第一章：简介

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

课程讲师与助教

- 主讲人：　连德富 特任教授 | 博士生导师
  @大数据学院 | 大数据分析与应用安徽省重点实验室
- 办公室：　西区科技楼 东楼715室
- 邮箱：　liandefu@ustc.edu.cn
- 研究方向：数据挖掘、机器学习、深度学习

助教

张永停

- 硕士生@信息学院
- zytabcd@mail.ustc.edu.cn
- 17398385209

顾言午

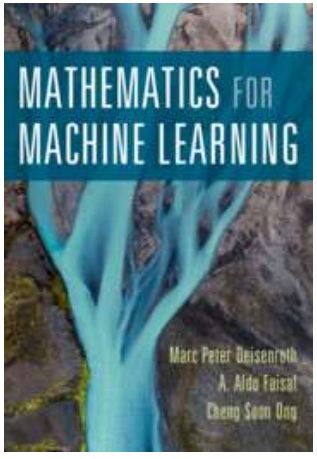
- 本科生@大数据学院
- yanwugu@mail.ustc.edu.cn
- 13017762508

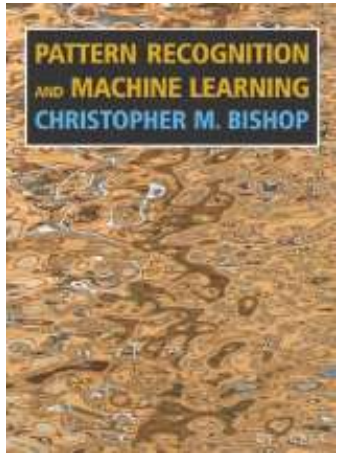
吴本伟

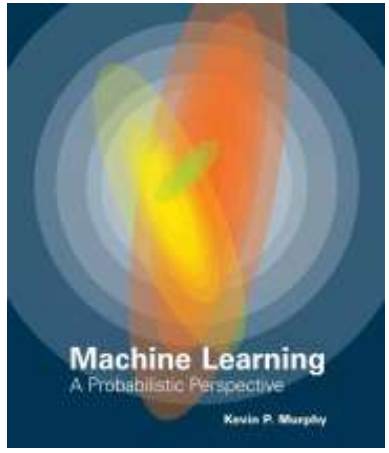
- 博士生@计算机学院
- wubenwei@mail.ustc.edu.cn
- 15055181994

参考资料

- 教科书
  - 周志华《机器学习》清华大学出版社
- 参考书
  - 李航 《统计机器学习》清华大学出版社
  - Christopher M. Bishop 《Pattern Recognition and Machine Learning》 Springer
  - Kevin P. Murphy 《Machine Learning: A probabilistic perspective》 MIT Press
  - Marc Peter Deisenroth et al.《 Mathematics for Machine Learning 》 Cambridge







阅读材料

- 机器学习领域的重要会议
  - ICML (International Conference on Machine Learning)
  - NeurIPS (Conference on Neural Information Processing Systems)
  - ICLR (International Conference on Learning Representations)
  - COLT (Conference on Learning Theory)
  - ECML(European Conference on Machine Learning)
  - ACML (Asian Conference on Machine Learning)
  - CCML (中国机器学习大会)
- 机器学习领域的重要期刊
  - Journal of Machine Learning Research (JMLR)
  - Machine Learning Journal (MLJ)
  - IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI)

课程安排与考核方式

笔试	期中占比10% 期末占比30%
作业	占比25% 大概14次作业，平均每次4-5道题，每2次提交一次作业
上机	占比25% 单独完成6次上机小实验
点名	占比5% 10次点名，每次10人
建议	占比5% 课程结束后提出有信息量的建议

总成绩=期末（30%）+作业（25%）+上机（25%）+期中（10%）+点名（5%）+建议（5%）

课程安排与考核方式

笔试	期中 期末
作业	占比 大作业
上机	占比 单独
点名	占比 10次
建议	占比 课程

群名称：2022机器学习概论群

群　号：632948557

总成绩=期末(25%)
+期中(5%)

课程目标

- 理解机器学习的基本概念
- 理解机器学习基础理论，熟练掌握机器学习算法原理和工程实现
- 掌握机器学习在实际问题中的应用
  - 特征抽取与预处理
  - 模型选择与调参
  - 实验方法

人工智能

- 人工智能（artificial intelligence，AI）就是让机器具有人类的智能。
  - “计算机控制”+“智能行为”

人工智能就是要让机器的行为看起来就像是人所表现出的智能行为一样。

John McCarthy（1927-2011）

- 人工智能这个学科的诞生有着明确的标志性事件，就是1956年的达特茅斯（Dartmouth）会议。在这次会议上，“人工智能”被提出并作为本研究领域的名称。

弱人工智能 VS 强人工智能

- 弱人工智能：限制领域人工智能（Narrow AI）或应用型人工智能（Applied AI），指的是专注于且只能解决特定领域问题的人工智能

- 强人工智能：又称通用人工智能（Artificial General Intelligence）或完全人工智能（Full AI），指的是可以胜任人类所有工作的人工智能。

  强人工智能具备以下能力

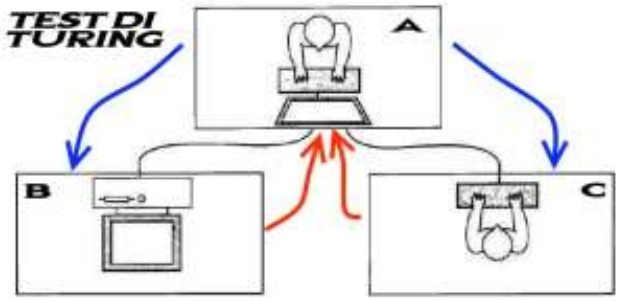
  1. 存在不确定性因素时进行推理，使用策略，解决问题，制定决策的能力
  2. 知识表示的能力，包括常识性知识的表示能力
  3. 规划能力
  4. 学习能力
  5. 使用自然语言进行交流沟通的能力
  6. 将上述能力整合起来实现既定目标的能力

图灵测试

“一个人在不接触对方的情况下，通过一种特殊的方式，和对方进行一系列的问答。如果在相当长时间内，他无法根据这些问题判断对方是人还是计算机，那么就可以认为这个计算机是智能的”。

---Alan Turing [1950]
《Computing Machinery and Intelligence》

Alan Turing



人工智能的三个阶段

- 让机器具有人类的智能

感知智能

- 基于视觉、听觉及各种传感器的信息处理
- 表现为“能听会说、能看会认”
- 机器视觉、语音识别、文字识别

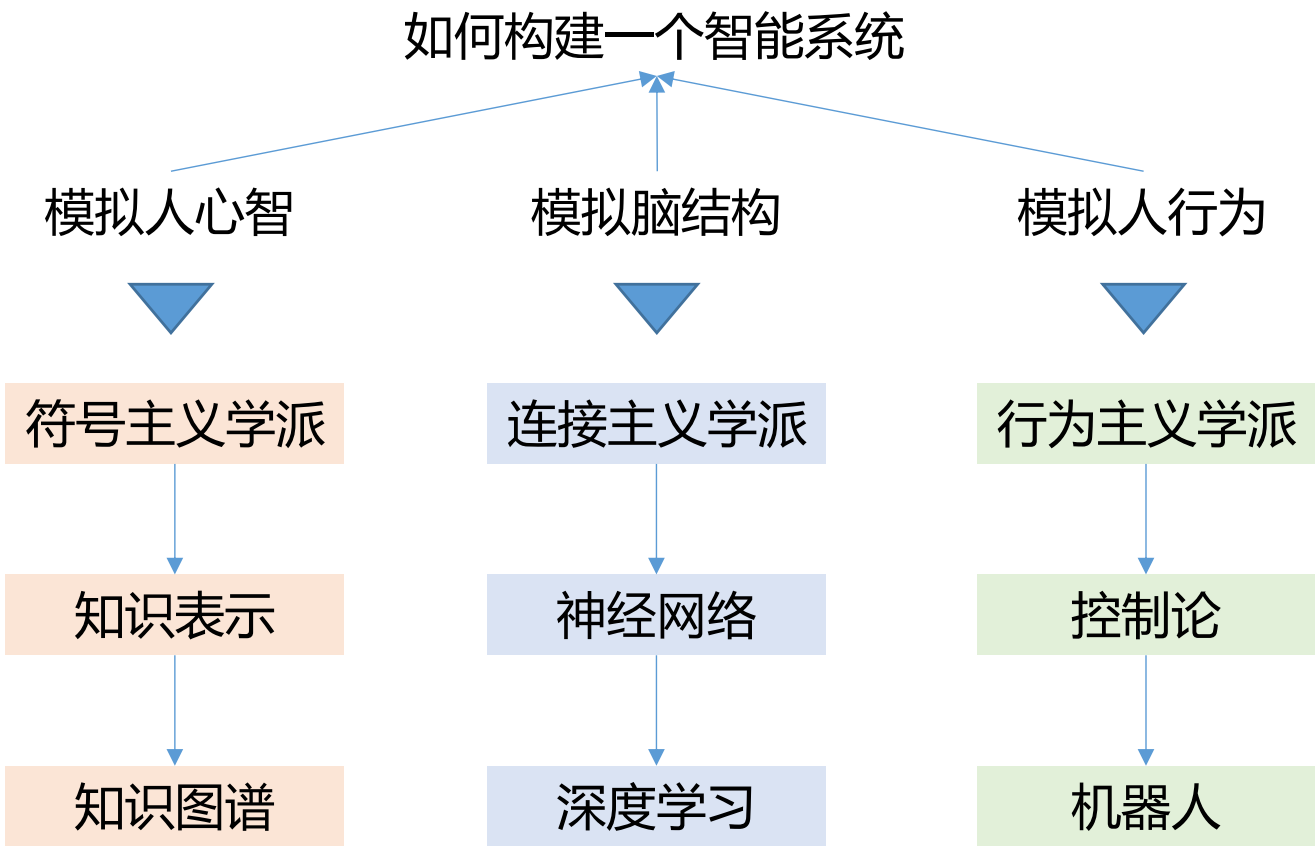
认知智能

- 更高层的语义处理、推理、规划、记忆、学习
- 表现为“能理解、会思考、有认知”
- 知识表示、模式识别、机器学习、自然语言处理

决策智能

- 复杂问题下，提升人机信任度，增强人类与智能系统交互协作智能
- 表现为“自主性”
- 规划、数据挖掘、强化学习

人工智能流派


如何构建一个智能系统
模拟人心智
模拟脑结构
模拟人行为
符号主义学派
连接主义学派
行为主义学派
知识表示
神经网络
控制论
知识图谱
深度学习
机器人

什么是机器学习

Learning is any process by which a system improves performance from experience.

学习是系统从经验中提高性能的任何过程

美国著名学者、计算机科学家和心理学家

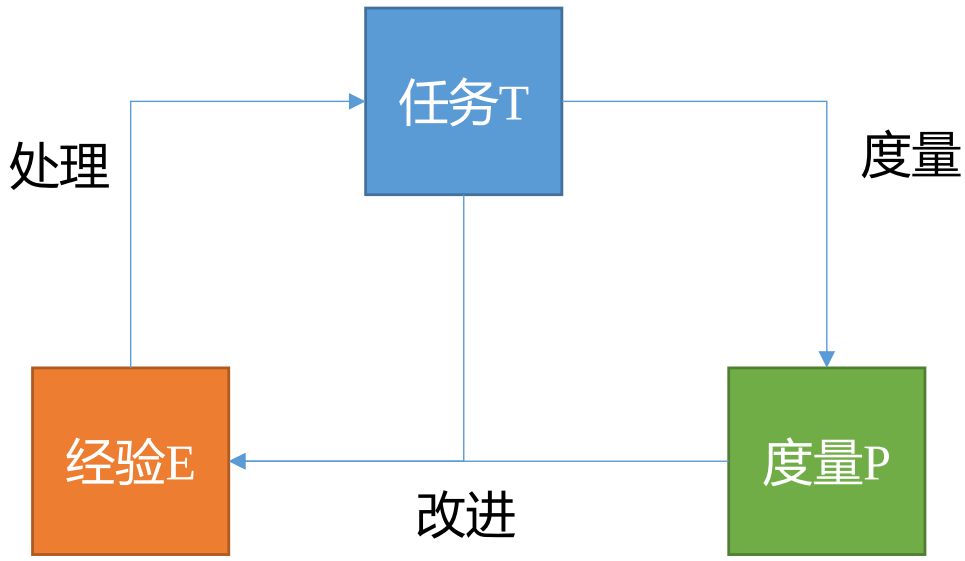
——Herbert Simon (司马贺)
Carnegie Mellon University

Turing Award (1975)
artificial intelligence, the psychology of human cognition

Nobel Prize in Economics (1978)
decision-making process within economic organizations

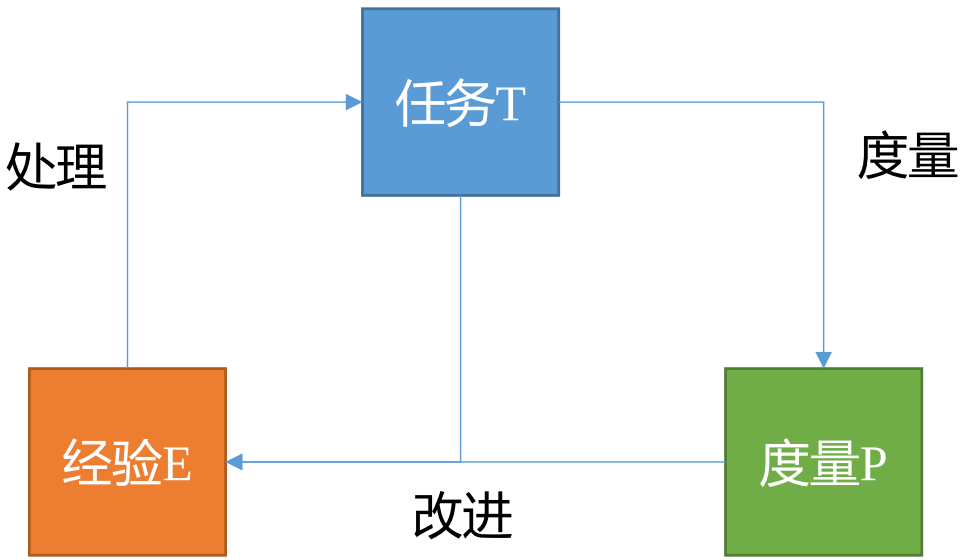
什么是机器学习

- 对于某类任务T和性能度量P，如果一个计算机程序在某些任务T上以P度量的性能随着经验E的增加而提高，那么我们称这个计算机程序是在从经验E中学习 —— Tom Mitchell



机器学习致力于研究如何通过计算的手段，利用经验来改善系统自身的性能，从而在计算机上从数据中产生“模型”，用于对新的情况给出判断。

什么是机器学习



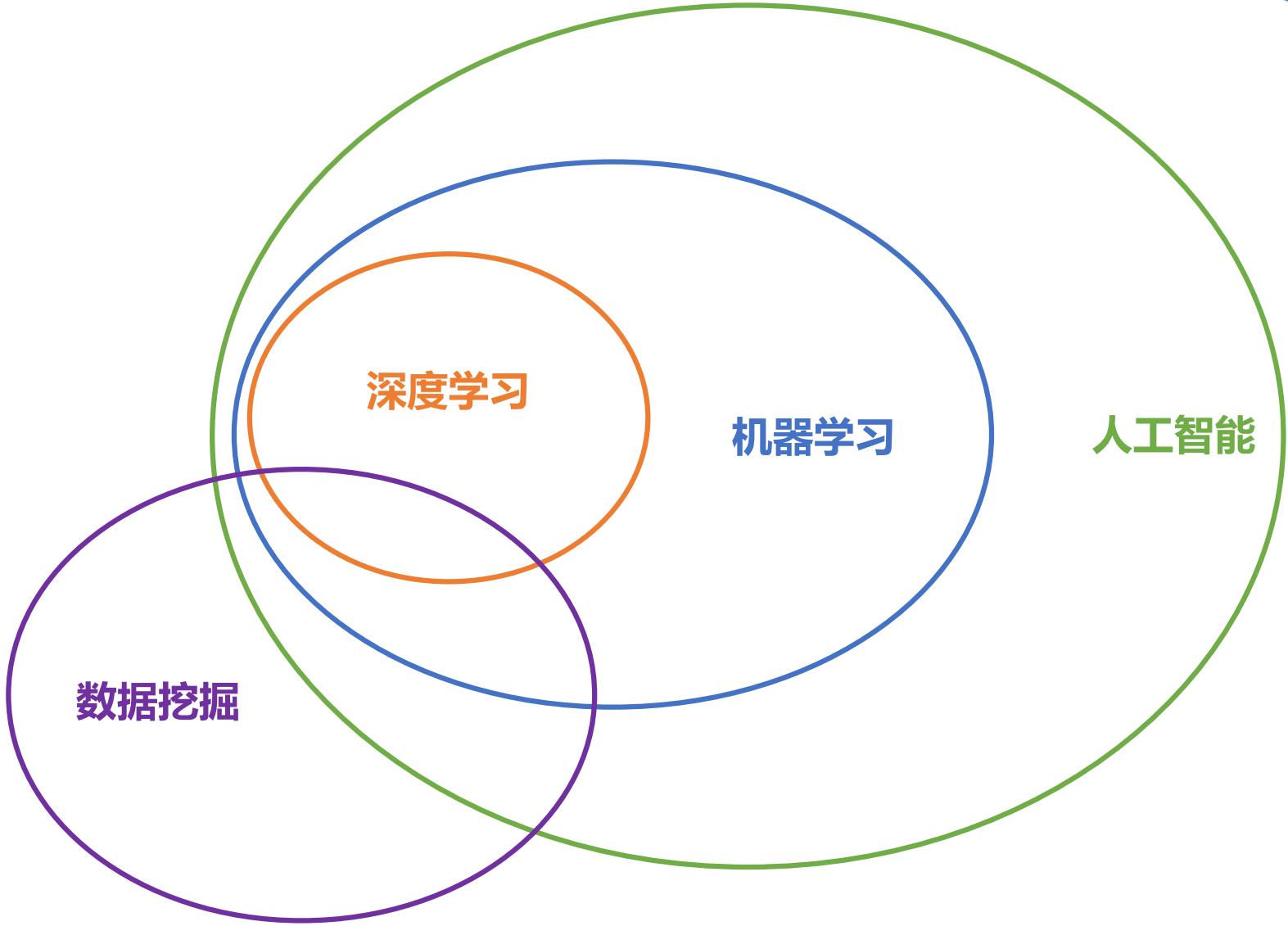
邮件分类

- T：将电子邮件分类为垃圾邮件或合法邮件
- P：被分类会垃圾邮件的比例
- E：邮件数据库和人工标记

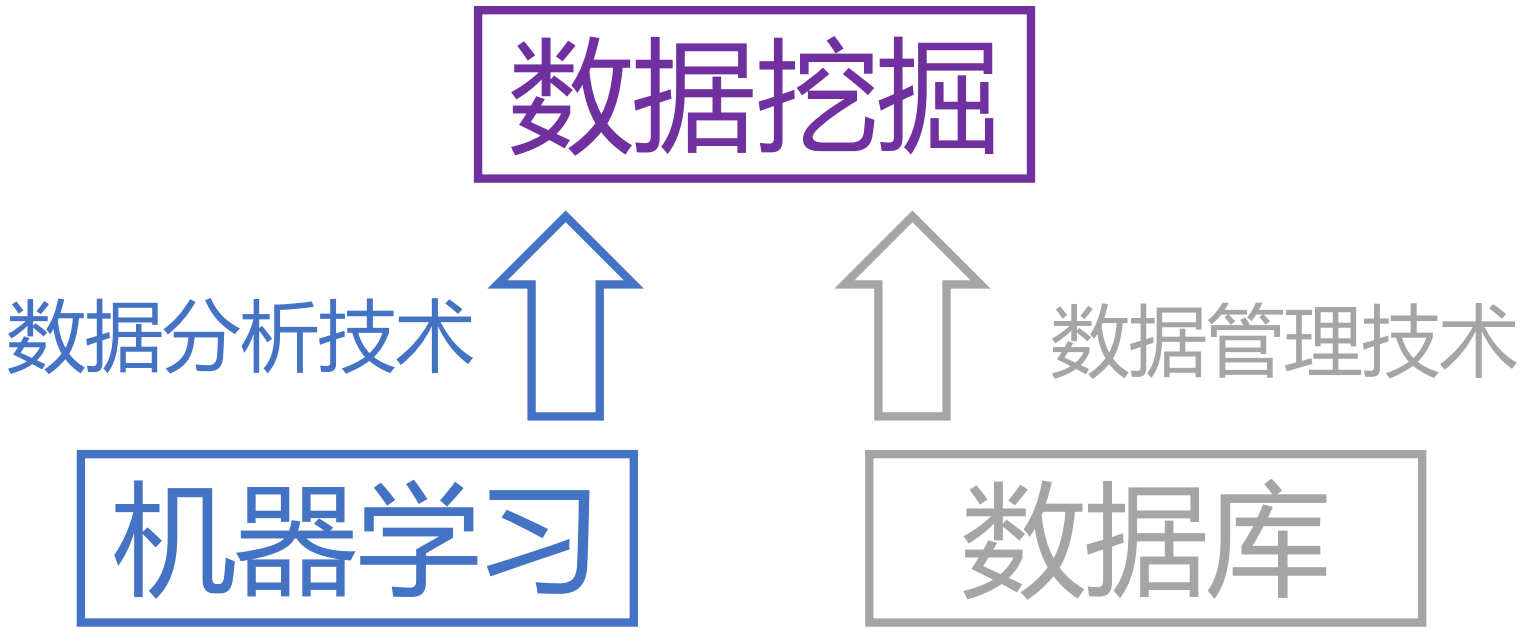
光学字符识别

- T：识别手写字符
- P：有多少比例字符被识别
- E：手写字符数据库和人工标记

什么是机器学习

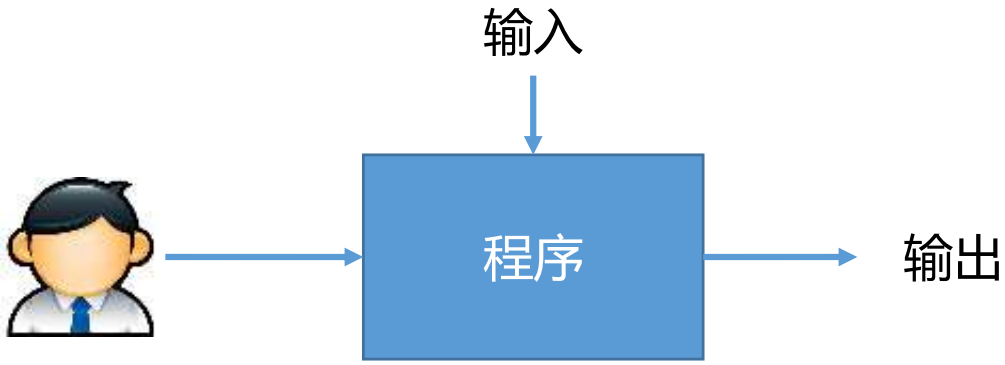


什么是机器学习

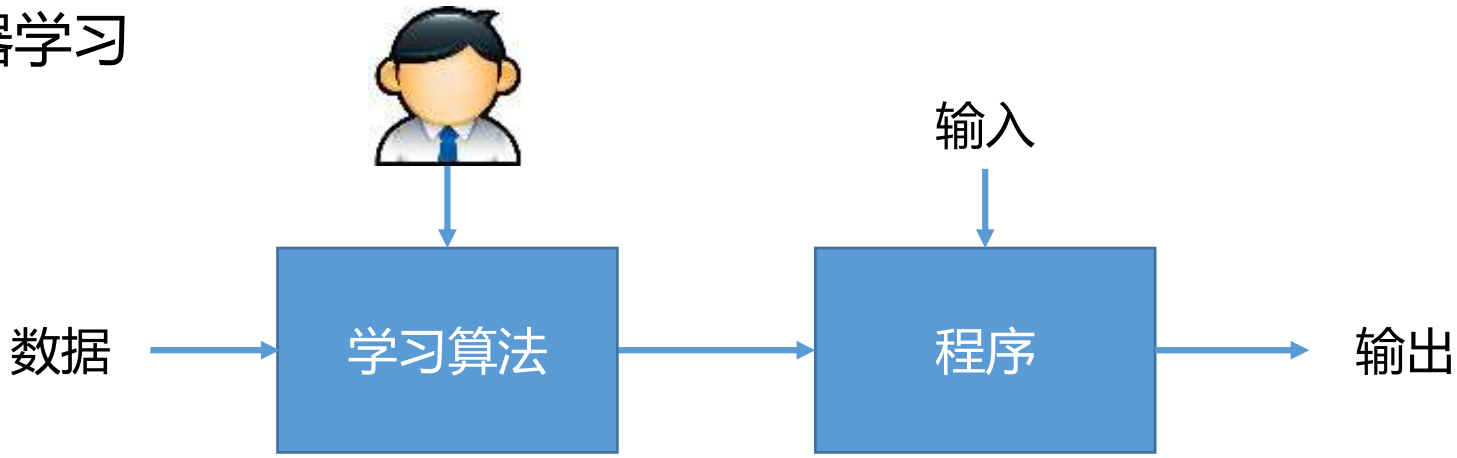


机器学习是什么

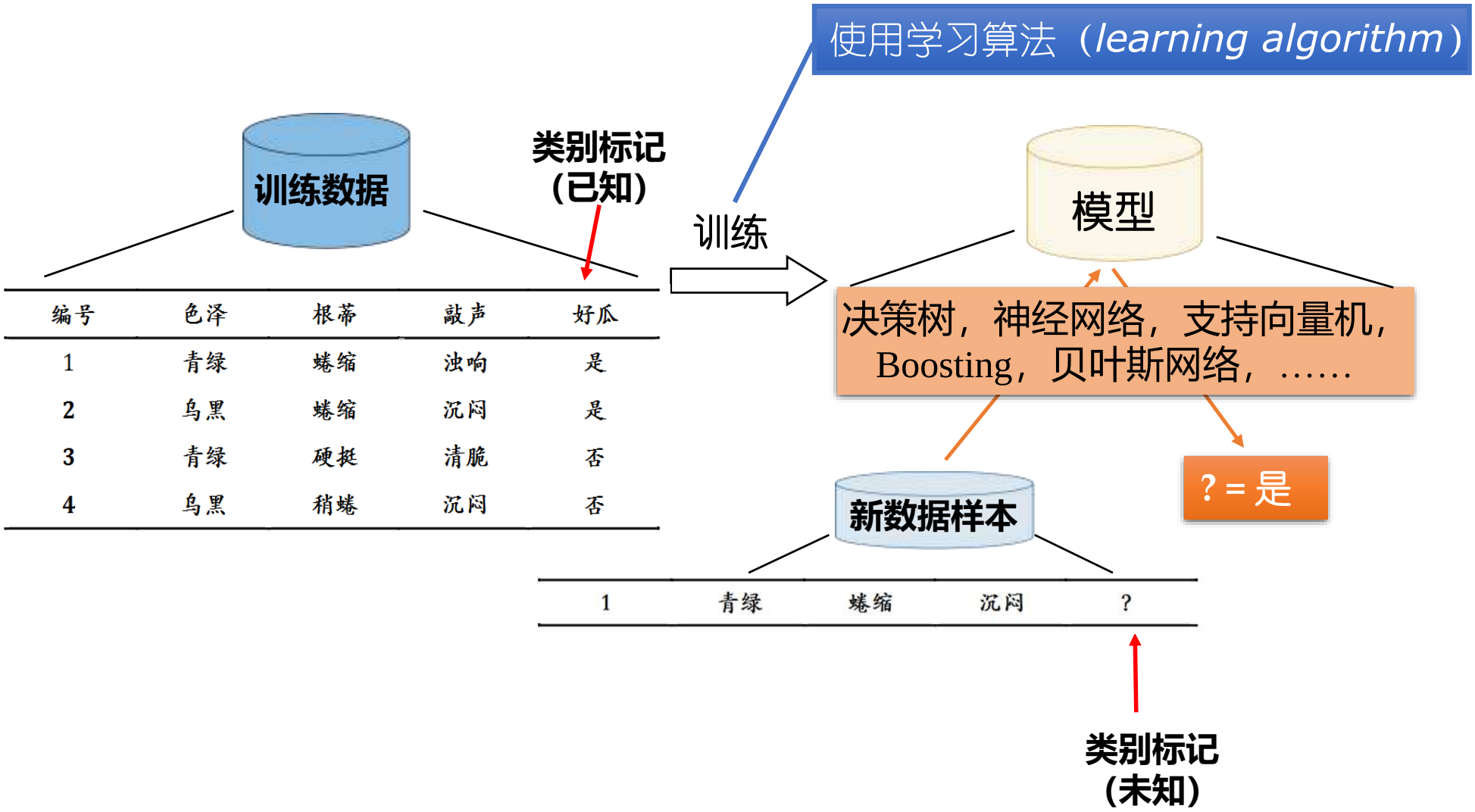
- 传统编程



- 机器学习



典型机器学习过程



编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	沉闷	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

1	青绿	蜷缩	沉闷	?

机器学习的基本概念—数据（经验E）

特征 ↑　　标记 ↑

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是 → 训练样本
2	乌黑	蜷缩	沉闷	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否
1	青绿	蜷缩	沉闷	?

训练集 ←　　测试集 ←

机器学习的基本概念—任务T

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	沉闷	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

↑ 标记

- 按照**标记**区分
  - 分类：**标记**为离散值（二分类、多分类）
  - 回归：**标记**为连续值（瓜的成熟度）
  - 聚类：没有**标记**

机器学习的基本概念—任务T

- 按照标记区分

- 分类：标记为离散值（二分类、多分类）
- 回归：标记为连续值（瓜的成熟度）

监督学习
Supervised Learning

- 聚类：没有标记

无监督学习
Unsupervised Learning

监督学习
Supervised Learning + 无监督学习
Unsupervised Learning = 半监督学习
Semi-supervised Learning

机器学习的基本概念—泛化能力

- 机器学习的目标是使得学到的模型能很好的适用于“新样本”，而不仅仅是训练集合，我们称模型适用于新样本的能力为泛化(generalization)能力。

- 通常假设样本空间中的样本服从一个未知分布 $\mathcal{D}$，样本从这个分布中独立获得，即“独立同分布”(i.i.d)。一般而言训练样本越多越有可能通过学习获得强泛化能力的模型

机器学习的基本概念—假设空间

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	沉闷	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

归纳学习 inductive learning

假设
Hypothesis

(色泽=?) ∧ (根蒂=?) ∧ (敲声=?) ↔ 好瓜

1. 青绿
2. 乌黑
3. * (通配符)

在假设空间中搜索不违背训练集的假设

假设空间大小：3*4*4+1=49

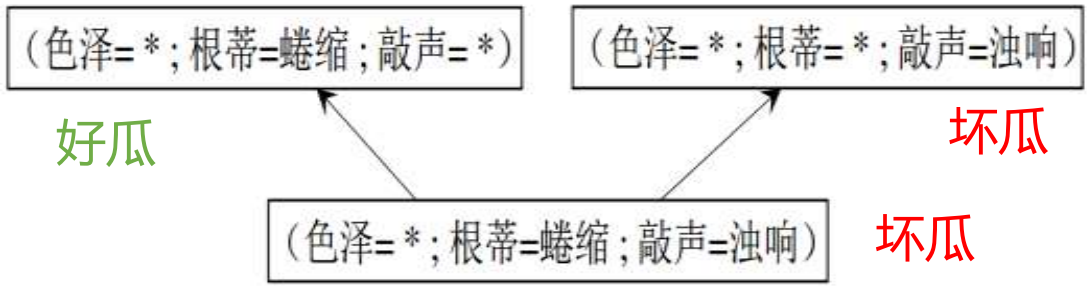
∅ 表示好瓜概念不成立

机器学习的基本概念—归纳偏好

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	沉闷	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

归纳学习 inductive learning

假设空间中有三个与训练集一致的假设

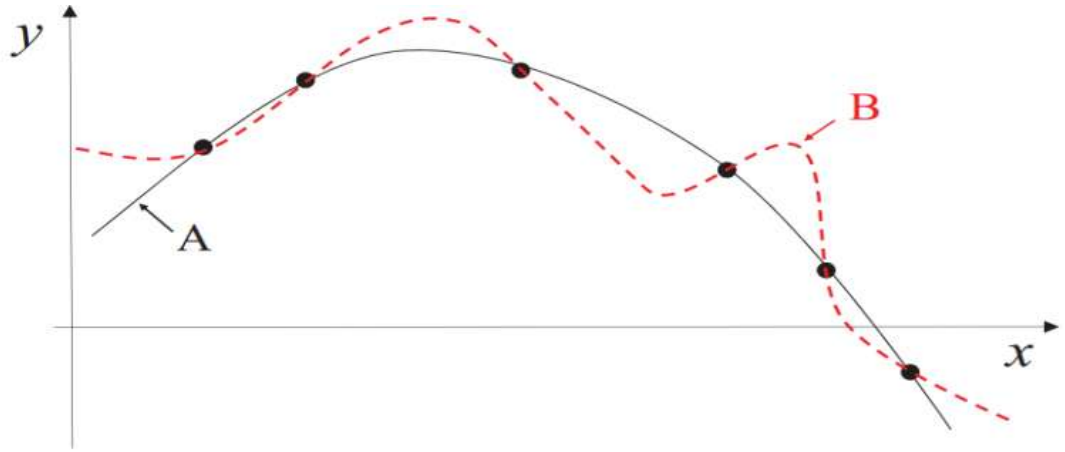


他们对(色泽=青绿;根蒂=蜷缩;敲声=沉闷)的瓜会预测结果不同

机器学习的基本概念—归纳偏好

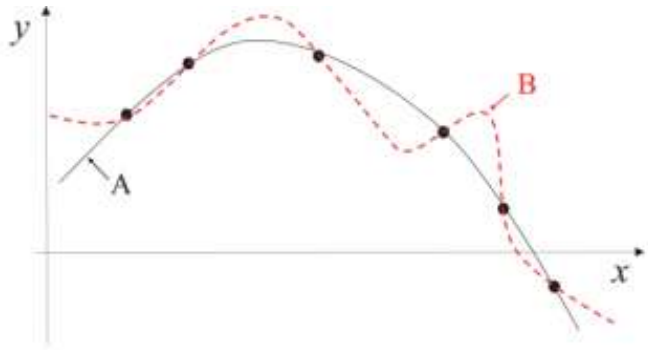
- 归纳偏好可看作学习算法自身在一个可能很庞大的假设空间中对假设进行选择的启发式或“价值观”.

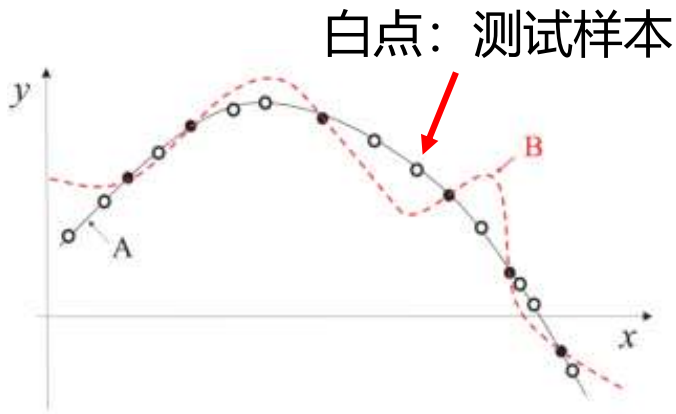
- “奥卡姆剃刀”是一种常用的、自然科学研究中最基本的原则，即“若有多个假设与观察一致，选最简单的那个”.



