中山大學
SUN YAT-SEN UNIVERSITY

机器学习
Machine Learning

沈颖 副教授

sheny76@mail.sysu.edu.cn

林振洲 助教

linzhzh6@mail2.sysu.edu.cn

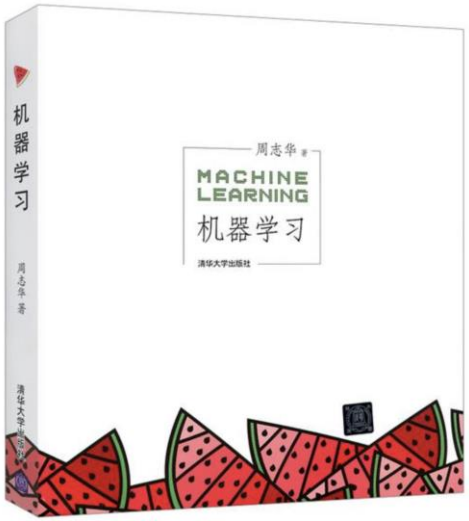
20级智能科学与技术专业

2021学年秋季学期

课程安排

机器学习

理论（36学时）



机器学习/周志华著.--北京：清华大学出版社，2016

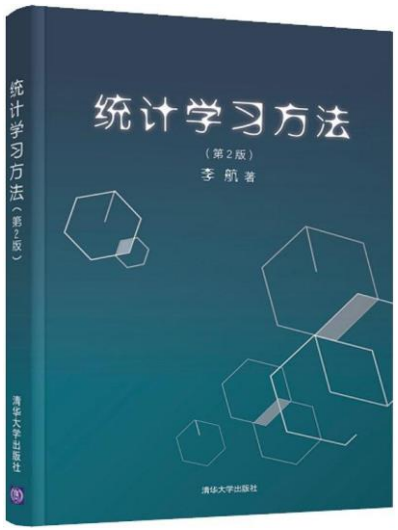
勘误修订：
https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/MLbook2016.htm#

1. 绪论（1.1至1.3）
2. 模型评估与选择（除2.4以外）
3. 线性模型（除3.4以外）
4. 神经网络（除5.5以外）
5. 支持向量机（除6.5、6.6以外）
6. 贝叶斯分类器（7.1至7.4）
7. 决策树（4.1至4.2）
8. 集成学习（除8.5以外）
9. 聚类（除9.2、9.4.2、9.5和9.6以外）
10. 降维（10.1和10.3）
11. 强化学习（16.1至16.4）

课程安排

统计学习方法

理论（36学时）



统计学习方法/李航著.--北京：清华大学出版社，2019

勘误修订：
http://blog.sina.cn/dpool/blog/s/blog_7ad48fee0102yjdu.html?type=-1

1. 绪论（第1章全部）
2. 模型评估与选择（第6章6.1全部）
3. 线性模型（第1章全部）
4. 神经网络（第2章除2.3.2和2.3.3以外）
5. 支持向量机（第7章全部）
6. 贝叶斯分类器（第4章全部）
7. 决策树（第4章4.1和4.2全部）
8. 集成学习（第8章除8.2和8.4.3以外）
9. 聚类（第14章全部）
10. 降维（第3、16章全部）
11. 强化学习（无）

课程安排

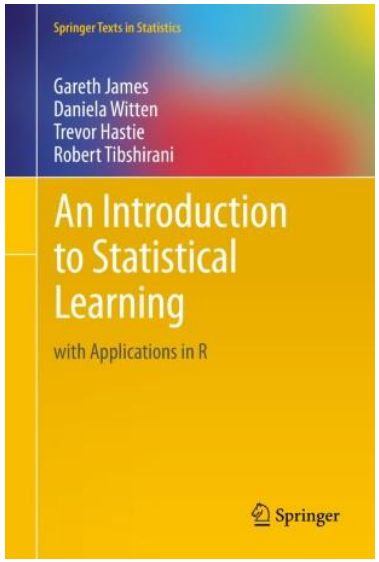
1. 绪论
2. 模型评估与选择
3. 线性模型
4. 神经网络
5. 支持向量机
6. 贝叶斯分类器
7. 决策树
8. 集成学习
9. 聚类
10. 降维
11. 强化学习

平时25%
1. 线性模型
2. 支持向量机
3. 集成学习
4. 考勤和纪律

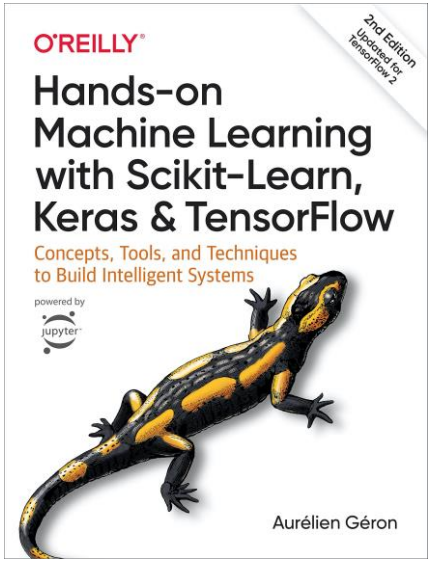
期中15% → 神经网络

期末60%
1. 问答题
2. 推导题
3. 计算题

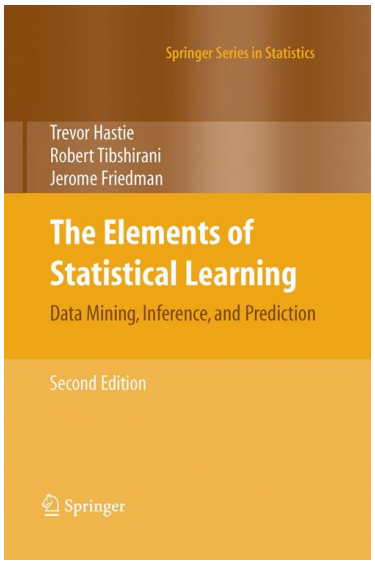
参考书籍



Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*, Springer, 2013.
http://faculty.marshall.usc.edu/gareth-james/ISL/



Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2019.
https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/



Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd Edition), Springer, 2009.
https://web.stanford.edu/~hastie/ElemStatLearn//

网络资源

网络教学

- Prof. Andrew Ng (Stanford University)，吴恩达教授
- Prof. Hung-Yi Lee (National Taiwan University)，李宏毅教授

学术文章

- Journal of Machine Learning Research, JMLR
  https://www.jmlr.org/
- International Conference on Machine Learning, ICML
  https://icml.cc/
- Annual Conference on Learning Theory, COLT
  http://learningtheory.org/colt2020/cfp.html

1. 绪论
2. 模型评估与选择
3. 线性模型
4. 神经网络
5. 支持向量机
6. 贝叶斯分类器
7. 决策树
8. 集成学习
9. 聚类
10. 降维
11. 强化学习

第1章 绪论

1. 机器学习的概念内涵

（定义、特点）

2. 机器学习的一般分类

（三类的定义、本质、术语、形式化描述）

3. 机器学习的关键要素

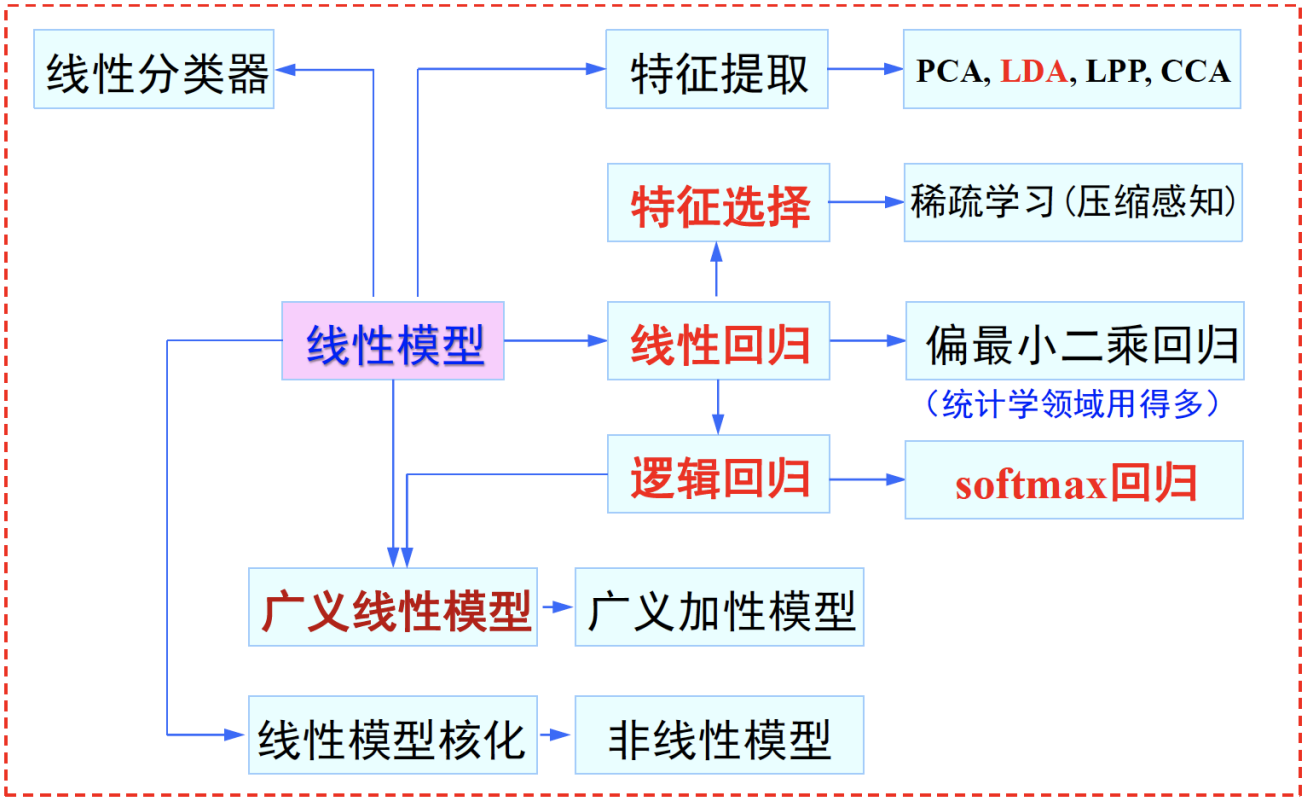
（模型的形式，策略的概念、损失函数、期望损失、经验损失、两种策略，算法的概念）

绪论——各章节安排

1. 绪论
2. 模型评估与选择
3. 线性模型
4. 神经网络
5. 支持向量机
6. 贝叶斯分类器
7. 决策树
8. 集成学习
9. 聚类
10. 降维
11. 强化学习

线性模型

- 将数据线性地变换到另一个维度

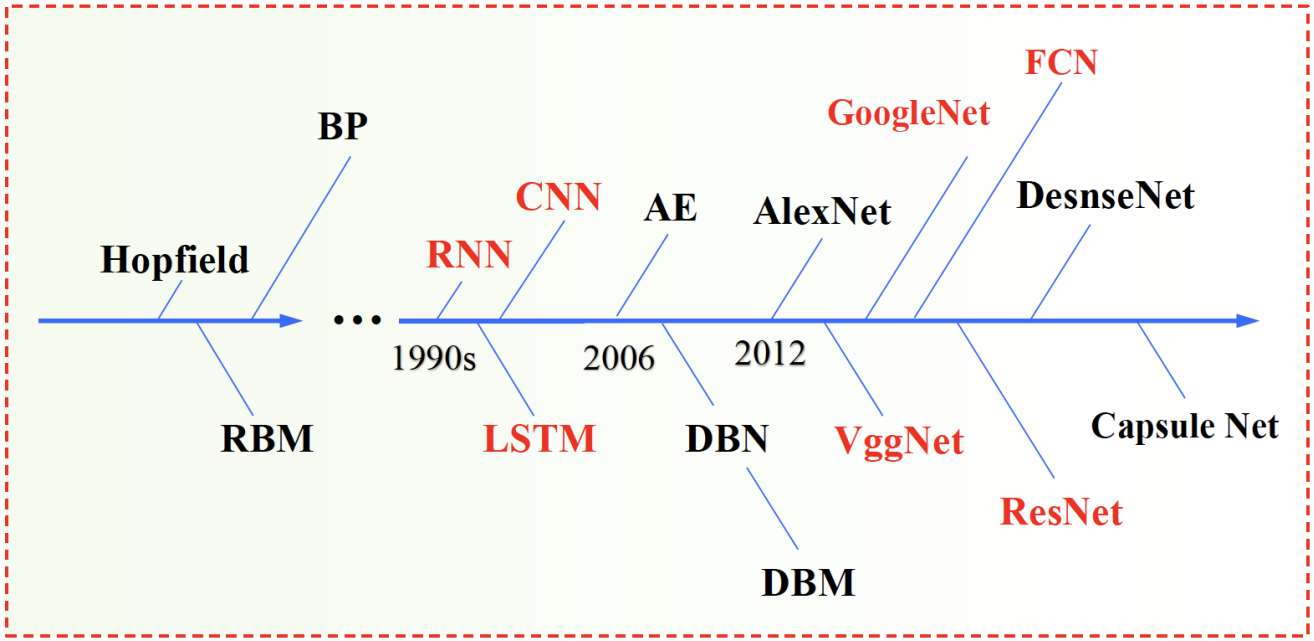


绪论——各章节安排

1. 绪论
2. 模型评估与选择
3. 线性模型
4. 神经网络
5. 支持向量机
6. 贝叶斯分类器
7. 决策树
8. 集成学习
9. 聚类
10. 降维
11. 强化学习

神经网络与深度学习

- 通用网络的发表之路

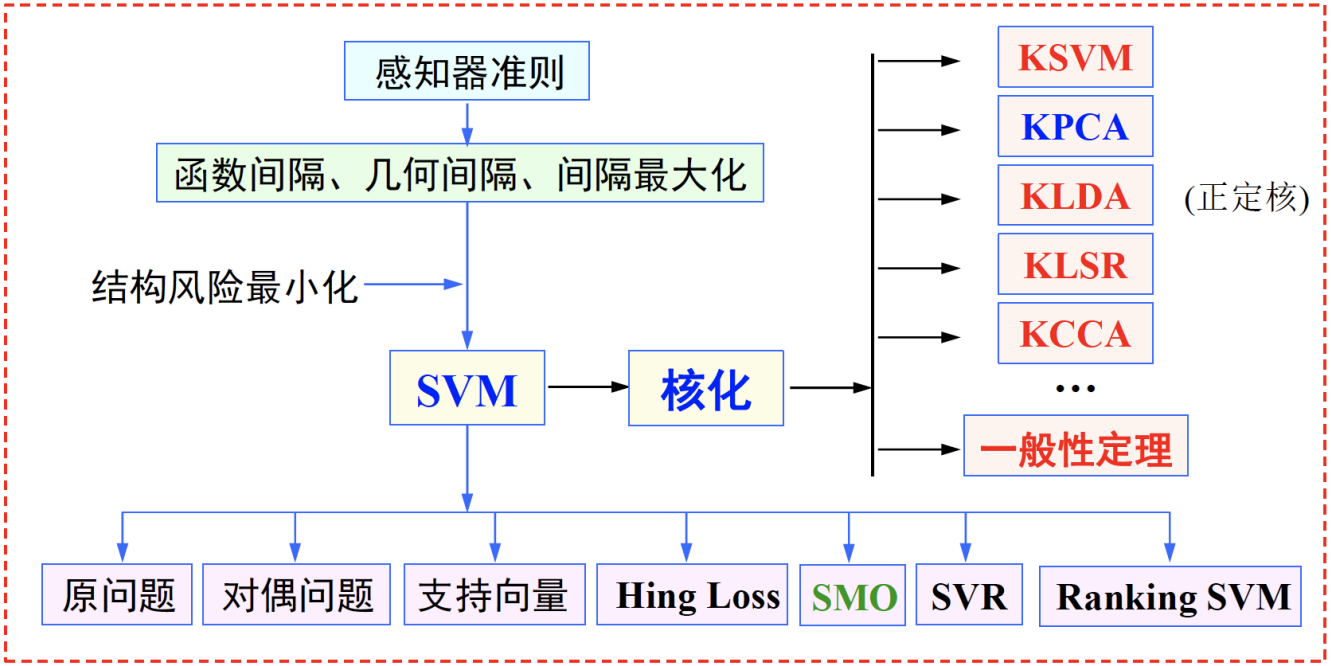


绪论——各章节安排

1. 绪论
2. 模型评估与选择
3. 线性模型
4. 神经网络
5. 支持向量机
6. 贝叶斯分类器
7. 决策树
8. 集成学习
9. 聚类
10. 降维
11. 强化学习

支持向量机与核方法

- 机器学习的经典内容

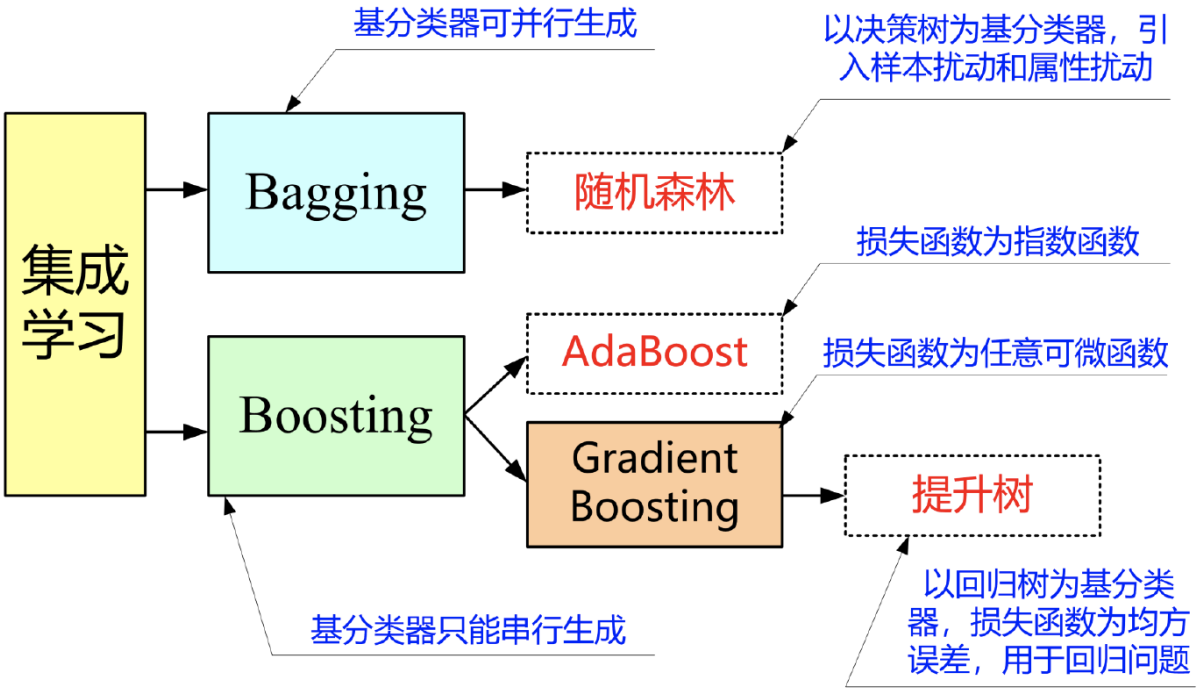


绪论——各章节安排

1. 绪论
2. 模型评估与选择
3. 线性模型
4. 神经网络
5. 支持向量机
6. 贝叶斯分类器
7. 决策树
8. 集成学习
9. 聚类
10. 降维
11. 强化学习

集成学习

- 通过学习器集成，将弱学习器提升成强学习器的一类方法

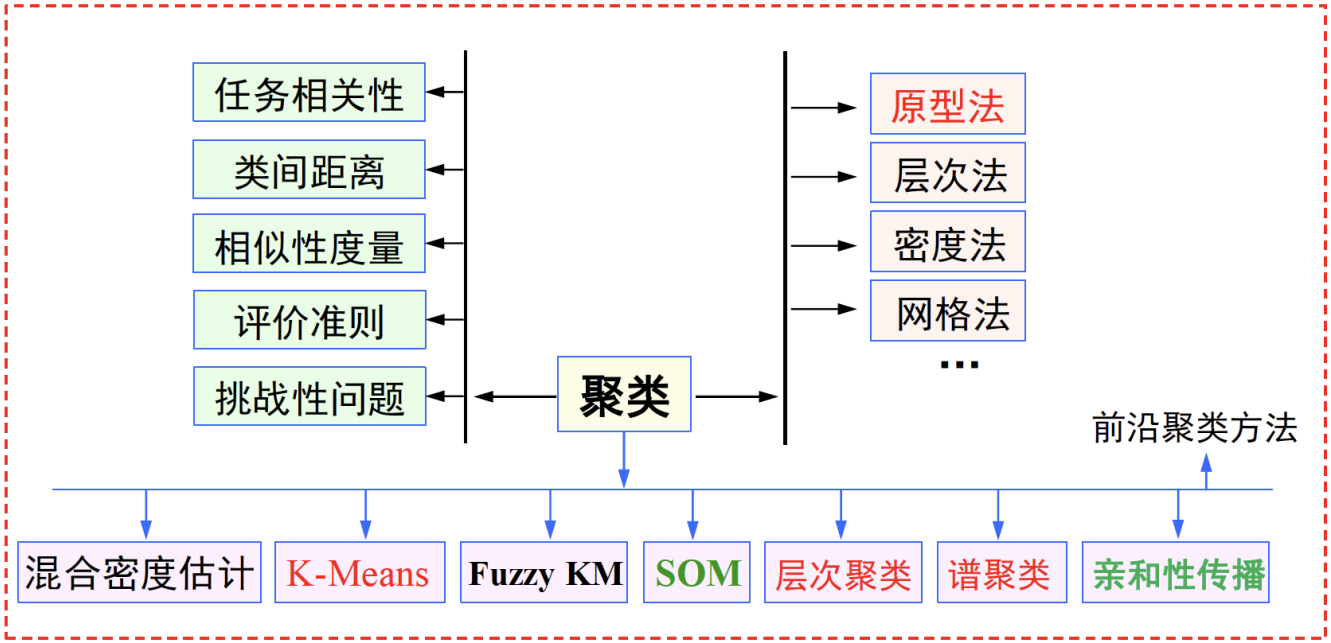


绪论——各章节安排

1. 绪论
2. 模型评估与选择
3. 线性模型
4. 神经网络
5. 支持向量机
6. 贝叶斯分类器
7. 决策树
8. 集成学习
9. 聚类
10. 降维
11. 强化学习

数据聚类

- 将数据划分为相似的子集

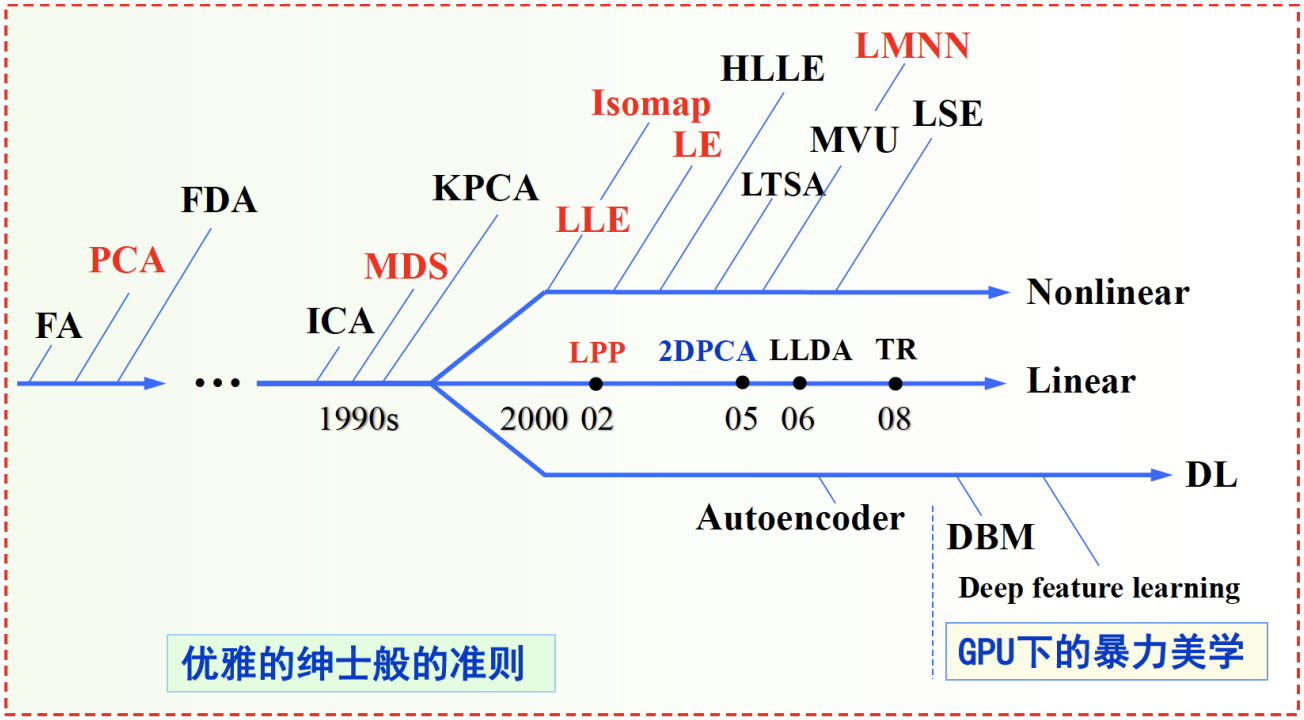


绪论——各章节安排

1. 绪论
2. 模型评估与选择
3. 线性模型
4. 神经网络
5. 支持向量机
6. 贝叶斯分类器
7. 决策树
8. 集成学习
9. 聚类
10. 降维
11. 强化学习

降维与距离度量学习

- 降维

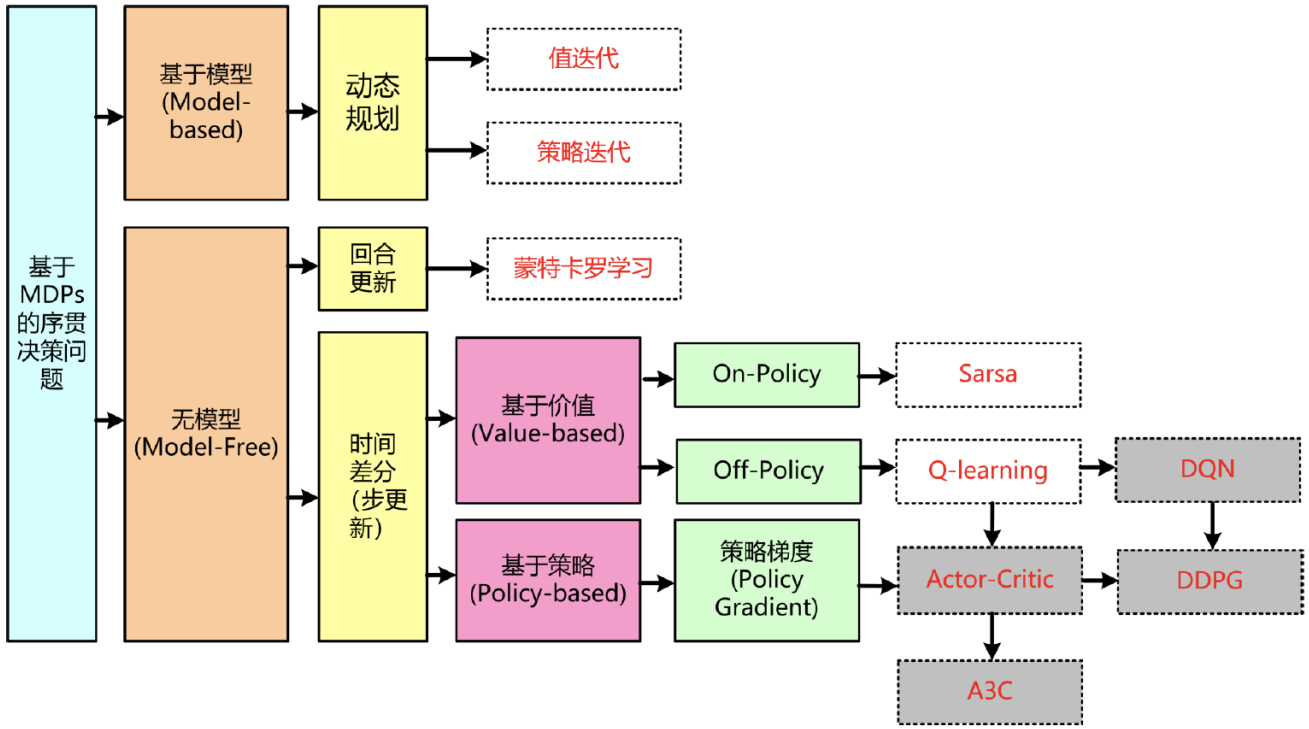


绪论——各章节安排

1. 绪论
2. 模型评估与选择
3. 线性模型
4. 神经网络
5. 支持向量机
6. 贝叶斯分类器
7. 决策树
8. 集成学习
9. 聚类
10. 降维
11. 强化学习

强化学习

- Agent基于环境反馈来学习最优的行动策略



绪论——本课程目标

- 通过本课程的学习，希望大家能够了解机器学习基本原理，掌握机器学习的基本思想和关键算法，了解机器学习最新研究成果和前沿研究动态，为解决相关实际问题提供知识储备。
  - ✓ 看得懂机器学习模型
  - ✓ 能解释机器学习模型
  - ✓ 能求解机器学习模型
  - ✓ 能自己构造学习模型
  - ✓ 可以提炼、描述实际应用中的学习问题和工作步骤
  - ✓ 能够求解自己的学习模型并加以改进
  - ✓ 能够给自己带来价值提升

绪论——什么是机器学习

“数据”

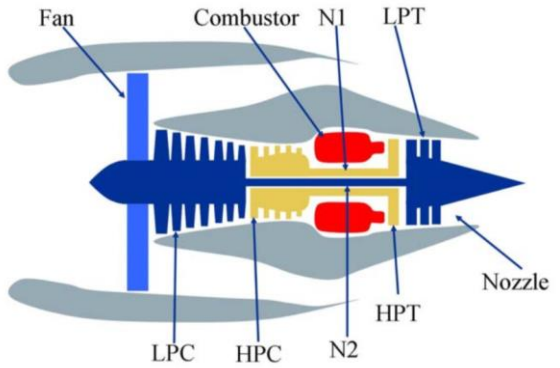


Figure 1. Simplified diagram of engine simulated in C-MAPSS [11].

C-MAPSS 是 NASA 用于模拟真实大型商用涡扇发动机的工具。
数据用于 IEEE PHM 2008 数据挑战赛。*

Symbol	*Description*	*Units*
Parameters available to participants as sensor data		
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/P2)	--
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	--
farB	Burner fuel-air ratio	--
htBleed	Bleed Enthalpy	--
Nf_dmd	Demanded fan speed	rpm
PCNfR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

21 channels of parameters are monitored and analyzed.

- Sensory data of 519 engines are used, which were obtained under 6 operational conditions with different combinations of altitude (0-42K feet), Mach number (0-0.84), and throttle resolver angle (20-100 degree).
- Run-to- failure data of 260 engines are used as training data, and 259 engines without run-to-failure are used as testing data.

*Saxena, et al. "Damage propagation modeling for aircraft engine run-to-failure simulation." 2008 international conference on prognostics and health management. IEEE, 2008.

绪论——什么是机器学习

“模型”

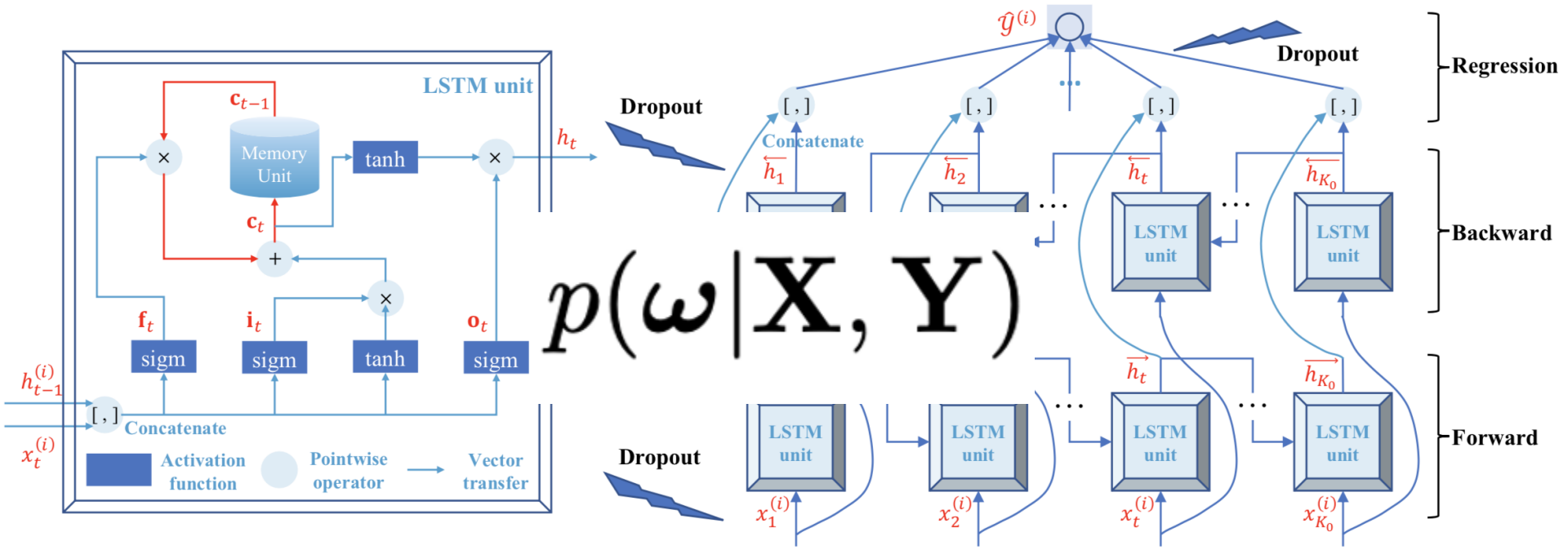


Fig. 3: A simple illustration of bi-directional LSTM

基于贝叶斯双向 LSTM 的方法

该方法用于处理具有复杂时间序列结构的状态监测数据，
可用于工业系统预测。

绪论——什么是机器学习

“策略”

- 学习策略是指学习过程中系统所采用的推理策略。一个学习系统总是由学习和环境两部分组成。由环境提供信息，学习部分则实现信息转换，用能理解的形式记忆下来，并从中获取有用的信息。
  - **机械学习 (Rote learning)**：无需任何推理或其它知识转换，直接吸取环境所提供的信息。
  - **示教学习 (Learning from instruction 或 Learning by being told)**：与人类社会的学校教学方式相似，学习的任务是建立一个系统，使它能接受教导和建议，并有效地存贮和应用学到知识。
  - **演绎学习 (Learning by deduction)**：所用推理形式为演绎推理。推理从公理出发，经过逻辑变换推导出结论。

绪论——什么是机器学习

“策略”

- 按学习策略
  - **类比学习 (Learning by analogy)**：利用不同领域（源域、目标域）中的知识相似性，通过类比从源域知识推导出目标域相应知识，从而实现学习。
  - **基于解释的学习 (Explanation-based learning, EBL)**：根据环境中的目标概念、相关概念例子、领域理论及可操作准则，构造一个解释，用于说明为什么该例子能满足目标概念，然后将解释进行推广（比如：一个满足可操作准则的充分条件）。（知识库构建）
  - **归纳学习 (Learning from induction)**：归纳学习基于环境提供的关于某个概念的一些实例或反例，通过归纳推理得出该概念的一般描述。

绪论——什么是机器学习

“策略”

- 基于所获取知识的表示形式
  - **产生式规则**：学习系统中的学习行为主要是：生成、泛化、特化或合成产生式规则。
  - **框架和模式**（schema）：每个框架包含一组槽，用于描述事物（概念和个体）的各个方面。（知识工程中的本体构造）
  - **图**：用以描述时空关联数据的结构。
  - **神经网络**：在联接学习中，学习所获取的知识，最后归纳为一个神经网络。

绪论——什么是机器学习

“算法”

$$\mathrm{KL}(q_\phi(\boldsymbol{\omega})||p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})) = \int q_\phi(\boldsymbol{\omega}) \log \frac{q_\phi(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})} \mathrm{d}\boldsymbol{\omega}.$$

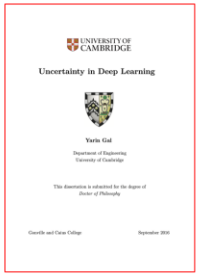
$$\mathrm{KL}(q_\phi(\boldsymbol{\omega})||p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})) = \int q_\phi(\boldsymbol{\omega}) \log \frac{q_\phi(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})} \mathrm{d}\boldsymbol{\omega}.$$

a stochastic estimator with S being a random subset of $\{1, \cdots, N\}$

$$\mathrm{KL}(q_\phi(\boldsymbol{\omega})||p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})) \propto \mathbb{E}_{S,\epsilon}[\widehat{\mathcal{D}}(\phi, S, \epsilon)]$$

A stochastic optimizer can be used to obtain an optimum ϕ^* for minimizing $\widehat{\mathcal{D}}(\phi, S, \epsilon)$, and this ϕ^* is also an optimum to $\mathrm{KL}\left(q_\phi(\boldsymbol{\omega})||p(\boldsymbol{\omega}|\mathbf{X},\mathbf{Y})\right)$

Key point: $q_\phi(\boldsymbol{\omega})$ is … differentiable transfo… parameter-free distr…



Implementing VI with $q_\phi(\boldsymbol{\omega})$ under $\boldsymbol{\omega} = \mathrm{diag}(\boldsymbol{\epsilon})\phi$ and $\boldsymbol{\epsilon}$ being a product of Bernoulli distributions is equivalent to implement dropout in deep network with $\mathrm{EL}(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{N} \left\| y^{(i)} - \mathbf{f}^{\boldsymbol{\omega}}(\mathbf{x}^{(i)}) \right\|^2 / 2N$.

K-L散度（相对熵）=交叉熵-熵

$$D_{KL}(P||Q) = -\sum_i P(i) \ln \frac{Q(i)}{P(i)} = \sum_i P(i) \ln \frac{P(i)}{Q(i)}$$

“应用”

Bayesian Bi-directional LSTM

```
BayesBLSTM
MainInput  (20×14)
Bi-LSTM    (18, True, 0.20, 0.20,
            tanh, hard_sigmoid)
Bi-LSTM    (18, True, 0.20, 0.20,
            tanh, hard_sigmoid)
AuxInput   (20×4)   ── Concatenate ()
Bi-LSTM    (31, False, 0.20, 0.20,
            tanh, hard_sigmoid)
Dropout    (0.20)
Dense      (200, linear)
Dropout    (0.20)
Dense      (100, linear)
Dropout    (0.20)
Dense      (1, linear)
```

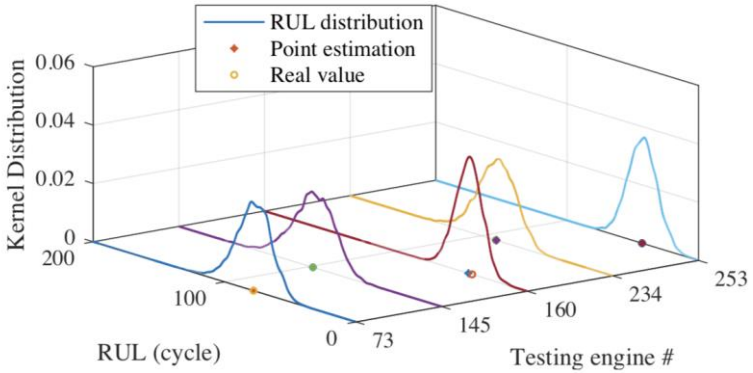


Fig. 8: Prognostics with uncertainty quantification for engines

知识计算：
基于RUL 预测的
引擎诊断

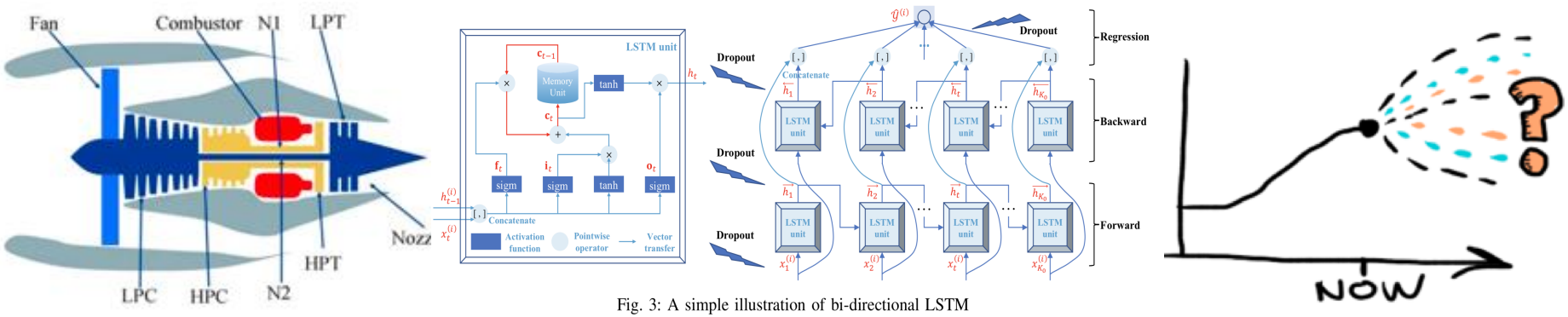


Fig. 3: A simple illustration of bi-directional LSTM

知识推理：决策支持

机器学习应用

- 计算机视觉 （图像识别、身份识别、行为监控/交通监控）
- 数据挖掘 （推荐系统、互联网广告投放、新型商务与政务）
- 自然语言处理 （自动新闻摘要、知识图谱、网页分类、机器翻译）
- 语音处理 （语音识别、同声翻译、跨语言翻译与语音合成）
- 智能人机交互 （表情、手势、声音、符号、人机对抗、游戏）
- 人类健康 （医学图像、体测数据）
- 空间探测与环境资源监测（卫星/航空遥感图像、情报生成）
- 网络安全与舆情分析 （互联网、大数据、证券市场分析）
- 生物信息学 （ DNA序列测序）
- 工业应用 （零部件/物品分类、损伤检测）
- 文档数字化 （历史书籍报纸、档案、手稿、标牌等）
- 交通分析 （自动驾驶、出行服务、智能管理、车牌识别）
- 网络搜索、信息提取和过滤

（文本、图像、视频、音频、多媒体 、邮件过滤）

应用实例

- ✓ 商品图片分类
- ✓ 图像句子描述
- ✓ 目标检测与识别
- ✓ 人脸识别/开放条件下行人再识别
- ✓ 广告点击行为预测
- ✓ 微生物种类判别
- ✓ 基于运营商数据的个人征信评估
- ✓ 基于文本内容的垃圾短信/邮件识别
- ✓ 中文句子内容精准分析
- ✓ P2P网络借贷平台的经营风险量化分析
- ✓ 客户用电异常行为分析
- ✓ 自动驾驶场景中的交通标志检测
- ✓ 市民出行公交预测
- ✓ 大数据精准营销中用户画像
- ✓ 微额借款用户人品预测
- ✓ 验证码识别
- ✓ 客户流失率预测
- ✓ 汽车4S店邮件营销方案
- ✓ … …

- ✓ 机场客流量分布预测
- ✓ 音乐流行趋势预测
- ✓ 需求预测与仓储规划方案
- ✓ 新浪微博互动量预测
- ✓ 货币基金资金流入流出预测
- ✓ 电影票房预测
- ✓ 产品价格预测分析
- ✓ 微博传播规模和传播深度预测
- ✓ 网约车出行流量预测
- ✓ 商品质量（红酒）评分
- ✓ 搜索引擎的搜索量和股价波动
- ✓ 股价走势预测
- ✓ 地震预报、气象分析
- ✓ 基于用户位置信息的商业选址
- ✓ 互联网情绪指标分析
- ✓ 基于用户轨迹的商户精准营销
- ✓ 推荐系统（穿衣搭配、购买）
- ✓ 交通事故成因分析
- ✓ … …

绪论——机器学习的一般分类

机器学习是一个范围宽阔、内容繁多、应用广泛的领域，目前不存在一个统一的理论体系涵盖所有内容。

一般分类

- 监督学习（supervised learning）

  线性模型、神经网络、支持向量机、决策树、集成学习、 *k*近邻

- 无监督学习（unsupervised learning）

  聚类（*k*均值、高斯混合）、降维（ 主成分分析）

- 强化学习（reinforcement learning）

  K-摇臂赌博机、有模型学习、无模型学习

- 半监督学习（semi-supervised learning）
- 迁移学习（transfer learning）
- 主动学习（active learning）

绪论——机器学习的一般分类（监督学习）

监督学习：从标注数据中学习预测模型的机器学习问题。

标注数据表示输入输出的对应关系，预测模型对给定的输入产生相应的输出。

输出

输入

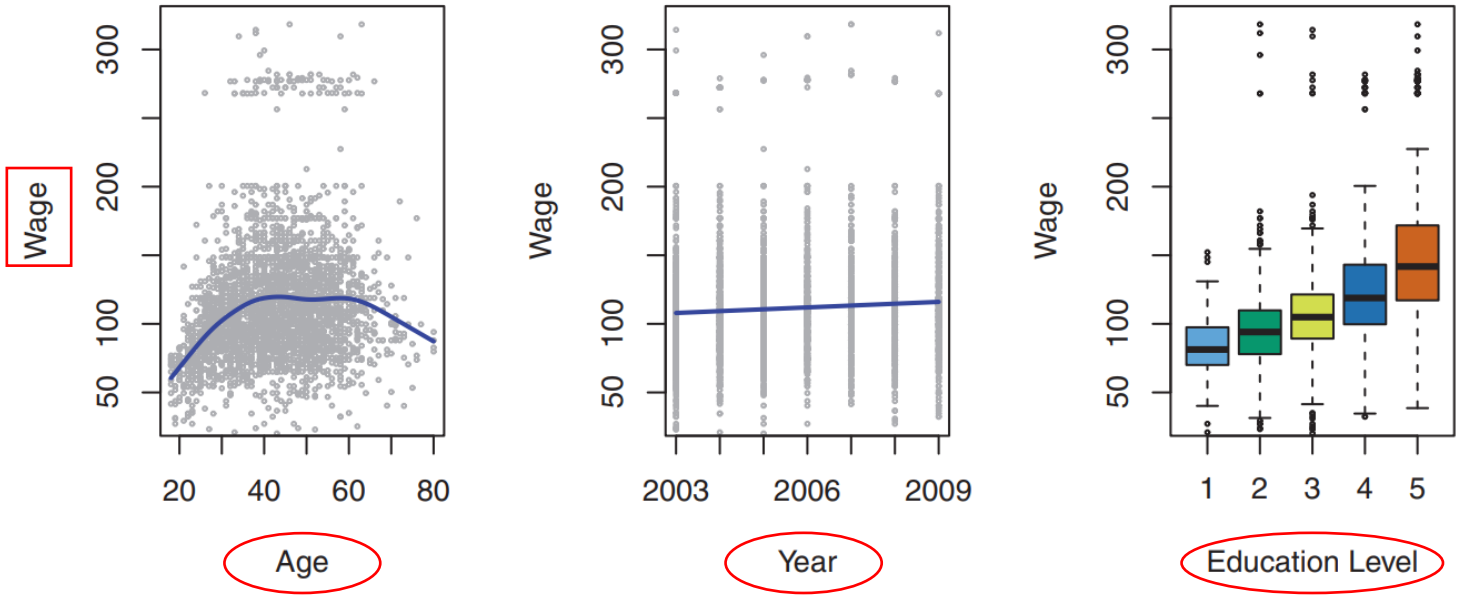
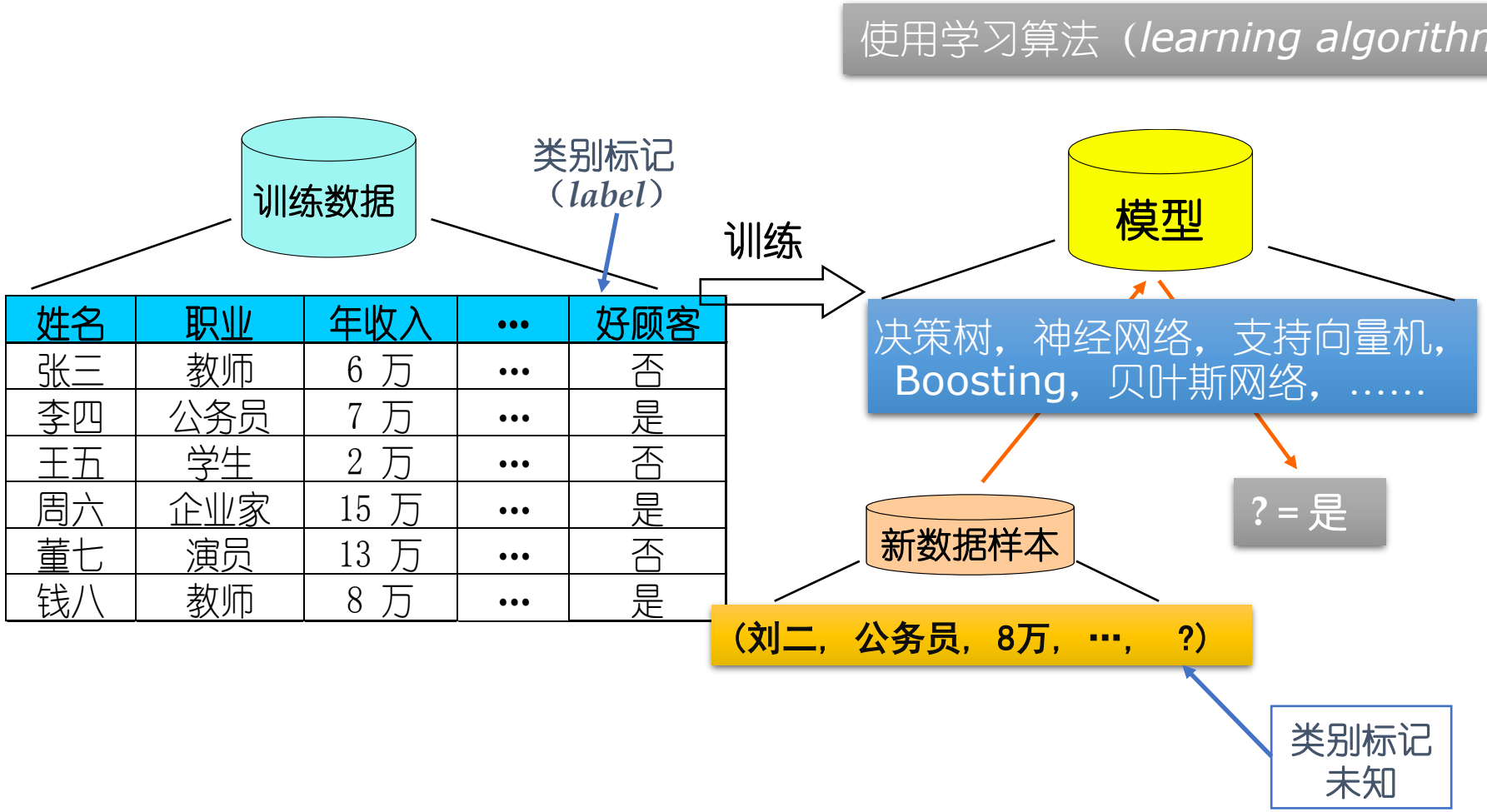


