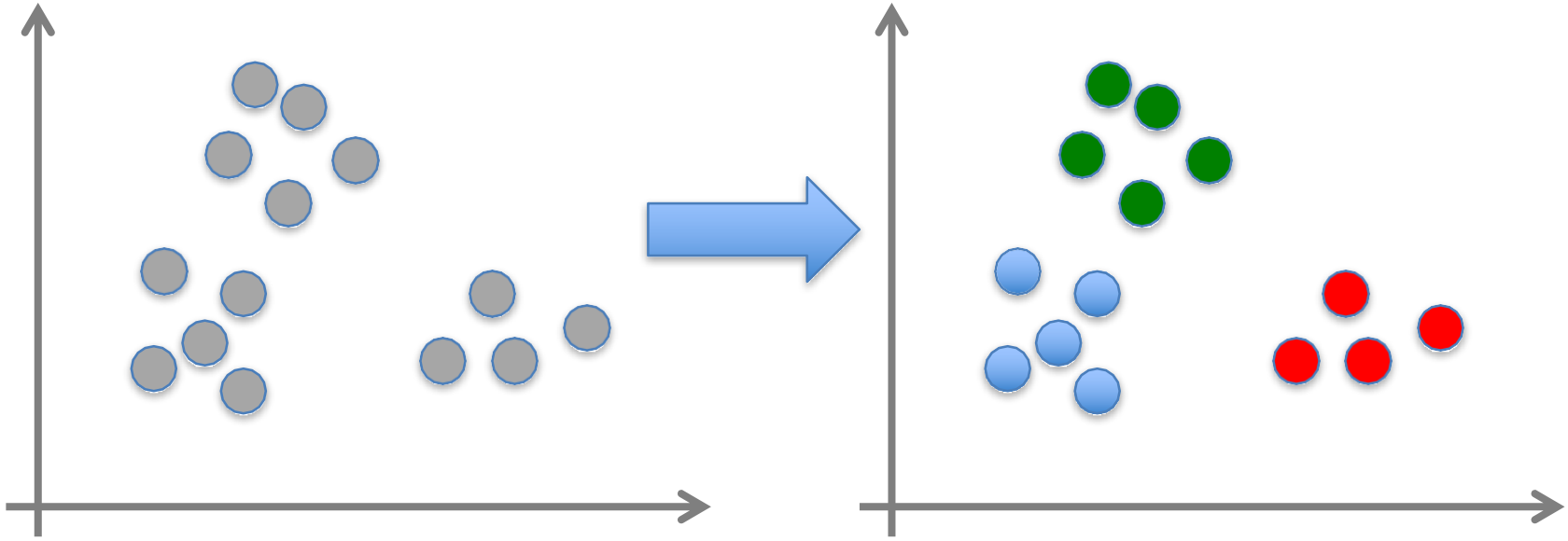
$$w_0 = \bar{y} - w_1 \bar{x}$$

$$w_1 = \frac{\sum_{i=1}^{n} x_i (y_i - \bar{y})}{\sum_{i=1}^{n} x_i (x_i - \bar{x})}$$

$$= \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

无监督学习

- 给定$x_1, x_2, \ldots, x_n$，无监督学习旨在输出x中的隐含结构。
- 无监督学习中的聚类示例：



混合学习

- 有监督学习具有分类精细、准确的优点，但学习过程复杂。
- 无监督学习具有分类灵活、算法简练的优点，但学习过程较慢。
- 混合学习过程一般事先用无监督学习抽取输入数据的特征，然后将这种内部表示提供给有监督学习进行处理，以达到输入输出的某种映射。由于对输入数据进行了预处理，将会使有监督学习以及整个学习过程加快。

强化学习

- 强化学习（Reinforcement Learning）就是智能系统从环境到行为进行映射的学习，其目的是使强化信号（回报函数值）最大。

- 强化学习中由环境提供的强化信号会对产生动作的好坏作一种评价，而不是告诉强化学习系统（Reinforcement Learning System, RLS）如何去产生正确的动作。由于外部环境提供的信息很少，RLS必须靠自身的经历进行学习。据此，RLS在行动-评价的环境中获得知识，并改进行动方案以适应环境。

机器学习的发展史

- 1950s
  - Samuel's checker player
  - Selfridge's Pandemonium
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - Learning in the limit theory
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Winston's arch learner
  - Expert systems and the knowledge acquisition bottleneck
  - Quinlan's ID3
  - Michalski's AQ and soybean diagnosis
  - Scientific discovery with BACON
  - Mathematical discovery with AM

Slide credit: Ray Mooney

机器学习的发展史

- 1980s:
  - Advanced decision tree and rule learning
  - Explanation-based Learning (EBL)
  - Learning and planning and problem solving
  - Utility problem
  - Analogy
  - Cognitive architectures
  - Resurgence of neural networks (connectionism, backpropagation)
  - Valiant's PAC Learning Theory
  - Focus on experimental methodology
- 1990s
  - Data mining
  - Adaptive software agents and web applications
  - Text learning
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
  - Ensembles: Bagging, Boosting, and Stacking
  - Bayes Net learning

Slide credit: Ray Mooney

机器学习的发展史

- 2000s
  - Support vector machines & kernel methods
  - Graphical models
  - Statistical relational learning
  - Transfer learning
  - Sequence labeling
  - Collective classification and structured outputs
  - Computer Systems Applications (Compilers, Debugging, Graphics, Security)
  - E-mail management
  - Personalized assistants that learn
  - Learning in robotics and vision
- 2010s
  - Deep learning systems
  - Learning for big data
  - Bayesian methods
  - Multi-task & lifelong learning
  - Applications to vision, speech, social networks, learning to read, etc.

Slide credit: Ray Mooney