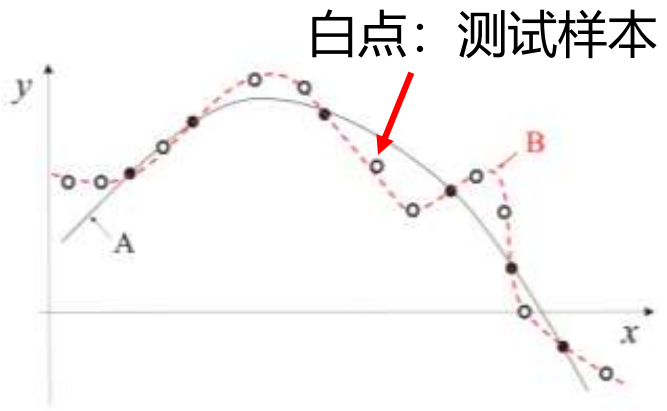
存在多条曲线与有限样本训练集一致

机器学习的基本概念—NFL





A优于B



B优于A

没有免费的午餐 No Free Lunch

一个算法$\mathfrak{L}_a$如果在某些问题上比另一个算法$\mathfrak{L}_b$好，必然存在另一些问题，$\mathfrak{L}_b$ 比 $\mathfrak{L}_a$ 好

机器学习的基本概念—NFL

- 假设样本空间 $\mathcal{X}$和假设空间 $\mathcal{H}$ 离散,令 $P(h|\boldsymbol{X},\mathfrak{L}_a)$ 代表算法$\mathfrak{L}_a$基于训练数据$\boldsymbol{X}$产生假设h的概率，在令f代表要学的目标函数，在训练集之外所有样本上的总误差为

$$E_{ote}(\mathfrak{L}_a|X,f)=\sum_{h}\sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}}P(\boldsymbol{x})\mathbb{I}\big(h(\boldsymbol{x})\neq f(\boldsymbol{x})\big)P(h|\boldsymbol{X},\mathfrak{L}_a)$$

$\mathbb{I}(\cdot)$为指示函数，若 $\cdot$ 为真取值1，否则取值0

考虑二分类问题，目标函数可以为任何函数$\mathcal{X}\mapsto\{0,1\}$，函数空间为$\{0,1\}^{|\mathcal{X}|}$。对所有可能$f$按**均匀分布**对误差求和，有如下结论：

$$\sum_{f}E_{ote}(\mathfrak{L}_a|\boldsymbol{X},f)=2^{|\mathcal{X}|-1}\sum_{\boldsymbol{x}\in\mathcal{X}\setminus\boldsymbol{X}}P(\boldsymbol{x})$$

总误差与学习算法无关！

机器学习的基本概念—NFL

考虑二分类问题，目标函数可以为任何函数$\mathcal{X} \mapsto \{0,1\}$，函数空间为$\{0,1\}^{|\mathcal{X}|}$。
对所有可能f按**均匀分布**对误差求和，有如下结论：

$$\sum_f E_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = 2^{|\mathcal{X}|-1} \sum_{\boldsymbol{x}\in\mathcal{X}\setminus\boldsymbol{X}} P(\boldsymbol{x})$$

- 证明如下

$$\sum_f E_{ote}(\mathfrak{L}_a|\boldsymbol{X}, f) = \sum_f \sum_h \sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}} P(\boldsymbol{x})\mathbb{I}\big(h(\boldsymbol{x}) \neq f(\boldsymbol{x})\big)P(h|\boldsymbol{X}, \mathfrak{L}_a)$$

$$= \sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}} P(\boldsymbol{x}) \sum_h P(h|\boldsymbol{X}, \mathfrak{L}_a) \sum_f \mathbb{I}\big(h(\boldsymbol{x}) \neq f(\boldsymbol{x})\big)$$

函数空间大小$2^{|\mathcal{X}|}$
f均分分布，一半的f对$\boldsymbol{x}$的预测和h不一致

$$= \sum_{\boldsymbol{x}\in\mathcal{X}-\boldsymbol{X}} P(\boldsymbol{x}) \sum_h P(h|\boldsymbol{X}, \mathfrak{L}_a) \frac{1}{2} 2^{|\mathcal{X}|}$$

$$= 2^{|\mathcal{X}|-1} \sum_{\boldsymbol{x}\in\mathcal{X}\setminus\boldsymbol{X}} P(\boldsymbol{x})$$

机器学习发展历程—推理期

- 基于符号知识表示，通过演绎推理技术取得了巨大成绩

- A. Newell和H. Simon的“逻辑理论家”(Logic Theorist)程序以及伺候的“通用问题求解”(General Problem Solving)程序等在当时取得了令人振奋的结果。
  - 1952年证明了著名数学家罗素和怀特海名著《数学原理》38条定理
  - 1963年证明了全部52条定理
  - 定理2.85比罗素和怀特海证明的更巧妙
  - 获得了1975年的图灵奖

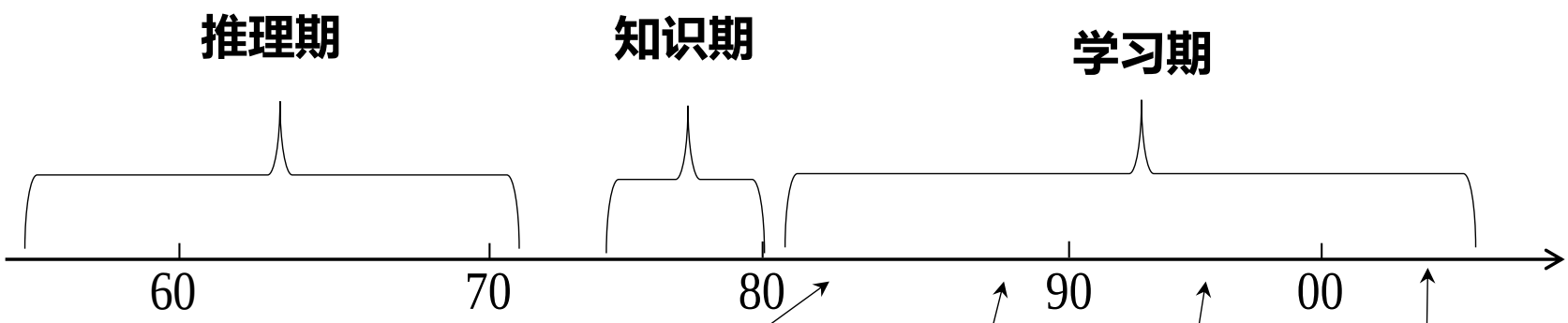
机器学习发展历程—知识期

- 基于符号知识表示，通过获取和利用领域知识建立专家系统取得大量成果

- “知识工程” 之父Edward A. Feigenbaum 研制了世界上第一个专家系统DENDRAL，并获得了1994年的图灵奖
  - DENDRAL输入的是质谱仪的数据，输出是给定物质的化学结构。
  - 捕捉化学家的化学分析知识，把知识提炼成规则。
  - 质谱数据就是关于原子重量的信息，能够帮助化学家确定一种化合物的结构和性质。

- 但是由人来总结知识再交给计算机相当困难。

机器学习发展历程



符号主义学习：决策树和基于逻辑的学习

- 决策树：以信息论为基础，最小化信息熵，模拟了人类对概念进行判定的树形流程
- 基于逻辑的学习（Inductive Logic Programming）：使用一阶逻辑进行知识表示，通过修改扩充逻辑表达式对数据进行归纳

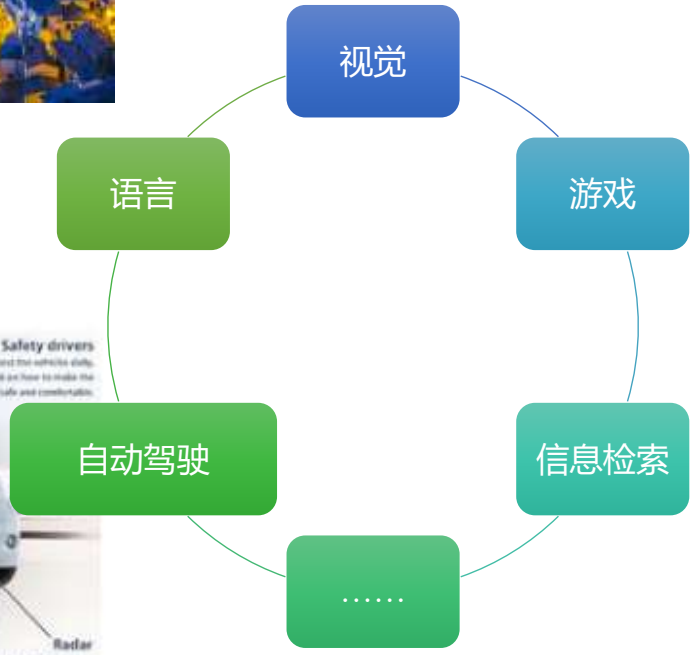
连接主义学习：基于神经网络

统计学习：支持向量机和核方法

连接主义学习：深度学习

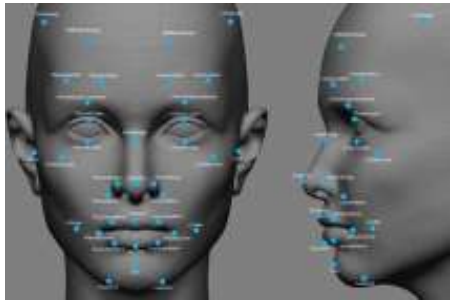
机器学习应用


Laser
Safety drivers
Processor
Position sensor
Orientation sensor
Radar


视觉
游戏
信息检索
……
自动驾驶
语言

机器学习取得了巨大的进展

- **语音识别**：微软英语语音识别实现词错率5.9%的突破，第一次**超越人类**
- **人脸识别**：Facebook的人脸识别系统DeepFace达到97.53%的准确率，**达到人类水平**
- **图像识别**：微软在ImageNet图像数据库上，达到4.94%的错误率，**低于人类5.1%的错误率**
- **人机对弈**：AlphaGo以4：1的战绩击败李世石，Master在围棋快棋上击败柯杰，聂卫平等高手，取得**60胜0负**的战绩

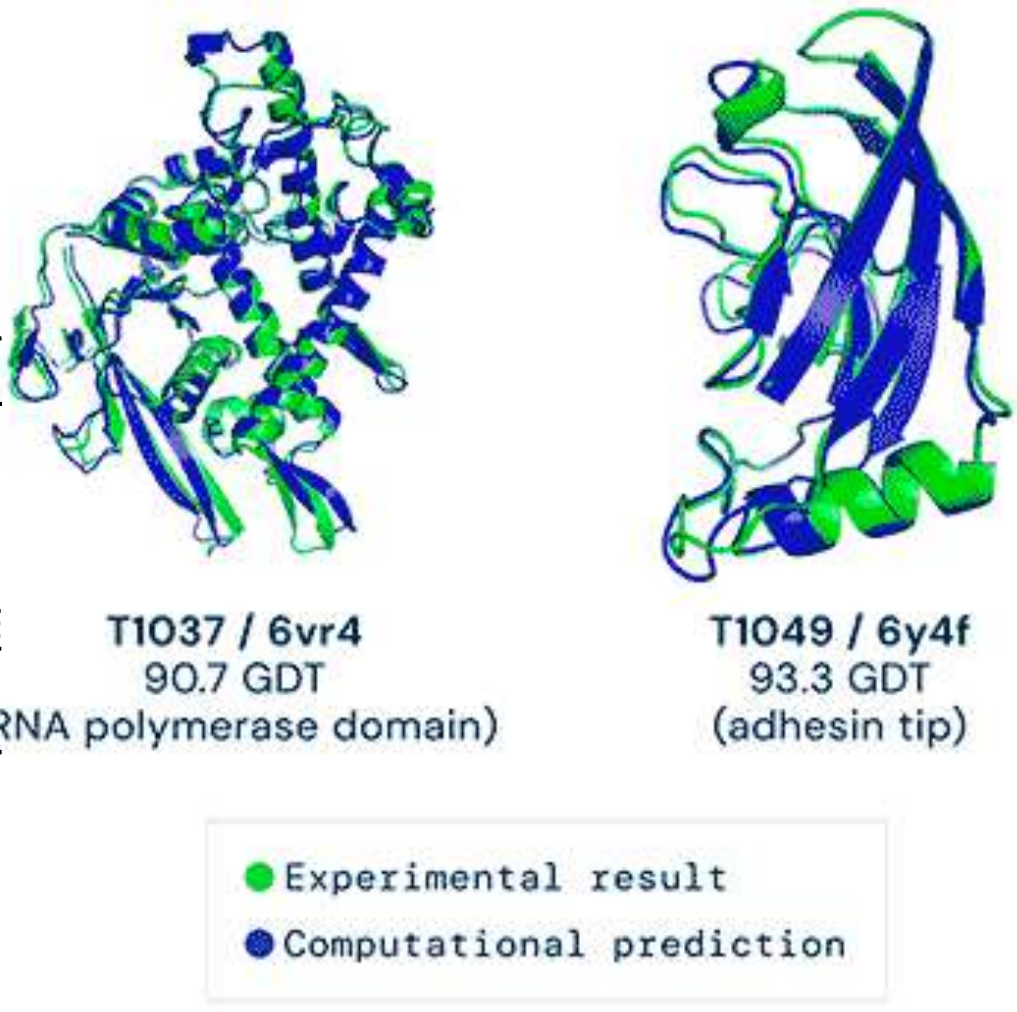



人工智能取得了巨大的进展

- 蛋白质结构预测（AlphaFold）：

AlphaFold在数天内通过蛋白质序列来预测蛋白质结构，此前科学家识别蛋白质形状需花费数年时间。

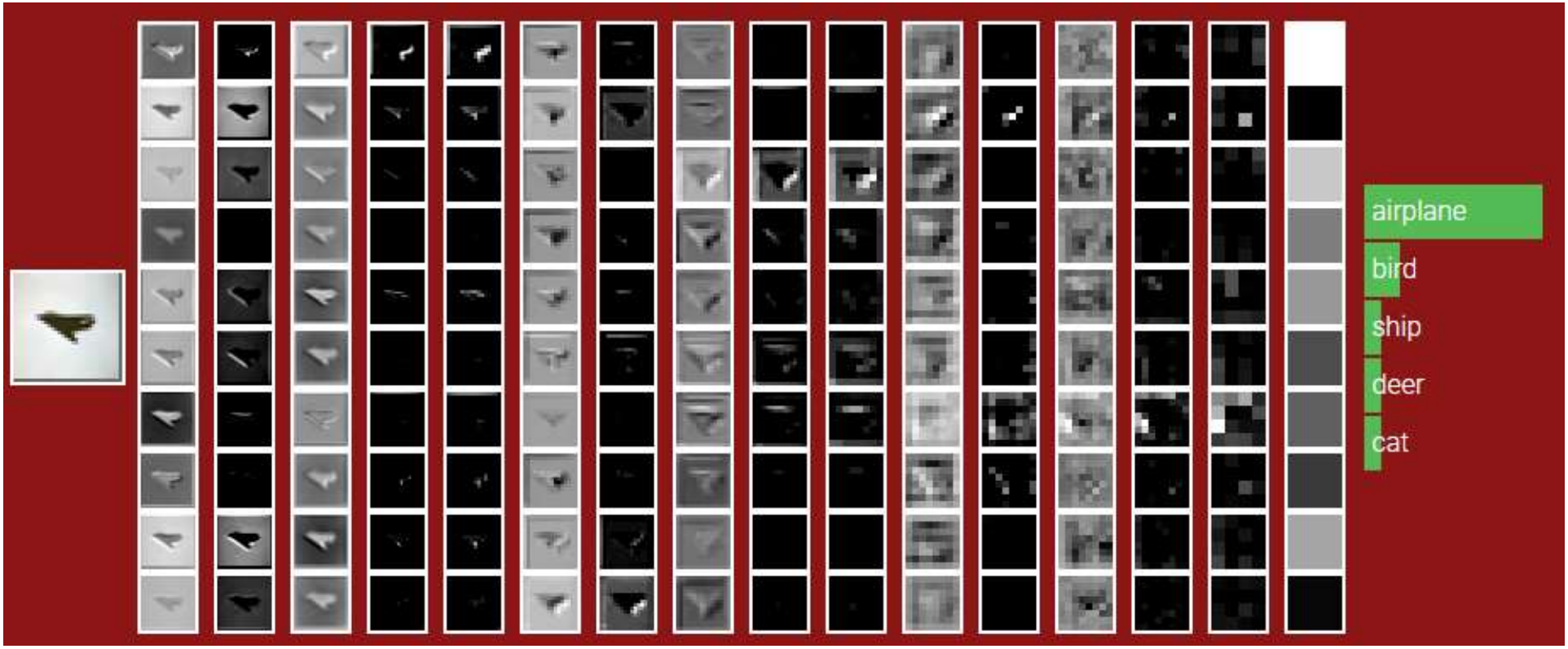
蛋白质通过卷曲折叠会构成三维结构，蛋白质的功能正由其结构决定，了解蛋白质结构有助于开发治疗疾病的药物。



机器学习应用—图片搜索


Google
图片
924f8...5a5f7.jpeg
蔡徐坤 打
全部
图片
地图
购物
找到约 576 条结果 （用时 0.83 秒）
图片尺寸:
440 × 262
查找该图片的其他尺
全部尺寸 - 小尺寸 -
可能相关的搜索查询：
蔡徐坤 打 篮球
外观类似的图片

机器学习应用—图片分类

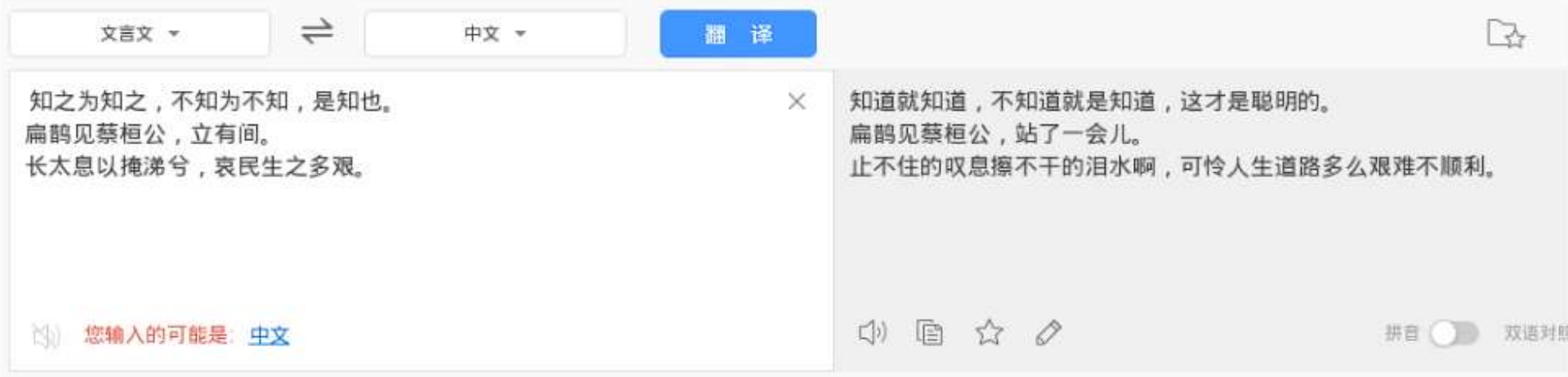


机器学习应用—换脸

学习算法利用人脸捕捉，让你在视频里实时扮演另一个人，简单来讲，就是可以把你的面部表情实时移植到视频里正在发表演讲的美国总统身上。

机器学习应用—翻译


Google
翻译
关闭即时翻译
中文 英语 德语 检测语言
英语 中文(简体) 日语
翻译
Google神经机器翻译系统面世。该系统克服在大型数据集上工作的挑战，不再将句子分解为词和短语独立翻译，而是翻译完整的句子，使得误差降低了55%-85%以上。目前这项技术已经运用于Google Translate的汉英翻译。
112/5000
The Google Neuro Machine Translation System is available. The system overcomes the challenge of working on large data sets, no longer decomposing sentences into independent translations of words and phrases, but translating complete sentences, reducing errors by more than 55%-85%. This technology is currently used in Chinese-English translation of Google Translate.
提出修改建议
Google shénjīng jīqì fānyì xìtǒng miànshì. Gāi xìtǒng kèfú zài dàxíng shùjù jí shàng gōngzuò de tiǎozhàn, bù zài jiāng jùzi fēnjiě wéi cí hé duǎnyǔ dúlì fānyì, ér shì fānyì wánzhěng de jùzi, shǐdé wùchā jiǎngdīle 55%-85%yǐshàng. Mùqián zhè xiàng jìshù yǐjīng yùnyòng yú Google Translate de hàn yīng fānyì.


文言文
中文
翻 译
知之为知之，不知为不知，是知也。
扁鹊见蔡桓公，立有间。
长太息以掩涕兮，哀民生之多艰。
您输入的可能是：中文
知道就知道，不知道就是知道，这才是聪明的。
扁鹊见蔡桓公，站了一会儿。
止不住的叹息擦不干的泪水啊，可怜人生道路多么艰难不顺利。
拼音
双语对照

机器学习应用—聊天机器人

机器学习应用—写诗

语音作诗

藏头诗

看图作诗

机器学习应用—作曲编曲

念奴娇·赤壁怀古

【作者】苏轼【朝代】宋

大江东去，浪淘尽，千古风流人物。故垒西边，人道是，三国周郎赤壁。乱石穿空，惊涛拍岸，卷起千堆雪。江山如画，一时多少豪杰。遥想公瑾当年，小乔初嫁了，雄姿英发。羽扇纶巾，谈笑间，樯橹灰飞烟灭。故国神游，多情应笑我，早生华发。人生如梦，一尊还酹江月。

http://tv.cctv.com/2018/08/18/VIDE1b0kQ8p05S3SMJYFtadm180818.shtml

机器学习应用—下棋

AlphaGo以3：0战胜柯洁

机器学习应用—德州扑克

- 在无限制德州扑克六人对决的比赛中，由 Facebook 与CMU共同开发的德扑 AI Pluribus 成功战胜了五名专家级人类玩家

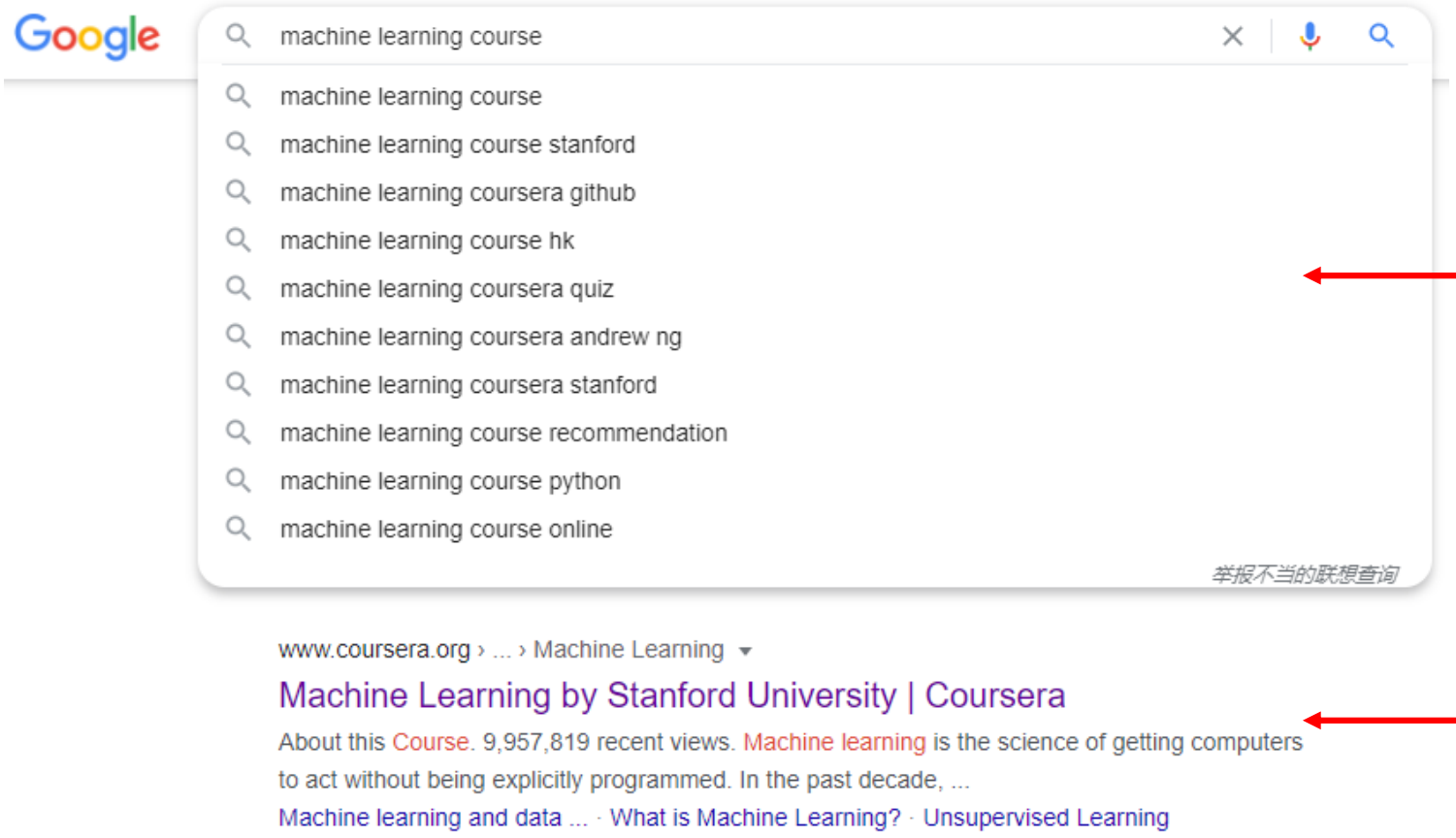
机器学习应用—麻将

微软亚洲研究院研发的麻将AI Suphx， 在国际知名麻将平台“天凤”上荣升十段，稳定段位显著超越人类顶级选手

游戏AI历史


TD-Gammon
1956年
Arthur Samuel开发的
AI程序首先应用于国际跳棋
1992年
Hans Berliner开发的AI程序
在双陆棋中取得突破
1994年
Jonathan Schaeffer
团队开发的AI程序
战胜国际跳棋
世界冠军
Chinook
许峰雄团队开发的AI程序
击败国际象棋世界冠军
1997年
Deep Blue
DeepMind团队
开发的AI程序
战胜围棋世界冠军
2016年
AlphaGo
2017年
下一个里程碑在哪里？
卡耐基梅隆大学、Facebook AI以及阿尔伯特大学开发的
AI程序在德州扑克取得突破
Libratus
DeepStack
Pluribus

机器学习应用—搜索



查询词建议

相关性计算
&PageRank

机器学习应用—推荐

机器学习

由AlphaGo的武林秘籍，赢得人机大战的必由之路！人工智能大牛周志华教授巨著，全面揭开机器学习的奥秘

周志华 著

京东价：¥68.40 (7.6折) [定价：¥90.00] [降价通知]

累计评价 10万+

促销信息 换购 购买1件可优惠换购热销商品 立即换购 >>

加价购 满10元另加28.90元，或满12元另加16.90元，或满15元另加9.90元，

即可在购物车换购热销商品 详情 >>

以上促销可在购物车任选其一

增值业务 助力环保，传递知识，旧书换新

排 名 自营 计算机与互联网销量榜 第 4 位

配 送 至 安徽合肥市蜀山区蜀山街道 有货

由 京东 发货，并提供售后服务。23:10前下单，预计明天(09月14日)送达

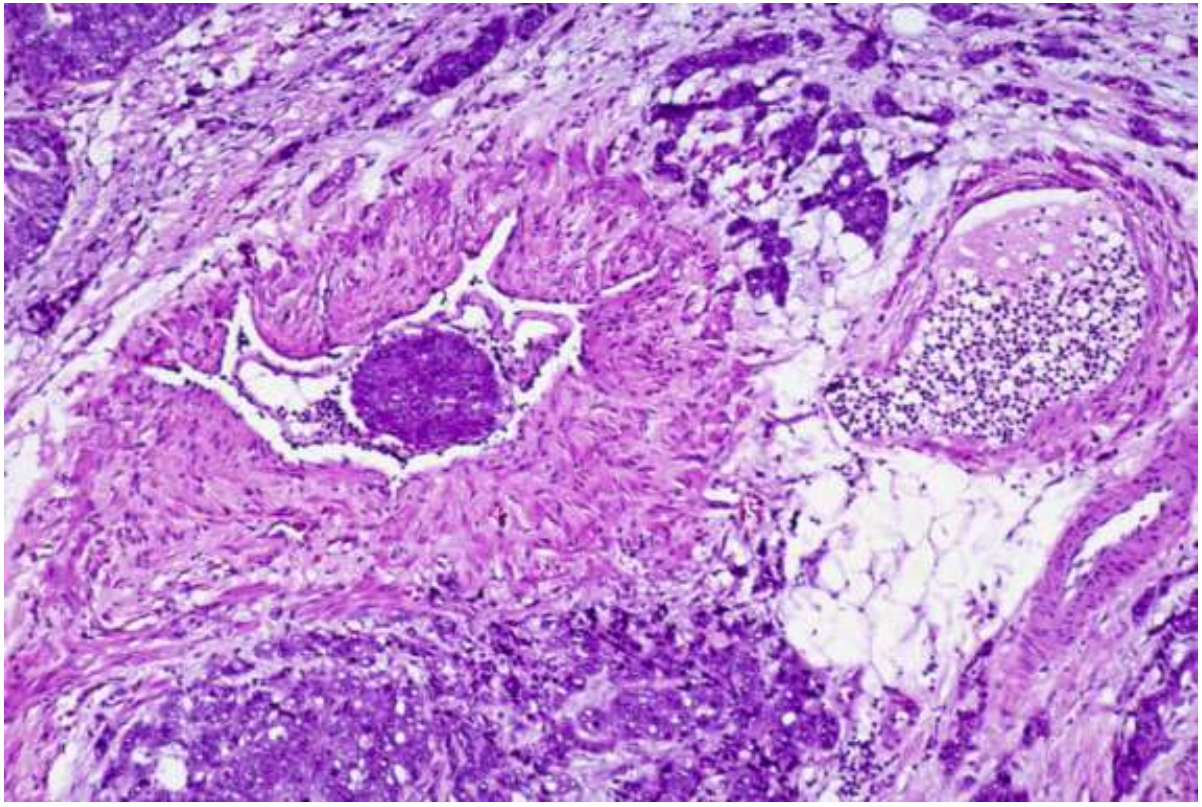
重 量 0.92kg

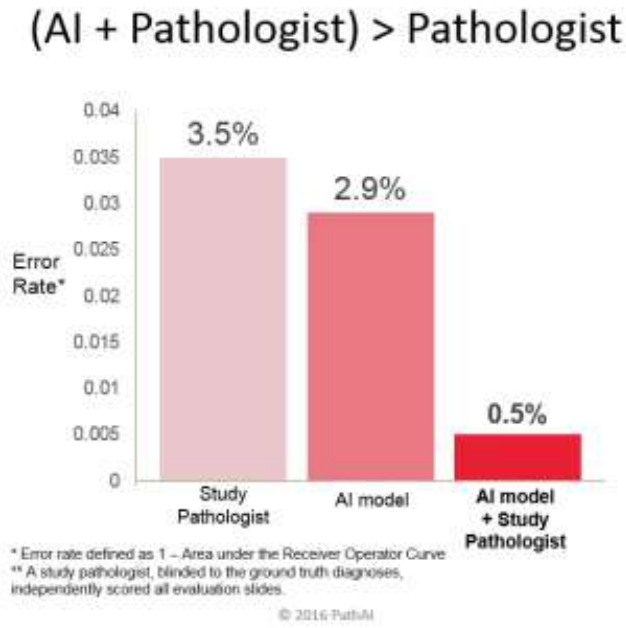
关注 分享 举报

猜你喜欢

机器学习应用—疾病诊断

- 乳腺癌诊断

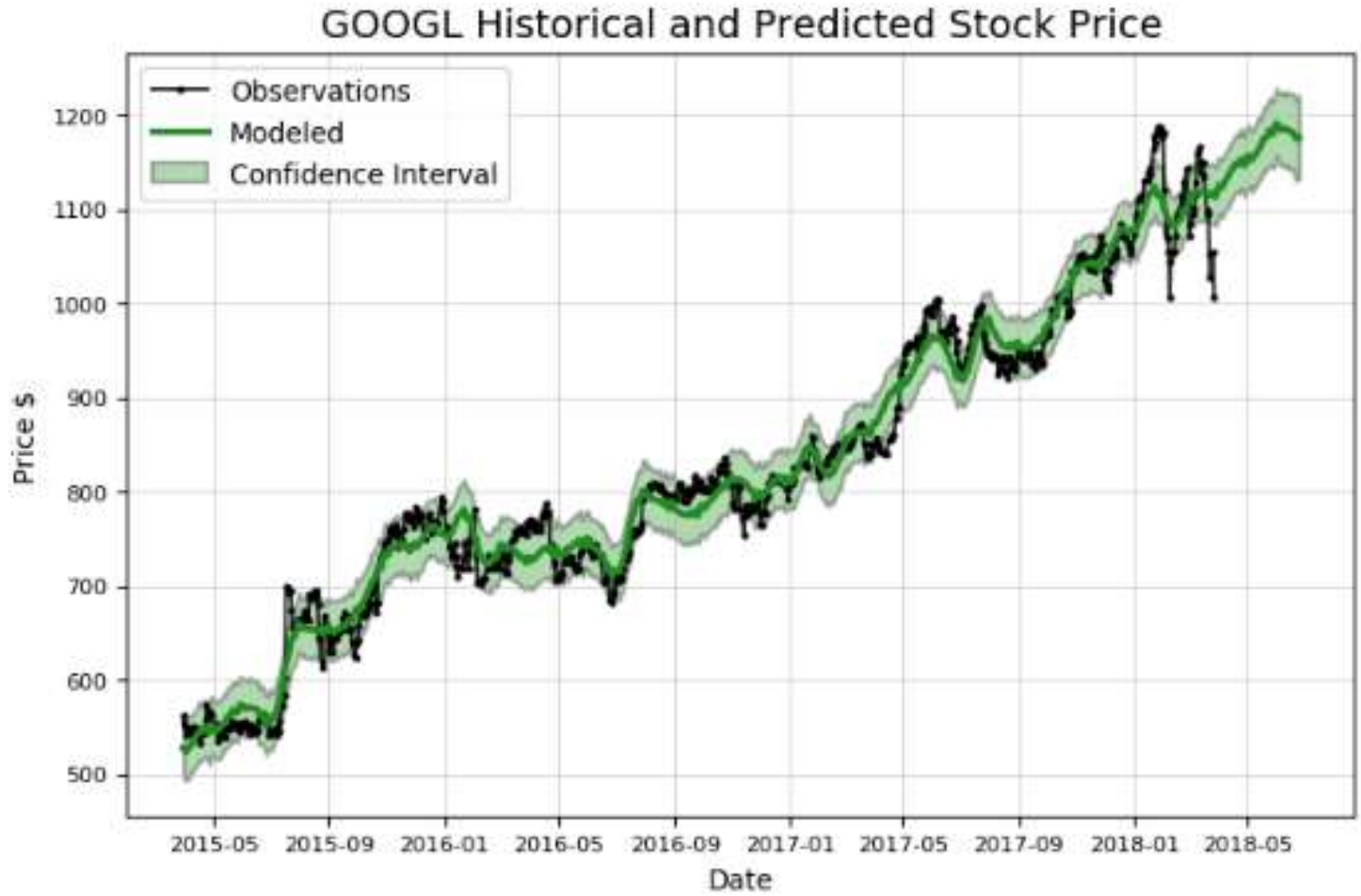




https://blogs.nvidia.com/blog/2016/09/19/deep-learning-breast-cancer-diagnosis/

机器学习应用—股价预测

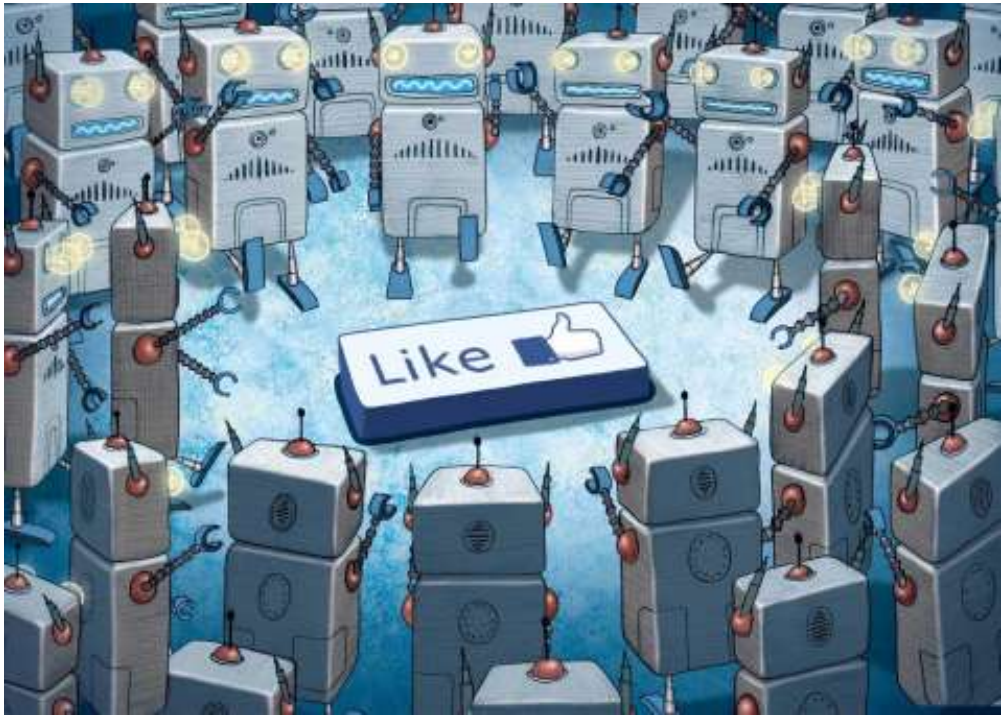
Predicted Price on 2018-06-25 00:00:00 = $1175.64



https://towardsdatascience.com/how-to-use-machine-learning-to-possibly-become-a-millionaire-predicting-the-stock-market-33861916e9c5