FIGURE 1.1. Wage *data, which contains income survey information for males from the central Atlantic region of the United States.* Left: wage *as a function of* age. *On average,* wage *increases with* age *until about* 60 *years of age, at which point it begins to decline.* Center: wage *as a function of* year. *There is a slow but steady increase of approximately* $10,000 *in the average* wage *between* 2003 *and* 2009. Right: *Boxplots displaying* wage *as a function of* education, *with* 1 *indicating the lowest level (no high school diploma) and* 5 *the highest level (an advanced graduate degree). On average,* wage *increases with the level of education.*

绪论——机器学习的一般分类（监督学习）

监督学习：从标注数据中学习预测模型的机器学习问题。

标注数据表示输入输出的对应关系，预测模型对给定的输入产生相应的输出。



绪论——机器学习的一般分类（无监督学习）

无监督学习：从无标注数据中学习分析模型的机器学习问题。

无标注数据是“自然”得到的数据，分析模型表示数据的类别、转换等。

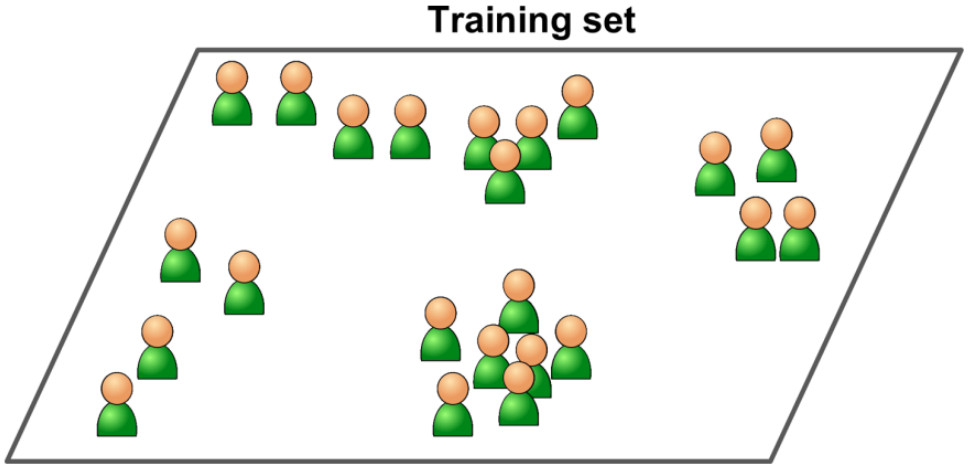


Figure 1-7. An unlabeled training set for unsupervised learning

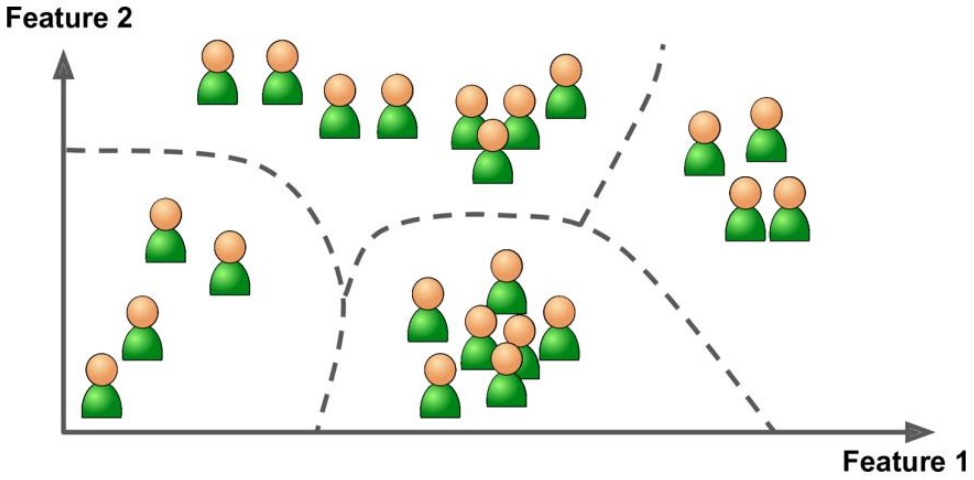


Figure 1-8. Clustering

绪论——机器学习的一般分类（无监督学习）

无监督学习：从无标注数据中学习分析模型的机器学习问题。

无标注数据是“自然”得到的数据，分析模型表示数据的类别、转换等。

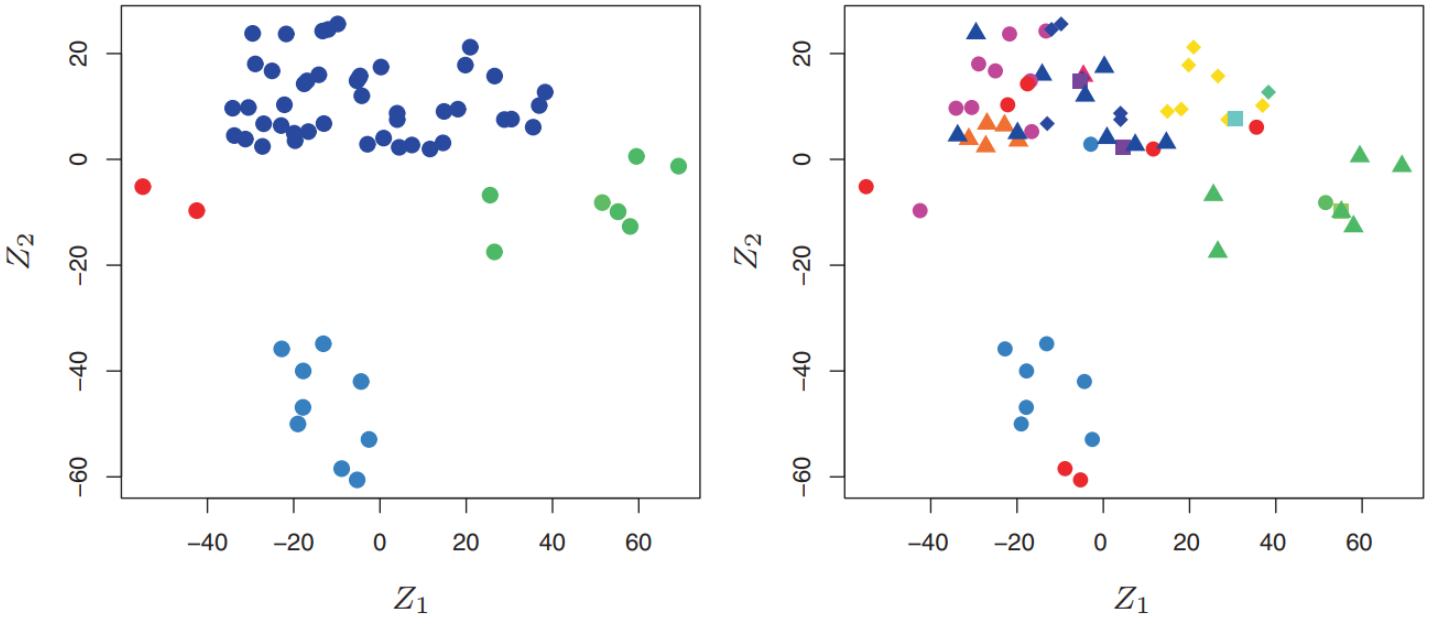


FIGURE 1.4. Left: *Representation of the* NCI60 *gene expression data set in a two-dimensional space,* Z_1 *and* Z_2*. Each point corresponds to one of the* 64 *cell lines. There appear to be four groups of cell lines, which we have represented using different colors.* Right: *Same as left panel except that we have represented each of the* 14 *different types of cancer using a different colored symbol. Cell lines corresponding to the same cancer type tend to be nearby in the two-dimensional space.*

聚类：单遍聚类、层次聚类、质心聚类、密度聚类

绪论——机器学习的一般分类（强化学习）

强化学习：智能系统在与环境的连续互动中学习最优行为策略的机器学习问题。智能系统观测到的是与环境互动得到的数据序列，主要目的是学习得到与环境互动的最优策略。

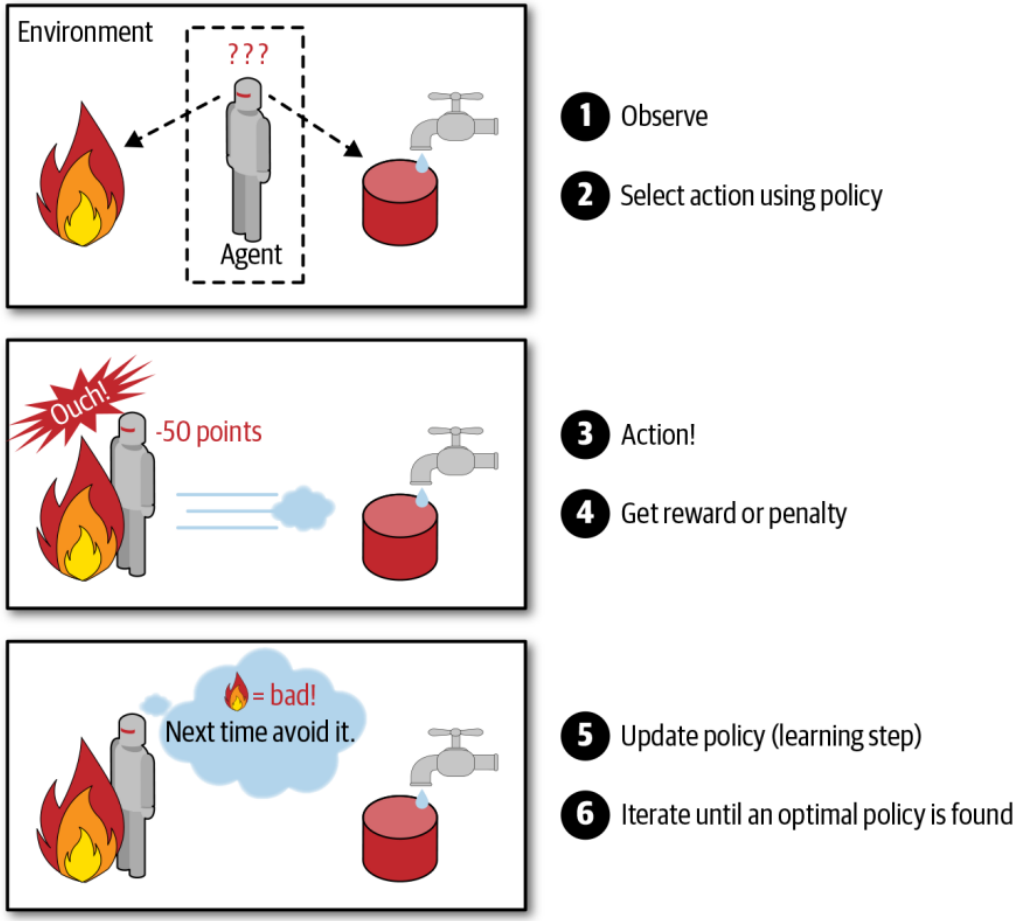


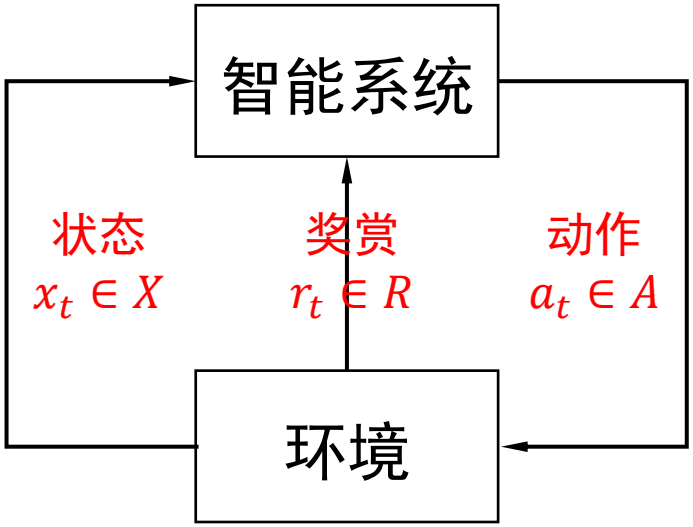
Figure 1-12. Reinforcement Learning

绪论——机器学习的一般分类（强化学习）

强化学习：智能系统在与环境的连续互动中学习最优行为策略的机器学习问题。智能系统观测到的是与环境互动得到的数据序列，主要目的是学习得到与环境互动的最优策略。

马尔科夫决策过程（Markov decision process）是状态、动作、奖励序列上的随机过程，由四元组$\langle X, A, P, R\rangle$组成。$X$是机器所处环境中的状态空间，$A$是机器所能采取的动作构成的动作空间，$P$是状态转移概率函数，$R$是奖赏函数。

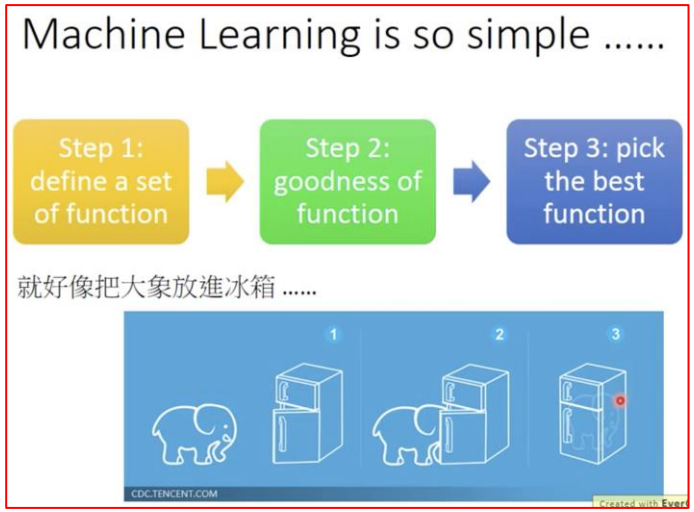
本质：学习最优的序贯策略

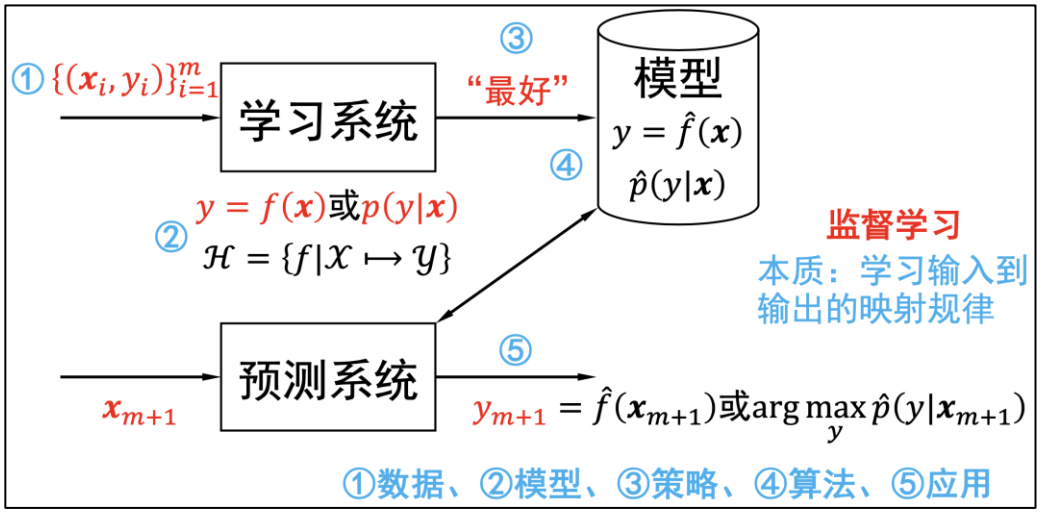


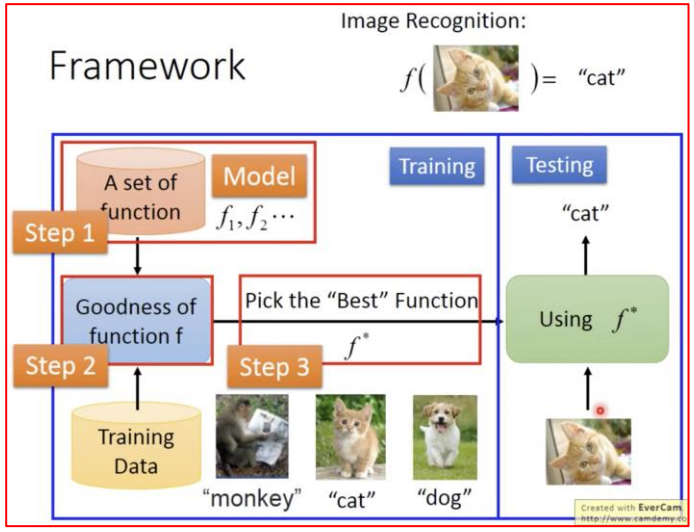
系统以长期奖励的最大化为目的，通过不断地“试错”（trail and error），从所有可能的策略中学得最优策略。

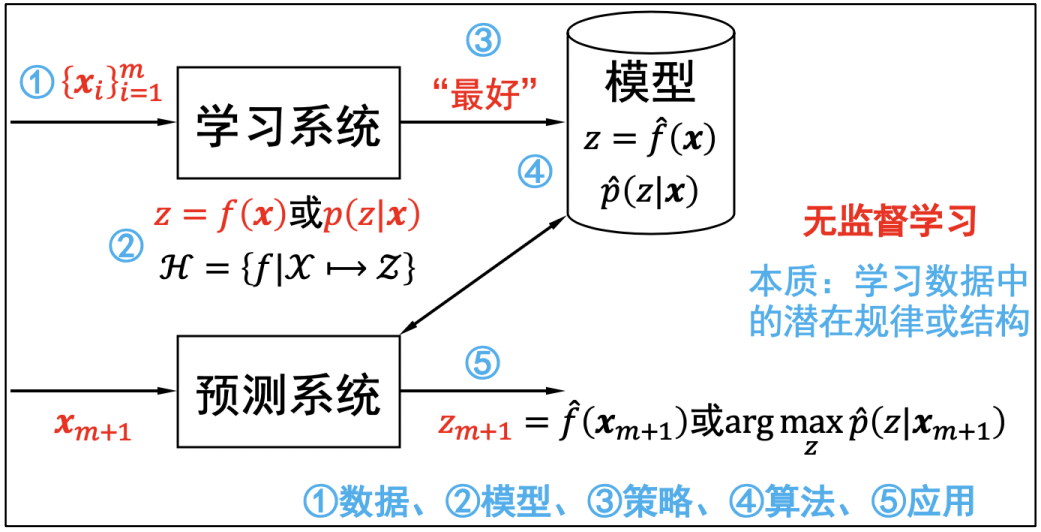
绪论——机器学习的关键要素

“数据”、“模型”、“策略”、“算法”









绪论——机器学习的关键要素（模型评估与选择）

表示	准则	优化
—近邻法	—准确率/错误率	—组合优化
—支持向量机	—召回率	—贪心搜索
—超平面分类器	—平方误差	—连续优化
—朴素贝叶斯分类	—后验概率	—无/有约束优化
—逻辑斯蒂回归	—似然	—梯度下降
—神经网络	—信息增益	—共轭梯度
—决策树	—KL距离	—线性规划
—图模型	—利润	—二次规划
—贝叶斯网络	—成本/效用	—半正定规划
—命名规则	—距离	

中山大學
SUN YAT-SEN UNIVERSITY

第2章 模型评估与选择

1. 训练误差与测试误差
2. 过拟合与模型选择
3. 性能度量
4. 偏差与方差

沈颖 副教授
sheny76@mail.sysu.edu.cn

训练误差与测试误差

机器学习的目的是使学得的模型不仅对已知数据而且对未知数据都能有很好的预测能力。

- 损失函数（loss function）度量模型一次预测的好坏
- 期望损失（expected loss ）度量平均意义下模型预测的好坏
- 经验损失（empirical risk）度量关于训练数据集的平均损失

当**损失函数**给定时，

- 模型的训练误差（training error）
- 模型的测试误差（testing error）

可用于评价学习方法（**经验损失**）

训练误差与测试误差

训练误差的大小：

- 对判定给定的问题是不是一个容易学习的问题具有重要意义。
- 如果训练误差过大，则说明存在“欠拟合”。
- 如果训练误差过小，则说明存在“过拟合”。

训练误差与测试误差

过拟合和欠拟合

如果训练误差过小，则说明存在“过拟合”

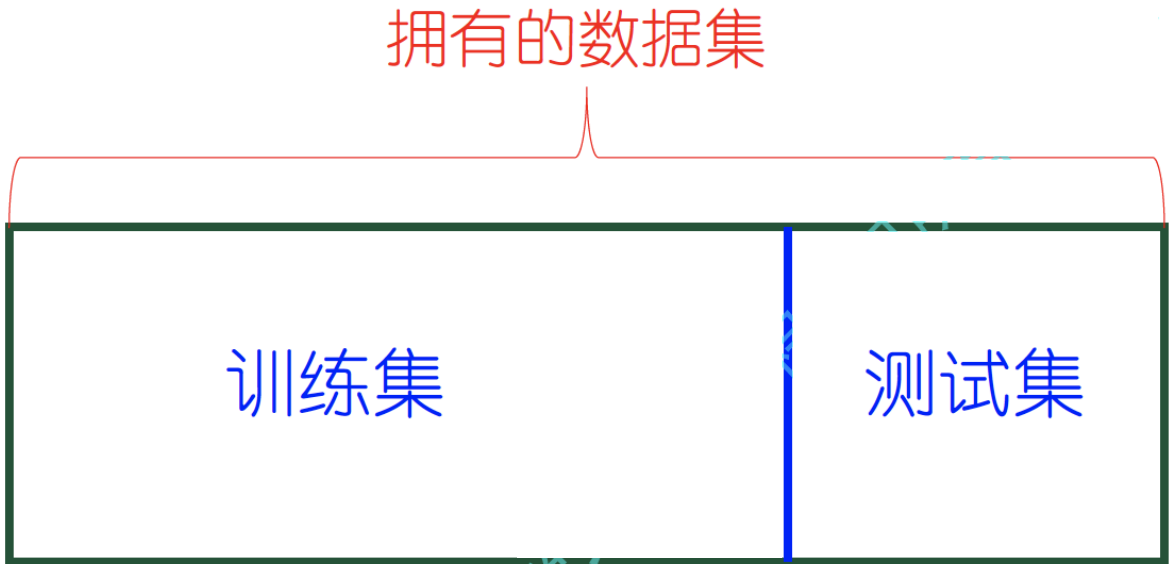
如果训练误差过大，则说明存在“欠拟合”

过拟合与模型选择

在实际应用中数据是不充足的，为了选择好的模型，可以采用

- 留出法（hold-out）
- 交叉验证法（cross validation）
- 自助法（bootstrap）

留出法：



- 直接将数据集划分为两个互斥集合
- 训练/测试集划分要尽可能保持数据分布的一致性
- 训练/测试样本比例通常为2:1~4:1
- 一般若干次随机划分，重复实验取平均值

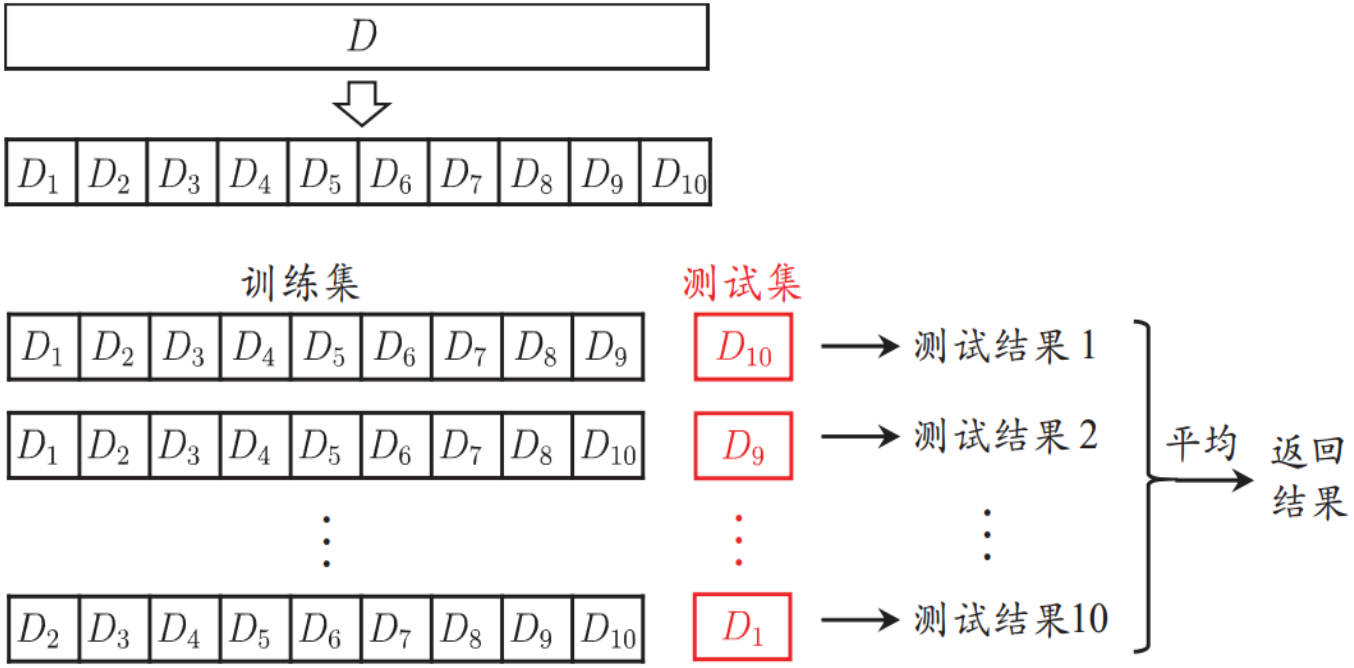
交叉验证法：

将数据集分层采样划分为k个大小相似的互斥子集，

每次用$k-1$个子集的并集作为训练集，

余下的子集作为测试集，

最终返回k个测试结果的均值， k最常用的取值是10。



10 折交叉验证示意图

过拟合与模型选择

交叉验证法：

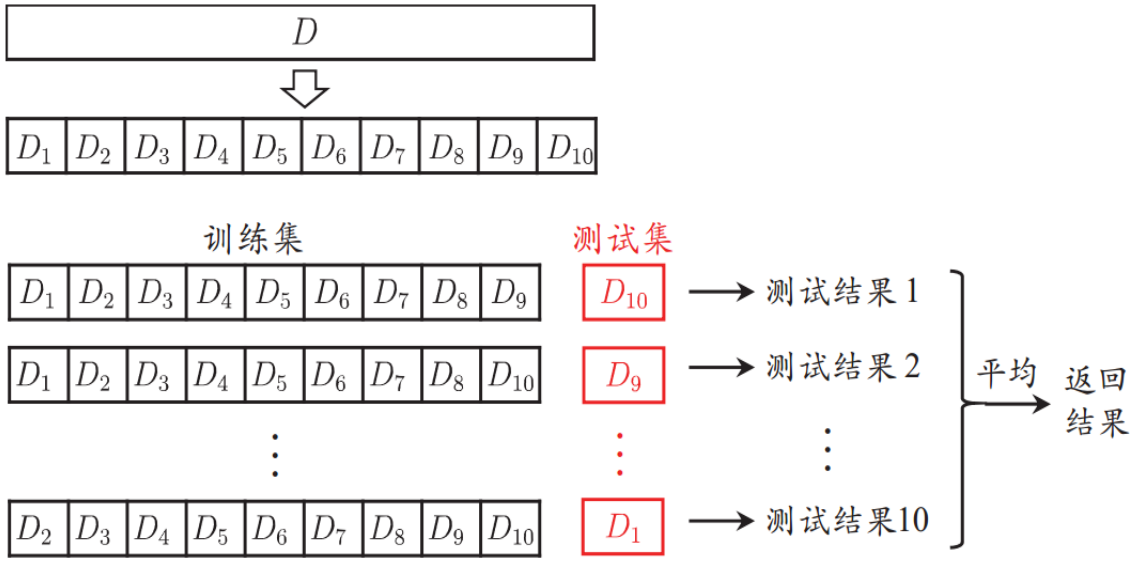
与留出法类似，将数据集D划分为k个子集同样存在多种划分方式，

为了减小因样本划分不同而引入的差别，

k折交叉验证通常随机使用不同的划分重复p次，

最终的评估结果是这p次k折交叉验证结果的均值，例如常见的

“10次10折交叉验证”

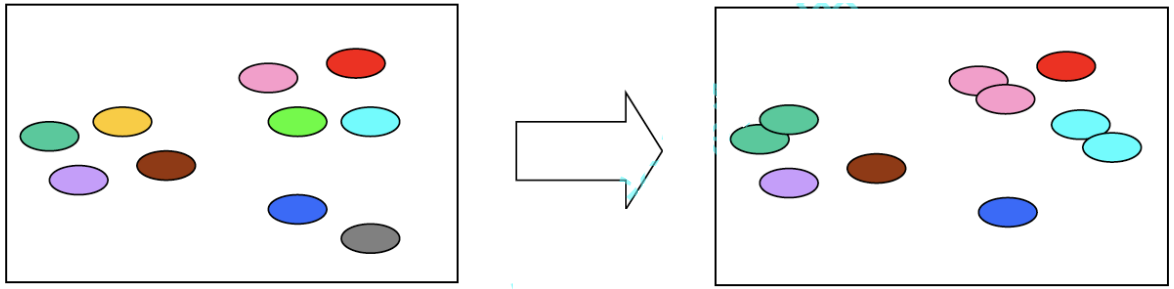


10 折交叉验证示意图

自助法：以自助采样法为基础，对数据集D有放回采样m次得到训练集D'，用D/D'做测试集。

基于“自助采样” (bootsrap sampling)

亦称“有放回采样”、“可重复采样”



- 训练集与原样本集同规模：实际模型与预期模型都使用m个训练样本
- 数据分布有所改变：从初始数据集中产生多个不同的训练集，对集成学习有很大的好处

中山大學
SUN YAT-SEN UNIVERSITY

第2章 模型评估与选择

1. 训练误差与测试误差
2. 过拟合与模型选择
3. 性能度量
4. 偏差与方差

性能度量——回归问题

性能度量是衡量模型泛化能力的评价标准，反映了任务需求；

使用不同的性能度量往往会导致不同的评判结果。

什么样的模型是“好”的，不仅取决于算法和数据，还取决于任务需求

性能度量——分类问题

回归任务最常用的性能度量是“均方误差”

分类任务最常用的性能度量是“错误率”和“精度”：

错误率

$$E(\hat{f};D)=\frac{1}{n}\sum_{j=1}^{n}\mathbb{I}(\hat{f}(\boldsymbol{x}_j)\neq y_j)$$

精度（正确率）

$$acc(\hat{f};D)=\frac{1}{n}\sum_{j=1}^{n}\mathbb{I}(\hat{f}(\boldsymbol{x}_j)=y_j)$$

性能度量——分类问题

信息检索、Web搜索等场景中经常需要衡量正例被预测出来的比率或者预测出来的正例中正确的比率，
此时查准率和查全率比错误率和精度更适合。

分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	TP（真正例）	FN（假反例）
反例	FP（假正例）	TN（真反例）

查准率 $P = \frac{TP}{TP + FP}$

查全率 $R = \frac{TP}{TP + FN}$

性能度量——分类问题

信息检索、Web搜索等场景中经常需要衡量正例被预测出来的比率或者预测出来的正例中正确的比率，
此时查准率和查全率比错误率和精度更适合。

分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	TP（真正例）	FN（假反例）
反例	FP（假正例）	TN（真反例）

查准率 $P = \frac{TP}{TP + FP}$

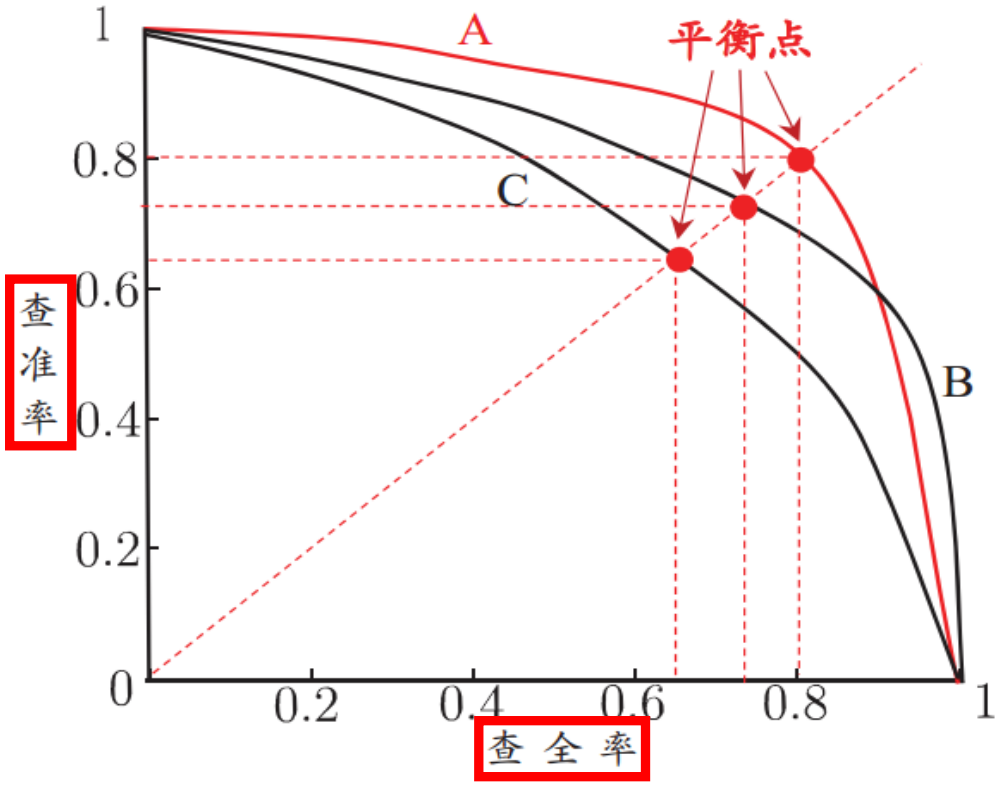
查全率 $R = \frac{TP}{TP + FN}$

性能度量——分类问题

根据学习器的预测结果按正例可能性大小对样例进行排序，

并逐个把样本作为正例进行预测，

则可以得到查准率-查全率曲线，简称“P-R曲线”



查准率 $P = \frac{TP}{TP + FP}$

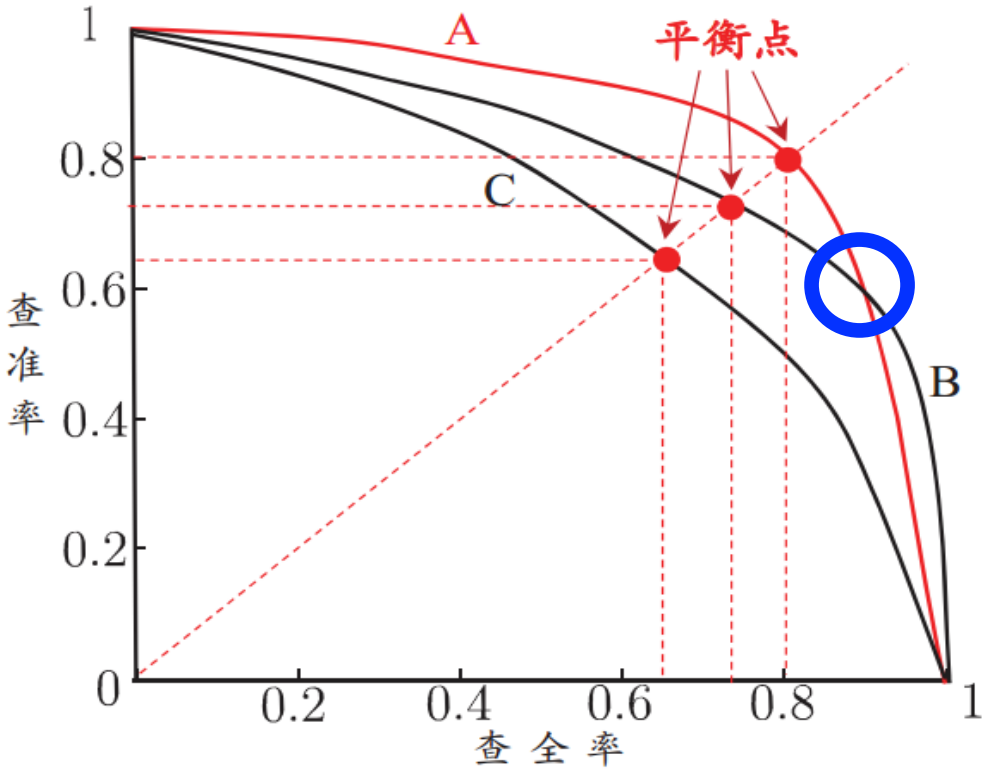
查全率 $R = \frac{TP}{TP + FN}$

性能度量——分类问题

根据学习器的预测结果按正例可能性大小对样例进行排序，

并逐个把样本作为正例进行预测，

则可以得到查准率-查全率曲线，简称“P-R曲线”。



平衡点是曲线上“查准率=查全率”时的取值，

可用来度量P-R曲线

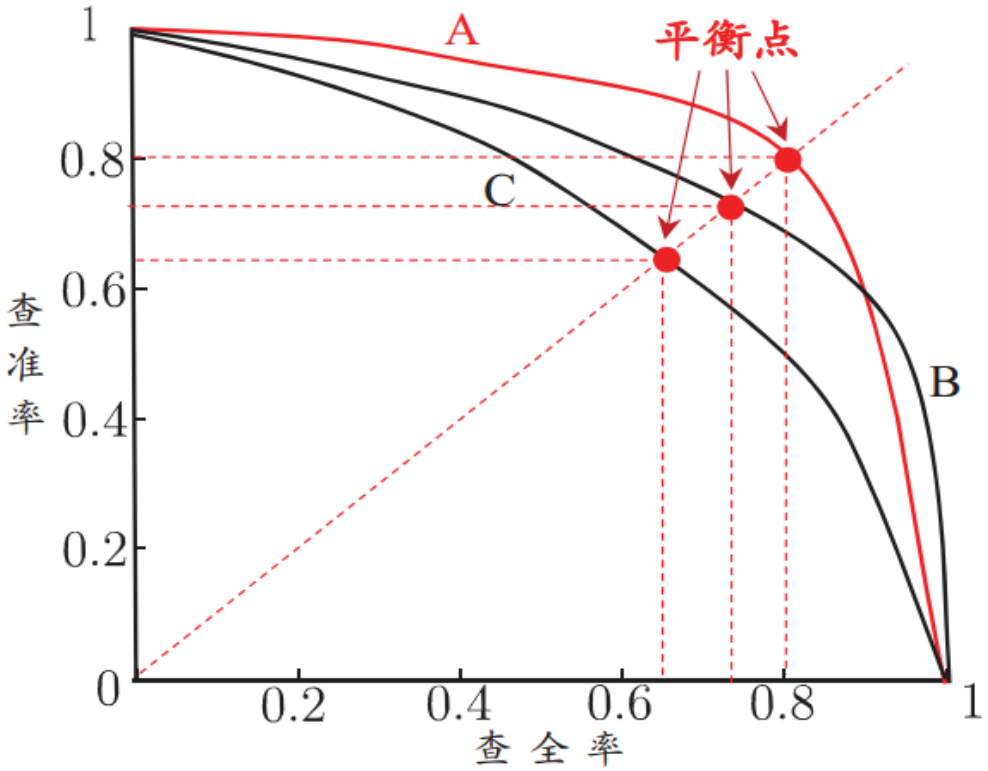
有交叉的分类器性能高低

性能度量——分类问题

根据学习器的预测结果按正例可能性大小对样例进行排序，

并逐个把样本作为正例进行预测，

则可以得到查准率-查全率曲线，简称“P-R曲线”。



PR图:

- 学习器A 优于 学习器C
- 学习器B 优于 学习器C
- 学习器A 优于 学习器B

性能度量——分类问题

比P-R曲线平衡点更用常用的是$F1$度量：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{样例总数} + TP - TN}$$

分类结果混淆矩阵

真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

性能度量——分类问题

比P-R曲线平衡点更用常用的是$F1$度量：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{样例总数} + TP - TN}$$

Error Type	**TextCNN**			**Bert-wwm**		
	P	R	F_1	P	R	F_1
Improper Collocation of Words	85.15	49.90	62.93	97.69	54.90	**70.29**
A Mixture of Sentences	91.26	84.60	87.80	98.96	85.50	**91.74**
Redundancy of Sentence Components	78.46	62.59	69.63	92.58	73.35	**81.85**
Shortage of Logicality	88.58	67.50	76.62	96.06	68.30	**79.84**
Spelling Mistake	69.76	90.90	78.94	83.26	97.00	**89.61**
Incomplete Sentence	77.24	70.60	73.77	78.07	80.80	**79.41**
Improper Words Order	75.02	79.60	77.24	70.14	88.80	**78.38**
Correct Sentences	57.77	96.30	72.22	62.85	98.80	**76.83**
Total (Macor index)	77.91	75.25	74.89	84.95	81.31	**80.99**

Table 4: Comparison between TextCNN and Bert-wwm for Chinese Grammatical Error Identification.

性能度量——分类问题

- 每个类

准确率: Precision=a/(a+b)

召回率: Recall=a/(a+c),

遗漏率: miss rate=1-recall

		Human	
		True	False
Classifier	Yes	a 真正例	b 假正例
	No	c 假负例	d 真负例

性能度量——分类问题

		Human	
		True	False
Classifier	Yes	a 真正例	b 假正例
	No	c 假负例	d 真负例

- 每个类

准确率: Precision=a/(a+b)

召回率: Recall=a/(a+c),

遗漏率: miss rate=1-recall

正确率: accuracy=(a+d)/(a+b+c+d),

错误率: error=(b+c)/(a+b+c+d)=1-accuracy

性能度量——分类问题

		Human	
		True	False
Classifier	Yes	a 真正例	b 假正例
	No	c 假负例	d 真负例

- 每个类

准确率: Precision=a/(a+b)

召回率: Recall=a/(a+c),

遗漏率: miss rate=1-recall

正确率: accuracy=(a+d)/(a+b+c+d),

错误率: error=(b+c)/(a+b+c+d)=1-accuracy

Precision是分类器预测为**某一个类别**的正确率的评价,

Accuracy是对分类器**整体**上的正确率的评价。

宏平均 Macro-average

首先计算出**每个类**的正确率和召回率值，在对所有**取平均**得到总的正确率和召回率的值。

Class	Predicted Class	Correct?
Orange	Lemon	0
Orange	Lemon	0
Orange	Apple	0
Orange	Orange	1
Orange	Apple	0
Lemon	Lemon	1
Lemon	Apple	0
Apple	Apple	1
Apple	Apple	1

$$\textbf{Macro-average}=\frac{1}{m}\sum_{i=1}^{m} average_i$$

Class	Accuracy
Orange	1/5=0.20
Lemon	1/2=0.50
Apple	2/2=1.00

宏平均：(0.20+0.50+1.00)/3=**0.57**

微平均 Micro-average

结合不同类别的贡献大小来计算平均值

Class	Predicted Class	Correct?
Orange	Lemon	0
Orange	Lemon	0
Orange	Apple	0
Orange	Orange	1
Orange	Apple	0
Lemon	Lemon	1
Lemon	Apple	0
Apple	Apple	1
Apple	Apple	1

$$Micro - F = \sum_{i=1}^{m}(n_i \cdot F_i) \Big/ \sum_{i=1}^{m} n_i$$

微平均: 4/9=**0.44**

(预测正确的样本个数)

性能度量——分类问题

“假正例率”为横轴，

“真正例率”为纵轴，

可得到ROC曲线，

全称“受试者工作特征”

真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

假正例率 $FPR = \dfrac{FP}{TN + FP}$

真正例率 $TPR = \dfrac{TP}{TP + FN}$

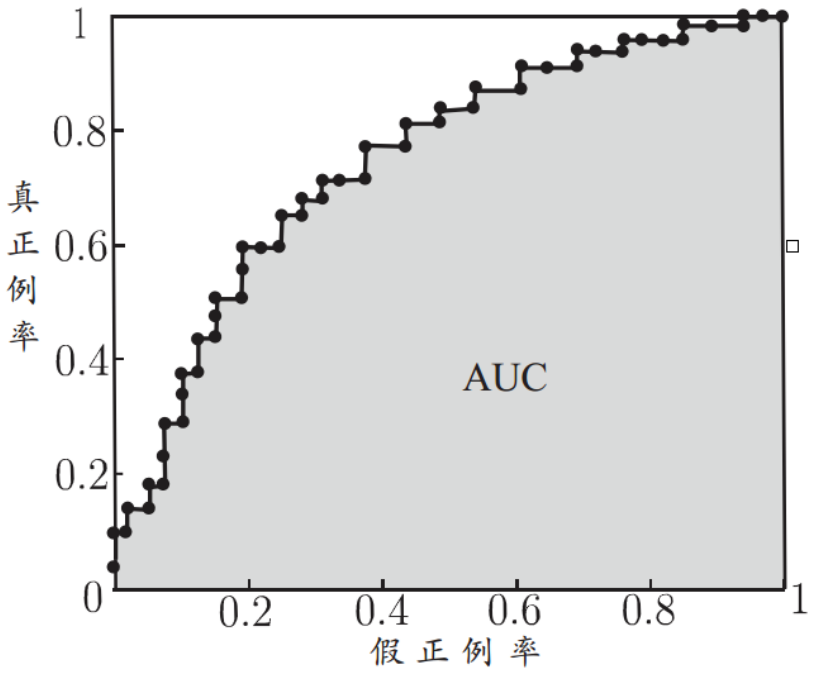
性能度量——分类问题

“假正例率”为横轴，

“真正例率”为纵轴，

可得到ROC曲线，

全称“受试者工作特征”

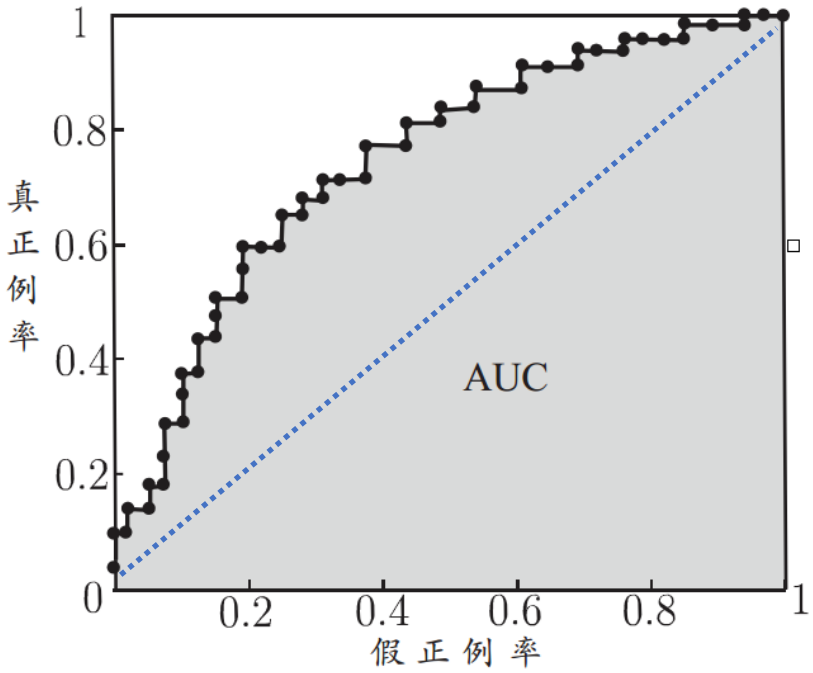


基于有限样例绘制的 ROC 曲线与 AUC

性能度量——分类问题

若某个学习器的ROC曲线被另一个学习器的曲线“包住”，则后者性能优于前者；

否则如果曲线交叉，可以根据ROC曲线下面积大小进行比较，也即**AUC值**。



基于有限样例绘制的 ROC 曲线与 AUC

假设ROC曲线由为$\{(x_1, y_1), \cdots, (x_n, y_n)\}$的点按需连接而成且有$x_1 = 0, x_n = 1$，则**AUC**可估算为：

$$\mathrm{AUC} = \frac{1}{2}\sum_{j=1}^{m-1}(x_{j+1} - x_j)(y_{j+1} + y_j)$$

中山大學
SUN YAT-SEN UNIVERSITY

第2章 模型评估与选择

1. 训练误差与测试误差
2. 过拟合与模型选择
3. 性能度量
4. 偏差与方差

“误差”包含了哪些因素？

如何去解释？

“偏差-方差分解”可以用来帮助解释泛化性能。

偏差-方差分解试图对学习算法期望的泛化性能进行拆解。

偏差与方差

泛化误差可分解为偏差、方差与噪声之和：

- 偏差度量了学习算法期望预测与真实结果的偏离程度；即刻画了学习算法本身的拟合能力；
- 方差度量了同样大小训练集的变动所导致的学习性能的变化；即刻画了数据扰动所造成的影响；
- 噪声表达了在当前任务上任何学习算法所能达到的期望泛化误差的下界；即刻画了学习问题本身的难度。

泛化性能是由学习方法的能力、数据的充分性以及学习任务本身的难度所共同决定的。

给定学习任务为了取得好的泛化性能，需要使偏差小（充分拟合数据）而且方差较小（减少数据扰动产生的影响）。

中山大學
SUN YAT-SEN UNIVERSITY

第3章 线性模型

1. 基本形式
2. 单变量线性回归
3. 多元线性回归
4. 广义线性模型
5. 对数几率回归
6. Softmax回归

沈颖 副教授

sheny76@mail.sysu.edu.cn

线性模型——基本形式

线性模型一般形式

$$f(\boldsymbol{x}) = \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_d x_d + b$$

其中x_i是$\boldsymbol{x}$在第i个属性上的取值

$\boldsymbol{x} = (x_1; x_2; \cdots; x_d)$是由$d$个属性描述的示例

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

线性模型——基本形式

线性模型一般形式

$$f(\boldsymbol{x}) = \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_d x_d + b$$

其中x_i是$\boldsymbol{x}$在第i个属性上的取值

$\boldsymbol{x} = (x_1; x_2; \cdots; x_d)$是由$d$个属性描述的示例

$\boldsymbol{\omega}$直观表达了各属性在预测中的重要性，具有有很好的可解释性

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

线性模型——基本形式

线性模型一般形式

$$f(\boldsymbol{x}) = \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_d x_d + b$$

向量形式

$$f(\boldsymbol{x}) = \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x} + b$$

- 形式简单、易于建模
- 可解释性
- 许多非线性模型可在线性模型的基础上通过引入高维映射或者层级结构来得到。

线性模型——基本形式

给定数据集$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$，
其中$\boldsymbol{x}_i = (x_{i1}; x_{i2}; \cdots; x_{id})$，$y_i \in \mathcal{Y}$。

线性回归（linear regression）的目的：

- 学得一个线性模型，从而
- 尽可能准确地预测实值输出标记。

$$f(\boldsymbol{x}_i) = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b，\text{使得} f(\boldsymbol{x}_i) \simeq y_i$$

$$f(\boldsymbol{x}_i) = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b，\ 使得 f(\boldsymbol{x}_i) \simeq y_i$$

离散属性处理

- 有“序”关系：连续化为连续值

  二值属性“身高”的取值"高""矮"可转化为{1.0, 0.0}

  三值属性"高度"的取值"高""中""低"可转化为{1.0, 0.5, 0.0}

- 无“序”关系：有k个属性值，则转换为k维向量

  例如属性“瓜类”的取值“西瓜”“南瓜”“黄瓜”可转化为(0, 0, 1), (0, 1, 0), (1, 0, 0)

中山大學
SUN YAT-SEN UNIVERSITY

第3章 线性模型

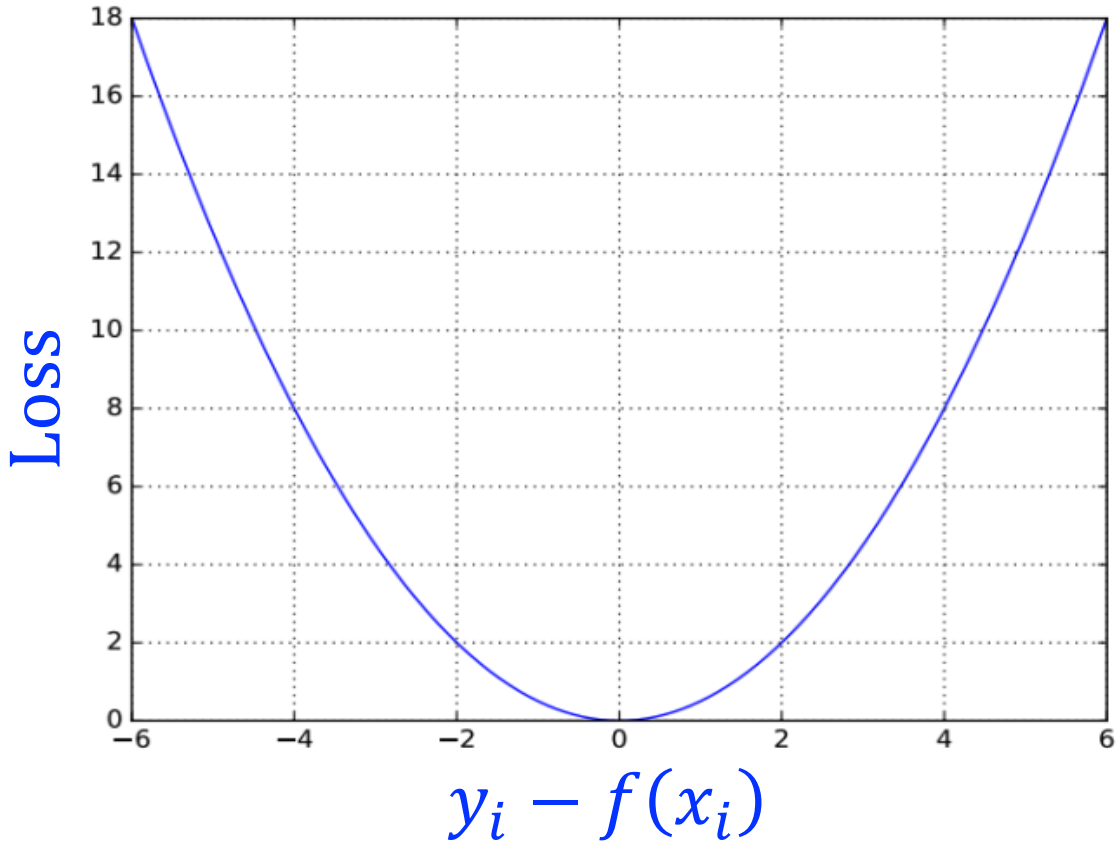
1. 基本形式
2. 单变量线性回归
3. 多元线性回归
4. 广义线性模型
5. 对数几率回归
6. Softmax回归

单变量线性回归

数据： $D = \{(x_i, y_i)\}_{i=1}^{m}$，其中$x_i, y_i \in \mathbb{R}$

模型： $f(x_i) = \omega x_i + b$，使得$f(x_i) \simeq y_i$

策略：平方损失， $\mathcal{L}\big(y_i, f(x_i)\big) = \big(y_i - f(x_i)\big)^2$



离样本真实值越大，损失越大

单变量线性回归

线性回归试图学得

$$f(\boldsymbol{x}_i) = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b，\text{使得} f(\boldsymbol{x}_i) \simeq y_i$$

如何确定$\boldsymbol{\omega}$ 和 b 呢？

显然，关键在于如何衡量$f(\boldsymbol{x})$与$\boldsymbol{y}$之间的差别

单变量线性回归

线性回归试图学得

$$f(\boldsymbol{x}_i) = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b，\text{使得} f(\boldsymbol{x}_i) \simeq y_i$$

如何确定$\boldsymbol{\omega}$ 和 b 呢？

显然，关键在于如何衡量$f(\boldsymbol{x})$与$\boldsymbol{y}$之间的差别

上一节课介绍的均方误差，是回归任务中最常用的性能度量

单变量线性回归

线性回归试图学得

$$f(\boldsymbol{x}_i) = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b, \text{ 使得} f(\boldsymbol{x}_i) \simeq y_i$$

试图基于均方误差最小化来进行模型求解

$$(\omega^*, b^*) = \underset{(\omega,b)}{\operatorname{argmin}} \sum_{i=1}^{m} \big(y_i - f(x_i)\big)^2 = \underset{(\omega,b)}{\operatorname{argmin}} \sum_{i=1}^{m} (y_i - \omega x_i - b)^2$$

均方误差有非常好的几何意义
它对应了常用的欧几里得距离或简称"欧氏距离"
(Euclidean distance).

$$(\omega^*, b^*) = \underset{(\omega,b)}{\operatorname{argmin}} \sum_{i=1}^{m} \big(y_i - f(x_i)\big)^2$$

闵可夫斯基距离（闵式距离）

$$d(\boldsymbol{a}, \boldsymbol{b}) = \left(\sum_{k=1}^{M} (a_k - b_k)^q \right)^{1/q}$$

闵可夫斯基距离（闵式距离）

$$d(\boldsymbol{a},\boldsymbol{b}) = \left(\sum_{k=1}^{M}(a_k - b_k)^q\right)^{1/q}$$

a, b 分别为两个待比较文本的**向量表示**

k 表示每个样品的**指标数**，$k = 1,2,\ldots M$

q 表示**阶数**，通常取值$q = 1,2$ 或者正无穷

闵可夫斯基距离（闵式距离）

$$d(a,b)=\left(\sum_{k=1}^{M}(a_k-b_k)^q\right)^{1/q}$$

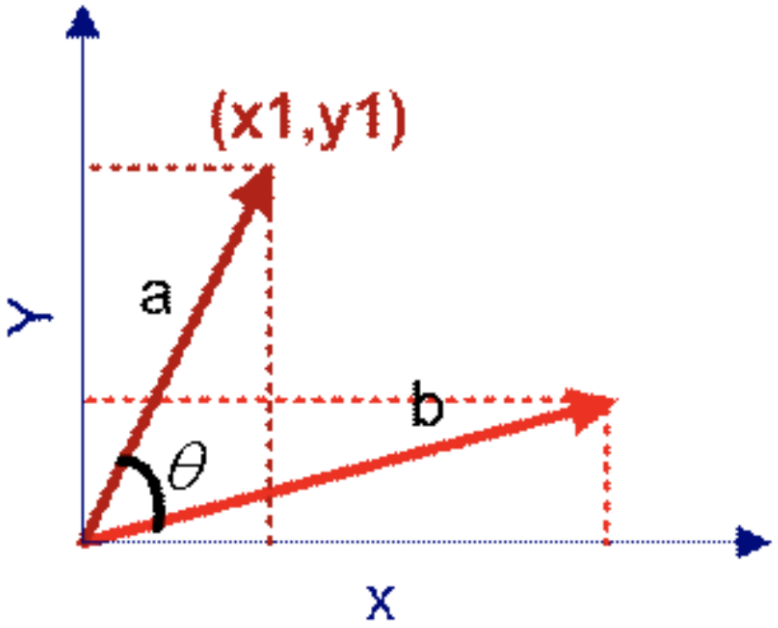
q 表示**阶数**，

通常取值$q=1,2$ 或者正无穷

根据 q 取值的不同，闵式距离可以分为**欧氏距离**、**曼哈顿距离**、**切比雪夫距离**

2. 基于夹角余弦的度量

余弦相似度通过测量两个向量之间夹角的**余弦值**度量它们之间的**相似性**。

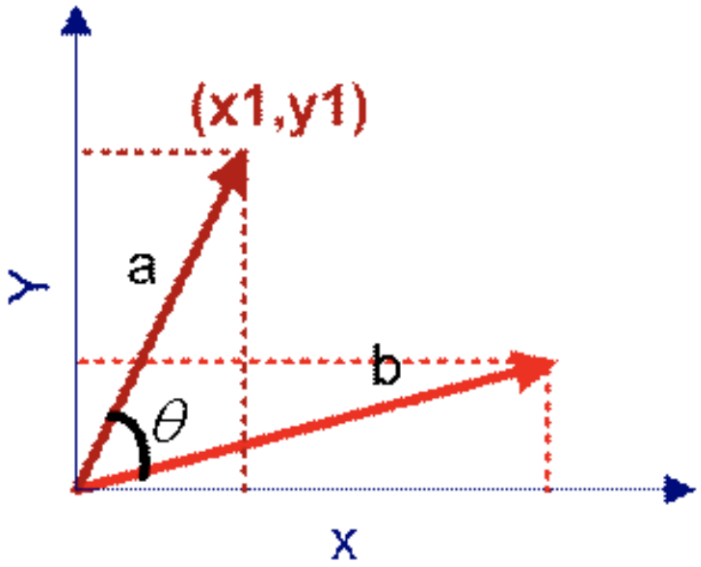


$$\cos(\theta) = \frac{a \bullet b}{||a|| \times ||b||}$$

$$= \frac{(x_1, y_1) \bullet (x_2, y_2)}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$

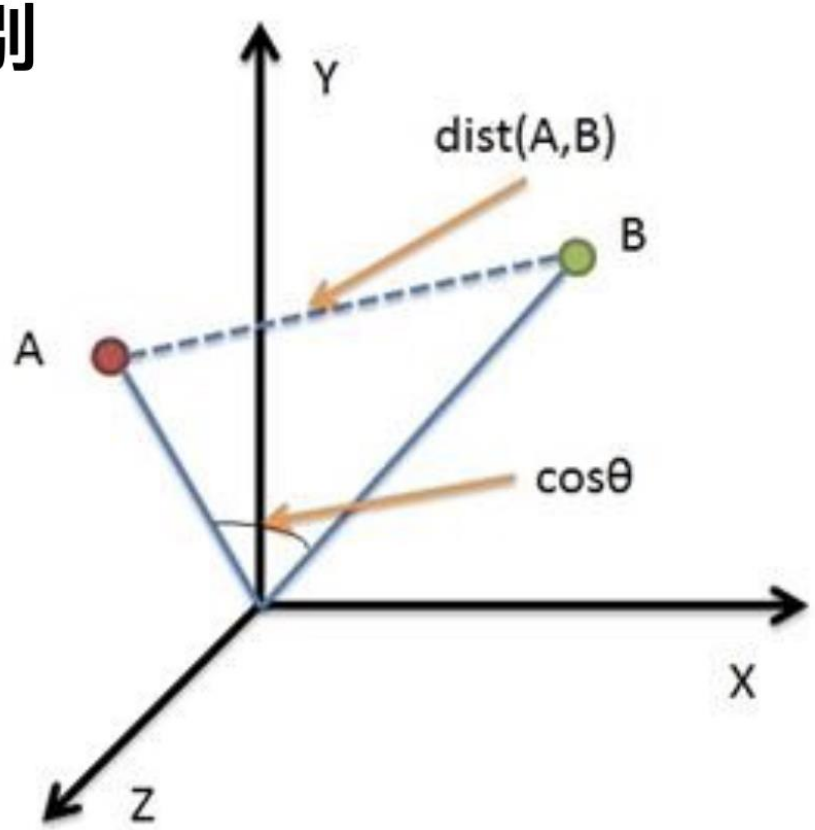
$$= \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$

单变量线性回归



- 余弦相似度通常用于正空间，因此其取值范围通常为**[−1，1]**。
- 两个向量有**相同的指向**时，余弦相似度的值为**1**；
- 两个**向量夹角为90°** 时，余弦相似度的值为**0**；
- 两个向量**指向完全相反**的方向时，余弦相似度的值为 **-1**。

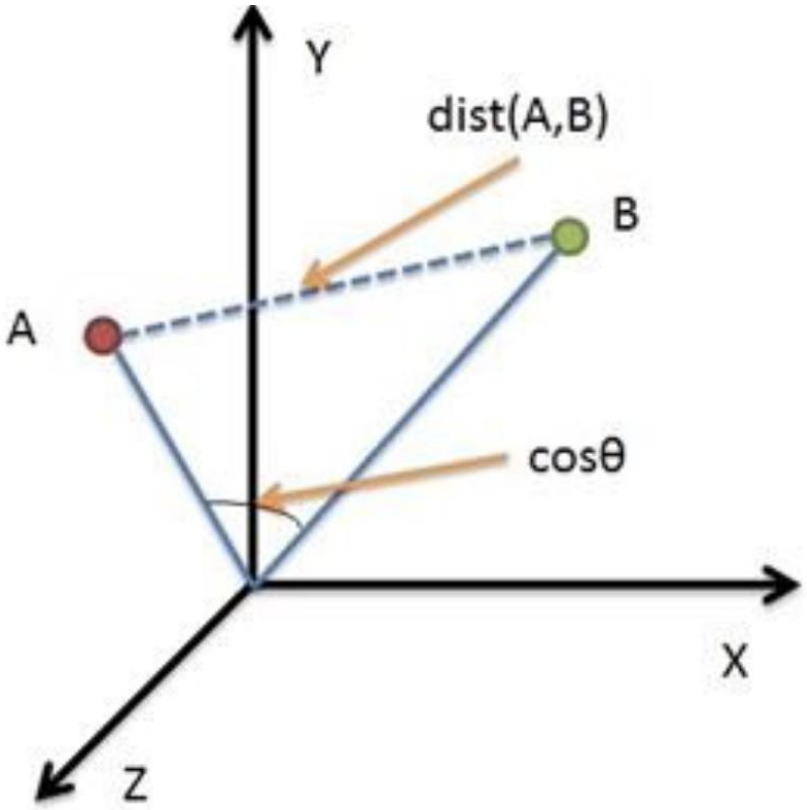
欧氏距离和余弦距离的区别



欧氏距离衡量的是空间各点的**绝对距离**，跟各个点所在的**位置坐标**直接相关；

余弦距离衡量的是空间向量的**夹角**，更加体现在**方向上的差异**，而不是位置。

欧氏距离和余弦距离的区别



如保持**A点**位置不变，**B点**朝原方向**远离坐标轴原点**，

则**余弦距离**保持不变，

而**欧氏距离** 发生改变。

单变量线性回归

线性回归试图学得

$$f(\boldsymbol{x}_i) = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b, \text{ 使得} f(\boldsymbol{x}_i) \simeq y_i$$

试图基于均方误差最小化来进行模型求解

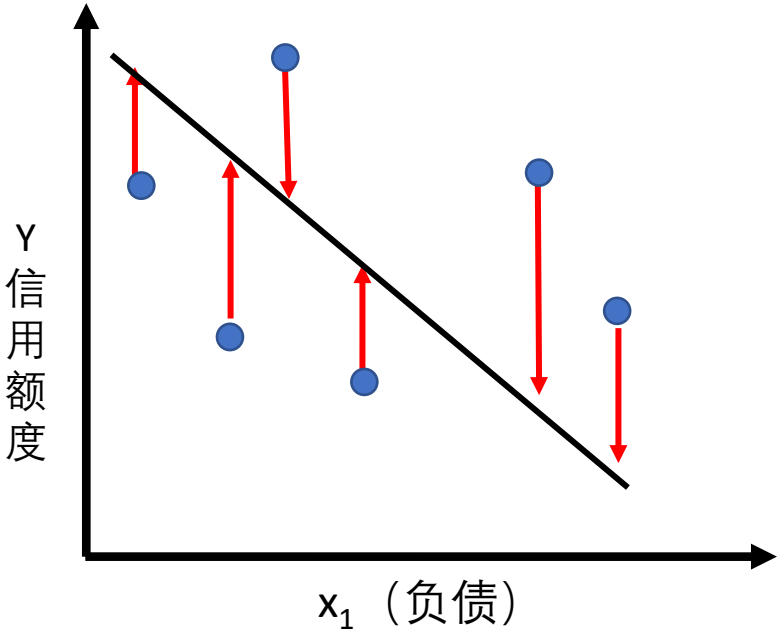
$$(\omega^*, b^*) = \underset{(\omega,b)}{\operatorname{argmin}} \sum_{i=1}^{m} \big(y_i - f(x_i)\big)^2 = \underset{(\omega,b)}{\operatorname{argmin}} \sum_{i=1}^{m} (y_i - \omega x_i - b)^2$$

基于均方误差最小化来进行模型求解的方法称为
"最小二乘法"（least square method）.

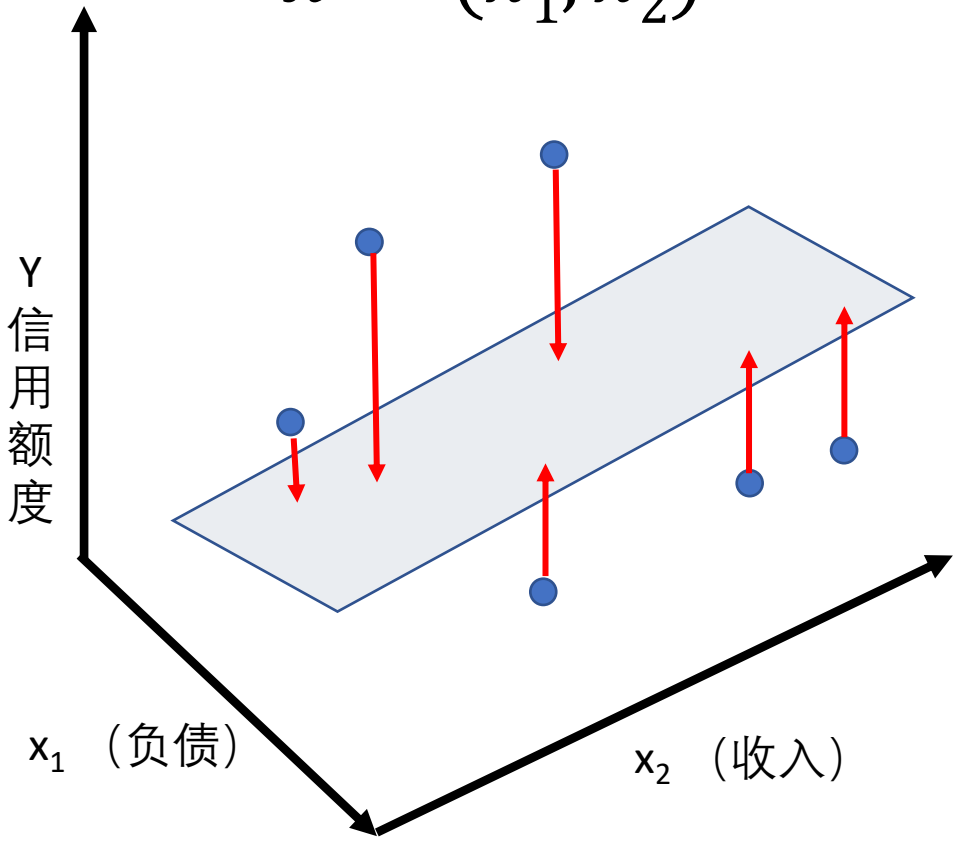
单变量线性回归

在线性回归中，最小二乘法就是试图找到一条直线，
使所有样本到直线上的欧氏距离之和最小.

$$x = (x_1)$$



$$x = (x_1, x_2)$$



单变量线性回归

算法：

线性回归模型的最小二乘“参数估计”，

就是求解ω和b使$E_{(\omega,b)} = \sum_{i=1}^{m}(y_i - \omega x_i - b)^2$最小化的过程

$E_{(\omega,b)}$是关于ω和b的凸函数，

当它关于ω和b的导数均为零时，可得到ω和b的最优解

算法：

线性回归模型的最小二乘“参数估计”，

就是求解ω和b使$E_{(\omega,b)} = \sum_{i=1}^{m}(y_i - \omega x_i - b)^2$最小化的过程

将$E_{(\omega,b)}$分别对ω和b求导：

$$\frac{\partial E_{(\omega,b)}}{\partial \omega} = 2\left(\omega\sum_{i=1}^{m} {x_i}^2 - \sum_{i=1}^{m}(y_i - b)x_i\right)$$

$$\frac{\partial E_{(\omega,b)}}{\partial b} = 2\left(mb - \sum_{i=1}^{m}(y_i - \omega x_i)\right)$$

$$\frac{\partial E_{(\omega,b)}}{\partial \omega} = 2\left(\omega \sum_{i=1}^{m} {x_i}^2 - \sum_{i=1}^{m}(y_i - b)x_i\right)$$

$$\frac{\partial E_{(\omega,b)}}{\partial b} = 2\left(mb - \sum_{i=1}^{m}(y_i - \omega x_i)\right)$$

令以上关于ω和b的导数皆为0，
可得到ω和b 最优解的闭式解

$$\omega^* = \frac{\sum_{i=1}^{m} y_i(x_i - \bar{x})}{\sum_{i=1}^{m} {x_i}^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2} \qquad b^* = \frac{1}{m}\sum_{i=1}^{m}(y_i - \omega x_i)$$

$\bar{x} = \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)$ 为 x 的均值

单变量线性回归

$$\frac{\partial E_{(\omega,b)}}{\partial \omega} = 2\left(\omega\sum_{i=1}^{m} {x_i}^2 - \sum_{i=1}^{m}(y_i - b)x_i\right)$$

$$\frac{\partial E_{(\omega,b)}}{\partial b} = 2\left(mb - \sum_{i=1}^{m}(y_i - \omega x_i)\right)$$

令以上关于ω和b的导数皆为0，
可得到ω和b 最优解的闭式解

$$\omega^* = \frac{\sum_{i=1}^{m} y_i(x_i - \bar{x})}{\sum_{i=1}^{m} {x_i}^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2}$$

$$b^* = \frac{1}{m}\sum_{i=1}^{m}(y_i - \omega x_i)$$

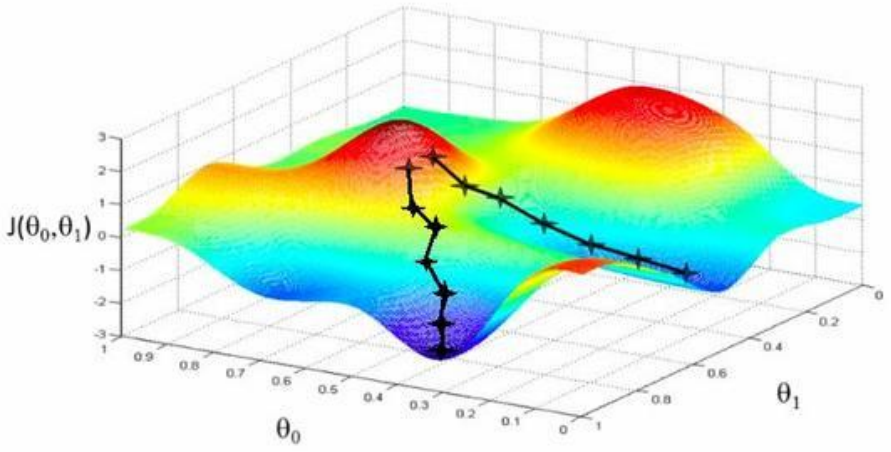
最终学得的单变量线性回归模型为：$\hat{f}(x_{m+1}) = \omega^* x_{m+1} + b^*$

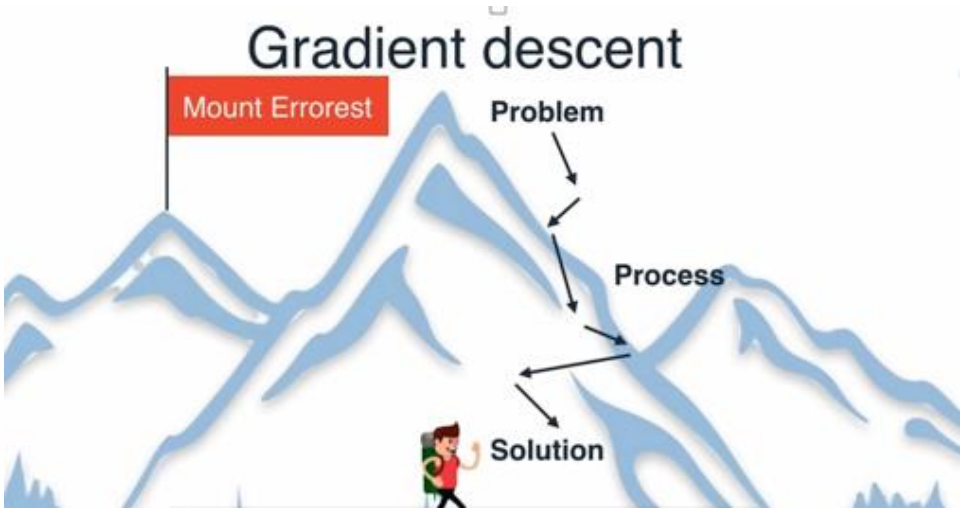
梯度下降法

算法： 经典的数值优化算法，如梯度下降法（gradient descent method）、牛顿法（Newton method）

梯度下降法（gradient descent method）

梯度下降法是一种迭代算法。



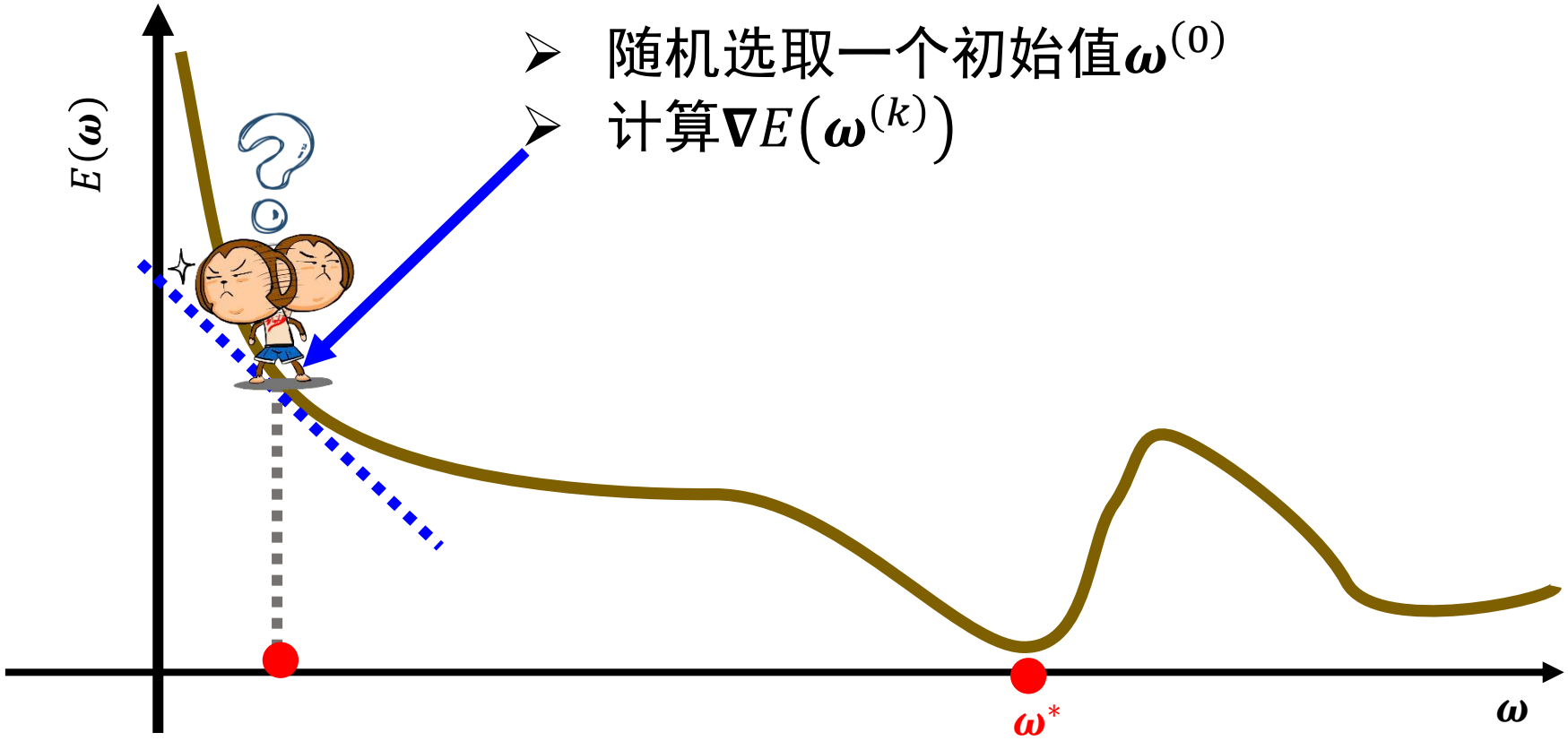


梯度下降法

梯度下降法算法

输入：目标函数$E(\boldsymbol{\omega})$，梯度函数$\nabla E(\boldsymbol{\omega})$；

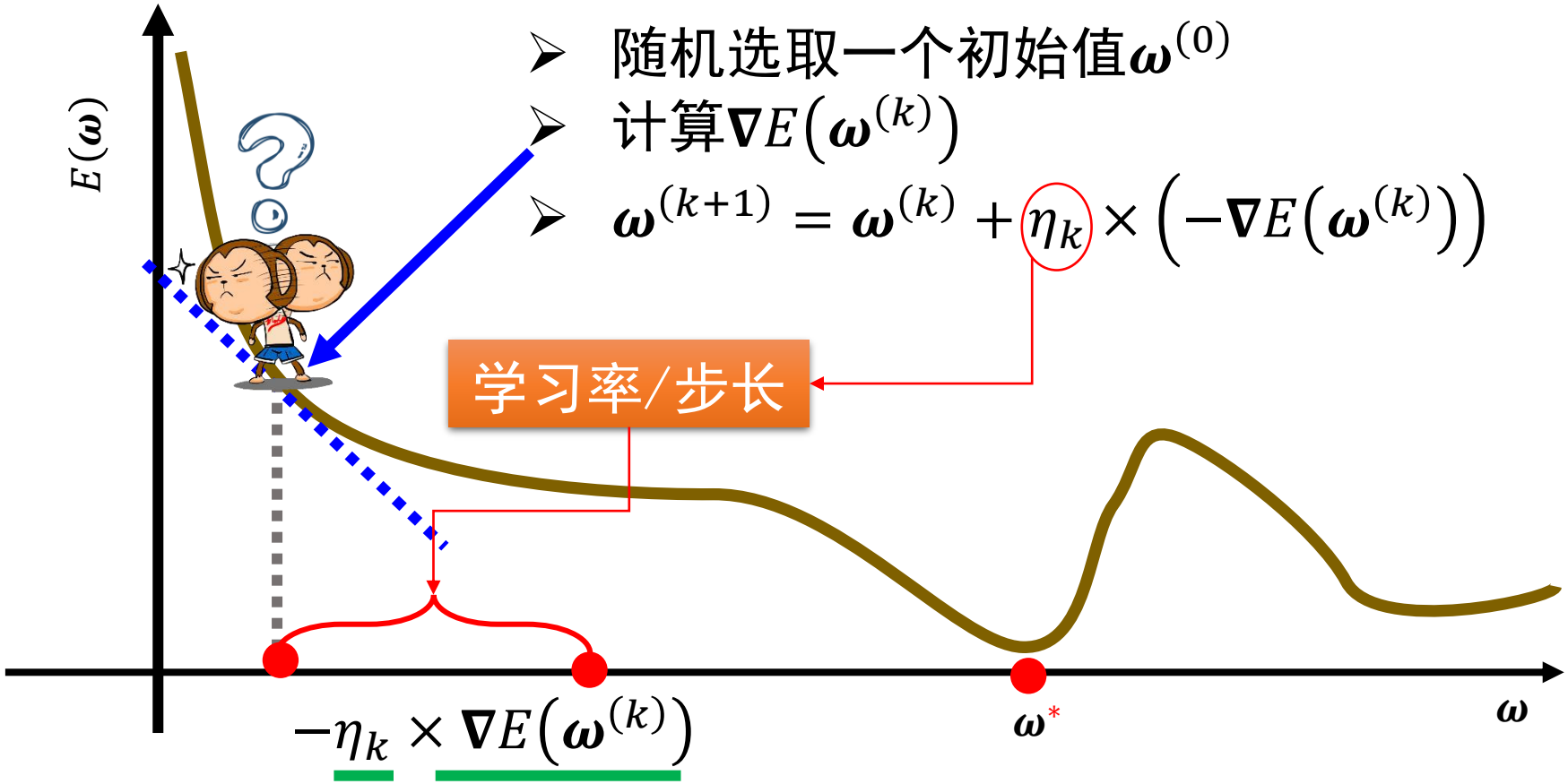
输出：$E(\boldsymbol{\omega})$的极小点$\boldsymbol{\omega}^*$。



梯度下降法算法

输入：目标函数$E(\boldsymbol{\omega})$，梯度函数$\nabla E(\boldsymbol{\omega})$；

输出：$E(\boldsymbol{\omega})$的极小点$\boldsymbol{\omega}^*$。



梯度下降法——以求解对数几率模型为例

梯度下降法算法

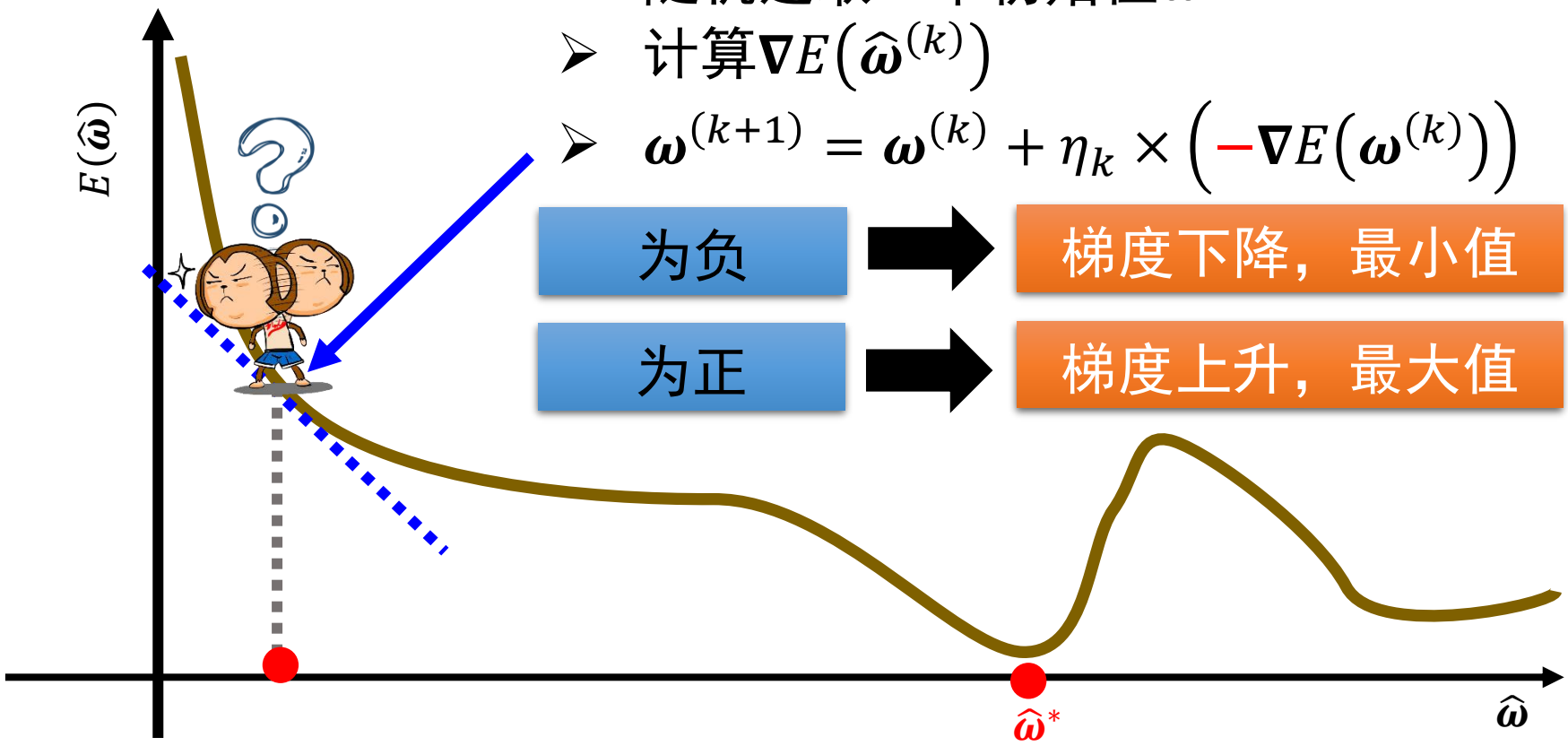
输入：目标函数$E(\hat{\boldsymbol{\omega}})$，梯度函数$\nabla E(\hat{\boldsymbol{\omega}})$；

输出：$E(\hat{\boldsymbol{\omega}})$的极小点$\hat{\boldsymbol{\omega}}^*$。

- 随机选取一个初始值$\hat{\boldsymbol{\omega}}^{(0)}$
- 计算$\nabla E\left(\hat{\boldsymbol{\omega}}^{(k)}\right)$
- $\boldsymbol{\omega}^{(k+1)} = \boldsymbol{\omega}^{(k)} + \eta_k \times \left(-\nabla E\left(\boldsymbol{\omega}^{(k)}\right)\right)$

为负 → 梯度下降，最小值

为正 → 梯度上升，最大值



梯度下降法

梯度下降法算法

梯度上升 找最大值 $\boldsymbol{\omega}^{(k+1)} = \boldsymbol{\omega}^{(k)} + \eta_k \times \left(\nabla E\left(\boldsymbol{\omega}^{(k)}\right)\right)$

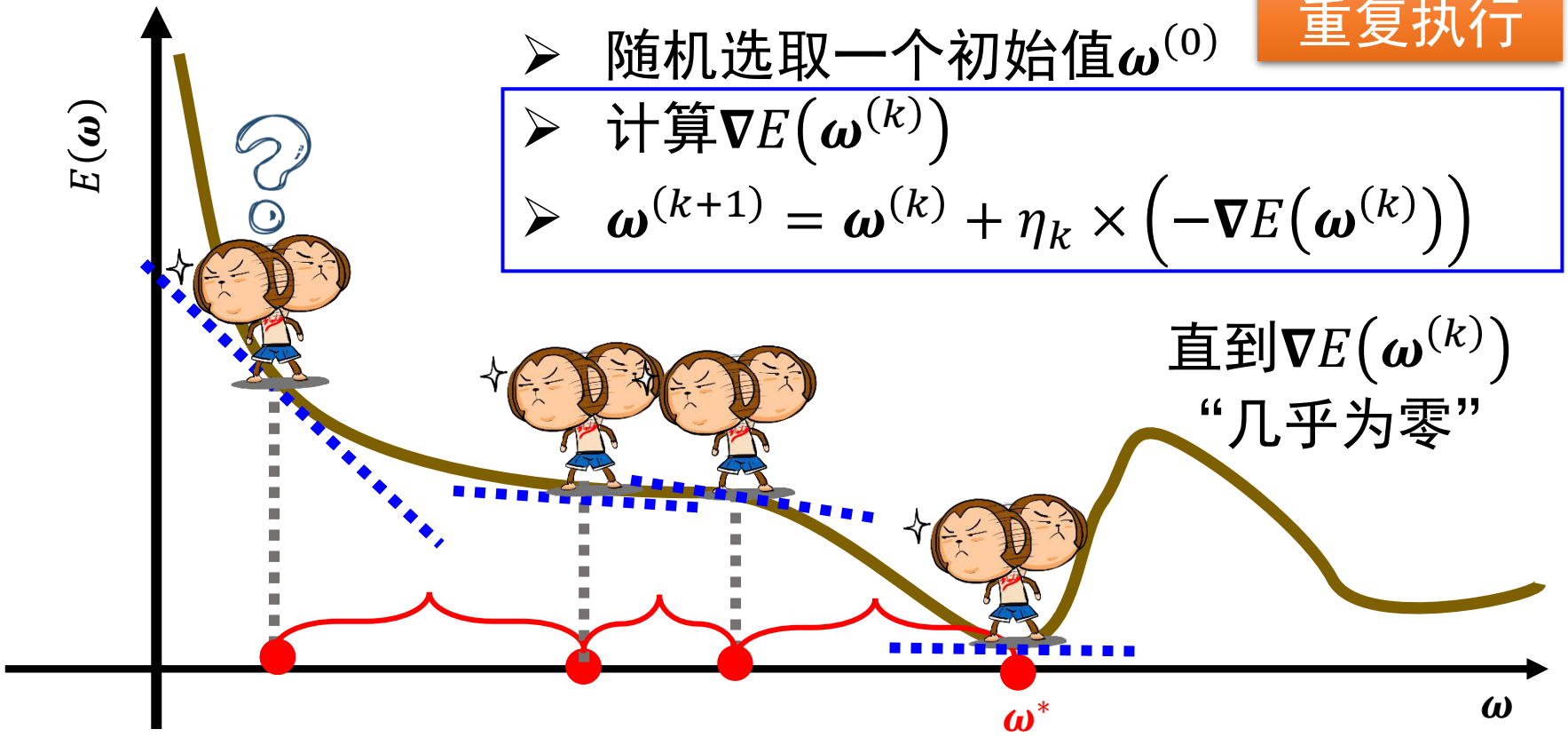
输入：目标函数$E(\boldsymbol{\omega})$，梯度函数$\nabla E(\boldsymbol{\omega})$；

输出：$E(\boldsymbol{\omega})$的极小点$\boldsymbol{\omega}^*$。

重复执行

- 随机选取一个初始值$\boldsymbol{\omega}^{(0)}$
- 计算$\nabla E\left(\boldsymbol{\omega}^{(k)}\right)$
- $\boldsymbol{\omega}^{(k+1)} = \boldsymbol{\omega}^{(k)} + \eta_k \times \left(-\nabla E\left(\boldsymbol{\omega}^{(k)}\right)\right)$

直到$\nabla E\left(\boldsymbol{\omega}^{(k)}\right)$“几乎为零”



第3章 线性模型

1. 基本形式
2. 单变量线性回归
3. 多元线性回归
4. 广义线性模型
5. 对数几率回归
6. Softmax回归

多元线性回归（multivariate linear regression）

数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$，其中$\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$

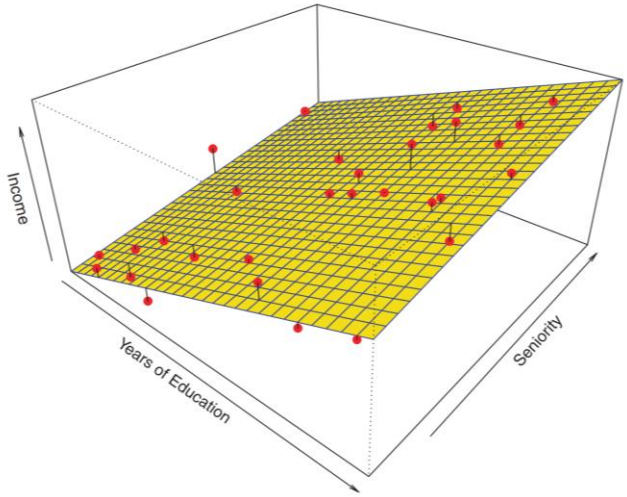
多回归任务：将向量映射为向量

设目标值$\mathbf{y}_i$，$i = 1, 2, \ldots, n$，为 m 维空间向量，则需要学习 m 个线性变换：

$$f_i(\mathbf{x}) = w_{1i}x_1 + w_{2i}x_2 + \cdots + w_{di}x_d + b_i$$

$$= \mathbf{w}_i^T \mathbf{x} + b_i$$

$i = 1, 2, \cdots, m$



$f_i(\mathbf{x})$ 负责将$\mathbf{x}$映射到其目标向量$\mathbf{y}$的第 i 个分量。

多元线性回归（multivariate linear regression）

数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$，其中$\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$

多回归任务：将向量映射为向量

设目标值$\mathbf{y}_i$，$i = 1, 2, \ldots, n$，为 m 维空间向量，则需要学习 m 个线性变换：

$$f_i(\mathbf{x}) = w_{1i}x_1 + w_{2i}x_2 + \cdots + w_{di}x_d + b_i$$
$$= \mathbf{w}_i^T \mathbf{x} + b_i$$

试图学习一组线性变换
以尽可能地预测实值输出标记向量

多元线性回归（multivariate linear regression）

单变量线性回归模型为：$\hat{f}(x_{m+1}) = \omega^* x_{m+1} + b^*$

数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$，其中$\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$

模型：$f(\boldsymbol{x}_i) = \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}_i + b$，使得$f(\boldsymbol{x}_i) \simeq y_i$，其中$\boldsymbol{\omega} \in \mathbb{R}^d$

策略：平方损失，$\mathcal{L}\big(y_i, f(\boldsymbol{x}_i)\big) = \big(y_i - f(\boldsymbol{x}_i)\big)^2$

算法：最小二乘“参数估计”，得到$\hat{\boldsymbol{\omega}}^*$

多元线性回归模型为：$\hat{f}(\hat{\boldsymbol{x}}_{m+1}) = \hat{\boldsymbol{x}}_{m+1}^{\mathrm{T}} \hat{\boldsymbol{\omega}}^*$，

其中$\hat{\boldsymbol{x}}_{m+1} = (\boldsymbol{x}_{m+1}; 1) \in \mathbb{R}^{d+1}, \hat{\boldsymbol{\omega}}^* = (\boldsymbol{\omega}^*; b^*) \in \mathbb{R}^{d+1}$

中山大學
SUN YAT-SEN UNIVERSITY

第3章 线性模型

1. 基本形式
2. 单变量线性回归
3. 多元线性回归
4. 广义线性模型
5. 对数几率回归
6. Softmax回归

广义线性模型（generalized linear model）

$y = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x} + b$转换成

- $\ln y = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x} + b$或者
- $y = e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b}$

一般形式

$$y = g^{-1}\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x} + b\right),$$

单调可微函数的联系函数（link function）

对数线性回归是广义线性模型在 $g(\cdot) = \ln(\cdot)$ 时的特例.