机器学习应用—异常检测

- 社交网络中异常账号检测



机器学习应用—虚假新闻检测

提问较真　　疫情数据　　北京疫情

新冠肺炎男性患者的死亡率更高 确实如此

2020-09-10

新冠死者平均只损失了几个月的寿命 谣言

2020-09-09

机器学习应用—自动驾驶

机器学习应用—自动驾驶

机器学习的部分数学基础

- 矩阵
- 概率分布
- 优化

机器学习的部分数学基础—矩阵

- 矩阵为二维数组，用大写粗斜体表示 $\boldsymbol{A} \in \mathbb{R}^{m \times n}$
  - $\boldsymbol{A}_{:,i}$表示第i列
  - $\boldsymbol{A}_{i,:}$ 表示第i行
  - $A_{i,j}$ 表示第i行第j列元素
- 矩阵范数
  - 函数$f(\boldsymbol{A})$衡量矩阵的大小，满足三个条件
    - $f(\boldsymbol{A}) \geq \boldsymbol{0}$，等号成立当且仅当$\boldsymbol{A} = 0$
    - $f(\boldsymbol{\alpha A}) = \boldsymbol{\alpha} f(\boldsymbol{A})$
    - $f(\boldsymbol{A} + \boldsymbol{B}) \leq f(\boldsymbol{A}) + f(\boldsymbol{B})$
  - Frobenius norm
    - $\|\boldsymbol{A}\|_F = \sqrt{\sum_i \sum_j |A_{i,j}|^2}$ ⇒ $\|\boldsymbol{A}\|_F^2 = \mathrm{tr}(\boldsymbol{A}^\top \boldsymbol{A})$
  - p-诱导范数
    - $\|\boldsymbol{A}\|_p = \sup_{x \neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p}$ $\|\boldsymbol{x}\|_p = \sqrt[p]{x_1^p + \cdots + x_d^p}$

⇒ $\|\boldsymbol{A}\|_2 = \sigma_{max}(A)$

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

行

列

机器学习的部分数学基础—矩阵导数

- $\left(\nabla f(\boldsymbol{x})\right)_i = \frac{\partial f(\boldsymbol{x})}{\partial x_i}$ $\qquad$ $\left(\nabla^2 f(\boldsymbol{x})\right)_{ij} = \frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}$ (海森矩阵)

- $\left(\frac{\partial x}{\partial \boldsymbol{A}}\right)_{ij} = \frac{\partial x}{\partial A_{ij}}$

- $\frac{\partial \operatorname{tr}(\boldsymbol{AB})}{\partial A_{ij}} = B_{ji}$ ⇒ $\frac{\partial \operatorname{tr}(\boldsymbol{AB})}{\partial \boldsymbol{A}} = \boldsymbol{B}^\top$ $\qquad$ $\frac{\partial \operatorname{tr}(\boldsymbol{A}^\top \boldsymbol{B})}{\partial \boldsymbol{A}} = \boldsymbol{B}$

- $\frac{\partial \|\boldsymbol{A}\|_F^2}{\partial \boldsymbol{A}} = \frac{\partial \operatorname{tr}(\boldsymbol{A}^\top \boldsymbol{A})}{\partial \boldsymbol{A}} = 2\boldsymbol{A}$

机器学习的部分数学基础—矩阵导数

- $\left(\frac{\partial \boldsymbol{A}}{\partial x}\right)_{ij} = \frac{\partial A_{ij}}{\partial x}$ $\left(\frac{\partial x}{\partial \boldsymbol{A}}\right)_{ij} = \frac{\partial x}{\partial A_{ij}}$
- $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$ ⇒ $\frac{\partial \boldsymbol{A}^{-1}\boldsymbol{A}}{\partial x} = \frac{\partial \boldsymbol{A}^{-1}}{\partial x}\boldsymbol{A} + \boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x} = \boldsymbol{0}$ ⇒ $\frac{\partial \boldsymbol{A}^{-1}}{\partial x} = -\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\boldsymbol{A}^{-1}$
- 试计算 $\frac{\partial \det(\boldsymbol{A})}{\partial \boldsymbol{A}}$ ⇒ $\frac{\partial \det(\boldsymbol{A})}{\partial \boldsymbol{A}} = \boldsymbol{C} = \mathrm{adj}(\boldsymbol{A})^{\mathsf{T}}$（伴随矩阵） ⇒ $\frac{\partial \ln\det(\boldsymbol{A})}{\partial \boldsymbol{A}} = (\boldsymbol{A}^{-1})^{\mathsf{T}}$
- 回忆 $\det(\boldsymbol{A}) = \sum_i A_{ij}C_{ij}$，$C_{ij}$表示方阵$\boldsymbol{A}$关于$A_{ij}$的代数余子式
- 回忆矩阵逆的运算 $\boldsymbol{A}^{-1} = \det(\boldsymbol{A})^{-1}\,\mathrm{adj}(\boldsymbol{A})$
- 试计算$\frac{\partial \ln\det(\boldsymbol{A})}{\partial x}$

机器学习的部分数学基础—特征值分解

- 对于方阵$\boldsymbol{A}$，特征向量方程
  - $\boldsymbol{Av} = \lambda \boldsymbol{v}$，$\boldsymbol{v}$ 特征向量，λ为特征值　　$\boldsymbol{A}(\alpha v) = \lambda(\alpha \boldsymbol{v})$， 考虑单位特征向量

- 可对角化矩阵的特征值分解为
  - $\boldsymbol{A} = \boldsymbol{V}\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{V}^{-1}$，$\boldsymbol{\lambda}$对应特征值
  - $\boldsymbol{V}$中的每一列为特征向量

- 机器学习算法常常涉及 实对称矩阵
  - 可对角化的
  - 特征值是实数，特征值为正数的矩阵为**正定阵**，非负的为**半正定矩**
  - $\boldsymbol{V}$ 为正交矩阵，满足$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{I}$

机器学习的部分数学基础—奇异值分解

- 奇异值分解类似于特征值分解，但是对任意矩阵都成立
- 对于任意大小为$m \times n$的矩阵$\boldsymbol{A}$，$\boldsymbol{A}^{\top}\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$
  - 令$\boldsymbol{A}\boldsymbol{v} = \sigma \boldsymbol{u}$，那么$\boldsymbol{A}^{T}\sigma \boldsymbol{u} = \lambda \boldsymbol{v}$，分别左乘$\boldsymbol{A}$ 得到 $\boldsymbol{A}\boldsymbol{A}^{\top}\boldsymbol{u} = \frac{\lambda}{\sigma}\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{u}$

$\boldsymbol{u}$对应$\boldsymbol{A}\boldsymbol{A}^{\top}$的特征值为$\lambda$特征向量

  - 在$\boldsymbol{A}\boldsymbol{v} = \sigma \boldsymbol{u}$两边分别乘以$\boldsymbol{u}$，那么$\boldsymbol{u}^{\top}\boldsymbol{A}\boldsymbol{v} = \sigma$
  - 在$\boldsymbol{A}^{T}\sigma \boldsymbol{u} = \lambda \boldsymbol{v}$两边分别乘以$\boldsymbol{v}$，那么 $\boldsymbol{v}^{T}\boldsymbol{A}^{T}\boldsymbol{u} = \frac{\lambda}{\sigma}$

$$\sigma^2 = \lambda$$

- 将$\boldsymbol{A}\boldsymbol{v} = \sigma \boldsymbol{u}$写成矩阵形式为$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma} = \mathrm{diag}([\sigma_1, \ldots, \sigma_r, 0, \cdots, 0])$$
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$$
$$r = \mathrm{rank}(A)$$

- 由于$\boldsymbol{V}$是正交矩阵，所以$\boldsymbol{A} = \boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^{\top} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$

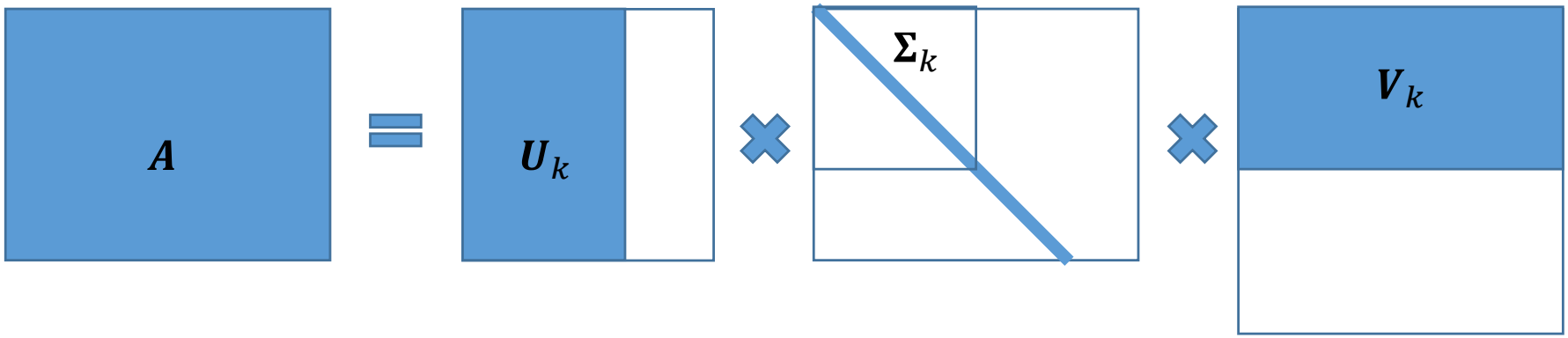
$\boldsymbol{\Sigma}$ 的大小为$m \times n$
$\boldsymbol{U}$的大小为$m \times m$
$\boldsymbol{V}$的大小为$n \times n$

$\boldsymbol{U}$的列向量为左奇异向量，$\boldsymbol{V}$的列向量为右奇异向量

机器学习的部分数学基础—奇异值分解

- 截断奇异值分解

$$\boldsymbol{A} \approx \boldsymbol{U}_k \, \boldsymbol{\Sigma}_k \, \boldsymbol{V}_k^T$$

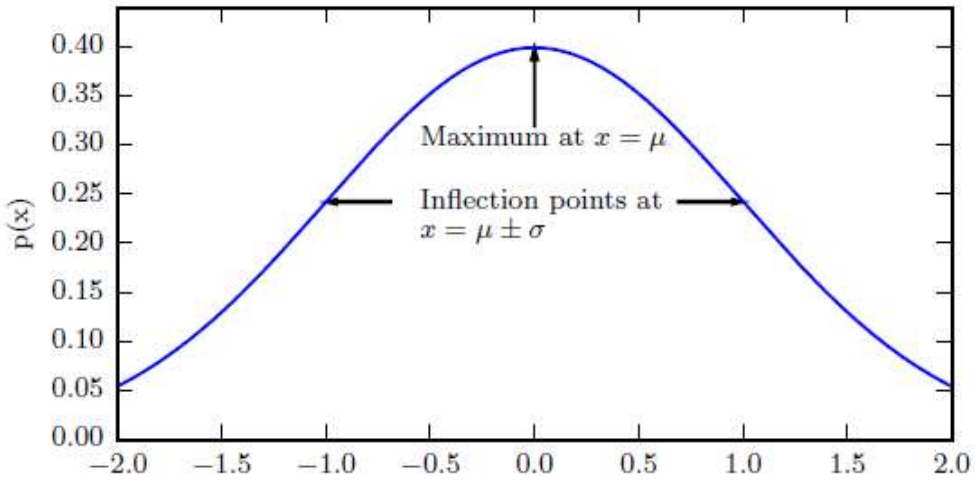


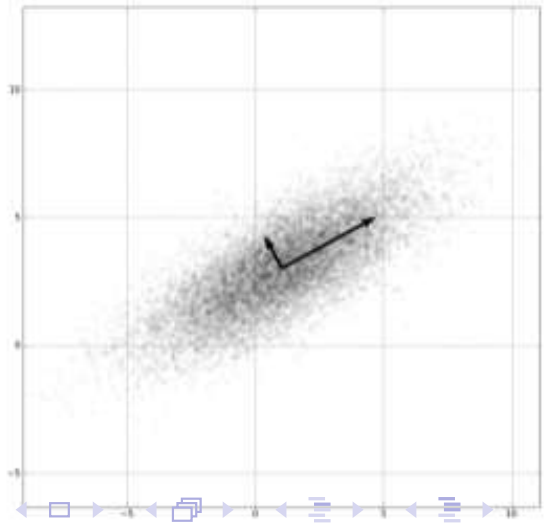
$\boldsymbol{U}_k$的大小为$m \times k$　　$\boldsymbol{\Sigma}_k$ 的大小为$k \times k$　　$\boldsymbol{V}_k$的大小为n $\times k$

$\boldsymbol{U}_k^{\mathrm{T}} \boldsymbol{U}_k = \boldsymbol{I}_k$　　$\boldsymbol{\Sigma}_k = \mathrm{diag}([\sigma_1, \ldots, \sigma_k])$　　$\boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{V}_k = \boldsymbol{I}_k$

机器学习的部分数学基础—概率分布

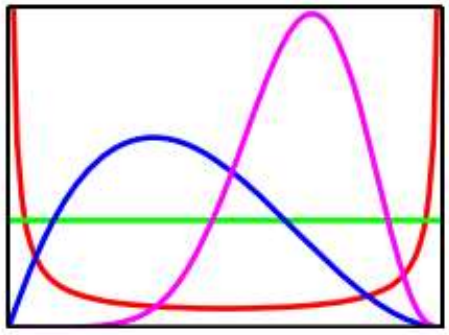
- 高斯分布、正态分布
  - $\mathcal{N}\left(x;\mu,\sigma^2\right)=\sqrt{\frac{1}{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$ 均值μ 标准差σ
  - $\mathcal{N}(x;\mu,\beta^{-1})=\sqrt{\frac{\beta}{2\pi}}\exp\left(-\frac{\beta}{2}(x-\mu)^2\right)$ 均值μ Scale β
- 多元正态分布
  - $\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})=\frac{1}{\sqrt{(2\pi)^n\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$
  - $\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\beta}^{-1})=\sqrt{\frac{\det(\boldsymbol{\beta})}{(2\pi)^n}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\beta}(\boldsymbol{x}-\boldsymbol{\mu})\right)$



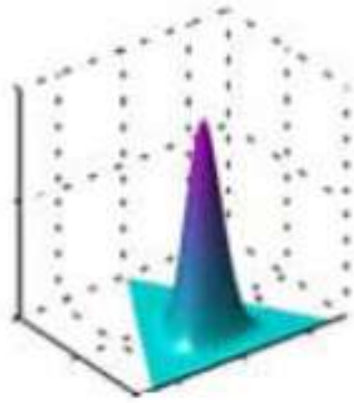


机器学习的部分数学基础—概率分布

- 贝塔分布 （Beta Distribution）是 随机变量 $\mu \in [0,1]$上的分布
  - $Beta(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1}$
  - $\mathbb{E}[\mu] = \frac{a}{a+b}$
  - $var[\mu] = \frac{ab}{(a+b)^2(a+b+1)}$



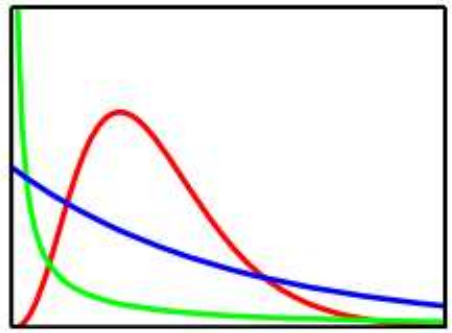
- 狄利克雷分布 （Dirichlet Distribution）是 贝塔分布的多元扩展
  - 多个连续变量$\mu_i \in [0,1]$的概率分布，满足 $\sum_i \mu_i = 1$
  - $\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)}\prod_i \mu_i^{\alpha_i - 1}$
  - $\mathbb{E}[\mu_i] = \frac{\alpha_i}{\sum_i \alpha_i}$



机器学习的部分数学基础—概率分布

- 伽玛分布 （Gamma Distribution）是正数随机变量 $\tau > 0$上的分布

  - $Gam(\tau|a,b) = \frac{1}{\Gamma(a)} b^a \tau^{a-1} e^{-b\tau}$
  - $\mathbb{E}[\tau] = \frac{a}{b}$
  - $var[\tau] = \frac{a}{b^2}$



机器学习的部分数学基础—KL散度

- KL散度：衡量两个分布的差异
  - $D_{KL}(P||Q) = \mathbb{E}_{x\sim P}\left[\log\frac{P(x)}{Q(x)}\right]$
  - 非负, P=Q时为零
  - $D_{KL}(P||Q) \neq D_{KL}(Q||P)$，但理论上最小值均当P=Q
- $D_{KL}(P||Q) = \mathbb{E}_{x\sim P}\left[\log\frac{P(x)}{Q(x)}\right] = \mathbb{E}_{x\sim P}[\log P(x)] - \mathbb{E}_{x\sim P}[\log Q(x)]$

$-H(P)$ P的熵

$H(P,Q)$ P和Q的交叉熵

机器学习的部分数学基础—优化

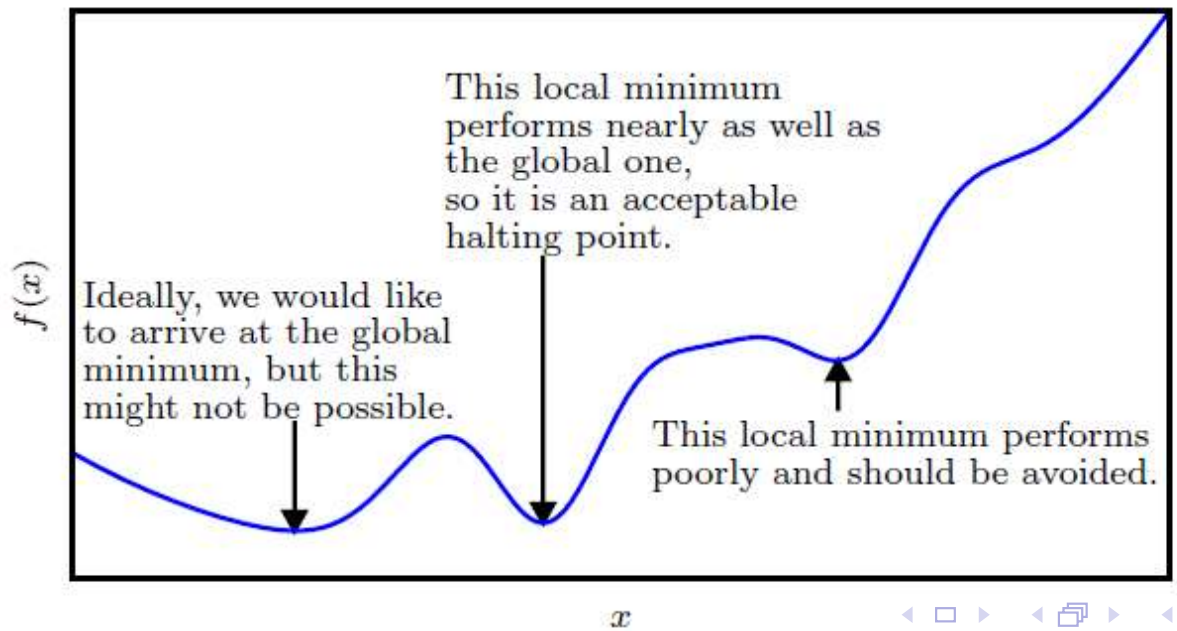
- 优化是什么？

求解目标函数在约束内的最小值

$$\min_{x} f(x)$$
$$s.t. \quad g_i(x) \leq 0, i = 1,2,\dots,m$$
$$h_j(x) = 0, j = 1,2,\cdots,n$$

机器学习的部分数学基础—优化

- 临界点 (critical point)、驻点 (stationary point)
  - $f'(x) = 0$
  - 可能是局部最小点、局部最大点、鞍点
- 局部最小点 x
  - 对于x的ϵ邻域上任意的c, $|x - c| < \epsilon$， $f(x) < f(c)$
- 局部最大点 x
  - 对于x的ϵ邻域上任意的c, $|x - c| < \epsilon$， $f(x) > f(c)$

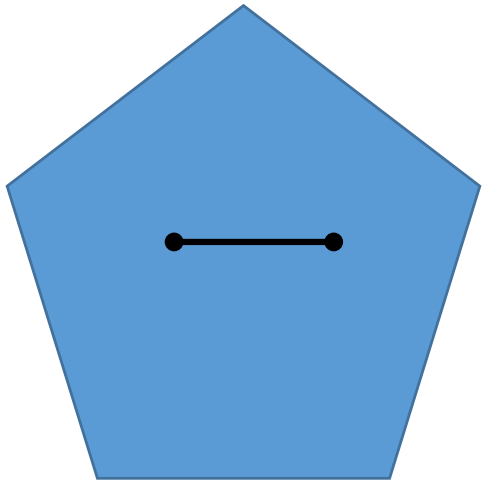


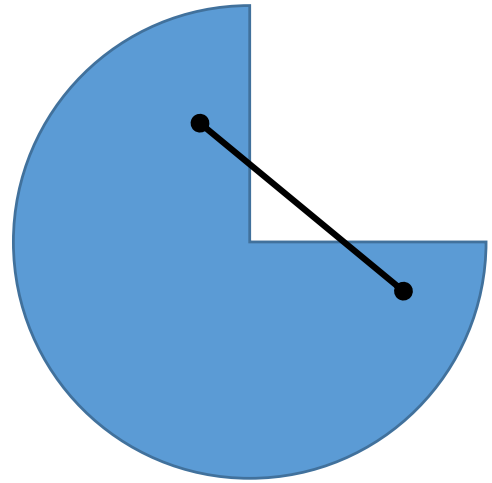
机器学习的部分数学基础—凸函数

给定集合$C \subseteq \mathbb{R}^n$。若$\forall\, x, y \in C$满足

$$\forall t \in [0,1], tx + (1-t)y \in C$$

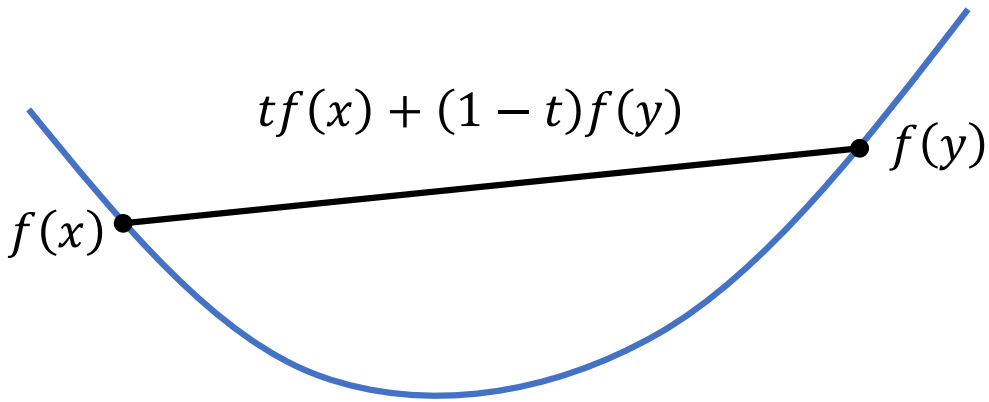
那么集合C为凸集





机器学习的部分数学基础—凸函数

给定一个函数$f: \mathbb{R}^n \mapsto R$。如果满足$dom(f)$是凸集而且 $\forall x, y \in dom(f)$,
$$\forall t \in [0,1], f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$
那么函数f是凸函数



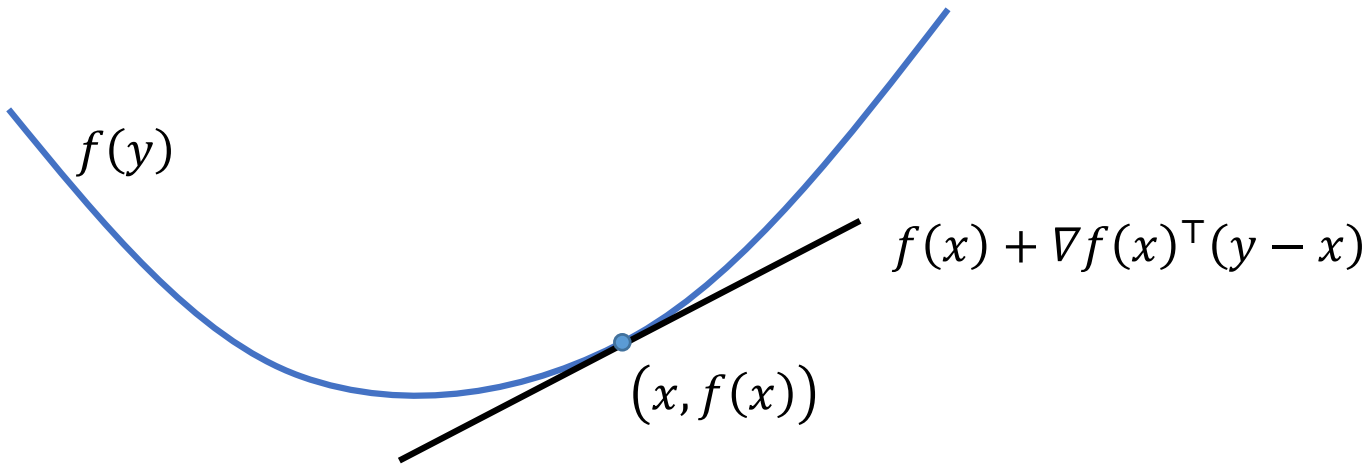
机器学习的部分数学基础—凸函数

- 指数函数 $\exp(ax)$
- 负对数函数 $-\log(x)$
- 反射函数 $\boldsymbol{a}^\top \boldsymbol{x} + b$
- 二次函数 $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + 2\boldsymbol{b}^\top \boldsymbol{x} + c$ （$\boldsymbol{A}$半正定）
- 范数 $\|\boldsymbol{x}\|_p = \sqrt[p]{\sum_i |x_i|^p}$
- 最大函数 $f(\boldsymbol{x}) = \max\{x_1, \cdots, x_n\}$
- Softplus $\log(1 + \exp(x))$
- LogSumExp $\log(\sum_i \exp(x_i))$
- LogDeterminant $-\log\det(X)$在半正定矩阵定义域上

机器学习的部分数学基础—凸函数

一阶条件

假设函数f可微，那么f是凸函数当且仅当 $\forall\, x, y \in dom(f)$，
$f(y) \geq f(x) + \nabla f(x)^{\mathsf{T}}(y - x)$



机器学习的部分数学基础—凸优化

二阶条件

假设函数f二阶可微，那么f是凸函数当且仅当 $\forall x \in dom(f)$，$\nabla^2 f(x) \succcurlyeq 0$，即海森矩阵半正定

机器学习的部分数学基础—凸优化

- 凸优化问题

$$\min_{x} f(\boldsymbol{x})$$
$$s.t. \quad g_i(\boldsymbol{x}) \leq 0, i = 1,2,\ldots,m$$
$$\boldsymbol{a}_i^{\top}\boldsymbol{x} = b, j = 1,2,\cdots,n$$

其中$f(\boldsymbol{x}), g_i(\boldsymbol{x})$是凸函数

机器学习的部分数学基础—凸优化

- 凸优化中，局部最优等价于全局最优

假设函数f可微凸函数，那么x 是 f 的全局最优当且仅当，

$$\nabla f(\boldsymbol{x}) = 0$$

证明：因为$\nabla f(\boldsymbol{x}) = 0$. 所以

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{x}) = f(\boldsymbol{x})$$

机器学习的部分数学基础—优化算法

- 无约束优化：梯度下降

- 目标： $\min_x f(x)$

while $\|\nabla f(x_t)\| > \delta$ do
$\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \alpha \nabla f(\boldsymbol{x}_t)$
end while

机器学习的部分数学基础—优化算法

- 有约束优化：拉格朗日乘子法

$$\min_{x} f(\boldsymbol{x})$$
$$s.t. \quad g_i(\boldsymbol{x}) \le 0, i = 1,2,\dots,m$$
$$h_j(\boldsymbol{x}) = 0, j = 1,2,\cdots,n$$

- 引入拉格朗日函数

$$L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) = f(\boldsymbol{x}) + \sum_i u_i\, g_i(\boldsymbol{x}) + \sum_j v_j\, h_j(\boldsymbol{x})$$

其中$u_i \ge 0$

机器学习的部分数学基础—优化算法

拉格朗日函数

$$L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{x}) + \sum_i u_i\, g_i(\boldsymbol{x}) + \sum_j v_j\, h_j(\boldsymbol{x})$$

其中$u_i \geq 0$

- 有如下结论

$\forall\, \boldsymbol{u} \geqslant 0, \boldsymbol{v}$, 和可行解$\boldsymbol{x}$, 满足

$$L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) \leq f(\boldsymbol{x})$$

机器学习的部分数学基础—优化算法

原问题

$$\min_{x} f(\boldsymbol{x})$$
$$s.t. \quad g_i(\boldsymbol{x}) \leq 0, i = 1,2,\ldots,m$$
$$h_j(\boldsymbol{x}) = 0, j = 1,2,\cdots,n$$

对偶问题

凸优化

$$\max_{\boldsymbol{u},\boldsymbol{v}} g(\boldsymbol{u},\boldsymbol{v})$$
$$s.t. \quad \boldsymbol{u} \succcurlyeq 0$$

其中$g(\boldsymbol{u},\boldsymbol{v}) = \min_{\boldsymbol{x}} L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v})$

机器学习的部分数学基础—优化算法

- 假设可行解集为C, $f^{\star}$为原问题最优解，那么满足

$$f^{\star} \geq \min_{\boldsymbol{x} \in C} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) \geq \min_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) = g(\boldsymbol{u}, \boldsymbol{v})$$

- 进一步得到弱对偶性

$$f^{\star} \geq g^{\star} = \max_{\boldsymbol{u}, \boldsymbol{v}} g(\boldsymbol{u}, \boldsymbol{v})$$

机器学习的部分数学基础—优化算法

- 强对偶性

$$f^\star = g^\star = \max_{\boldsymbol{u},\boldsymbol{v}} g(\boldsymbol{u},\boldsymbol{v})$$

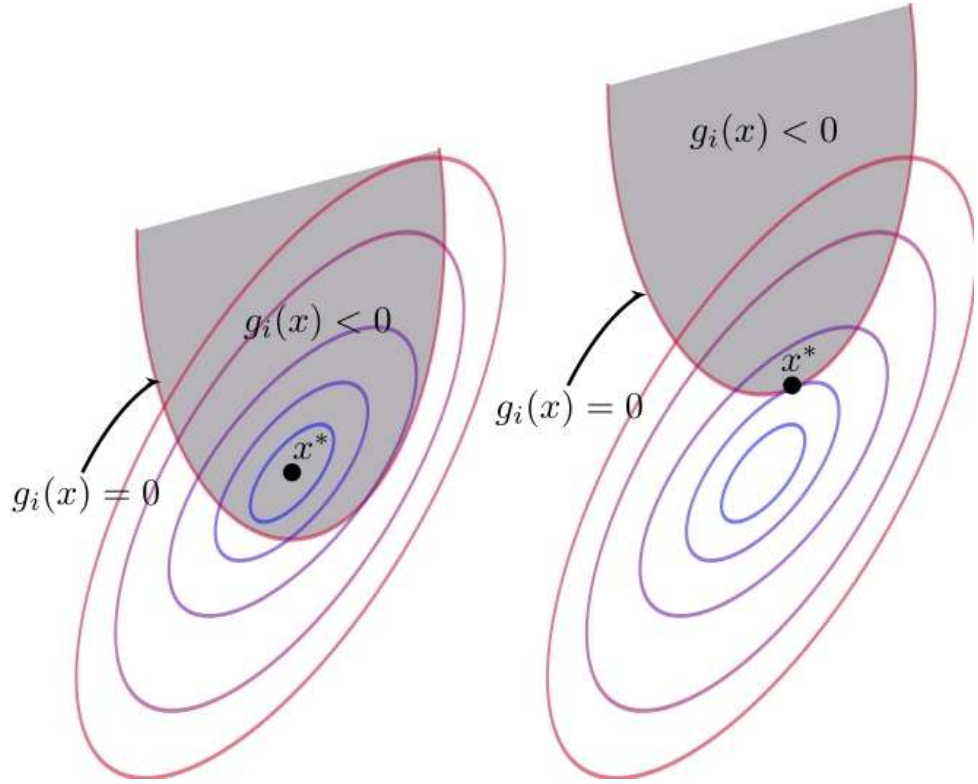
Slater条件

原问题为凸优化问题，且可行域中至少有一个点使得不等式约束严格成立

机器学习的部分数学基础—优化算法

- 最优解的必要条件：KKT条件

$$\nabla_{\mathbf{x}} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) = \mathbf{0}$$
$$g_i(\boldsymbol{x}) \leq 0$$
$$h_j(\boldsymbol{x}) = 0$$
$$\mu_i \geq 0$$
$$\mu_i g_i(\boldsymbol{x}) = 0$$



作业

- 1. 计算$\frac{\partial \ln \det(\boldsymbol{A})}{\partial x}$
- 2. 书习题1.2
- 3. 已知随机变量 $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2] \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, 计算$P(\boldsymbol{x}_1), P(\boldsymbol{x}_1|\boldsymbol{x}_2)$
- 4. 证明范数 $\|\boldsymbol{x}\|_p$是凸函数
- 5. 证明判定凸函数的0阶和1阶条件相互等价

$$\forall x, y \in dom(f), \forall t \in [0,1], f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$$

$$\Updownarrow$$

$$\forall\, x, y \in dom(f),\ \ f(y) \geq f(x) + \nabla f(x)^{\mathsf{T}}(y - x)$$

第二章：模型评估与选择

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

模型评估与选择

- 模型评估
  - 给定一个数据集，如何估计一个模型的“泛化” 能力?