中山大學
SUN YAT-SEN UNIVERSITY

第3章 线性模型

1. 基本形式
2. 单变量线性回归
3. 多元线性回归
4. 广义线性模型
5. 对数几率回归
6. 多分类学习
7. Softmax回归

沈颖 副教授

sheny76@mail.sysu.edu.cn

对数几率回归

- **如何用线性回归做分类任务，应该怎么做呢？**
- 一种常用的做法是采用类别标签向量：
  - 对于两类分类问题，我们期望正例样本被回归到“+1”,负例样本被回归到“-1”。
  - 对于多类问题，通常将样本回归到**one-zero hot**向量，该向量只有一个元素为1，其余元素全为零。

对数几率回归则利用广义线性回归的思想，期望找到一个单调可微函数**将分类任务的真实标记与线性回归模型的预测值**联系起来。

对数几率回归

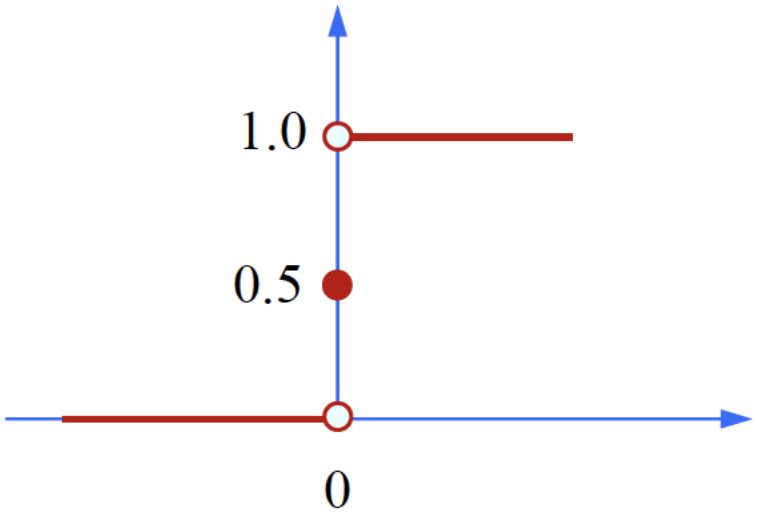
数据：$D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，$\boldsymbol{x}_i\in\mathbb{R}^d, y_i\in\{0,1\}$

模型：$f(\boldsymbol{x}_i)=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+b$，使得$f(\boldsymbol{x}_i)\simeq g(y_i)$，其中$\boldsymbol{\omega}\in\mathbb{R}^d$

考虑二分类任务，当输出标记$y_i\in\{0,1\}$时，
需要将$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+b$的实值转换为0/1值

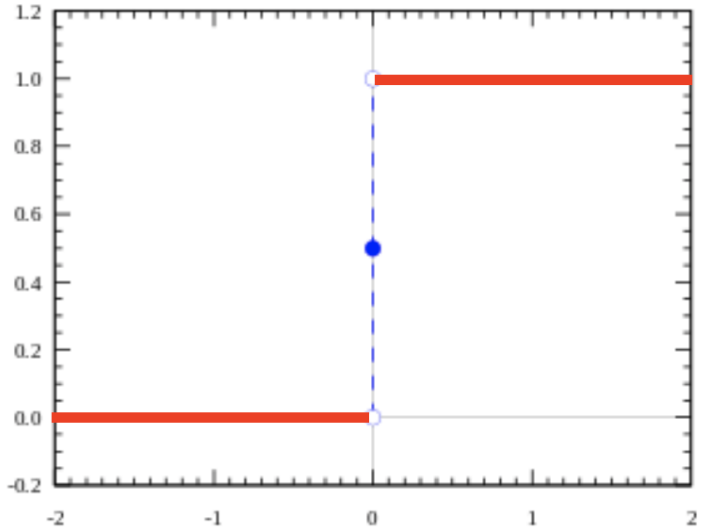
最理想的是“单位阶跃函数”（unit-step function）

$$y=\begin{cases}0, & z=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b<0\\ 0.5, & z=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b=0\\ 1, & z=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b>0\end{cases}$$



若预测值 z 大于零就判为正例，
小于零则判为反例，
预测值为临界值零则可任意判别

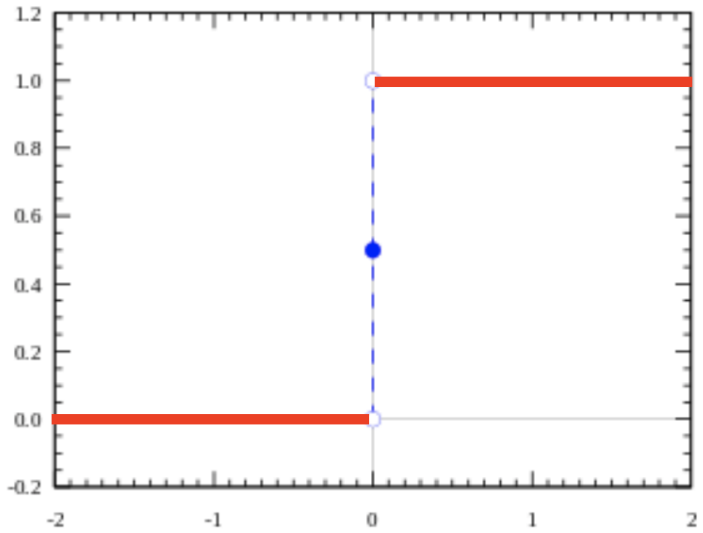
阶跃函数



最简单的二分类非线性激活函数——阶跃函数（Step Function）

当输入大于0就被分类到 1（100% 被激活），小于0就分到 0（没有被激活）

阶跃函数



最简单的二分类非线性激活函数——阶跃函数（Step Function）

当输入大于0就被分类到 1（100% 被激活 ），小于0就分到 0（没有被激活）

可是激活值只有100%或者0%有失偏颇，我们希望它可以是0%—100%任意值。

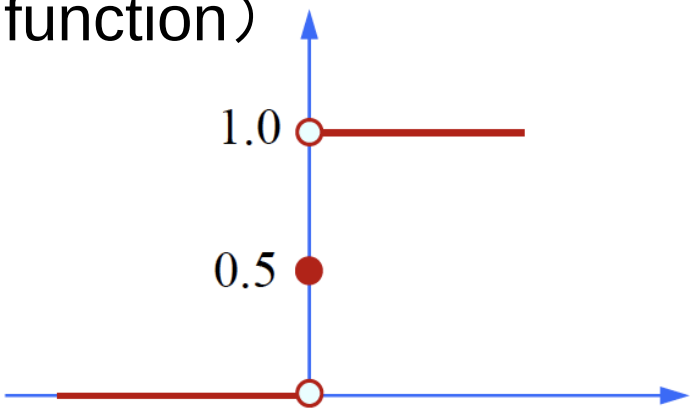
即，值越大，激活程度越高。

对于分类任务，也就意味着它属于这一类的概率越大

对数几率回归

理想的“单位阶跃函数”（unit-step function）

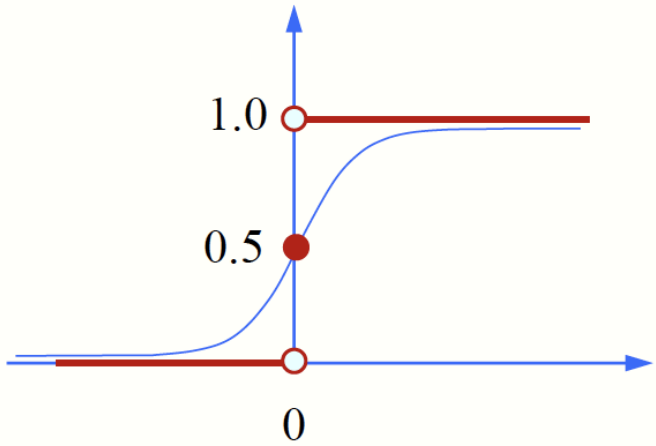
$$y=\begin{cases}0, & z=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b<0\\ 0.5, & z=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b=0\\ 1, & z=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b>0\end{cases}$$



- 阶跃函数不连续，且反函数不存在。需要找到可近似单位阶跃函数的替换函数（surrogate function）。
- 对数几率函数（logistic function）正是这样的函数（一种sigmoid函数）：

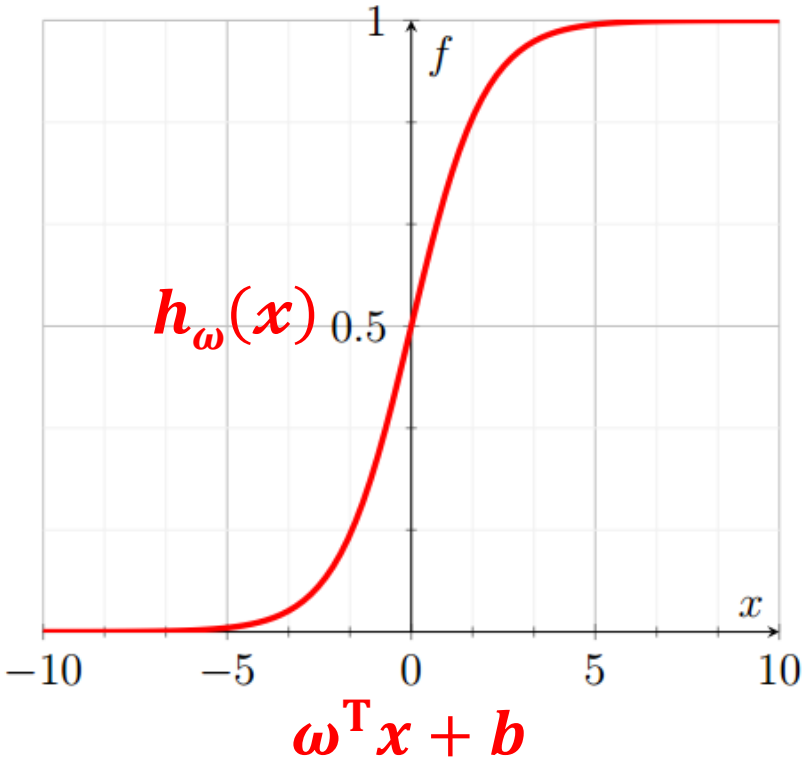
$$y=\frac{1}{1+e^{-z}}$$

对数函数



将z值转化为一个接近0或1的y值，并且其输出值在$z=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b=0$ 附近变化很陡

Sigmoid函数



优点：

- 梯度的“平滑性”
- 输出在“0 – 1区间”

缺点：

- 梯度消失问题：神经网络使用 Sigmoid 激活函数进行反向传播时，输出接近0或1的神经元其梯度趋近于0
- 计算成本问题：涉及指数计算
- 不以零为中心：Sigmoid 输出不以零为中心

对数几率回归

logistic regression，亦称逻辑斯特回归

特别需注意到，虽然它的名字是"回归"，但却是分类学习方法.

优点：

1. 直接对分类可能性进行建模，无需事先假设数据分布。

↓

避免了假设分布不准确所带来的问题

2. 不仅预测出"类别"，而是可得到近似概率预测

↓

对许多需利用概率辅助决策的任务很有用

3. 对率函数是任意阶可导的凸函数，有很好的数学性质，
现有的许多数值优化算法都可直接用于求取最优解

中山大學
SUN YAT-SEN UNIVERSITY

第3章 线性模型

1. 基本形式
2. 单变量线性回归
3. 多元线性回归
4. 广义线性模型
5. 对数几率回归
6. 多分类学习
7. Softmax回归

多分类学习

数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$，$y_i \in \{C_i, \cdots, C_N\}$

多分类学习方法

- 二分类学习方法推广到多类（如多项对数几率回归）
- 利用二分类学习器解决多分类问题（常用）
  - 对问题进行拆分，为拆出的每个二分类任务训练一个分类器
  - 对于每个分类器的预测结果进行集成以获得最终的多分类结果

拆分策略

- 一对一（One vs. One, OvO）
- 一对其余（One vs. Rest, OvR）
- 多对多（Many vs. Many, MvM）

- **拆分策略** —对—：

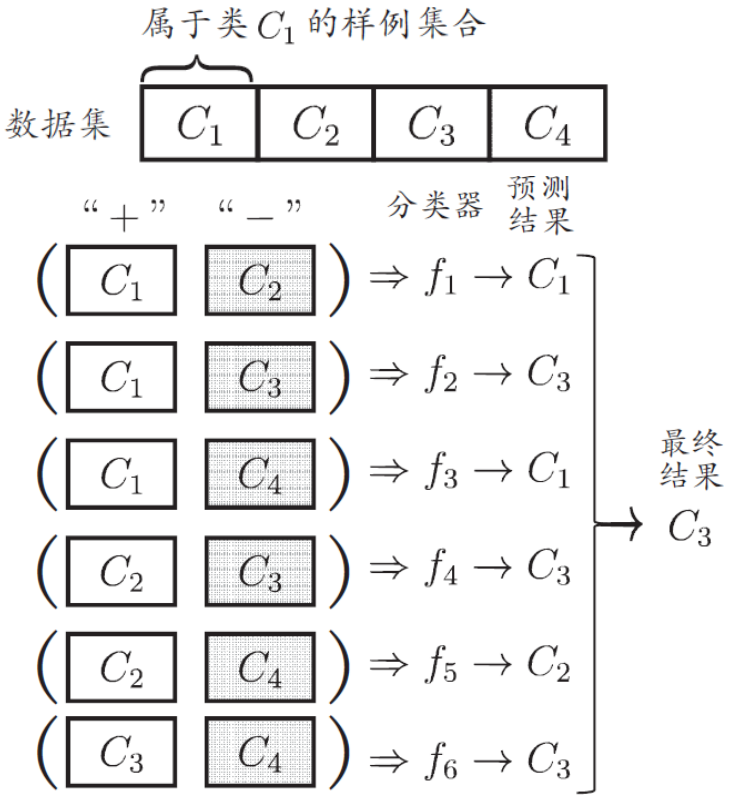
数据：$D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，$y_i\in\{C_i,\cdots,C_N\}$

拆分阶段

- N个类别，两两配对（如C_i和C_j，将C_i作为正例C_j作为反例）
  - $N(N-1)/2$ 个二类任务
- 各个二类任务，学习分类器
  - $N(N-1)/2$ 个二类分类器

测试阶段

- 新样本提交给所有分类器预测
  - $N(N-1)/2$个分类结果
- 投票产生最终分类结果
  - 被预测最多的类别为最终类别



- **拆分策略** 一对多（一对其余）

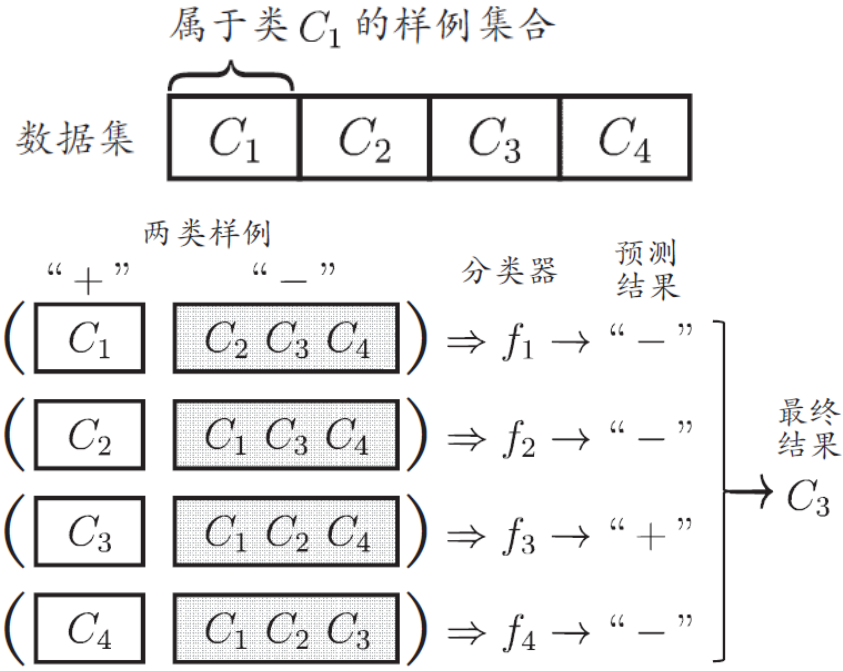
数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$，$y_i \in \{C_i, \cdots, C_N\}$

拆分阶段

- N个类别，某一类作为正例，所有其他类作为反例
  - N 个二类任务
- 各个二类任务，学习分类器
  - N 个二类分类器

测试阶段

- 新样本提交给所有分类器预测
  - N个分类结果
- 比较各分类器预测置信度或投票
  - 置信度最大类别作为最终类别



多分类学习——两种策略比较

一对一

- 训练 $N(N-1)/2$ 个分类器，存储开销大
- 训练只用两个类的样例，训练时间短
- 测试时间长

一对其余

- 训练N个分类器，存储开销小
- 训练用到全部训练样例，训练时间长
- 测试时间短

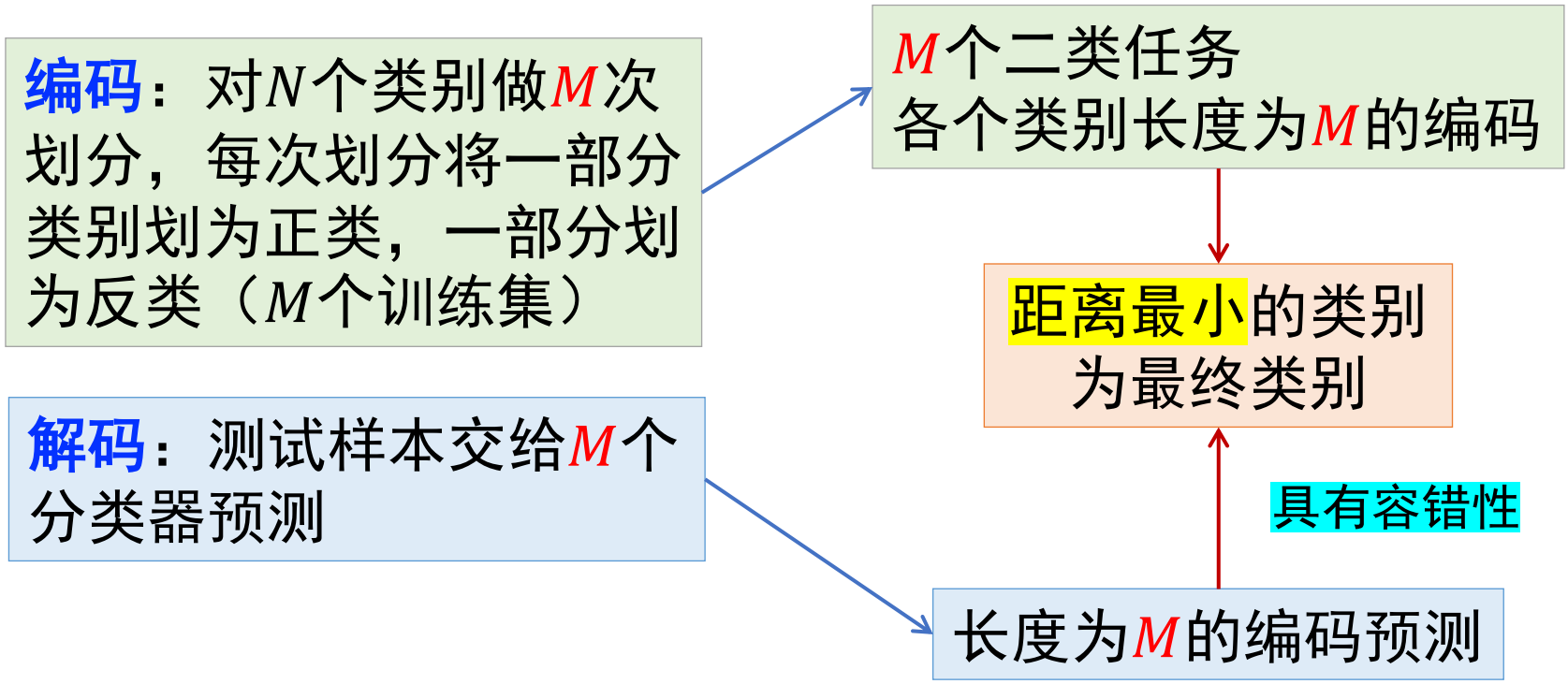
预测性能取决于具体数据分布，
多数情况下两者差不多

多对多（Many vs Many, MvM）

- 若干类作为正类，若干类作为反类

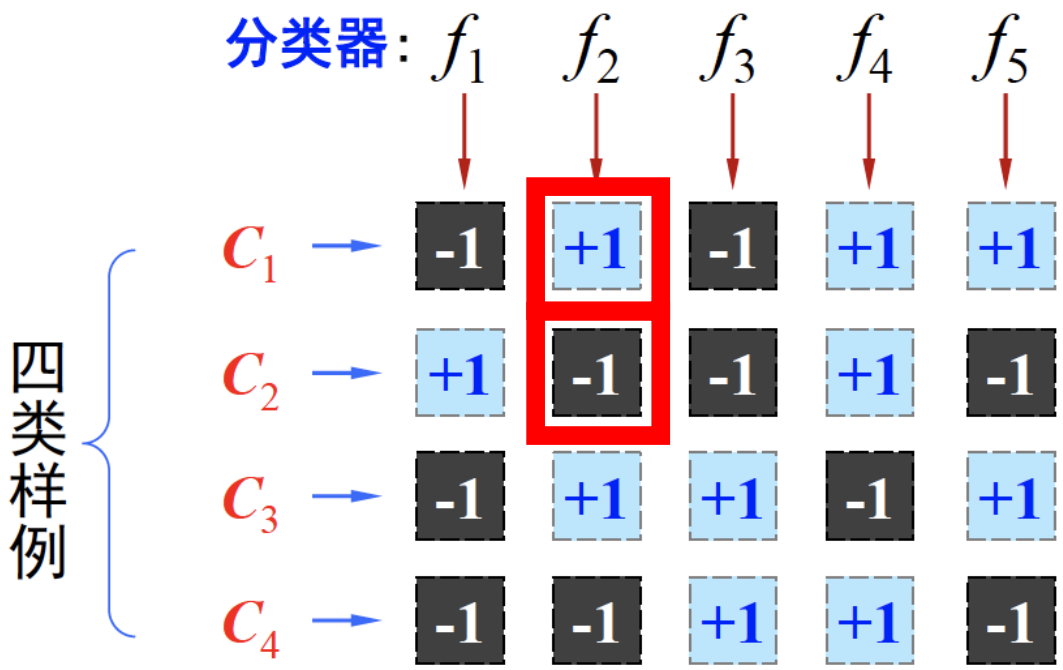
纠错输出码（Error Correcting Output Code, ECOC）

将编码的思想引入类别拆分，并尽可能在解码过程中具有容错性.



多分类学习——多对多

- **ECOC二元码示意图：** 将每个类别分别指定为正类和反类



海明距离	欧氏距离
3	$2\sqrt{3}$
4	4
1	2
2	$2\sqrt{2}$

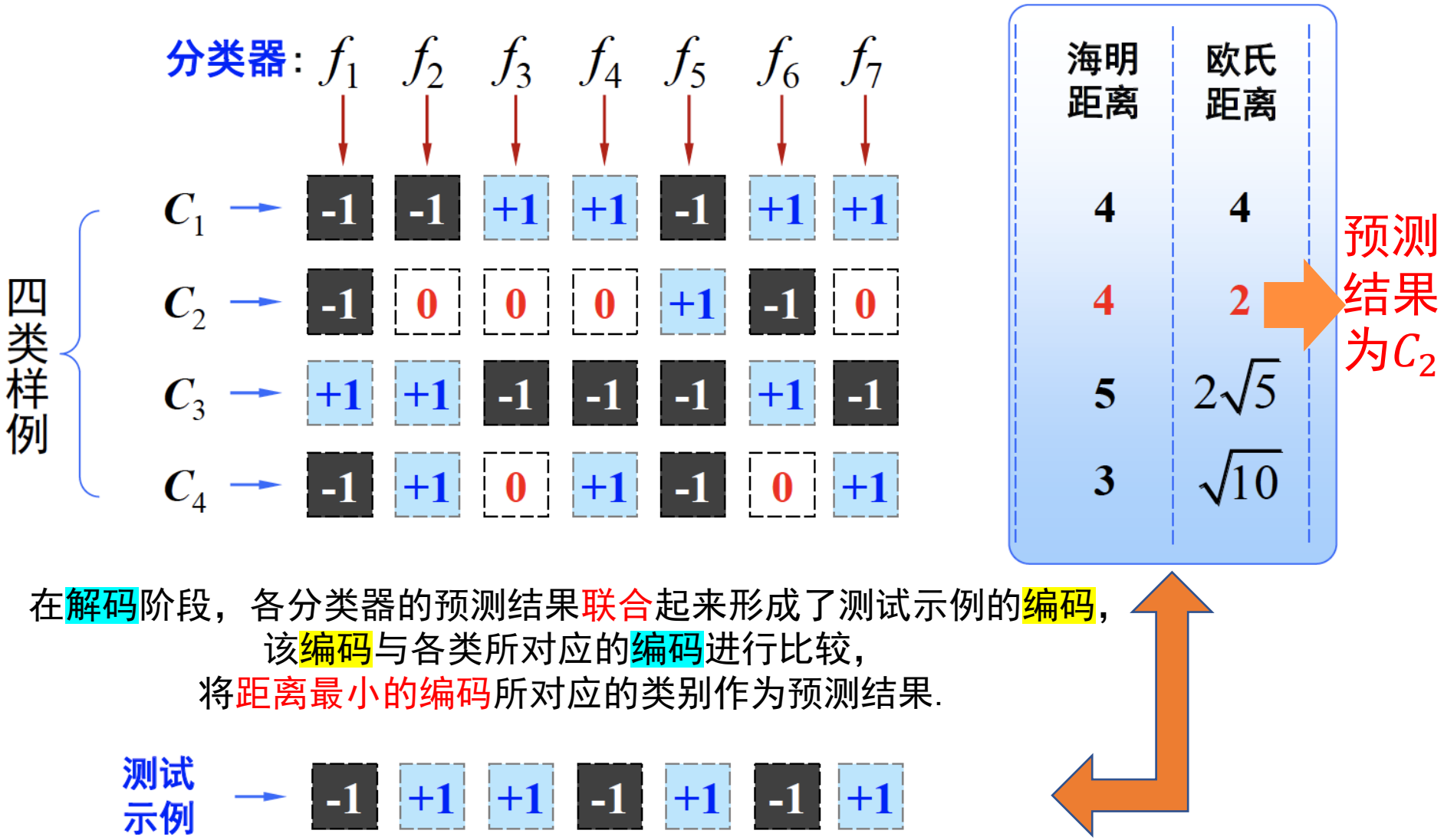
预测结果为C_3

在解码阶段，各分类器的预测结果联合起来形成了测试示例的编码，
该编码与各类所对应的编码进行比较，
将距离最小的编码所对应的类别作为预测结果.

测试示例 → -1 -1 +1 -1 +1

多分类学习——多对多

- ECOC三元码示意图 (0:表示不考虑此类):



在解码阶段，各分类器的预测结果联合起来形成了测试示例的编码，
该编码与各类所对应的编码进行比较，
将距离最小的编码所对应的类别作为预测结果.

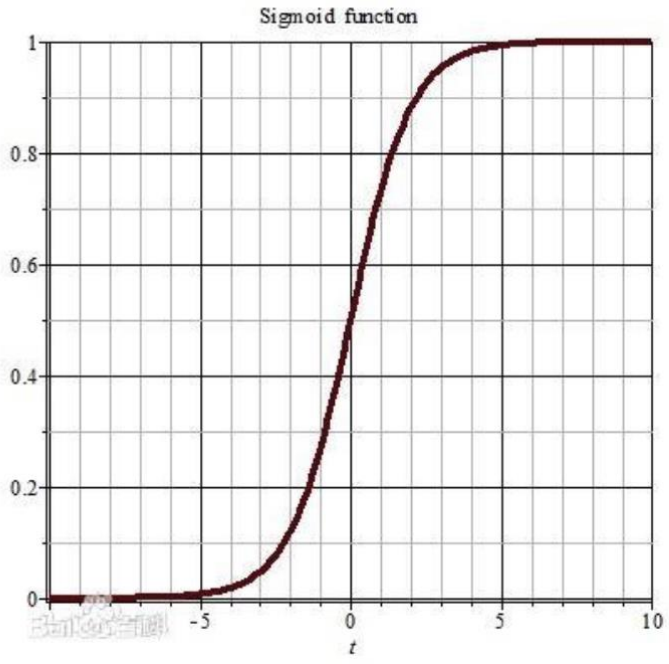
中山大學
SUN YAT-SEN UNIVERSITY

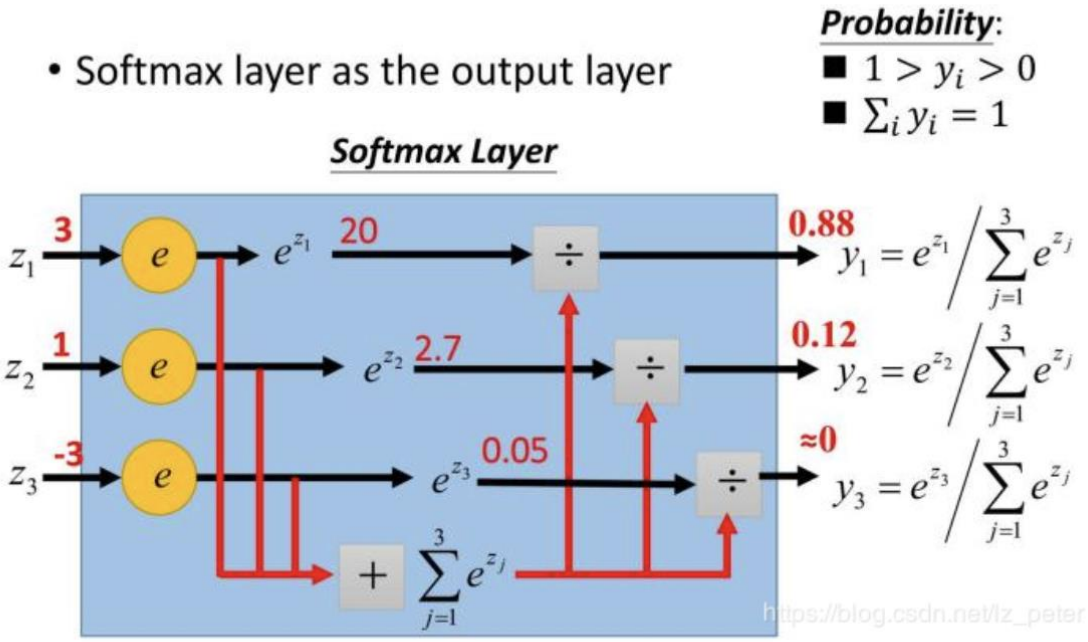
第3章 线性模型

1. 基本形式
2. 单变量线性回归
3. 多元线性回归
4. 广义线性模型
5. 对数几率回归
6. 多分类学习
7. Softmax回归

Softmax函数

- softmax函数，又称归一化指数函数。
- 它是二分类函数sigmoid在多分类上的推广，目的是将多分类的结果以概率的形式展现出来。





中山大學
SUN YAT-SEN UNIVERSITY

第3章 线性模型

1. 基本形式

2. 单变量线性回归

（数据、模型、策略、算法）

3. 多元线性回归

（数据、模型、策略、算法）

4. 广义线性模型

（数据、模型、策略、算法）

5. 对数几率回归

（数据、模型、策略、算法）

6. Softmax回归方法

（交叉熵）

中山大學
SUN YAT-SEN UNIVERSITY

第4章 决策树

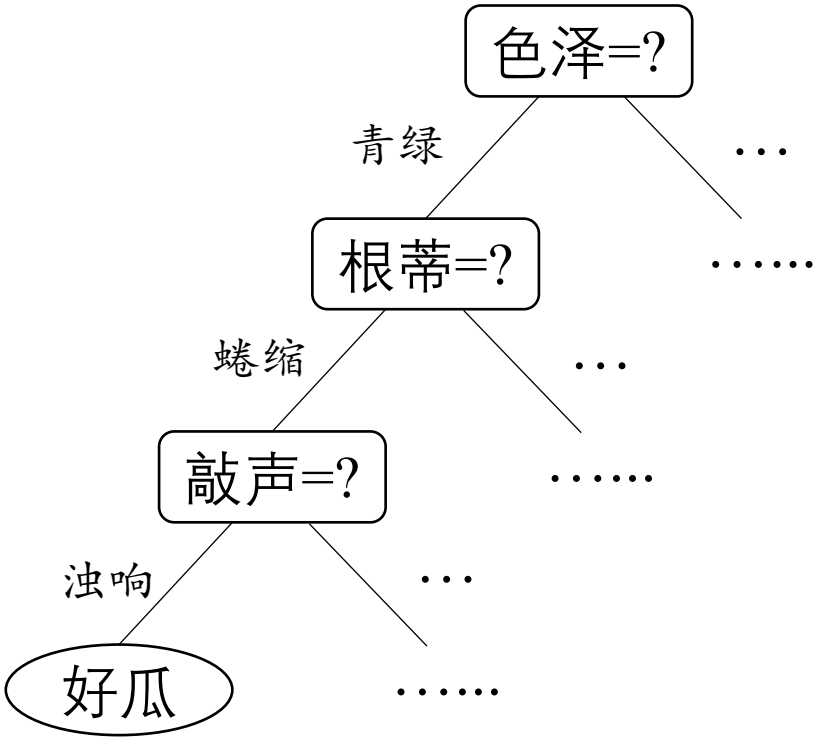
1. 基本流程
2. 划分选择
3. 剪枝处理
4. 缺失值处理

沈颖 副教授

sheny76@mail.sysu.edu.cn

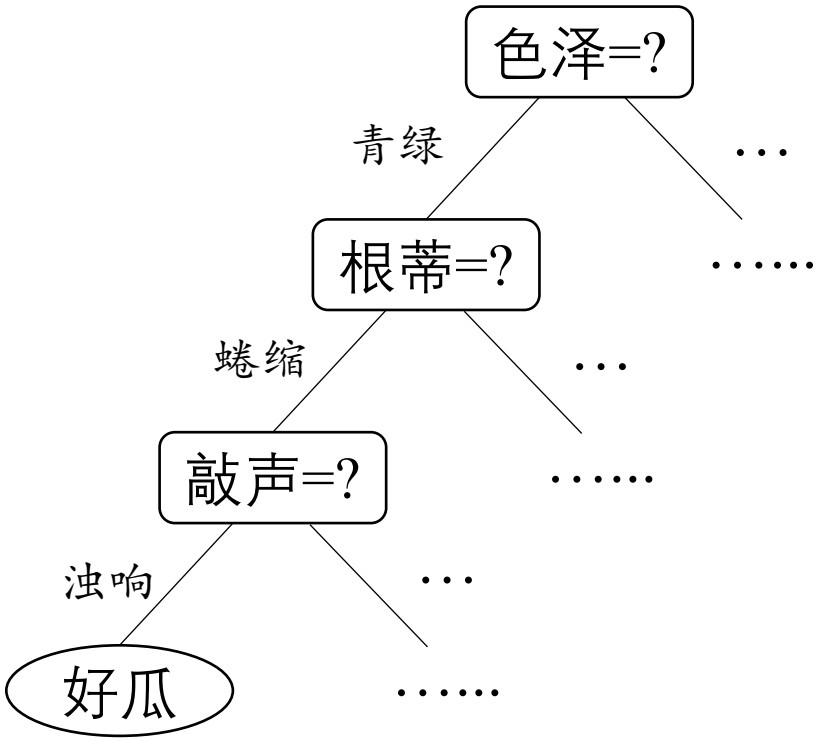
- 决策树（decision tree）

  决策树是一种基本的分类与回归方法，决策树模型呈树形结构，主要由节点（根节点、内部节点和叶节点）和有向边组成。



决策树（decision tree）

决策树是一种基本的分类与回归方法，决策树模型呈树形结构，主要由节点（根节点、内部节点和叶节点）和有向边组成。



- 根节点（一个）
  包含样本全集
- 分支节点（若干）
  对应于一个属性的测试
- 叶节点（若干）
  对应于决策结果。
  即当前属性集为空，或是所有样本在所有属性上取值相同，无法划分;

决策树学习（分类问题）

基本思想

数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$

其中$\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{id}\} \in \mathbb{R}^d, y_i \in \{1,2,\cdots,K\}$为类标记

Features				Classification
Outlook 天气预告	Temp 温度	Humidity 湿度	Wind 风	Class Play Yes or No 赛事能否进行
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast 多云	Hot	High	Weak	Yes
Rain	Midd	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Midd	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Midd	Normal	Weak	Yes
Sunny	Midd	Normal	Strong	Yes

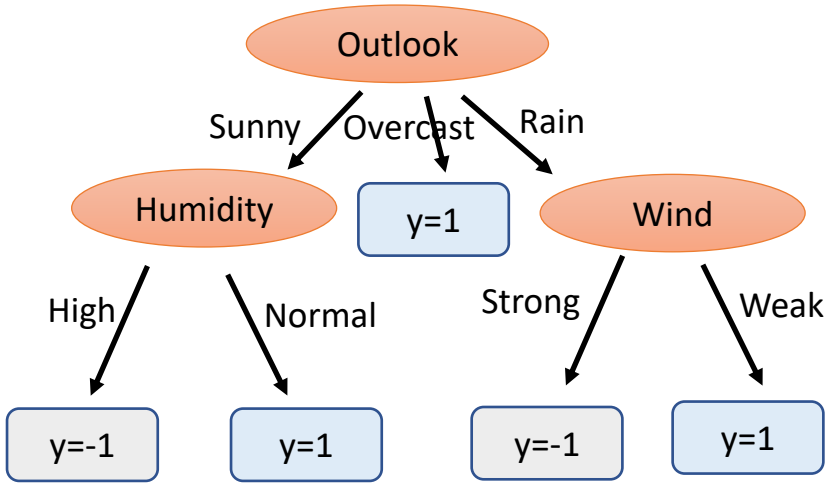
决策树学习（分类问题）

基本思想

数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$

其中$\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{id}\} \in \mathbb{R}^d, y_i \in \{1,2,\cdots,K\}$为类标记

模型： 分类决策树模型（一种由结点和有向边组成的用于描述对实例进行分类的树形结构）



决策树学习（分类问题）

- 基本思想

数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$

其中$\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{id}\} \in \mathbb{R}^d, y_i \in \{1,2, \cdots, K\}$为类标记

模型：分类决策树模型（一种由结点和有向边组成的用于描述对实例进行分类的树形结构）

策略：找到一个与训练数据矛盾较小同时泛化能力较强的决策树（可采用正则化的对数似然函数来作为损失函数）

决策树学习（分类问题）

基本思想

数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$

其中$\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{id}\} \in \mathbb{R}^d, y_i \in \{1,2,\cdots,K\}$为类标记

模型：分类决策树模型（一种由结点和有向边组成的用于描述对实例进行分类的树形结构）

策略：找到一个与训练数据矛盾较小同时泛化能力较强的决策树（可采用正则化的对数似然函数来作为损失函数）

算法：启发式方法（ID3、C4.5、CART等）

- 通常是一个递归地选择最优特征，并根据该特征对训练数据进行分割，使得对各个子数据集有一个最好的分类的过程。

决策树学习（分类问题）

基本思想

算法：启发式方法（ID3、C4.5、CART等）

- 开始，构建根节点，将所有训练数据都放在根节点。
- 同时，选择一个最优属性，按照这一属性将训练数据集分割成子集，使得各个子集有一个当前条件下最好的分类。
  - 如果，这些子集已经能够被基本正确分类，那么构建叶节点，并将这些子集分到所对应的叶节点中去；
  - 如果，还有子集不能被正确分类，那么构建中间节点，对这些子集选择新的最优属性，继续对其进行分割。
- 然后，如此递归地进行下去，直至所有训练数据子集被基本正确分类，或者没有合适的属性为止。
- 最后，构建叶节点，每个子集都被分到叶节点得到各自的类。

中山大學
SUN YAT-SEN UNIVERSITY

第4章 决策树

1. 基本流程
2. 划分选择
3. 剪枝处理
4. 缺失值处理

西瓜书：第4章4.1和4.2全部

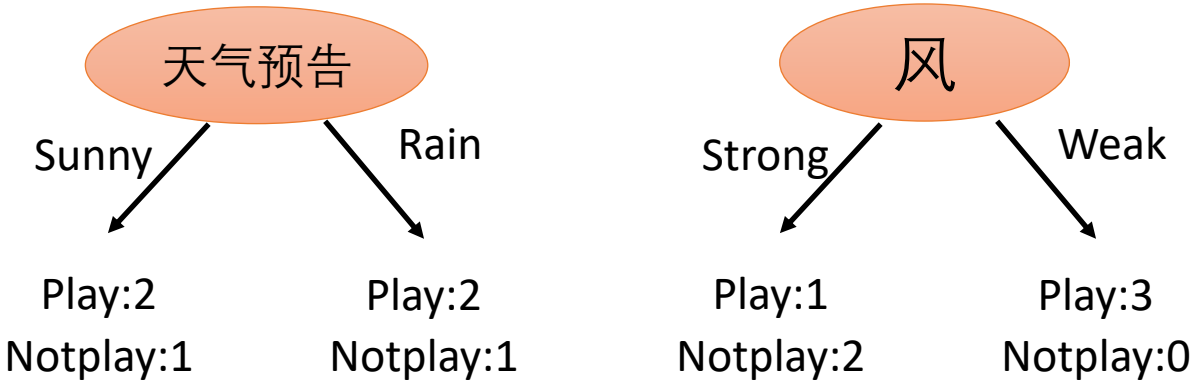
划分选择

决策树学习的关键在于如何选择最优划分属性。

一般而言，随着划分过程不断进行，我们希望决策树的分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”(purity)越来越高

天气预告	风	湿度	赛事进行?
Sunny	Weak	High	1
Sunny	Strong	High	-1
Sunny	Weak	Normal	1
Rain	Strong	Normal	1
Rain	Weak	High	1
Rain	Strong	High	-1

卡方检验—CHAID决策树



划分选择（特征选择）

- 如何选择最优划分属性

  一般而言，随着划分过程不断进行，我们希望决策树的分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”（purity）越来越高

- 经典的属性划分方法
  - 信息增益
  - 增益率
  - 基尼指数

划分选择——信息增益

- 信息增益（ID3决策树）

离散属性a有V个可能的取值$\{a^1, a^2, \cdots, a^V\}$，用$a$来进行划分，则会产生$V$个分支结点，其中第$v$个分支结点包含了$D$中所有在属性$a$上取值为$a^v$的样本，记为$D^v$。则可计算出用属性$a$对样本集$D$进行划分所获得的“信息增益”：

$$\mathrm{Gain}(D,a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v)$$

划分前的信息熵　　划分后的信息熵

第v个分支的权重，样本越多越重要

弱点：对可取值数目较多的属性有所偏好

划分选择——增益率

- 增益率（C4.5决策树）

离散属性a有V个可能的取值$\{a^1, a^2, \cdots, a^V\}$，用$a$来进行划分，则会产生$V$个分支结点，其中第$v$个分支结点包含了$D$中所有在属性$a$上取值为$a^v$的样本，记为$D^v$。则可计算出用属性$a$对样本集$D$进行划分所获得的“增益率”：

$$\text{Gain\_ratio}(D,a) = \frac{\text{Gain}(D,a)}{\text{IV}(a)}$$

$$\text{IV}(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

可能存在的问题：对可取值数目较少的属性有所偏好

划分选择——增益率

- 信息增益（ID3决策树）：对可取值数目较多的属性有所偏好
- 增益率（C4.5决策树）：对可取值数目较少的属性有所偏好

先从候选划分属性中，找出**信息增益**高于平均水平的，

再从中选取**增益率**高的

决策树学习（分类问题）

基本思想

数据：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$

其中$\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{id}\} \in \mathbb{R}^d, y_i \in \{1,2,\cdots,K\}$为类标记

模型：分类决策树模型（一种由结点和有向边组成的用于描述对实例进行分类的树形结构）

策略：找到一个与训练数据矛盾较小同时泛化能力较强的决策树（也可采用正则化的对数似然函数来作为损失函数）

算法：启发式方法（ID3、C4.5、CART等）

- ID3决策树学习算法[Quinlan, 1986]以信息增益为准则来选择划分属性
- C4.5决策树学习算法[Quinlan, 1993] 先从候选划分属性中找出信息增益高于平均水平的属性，再从中选取增益率最高的
- CART决策树学习算法[Breiman et al., 1984]采用基尼指数来选择划分属性

决策树学习（分类问题）

基本思想

算法：启发式方法（ID3、C4.5、CART等）

- 开始，构建根节点，将所有训练数据都放在根节点。
- 同时，选择一个最优属性，按照这一属性将训练数据集分割成子集，使得各个子集有一个当前条件下最好的分类。
  - 如果，这些子集已经能够被基本正确分类，那么构建叶节点，并将这些子集分到所对应的叶节点中去；
  - 如果，还有子集不能被正确分类，那么构建中间节点，对这些子集选择新的最优属性，继续对其进行分割。
- 然后，如此递归地进行下去，直至所有训练数据子集被基本正确分类，或者没有合适的属性为止。
- 最后，构建叶节点，每个子集都被分到叶节点得到各自的类。