- 模型选择
  - 给定一个数据集，如何根据“泛化” 能力，选出最好的模型或选出最好的参数配置

模型评估—经验误差与过拟合

误差：样本真实输出与预测输出之间的差异，可以是错误率

训练集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜	预测
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是	否
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是	否
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否	是
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否	否

训练误差
经验误差

测试集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜	预测
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是	否
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否	是
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否	是
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否	否

测试误差

模型评估—经验误差与过拟合

训练集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜	预测
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是	否
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是	否
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否	是

训练误差
经验误差

泛化误差：除训练集外的所有样本上的误差

测试集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜	预测
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是	否
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否	是
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否	是
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否	否

测试误差

模型评估—经验误差与过拟合

- 事先可能不知道新样本的特征，只能努力使经验误差最小化
- 很多时候虽然能在训练集上做到分类错误率为零，但多数情况下这样的学习器并不好

过拟合

学习器把训练样本学习的“太好”，将训练样本本身的特点当做所有样本的一般性质，导致泛化性能下降

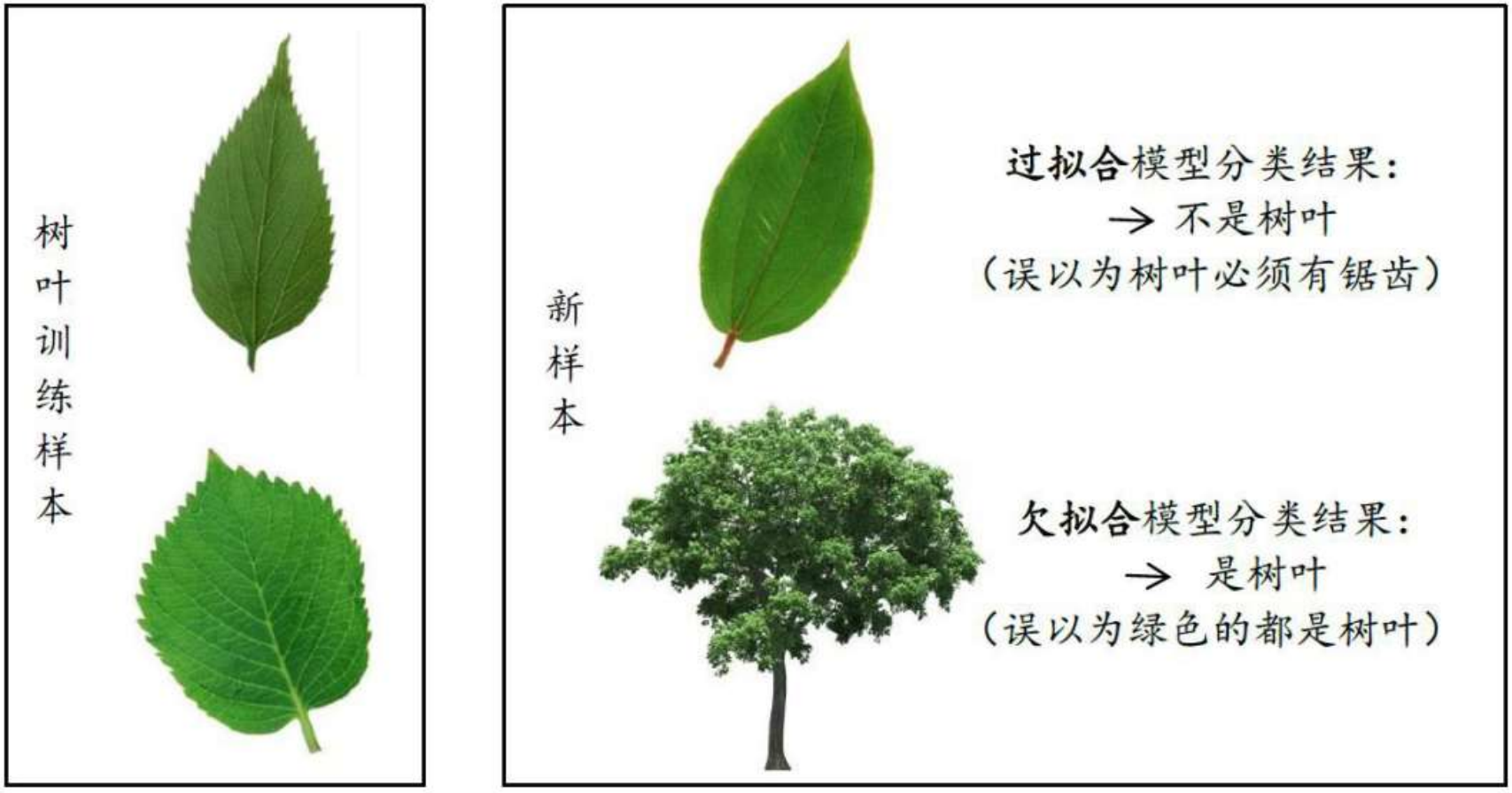
解决办法：优化目标加正则项；early stop

欠拟合

对训练样本的一般性质尚未学好

解决办法：决策树:拓展分支；神经网络：增加训练轮数

模型评估—过拟合与欠拟合



过拟合、欠拟合的直观类比

模型评估—评估方法

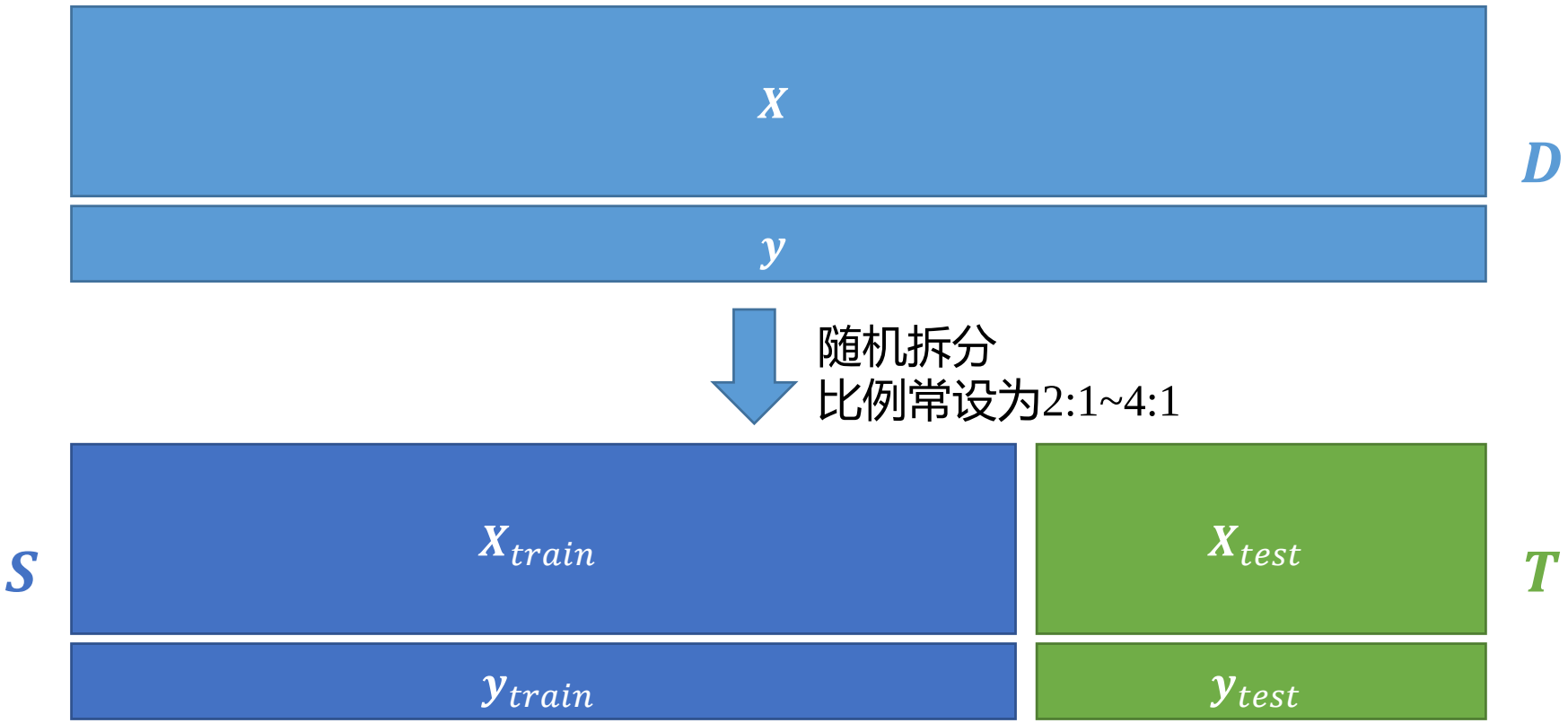
- 需要一个测试集来测试学习器对新样本的判别能力

- 假设测试集是从样本真实分布中独立采样获得，以测试集上的测试误差作为泛化误差的近似

测试集要和训练集中的样本尽量互斥，即测试样本尽量不在训练集中出现、未在训练集中使用过

评估方法—留出法 (hold-out)

- 假设数据集合为$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$



满足$D = S \cup T$且$S \cap T = \emptyset$

评估方法—留出法 (hold-out)

2:1随机拆分

训练集S

正负样本在训练集、测试集中的分布与数据集的不一致，**误差估计在可能会产生偏差**

测试集T

评估方法—留出法 (hold-out)

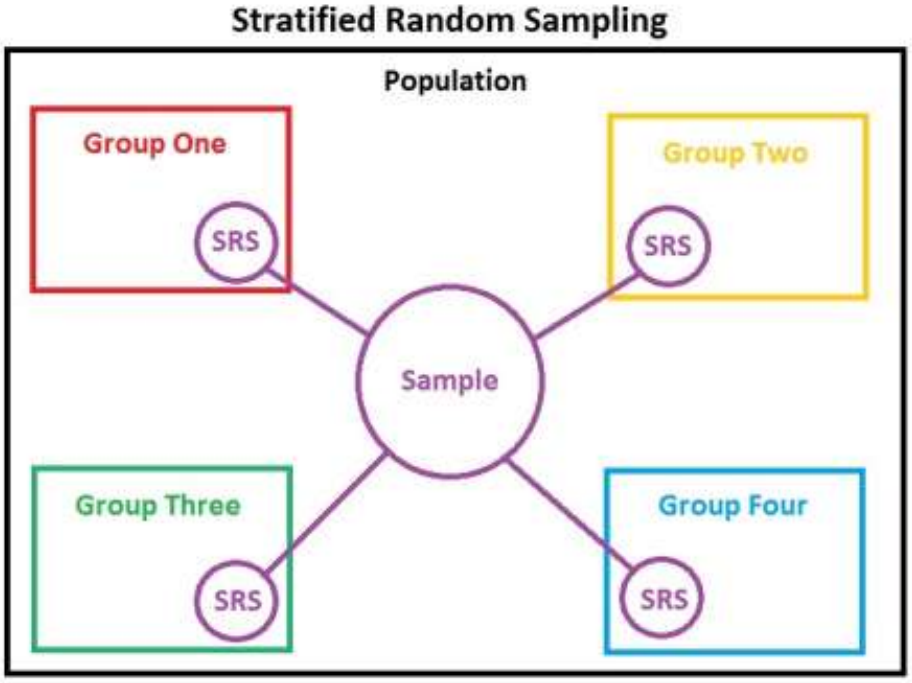
2:1随机拆分

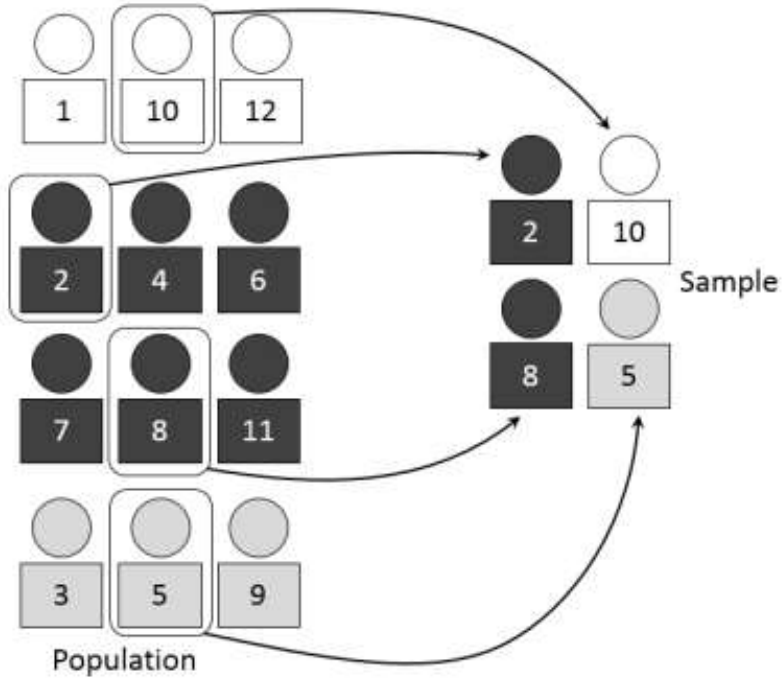
2:1随机拆分

训练集S

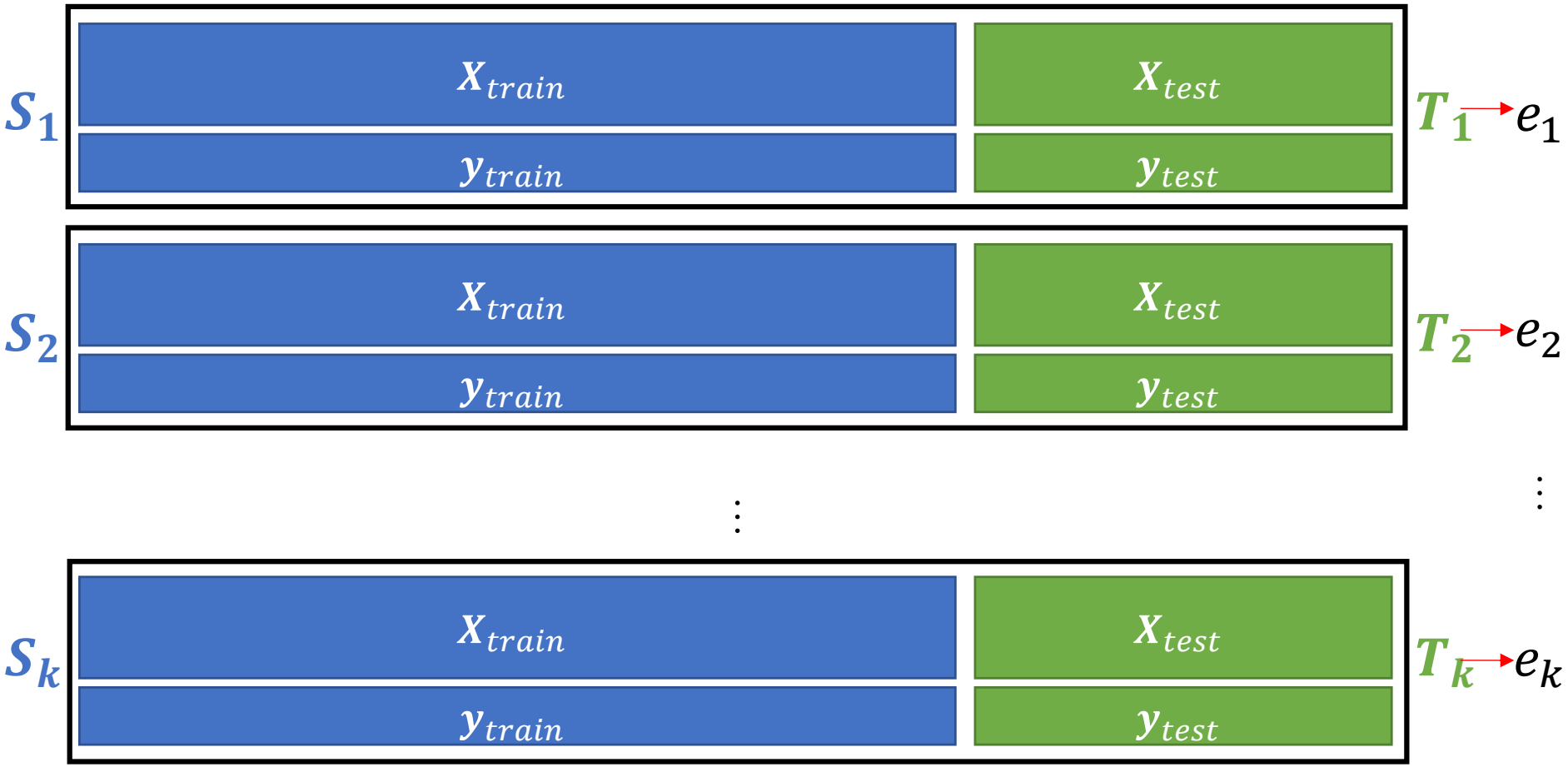
测试集T

评估方法—留出法 (hold-out)





评估方法—留出法 (hold-out)

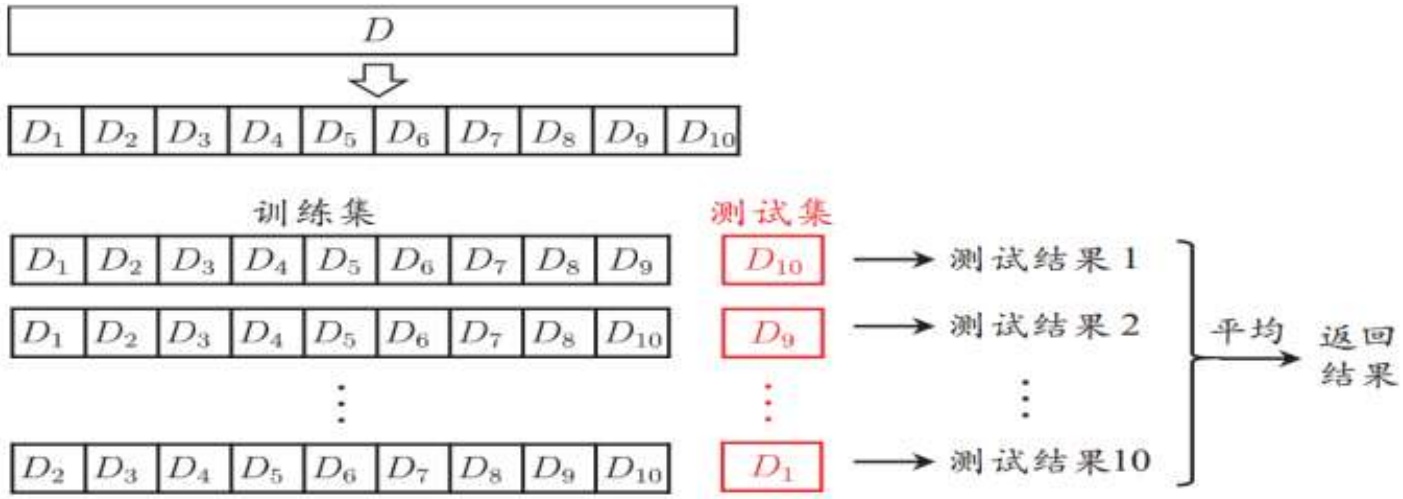


$$e = \frac{1}{K}(e_1 + e_2 + \cdots + e_k)$$

评估方法—交叉验证法

1. 将数据集分层采样划分为K个大小相似的互斥子集
2. 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
3. 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

评估方法—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”

评估方法—交叉验证法

- 当$k = m$，（每个样本一个集合），得到留一法

不受随机样本划分方式的影响
结果往往比较准确
当数据集比较大时，计算开销难以忍受

评估方法—自助法

- 希望评估D训练出的模型，但是实际评估模型使用了更小数据集

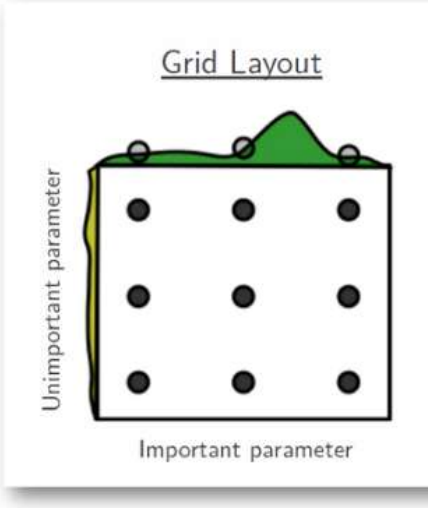
- 以自助采样法为基础，对数据集 D 有放回采样 m 次得到训练集 D'，$D \setminus D'$ 用做测试集。

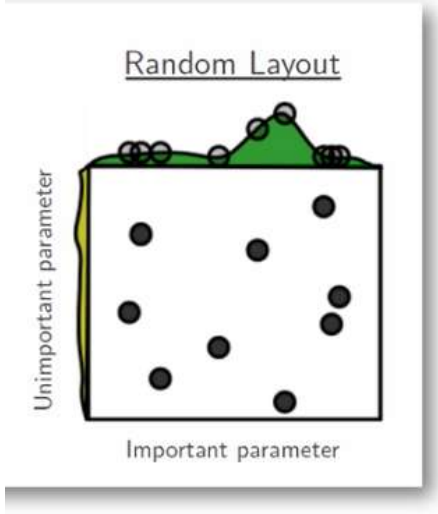
样本在m次采样中始终不被采样到的概率 $= \lim_{m\to\infty}\left(1-\frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368$

- 实际模型与预期模型都使用 m 个训练样本
- 约有1/3的样本没在训练集中出现
- 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
- 自助法在数据集较小、难以有效划分训练/测试集时很有用
- 由于改变了数据集分布可能引入估计偏差，在数据量足够时，留出法和交叉验证法更常用。

模型评估—调参和最终模型

- 大多数学习算法都有些参数需要设定，不同参数设置，导致学得模型的性能有显著差别
- 模型选择，包括学习算法选择和参数配置的设定，后者称为调参
- 调参的一般过程：
  - 将训练集划分为训练集和验证集
  - 通过网格法或随机法进行参数搜索，计算出验证集上的误差
  - 选出最佳的参数配置，在训练集上重新训练





模型评估—性能度量

- 性能度量是衡量模型泛化能力的评价标准
- 反映任务需求，使用不同的性能度量往往会导致不同的评判结果
- 在预测任务中，给定样例集$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$，评估学习器的性能$f$也即把预测结果$f(x)$和真实标记比较

回归任务最常用的性能度量是“均方误差”

$$E(f, D) = \frac{1}{m}\sum_{m}(f(\boldsymbol{x}_i) - y_i)^2$$

假设知道数据的分布，那么均方误差表达为

$$E(f, D) = \int_{\boldsymbol{x}\sim D}(f(\boldsymbol{x}) - y)^2\, p(\boldsymbol{x})d\boldsymbol{x}$$

模型评估—性能度量

- 对于分类任务,错误率和精度是最常用的两种性能度量:

错误率：分错样本占样本总数的比例

$$E(f,D)=\frac{1}{m}\sum_{i}\mathbb{I}(f(\boldsymbol{x}_i)\neq y_i)$$

精度：分对样本占样本总数的比率

$$acc(f,D)=\frac{1}{m}\sum_{i}\mathbb{I}(f(\boldsymbol{x}_i)=y_i)=1-E(f,D)$$

模型评估—性能度量

- 错误率和精度虽然常用，但不能满足所有任务需求
  - 比如，挑出的西瓜中有多少比例是好瓜，有多少比例的好瓜被挑选出来
  - 信息检索等场景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

- 查准率和查全率比错误率和精度更适合

混淆矩阵 (confusion matrix)

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)

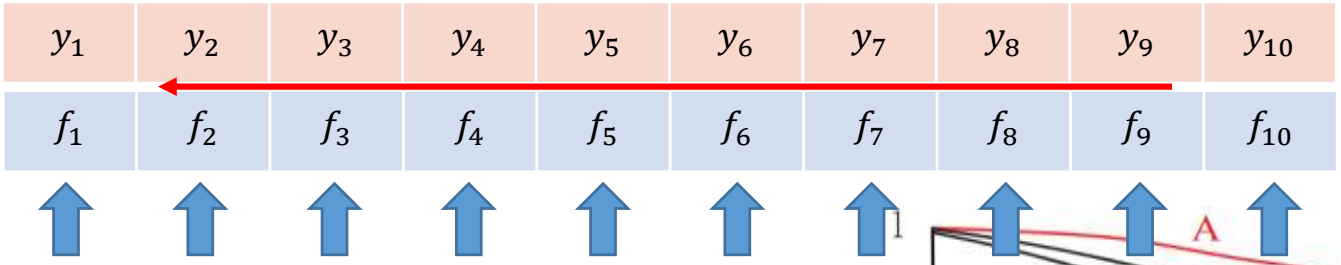
查全率 (Recall) $R = \frac{TP}{TP+FN}$

查准率(Precision) $P = \frac{TP}{TP+FP}$

模型评估—性能度量

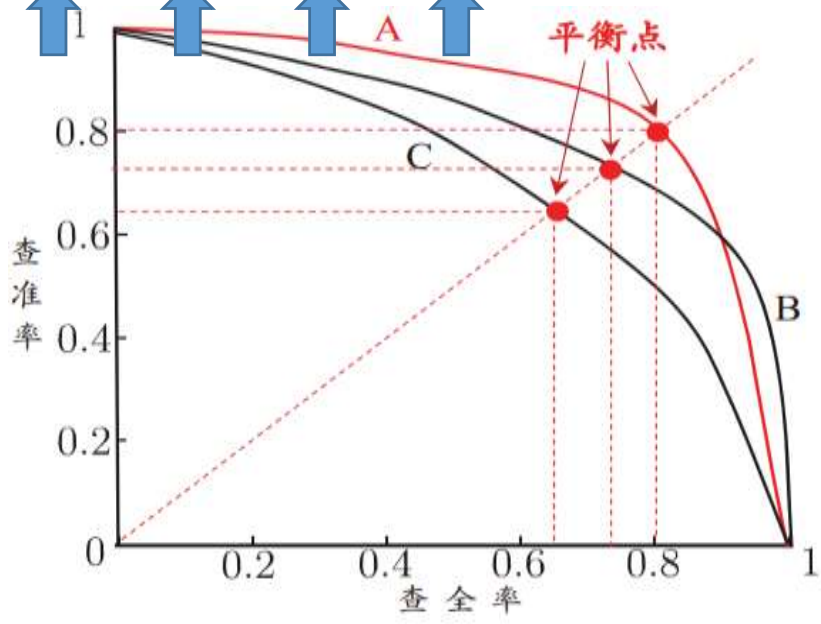
- 查准率和查全率是一堆矛盾的度量
  - 查准率高时，查全率低；查全率高时，查准率低

如何权衡这两个指标呢？



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

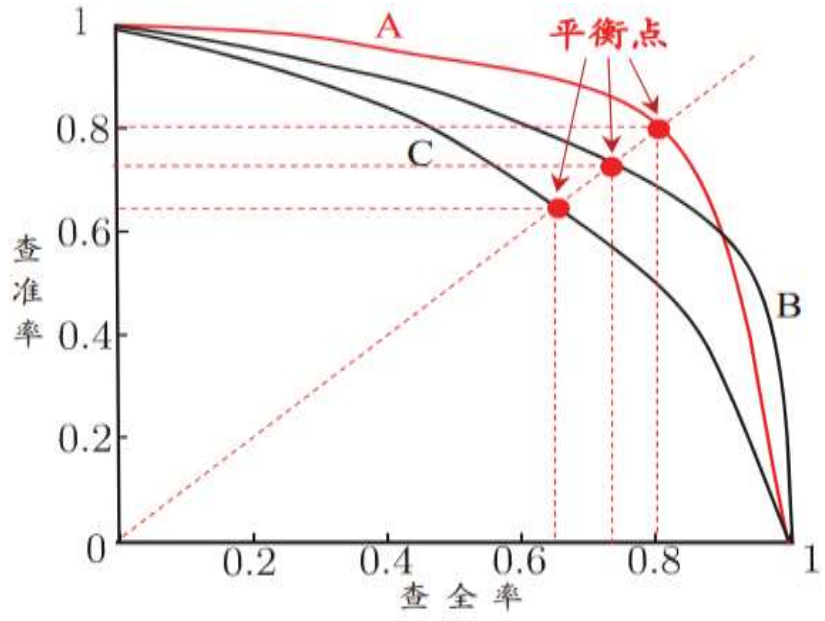
得到查准率-查全率曲线，简称“P-R曲线”



P-R曲线与平衡点示意图

模型评估—性能度量

- 如果一个学习器的P-R曲线被另一个学习器的曲线完全包住，那么后者性能更优
- 如果发生了交叉，则难以判断孰优孰劣
- 可以估算P-R曲线下的面积，但是估算比较困难
- 通过平衡点来权衡这两者指标



P-R曲线与平衡点示意图

平衡点是曲线上“查准率=查全率”时的取值，可用来用于度量P-R曲线有交叉的分类器性能高低

模型评估—性能度量

- 比P-R曲线平衡点更用常用的是F1度量：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{1}{\frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)}$$

- 比F1更一般的形式 F_β ，

$$F_\beta = \frac{\left(1 + \beta^2\right) \times P \times R}{\beta^2 P + R} = \frac{\frac{1}{\beta} + \beta}{\left(\frac{1}{\beta}\frac{1}{P} + \beta\frac{1}{R}\right)}$$

$\beta = 1$：标准的F1
$\beta > 1$：偏重查全率
$\beta < 1$：偏重查准率

模型评估—性能度量

- 如果有多个二分类混淆矩阵
  - 多次训练/测试、多个数据集训练/测试、多分类中每两两类别的组合

先在各个混淆矩阵上分别计算出查准率和查全率，记为$(P_1, R_1), (P_2, R_2), \cdots, (P_n, R_n)$。再计算均值，得到宏查准率（macro-P）、宏查全率（macro-R）和相应的宏F1（macro-F1）。

$$\text{macro-P} = \frac{1}{n}\sum_i P_i$$

$$\text{macro-R} = \frac{1}{n}\sum_i R_i$$

先在各个混淆矩阵对应元素平均，得到TP、FP、TN、FN的平均值，在基于平均值计算微查准率（micro-P）、微查全率（micro-R）和相应的微F1（micro-F1）

$$\text{micro-P} = \frac{\overline{TP}}{\overline{TP}+\overline{FP}}$$

$$\text{micro-R} = \frac{\overline{TP}}{\overline{TP}+\overline{FN}}$$

模型评估—性能度量

- 类似P-R曲线，根据学习器的预测结果对样例排序，并逐个作为正例进行预测，以“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（Receiver Operating Characteristics）”.