中山大學
SUN YAT-SEN UNIVERSITY

第5章 神经网络

1. 神经元模型
2. 感知机*与多层网络
3. 误差逆传播算法
4. 全局最小和局部最小
5. 深度学习

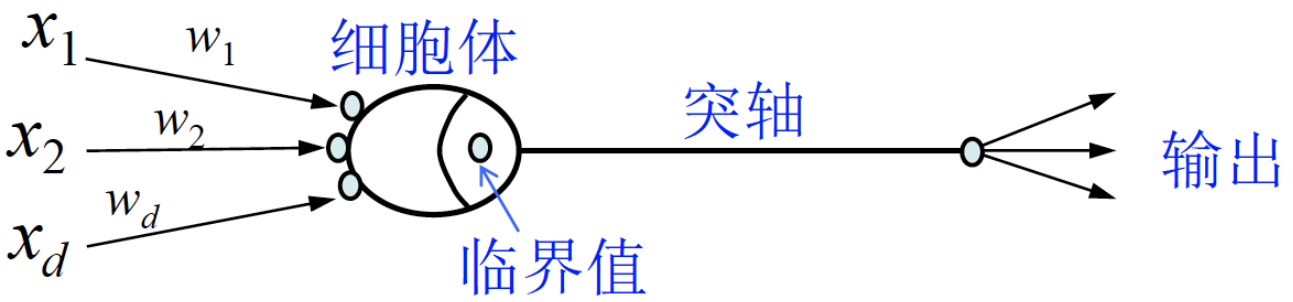
沈颖 副教授

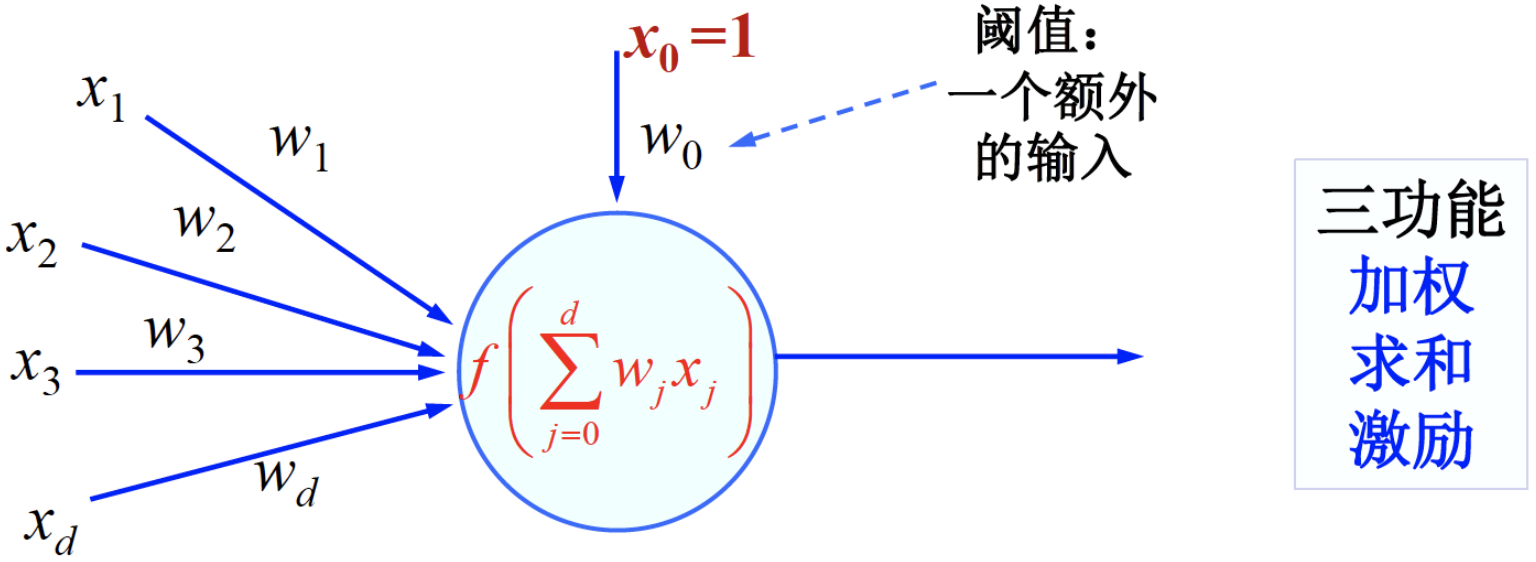
sheny76@mail.sysu.edu.cn

*《统计学习方法》第2章除2.3.2和2.3.3以外

神经元模型

- 处理单元的基本结构和功能





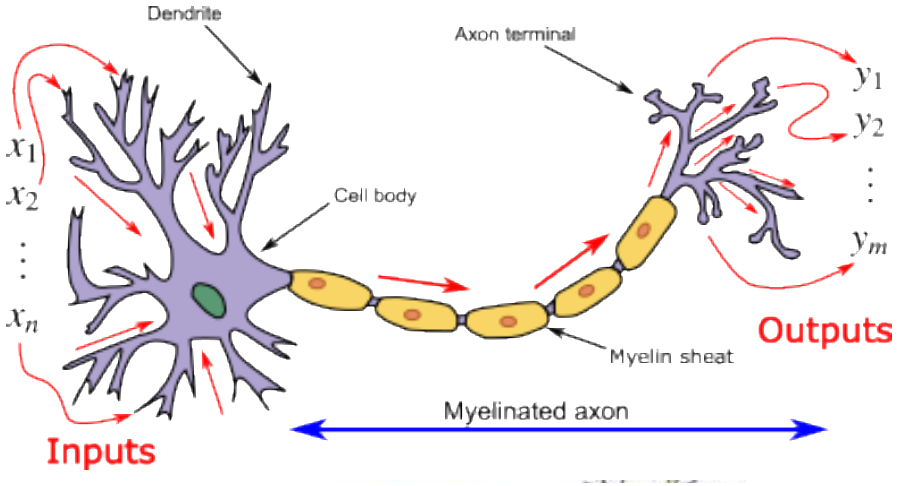
x: 收到信号　**w**: 信号的权重　$f(.)$: 激励函数

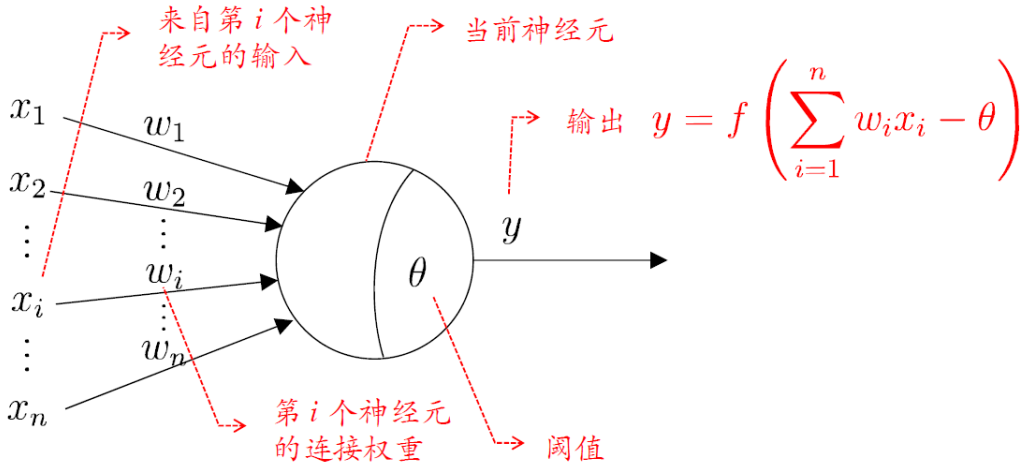
神经元模型

神经元（neuron）模型

M-P神经元模型（ McCulloch and Pitts, 1943 ）

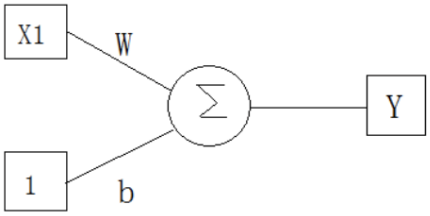
- **输入**：来自其他n个神经元传递过来的输入信号
- **处理**：输入信号通过带权重的连接进行传递, 神经元接受到总输入值将其与神经元的阈值进行比较
- **输出**：通过激活函数的处理以得到输出





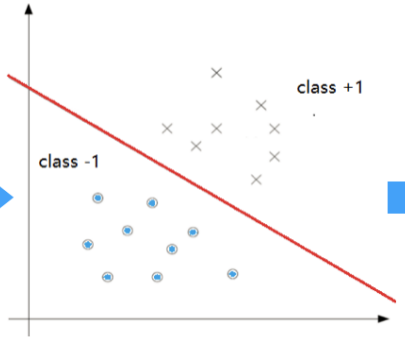
神经元（neuron）模型

激活函数（activation function ）

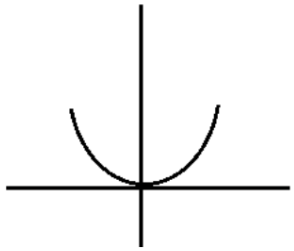


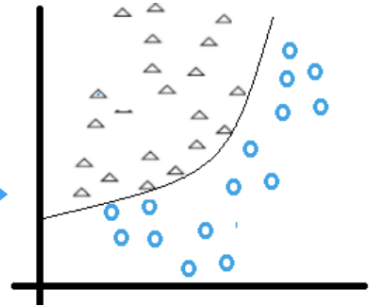
普通的神经网络

$f(x;\theta) = Wx + b$



无论怎么移动和反转，都无法解决非线性可分问题





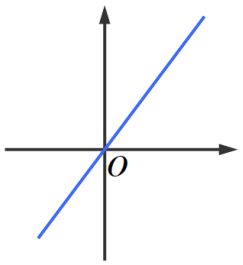
变成最简单的非线性神经网络

$f(x;\theta) = W\varphi(x) + b$

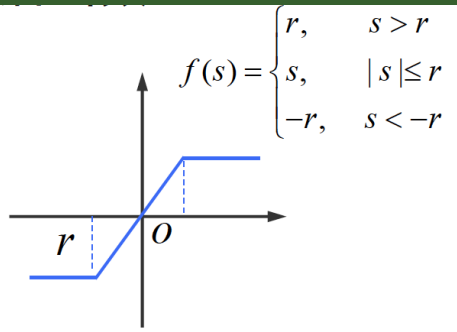
激活函数：

- 是神经网络逼近复杂函数的关键
- 神经网络的输出压缩至特定边界的关键

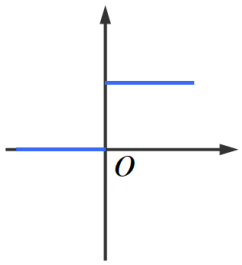
转移函数-激活函数



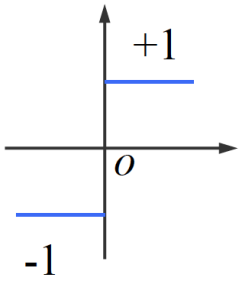
线性函数



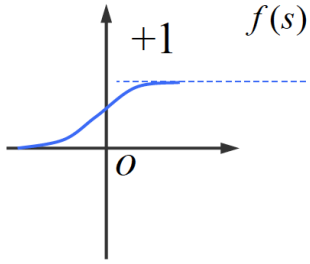
斜坡函数



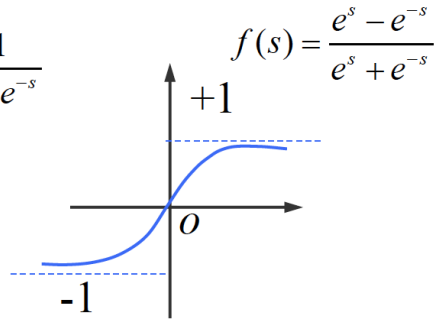
阶跃函数



符号函数



Sigmoid函数



双曲正切函数(tanh)

每个激活函数都要考虑输入输出以及数据变化

激活函数可

1）提高模型鲁棒性和非线性表达能力，

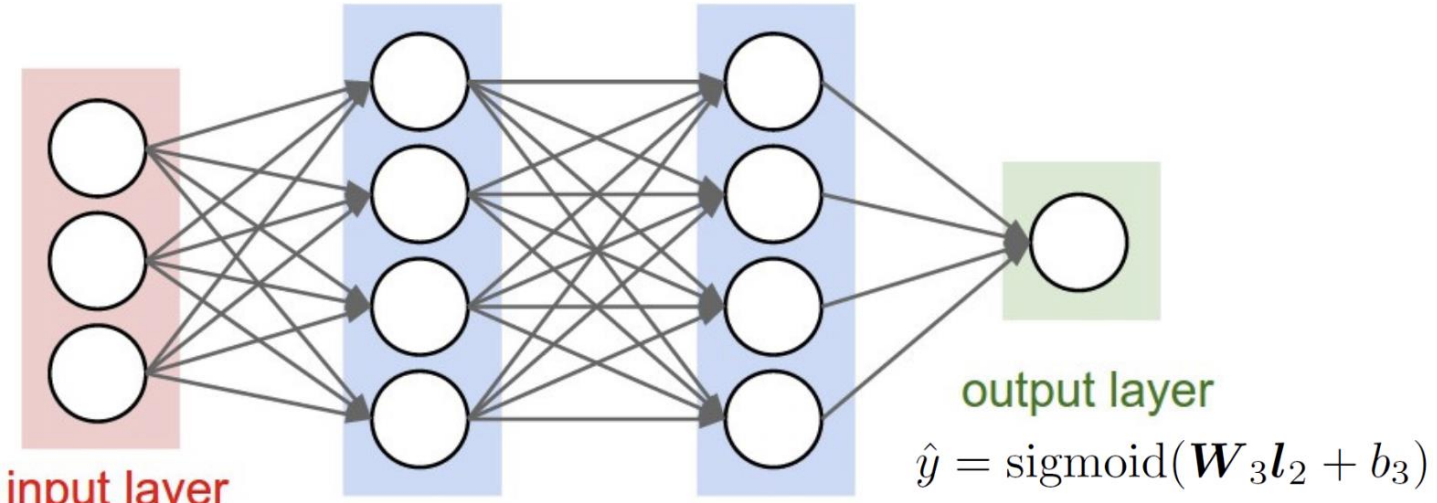
2）缓解梯度消失问题，

3）将特征图映射到新的特征空间从而更有利于训练，

4）加速模型收敛。

- 神经网络模型

  将若干神经元按一定的层次结构连接起来，可得到神经网络。从计算机科学的角度看，可将神经网络视为包含了若干参数的数学模型，这个模型是由若干个函数，例如$y_j = f\left(\sum_i \omega_i x_i - \theta_j\right)$相互(嵌套)代入而得



中山大學
SUN YAT-SEN UNIVERSITY

第5章 神经网络

1. 神经元模型
2. 感知机*与多层网络
3. 误差逆传播算法
4. 全局最小和局部最小
5. 深度学习

*《统计学习方法》第2章除2.3.2和2.3.3以外

感知机（逻辑运算问题）

感知机由两层神经元组成，

输入层接受外界输入信号传递给输出层，

输出层是M-P神经元（阈值逻辑单元）

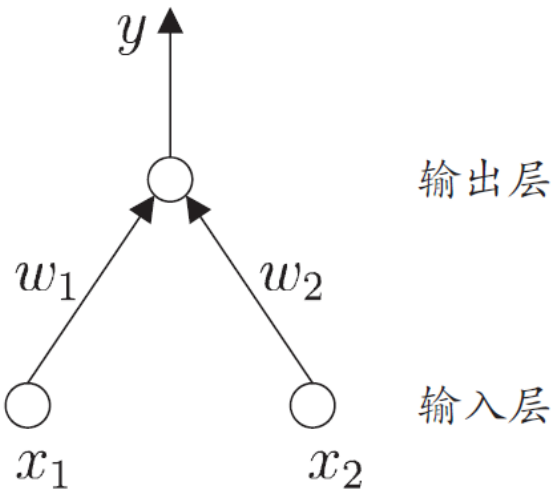


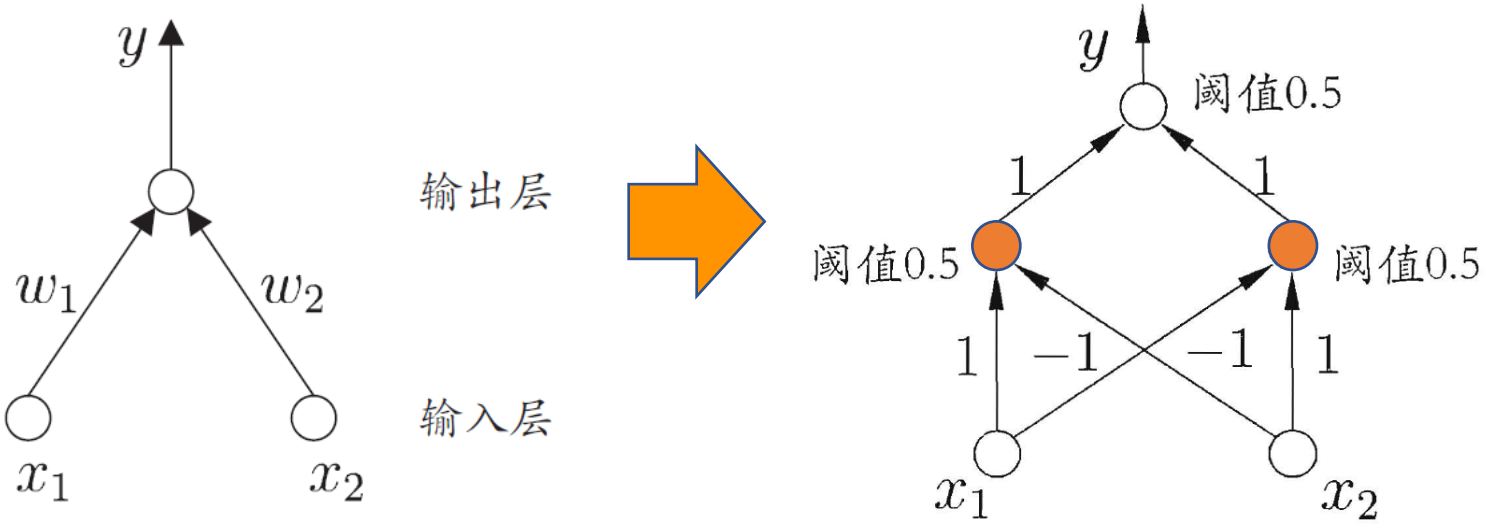
图 5.3 两个输入神经元的感知机网络结构示意图

感知机能够容易地实现逻辑与and、或or、非not运算

感知机

要解决非线性可分问题，需考虑多层功能神经元：

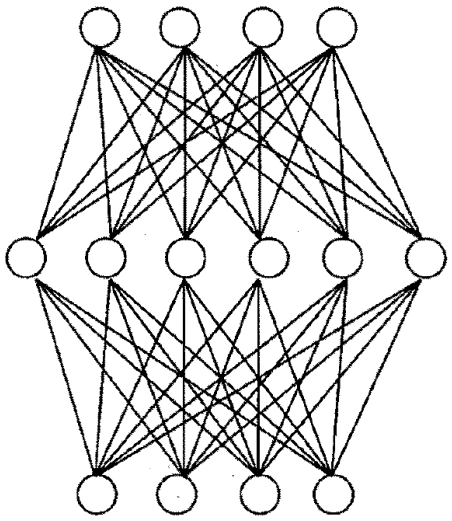
- 输出层与输入层之间的一层神经元，被称为隐层或隐含层（hidden layer）
- 隐含层和输出层神经元都是拥有激活函数的功能神经元



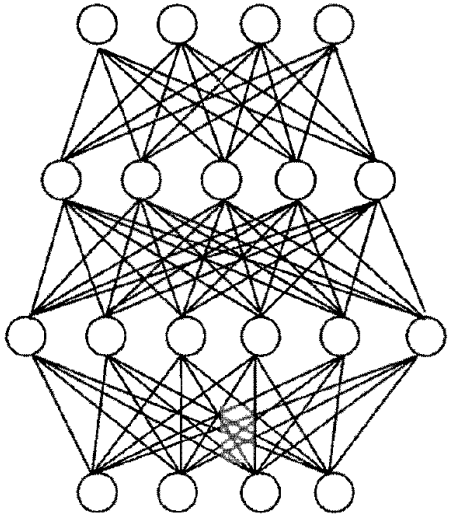
感知机

更常见的，多层前馈神经网络：

- 神经网络是一种层级结构，
- 每层神经元与下一层神经元全互连，
- 神经元之间不存在同层连接，也不存在跨层连接。



(a) 单隐层前馈网络



(b) 双隐层前馈网络

中山大學
SUN YAT-SEN UNIVERSITY

第5章 神经网络

1. 神经元模型
2. 感知机*与多层网络
3. 误差逆传播算法
4. 全局最小和局部最小
5. 深度学习

沈颖 副教授

sheny76@mail.sysu.edu.cn

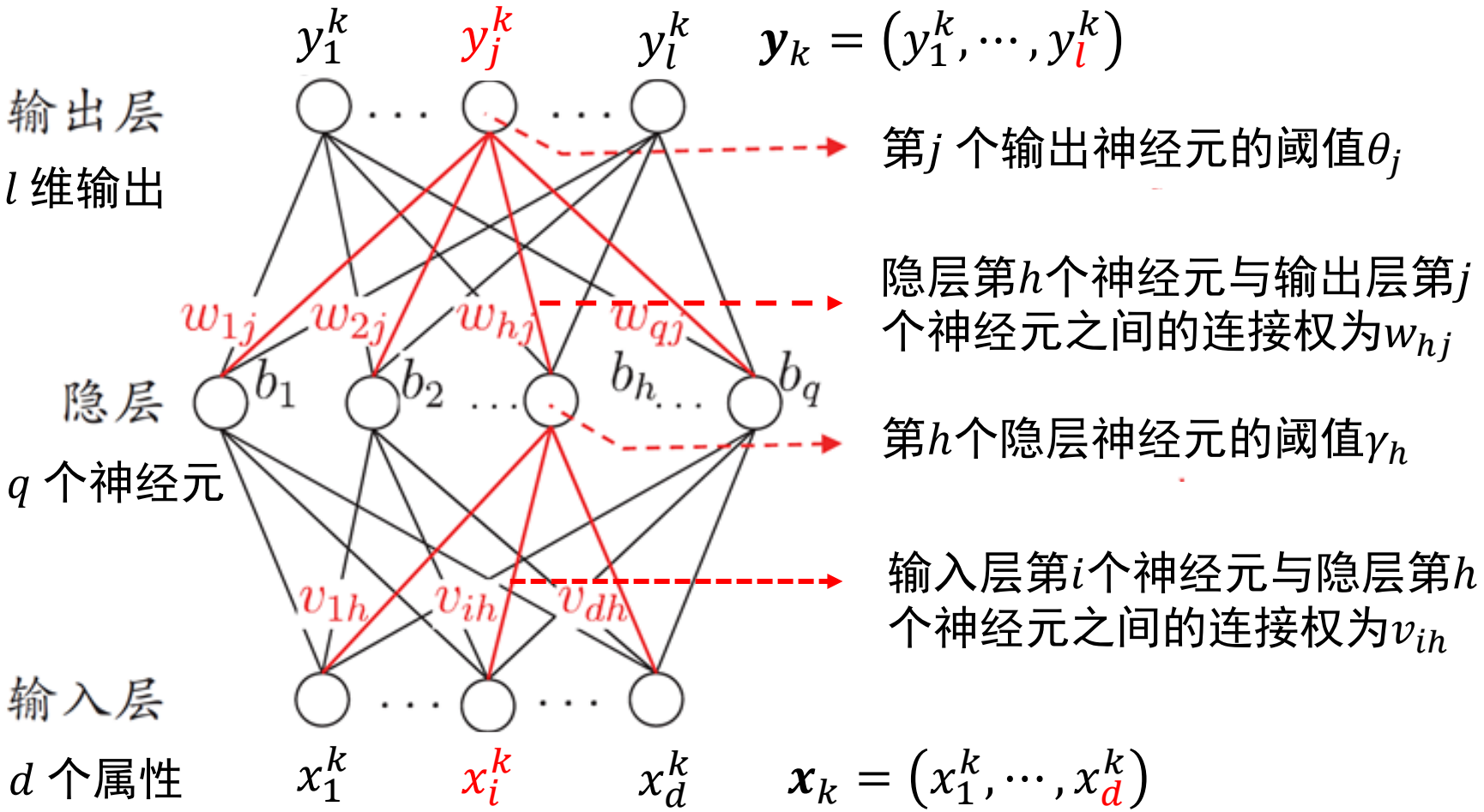
*《统计学习方法》第2章除2.3.2和2.3.3以外

误差逆传播算法

◘ 多层前馈神经网络（multi-layer feedforward neural networks）

数据：$D=\{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2),\cdots,(\boldsymbol{x}_m,\boldsymbol{y}_m)\}$，$\boldsymbol{x}_i\in\mathbb{R}^d,\boldsymbol{y}_i\in\mathbb{R}^l$

模型：所有神经元的激活函数都为sigmoid函数

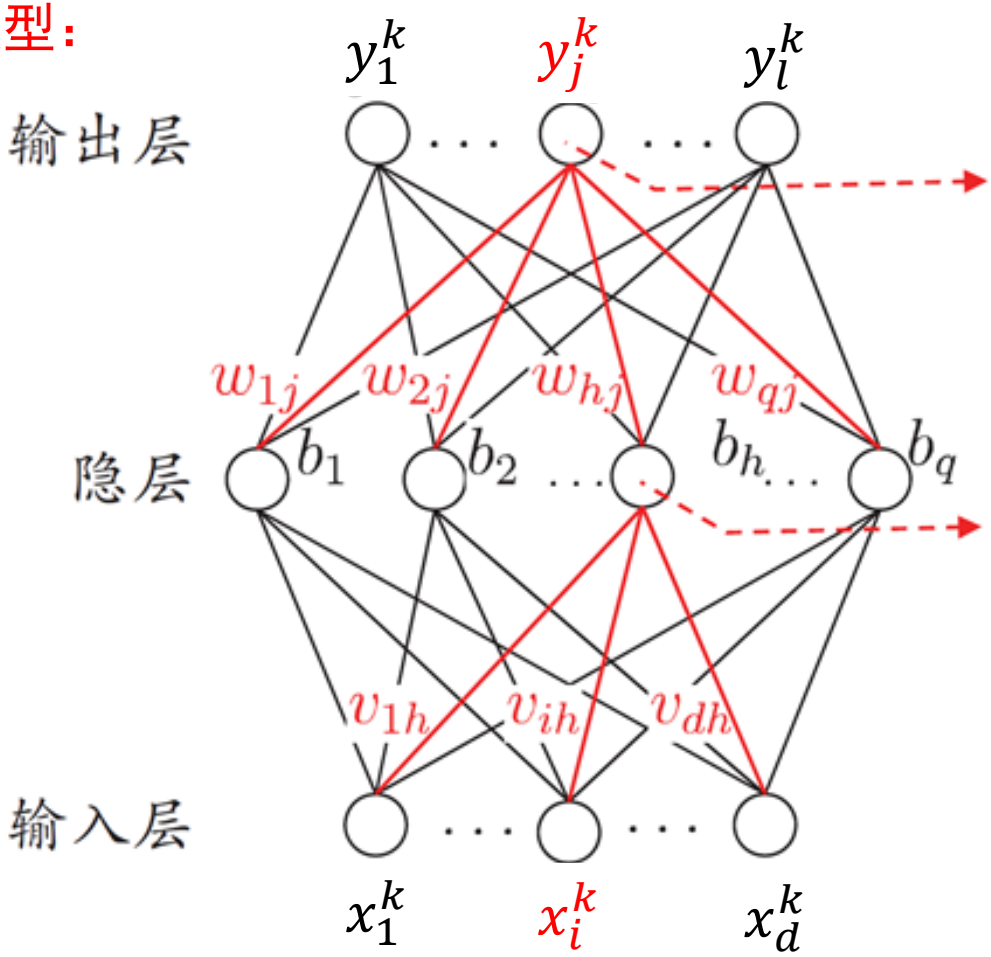


误差逆传播算法

◘ 多层前馈神经网络（multi-layer feedforward neural networks）

数据：$D=\{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2),\cdots,(\boldsymbol{x}_m,\boldsymbol{y}_m)\}$，$\boldsymbol{x}_i\in\mathbb{R}^d,\boldsymbol{y}_i\in\mathbb{R}^l$

模型：



第j 个输出神经元的输入输出

$$\beta_j=\sum_{h=1}^{q}\omega_{hj}b_h$$

$$y_j^k=f_{sigmoid}(\beta_j-\theta_j)$$

第h个隐层神经元的输入输出

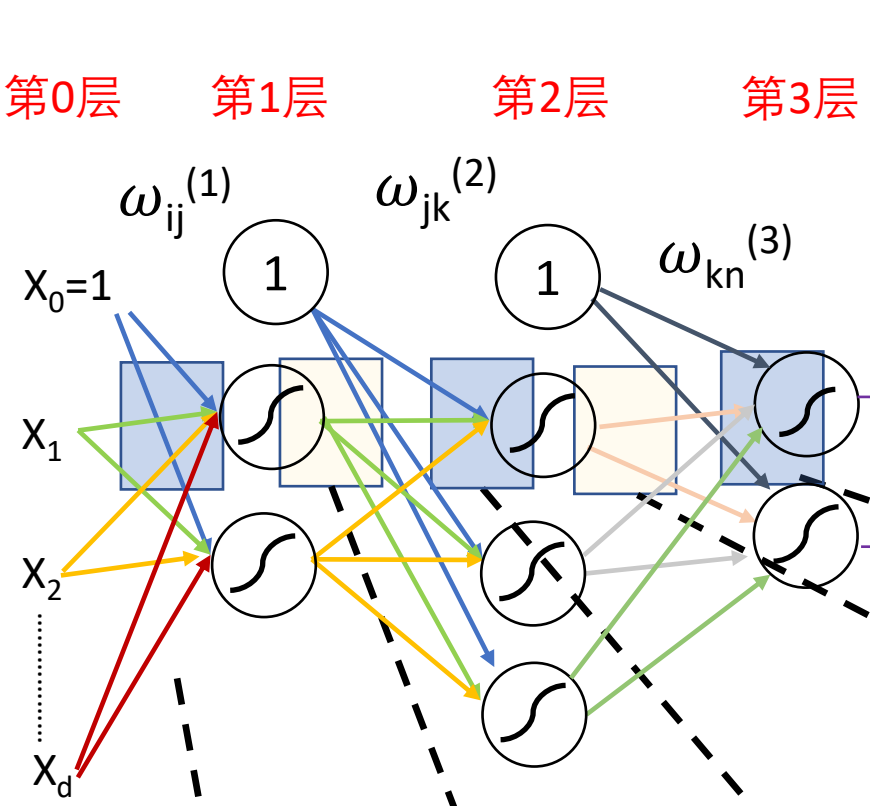
$$\alpha_h=\sum_{i=1}^{d}v_{ih}x_i^k$$

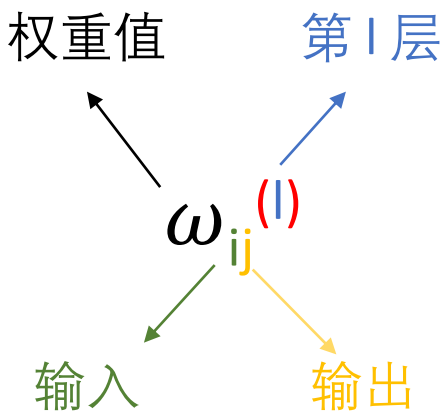
$$b_h=f_{sigmoid}(\alpha_h-\gamma_h)$$

所有神经元的激活函数都为sigmoid函数

多层神经网络

此过程为前馈计算



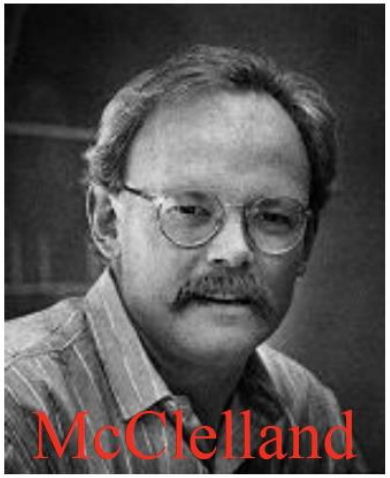


$$\hat{y}_n = g(z_n^{(3)})$$

$$z_n^{(3)} = \sum_{k=0}^{d2} \omega_{kn}^{(3)} x_k^{(2)}$$

$$z_j^{(1)} = \sum_{i=0}^{d0} \omega_{ij}^{(1)} x_i^{(0)} \qquad x_j^{(1)} = g(z_j^{(1)}) \qquad z_k^{(2)} = \sum_{j=0}^{d1} \omega_{jk}^{(2)} x_j^{(1)} \qquad x_k^{(2)} = g(z_k^{(2)})$$

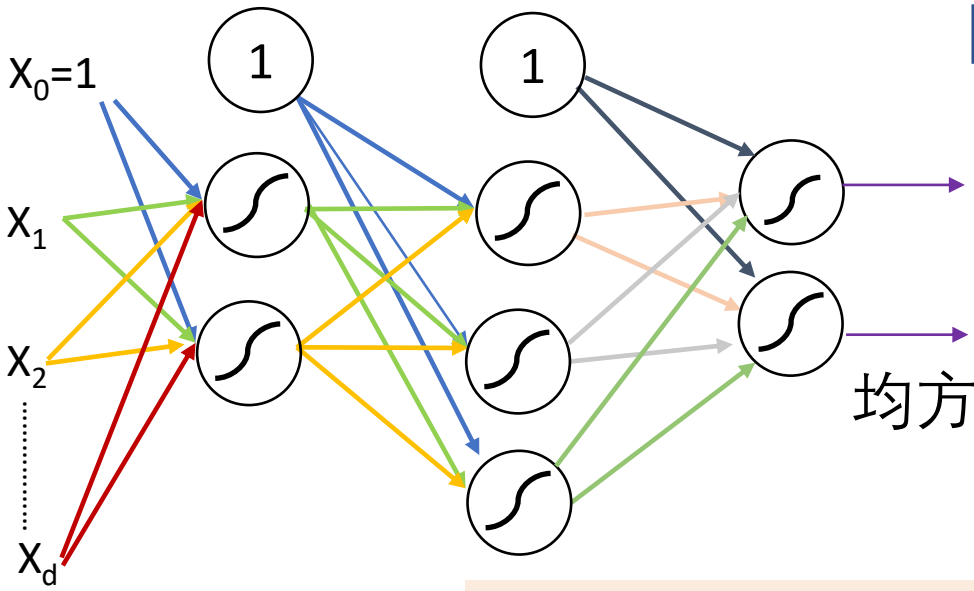
- D. Rumelhart，J. McClelland于1985年提出了误差反向传播(Back Propagation, BP)学习算法



- 基本原理
  - 利用输出后的误差来估计**输出层的前一层**的误差，再用这个误差估计更前一层的误差，如此一层一层地反传下去，从而获得所有其它各层的误差估计

误差反向传播算法

- Rumelhart and McClelland, 1986,《*Parallel Distributed Proceesing:Explorations in the Microstructures of Cognition*》



依此计算可的最后的输出

$$\hat{y}_n = g(z_n^{(3)})$$

均方误差 $E_m = \frac{1}{2}\sum_{n=1}^{d3}(\hat{y}_n{}^m - y_n{}^m)^2$

神经网络可以理解为通过样本数据逐渐学习一组合适的权重值，达到最佳判断效果（最小化误差）

任意参数的更新估计式

$$w \leftarrow w + \Delta w$$

误差逆传播算法

误差反向传播算法

- 误差反向传播训练算法
  - 属于监督学习算法，**通过调节各层的权重，使网络学会**由“输入-输出对”组成的训练组。
  - BP算法核心是梯度下降法。
  - 权重先从输出层开始修正，再依次修正各层权重
    - 首先修正：**“输出层至最后一个隐含层”**的连接权重
    - 再修正：**“最后一个隐含层至倒数第二个隐含层”**的连接权重，….
    - 最后修正：**“第一隐含层至输入层”**的连接权重。

学习的本质：对网络各连接权重作动态调整！

中山大學
SUN YAT-SEN UNIVERSITY

第5章 神经网络

1. 神经元模型
2. 感知机*与多层网络
3. 误差逆传播算法
4. 全局最小和局部最小
5. 深度学习

*《统计学习方法》第2章除2.3.2和2.3.3以外

全局最小和局部最小

- 若用 E 表示神经网络在训练集上的误差，则它显然是关于连接权w和阈值θ的函数.
- 此时，神经网络的训练过程可看作一个参数寻优过程，即在参数空间中，寻找一组最优参数使得E最小.
- 我们常会谈到两种“最优”：
  - "局部极小"（local minimum）
  - "全局最小"（global minimum）

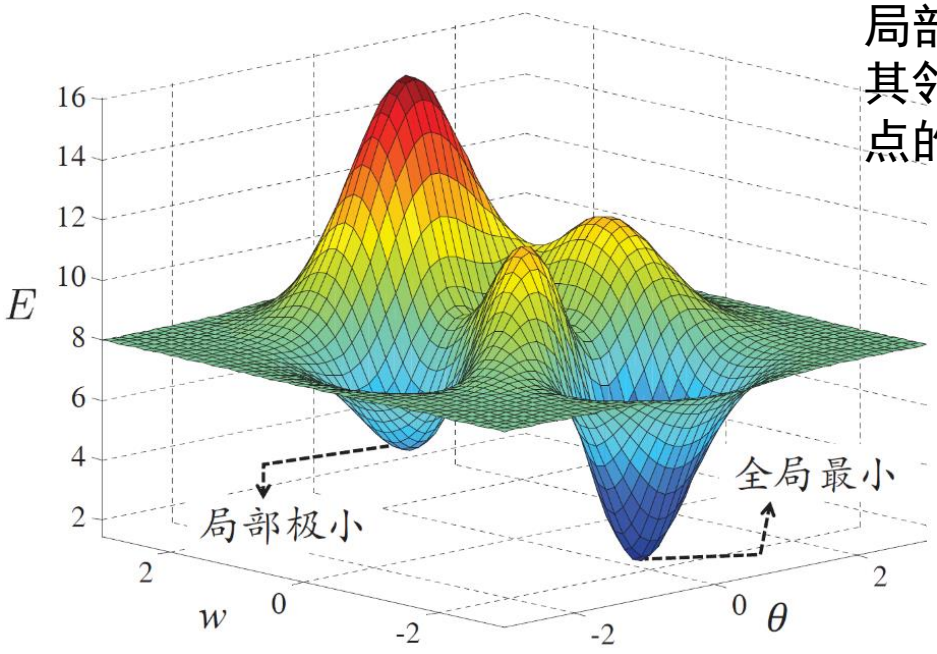
全局最小和局部最小

- 对$\boldsymbol{\omega}^*$和θ^*，若存在$\epsilon > 0$使得

$$\forall(\boldsymbol{\omega};\theta) \in \{\|(\boldsymbol{\omega};\theta) - (\boldsymbol{\omega}^*;\theta^*)\| \leq \epsilon\}$$

都有$E(\boldsymbol{\omega};\theta) \geq E(\boldsymbol{\omega}^*;\theta^*)$成立，则为局部极小解；

若对参数空间中的任意$(\boldsymbol{\omega};\theta)$，都有$E(\boldsymbol{\omega};\theta) \geq E(\boldsymbol{\omega}^*;\theta^*)$成立，则为全局最小解。



局部极小解是参数空间中的某个点，其邻域点的误差函数值均不小于该点的函数值；

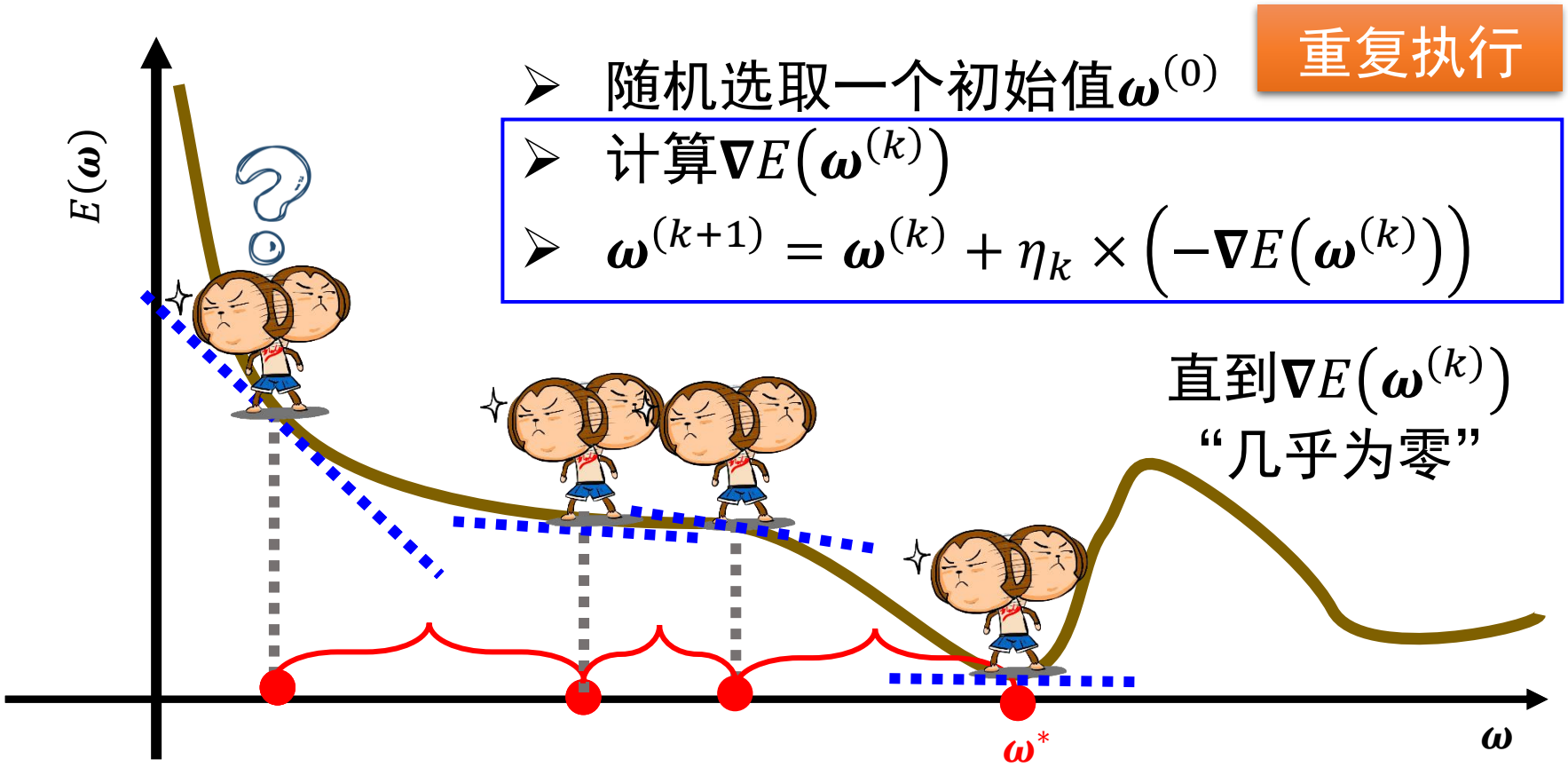
全局最小解则是指参数空间中所有点的误差函数值均不小于该点的误差函数值.

全局最小和局部最小

“跳出”局部最小的策略

基于**梯度**的搜索是使用最为广泛的参数寻优方法。

如果误差函数仅有**一个**局部极小，那么此时找到的局部极小就是**全局最小**；



全局最小和局部最小

“跳出”局部最小的策略

基于梯度的搜索是使用最为广泛的参数寻优方法。如果误差函数仅有一个局部极小，那么此时找到的局部极小就是全局最小；
然而，如果误差函数具有多个局部极小，则不能保证找到的解是全局最小。

在现实任务中，通常采用以下策略“跳出”局部极小，从而达到全局最小：

- 多组不同的初始参数优化神经网络，选取误差最小的解作为最终参数
- 模拟退火技术：每一步都以一定的概率接受比当前解更差的结果，从而有助于跳出局部极小
- 随机梯度下降：与标准梯度下降法精确计算梯度不同，随机梯度下降法在计算梯度时加入了随机因素

中山大學
SUN YAT-SEN UNIVERSITY

第5章 神经网络

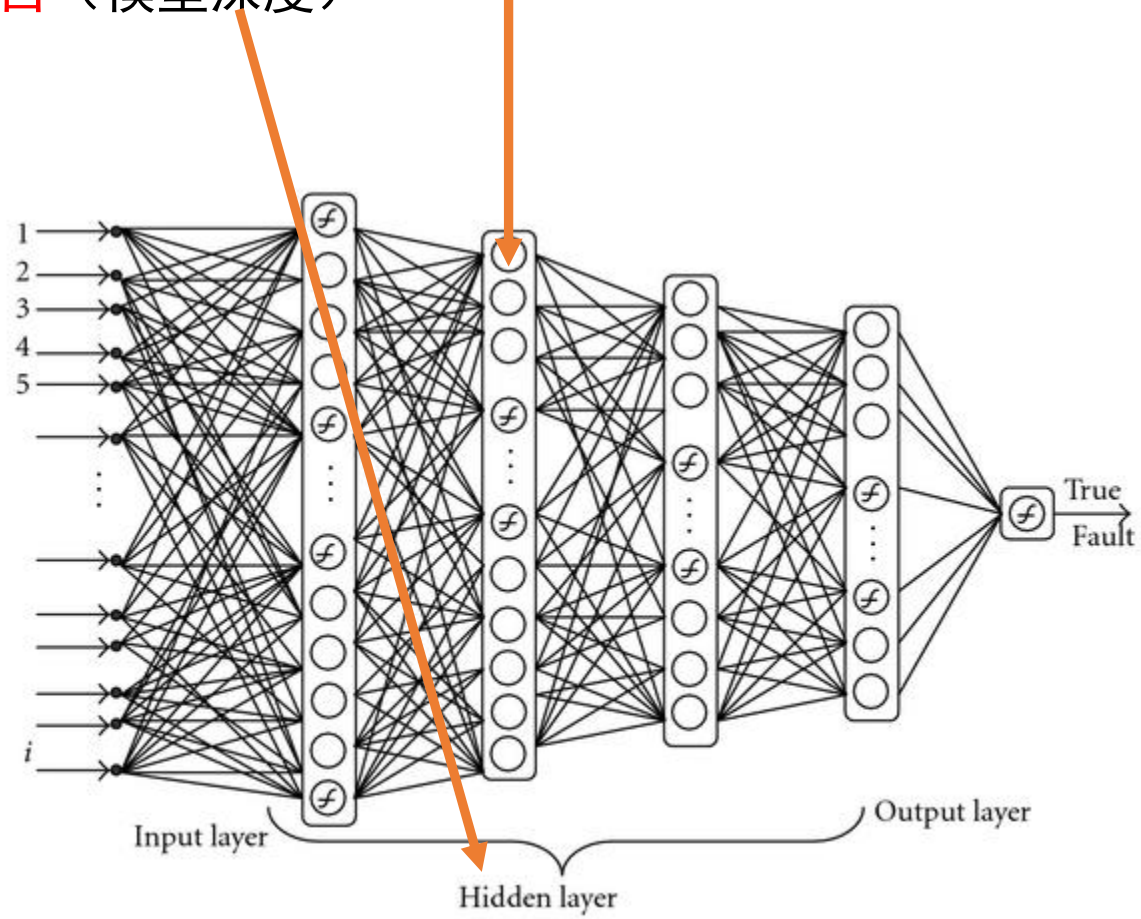
1. 神经元模型
2. 感知机*与多层网络
3. 误差逆传播算法
4. 全局最小和局部最小
5. 深度学习

*《统计学习方法》第2章除2.3.2和2.3.3以外

- 典型的深度学习模型就是很深层的神经网络

模型复杂度

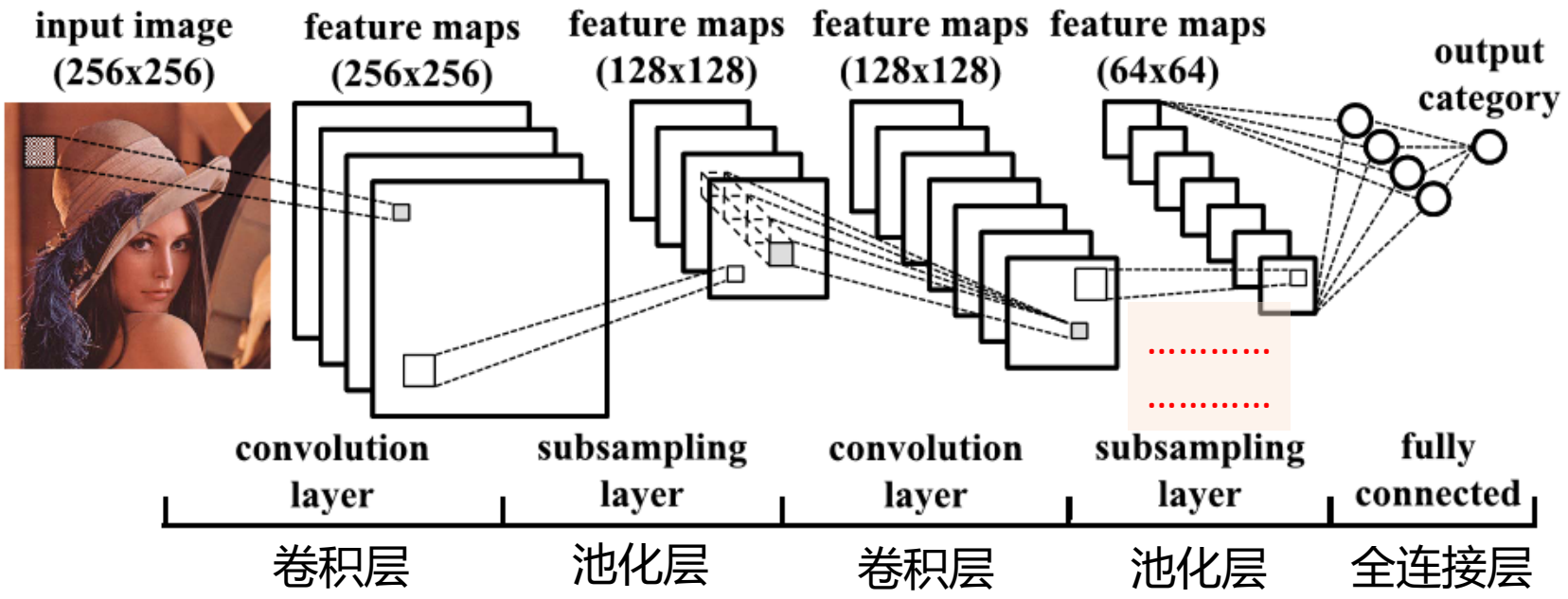
- 增加隐层神经元的数目（模型宽度）
- 增加隐层数目（模型深度）



深度学习

- 基础网络模型
  - 前馈神经网络
  - 卷积神经网络
  - 循环神经网络
  - 网络优化与正则化
  - 记忆与注意力机制
  - 无监督学习
- 进阶模型
  - 概率图模型
  - 玻尔兹曼机
  - 深度信念网络
  - 深度生成模型
  - 深度强化学习

卷积神经网络(CNN)



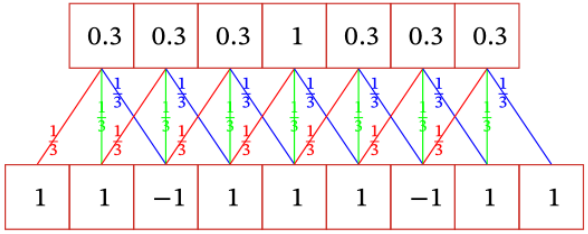
每个卷积层包含多个特征映射，每个特征映射是一个由多个神经元构成的“平面”，通过一种卷积滤波器提取输入的一种特征

采样层亦称“池化层pooling”，其作用是基于局部相关性原理进行亚采样，从而在减少数据量的同时，保留有用信息。

连接层就是传统神经网络对隐层与输出层的全连接

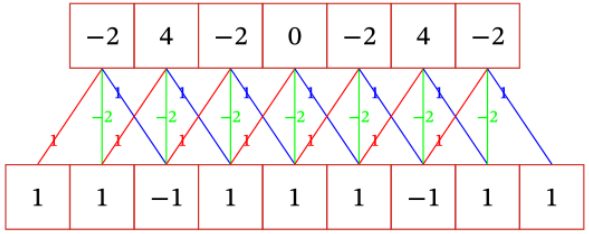
卷积

- 不同的滤波器来提取信号序列中的不同特征



(a) 滤波器 [1/3, 1/3, 1/3]