混淆矩阵 (confusion matrix)

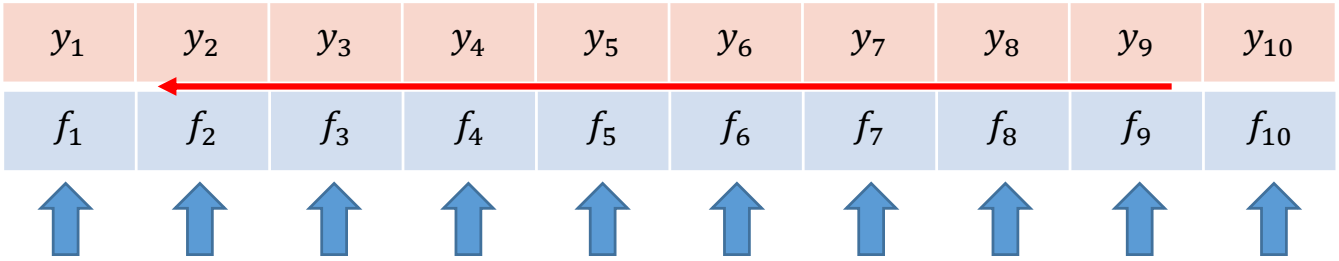
真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

$$TPR = \frac{TP}{TP + FN}$$ 真正例率

$$FPR = \frac{FP}{FP + TN}$$ 假正例率

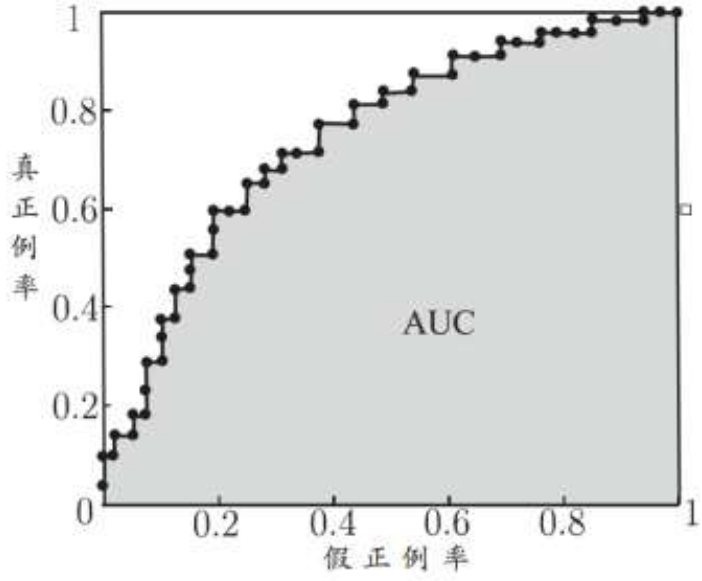
模型评估—性能度量

ROC图的绘制：



给定m^+个正例和m^-个负例，根据学习器预测结果对样例进行排序。

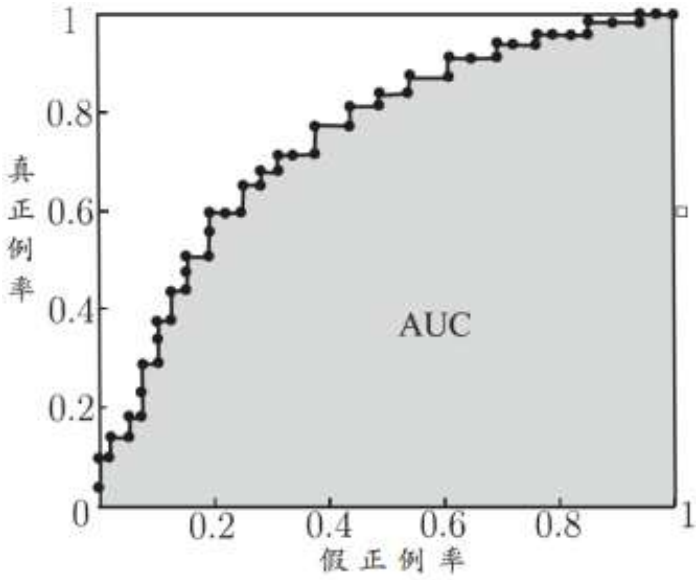
将分类阈值设为每个样例的预测值，当前标记点坐标为(x, y)，当前若为真正例，则对应标记点的坐标为$\left(x, y+\frac{1}{m^+}\right)$；当前若为假正例，则对应标记点的坐标为$\left(x+\frac{1}{m^{-1}}, y\right)$。然后用线段连接相邻点



基于有限样例绘制的 ROC 曲线与 AUC

模型评估—性能度量

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$\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$的点按序连接而形成 $(x_1 = 0, x_m = 1)$，则AUC可估算为

$$\mathrm{AUC} = \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1} - x_i) \times (y_i + y_{i+1})$$

AUC衡量了样本预测的排序质量

性能度量—代价敏感错误率

- 现实任务中不同类型的错误所造成的后果很可能不同，为了权衡不同类型错误所造成的不同损失，可为错误赋予“非均等代价”。

- 以二分类为例，根据领域知识设定“代价矩阵”，如下表所示，$cost_{ij}$表示将第i类样本预测为第j类样本的代价。一般，$cost_{ii}=0$

真实类别	预测类别	
	第0类	第1类
第0类	0	$cost_{01}$
第1类	$cost_{10}$	0

- 损失程度相差越大，$cost_{01}$与 $cost_{10}$值的差别越大。

性能度量—代价敏感错误率

- 在非均等代价下，不再最小化错误次数，而是最小化“总体代价”，则“代价敏感”错误率相应的为：

$$E(f;D,cost)=\frac{1}{m}\left(\sum_{x_i\in D^+}\mathbb{I}(f(x_i)\neq y_i)\times cost_{01}+\sum_{x_i\in D^-}\mathbb{I}(f(x_i)\neq y_i)\times cost_{10}\right)$$

性能度量—代价曲线

- 在非均等代价下，ROC曲线不能直接反映出学习器的期望总体代价，而“代价曲线”可以

- 代价曲线的横轴是取值为[0,1]的正例概率代价

$$P(+)cost = \frac{p \times cost_{01}}{p \times cost_{01} + (1-p) \times cost_{10}}$$

- 纵轴是取值为[0,1]的归一化代价

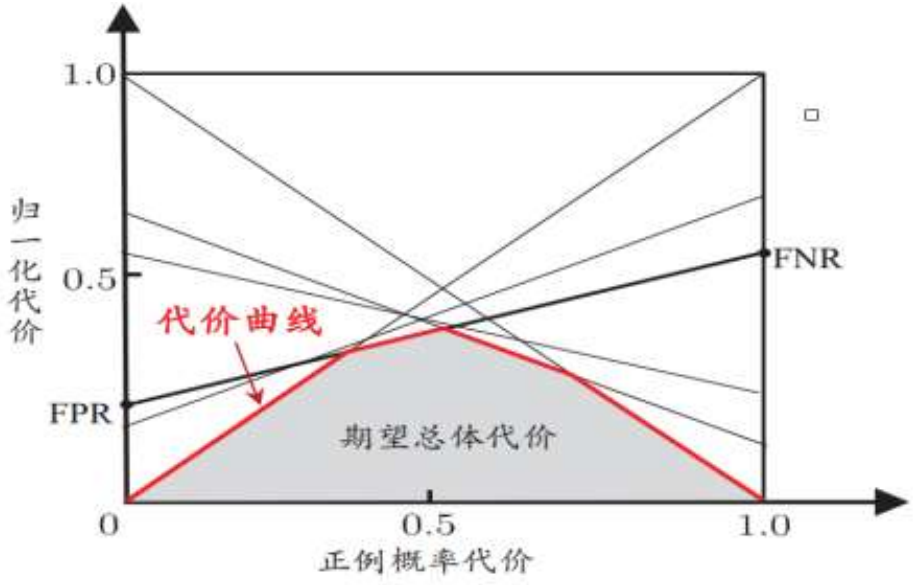
$$\mathrm{cost}_{\mathrm{norm}} = \mathrm{FNR} \times P(+)cost + FPR \times (1 - P(+)cost)$$

$FNR = 1 - TPR$ 假负例率

性能度量—代价曲线

代价曲线图的绘制：

- ROC曲线上每个点对应了代价曲线上的一条线段，设ROC曲线上点的坐标为(TPR,FPR),则可相应计算出FNR,然后在代价平面上绘制一条从(0,FPR)到(1,FNR)的线段，线段下的面积即表示了该条件下的期望总体代价
- 将ROC曲线上的每个点转化为代价平面上的一条线段，然后取所有线段的下界，围成的面积即为所有条件下学习器的期望总体代价。



代价曲线与期望总体代价

模型评估—比较检验

关于性能比较

测试性能并不等于泛化性能
测试性能随着测试集的变化而变化
很多机器学习算法本身有一定的随机性

直接选取相应评估方法在相应度量下比大小的方法不可取！

假设检验为学习器性能比较提供了重要依据，基于其结果我们可以推断出若在测试集上观察到学习器A比B好，则A的泛化性能是否在统计意义上优于B，以及这个结论的把握有多大。

模型评估—二项检验

设泛化错误率为 ϵ，若测试错误率为 $\hat{\epsilon}$，对$\epsilon \leq \epsilon_0$进行假设检验

- 假定测试样本从样本总体分布中独立采样而来，可以使用“二项检验”对 $\epsilon \leq \epsilon_0$ 进行假设检验

- 求解$1-\alpha$概率内能看到的最大错误概率

$$\bar{\epsilon} = \min \epsilon \ \ s.t. \sum_{i=\epsilon \times m+1}^{m} \binom{m}{i} \epsilon_0^i (1-\epsilon_0)^{m-i} < \alpha$$

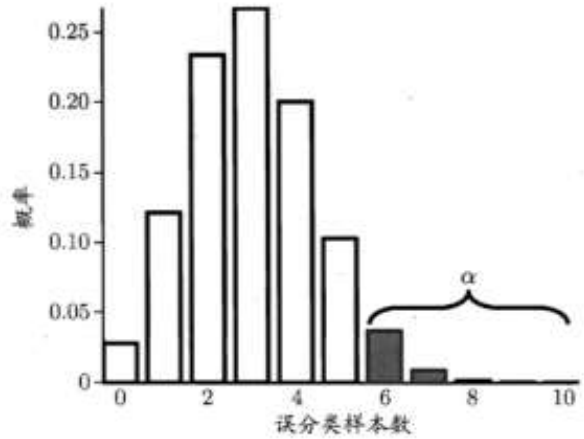


图 2.6 二项分布示意图($m = 10, \epsilon = 0.3$)

- 若$\hat{\epsilon} < \bar{\epsilon}$，则在$\alpha$显著度下，假设$\epsilon \leq \epsilon_0$不能被拒绝
- 否则，该假设被拒绝，即在α显著度下认为$\epsilon > \epsilon_0$

模型评估—t检验

设泛化错误率为 ϵ，若k个测试错误率为$\hat{\epsilon}_1, \hat{\epsilon}_2, \cdots, \hat{\epsilon}_k$，对$\epsilon = \epsilon_0$进行假设检验

- 多次留出法或交叉验证法进行训练/测试时可使用“t检验”
- 平均测试错误率μ和方差σ^2

$$\mu = \frac{1}{k}\sum_i \hat{\epsilon}_i$$

$$\sigma^2 = \frac{1}{k-1}\sum_i (\hat{\epsilon}_i - \mu)^2$$

- 考虑到这k个测试错误率是泛化错误率ϵ_0的独立采样，则

$$\tau_t = \frac{\sqrt{k}(\mu - \epsilon_0)}{\sigma}$$

- 服从自由度为k-1的t分布

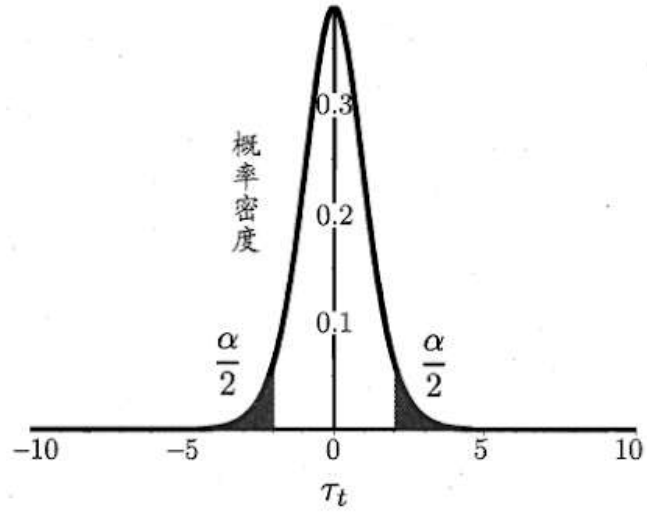
模型评估—t检验

设泛化错误率为 ϵ，若k个测试错误率为$\hat{\epsilon}_1, \hat{\epsilon}_2, \cdots, \hat{\epsilon}_k$，对$\epsilon = \epsilon_0$进行假设检验

$$\tau_t = \frac{\sqrt{k}(\mu - \epsilon_0)}{\sigma}$$

服从自由度为k-1的t分布

- 考虑双边假设前提下，求解$1 - \alpha$概率内能看到的最大错误概率，对应右图阴影部分



- 若$\tau_t \in [t_{-\alpha/2}, t_{\alpha/2}]$，则在$\alpha$显著度下，假设$\mu = \epsilon_0$不能被拒绝，即泛化误差率为$\epsilon_0$
- 否则，该假设被拒绝，即泛化误差率与ϵ_0有显著不同

模型评估—交叉验证t检验

现实任务中，更多时候需要对不同学习器性能进行比较

- 对两个学习器A和B，若k折交叉验证得到的测试错误率分别为$\epsilon_1^A,\cdots,\epsilon_k^A$和$\epsilon_1^B,\cdots,\epsilon_k^B$，可用k折交叉验证“成对t检验”进行检验
- 先对每个结果求差$\Delta_i=\epsilon_i^A-\epsilon_i^B$
- 计算差值的均值μ和方差σ^2

•在显著度α下，若变量$\tau_t=\left|\frac{\sqrt{k}\mu}{\sigma}\right|$小于临界值$t_{\alpha/2,k-1}$，则假设不能被拒绝，即两个学习器没有显著差别。

否则认为两个学习器有显著差别，平均错误率小的那个学习器性能更优。

模型评估—交叉验证t检验

假设检验的前提是测试错误率为泛化错误率的独立采样

然而由于样本有限，使用交叉验证导致训练集重叠，测试错误率并不独立，从而过高估计假设成立的概率

为缓解这一问题，可采用“5*2交叉验证”法.

模型评估—5*2交叉验证t检验

- 5*2折交叉验证就是做5次二折交叉验证
- 每次二折交叉验证之前将数据打乱，使得5次交叉验证中的数据划分不重复。
- 为缓解测试数据错误率的非独立性，仅计算第一次2折交叉验证结果的平均值$\mu = 0.5\left(\Delta_1^1 + \Delta_1^2\right)$和每次二折实验计算得到的方差 $\sigma_i^2 = \left(\Delta_i^1 - \frac{\Delta_i^1+\Delta_i^2}{2}\right)^2 + \left(\Delta_i^2 - \frac{\Delta_i^1+\Delta_i^2}{2}\right)^2$
- 变量$\tau_t = \frac{\mu}{\sqrt{0.2\sum_{i=1}^{5}\sigma_i^2}}$服从自由度为5的t分布

模型评估—McNemar检验

- 对于二分类问题，留出法不仅可以估计出学习器A和B的测试错误率，还能获得两学习器分类结果的差别，如下表所示

两学习器分类差别列联表

算法 B	算法 A	
	正确	错误
正确	e_{00}	e_{01}
错误	e_{10}	e_{11}

对两学习器性能相同进行假设 等价于对$e_{01} = e_{10}$进行假设检验

McNemar 检验考虑变量

$$\tau_{\chi^2} = \frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}}$$

该变量服从自由度为1的卡方分布

模型评估—Friedman检验

- 交叉验证t检验和McNemar检验都是在一个数据集上比较两个算法的性能
- Friedman检验在一组数据集上对多个算法进行比较

- 假设用D_1, D_2, D_3, D_4四个数据集对算法 A, B, C进行比较
- 使用留出法或交叉验证法得到每个算法在每个数据集的测试结果
- 然后在每个数据集上根据性能好坏排序，并赋序值1,2,…
- 若算法性能相同则平分序值,继而得到每个算法的平均序值

算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

模型评估—Friedman检验

- 由平均序值进行Friedman检验来判断这些算法是否性能都相同

算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

假设在N数据集上比较k个算法，令r_i 表示第i个算法的平均序值。**若这些算法性能都相同，则它们的平均序值应当相同**

若不考虑平分序值情况，那么

$$\mathrm{E}[r_i] = \frac{k+1}{2}, \mathrm{var}(r_i) = \frac{\left(k^2-1\right)}{12N}$$

当k和N都较大时，变量

$$\tau_{\chi^2} = \frac{12N}{k(k+1)}\left(\sum_{i=1}^{k} r_i^2 - \frac{k(k+1)^2}{4}\right)$$

服从自由度为k-1的卡方分布

模型评估—Nemenyi后续检验

- 若“所有算法的性能相同”这个假设被拒绝，说明算法的性能显著不同，此时可用Nemenyi后续检验进一步区分算法。

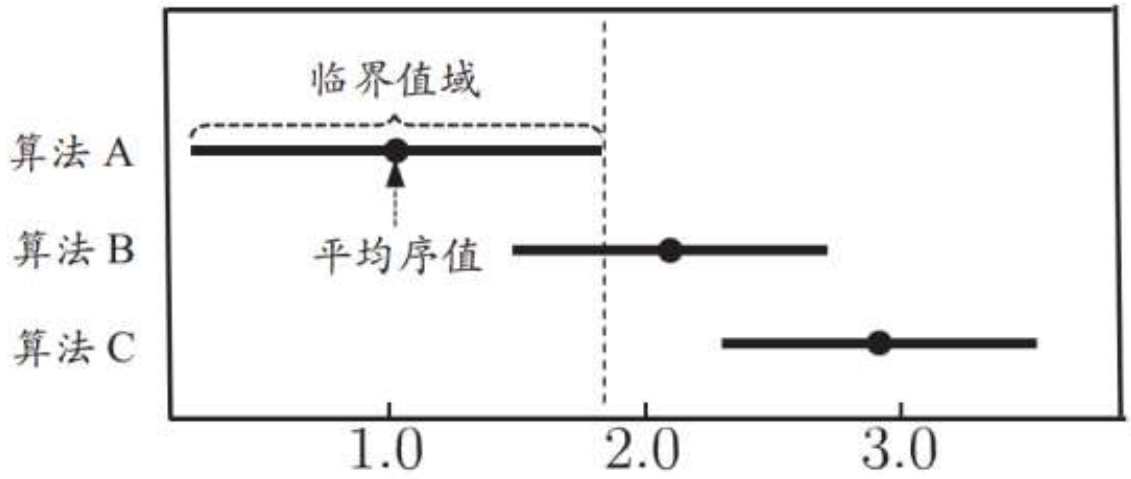
- Nemenyi检验计算平均序值差别的临界阈值

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

- 如果两个算法的平均序值之差超出了临界阈值CD，则以相应的置信度拒绝“两个算法性能相同”这一假设。

模型评估—Nemenyi后续检验

- 根据上例的序值结果可绘制如下Friedman检验图，横轴为平均序值，每个算法圆点为其平均序值，线段为临界阈值的大小。



- 若两个算法有交叠(A和B)，则说明没有显著差别
- 否则有显著差别(A和C),算法A明显优于算法C

模型评估—泛化性能解释

- 通过实验可以估计学习算法的泛化性能，而“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华错误率进行拆解。

- 对测试样本$\boldsymbol{x}$，令y_D 为 $\boldsymbol{x}$ 在数据集中的标记，y为 $\boldsymbol{x}$ 的真实标记，$f(x;D)$为训练集 D 上学得模型 f 在 $\boldsymbol{x}$ 上的预测输出。

- 以回归任务为例

学习算法的期望预期为：$\bar{f}(\boldsymbol{x}) = \mathbb{E}_D[f(\boldsymbol{x};D)]$

使用样本数目相同的不同训练集产生的方差为：$\mathrm{var}(\boldsymbol{x}) = \mathbb{E}_D\left[\left(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x})\right)^2\right]$

输出噪声为：$\varepsilon^2 = \mathbb{E}_D[(y_D - y)^2]$

期望输出与真实标记的差别称为偏差为：$bias^2(\boldsymbol{x}) = \left(\bar{f}(\boldsymbol{x}) - y\right)^2$

模型评估—泛化性能解释

- 为便与讨论，假定噪声期望为0，也即 $\mathbb{E}_D[y_D - y] = 0$，

对泛化误差分解

$$
\begin{aligned}
E(f, D) &= \mathbb{E}_D[(f(\boldsymbol{x}; D) - y_D)^2] \\
&= \mathbb{E}_D\left[\left(f(\boldsymbol{x}; D) - \bar{f}(\boldsymbol{x}) + \bar{f}(\boldsymbol{x}) - y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f(\boldsymbol{x}; D) - \bar{f}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_D\left[\left(\bar{f}(\boldsymbol{x}) - y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f(\boldsymbol{x}; D) - \bar{f}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_D\left[\left(\bar{f}(\boldsymbol{x}) - y + y - y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f(\boldsymbol{x}; D) - \bar{f}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_D\left[\left(\bar{f}(\boldsymbol{x}) - y\right)^2\right] + \mathbb{E}_D[(y - y_D)^2]
\end{aligned}
$$

$var(\boldsymbol{x})$　　$bias^2(\boldsymbol{x})$　　ε^2

泛化误差可分解为偏差、方差与噪声之和

模型评估—泛化性能解释

$$E(f,D) = \mathbb{E}_D\left[\left(f(x;D) - \bar{f}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_D\left[\left(\bar{f}(\boldsymbol{x}) - y\right)^2\right] + \mathbb{E}_D[(y - y_D)^2]$$

$var(\boldsymbol{x})$ $bias^2(\boldsymbol{x})$ ε^2

- 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

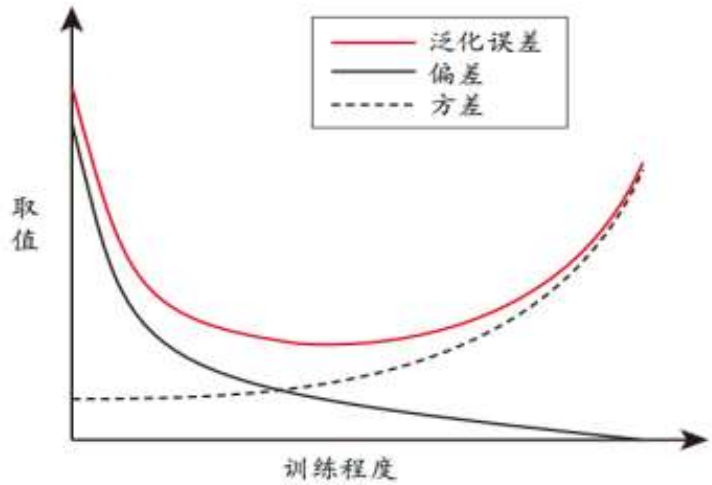
泛化性能是由学习算法的能力、数据的充分性以及学习任务本身的难度所共同决定的。给定学习任务为了取得好的泛化性能，需要使偏差小(充分拟合数据)而且方差较小(减少数据扰动产生的影响)。

模型评估—泛化性能解释

- 一般来说，偏差与方差是有冲突的，称为偏差-方差窘境。

- 如右图所示，假如我们能控制算法的训练程度：

  - 在训练不足时，学习器拟合能力不强，训练数据的扰动不足以使学习器的拟合能力产生显著变化，此时偏差主导泛化错误率
  - 随着训练程度加深，学习器拟合能力逐渐增强，方差逐渐主导泛化错误率
  - 训练充足后，学习器的拟合能力非常强，训练数据的轻微扰动都会导致学习器的显著变化，若训练数据自身非全局特性被学到则会发生过拟合。



泛化误差与偏差、方差的关系示意图

作业

- 习题2.2
- 习题2.4
- 习题2.5
- 习题2.9

第三章：线性模型

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

基本形式

- 线性模型一般形式

$$f(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b$$

$\boldsymbol{x} = (x_1; x_2; \cdots, x_d)$是由属性描述的示例，其中$x_i$是$\boldsymbol{x}$在第$i$个属性上的取值

- 向量形式

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathsf{T}} \boldsymbol{x} + b$$

$\boldsymbol{w} = (w_1; w_2; \cdots, w_d)$是属性的权重

线性模型优点

- 形式简单、易于建模
- 可解释性
- 非线性模型的基础：引入层级结构或高维映射

- 一个例子
  - 综合考虑色泽、根蒂和敲声来判断西瓜好不好
  - 其中根蒂的系数最大，表明根蒂最要紧；而敲声的系数比色泽大，说明敲声比色泽更重要

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

线性回归

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

- 线性回归目标
  - 学得一个线性模型以尽可能准确地预测实值输出标记

- 离散属性处理
  - 有“序”关系
    - 连续化为连续值
  - 无“序”关系
    - 有k个属性值，则转换为k维向量

线性回归

- 单一属性的线性回归目标

$$f(x_i) = w\,x_i + b \text{ 使得 } f(x_i) \approx y_i$$

- 参数/模型估计：最小二乘法（least square method）

$$(w^\star, b^\star) = \arg\min_{w,b} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$

$$= \arg\min_{w,b} \sum_{i=1}^{m} (wx_i + b - y_i)^2$$

线性回归 - 最小二乘法

- 最小化均方误差

$$E(w,b) = \sum_{i=1}^{m}(y_i - wx_i - b)^2$$

- 分别对w和b求导，可得

$$\frac{\partial E(w,b)}{\partial w} = 2\left(w\sum_i x_i^2 - \sum_i (y_i - b)x_i\right)$$

$$\frac{\partial E(w,b)}{\partial b} = 2\left(mb - \sum_i (y_i - wx_i)\right)$$

线性回归 - 最小二乘法

- 令导数梯度等于0，得到闭形式解

$$\frac{\partial E(w,b)}{\partial w} = w\sum_i x_i^2 - \sum_i (y_i - b)x_i$$

$$= w\sum_i x_i^2 - \sum_i (y_i - \bar{y} + w\bar{x})x_i$$

$$= w\left(\sum_i x_i^2 - \bar{x}\sum_i x_i\right) - \sum_i (y_i - \bar{y})x_i$$

$$= w\left(\sum_i x_i^2 - \bar{x}\sum_i x_i\right) - \left(\sum_i y_i x_i - \sum_i \bar{y}x_i\right)$$

$$= w\left(\sum_i x_i^2 - \bar{x}\sum_i x_i\right) - \left(\sum_i y_i x_i - \sum_i y_i\bar{x}\right)$$

$$= w\left(\sum_i x_i^2 - \bar{x}\sum_i x_i\right) - \sum_i y_i(x_i - \bar{x})$$

$$b = \frac{1}{m}\sum_i (y - wx_i) = \bar{y} - w\bar{x}$$

$$w = \frac{\sum_i y_i(x_i - \bar{x})}{\left(\sum_i x_i^2 - \frac{1}{m}\left(\sum_i x_i\right)^2\right)}$$

多元线性回归

- 给定数据集

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

$$\text{其中}\boldsymbol{x}_i = (x_{i1}; x_{i2}; \cdots, x_{id}),\ y_i \in \mathbb{R}$$

- 多元线性回归目标

$$f(\boldsymbol{x_i}) = \boldsymbol{w}^{\mathsf{T}}\ \boldsymbol{x}_i + b \text{ 使得 } f(\boldsymbol{x}_i) \approx y_i$$

多元线性回归

- 把$\boldsymbol{w}$和b吸收入向量形式 $\hat{\boldsymbol{w}} = (\boldsymbol{w}; b)$，数据集表示为

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ x_{21} & x_{22} & \cdots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^{\mathsf{T}} & 1 \\ \boldsymbol{x}_2^{\mathsf{T}} & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_m^{\mathsf{T}} & 1 \end{pmatrix}$$

$$\boldsymbol{y} = (y_1; y_2; \cdots; y_m)$$

多元线性回归 - 最小二乘法

- 最小二乘法（least square method）

$$\hat{\boldsymbol{w}}^{\star} = \arg\min_{\hat{\boldsymbol{w}}} \|\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}\|_2^2 = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}})^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}})$$

$E(\hat{\boldsymbol{w}})$

- 求$E(\hat{\boldsymbol{w}})$关于变量$\hat{\boldsymbol{w}}$的导数得到

$$\nabla_{\hat{\boldsymbol{w}}} E(\hat{\boldsymbol{w}}) = 2\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{y})$$

多元线性回归 - 满秩讨论

- $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$是满秩矩阵或正定矩阵，则

$$\nabla_{\hat{\boldsymbol{w}}} E(\hat{\boldsymbol{w}}) = 2\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\hat{\boldsymbol{w}} - \boldsymbol{y}) = 0 \quad \Rightarrow \quad \hat{\boldsymbol{w}}^{\star} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

- 把$\hat{\boldsymbol{w}}^{\star}$代回$f(\boldsymbol{x}_i)$，线性回归模型为

$$f(\hat{\boldsymbol{x}}_i) = \boldsymbol{x}_i(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

- 如果$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$不是满秩矩阵
  - 根据归纳偏好选择解
  - 引入正则化

一元线性回归

- 重新考虑一个特征的情形

$$\boldsymbol{X} = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{pmatrix} \qquad \boldsymbol{X}^\top \boldsymbol{X} = \begin{pmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & m \end{pmatrix} \qquad \boldsymbol{X}^\top \boldsymbol{y} = \begin{pmatrix} \sum_i x_i y_i \\ \sum_i y_i \end{pmatrix}$$

$$(\boldsymbol{X}^\top \boldsymbol{X})^{-1} = \frac{1}{m\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \begin{pmatrix} m & -\sum_i x_i \\ -\sum_i x_i & \sum_i x_i^2 \end{pmatrix}$$

$$(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y} = \frac{1}{m\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \begin{pmatrix} m\sum_i x_i y_i - \sum_i x_i \sum_j y_j \\ \sum_i x_i^2 \sum_j y_j - \sum_i x_i y_i \sum_j x_j \end{pmatrix}$$

一元线性回归

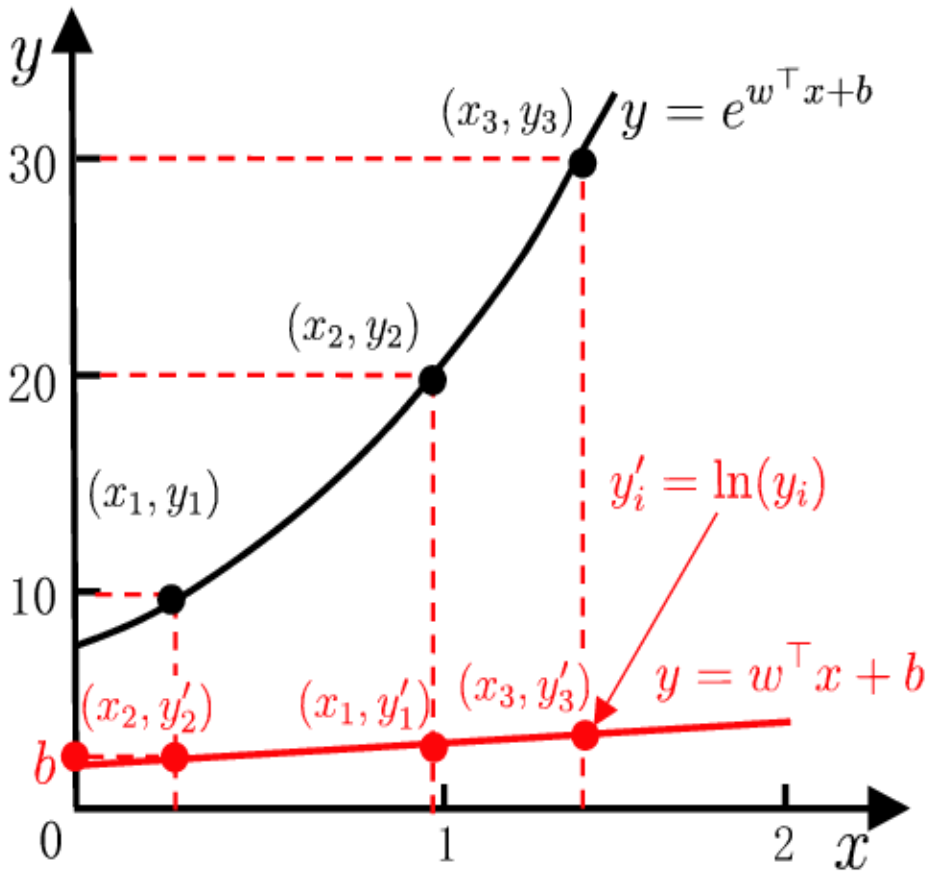
$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^{\top}\boldsymbol{X})^{-1}\boldsymbol{X}^{\top}\boldsymbol{y} = \frac{1}{m\sum_i x_i^2 - \left(\sum_i x_i\right)^2}\begin{pmatrix} m\sum_i x_i y_i - \sum_i x_i \sum_j y_j \\ \sum_i x_i^2 \sum_j y_j - \sum_i x_i y_i \sum_j x_j \end{pmatrix}$$

$$w = \frac{m\sum_i x_i y_i - \sum_i x_i \sum_j y_j}{m\sum_i x_i^2 - \left(\sum_i x_i\right)^2} = \frac{\sum_i x_i y_i - \sum_j y_j\,\bar{x}}{\sum_i x_i^2 - \frac{1}{m}\left(\sum_i x_i\right)^2} = \boxed{\frac{\sum_i (x_i - \bar{x}) y_i}{\sum_i x_i^2 - \frac{1}{m}\left(\sum_i x_i\right)^2}}$$

$$b = \frac{\sum_i x_i^2 \sum_j y_j - \sum_i x_i y_i \sum_j x_j}{m\sum_i x_i^2 - \left(\sum_i x_i\right)^2} = \frac{\overline{x^2}\sum_j y_j - \sum_i x_i y_i \bar{x}}{\sum_i x_i^2 - \frac{1}{m}\left(\sum_i x_i\right)^2} = \boxed{\frac{\sum_j y_j\left(\overline{x^2} - x_i\bar{x}\right)}{\sum_i x_i^2 - \frac{1}{m}\left(\sum_i x_i\right)^2}}$$

对数线性回归

- 输出标记的对数为线性模型逼近的目标



$$y = \boldsymbol{w}^\top \boldsymbol{x} + b$$

$$\ln y = \boldsymbol{w}^\top \boldsymbol{x} + b$$

线性回归 - 广义线性模型

- 一般形式

$$y = g^{-1}(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b)$$

- $g(\cdot)$称为链接函数 (link function)
  - 单调可微

- 对数线性回归 $g(\cdot) = \ln(\cdot)$ 就是广义线性模型的特列

二分类任务

- 预测值与输出标记

$$z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \qquad y \in \{0,1\}$$

- 寻找函数将分类标记与线性回归模型输出联系起来

- 最理想的函数——单位阶跃函数

$$y = \begin{cases} 0, & z < 0 \\ 0.5, & z = 0 \\ 1, & z > 0 \end{cases}$$

  - 预测值大于0就判别为正例，小于0判别为负例，预测值为临界值0时可以任意判别

二分类任务

- 单位阶跃函数缺点
  - 不连续，无法用在广义线性模型中

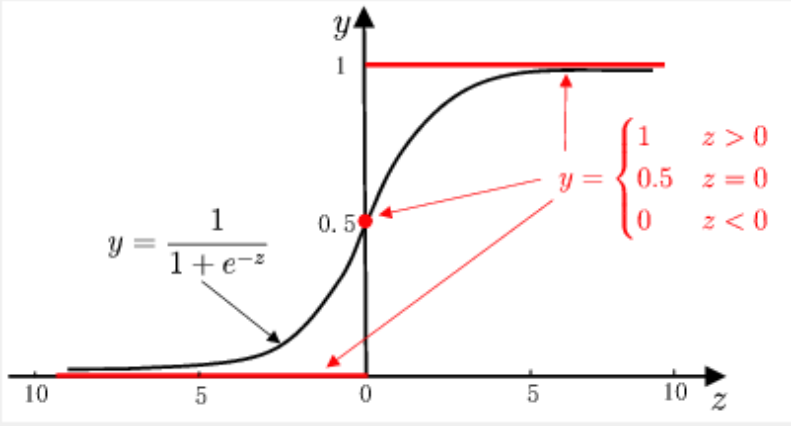
- 替代函数——对数几率函数（logistic function）

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

单调可微、任意阶可导

$$\sigma^{-1}(y) = \ln \frac{y}{1 - y}$$

单位阶跃函数与对数几率函数的比较



对数几率回归

- 运用对数几率函数

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad \Rightarrow \quad \hat{y} = \frac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}}$$

$\hat{y}$视为样本$\boldsymbol{x}$ 作为正例的可能性

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = \ln\frac{\hat{y}}{1-\hat{y}}$$

$\frac{\hat{y}}{1-\hat{y}}$称为几率，反映了$\boldsymbol{x}$作为正例的相对可能性

对数几率回归优点

- 无需事先假设数据分布
- 可得到“类别”的近似概率预测
- 可直接应用现有数值优化算法求取最优解

对数几率回归 - 极大似然法

如何优化参数$(\boldsymbol{w}, b)$?

$$\hat{y} = \frac{1}{1 + e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}} \triangleq P(y = 1|\boldsymbol{x}; \boldsymbol{w}, b)$$

$$P(y = 0|\boldsymbol{x}; \boldsymbol{w}, b) = \frac{e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}}{1 + e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}} = \frac{1}{1 + e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$$

给定数据集$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}, \quad y_i \in \{0,1\}$