低频信息



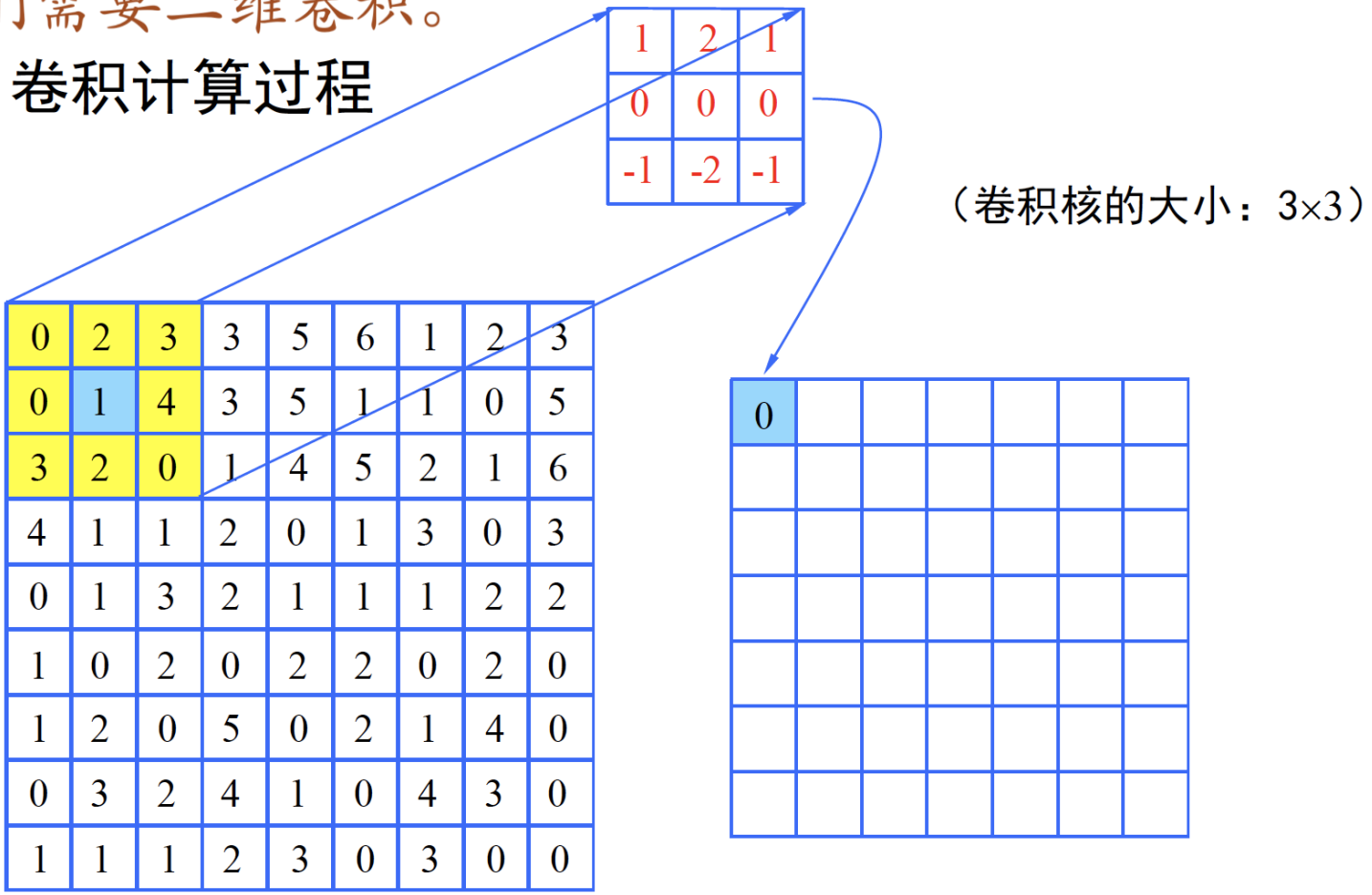
(b) 滤波器 [1, −2, 1]

高频信息

两维卷积

- 在图像处理中，图像是以二维矩阵的形式输入到神经网络中，因此我们需要二维卷积。

卷积计算过程



$0\times1 + 2\times2 + 3\times1 + 0\times0 + 1\times0 + 4\times0 + 3\times(-1) + 2\times(-2) + 0\times(-1) = 0$

卷积计算过程：

1	2	1
0	0	0
-1	-2	-1

（卷积核的大小：3×3）

0	2	3	3	5	6	1	2	3
0	1	4	3	5	1	1	0	5
3	2	0	1	4	5	2	1	6
4	1	1	2	0	1	3	0	3
0	1	3	2	1	1	1	2	2
1	0	2	0	2	2	0	2	0
1	2	0	5	0	2	1	4	0
0	3	2	4	1	0	4	3	0
1	1	1	2	3	0	3	0	0

0	8					

$2\times1 + 3\times2 + 3\times1 + 1\times0 + 4\times0 + 3\times0 + 2\times(-1) + 0\times(-2) + 1\times(-1) = 8$

卷积计算过程

1	2	1
0	0	0
-1	-2	-1

（卷积核的大小：3×3）

0	2	3	3	5	6	1	2	3
0	1	4	3	5	1	1	0	5
3	2	0	1	4	5	2	1	6
4	1	1	2	0	1	3	0	3
0	1	3	2	1	1	1	2	2
1	0	2	0	2	2	0	2	0
1	2	0	5	0	2	1	4	0
0	3	2	4	1	0	4	3	0
1	1	1	2	3	0	3	0	0

0	8	8				

$$3\times1 + 3\times2 + 5\times1 + 4\times0 + 3\times0 + 5\times0 + 0\times(-1) + 1\times(-2) + 4\times(-1) = 8$$

- 最大值采样（Maximum Pooling） pooling=汇聚

$$pool_{max}(R_k) = \max_{i \in R_k} a_i$$

- 最小值采样（Minimum Pooling）

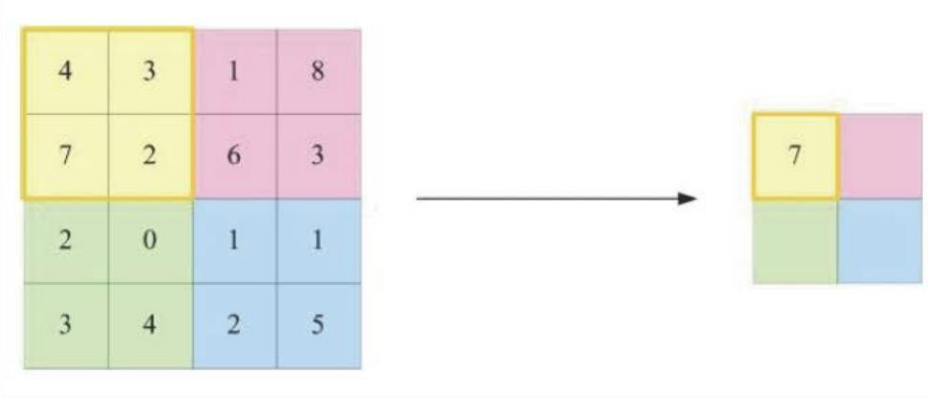
$$pool_{min}(R_k) = \min_{i \in R_k} a_i$$

- 平均值（Average Pooling）

$$pool_{ave}(R_k) = \frac{1}{|R_k|} \sum_{i \in R_k}^{|R_k|} a_i$$

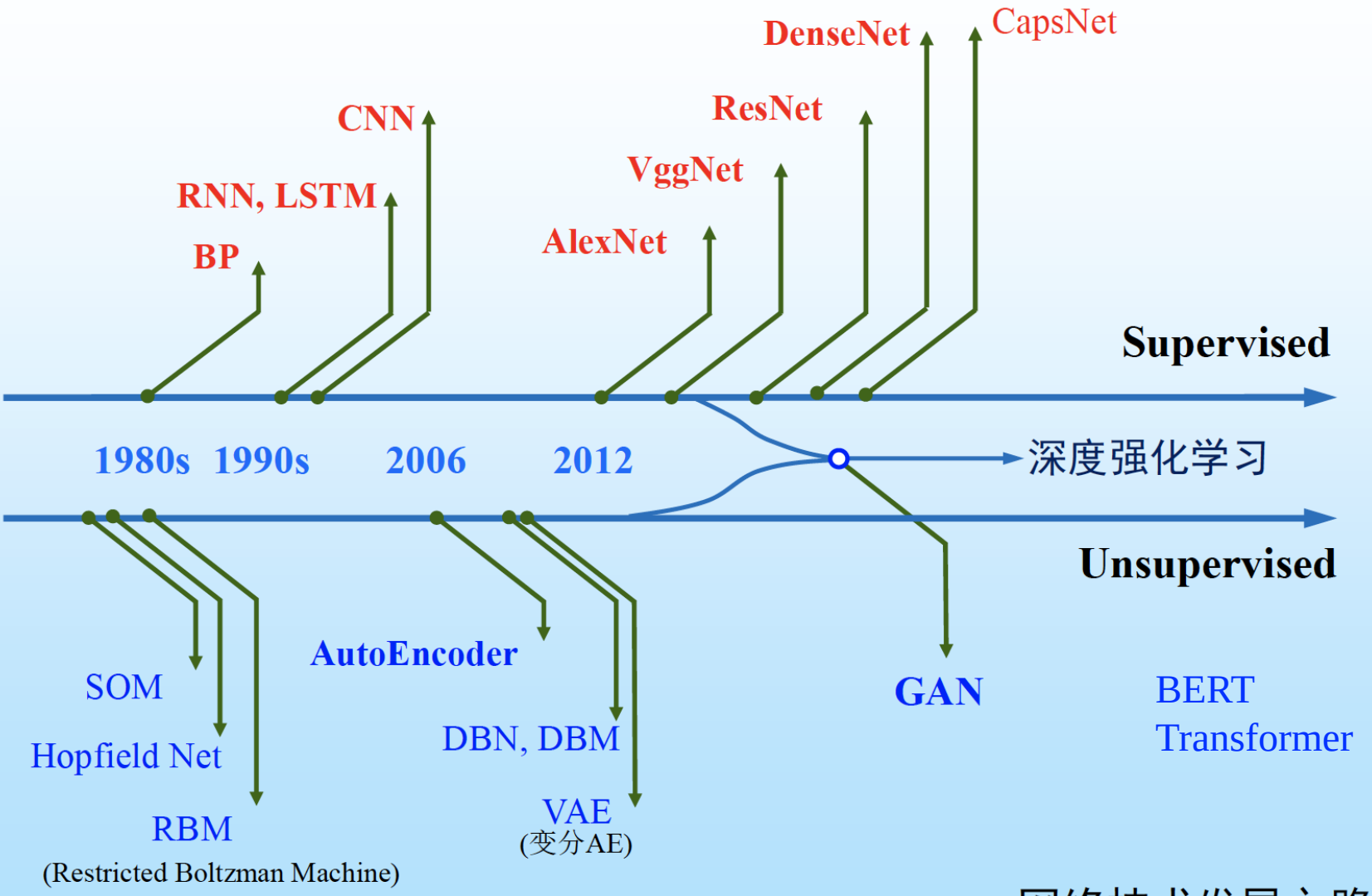
- TopK采样（ TopK Pooling）

$$pool_k(R_k) = \operatorname{topk}_{i \in R_k} a_i$$



深度卷积神经网络发展图

整体发展态势



深度学习

- 基础网络模型
  - 前馈神经网络
  - 卷积神经网络
  - 循环神经网络
  - 网络优化与正则化
  - 记忆与注意力机制
  - 无监督学习
- 进阶模型
  - 概率图模型
  - 玻尔兹曼机
  - 深度信念网络
  - 深度生成模型
  - 深度强化学习

我在2月1号放寒假离开中大

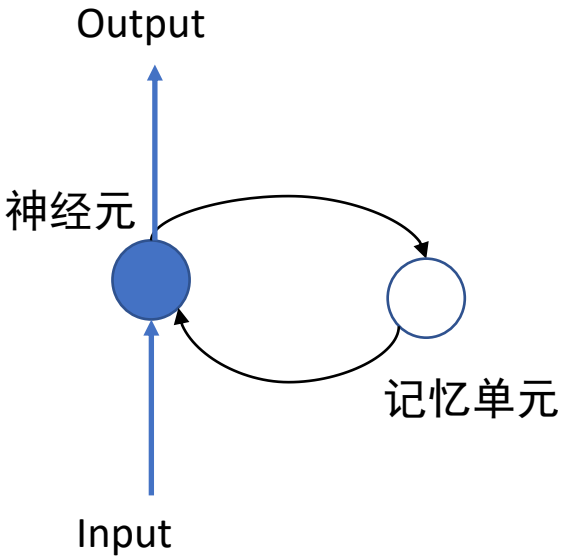
日期 地址

我在2月1号放寒假回中大

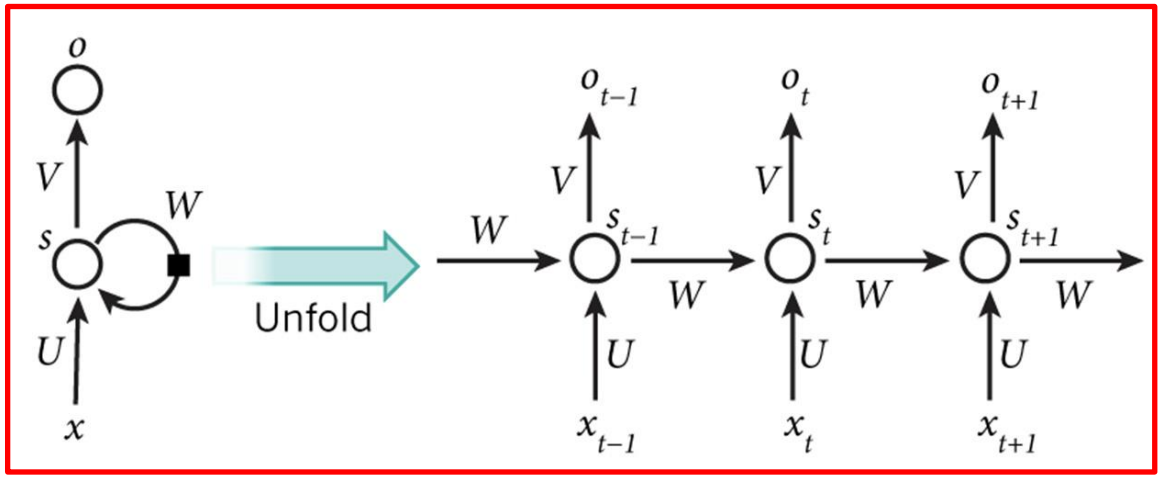
日期 地址

想知道地址：目的地还是出发地？

金融领域的应用：输入为公司公告文本



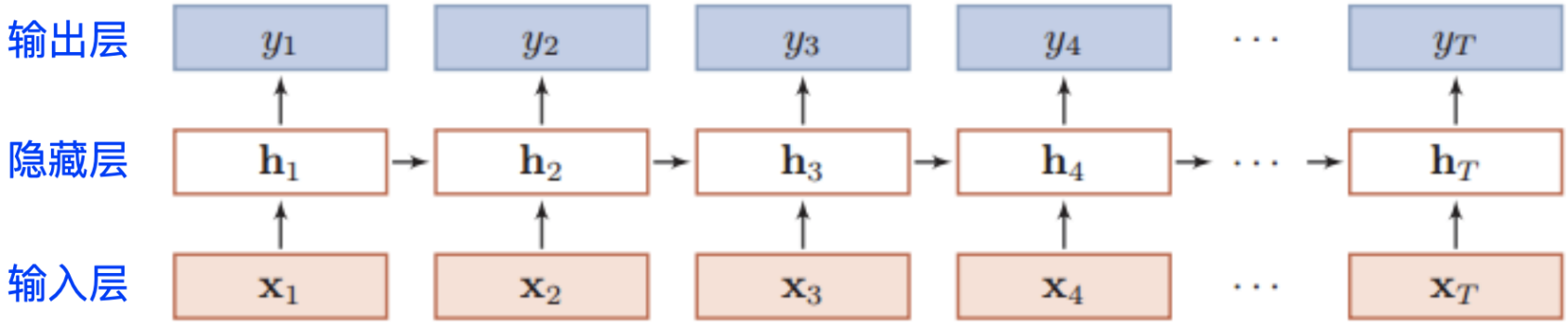
假设记忆单元初始化为0
神经元进行简单加和的操作



上下文依赖非常重要——我们需要网络能够有“记忆”，
不能看到“中大”就忘记之前看到的词“离开”或者“回”

(1) 单向循环神经网络

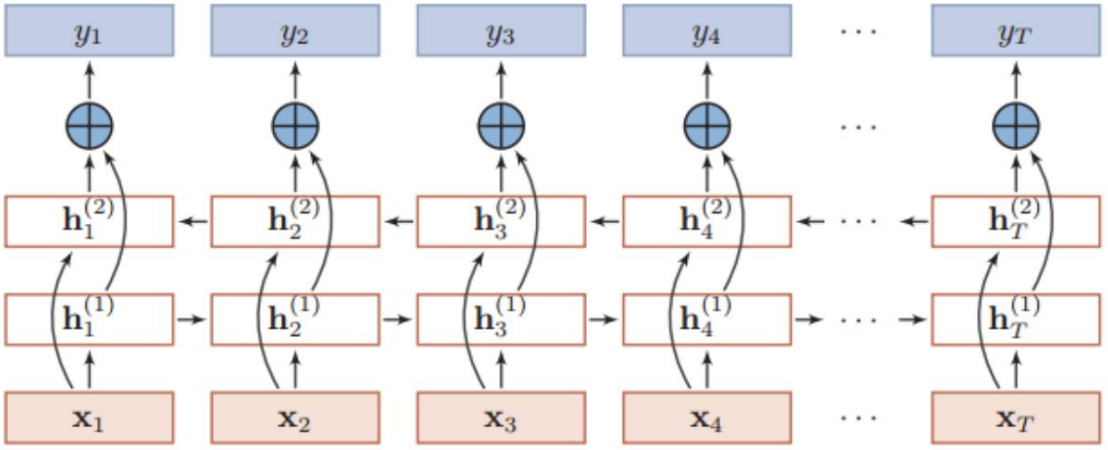
- 先来搞清楚只有一个隐藏层的循环神经网络的结构。
- 如下是一个按时间展开的循环神经网络图：



可以看到，连接不仅存在于相邻的层与层之间（比如输入层-隐藏层），
还存在于时间维度上的隐藏层与隐藏层之间（反馈连接，h_1 到 h_T）。

(2) 双向循环神经网络

- 双向循环神经网络（Bidirectional Recurrent Neural Network，Bi-RNN），它由两层循环神经网络组成，这两层网络都输入序列 x，但是信息传递方向相反。



中山大學
SUN YAT-SEN UNIVERSITY

第5章 神经网络

1. 神经元模型

（神经元、激活函数）

2. 感知机与多层网络

（感知机五要素、多层前馈神经网络五要素）

3. 误差逆传播算法

（推导过程）

4. 全局最小和局部最小

（概念、“跳出”策略）

5. 深度学习

（概念、增加模型复杂度的方法）

中山大學
SUN YAT-SEN UNIVERSITY

第7章 支持向量机*

1. 线性可分支持向量机与硬间隔最大化
2. 线性支持向量机与软间隔最大化
3. 非线性支持向量机与核函数
4. 序列最小最优化算法
5. 支持向量机回归

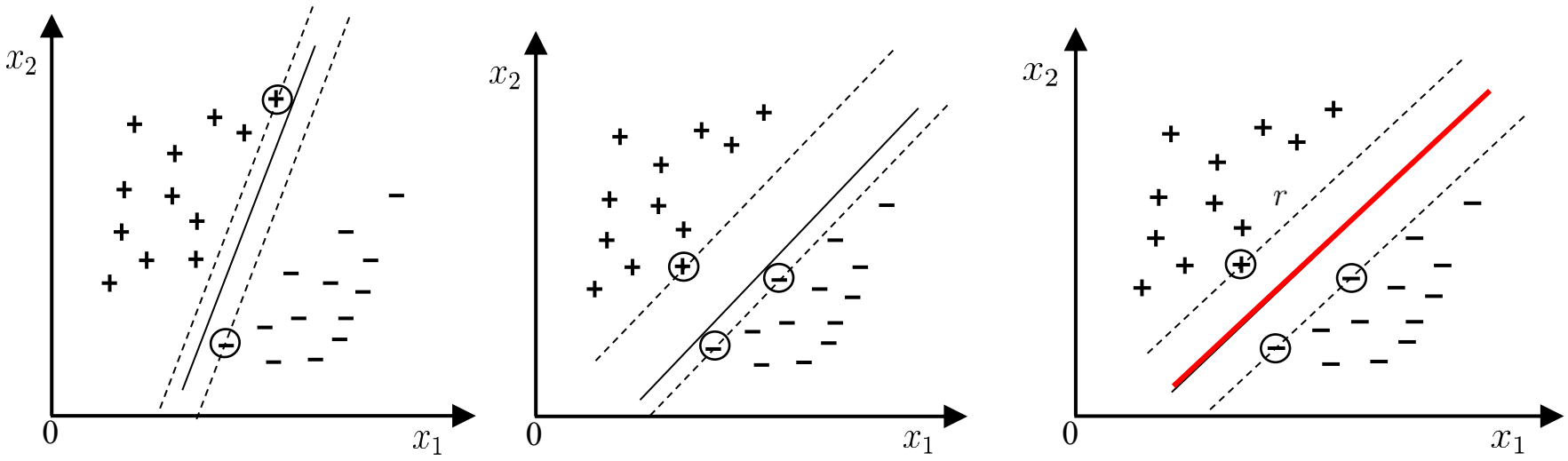
沈颖 副教授

sheny76@mail.sysu.edu.cn

*《统计学习方法》第7章中除7.2.4和7.3.2外所有内容

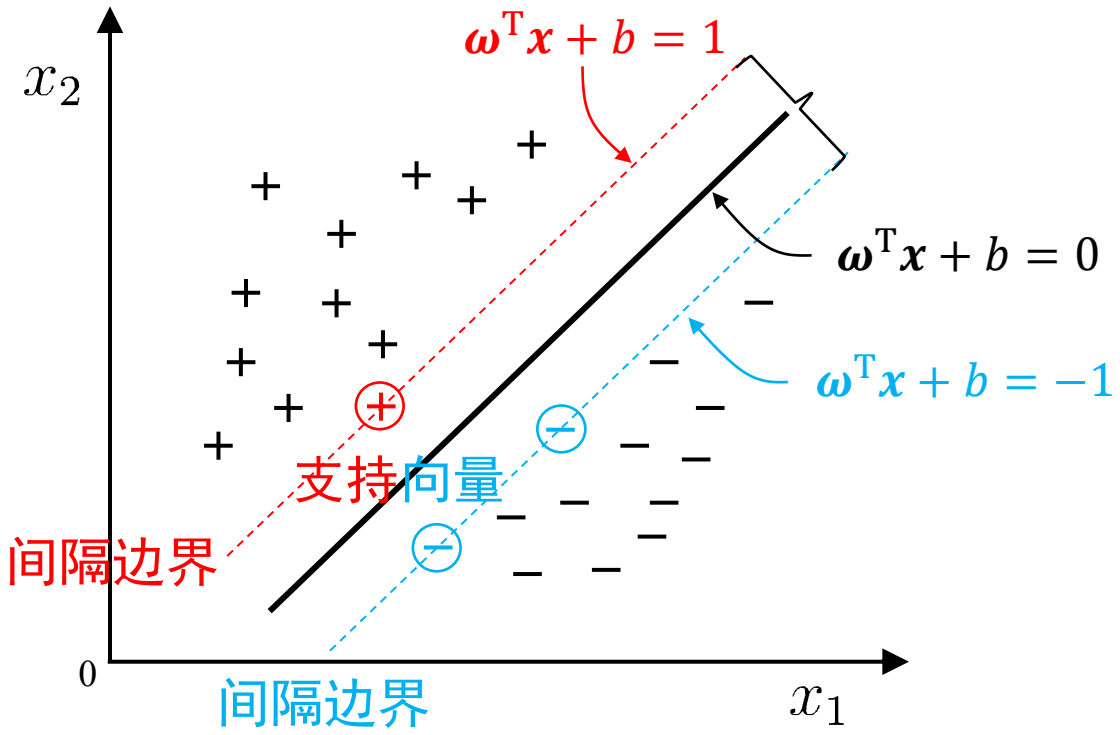
线性分类器回顾

将训练样本分开的超平面可能有很多，哪一个更好呢？



“正中间”的“容忍”性最好：

对接近两个类的分隔界的样本，红色的超平面受影响最小，分类结果最鲁棒，对未见示例的泛化能力最强



支持向量：在线性可分情况下，训练数据集的样本点中与分离超平面距离最近的样本点（使上述约束条件的等号成立）的示例称为支持向量(support vector)

概念——支持向量机

- 支持向量机（support vector machine, SVM）

  SVM是一种二分类模型，其基本模型是定义在特征空间上的间隔最大的分类器

  - 线性可分支持向量机

    当训练数据线性可分时，通过硬间隔最大化（hard margin maximization），学习的一种线性的分类器

  - 线性支持向量机（linear SVM）

    当训练数据近似线性可分时，通过软间隔最大化（soft margin maximization），学习的一种线性的分类器

  - 非线性支持向量机（non-linear SVM）

    当训练数据线性不可分时，通过使用核技巧（kernel trick）及软间隔最大化，学习的一种非线性的分类器

- **函数间隔定义**

对于给定的训练数据集T和超平面$(\boldsymbol{\omega}, b)$，定义超平面$(\boldsymbol{\omega}, b)$关于样本点$(\boldsymbol{x}_i, y_i)$的函数间隔为

$$\hat{r}_i = y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b\right)$$

函数间隔

函数间隔（functional margin）

超平面$(\boldsymbol{\omega}, b)$关于样本点$(\boldsymbol{x}_i, y_i)$的函数间隔需满足约束参数$\boldsymbol{\omega}$和b，使得间隔r最大

$$\max_{\boldsymbol{\omega},b} \frac{2}{\|\boldsymbol{\omega}\|}$$
$$\text{s.t.} \quad y_i(\boldsymbol{\omega}^T x_i + b) \geq 1, \qquad i = 1,2,\dots,m.$$

最大化向量模长$\|\boldsymbol{\omega}\|^{-1}$，
等价于最小化$\|\boldsymbol{\omega}\|^2$.

重写

$$\min_{\boldsymbol{\omega},b} \frac{1}{2}\|\boldsymbol{\omega}\|^2$$

支持向量机的基本型

$$\text{s.t.} \quad y_i(\boldsymbol{\omega}^T x_i + b) \geq 1, \qquad i = 1,2,\dots,m.$$

函数间隔

- 定义超平面$(\boldsymbol{\omega}, b)$关于样本点$(\boldsymbol{x}_i, y_i)$的函数间隔为

$$\hat{r}_i = y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b\right)$$

- 定义超平面$(\boldsymbol{\omega}, b)$关于样本点$(\boldsymbol{x}_i, y_i)$的几何间隔为

$$r_i = \frac{y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b\right)}{\|\boldsymbol{\omega}\|} = \frac{\hat{r}_i}{\|\boldsymbol{\omega}\|}$$

- 线性可分支持向量机

$$\min_{\boldsymbol{\omega},b} \quad \frac{1}{2}\|\boldsymbol{\omega}\|^2$$
$$\text{s.t.} \quad y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1, i = 1, \cdots, m$$

通过求解对偶问题得到原始问题的最优解

第一步：定义拉格朗日函数

$$L(\boldsymbol{\omega}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right)$$

$\boldsymbol{\alpha} = (\alpha_1; \cdots; \alpha_m)$ 拉格朗日乘子向量

$\alpha_i \geq 0$ 拉格朗日乘子

- 线性可分支持向量机

$$\begin{aligned} \min_{\boldsymbol{\omega},b} \quad & \frac{1}{2}\|\boldsymbol{\omega}\|^2 \\ \text{s.t.} \quad & y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1, i = 1, \cdots, m \end{aligned}$$

通过求解对偶问题得到原始问题的最优解

第一步：定义拉格朗日函数

$$L(\boldsymbol{\omega}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right)$$

根据拉格朗日对偶性，原始问题的对偶问题是极大极小问题：

$$\max_{\boldsymbol{\alpha}} \min_{\boldsymbol{\omega},b} L(\boldsymbol{\omega}, b, \boldsymbol{\alpha})$$

- 线性可分支持向量机

$$\begin{aligned}\min_{\boldsymbol{\omega},b}\quad & \frac{1}{2}\|\boldsymbol{\omega}\|^2\\ \text{s.t.}\quad & y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geq 1, i=1,\cdots,m\end{aligned}$$

通过求解对偶问题得到原始问题的最优解

第一步：定义拉格朗日函数

$$L(\boldsymbol{\omega},b,\boldsymbol{\alpha})=\frac{1}{2}\|\boldsymbol{\omega}\|^2+\sum_{i=1}^{m}\alpha_i\left(1-y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\right)$$

$$\boldsymbol{\alpha}=(\alpha_1;\cdots;\alpha_m)$$

根据拉格朗日对偶性，原始问题的对偶问题是极大极小问题：

$$\max_{\boldsymbol{\alpha}}\min_{\boldsymbol{\omega},b}L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$$

为了得到对偶问题的解，需先求$L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$对$\boldsymbol{\omega},b$的极小，

再求对$\boldsymbol{\alpha}$的极大

- 线性可分支持向量机

$$\min_{\boldsymbol{\omega},b} \quad \frac{1}{2}\|\boldsymbol{\omega}\|^2$$
$$\text{s.t.} \quad y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geq 1, i = 1, \cdots, m$$

通过求解对偶问题得到原始问题的最优解

第一步： 定义拉格朗日函数

$$L(\boldsymbol{\omega}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i\left(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right)$$

第二步： 求$\min_{\boldsymbol{\omega},b} L(\boldsymbol{\omega}, b, \boldsymbol{\alpha})$

将拉格朗日函数分别对$\boldsymbol{\omega}$和b求偏导数并令其等于0

$$\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha})}{\partial \boldsymbol{\omega}} = \boldsymbol{\omega} - \sum_{i=1}^{m} a_i y_i \boldsymbol{x}_i$$

$$\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha})}{\partial \boldsymbol{b}} = -\sum_{i=1}^{m} a_i y_i$$

$$\boldsymbol{\omega} = \sum_{i=1}^{m} a_i y_i \boldsymbol{x}_i$$
$$\sum_{i=1}^{m} a_i y_i = 0$$

$$\frac{\partial \|\boldsymbol{\omega}\|^2}{\partial \boldsymbol{\omega}} = 2\boldsymbol{\omega}$$
$$\frac{\partial \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}}{\partial \boldsymbol{\omega}} = \boldsymbol{x}$$

对偶问题

$$\max_{a}\left(\min_{b,w}\frac{1}{2}\|\boldsymbol{\omega}\|^2-\sum_{i=1}^{m}a_i\left(y_i(\boldsymbol{\omega}^T\boldsymbol{x}_i+b)-1\right)\right)$$

第三步：求$\min_{\boldsymbol{\omega},b}L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$对$\boldsymbol{\alpha}$的极大，即对偶问题

$$\boldsymbol{\omega}=\sum_{i=1}^{m}a_i\,y_i\boldsymbol{x}_i \qquad \sum_{i=1}^{m}a_i\,y_i=0$$

$$\max_{a}\left(\min_{b,\boldsymbol{\omega}}\frac{1}{2}\|\boldsymbol{\omega}\|^2-\sum_{i=1}^{m}a_i\left(y_i\boldsymbol{\omega}^T\boldsymbol{x}_i-1\right)-\sum_{i=1}^{m}a_i\,y_i b\right)$$

$$\max_{a}\left(\min_{b,\boldsymbol{\omega}}\frac{1}{2}\|\boldsymbol{\omega}\|^2+\sum_{i=1}^{m}a_i\left(1-y_i\boldsymbol{\omega}^T\boldsymbol{x}_i\right)\right)$$

$$\max_{a}\left(\min_{b,\boldsymbol{\omega}}\frac{1}{2}\|\boldsymbol{\omega}\|^2+\sum_{i=1}^{m}a_i-\sum_{i=1}^{m}a_i\,y_i\boldsymbol{\omega}^T\boldsymbol{x}_i\right)$$

$\boldsymbol{\omega}^T\boldsymbol{\omega}$

$$\max_{a}\left(-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}a_i\,a_j\,y_iy_jx_i^Tx_j+\sum_{i=1}^{m}a_i\right)$$

学习的对偶算法

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

$$\text{s.t.} \sum_{i=1}^{m} \alpha_i y_i = 0,$$

$$\alpha_i \geq 0, i = 1, \cdots, m$$

求极大

$$\min_{\boldsymbol{\alpha}} -\sum_{i=1}^{m} \alpha_i + \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

$$\text{s.t.} \sum_{i=1}^{m} \alpha_i y_i = 0,$$

$$\alpha_i \geq 0, i = 1, \cdots, m$$

求极小

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

$$\text{s.t.} \sum_{i=1}^{m} \alpha_i y_i = 0,$$

$$\alpha_i \geq 0, i = 1, \cdots, m$$

第四步：

求得最优解$\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_d^*)^{\mathrm{T}}$，

j为$\boldsymbol{\alpha}^*$中的一个正分量，存在下标j，使得$\alpha_j^* > 0$，

对于线性可分问题根据 KKT条件 求出$(\boldsymbol{\omega}^*, b^*)$

$$\boldsymbol{\omega}^* = \sum_{i=1}^{m} \alpha_i^* y_i \boldsymbol{x}_i, \qquad b^* = y_j - \sum_{i=1}^{m} \alpha_i^* y_i \boldsymbol{x}_i^T \boldsymbol{x}_j$$

对偶问题

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

$$\text{s.t.} \sum_{i=1}^{m} \alpha_i y_i = 0,$$

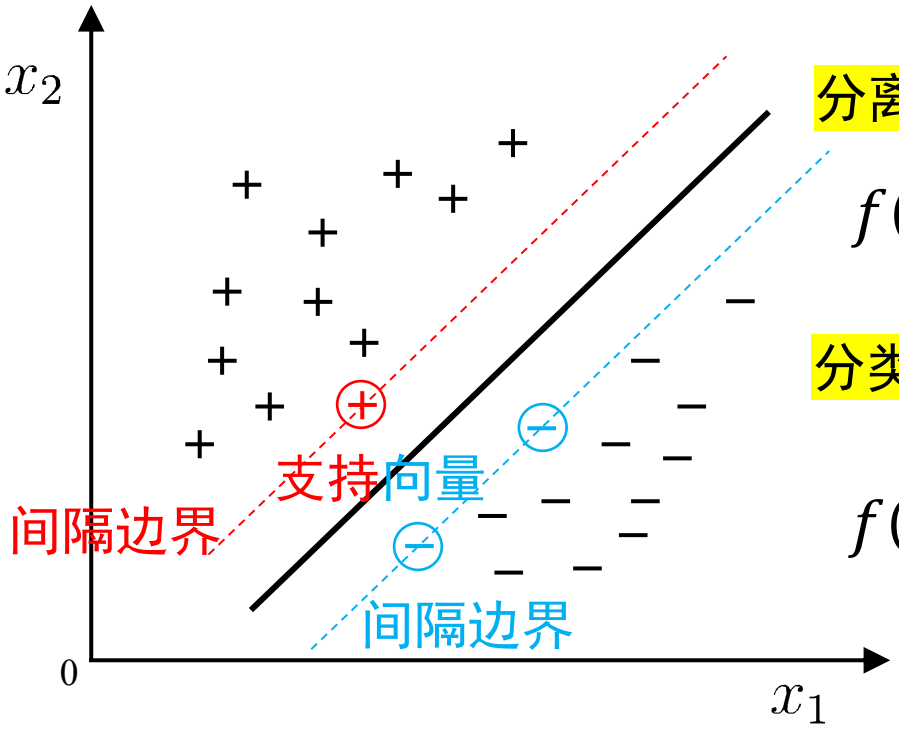
$$\alpha_i \geq 0, i = 1, \cdots, m$$

KKT条件：

$$\begin{cases} \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\omega}, b, \boldsymbol{\alpha}) = 0 \\ \nabla_b \mathcal{L}(\boldsymbol{\omega}, b, \boldsymbol{\alpha}) = 0 \\ \alpha_i \left(1 - \boldsymbol{y_i}(\boldsymbol{\omega}^T x_i + b)\right) = 0 \\ \boldsymbol{y_i}(\boldsymbol{\omega}^T x_i + b) \geq 1 \\ \alpha_i \geq 0, i = 1, \ldots m \end{cases}$$

第五步： 最终模型$f(\boldsymbol{x}) = \text{sign}\left(\boldsymbol{\omega}^{*\text{T}}\boldsymbol{x} + b^*\right)$

$$\boldsymbol{\omega} = \sum_{i=1}^{m} a_i y_i \boldsymbol{x}_i \qquad \sum_{i=1}^{m} a_i y_i = 0$$



分离超平面：

$$f(\boldsymbol{x}) = \boldsymbol{\omega}^{*\text{T}}\boldsymbol{x} + b^* = \sum_{i=1}^{m} \alpha_i^* y_i \boldsymbol{x}_i^{\text{T}}\boldsymbol{x} + b^*$$

分类决策函数：

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i^* y_i \boldsymbol{x}_i^{\text{T}}\boldsymbol{x} + b^*\right)$$

现象： 分类决策函数只依赖于输入$\boldsymbol{x}$和训练样本输入的内积

第8章 集成学习

1. 个体与集成
2. Boosting
3. Bagging与随机森林
4. 结合策略

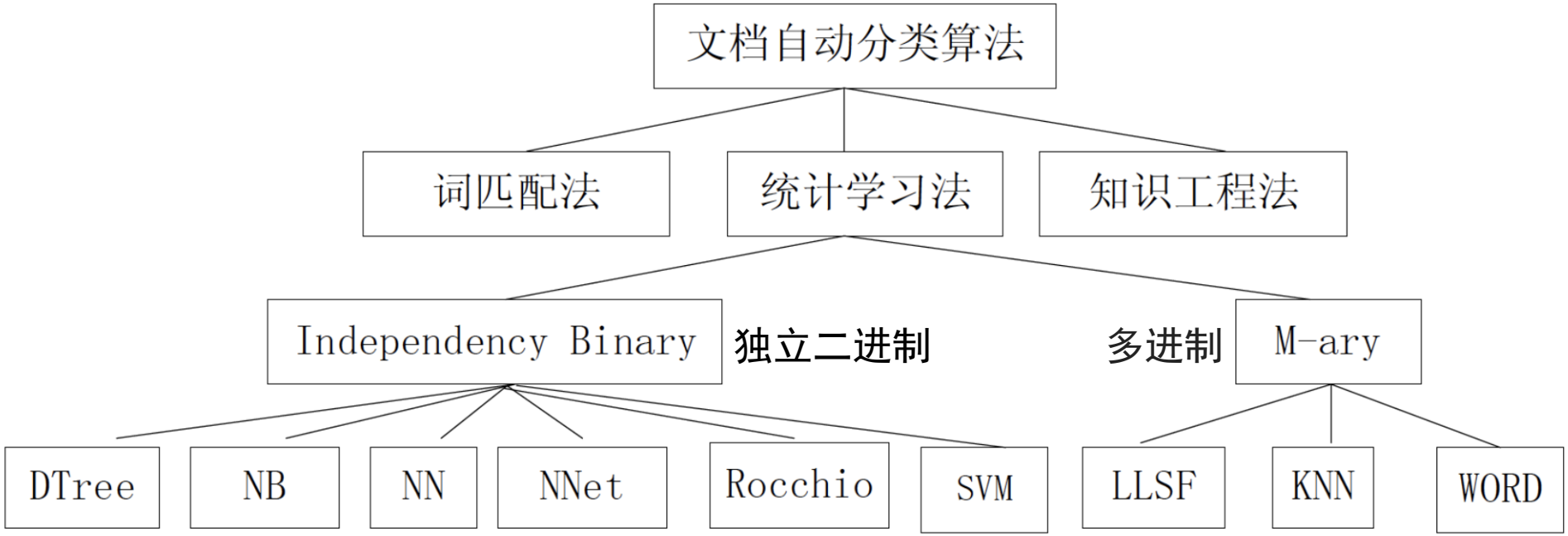
沈颖 副教授

sheny76@mail.sysu.edu.cn

机器学习-第8章（除8.5以外）

统计学习方法（第8章除8.2和8.4.3以外）

自动分类算法分类


文档自动分类算法
词匹配法
统计学习法
知识工程法
Independency Binary
独立二进制
多进制
M-ary
DTree
NB
NN
NNet
Rocchio
SVM
LLSF
KNN
WORD
决策树
朴素贝叶斯
神经网络
支持向量机
K阶近邻

概念——集成学习

- 集成学习（ensemble learning）
  - “三个臭皮匠，顶个诸葛亮”
  - “不积跬步，无以至千里”

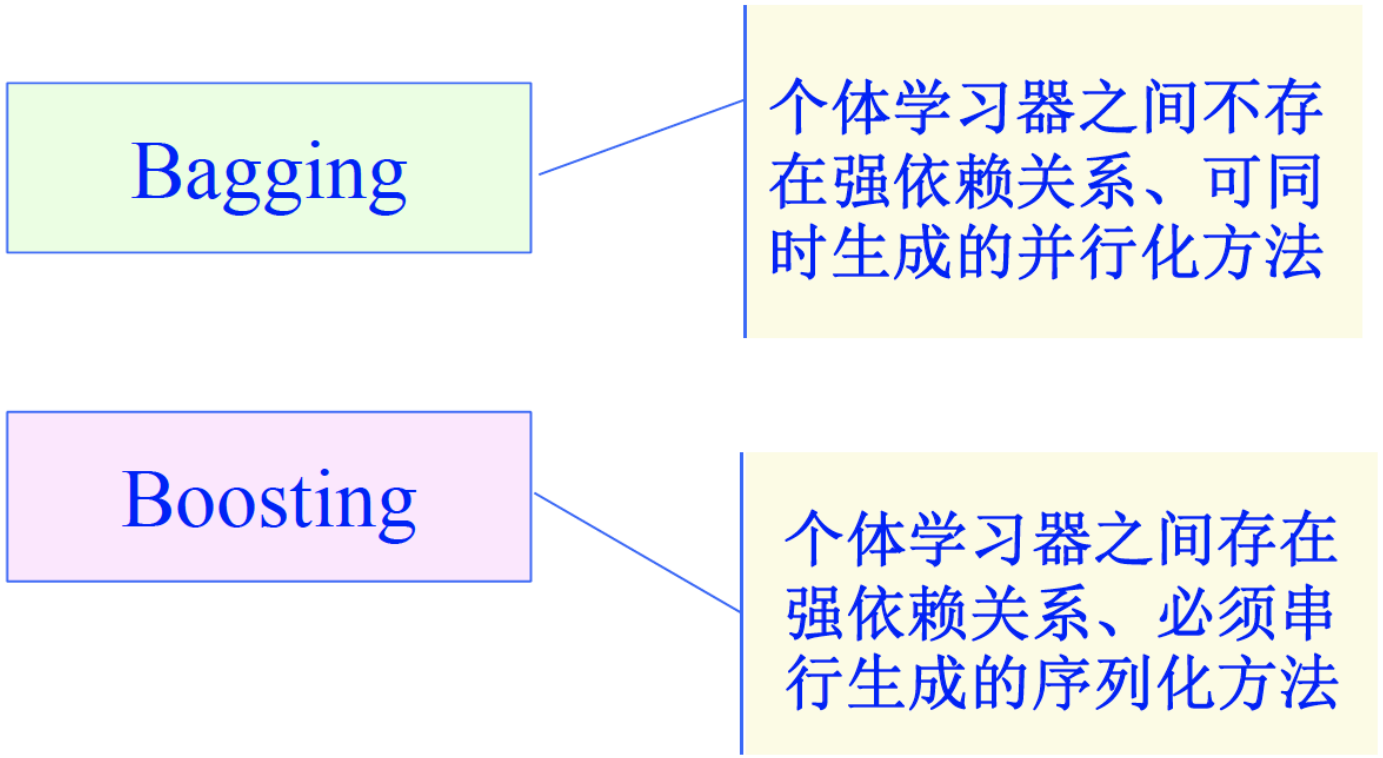
- 什么是集成学习

  集成学习通过构建并结合多个学习器来完成学习任务，
  有时也被称为多分类器系统（multi-classifier system）、
  基于委员会的学习（committee-based learning）等。

将多个学习器进行结合，
常可获得比单一学习器显著优越的泛化性能

集成学习方法分类

- 根据个体学习器的生成方式，集成学习方法分为两大类



集成学习的核心：如何产生并结合“好而不同”的个体学习器？

- 结合（怎样组合弱分类器）
  - 平均法：用于数值型输出，有简单平均法、加权平均法等
  - 投票法：用于分类任务，有绝对多数投票法、相对多数投票法、加权投票法等
  - 学习法：用于训练数据较多的情形，代表是Stacking

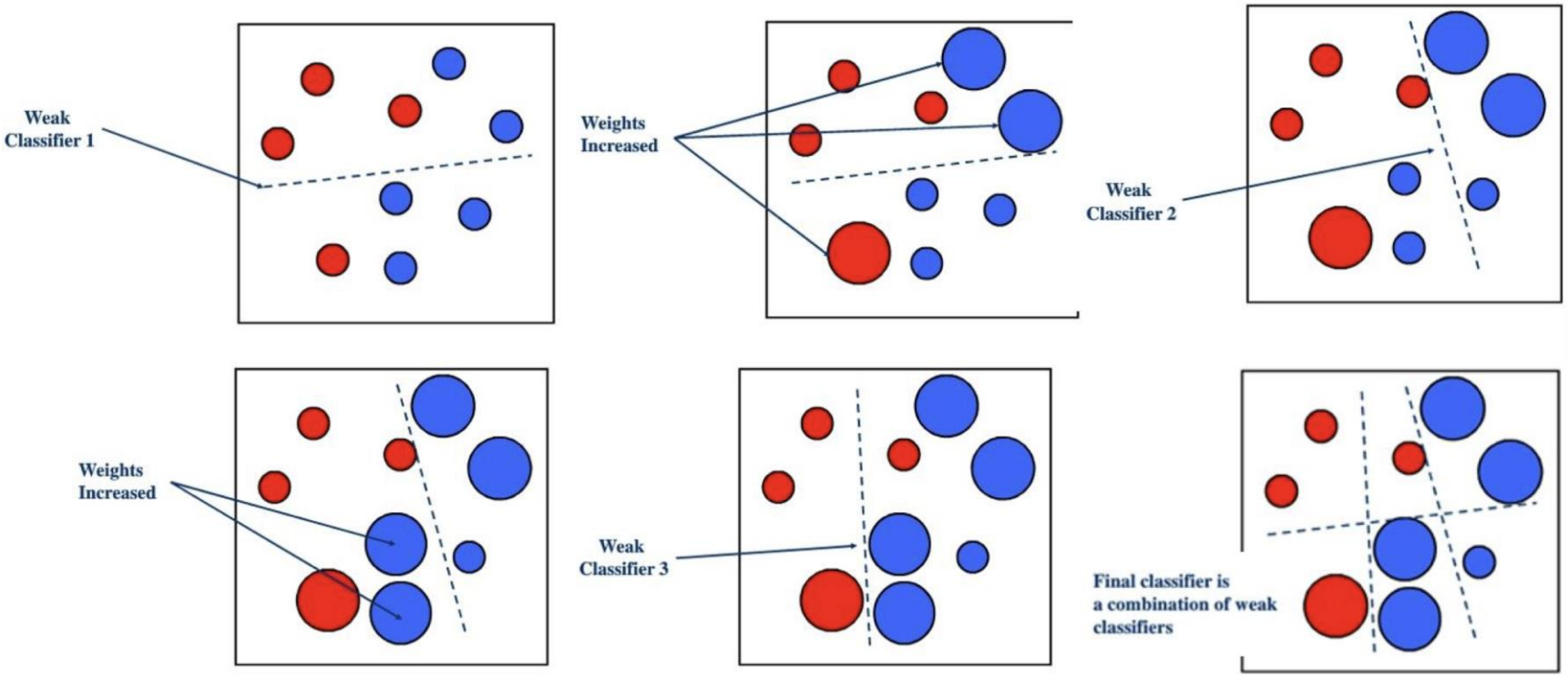
中山大學
SUN YAT-SEN UNIVERSITY

第8章 集成学习

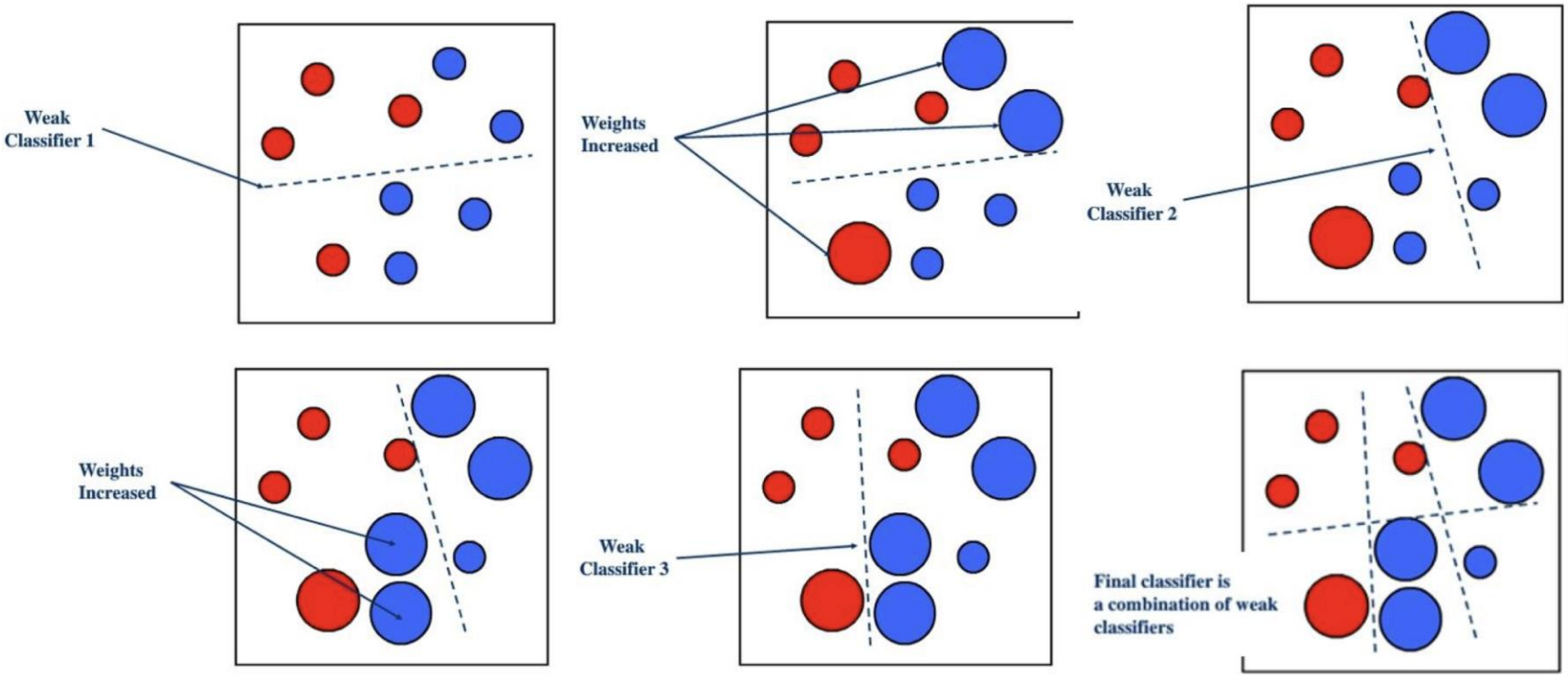
1. 个体与集成
2. Boosting
3. Bagging与随机森林
4. 结合策略