- 极大似然法 最大化 对数似然

$$\ell(\boldsymbol{w}, b) = \sum_{i=1}^{m} y_i \log P(y = 1|\boldsymbol{x}_i; \boldsymbol{w}, b) + (1 - y_i) \log P(y = 0|\boldsymbol{x}_i; \boldsymbol{w}, b)$$

$$= \sum_{i=1}^{m} \log P(y = y_i|\boldsymbol{x}_i; \boldsymbol{w}, b)$$

对数几率回归 - 极大似然法

- 极大似然法 最小化 负对数似然

$$\ell(\hat{\boldsymbol{w}}) = -\sum_{i=1}^{m} \log P(y = y_i|\hat{\boldsymbol{x}}_i; \hat{\boldsymbol{w}})$$

$$P(y = 1|\boldsymbol{x}; \boldsymbol{w}, b) = \frac{e^{\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}}{1 + e^{\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}} = \frac{e^{1\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}}{1 + e^{\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}} = \frac{e^{y\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}}{1 + e^{\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}}$$

$$P(y = 0|\boldsymbol{x}; \boldsymbol{w}, b) = \frac{1}{1 + e^{\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}} = \frac{e^{0\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}}{1 + e^{\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}} = \frac{e^{y\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}}{1 + e^{\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}}}$$

$$\ell(\hat{\boldsymbol{w}}) = -\sum_{i=1}^{m} \log \frac{e^{y_i\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}_i}}{1 + e^{\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}_i}} = \sum_{i=1}^{m} -y_i\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}_i + \log\left(1 + e^{\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}_i}\right)$$

对数几率回归 - 极大似然法

- 考察函数$f(x) = -ax + \ln(1 + \exp(x))$

一阶导数 $$f'(x) = -a + \frac{1}{1 + e^{-x}}$$

二阶导数 $$f''(x) = \left(\frac{1}{1 + e^{-x}}\right)' = \frac{e^{-x}}{(1 + e^{-x})^2} = \sigma(x)\big(1 - \sigma(x)\big)$$

因为$\sigma(x) \in (0,1)$，所以$f''(x) > 0$，因此$f(x)$是凸函数。

因为$\ell(\hat{\boldsymbol{w}})$是$f(x)$ 和$g(\boldsymbol{w}) = \hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}_i$的复合函数，即$\ell(\hat{\boldsymbol{w}}) = f\big(g(\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\boldsymbol{x}}_i)\big)$，
所以$\ell(\hat{\boldsymbol{w}})$是关于$\hat{\boldsymbol{w}}$的凸函数

对数几率回归 - 极大似然法

- 负对数似然 $\ell(\hat{\boldsymbol{w}}) = \sum_{i=1}^{m} \left(-y_i \hat{\boldsymbol{w}}^{\mathsf{T}} \hat{\boldsymbol{x}}_i + \log\left(1 + e^{\hat{\boldsymbol{w}}^{\mathsf{T}} \hat{\boldsymbol{x}}_i}\right)\right)$

一阶导数

$$\nabla \ell(\hat{\boldsymbol{w}}) = \sum_i f'(z_i) \frac{\partial z_i}{\partial \hat{\boldsymbol{w}}} = \sum_i \left(-y_i + \frac{1}{1 + e^{-z_i}} \right) \boldsymbol{x}_i$$

$$= -\sum_i \left(y_i - P(y = 1 | \hat{\boldsymbol{x}}_i; \hat{\boldsymbol{w}}) \right) \hat{\boldsymbol{x}}_i$$

p_1

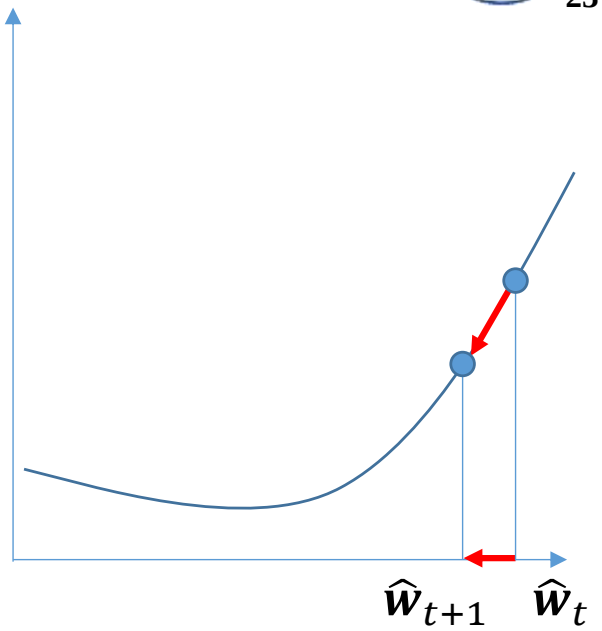
二阶导数

$$\nabla^2 \ell(\hat{\boldsymbol{w}}) = \frac{\partial \nabla \ell(\hat{\boldsymbol{w}})}{\partial \hat{\boldsymbol{w}}^{\mathsf{T}}} = \sum_i p_1 (1 - p_1) \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}$$

对数几率回归 - 极大似然法

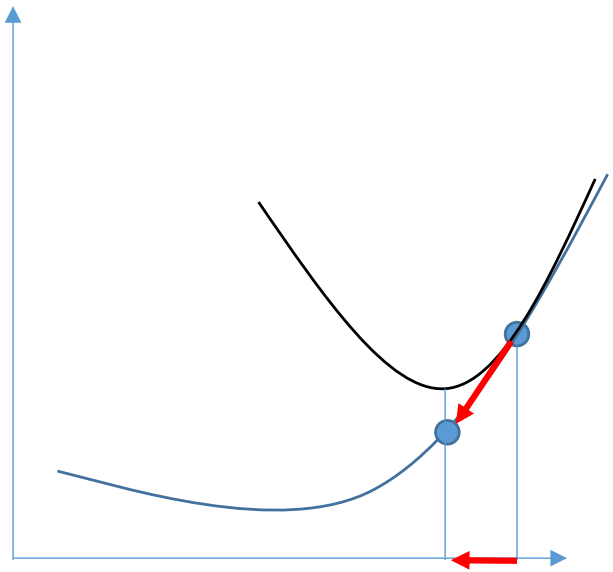
- 梯度下降法

while $\|\nabla \ell(\hat{\boldsymbol{w}})\| > \delta$ do
$\hat{\boldsymbol{w}}_{t+1} \leftarrow \hat{\boldsymbol{w}}_t - \alpha \nabla \ell(\hat{\boldsymbol{w}})$
end while



- 牛顿法

while $\|\nabla \ell(\hat{\boldsymbol{w}})\| > \delta$ do
$\hat{\boldsymbol{w}}_{t+1} \leftarrow \hat{\boldsymbol{w}}_t - \left(\nabla^2 \ell(\hat{\boldsymbol{w}})\right)^{-1} \nabla \ell(\hat{\boldsymbol{w}})$
end while



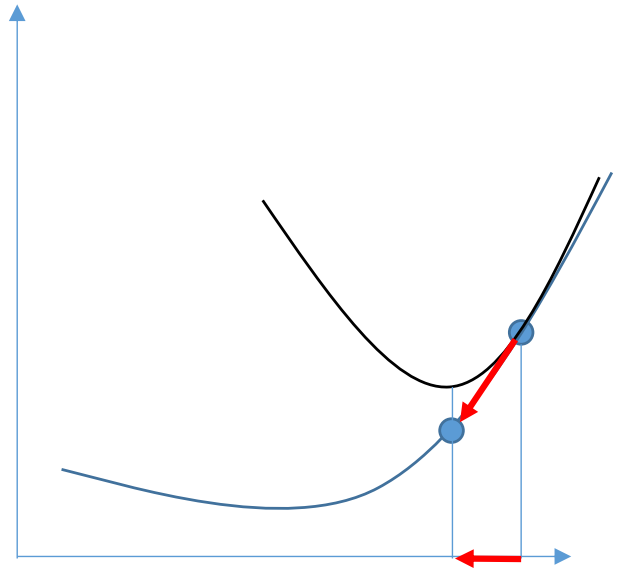
对数几率回归 - 极大似然法

牛顿法

while $\|\nabla \ell(\hat{\boldsymbol{w}})\| > \delta$ do

$\quad \hat{\boldsymbol{w}}_{t+1} \leftarrow \hat{\boldsymbol{w}}_t - \left(\nabla^2 \ell(\hat{\boldsymbol{w}})\right)^{-1} \nabla \ell(\hat{\boldsymbol{w}})$

end while



- 考虑$\ell(\hat{\boldsymbol{w}})$在$\hat{\boldsymbol{w}}_t$处的二阶泰勒展开

$$\ell(\hat{\boldsymbol{w}}) \approx \ell(\hat{\boldsymbol{w}}_t) + \nabla \ell(\hat{\boldsymbol{w}}_t)^{\mathsf{T}}(\hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}_t) + \frac{1}{2}(\hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}_t)^{\mathsf{T}} \nabla^2 \ell(\hat{\boldsymbol{w}})(\hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}_t)$$

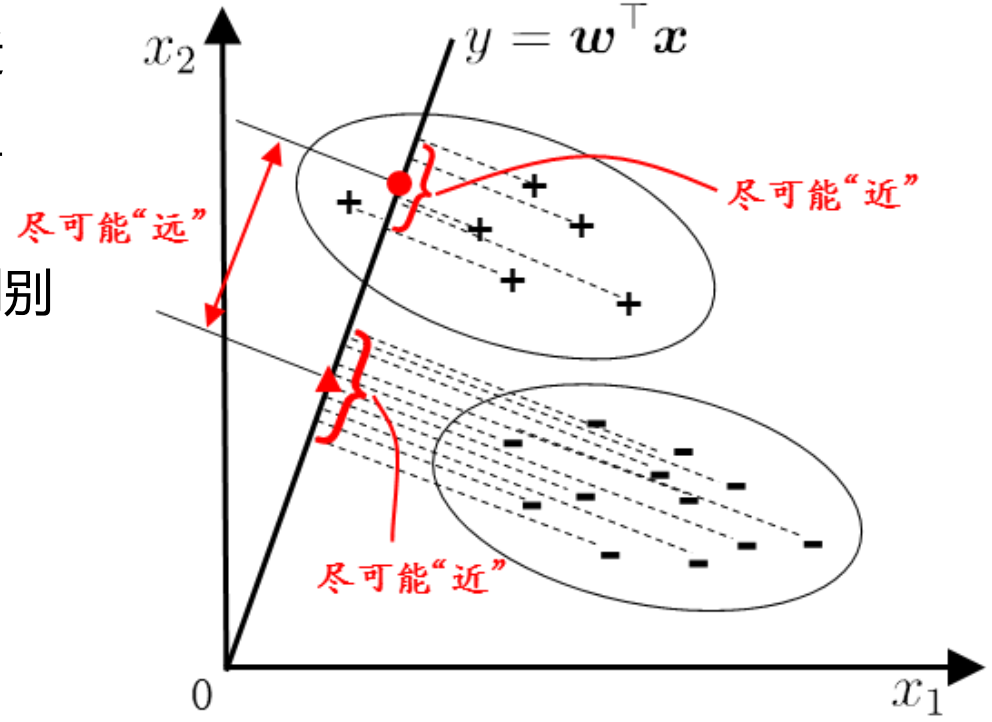
- 对$\hat{\boldsymbol{w}}$求导数并令其等于0，得到$\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}}_t - \left(\nabla^2 \ell(\hat{\boldsymbol{w}})\right)^{-1} \nabla \ell(\hat{\boldsymbol{w}})$

二分类任务– 线性判别分析

- 线性判别分析（Linear Discriminant Analysis）[Fisher, 1936]

投影到低纬空间，使得

- 欲使同类样例的投影点尽可能接近
- 欲使异类样例的投影点尽可能远离
- 新样本投影后根据投影位置进行判别



LDA也可被视为一种
监督降维技术

二分类任务– 线性判别分析

- 给定数据集$D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}, y_i \in \{0,1\}$

令

- 第c类示例的集合 X_c
- 第c类示例的均值向量 $\boldsymbol{\mu}_c = \frac{1}{|X_c|}\sum_{\boldsymbol{x}\in X_c}\boldsymbol{x}$
- 第c类示例的协方差矩阵 $\boldsymbol{\Sigma}_c = \sum_{x\in X_c}(\boldsymbol{x}-\boldsymbol{\mu}_c)(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\mathrm{T}}$

若将数据投影到方向$\boldsymbol{w}$确定的直线上

- 两类样本的中心在直线上的投影 $\frac{1}{|X_c|}\sum_{\boldsymbol{x}\in X_c}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_c$
- 两类样本的协方差 $\sum_{\boldsymbol{x}\in X_c}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_c)^2 = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_c\boldsymbol{w}$

二分类任务– 线性判别分析

- 欲使同类样例的投影点尽可能接近，

  可以让同类样例投影点的协方差尽可能小

- 欲使异类样例的投影点尽可能远离，

  可以让类中心之间的距离尽可能大

- 同时考虑两者，则可得到最大化目标

定义类内散度

$$S_b = (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^\top$$

$$J = \frac{\left\|\boldsymbol{w}^\top \boldsymbol{\mu}_0 - \boldsymbol{w}^\top \boldsymbol{\mu}_1\right\|_2^2}{\boldsymbol{w}^\top \boldsymbol{\Sigma}_0 \boldsymbol{w} + \boldsymbol{w}^\top \boldsymbol{\Sigma}_1 \boldsymbol{w}} = \frac{\boldsymbol{w}^\top (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^\top \boldsymbol{w}}{\boldsymbol{w}^\top (\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1) \boldsymbol{w}} = \frac{\boldsymbol{w}^\top \boldsymbol{S}_b \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_w \boldsymbol{w}}$$

$$S_w = \boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1$$

定义类间散度

二分类任务– 线性判别分析

$$\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^{\top}\boldsymbol{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\top}\boldsymbol{S}_w\boldsymbol{w}}$$

广义瑞利商
(generalized Rayleigh quotient)

- 若$\boldsymbol{w}$是一个解，则对于任意常数α, $\alpha\boldsymbol{w}$也是解

- 不失一般性，令$\boldsymbol{w}^{\top}\boldsymbol{S}_w\boldsymbol{w} = 1$，则等价于

$$\max_{\boldsymbol{w}} \boldsymbol{w}^{\top}\boldsymbol{S}_b\boldsymbol{w} \qquad \text{s.t } \boldsymbol{w}^{\top}\boldsymbol{S}_w\boldsymbol{w} = 1$$

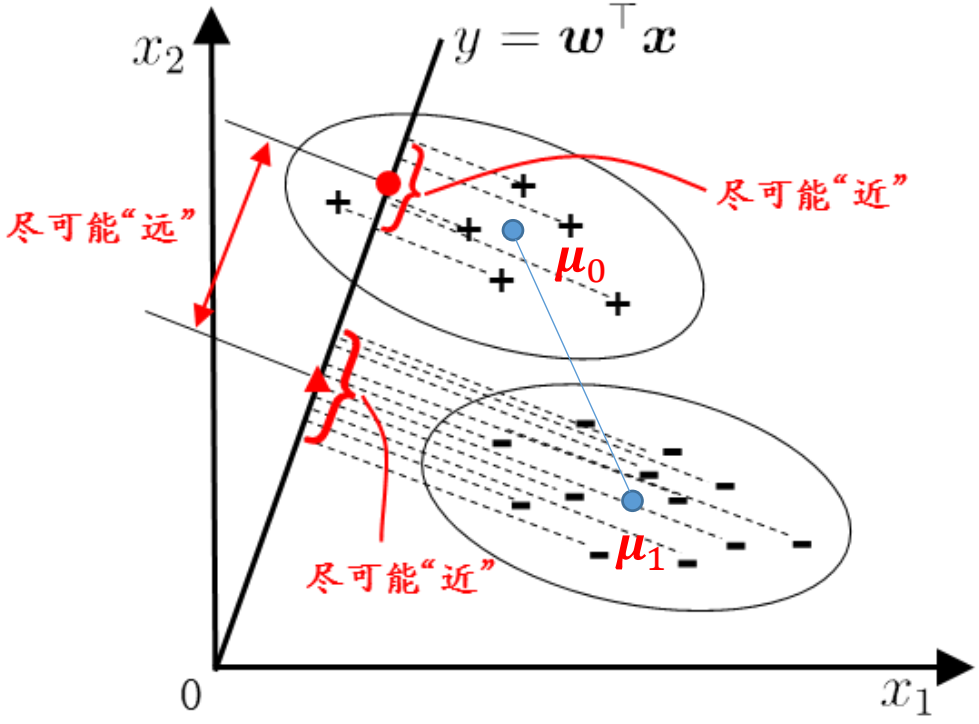
- 引入拉格朗日乘子λ，并令朗格拉日函数梯度等于0，可以得到

$$\boldsymbol{S}_b\boldsymbol{w} = \lambda\boldsymbol{S}_w\boldsymbol{w}$$

二分类任务– 线性判别分析

最优解 $\boldsymbol{S}_b\boldsymbol{w} = \lambda \boldsymbol{S}_w\boldsymbol{w}$

- $\boldsymbol{S}_b\boldsymbol{w} = (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^\top \boldsymbol{w} \propto (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)$
- 由此可得 $\boldsymbol{w} \propto \boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)$



LDA推广– 多分类任务

- 全局散度矩阵 $\boldsymbol{S}_t = \boldsymbol{S}_b + \boldsymbol{S}_w = \sum_i (\boldsymbol{x_i} - \boldsymbol{\mu})(\boldsymbol{x_i} - \boldsymbol{\mu})^\top$

- 类内散度矩阵 $\boldsymbol{S}_w = \sum_c \boldsymbol{S}_{w_c}$ $\boldsymbol{S}_{w_c} = \sum_{\boldsymbol{x} \in X_c} (\boldsymbol{x} - \boldsymbol{\mu}_c)(\boldsymbol{x} - \boldsymbol{\mu}_c)^\top$

- 类间散度矩阵 $\boldsymbol{S}_{w_c} = \sum_{\boldsymbol{x} \in X_c} \boldsymbol{x}\,\boldsymbol{x}^\top - m_c \boldsymbol{\mu}_c \boldsymbol{\mu}_c^\top$ $\boldsymbol{S}_t = \sum_i \boldsymbol{x}_i\,\boldsymbol{x}_i^\top - m\boldsymbol{\mu}\,\boldsymbol{\mu}^\top$

$$\boldsymbol{S}_b = \boldsymbol{S}_t - \boldsymbol{S}_{\mathrm{w}} = \sum_c m_c \boldsymbol{\mu}_c \boldsymbol{\mu}_c^\top - m\boldsymbol{\mu}\,\boldsymbol{\mu}^\top$$

$$= \sum_c m_c (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^\top$$

$$m\boldsymbol{\mu} = \sum_c m_c \boldsymbol{\mu}_c \qquad m = \sum_c m_c$$

LDA推广– 多分类任务

- 优化目标

$$\max_{\boldsymbol{W}} \frac{\mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{S}_b \boldsymbol{W})}{\mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{S}_w \boldsymbol{W})} = \frac{\sum_i \boldsymbol{w}_i^\top \boldsymbol{S}_b \boldsymbol{w}_i}{\sum_i \boldsymbol{w}_i^\top \boldsymbol{S}_w \boldsymbol{w}_i}$$

- 若$\boldsymbol{W}$是一个解，则对于任意常数α, $\alpha\boldsymbol{W}$也是解

- 等价于 $\max_{\boldsymbol{W}} tr(\boldsymbol{W}^\top \boldsymbol{S}_b \boldsymbol{W})$ s.t. $tr(\boldsymbol{W}^\top \boldsymbol{S}_w \boldsymbol{W}) = 1$

- 引入拉格朗日乘子λ，并令朗格拉日函数梯度等于0，可以得到广义特征值问题

$$\boldsymbol{S}_b \boldsymbol{W} = \lambda \boldsymbol{S}_w \boldsymbol{W}$$

$\boldsymbol{W}$的闭式解则是 $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$的N-1个最大广义特征值所对应的特征向量组成的矩阵

多分类学习

- 多分类学习方法
  - 二分类学习方法推广到多类，利用二分类学习器解决多分类问题

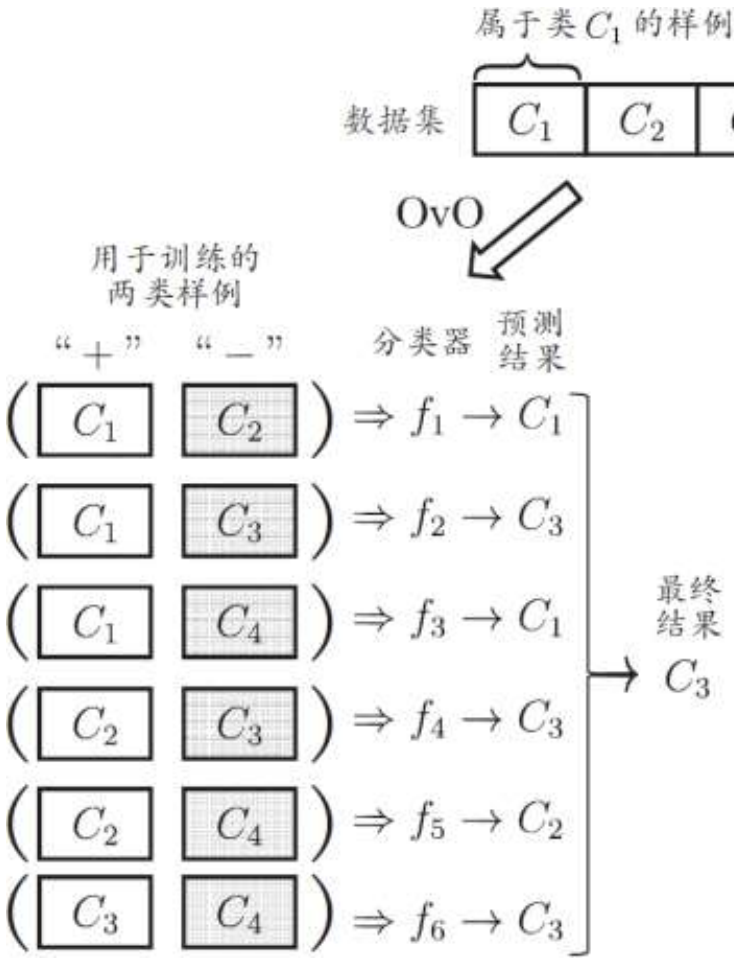
✓对问题进行拆分，为拆出的每个二分类任务训练一个分类器

✓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）

多分类学习– 一对一



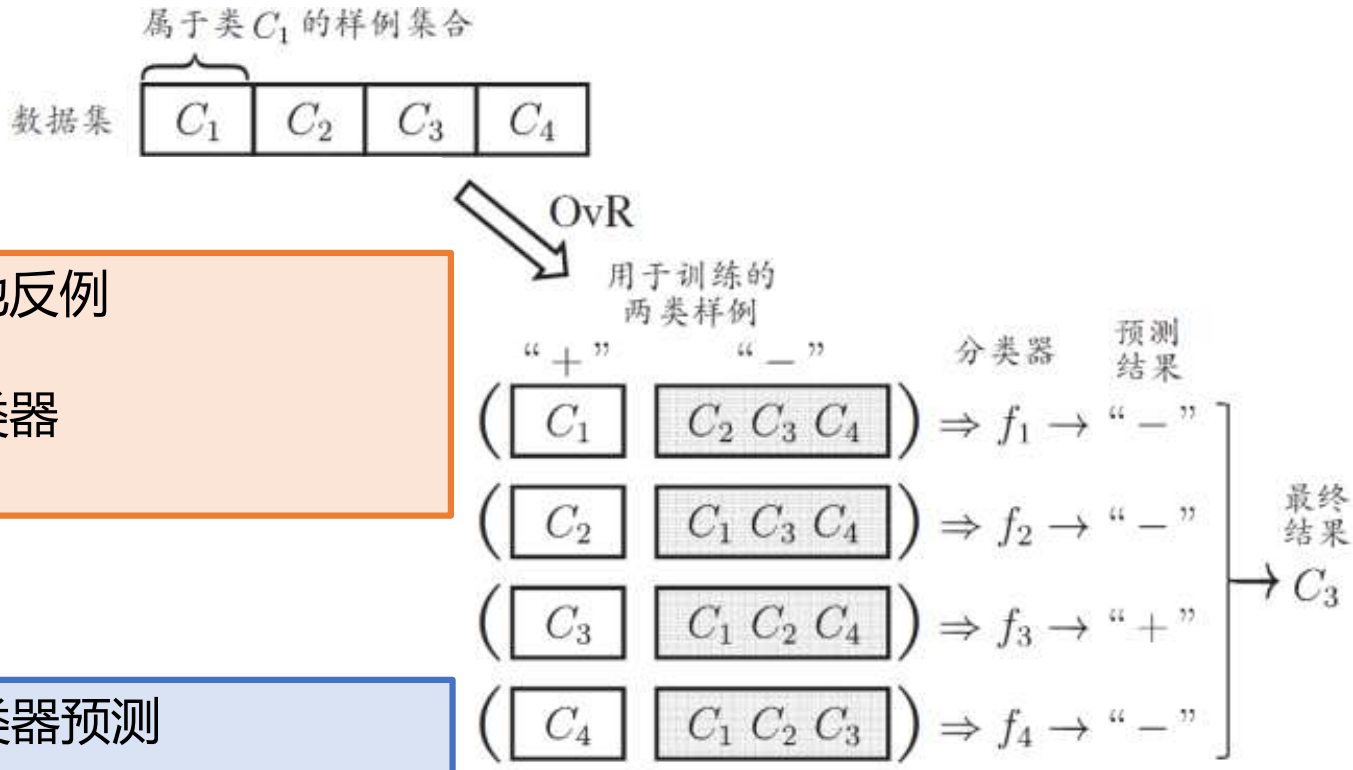
拆分阶段

- N个类别两两配对
  - 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

多分类学习– 一对其余



拆分阶段

- 某一类作为正例，其他反例
  - 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
  - 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，存储开销和测试时间大
- 训练只用两个类的样例，训练时间短

一对其余

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

预测性能取决于具体数据分布，多数情况下两者差不多

多分类学习– 多对多

- 多对多（Many vs Many, MvM）

  若干类作为正类，若干类作为反类

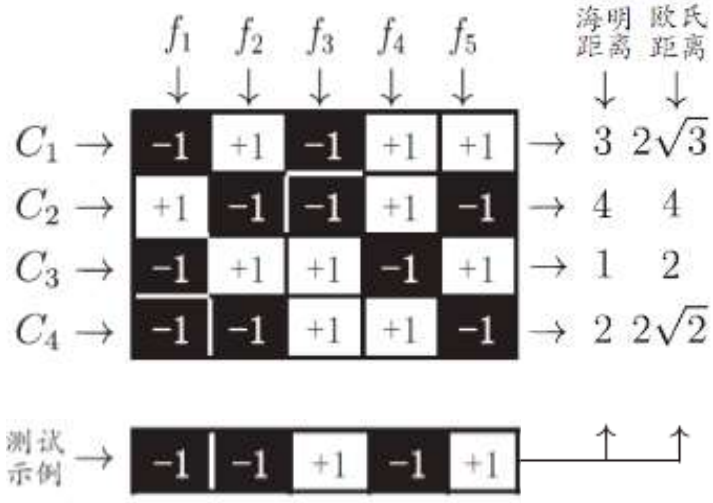
- 纠错输出码（Error Correcting Output Code, ECOC）

编码

对N个类别做M次划分，每次划分将一部分类别划为正类，一部分划为反类，形成二分类训练集

解码

M个分类器分别对测试样本进行预测，预测标记组成一个编码。将距离最小的类别为最终类别



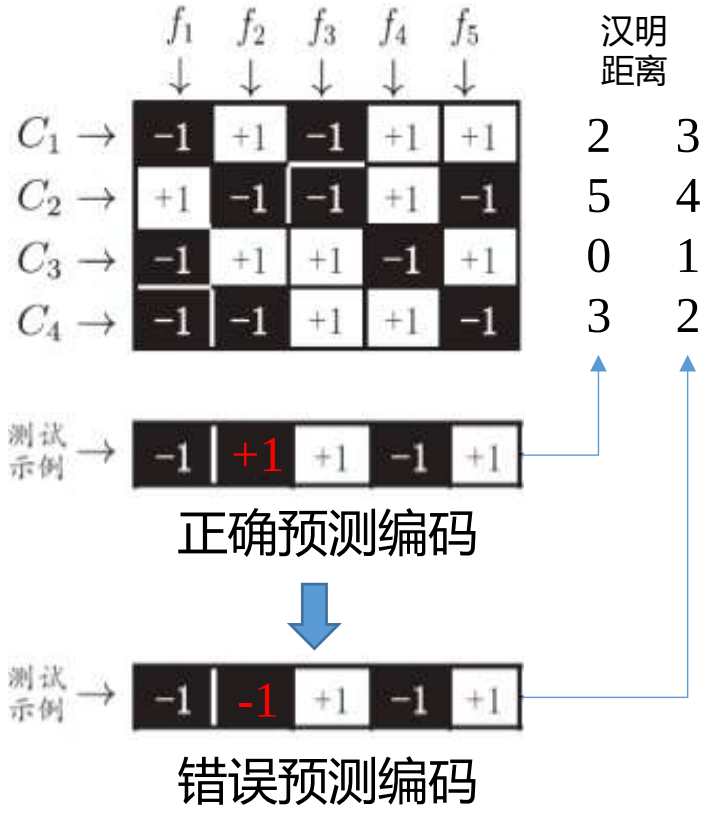
多分类学习– 多对多

编码

对N个类别做M次划分，每次划分将一部分类别划为正类，一部分划为反类，形成二分类训练集

解码

M个分类器分别对测试样本进行预测，预测标记组成一个编码。将距离最小的类别为最终类别



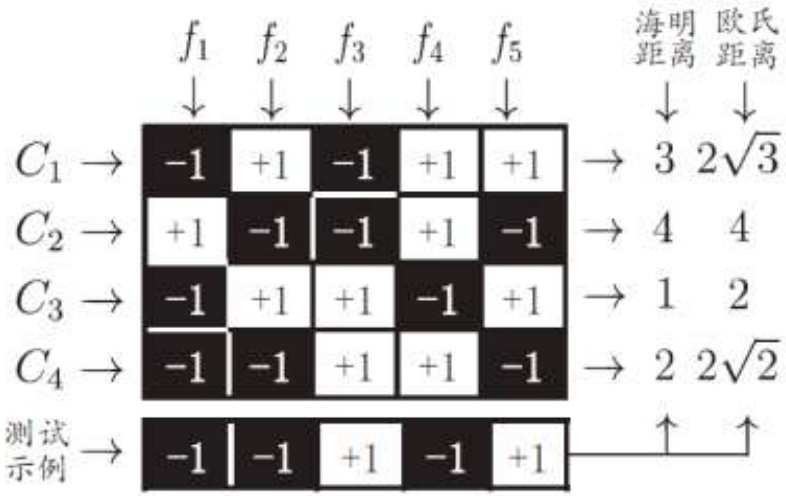
- 纠错能力
  - 若f_2预测错误
  - 仍然能产生正确的最终分类

✓ECOC编码越长、对分类器错误纠错能力越强

✓对同等长度编码，理论上任意两个类别之间的编码距离越远，则纠错能力越强

多分类学习– 多对多

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



(a) 二元 ECOC 码

[Dietterich and Bakiri,1995]

类别不平衡问题

- 类别不平衡（class imbalance）
  - 不同类别训练样例数相差很大情况（正类为小类）

类别平衡正例预测

分类决策规则：$\frac{y}{1-y} > 1$，则为正例

类别不平衡正例预测

分类决策规则：$\frac{y}{1-y} > \frac{m^+}{m^-}$，则正例

y为分类器预测值，表达了正例可能性，几率$\frac{y}{1-y}$反映相对可能性

- 分类器都基于类别平衡分类决策规则决策的，只能对预测值进行缩放

$$\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+} \quad \Rightarrow \quad y' = \frac{m^- y}{m^+(1-y) + m^- y}$$

类别不平衡问题

“训练集是真实样本总体的无偏采样” 假设往往不成立，未必能基于训练集观测几率来推断真实几率

解决办法

- 欠采样（undersampling）
  - 去除一些反例使正反例数目接近（EasyEnsemble [Liu et al.,2009]）
- 过采样（oversampling）
  - 增加一些正例使正反例数目接近（SMOTE [Chawla et al.2002]）
- 阈值移动（threshold-moving）

作业

- 3.2
- 3.7
- 在LDA多分类情形下，试计算类间散度矩阵S_b的秩并证明
- 给出公式3.45的推导过程
- 证明$\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}$是投影矩阵，并对线性回归模型从投影角度解释

第四章：决策树

主讲：连德富 特任教授 | 博士生导师

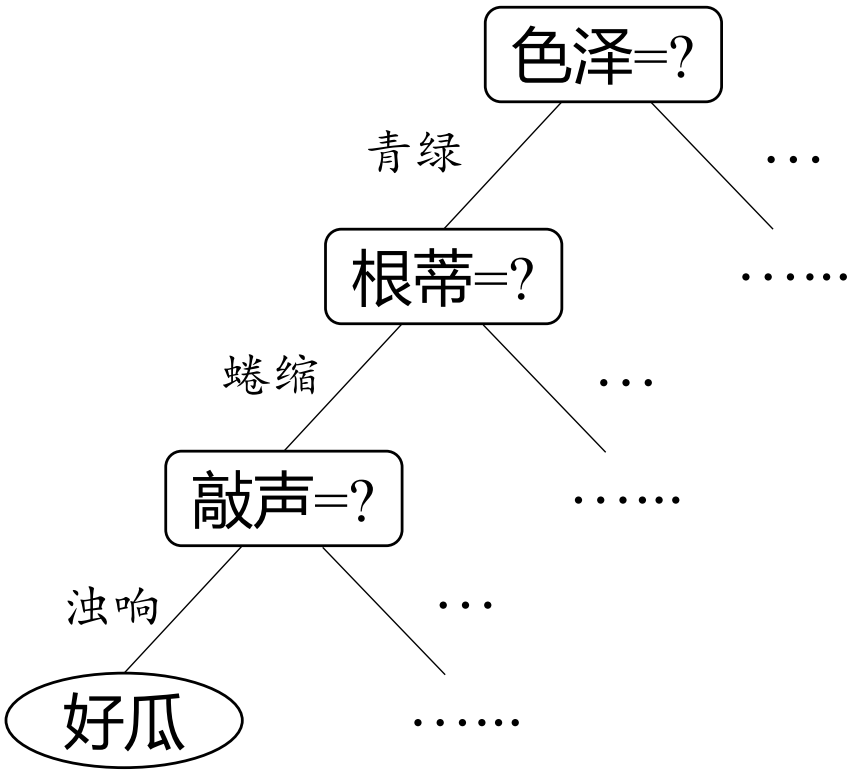
邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

基本流程

- 决策树基于树结构来进行预测



- 决策过程的最终结论对应了我们所希望的判定结果（叶子节点）
- 决策过程中提出的每个判定问题都是对某个属性的“测试”（内部节点）
- 每个测试的结果或是导出最终结论，或者导出进一步的判定问题，其考虑范围是在上次决策结果的限定范围之内
- 每个节点包含的样本集合根据属性测试划分到子节点中（根节点包括样本全集）
- 从根结点到每个叶结点的路径对应了一个判定测试序列

决策树学习的目的是为了产生一棵泛化能力强，即处理未见示例能力强的决策树

基本流程

决策树的生成是一个递归过程

Algorithm 1 决策树学习基本算法

输入:

- 训练集 $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$;
- 属性集 $A = \{a_1, \ldots, a_d\}$.