工作机制

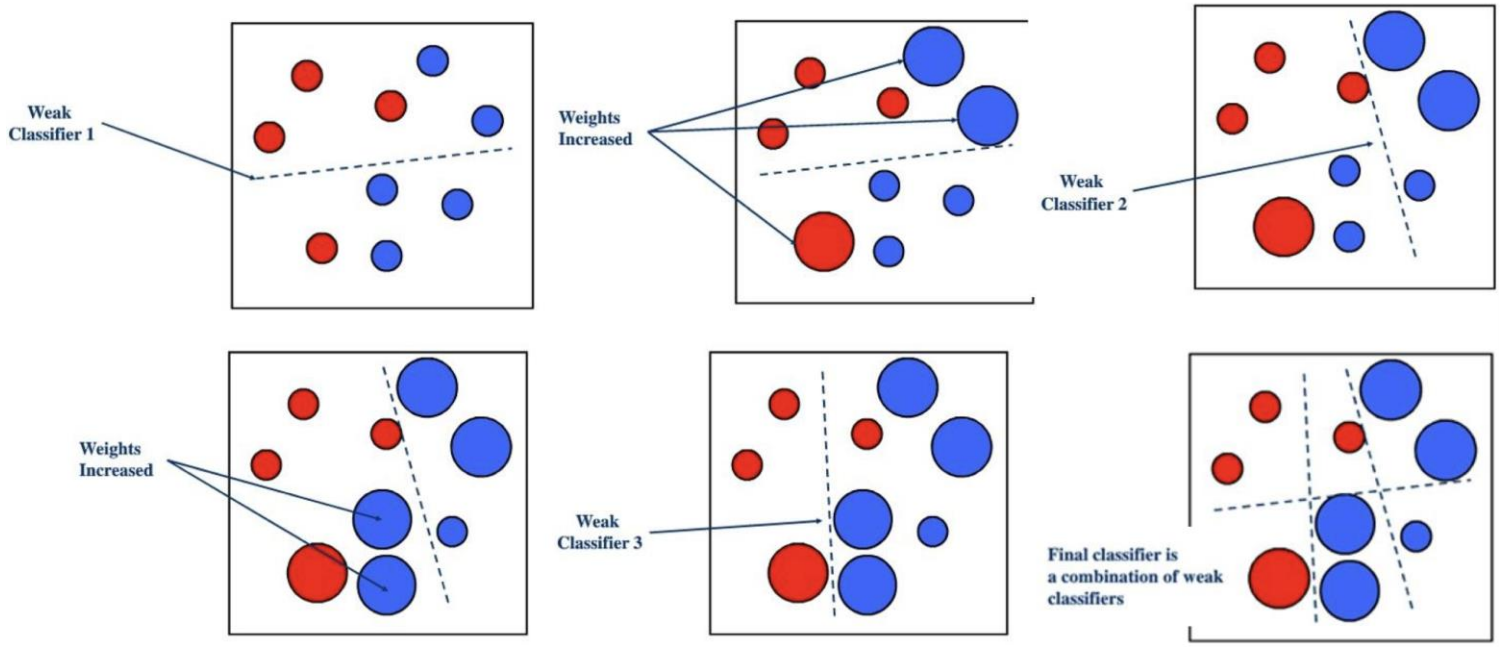
- 从初始训练集训练出一个基学习器，再根据基学习器的表现对训练样本分布进行调整，使得先前基学习器做错的训练样本在后续受到更多关注



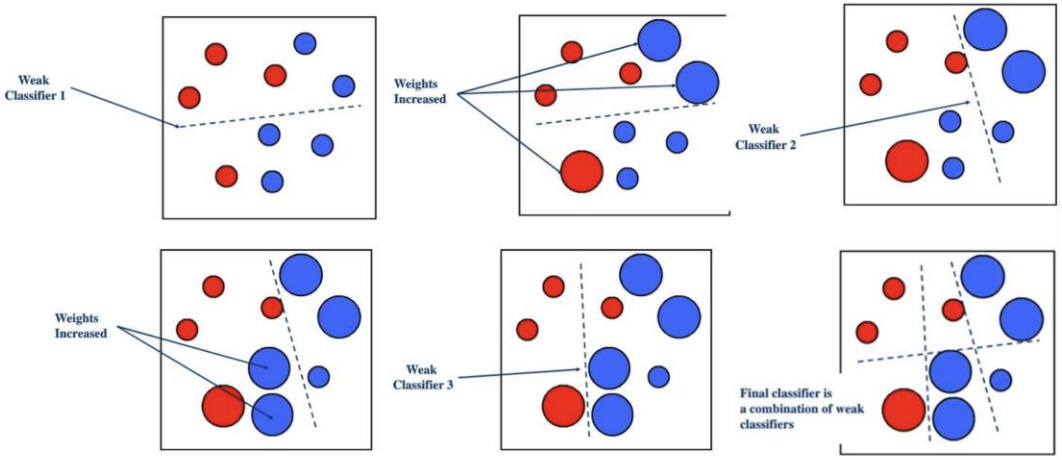
- 工作机制
  - 从初始训练集训练出一个基学习器，再根据基学习器的表现对训练样本分布进行调整，使得先前基学习器做错的训练样本在后续受到更多关注
  - 从调整样本分布后的训练集训练出下一个基学习器



- 工作机制
  - 从初始训练集训练出一个基学习器，再根据基学习器的表现对训练样本分布进行调整，使得先前基学习器做错的训练样本在后续受到更多关注
  - 从调整样本分布后的训练集训练出下一个基学习器
  - 重复进行，直至基学习器数目达到事先指定的值T



- 工作机制
  - 从初始训练集训练出一个基学习器，再根据基学习器的表现对训练样本分布进行调整，使得先前基学习器做错的训练样本在后续受到更多关注
  - 从调整样本分布后的训练集训练出下一个基学习器
  - 重复进行，直至基学习器数目达到事先指定的值T
  - 将这T个基学习器进行加权结合，把一个弱分类器提升为一个强分类器。



- 最具代表性的算法——AdaBoost

AdaBoost

Adaptive Boosting

A learning algorithm

Building a strong classifier from a lot of weaker ones

Boosting – AdaBoost推导

- AdaBoost 算法有多种推导方式，比较容易理解的是加性模型(additive model)，即基学习器的线性组合

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} a_t h_t(\boldsymbol{x})$$

- 指数损失函数

$$\ell(H|D) = \frac{1}{m}\sum_{t=1}^{m} \mathbb{I}[H(x_i) \neq y_i]$$

AdaBoost算法基本思想

- 基本想法：通过调整权值，产生一个新的样本分布及 h ，使前一个h错误率增加

$$\boldsymbol{a_t} = \frac{1}{2}\ln((1-\varepsilon_t)/\varepsilon_t)$$

IF 数据 i 在 $h_t(\boldsymbol{x})$ 判断失败

增大 ${w_i}^t$ to ${w_i}^{t+1}$

$${w_i}^{t+1} = {w_i}^t * \boldsymbol{a_t}$$

$$\Rightarrow \quad {w_i}^{t+1} = {w_i}^t * \exp(\boldsymbol{a_t})$$

IF 数据 i 在 $h_t(\boldsymbol{x})$ 判断成功

降低 ${w_i}^t$ to ${w_i}^{t+1}$

$${w_i}^{t+1} = \frac{{w_i}^t}{\boldsymbol{a_t}}$$

$$\Rightarrow \quad {w_i}^{t+1} = {w_i}^t * \exp(-\boldsymbol{a_t})$$

中山大學
SUN YAT-SEN UNIVERSITY

第8章 集成学习

1. 个体与集成
2. Boosting
3. Bagging与随机森林
4. 结合策略

Bagging（Bootstrap Aggregating）算法

Bagging是通过组合**随机生成的训练集**而改进分类的集成算法。

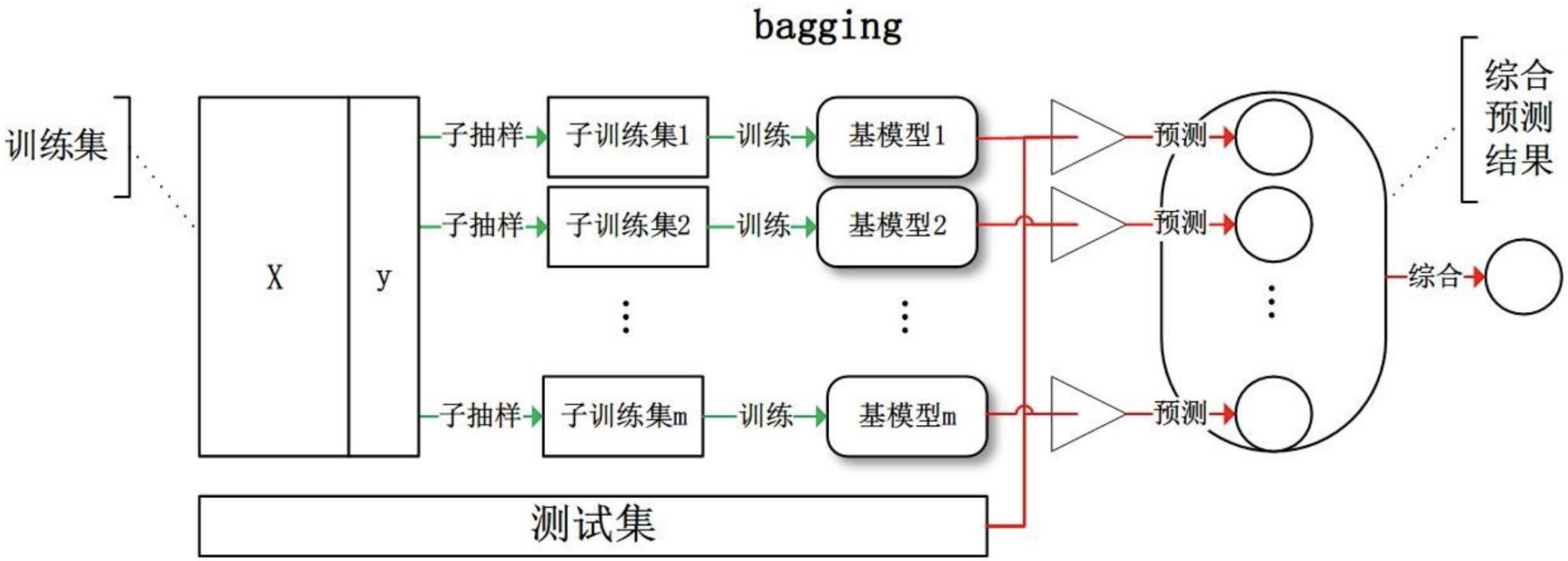
Bagging每次训练数据时只使用训练集中的某个子集作为当前训练集（有放回随机抽样），每一个训练样本在某个训练集中可以多次或不出现。

经过 T 次训练后，可得到 T 个不同的分类器。

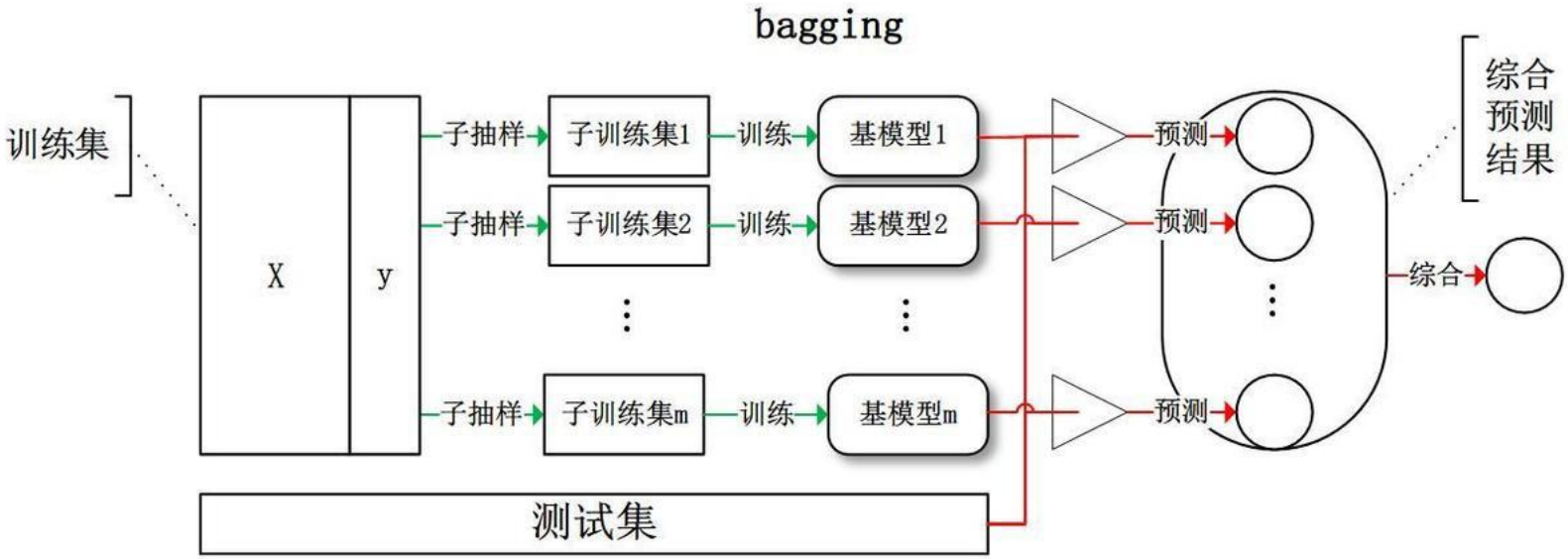
Bagging与随机森林

对一个测试样例进行分类时，分别调用这 T 个分类器，得到 T 个分类结果。

最后把这T个分类结果中出现次数多的类赋予测试样例。



Bagging算法特点



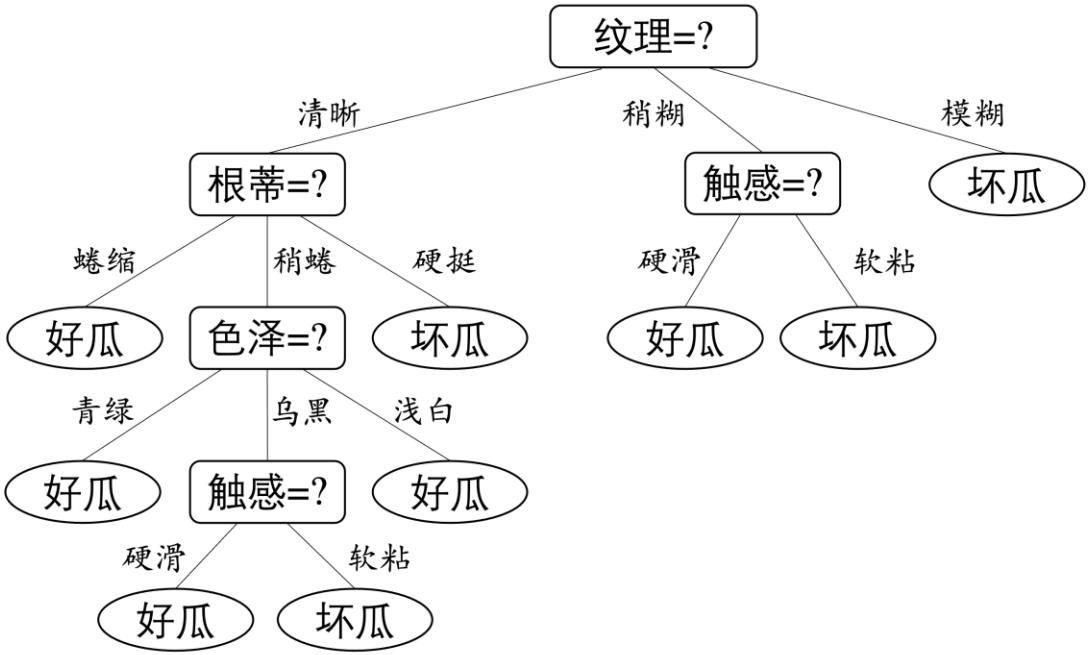
- **时间复杂度低**
  - 假定基学习器的计算复杂度为$O(m)$，采样与投票/平均过程的复杂度为$O(s)$，则bagging的复杂度大致为$T(O(m)+O(s))$
  - 由于$O(s)$很小且T是一个不大的常数
  - 训练一个bagging集成与直接使用基学习器的复杂度同阶

Bagging 是一个很高效的集成学习算法

随机森林

- ✓ 随机森林是bagging的一个扩展变种
- ✓ 采样的随机性（自助采样法）
- ✓ 属性选择的随机性（基决策树最优属性的选择）

传统决策树在选择划分属性时是在当前结点的属性集合（假定有d个属性）中选择一个最优属性



Bagging与随机森林

- 在随机森林中，对基决策树的每个结点，先从该结点的属性集合中随机选择一个包含k个属性的子集
- 然后再从这个自己中选择一个最优属性用于划分

随机森林对Bagging 只做了小改动，
Bagging 中基学习器的"多样性"仅通过样本扰动(通过对初始训练集采样)而来不同，
随机森林中基学习器的多样性不仅来自样本扰动，还来自属性扰动，
这就使得最终集成的泛化性能可通过个体学习器之间差异度的增加而进一步提升.

中山大學
SUN YAT-SEN UNIVERSITY

第8章 集成学习

1. 个体与集成
2. Boosting
3. Bagging与随机森林
4. 结合策略

结合策略

- 集成学习结合（怎样组合弱分类器）
  - 平均法：用于数值型输出，有简单平均法、加权平均法等
  - 投票法：用于分类任务，有绝对多数投票法、相对多数投票法、加权投票法等
  - 学习法：用于训练数据较多的情形，代表是Stacking

简单平均和加权平均

- 简单平均法

$$H(\boldsymbol{x}) = \frac{1}{T}\sum_{i=1}^{T} h_i(\boldsymbol{x})$$

- 加权平均法

$$H(\boldsymbol{x}) = \frac{1}{T}\sum_{i=1}^{T} \boldsymbol{\omega_i} h_i(\boldsymbol{x})$$

$$\omega_i \geq 0, \sum_{i=1}^{T} \omega_i = 1$$

平均法

投票法

- 绝对多数投票法（majority voting）

$$H(\boldsymbol{x}) = \begin{cases} c_j, \sum_{i=1}^{T} h_i^j(\boldsymbol{x}) > \frac{1}{2}\sum_{k=1}^{N}\sum_{i=1}^{T} h_i^k(\boldsymbol{x}) \\ \text{rejection, otherwise} \end{cases}$$

- 相对多数投票法（plurality voting）

$$H(\boldsymbol{x}) = c_{\arg\max_j \sum_{i=1}^{T} h_i^j(\boldsymbol{x})}$$

$h_i^j(\boldsymbol{x}) \in \{0,1\}$，若$h_i$将样本预测为类别$j$则取值为1，其他为0

- 加权投票法（weighted voting）

$$H(\boldsymbol{x}) = c_{\arg\max_j \sum_{i=1}^{T} \boldsymbol{\omega_i} h_i^j(\boldsymbol{x})}$$

中山大學
SUN YAT-SEN UNIVERSITY

第9章 聚类

1. 聚类任务
2. 划分聚类法概论
3. 划分聚类法—k均值算法
4. 高斯混合聚类
5. 层次聚类
6. 密度聚类
7. 谱聚类

沈颖 副教授

sheny76@mail.sysu.edu.cn

机器学习-第9章（除9.2、9.4.2、9.5和9.6以外）
统计学习方法-第14章

绪论——机器学习的一般分类（无监督学习）

无监督学习：从无标注数据中学习分析模型的机器学习问题。

无标注数据是“自然”得到的数据，分析模型表示数据的类别、转换等。

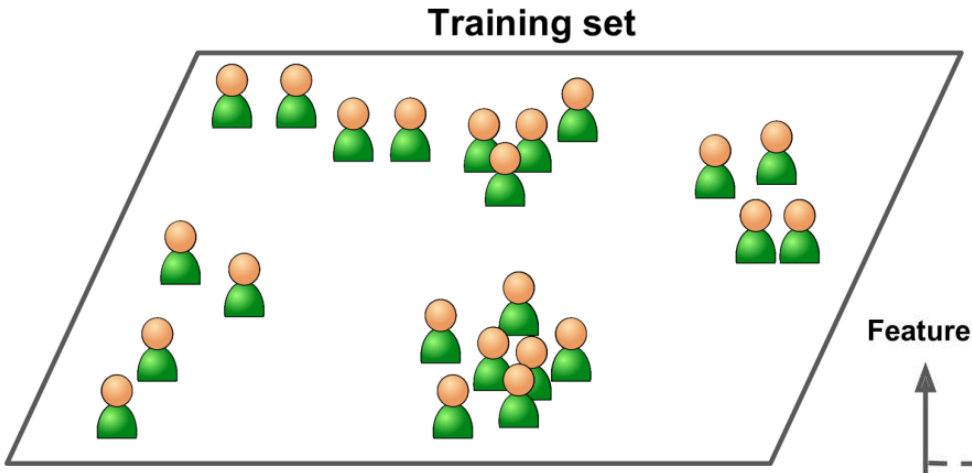


Figure 1-7. An unlabeled training set for unsupervised learning

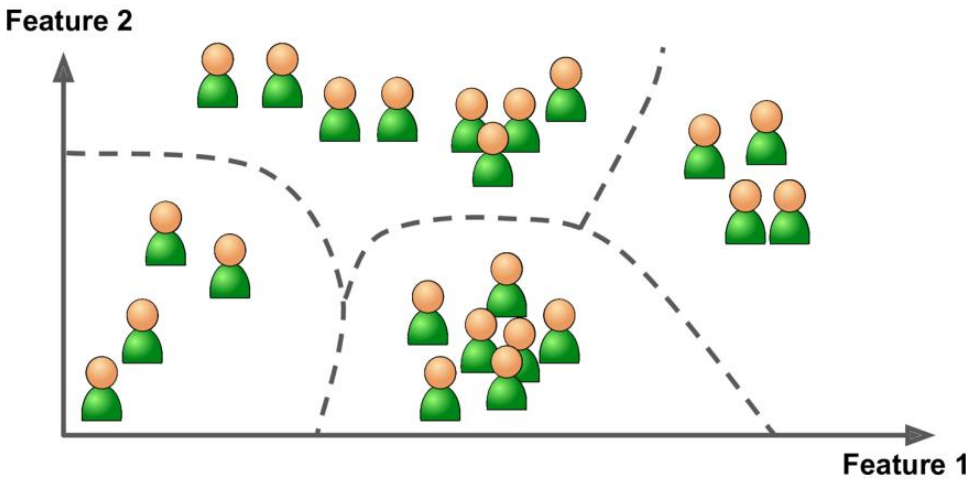


Figure 1-8. Clustering

无监督学习的基本想法：对给定数据（矩阵数据）进行某种“压缩”，从而找到数据的潜在结构。

假定损失最小的“压缩”得到的结果就是最本质的结构。

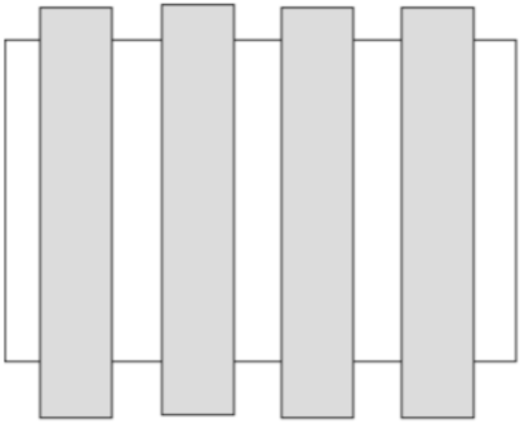
假设训练数据集由m个样本组成，每个样本是一个n维向量。训练数据可以由一个矩阵表示，每一行对应一个特征，每一列对应一个样本

$$\boldsymbol{X}=\begin{bmatrix} x_{11} & \cdots & x_{m1} \\ \vdots & & \vdots \\ x_{1n} & \cdots & x_{mn} \end{bmatrix}$$

样本（列）　特征（行）

考虑发掘数据的纵向结构

- 把相似的样本聚到同类，即对数据进行聚类



(a) 数据纵向结构

无监督学习的基本想法：对给定数据（矩阵数据）进行某种“压缩”，从而找到数据的潜在结构。

假定损失最小的“压缩”得到的结果就是最本质的结构。

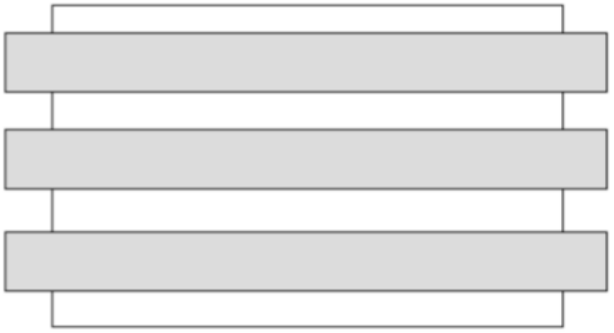
假设训练数据集由m个样本组成，每个样本是一个n维向量。训练数据可以由一个矩阵表示，每一行对应一个特征，每一列对应一个样本

样本

$$\boldsymbol{X}=\begin{bmatrix} x_{11} & \cdots & x_{m1} \\ \vdots & & \vdots \\ x_{1n} & \cdots & x_{mn} \end{bmatrix}$$ 特征

考虑发掘数据的横向结构

- 把高维空间的向量转换为低维空间的向量，即对数据进行降维



(b) 数据横向结构

典型的应用

- 聚类算法的典型应用
  - 空间数据分析
    - ✓ GPS数据
    - ✓ 手机信令数据
  - 推荐系统
    - ✓ 广告
  - 金融领域
    - ✓ 用户的交易数据

中山大學
SUN YAT-SEN UNIVERSITY

第9章 聚类

1. 聚类任务
2. 划分聚类法概论
3. 划分聚类法—k均值算法
4. 高斯混合聚类
5. 层次聚类
6. 密度聚类

机器学习-第9章（除9.2、9.4.2、9.5和9.6以外）

统计学习方法-第14章

属性类型

连续属性 vs 离散属性

- 连续属性（continuous attribute）

  在定义域上有无穷多个可能的取值：人的身高

- 离散属性（categorical attribute）

  在定义域上是有限个可能的取值：硬币只有 0.1 两种情况

二值离散型

- 二值离散型属性只有 0 和 1
  - 例如：Rain：{ Strong, Weak }
- 假设两个样本 $\boldsymbol{x}_i$, $\boldsymbol{x}_j$
  - $\boldsymbol{x}_i = (x_{i1}; x_{i2}; \ldots; x_{in})$
  - $\boldsymbol{x}_j = (x_{j1}; x_{j2}; \ldots; x_{jn})$
  - 均为 n 维特征向量，并且每维特征都是一个二值离散型数值

	Features			
	Outlook	Temperature	Humidity	Wind
样本1	Sunny	Cold	High	Weak
样本2	Sunny	Cold	High	Strong

Sunny：1　　Cloudy：0
Hot：　1　　Cold：0
High：　1　　Low：0
Strong：1　　Weak：0

二值离散型（非对称型）

- 比较两个人的患病情况：矩阵

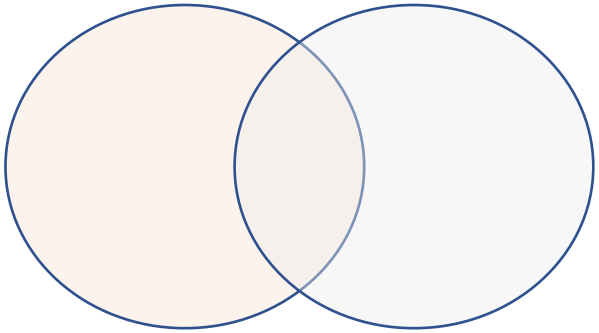
		样本2		
		0	**1**	**sum**
样本1	**0**	1 a	1 b	
	1	0 c	2 d	
	sum			4

Jaccard系数 $\frac{2}{3}$

Jaccard系数的一般形式

$$J = \frac{d}{b + c + d}$$

样本1患病情况 A

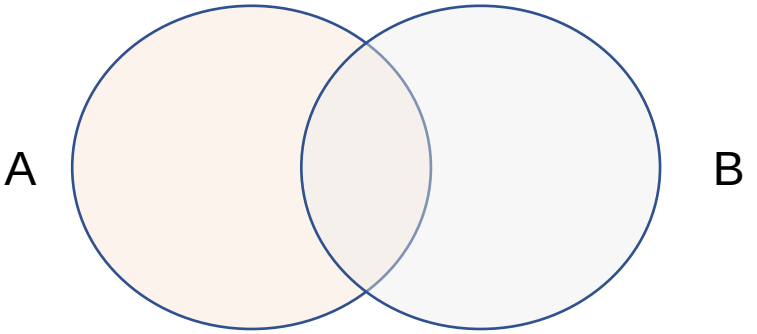


样本2患病情况 B

Jaccard系数

$$J = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Jaccard系数



Jaccard系数

$$J = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- 比较文本相似度，用于文本查重与去重
- 推荐系统
  - 图书 A—> 15人好评
  - 图书 B—> 30人好评
  - 图书 C—> 100人好评
  - 图书 A，B —> 共同好评10人
  - 图书 A，C —>共同好评15人

$$J(A,B) = \frac{10}{15 + 30 - 10} = 0.29$$

$$J(A,C) = \frac{15}{15 + 100 - 15} = 0.15$$

$$\boldsymbol{J(A,B) > J(A,C)}$$

属性类型

连续属性 vs 离散属性

- 连续属性（continuous attribute）
  在定义域上有无穷多个可能的取值
- 离散属性（categorical attribute）
  在定义域上是有限个可能的取值

有序属性 vs 无序属性

- 有序属性（ordinal attribute）
  例如定义域为{1,2,3}的离散属性，“1”与“2”比较接近、与“3”比较远，称为“有序属性”
- 无序属性（non-ordinal attribute）
  例如定义域为{飞机, 火车, 轮船}这样的离散属性，不能直接在属性值上进行计算，称为“无序属性”

- 对于无序属性，Value Difference Metric（VDM）

属性u上两个离散值a与b之间的VDM距离为

k为样本簇数

在第i个样本簇中在属性u上取值为a的样本数

$$\mathrm{VDM_p}(a,b)=\sum_{i=1}^{k}\left|\frac{m_{u,a,i}}{m_{u,a}}-\frac{m_{u,b,i}}{m_{u,b}}\right|^{p}$$

表示在属性u上取值为a的样本数

混合属性（有序和无序属性）

将闵可夫斯基距离与VDM距离相结合

$$\mathrm{MinkovDM_p}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{u=1}^{n_c} \left| x_{iu} - x_{ju} \right|^p + \sum_{u=n_c+1}^{n} \mathrm{VDM_p}(x_{iu}, x_{ju}) \right)^{\frac{1}{p}}$$

混合属性（有序和无序属性）

将闵可夫斯基距离与VDM距离相结合

$$\mathrm{MinkovDM_p}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left(\sum_{u=1}^{n_c}\left|x_{iu}-x_{ju}\right|^p+\sum_{u=n_c+1}^{n}\mathrm{VDM_p}(x_{iu},x_{ju})\right)^{\frac{1}{p}}$$

加权属性（weighted distance），以加权闵可夫斯基距离为例

$$\mathrm{dist_{wmk}}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left(\boldsymbol{\omega_1}\cdot\left|x_{i1}-x_{j1}\right|^p+\cdots+\boldsymbol{\omega_n}\cdot\left|x_{in}-x_{jn}\right|^p\right)^{\frac{1}{p}}$$

$$\omega_i\geq 0,\sum_{i=1}^{T}\omega_i=1$$

中山大學
SUN YAT-SEN UNIVERSITY

第9章 聚类

1. 聚类任务
2. 划分聚类法概论
3. 划分聚类法—k均值算法
4. 高斯混合聚类
5. 层次聚类
6. 密度聚类
7. 谱聚类

机器学习-第9章（除9.2、9.4.2、9.5和9.6以外）

统计学习方法-第14章

聚类分析方法

- 常用的聚类分析方法
  - 划分聚类法（Partition-based clustering ）
    - ✓ 以距离作为相似度度量方法
    - ✓ 将数据集划分为多个簇
    - ✓ k-means
  - 层次聚类方法（Hierarchical-based clustering）
    - ✓ 层次聚类试图在不同层次对数据集进行划分，从而形成树形的聚类结构。数据集划分既可采用“自底向上”的聚合策略，也可采用“自顶向下”的分拆策略。
    - ✓ AGNES算法（自底向上的层次聚类算法）
  - 密度聚类方法（Density-based clustering）
    - ✓ 假设聚类结构能通过样本分布的紧密程度来确定
    - ✓ DBSCAN

原型聚类

原型聚类又称为基于原型的聚类（prototype-based clustering），此类算法假设聚类结构能通过一组**原型刻画**

算法过程

通常情况下，算法先对原型进行**初始化**，再对原型进行**迭代更新求解**

著名算法

k均值算法、学习向量量化算法、高斯混合聚类算法

原型聚类——k均值算法

k均值算法

数据：$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$，$\boldsymbol{x}_j \in \mathbb{R}^n$

模型：基于原型（样本空间中具有代表性的点）的聚类结构

策略：最小化簇划分$C = \{C_1, C_2, \cdots, C_k\}$的平方误差

算法：迭代更新求解

$$E = \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in C_i} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|_2^2$$

其中$\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{x}$是簇$C_i$的均值向量

E值在一定程度上刻画了簇内样本围绕簇均值向量的紧密程度，E值越小，则簇内样本相似度越高

原型聚类——k均值算法

k均值算法

数据：$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m\}$，$\boldsymbol{x}_j \in \mathbb{R}^n$

模型：基于原型（样本空间中具有代表性的点）的聚类结构

策略：最小化簇划分$C = \{C_1, C_2, \cdots, C_k\}$的平方误差

$$E = \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in C_i} \|\boldsymbol{x} - \boldsymbol{\mu}_i\|_2^2$$

算法：迭代更新求解

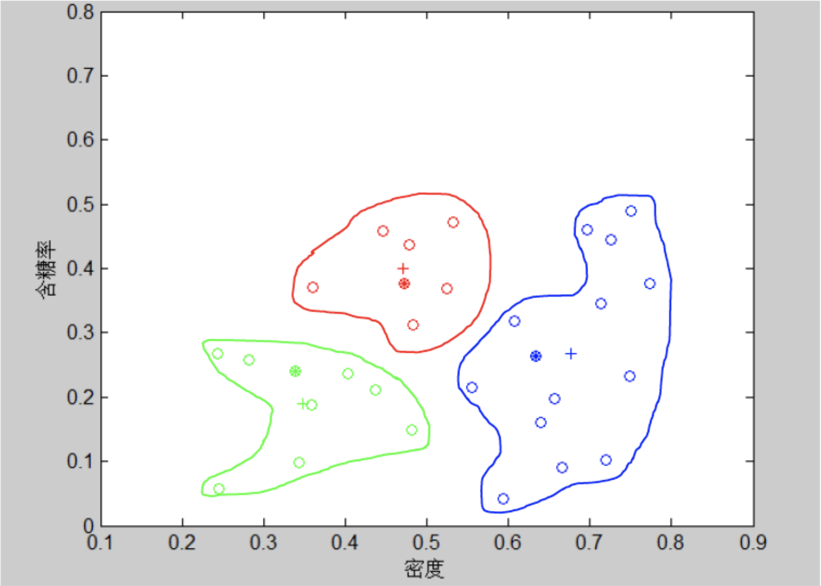
1. 选择k个类的中心（均值向量），将样本逐个派到与其最近的中心的类中，得到一个聚类结果；
2. 更新每个类的样本的均值向量，作为类的新的中心；
3. 重复以上步骤，直到收敛为止（当前均值向量均未更新）。

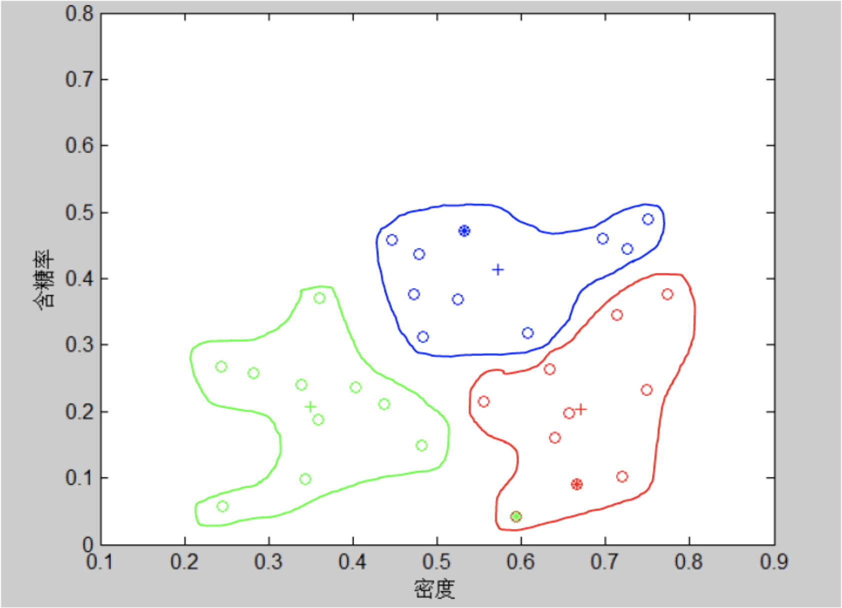
K-means总结

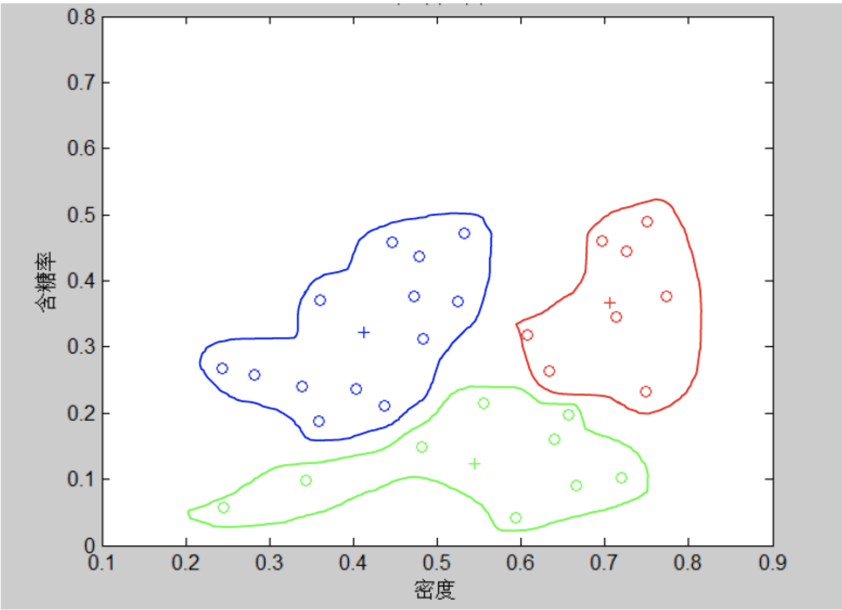
- K-均值聚类算法是一种典型的**动态聚类方法**，具有如下三个要点：
  - (1) 选择欧氏距离度量作为样本间的相似性度量。
  - (2) 采用最大似然估计或最小均方误差作为评价聚类的准则函数。
  - (3) 给定某个初始分类，然后采用迭代算法寻找准则函数的极值。
- **优点：**
  - 是解决聚类问题的一种经典算法，简单、快速。
  - 对处理大数据集，该算法仍可保持其高效率。
  - 对于密集簇，聚类效果很好。
- **缺点：**
  - 必须事先给定簇的个数，且对初始值敏感。
  - 不适合于发现非凸曲面的簇以及大小相差很大的簇。
  - 对噪声、孤立数据点、野点很敏感。

K-means总结

不同初始值可能会得到不同的聚类结果







中山大學
SUN YAT-SEN UNIVERSITY

第9章 聚类

1. 聚类任务
2. 划分聚类法概论
3. 划分聚类法—k均值算法
4. 高斯混合聚类
5. 层次聚类
6. 密度聚类
7. 谱聚类

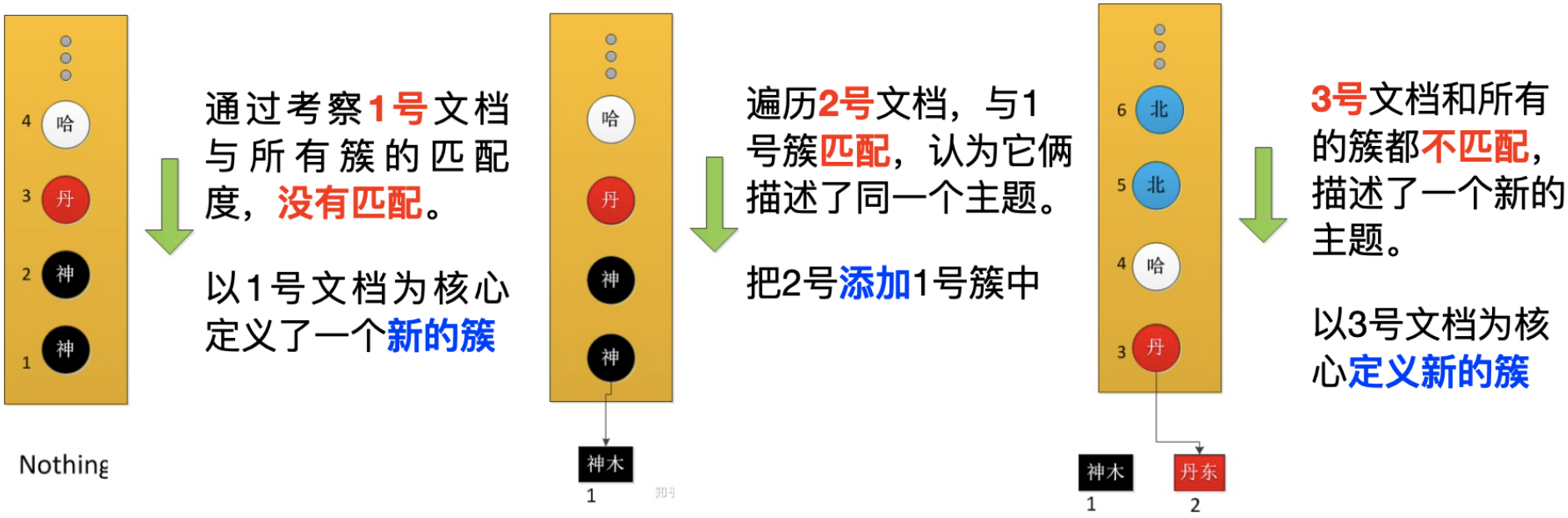
机器学习-第9章（除9.2、9.4.2、9.5和9.6以外）
统计学习方法-第14章

聚类分析方法

- 常用的聚类分析方法
  - 划分聚类法（Partition-based clustering ）
    - ✓ 以距离作为相似度度量方法
    - ✓ 将数据集划分为多个簇
    - ✓ k-means
  - 层次聚类方法（Hierarchical-based clustering）
    - ✓ 层次聚类试图在不同层次对数据集进行划分，从而形成树形的聚类结构。数据集划分既可采用“自底向上”的聚合策略，也可采用“自顶向下”的分拆策略。
    - ✓ AGNES算法（自底向上的层次聚类算法）
  - 密度聚类方法（Density-based clustering）
    - ✓ 假设聚类结构能通过样本分布的紧密程度来确定
    - ✓ DBSCAN

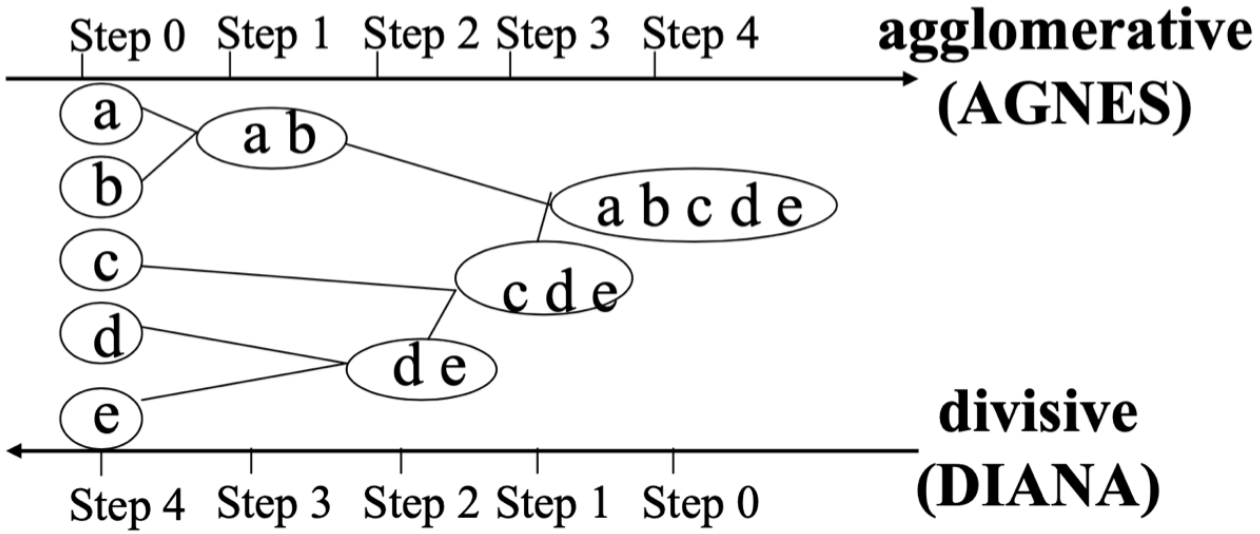
1. 单遍聚类

- “Single-pass”—“只遍历一回”
- 计算速度非常快
- “匹配”与否的判断——相似度和阈值



2. 层次聚类

层次聚类方法依据一种**层次架构**将数据逐层进行**聚合**或**分裂**，最终将数据对象组织成一棵**聚类树状**的结构。



中山大學
SUN YAT-SEN UNIVERSITY

第9章 聚类

1. 聚类任务
2. 划分聚类法概论
3. 划分聚类法—k均值算法
4. 高斯混合聚类
5. 层次聚类
6. 密度聚类
7. 谱聚类

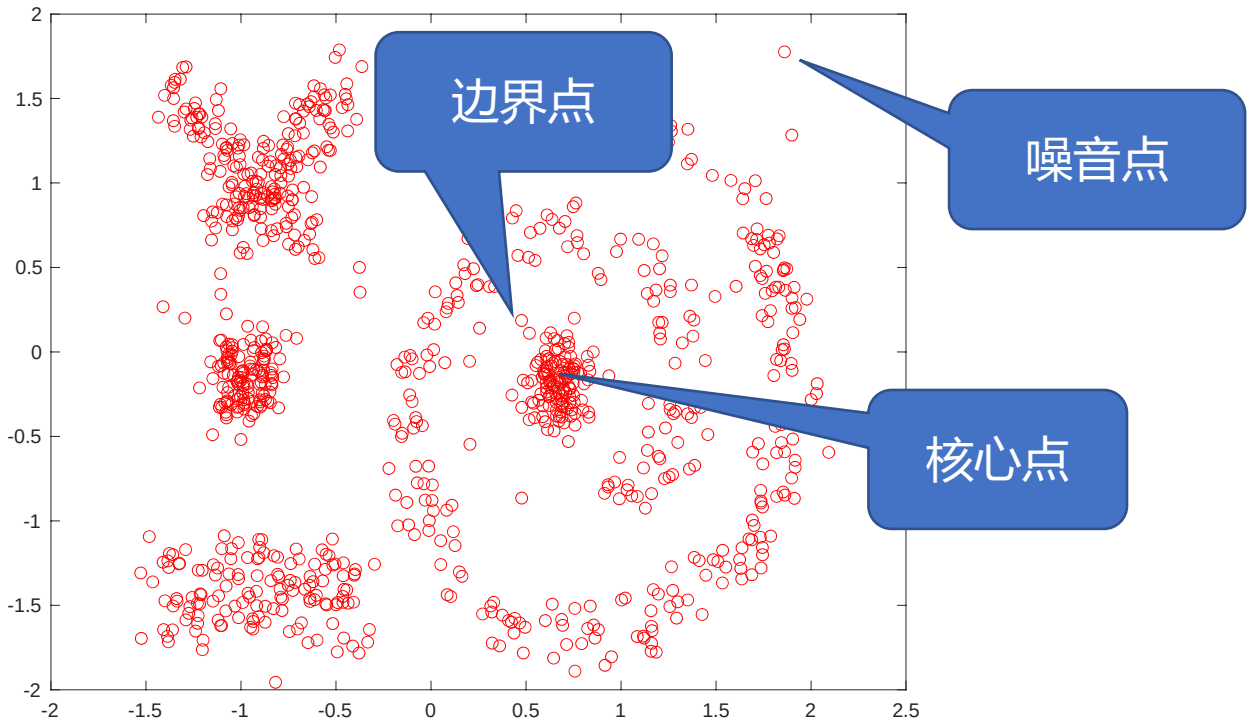
机器学习-第9章（除9.2、9.4.2、9.5和9.6以外）
统计学习方法-第14章

聚类分析方法

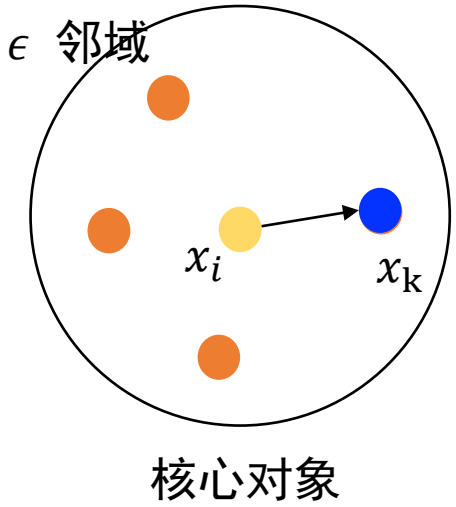
- 常用的聚类分析方法
  - 划分聚类法（Partition-based clustering ）
    - ✓ 以距离作为相似度度量方法
    - ✓ 将数据集划分为多个簇
    - ✓ k-means
  - 层次聚类方法（Hierarchical-based clustering）
    - ✓ 层次聚类试图在不同层次对数据集进行划分，从而形成树形的聚类结构。数据集划分既可采用“自底向上”的聚合策略，也可采用“自顶向下”的分拆策略。
    - ✓ AGNES算法（自底向上的层次聚类算法）
  - 密度聚类方法（Density-based clustering）
    - ✓ 假设聚类结构能通过样本分布的紧密程度来确定
    - ✓ DBSCAN

密度聚类的定义

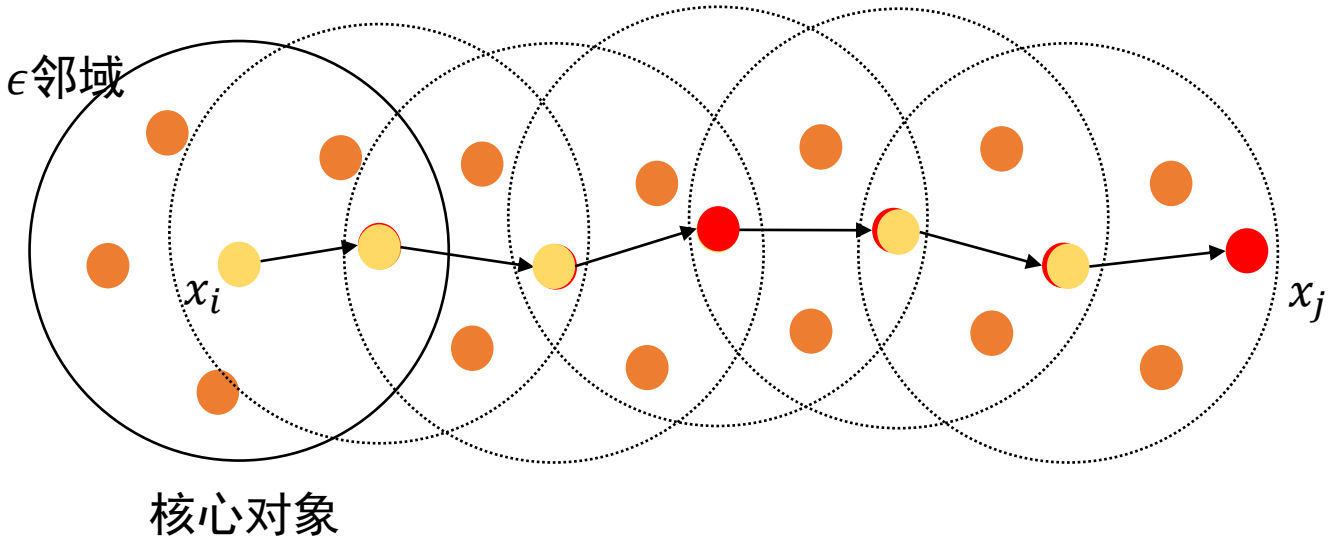
- 密度：在一定区域范围内，点的数量超过一个给定阈值
- 核心点：高密度点
- 边界点：低密度但邻近核心点
- 噪音点：既不是核心点也不是边界点



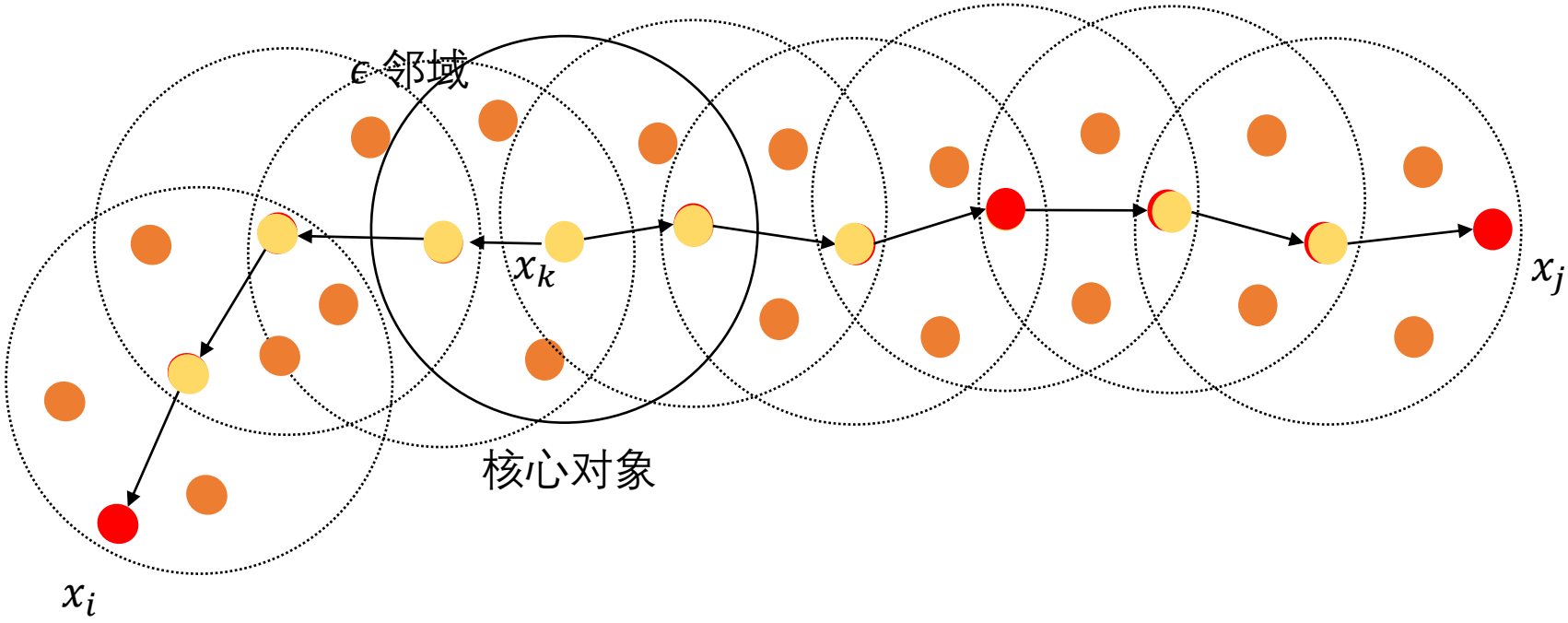
- **密度直达**：若样本 x_k 位于样本 x_i 的 ϵ 邻域中，且 x_i 是一个核心对象，则称样本 x_k 由 x_i 密度直达；



- 密度直达：若样本 x_k 位于样本 x_i 的 ϵ 邻域中，且 x_i是一个核心对象，则称样本 x_k 由 x_i密度直达；
- 密度可达：对样本 x_i 与 x_j ，若存在样本序列 $p_1, p_2, \ldots, p_n$，其中 $p_1 = x_i$,$p_n = x_j$，且 p_{i+1} 由 p_i 密度直达，则该两样本密度可达。



- **密度相连**：对样本 x_i 与 x_j，若存在样本 x_k 使得两样本均由 x_k 密度可达，则称该两样本密度相连。



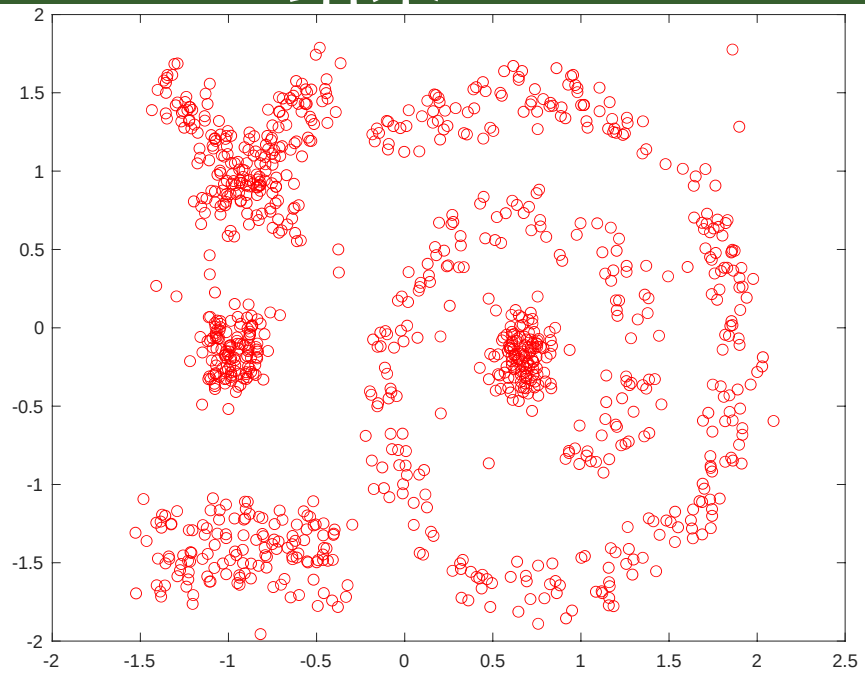
输入： 样本集$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
邻域参数$(\epsilon, MinPts)$.

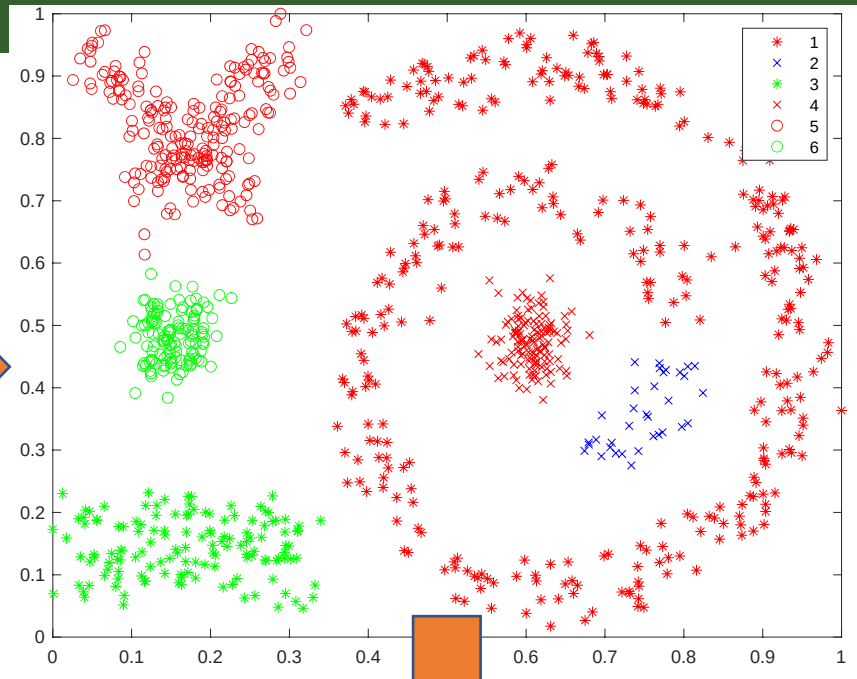
过程：

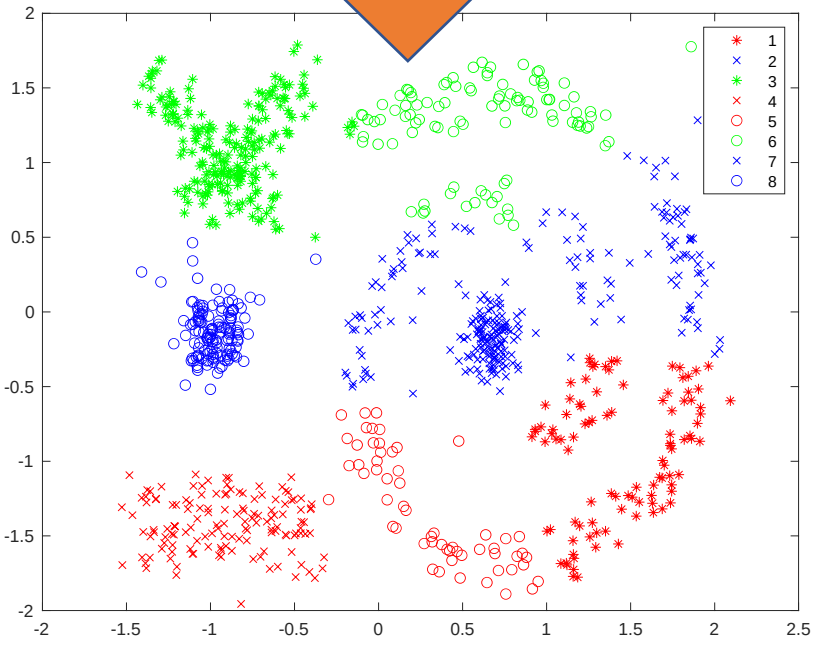
初始化核心对象集合：$\Omega = \emptyset$
for $j = 1, \ldots, m$ **do**
  确定样本$\boldsymbol{x}_j$的ϵ-邻域$N_\epsilon(\boldsymbol{x}_j)$;
  if $|N_\epsilon(\boldsymbol{x}_j)| \geq MinPts$ **then**
    将样本$\boldsymbol{x}_j$加入核心对象集合：$\Omega = \Omega \bigcup \{\boldsymbol{x}_j\}$
  end if
end for
初始化聚类簇数：$k = 0$
初始化未访问样本集合：$\Gamma = D$
while $\Omega \neq \emptyset$ **do**
  记录当前未访问样本集合：$\Gamma_{\text{old}} = \Gamma$;
  随机选取一个核心对象$\boldsymbol{o} \in \Omega$, 初始化队列 $Q =< \boldsymbol{o} >$;
  $\Gamma = \Gamma \setminus \{\boldsymbol{o}\}$;
  while $Q \neq \emptyset$ **do**
    取出队列Q中的首个样本$\boldsymbol{q}$;
    if $|N_\epsilon(\boldsymbol{q})| \geq MinPts$ **then**
      令$\Delta = N_\epsilon(\boldsymbol{q}) \bigcap \Gamma$;
      将Δ中的样本加入队列Q;
      $\Gamma = \Gamma \setminus \Delta$;
    end if
  end while
  $k = k + 1$, 生成聚类簇$C_k = \Gamma_{\text{old}} \setminus \Gamma$;
  $\Omega = \Omega \setminus C_k$
end while
return 簇划分结果

输出： 簇划分$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

DBSCAN结果


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.6
0.7
0.8
0.9
1
1
2
3
4
5
6


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
1
2
3
4
5
6
7
8

中山大學
SUN YAT-SEN UNIVERSITY

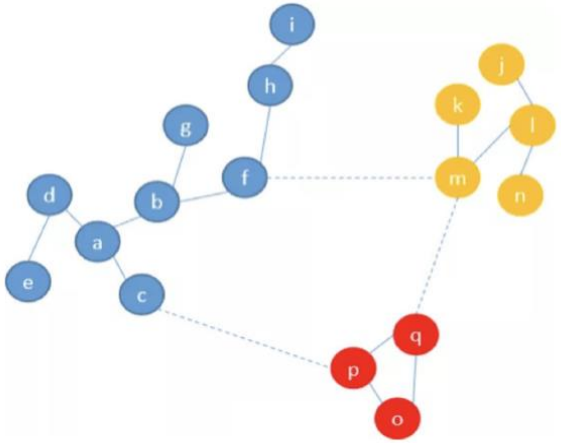
第9章 聚类

1. 聚类任务
2. 划分聚类法概论
3. 划分聚类法—k均值算法
4. 高斯混合聚类
5. 层次聚类
6. 密度聚类
7. 谱聚类

机器学习-第9章（除9.2、9.4.2、9.5和9.6以外）
统计学习方法-第14章

谱聚类

- 从图切割的角度，聚类就是要找到一种合理的分割图的方法，**分割后能形成若干个子图**。连接不同子图的边的权重尽可能小，子图内部边权重尽可能大。
- 谱聚类算法建立在**图论中的谱图理论**基础之上，其本质是将聚类问题转化为一个**图上的关于顶点划分的最优问题**。
- 谱聚类算法建立在**点对亲和性**基础之上，理论上能对**任意分布形状**的样本空间进行聚类。



中山大學
SUN YAT-SEN UNIVERSITY

第10章 降维

1. k近邻学习
2. 降维

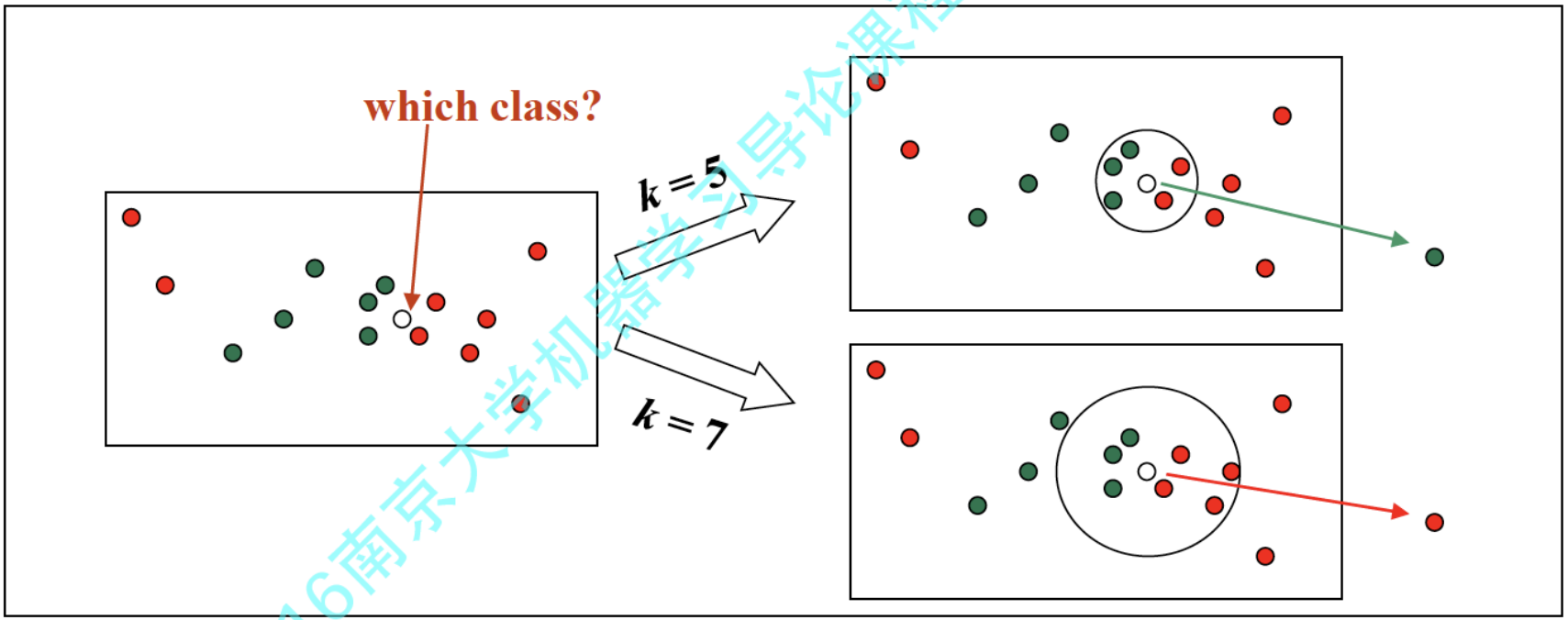
沈颖 副教授
sheny76@mail.sysu.edu.cn

机器学习-第10章（10.1和10.3）
统计学习方法-第3、16章全部

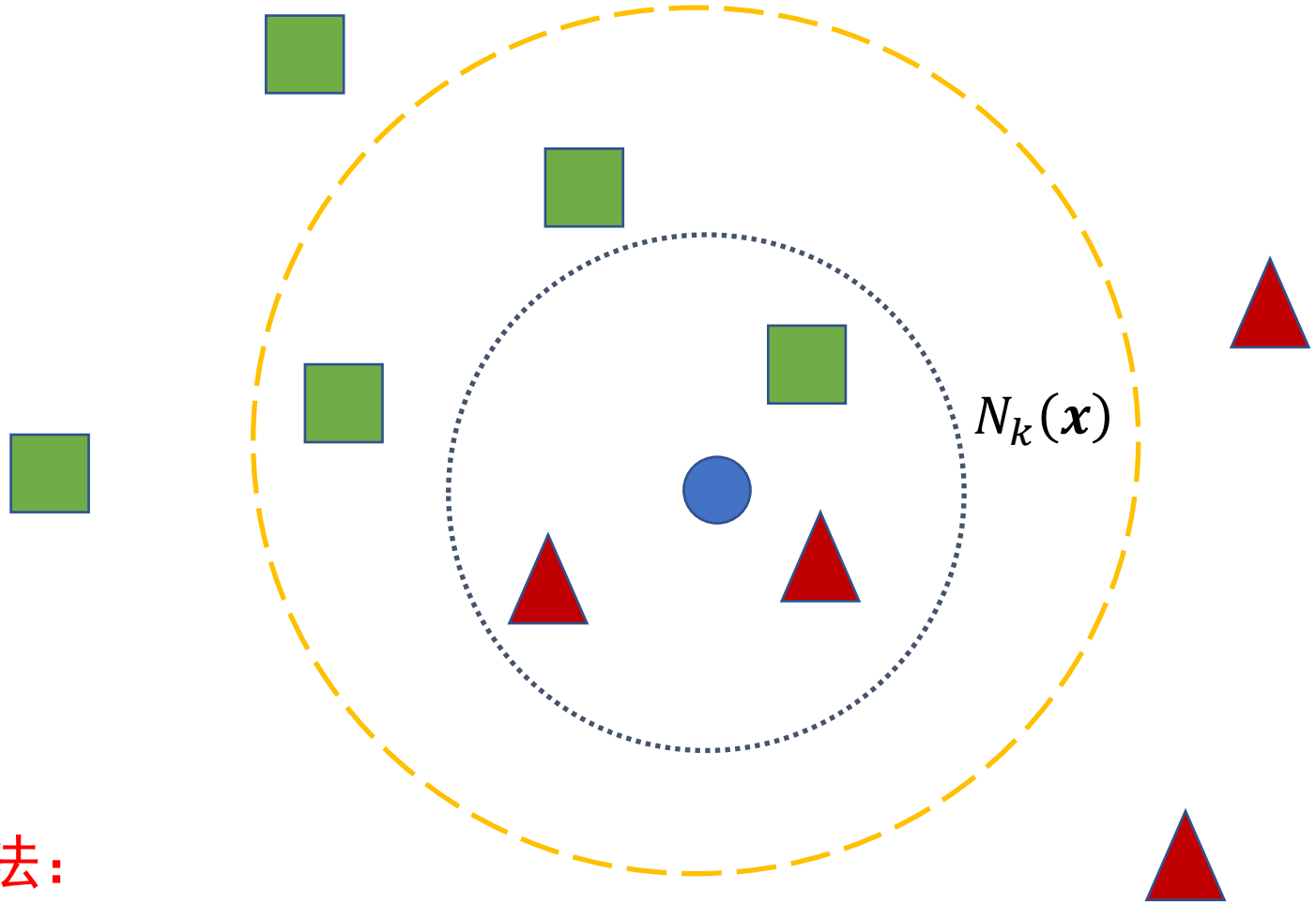
k 近邻 (k-Nearest Neighbor, kNN)

懒惰学习 (lazy learning) 的代表

此类学习技术在训练阶段仅仅是把样本保存起来，训练时间开销为零，待收到测试样本后再进行处理



kNN基本想法



算法：

- 根据给定的距离度量，在训练集D中找出与$\boldsymbol{x}$最邻近的k个点，涵盖这k个点的邻域记作$N_k(\boldsymbol{x})$
- 在$N_k(\boldsymbol{x})$中根据分类决策规则（如多数表决）决定$\boldsymbol{x}$的类别y

中山大學
SUN YAT-SEN UNIVERSITY

第10章 降维

1. k近邻学习

2. 降维

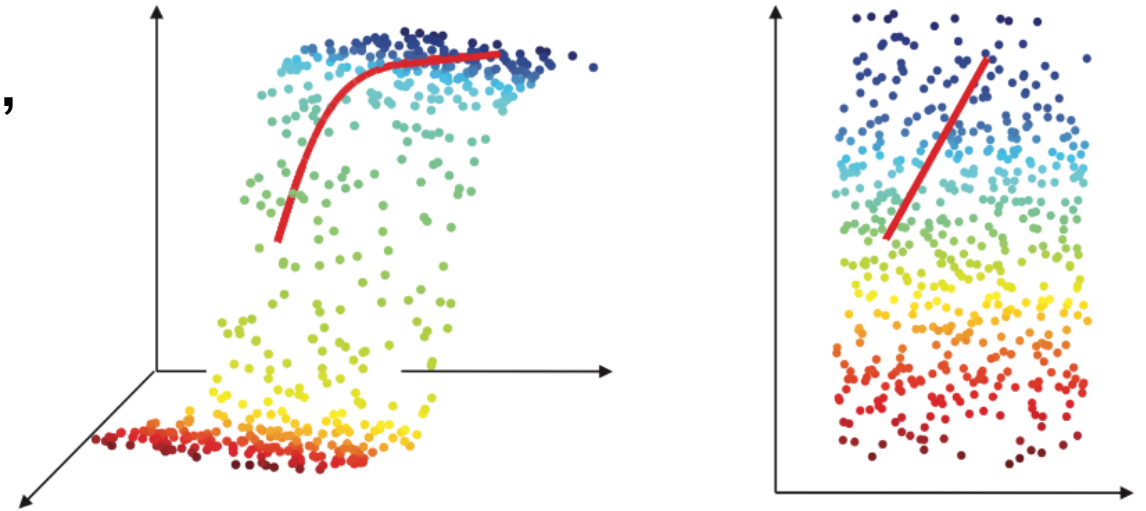
机器学习-第10章（10.1和10.3）

统计学习方法-第3、16章全部

缓解维数灾难的一个重要途径是降维

- 即通过某种数学变换，将原始高维属性空间转变为一个低维“子空间”（subspace），在这个子空间中样本密度大幅度提高，距离计算也变得更为容易。

数据样本虽然是高维的，但与学习任务密切相关的也许仅是某个低维分布，即高维空间中的一个低维“嵌入”(embedding)

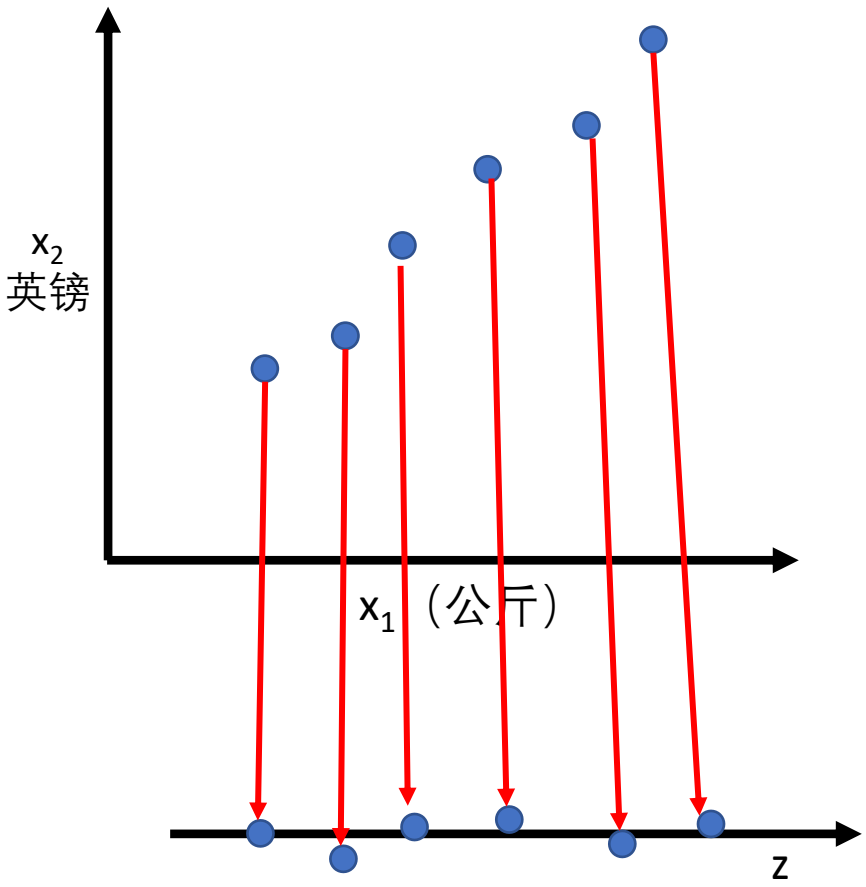


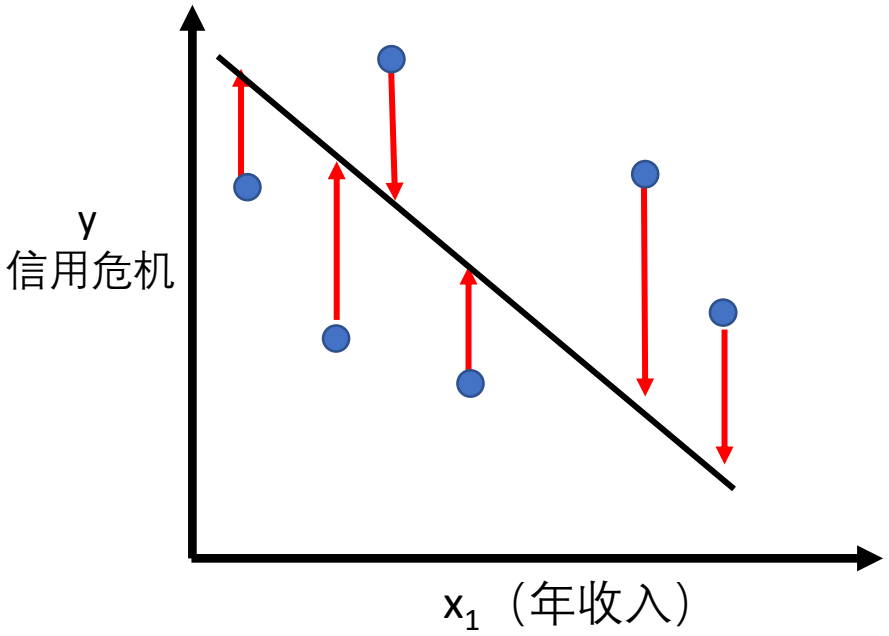
(a) 三维空间中观察到的样本点　　(b) 二维空间中的曲面

图 10.2　低维嵌入示意图

降维（Dimensionality Reduction）

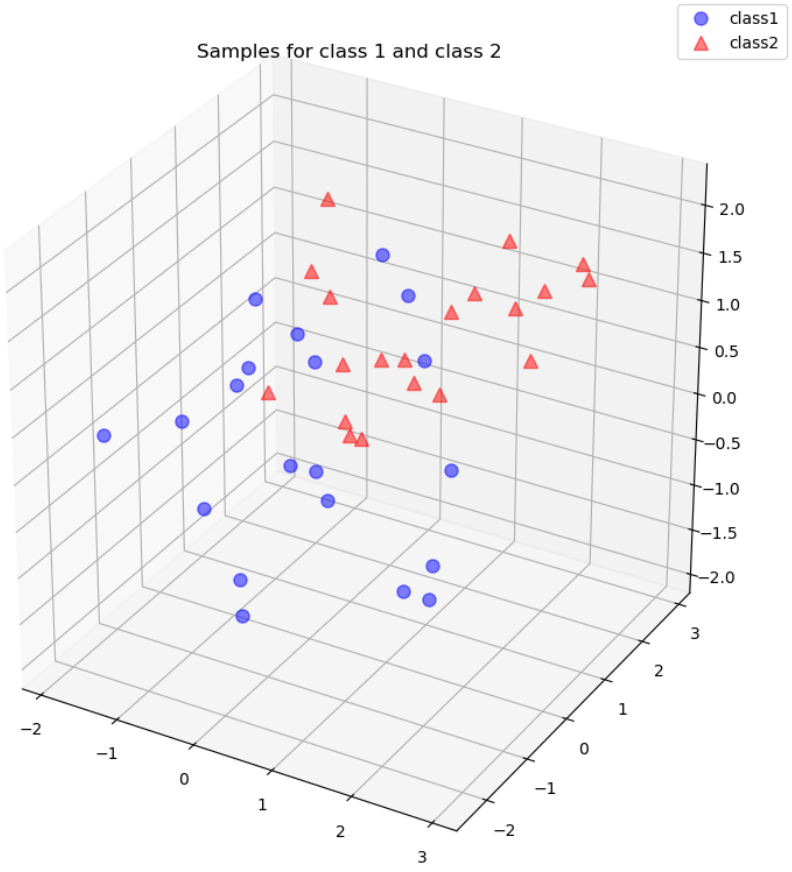
- 数据压缩、高效的计算
- 数据隐私保护

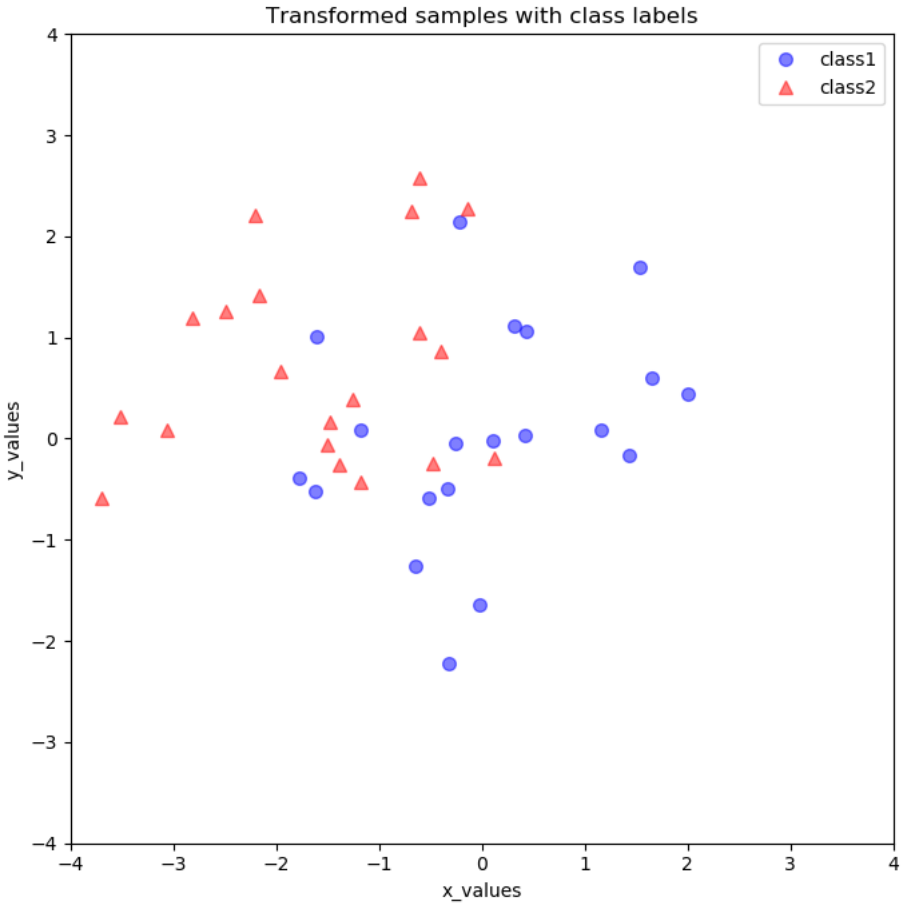




线性回归

多维


Samples for class 1 and class 2
class1
class2


Transformed samples with class labels
class1
class2
y_values
x_values

主成分问题建模与求解

主成分分析

序号	原始文档	降维后的文档	类别
$train\_d_1$	北京 理工 大学 计算机 专业 创建 于 1958 年 是 中国 最早 设立 计算机 专业 的 高校 之一	大学 计算机 计算机 高校	教育
$train\_d_2$	北京 理工 大学 学子 在 第四 届 中国 计算机 博弈 锦标赛 中 夺冠	大学 计算机	教育
$train\_d_3$	北京 理工 大学 体育馆 是 2008 年 中国 北京 奥林匹克 运动会 的 排球 预赛 场地	大学 运动会	体育
$train\_d_4$	第五届 东亚 运动会 中国 军团 奖牌 总数 创 新高 男女 排球 双双 夺冠	运动会	体育
$test\_d_1$	北京 理工 大学 是 理工 为主 工理文 协调 发展 的 全国 重点 大学	大学 大学	
$test\_d_2$	复旦 大学 排球 队 获得 本届 大学生 运动会 排球 比赛 冠军	大学 排球 运动会	

主成分分析

由线性相关变量表示的观测数据

利用正交变换转换

少数几个由线性无关变量表示的数据

线性无关的变量称为主成分

主成分的个数通常小于原始变量的个数，所以主成分分析属于降维方法

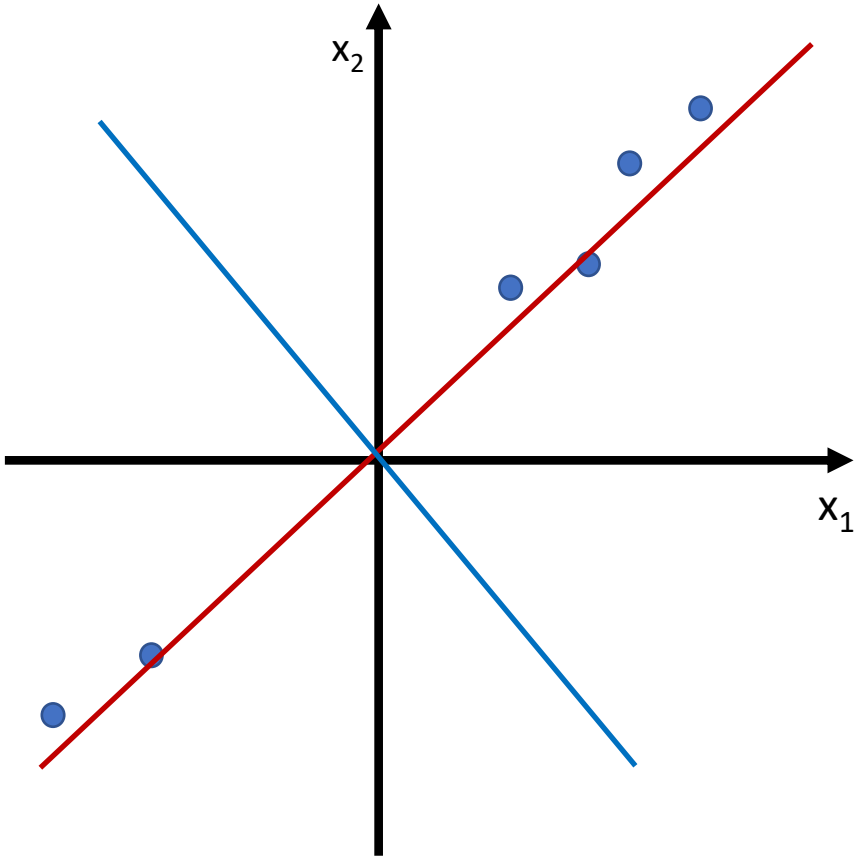
主成分分析（Principal Component Analysis, PCA）

主成分分析是最常用的一种降维方法.

对于正交属性空间中的样本点，如何用一个超平面（直线的高维推广）对所有样本进行恰当的表达？

- 若存在这样的超平面，那么它大概应具有这样的性质
  - 最近重构性：样本点到这个超平面的距离都足够近
  - 最大可分性：样本点在这个超平面上的投影能尽可能分开

- 若存在这样的超平面，那么它大概应具有这样的性质
  - 最近重构性：样本点到这个超平面的距离都足够近
  - 最大可分性：样本点在这个超平面上的投影能尽可能分开



中山大學
SUN YAT-SEN UNIVERSITY

第10章 强化学习

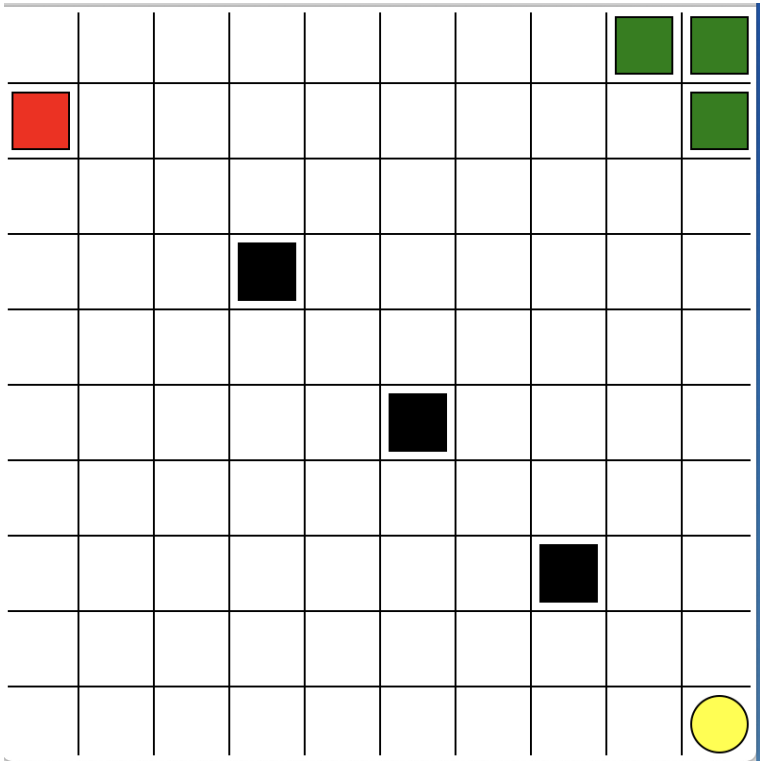
1. 强化学习概述
2. DRL基于价值的学习
3. DRL基于策略的学习

沈颖 副教授
sheny76@mail.sysu.edu.cn

强化学习基本元素

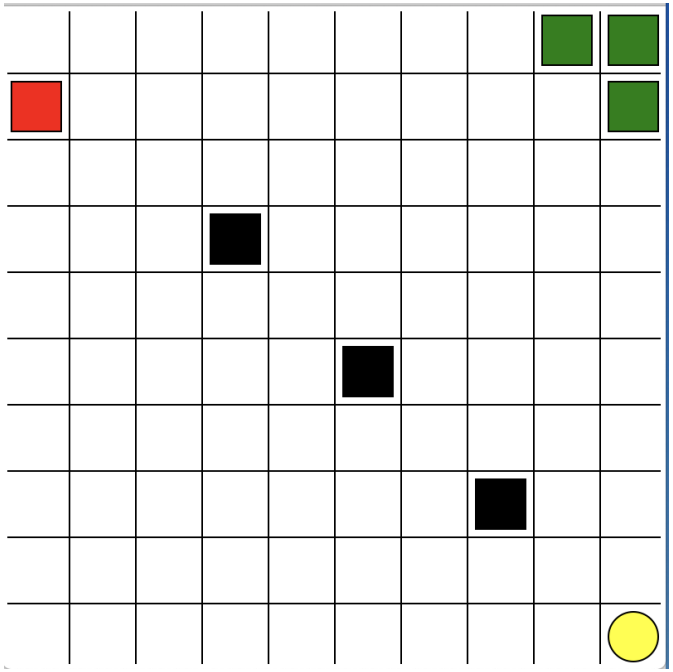
- History: 是observations, action, rewards组成的序列 $H_t = O_1, R_1, A_1, O_2, R_2, A_2, \ldots, O_{t-1}, R_{t-1}, A_{t-1}, O_t, R_t$
  - 时间 t 之前的所有可观察变量

迷宫游戏



强化学习基本元素

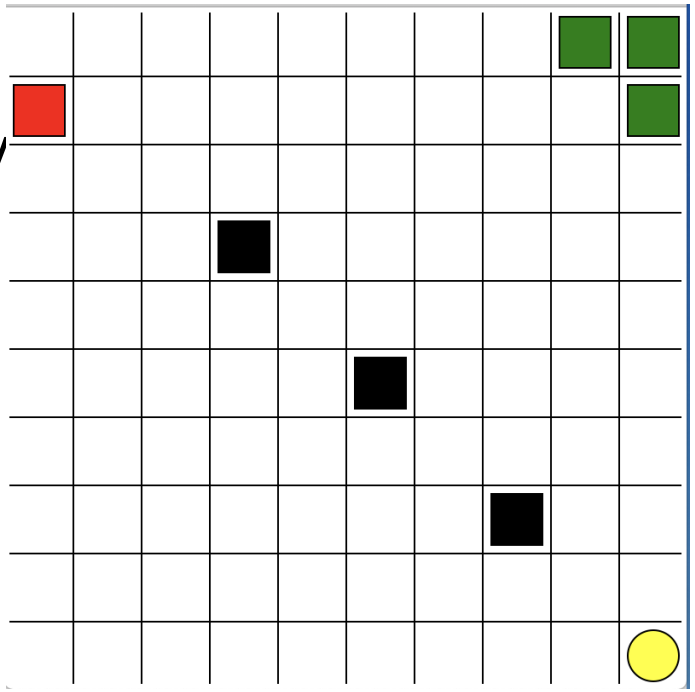
- History: 是observations, action, rewards组成的序列 $H_t = O_1, R_1, A_1, O_2, R_2, A_2, \ldots, O_{t-1}, R_{t-1}, A_{t-1}, O_t, R_t$
  - 时间 t 之前的所有可观察变量
- 接下来会发生什么取决于history:
  - Agent选择动作（actions）
  - 环境选择 observations/rewards



强化学习基本元素

- Policy：学习主体agent在给定时间的行为方式
  - 从state到action的一种映射
  - 确定策略（Deterministic policy）

$$a = \pi(s)$$

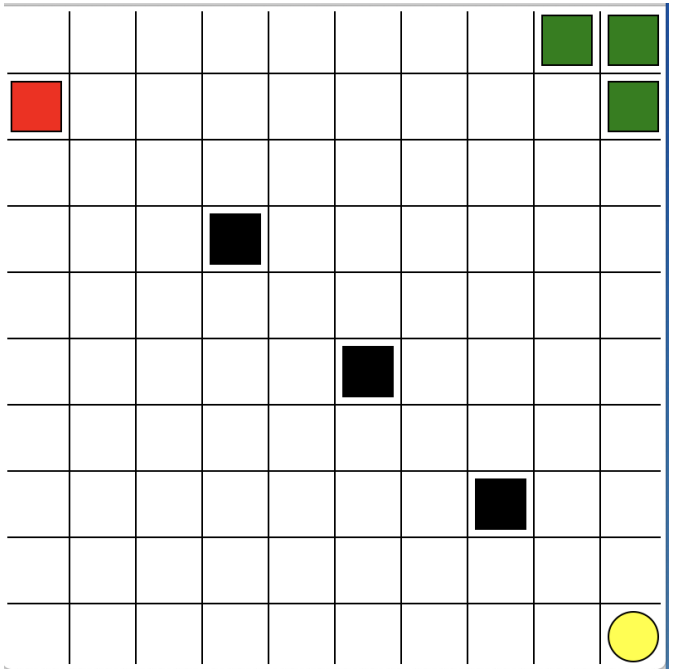


  - 随机策略（Stochastic policy）

$$\pi(a|s) = P(A_t = a|S_t = s)$$

强化学习基本元素

- 值函数（Value function）
  - 值函数是对未来累积奖赏的预测
  - 用于评价状态（states）的好坏
  - 因此在动作之间进行选择



状态值（State Value）：从长远来看，什么是好的状态，即累积奖赏

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\ldots | S_t = s]$$

动作值（Action Value）：
从长远来看，在特定状态下什么是好的动作

$$Q_\pi(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots\ldots | S_t = s, A_t = a]$$

强化学习代理（Agents）分类

- 基于价值的学习Value Based
  - 没有策略 No Policy
  - 有值函数 Value
- 基于策略的学习Policy Based
  - 有策略 Policy
  - 没有值函数 No Value Function
- Actor Critic
  - 有策略 Policy
  - 有值函数 Value Function

强化学习代理（Agents）分类

- 基于模型的强化学习 Model based RL
  - 依据策略或值函数
  - 构建关于环境的模型

- 无模型的强化学习 Model-free RL
  - 依据策略或值函数
  - 不构建关于环境的模型