过程: 函数 TreeGenerate(D, A)

```
1: 生成结点 node;
2: if D 中样本全属于同一类别 C then
3:     将 node 标记为 C 类叶结点; return
4: end if
5: if A = ∅ OR D 中样本在 A 上取值相同 then
6:     将 node 标记叶结点，其类别标记为 D 中样本数最多的类; return
7: end if
8: 从 A 中选择最优划分属性 a_*;
9: for a_* 的每一个值 a_*^v do
10:    为 node 生成每一个分枝; 令 D_v 表示 D 中在 a_* 上取值为 a_*^v 的样本子集;
11:    if D_v 为空 then
12:        将分枝结点标记为叶结点, 其类别标记为 D 中样本最多的类; return
13:    else
14:        以 TreeGenerate(D_v, A − {a_*}) 为分枝结点
15:    end if
16: end for
```

输出: 以 node 为根结点的一棵决策树

下述三种情况递归返回

（1）当前结点包含的样本全部属于同一类别

（2）当前属性集为空，或所有样本在所有属性上取值相同

（3）当前结点包含的样本集合为空

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

经典的属性划分方法：

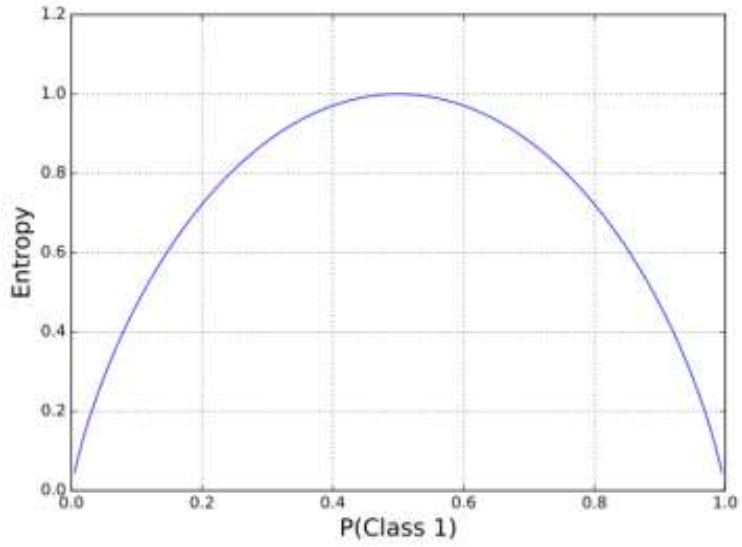
信息增益　增益率　基尼指数

划分选择—信息增益

- “信息熵”是度量样本集合纯度最常用的一种指标

假定当前样本集合D中第k类样本所占的比例为$p_k(\mathrm{k}=1,2,\cdots,|\mathcal{Y}|)$，则$D$的信息熵定义为

$$Ent(D)=-\sum_k p_k \log_2 p_k$$



二分类：

$$Ent(D)=-p_1\log p_1-(1-p_1)\log(1-p_1)$$

划分选择—信息增益

- 对于任意一个离散随机变量Y，信息熵 $H(Y)=-\sum_i P(Y=i)\log P(Y=i)$
- 信息熵是度量样本集合纯度最常用的一种指标

假定当前样本集合D中第k类样本所占的比例为$p_k(\mathrm{k}=1,2,\cdots,|\mathcal{Y}|)$，则$D$的信息熵定义为

$$Ent(D)=-\sum_k p_k \log_2 p_k$$

$Ent(D)$的值越小，则 D的纯度越高

若f为凹函数，则 $\sum_i \alpha_i f(x_i)\le f(\sum_i \alpha_i x_i)$

易证： $Ent(D)=\sum_k p_k \log_2 \frac{1}{p_k} \le \log_2 \sum_k p_k \frac{1}{p_k} = \log_2|\mathcal{Y}|$

均匀分布的熵最大

计算信息熵时约定：若 $p=0$，则 $p\log p=0$

划分选择—信息增益

- 信息增益是是变量间相互依赖性的度量，是联合分布和边缘分布乘积的相似程度度量

$$I(X,Y) = \sum_{y}\sum_{x} P(X=x, Y=y) \log \frac{P(X=x, Y=y)}{P(X=x)P(Y=y)}$$

$$= H(Y) - \sum_{x} P(X=x) H(Y|X=x)$$

$$= H(Y) - H(Y|X)$$

$$= H(X) - H(X|Y)$$

划分选择—信息增益

- 假设离散属性a有V个可能的取值$\{a^1, a^2, \cdots, a^V\}$
- 若用a来进行划分，则会产生V个分支结点；第v个分支结点包含D中所有在属性a上取值为a^v的样本，记为D^v

用属性a对样本集D进行划分所获得的“信息增益”：

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

$I(a,Y)$　$H(Y)$　$P(a^v)$　$H(Y|a^v)$

为分支结点权重，样本数越多的分支结点的影响越大

信息增益越大，则意味着使用属性a来进行划分所获得的“纯度提升”越大

- ID3决策树算法[Quinlan, 1986]以信息增益为准则来选择划分属性

划分选择—信息增益

信息增益实例

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

数据集包含17个训练样本，$|\mathcal{Y}| = 2$

正例占$p_1 = \frac{8}{17}$，反例占$p_2 = \frac{9}{17}$

计算得到根结点的信息熵为

$$Ent(D) = -\sum_{k} p_k \log_2 p_k = -\left(\frac{8}{17}\log_2\frac{8}{17} + \frac{9}{17}\log_2\frac{9}{17}\right) = 0.998$$

划分选择—信息增益

信息增益实例

以属性“色泽”为例，其对应的3个数据子集分别为 D^1(色泽=青绿)， D^2(色泽=乌黑)， D^3 (色泽=浅白)

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

子集D^1包含编号为{1,4,6,10,13,17} 6个样例，正例$p_1 = \frac{3}{6}$，反例$p_2 = \frac{3}{6}$

$$Ent(D^1) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1.000$$

子集D^2包含编号为{2,3,7,8,9,15 } 6个样例，正例$p_1 = \frac{4}{6}$，反例$p_2 = \frac{2}{6}$

$$Ent(D^2) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} = 0.918$$

子集D^3包含编号为{5,11,12,14,16} 5个样例，正例$p_1 = \frac{1}{5}$，反例$p_2 = \frac{4}{5}$

$$Ent(D^1) = -\frac{1}{5}\log_2\frac{1}{5} - \frac{4}{5}\log_2\frac{4}{5} = 0.722$$

属性“色泽”的信息增益为

$$\text{Gain}(D, \text{色泽}) = Ent(D) - \sum_v \frac{|D^v|}{|D|} Ent(D^v)$$

$$= 0.998 - \left(\frac{6}{17} \times 1.000 + \frac{6}{17} \times 0.918 + \frac{5}{17} \times 0.772\right) = 0.109$$

划分选择—信息增益

信息增益实例

			敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

$$\text{Gain}(D, \text{色泽}) = 0.109$$

$$\text{Gain}(D, \text{根蒂}) = 0.143$$

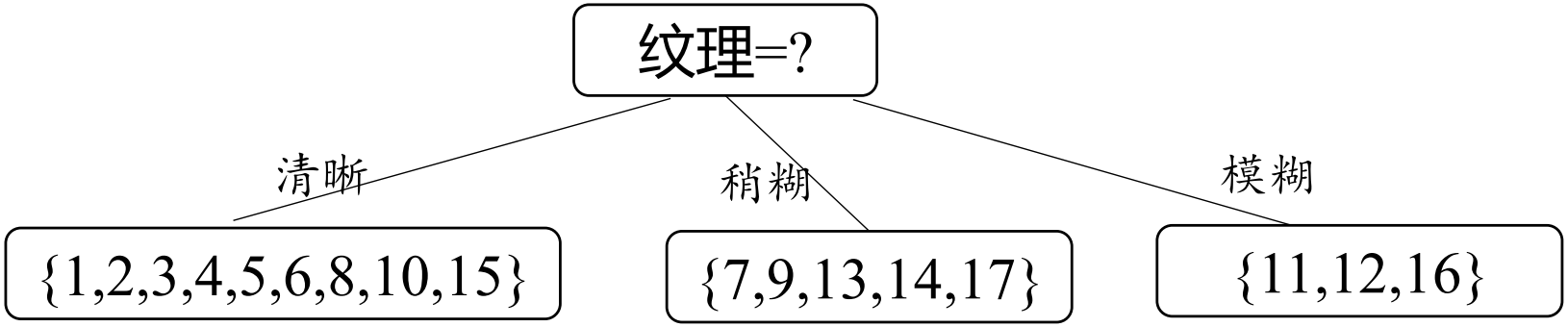
$$\text{Gain}(D, \text{敲声}) = 0.141$$

$$\text{Gain}(D, \text{纹理}) = 0.381$$

$$\text{Gain}(D, \text{脐部}) = 0.289$$

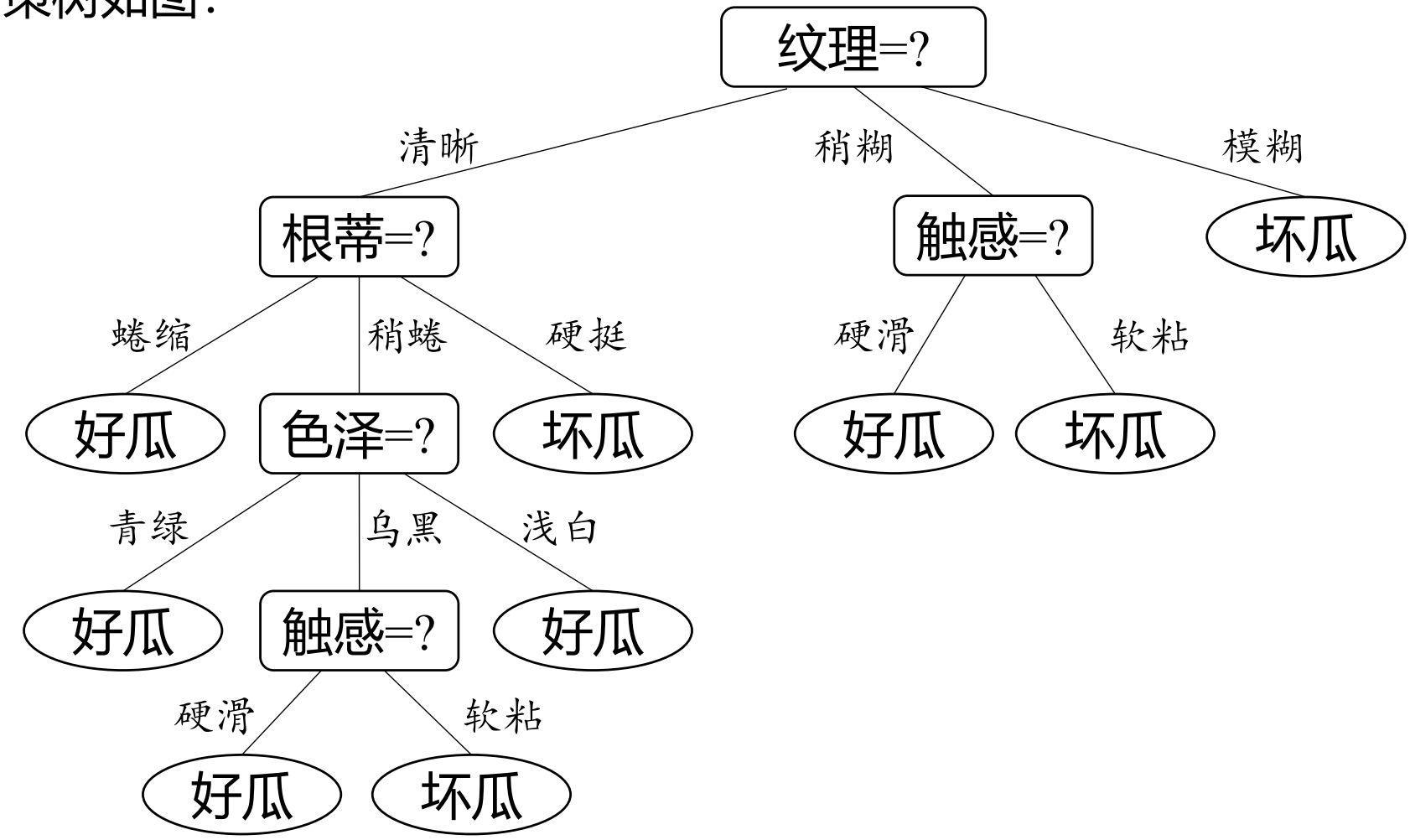
$$\text{Gain}(D, \text{触感}) = 0.006$$

属性“纹理”的信息增益最大，其被选为划分属性



划分选择—信息增益

决策树学习算法将对每个分支结点做进一步划分，最终得到的决策树如图：



划分选择—信息增益

存在的问题

若把“编号”也作为一个候选划分属性，则其信息增益一般远大于其他属性。显然，这样的决策树不具有泛化能力，无法对新样本进行有效预测

信息增益对可取值数目较多的属性有所偏好

划分选择—增益率

- 增益率定义：

$$\text{Gain}-\text{ratio}(D,a)=\frac{\text{Gain}(D,a)}{\text{IV}(a)} \qquad \text{其中 } \text{IV}(a)=-\sum_v \frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

称为属性a的“固有值”[Quinlan, 1993]，属性a的可能取值数目越多（即V越大），则$IV(a)$的值通常就越大

- 存在的问题

增益率准则对可取值数目较少的属性有所偏好

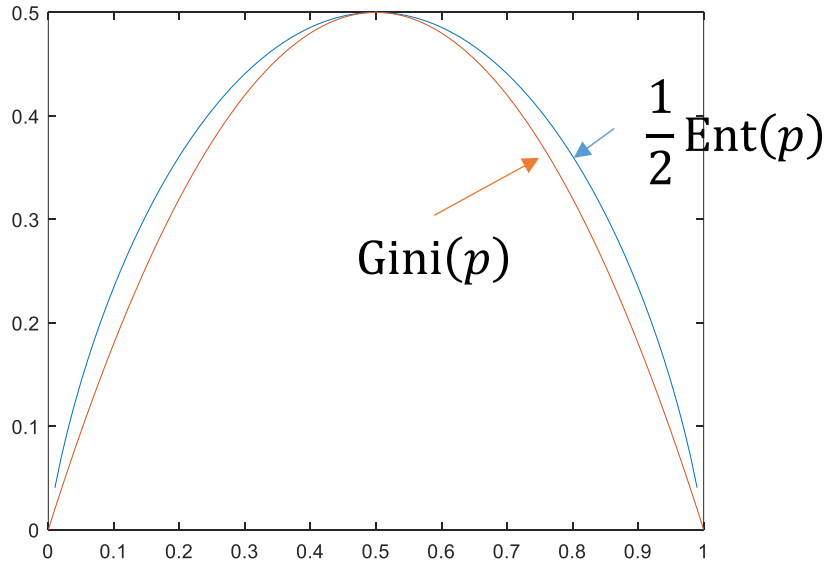
- C4.5 [Quinlan, 1993]使用了一个启发式：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选取增益率最高的

划分选择-基尼指数

- 数据集D的纯度可用“基尼值”来度量　　$\log x \approx x - 1$

$$\mathrm{Gini}(D) = \sum_{k=1}^{|y|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|y|} p_k^2 = \sum_{k=1}^{|y|} p_k(1 - p_k) \approx -\sum_{k=1}^{|y|} p_k \log p_k$$

- 反映了从D中随机抽取两个样本，其类别标记不一致的概率
- $\mathrm{Gini}(D)$越小，数据集D的纯度越高



划分选择-基尼指数

- 数据集D的纯度可用“基尼值”来度量

$$\mathrm{Gini}(D) = \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2$$

- 属性a的基尼指数定义为：

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

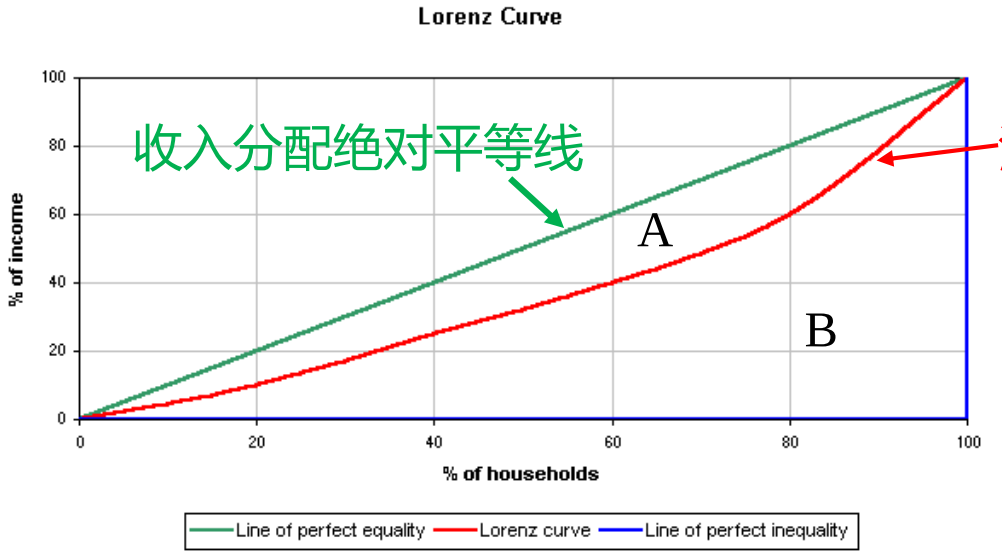
- 选择那个使划分后基尼指数最小的属性作为最优划分属性，即

$$a^\star = \arg\min_{a \in A} \mathrm{Gini-index}(D, a)$$

- CART [Breiman et al., 1984]采用“基尼指数”来选择划分属性

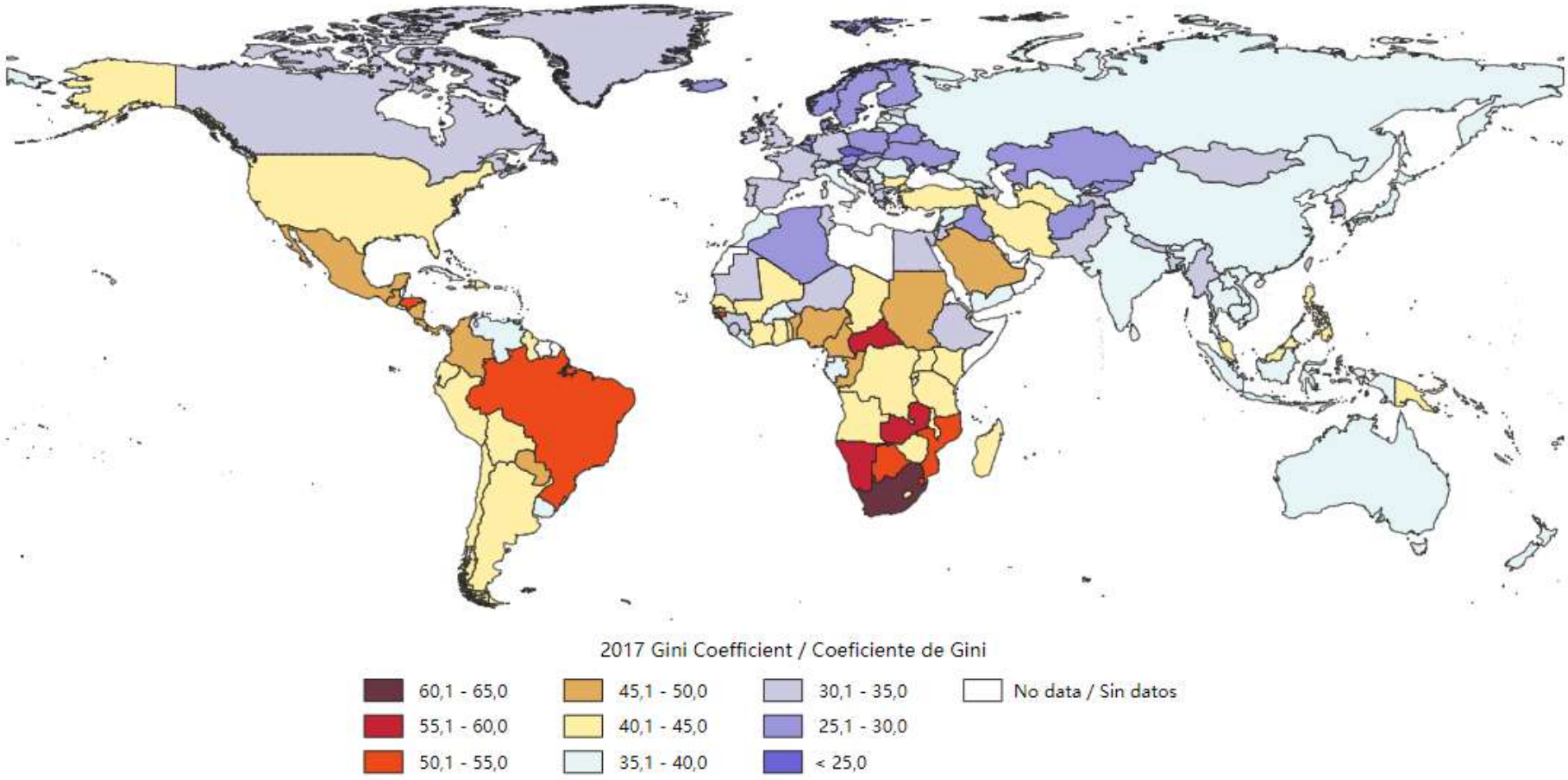
基尼系数

- 洛伦兹曲线是在过往财富分配数据上建立的累积分布函数所对应的曲线
- x%代表一部分（收入相似）家庭占整个社会家庭的比例，以y%代表该部分家庭的收入占整个社会收入的比例
- 基尼系数是根据洛伦兹曲线所定义的判断年收入分配公平程度的指标



基尼系数 $\frac{A}{A+B}$

基尼系数



2017年世界银行基尼系数世界地图。
基尼系数越小收入分配越平均，基尼系数越大收入分配越不平均。

剪枝处理

- 为什么剪枝
  - “剪枝”是决策树学习算法对付“过拟合”的主要手段
  - 可通过“剪枝”来一定程度避免因决策分支过多，以致于把训练集自身的一些特点当做所有数据都具有的一般性质而导致的过拟合

- 剪枝的基本策略
  - 预剪枝
  - 后剪枝

- 判断决策树泛化性能是否提升的方法
  - 留出法：预留一部分数据用作“验证集”以进行性能评估

剪枝处理

- 数据集

训练集

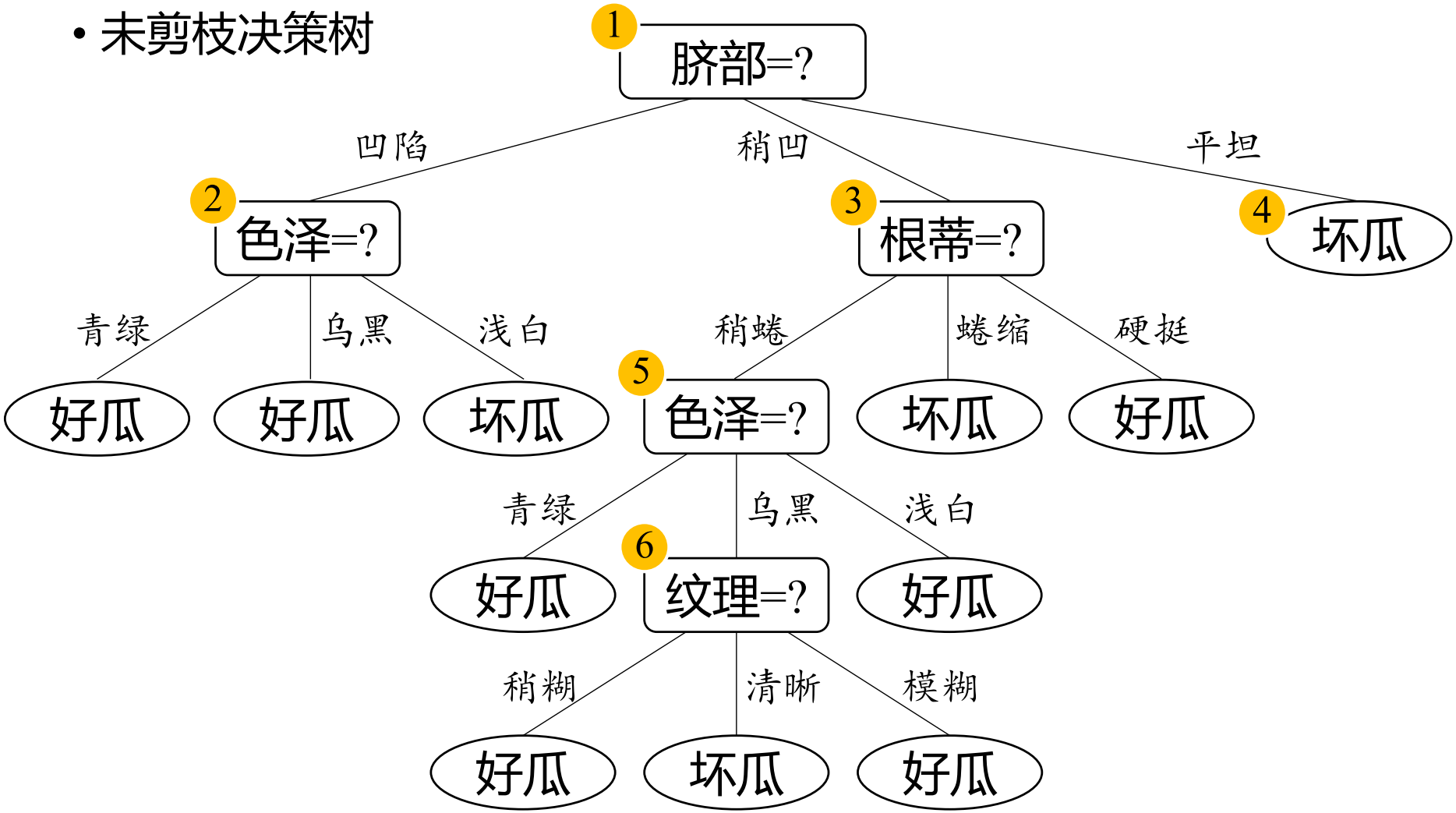
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

验证集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

剪枝处理

- 未剪枝决策树


1
脐部=?
凹陷
稍凹
平坦
2
色泽=?
3
根蒂=?
4
坏瓜
青绿
乌黑
浅白
稍蜷
蜷缩
硬挺
好瓜
好瓜
坏瓜
5
色泽=?
坏瓜
好瓜
青绿
乌黑
浅白
好瓜
6
纹理=?
好瓜
稍糊
清晰
模糊
好瓜
坏瓜
好瓜

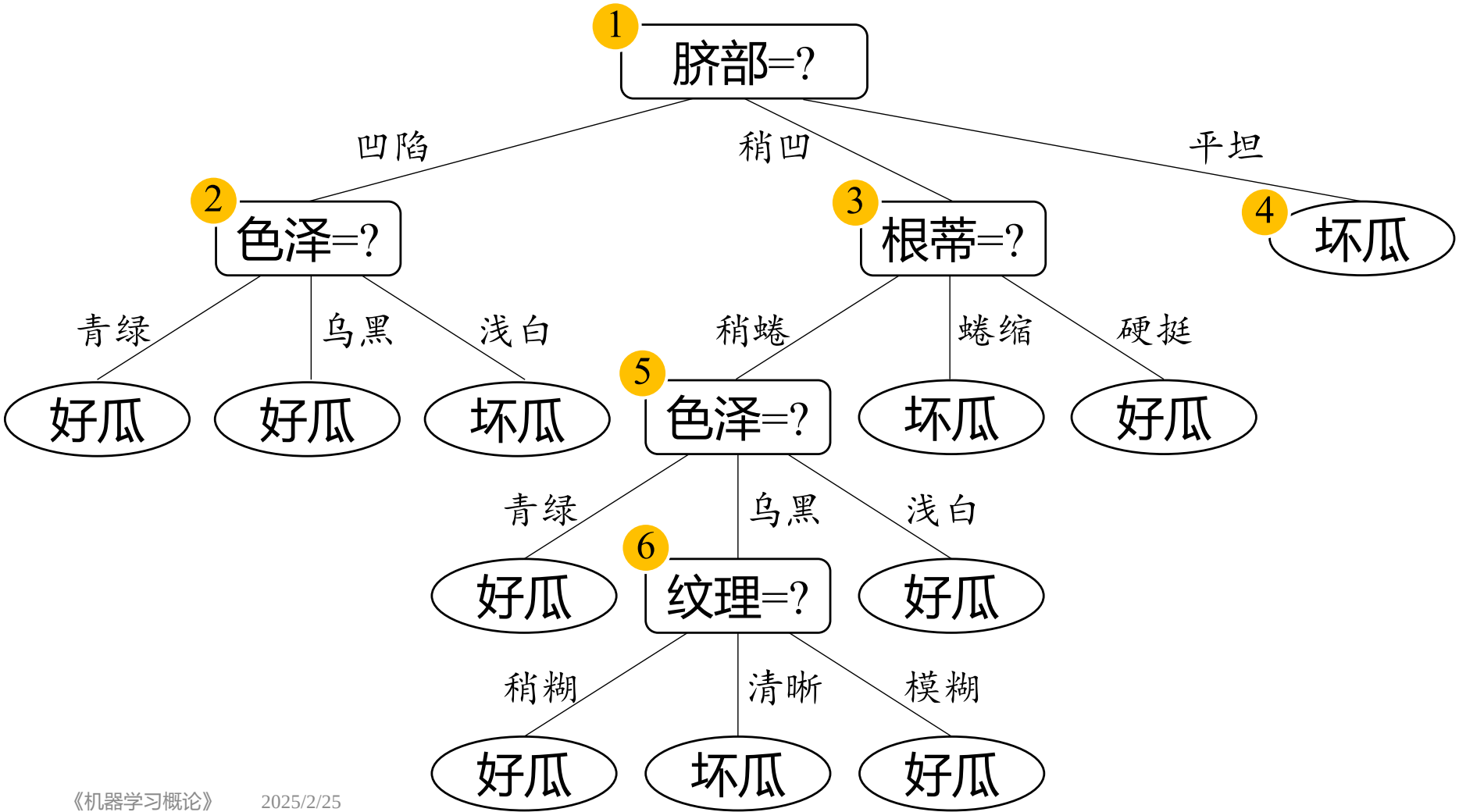
剪枝处理-预剪枝

- 决策树生成过程中，对每个结点在划分前先进行估计

- 若当前结点的划分不能带来决策树泛化性能提升，则停止划分并将当前结点记为叶结点，其类别标记为训练样例数最多的类别

剪枝处理

- 基于信息增益准则，选取属性“脐部”划分训练集
- 分别计算划分前及划分后的验证集精度，判断是否需要划分



剪枝处理—预剪枝

训练集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

验证集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

1 脐部=?

结点1：若不划分，则将其标记为叶结点，类别标记为训练样例中最多的类别，即好瓜。验证集中{4,5,8} 被分类正确，得到验证集精度为$\frac{3}{7} \times 100\% = 42.9\%$

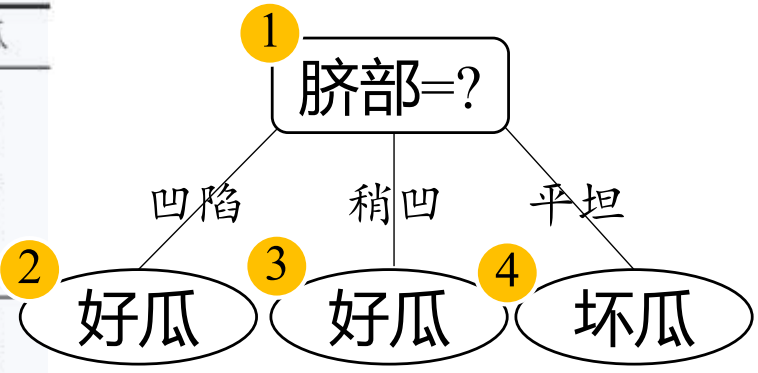
剪枝处理—预剪枝

训练集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

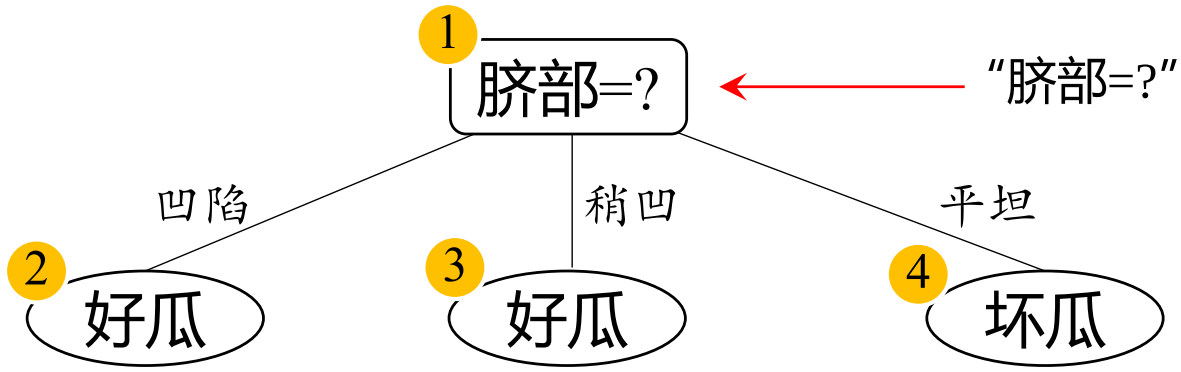
验证集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否



结点1：若划分，根据结点2 3 4的训练样例，将这3个结点分别标记为“好瓜”、“好瓜”、“坏瓜”。此时，验证集中编号{4,5,8,11,12}样例被划分正确，验证集精度$\frac{5}{7} \times 100\% = 71.4\%$

剪枝处理—预剪枝



“脐部=?”

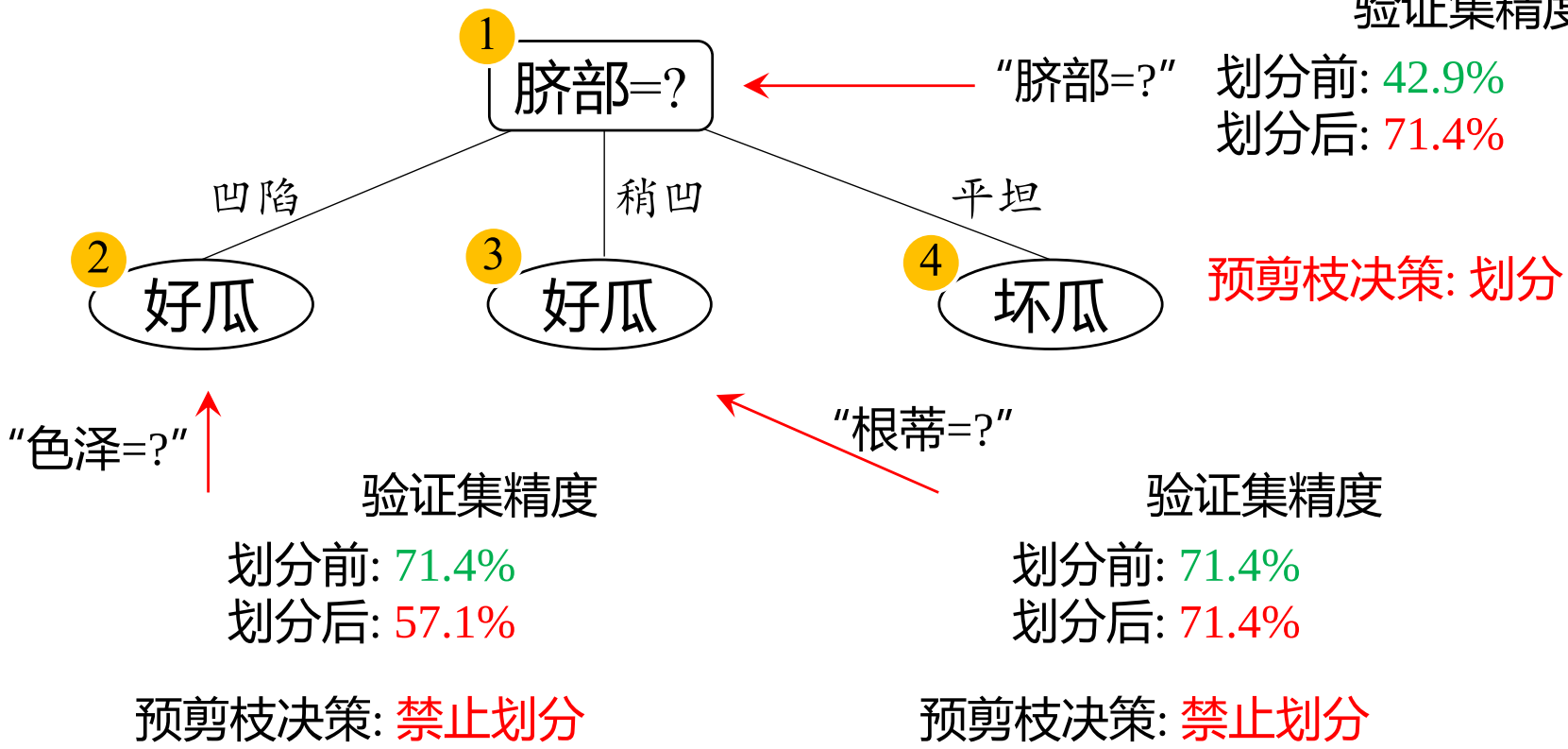
验证集精度

划分前: 42.9%

划分后: 71.4%

预剪枝决策: 划分

剪枝处理—预剪枝



最终得到仅有一层划分的决策树，称为"**决策树桩**"

剪枝处理—预剪枝

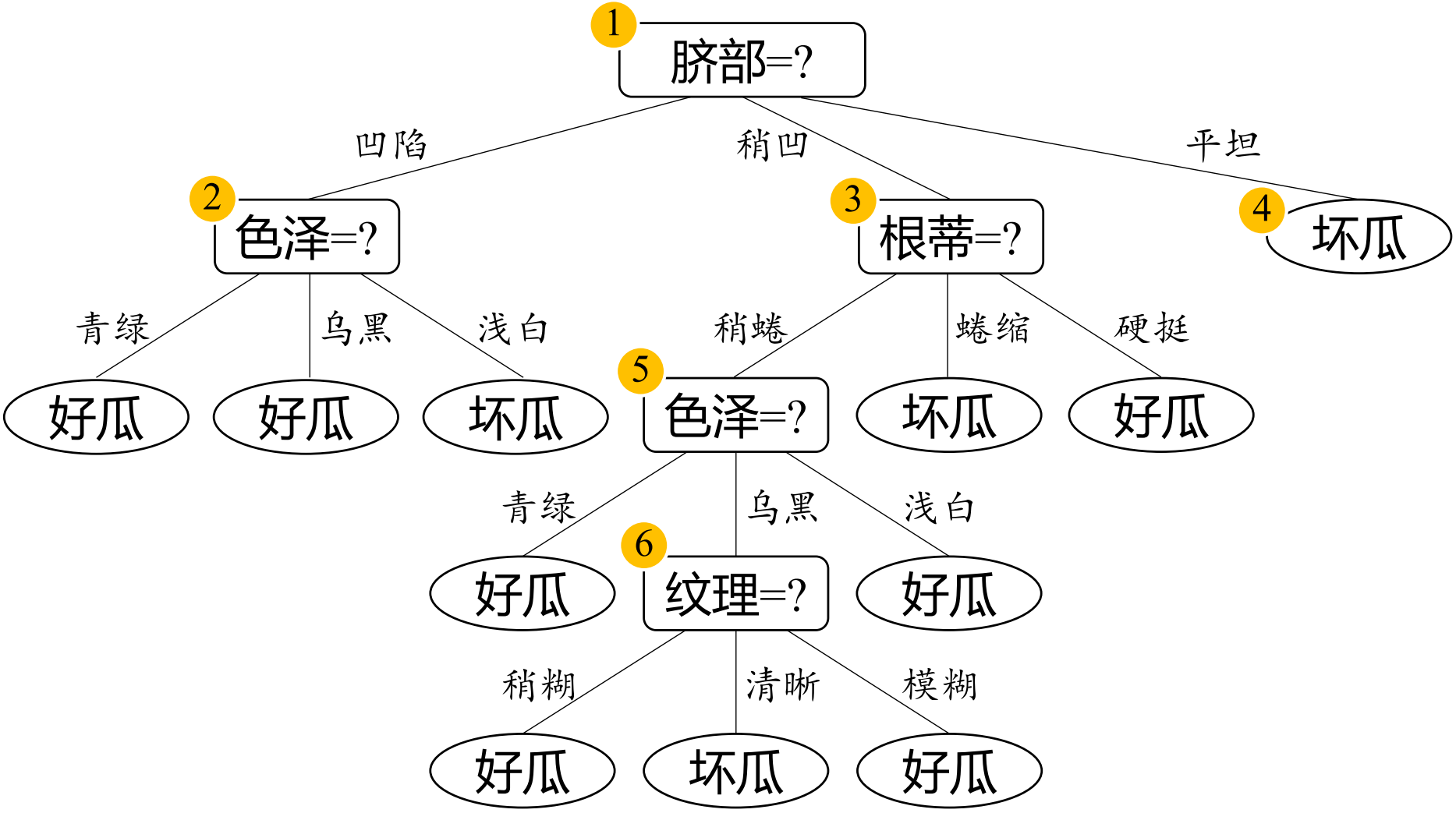
优点

- 降低过拟合风险
- 显著减少训练时间和测试时间开销

缺点

- 欠拟合风险：有些分支的当前划分虽然不能提升泛化性能，但在其基础上进行的后续划分却有可能导致性能显著提高。预剪枝基于“贪心”本质禁止这些分支展开，带来了欠拟合风险

剪枝处理—后剪枝

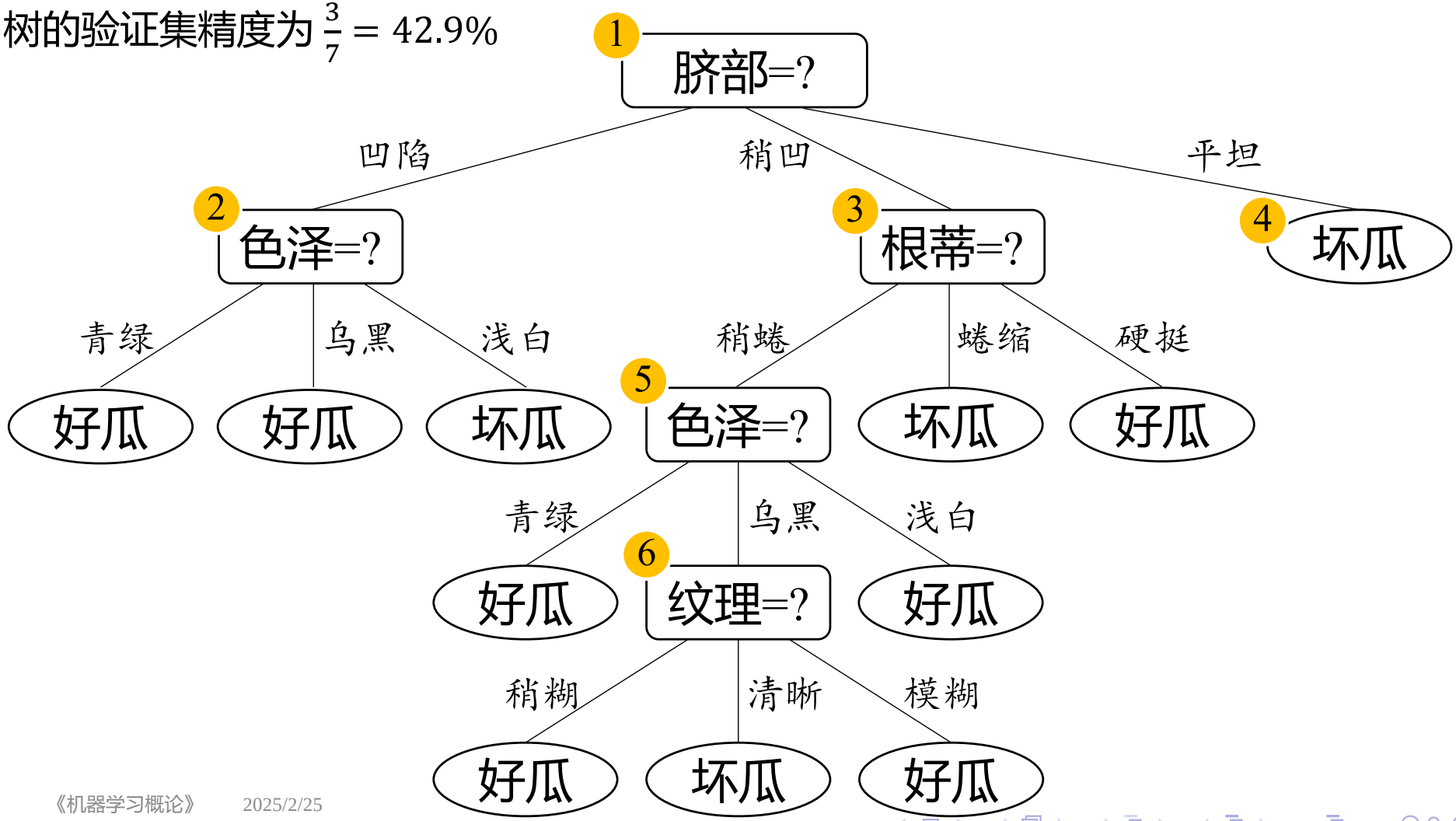


先从训练集生成一棵完整的决策树，然后自底向上地对非叶结点进行考察，若将该结点对应的子树替换为叶结点能带来决策树泛化性能提升，则将该子树替换为叶结点

剪枝处理—后剪枝

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

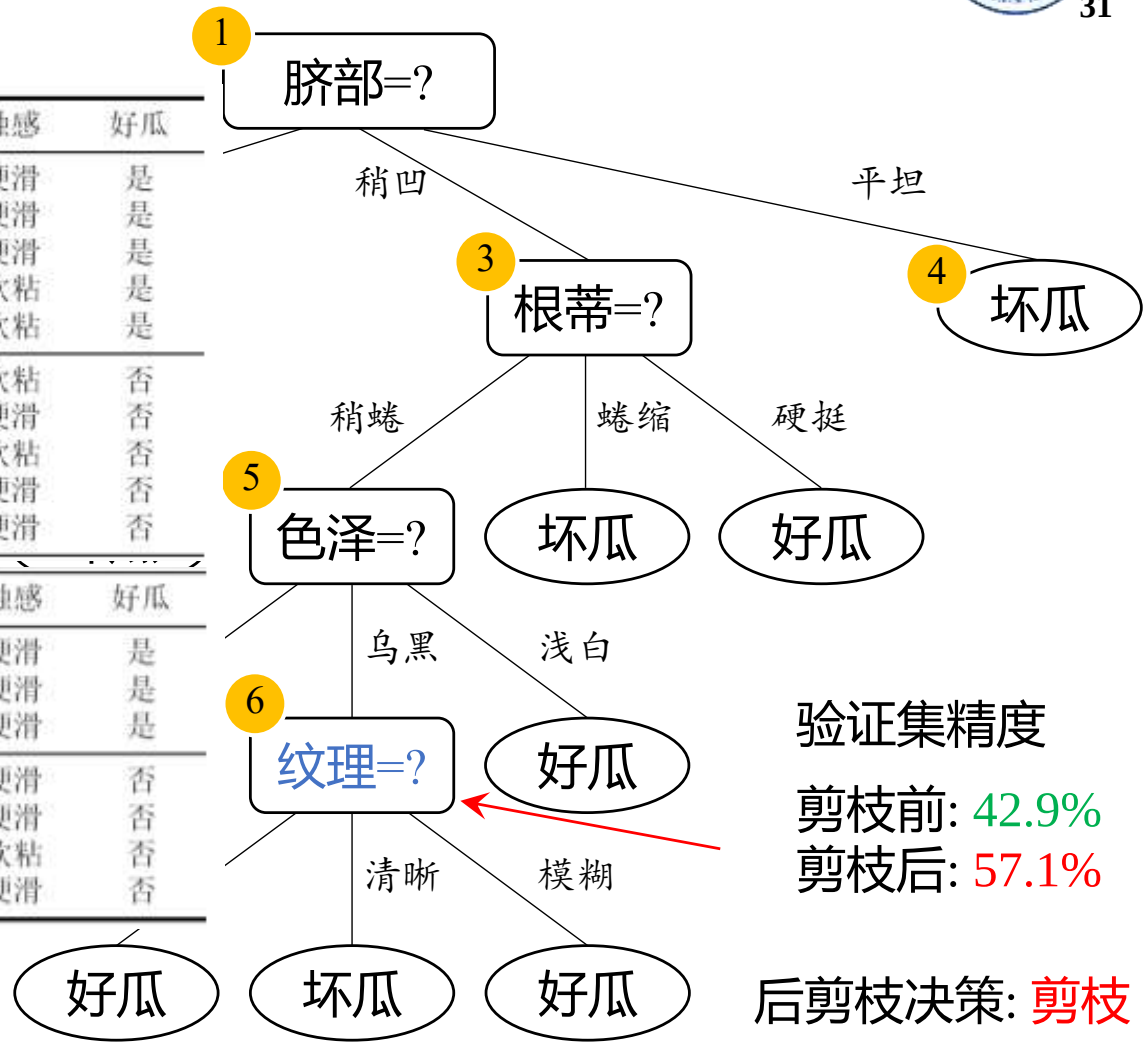
首先生成一棵完整的决策树，验证集{4,11,12 }判断正确，该决策树的验证集精度为 $\frac{3}{7} = 42.9\%$



剪枝处理—后剪枝

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否



验证集精度

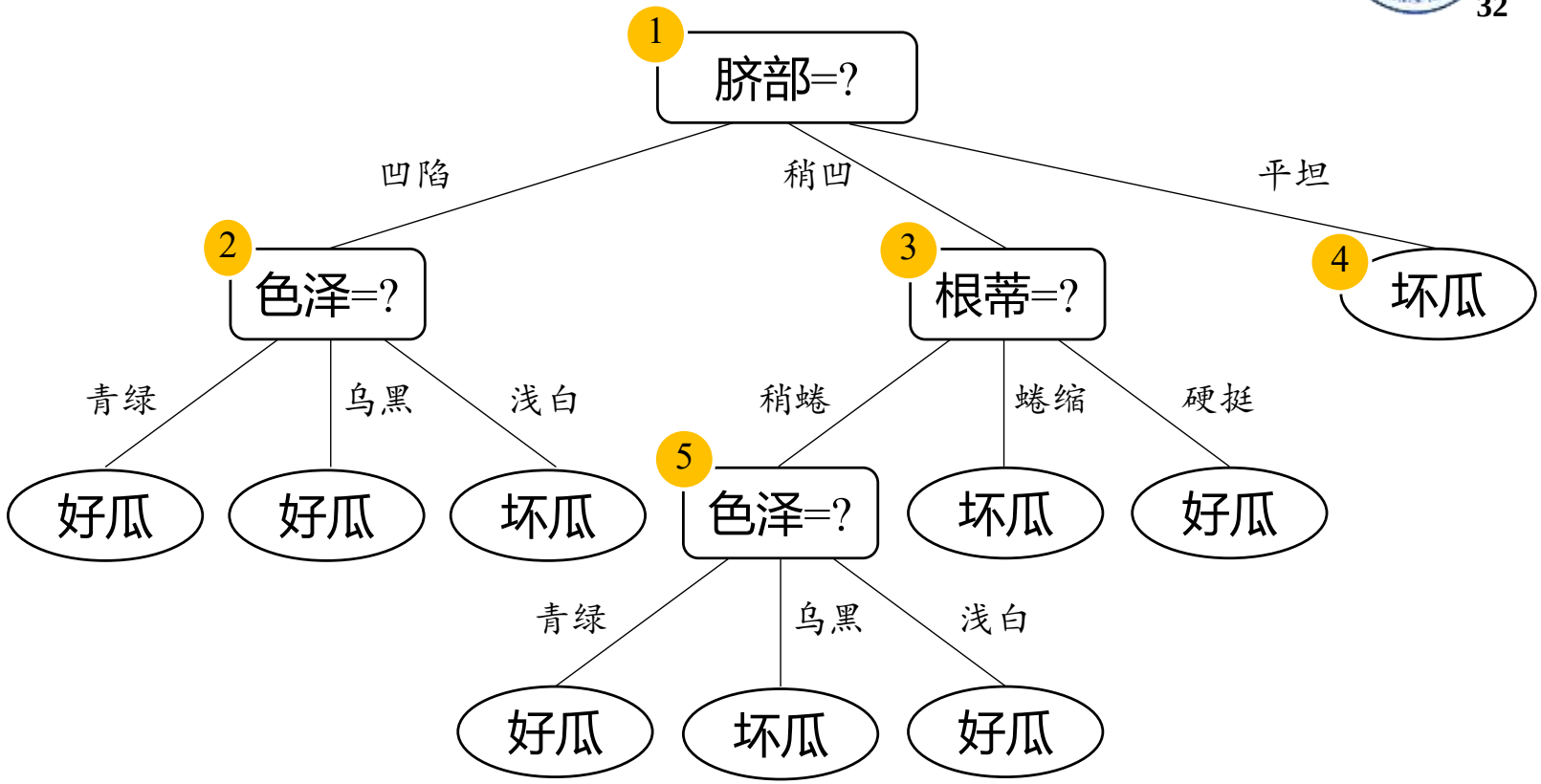
剪枝前: 42.9%

剪枝后: 57.1%

后剪枝决策: 剪枝

首先考虑结点 6，若将其替换为叶结点，根据落在其上的训练样本{7,15}将其标记为“好瓜”。 验证集{4,8,11,12}判断正确，精度提高至57.1%。

剪枝处理—后剪枝

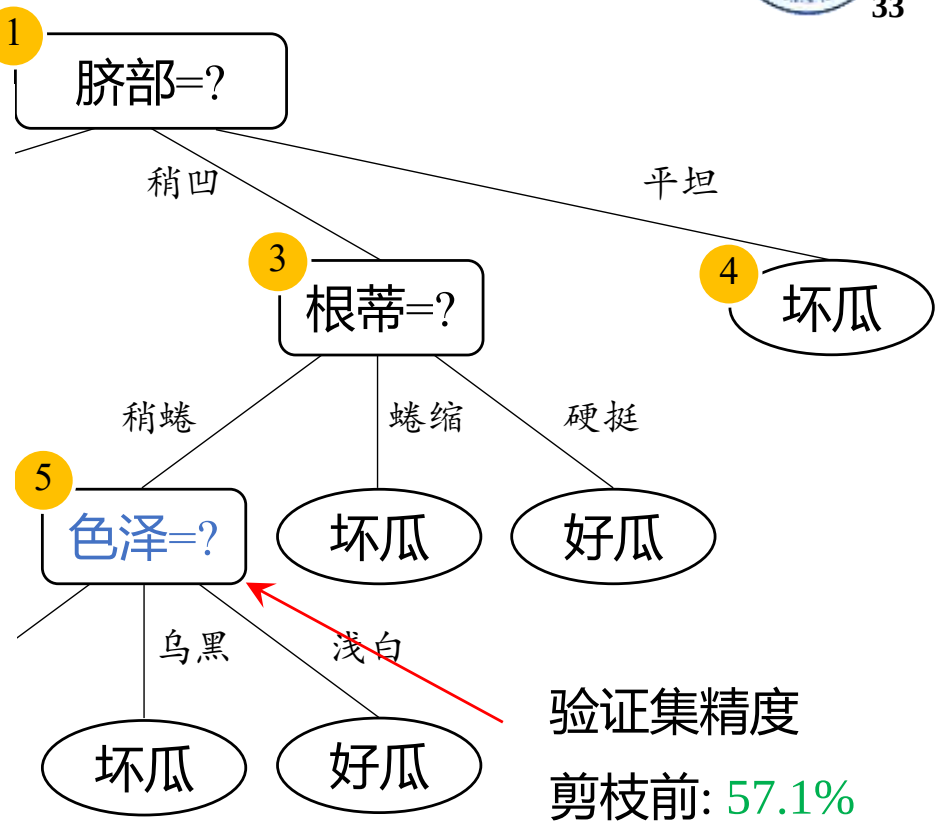


首先考虑结点 6 ，若将其替换为叶结点，根据落在其上的训练样本{7,15}将其标记为“好瓜”。 验证集{4,8,11,12}判断正确，精度提高至57.1%。

剪枝处理—后剪枝

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否



验证集精度

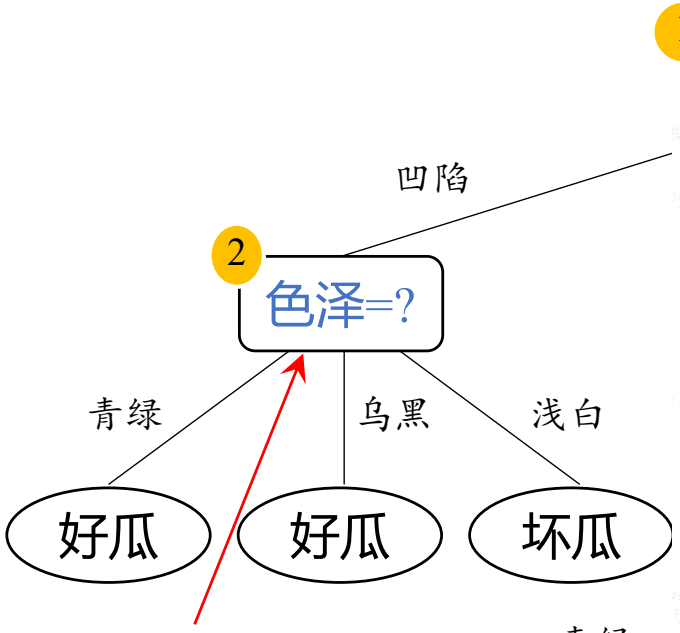
剪枝前: 57.1%

剪枝后: 57.1%

后剪枝决策: 不剪枝

然后考虑结点 5 ，若将其替换为叶结点，根据落在其上的训练样本{6,7,15}将其标记为“好瓜”。验证集{4,8,11,12}判断正确，精度仍然为57.1%。

剪枝处理—后剪枝



编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否

验证集精度

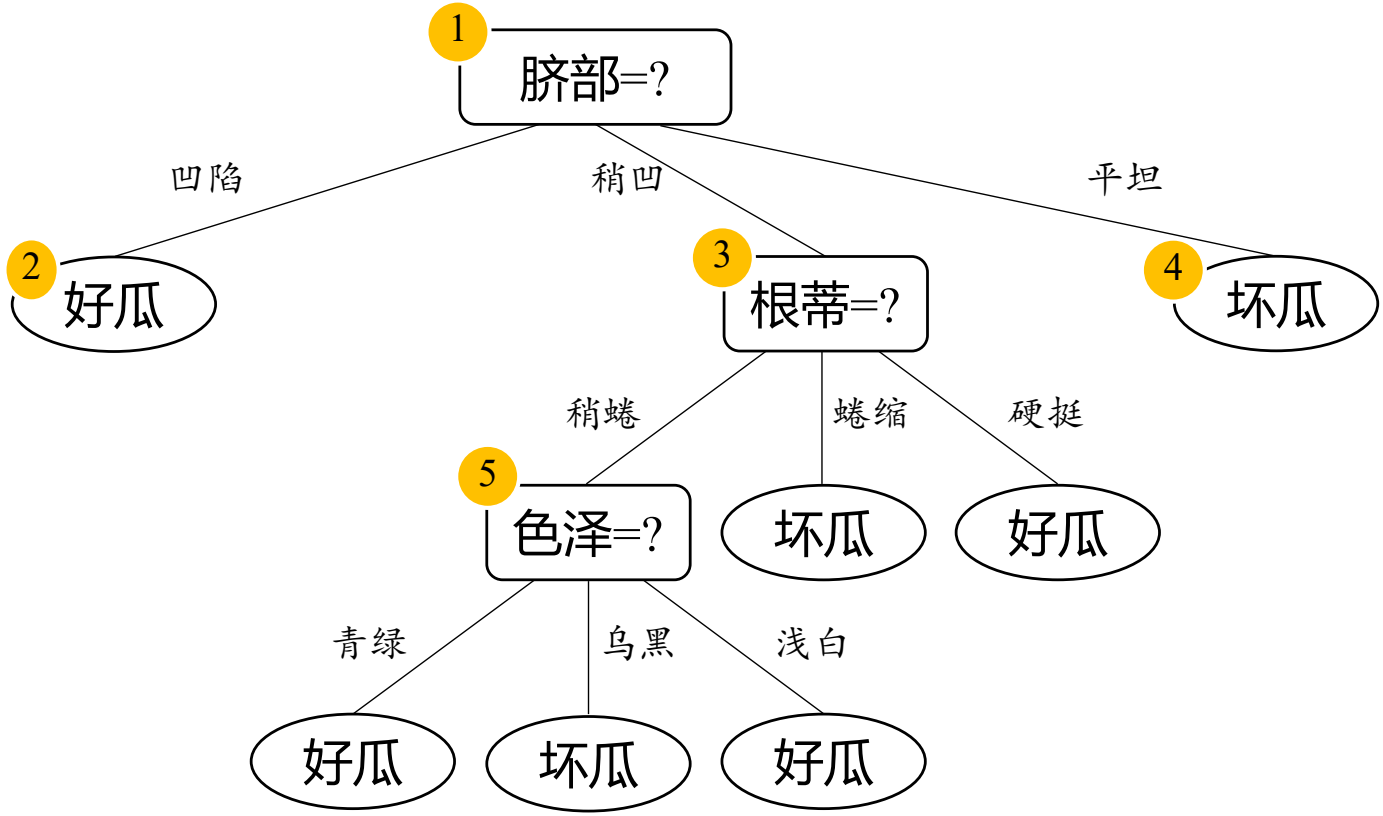
剪枝前: 57.1%

剪枝后: 71.4%

后剪枝决策: 剪枝

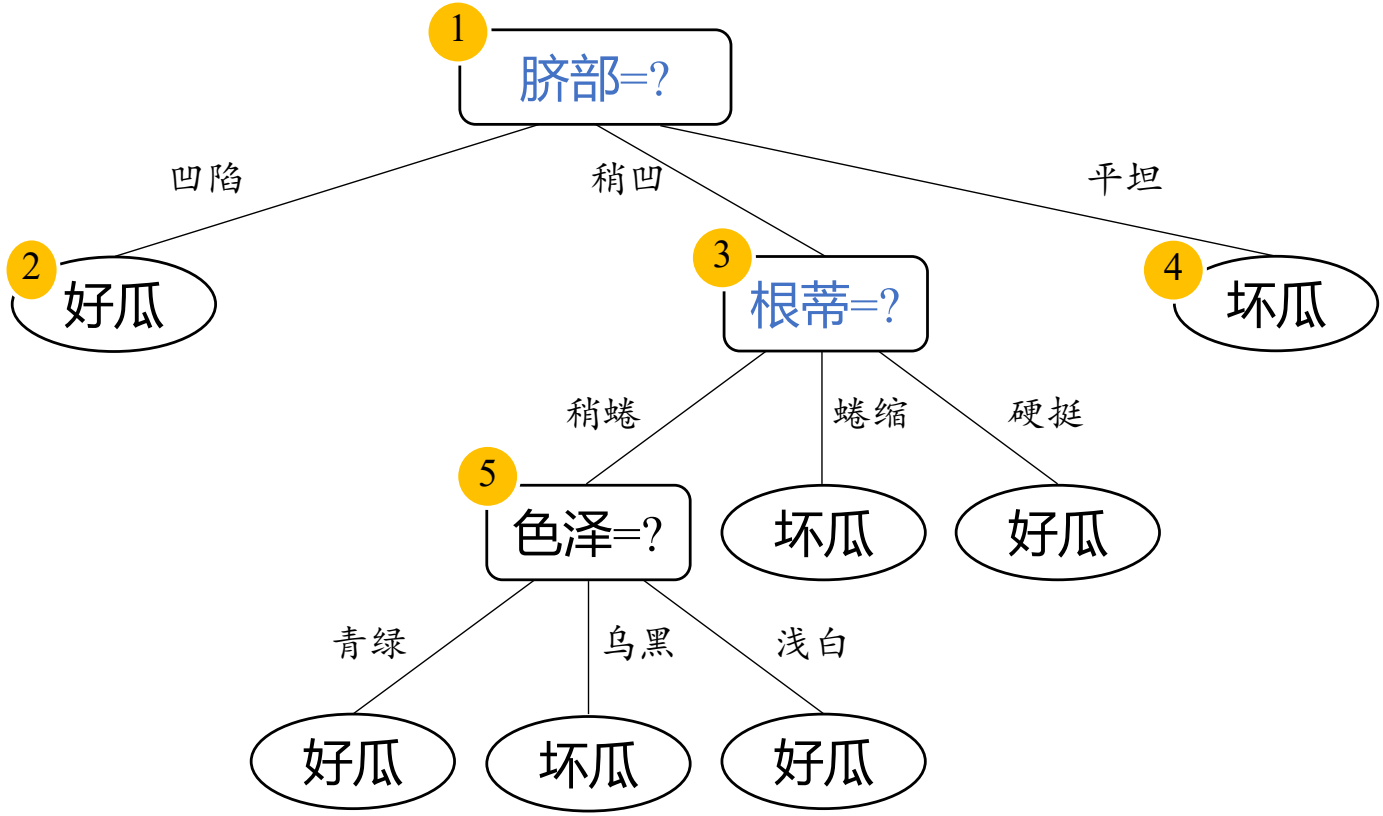
然后考虑结点 2 ，若将其替换为叶结点，根据落在其上的训练样本{1,2,3,14}将其标记为“好瓜”。验证集{4,5,8,11,12}判断正确，精度提高至71.4%。

剪枝处理—后剪枝



然后考虑结点 2 ，若将其替换为叶结点，根据落在其上的训练样本{1,2,3,14}将其标记为“好瓜”。验证集{4,5,8,11,12}判断正确，精度提高至71.4%。

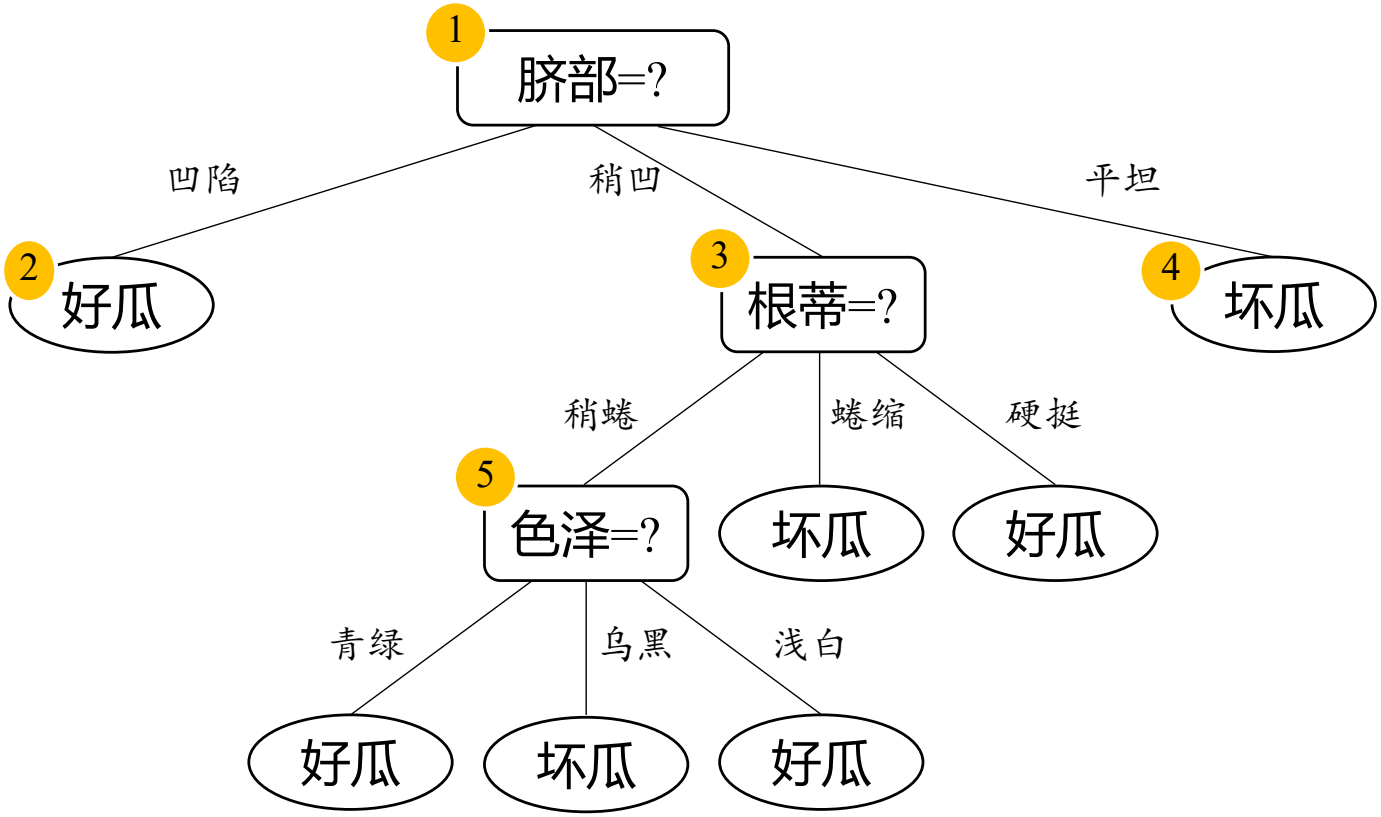
剪枝处理—后剪枝



考虑结点1 3，先后替换为叶结点，验证集精度均未提升，则分支得到保留

剪枝处理—后剪枝

最终基于后剪枝策略得到的决策树如图所示



剪枝处理—后剪枝

优点

- 后剪枝比预剪枝保留了更多的分支，欠拟合风险小，泛化性能往往优于预剪枝决策树

缺点

- 训练时间开销大：后剪枝过程是在生成完全决策树之后进行的，需要自底向上对所有非叶结点逐一考察

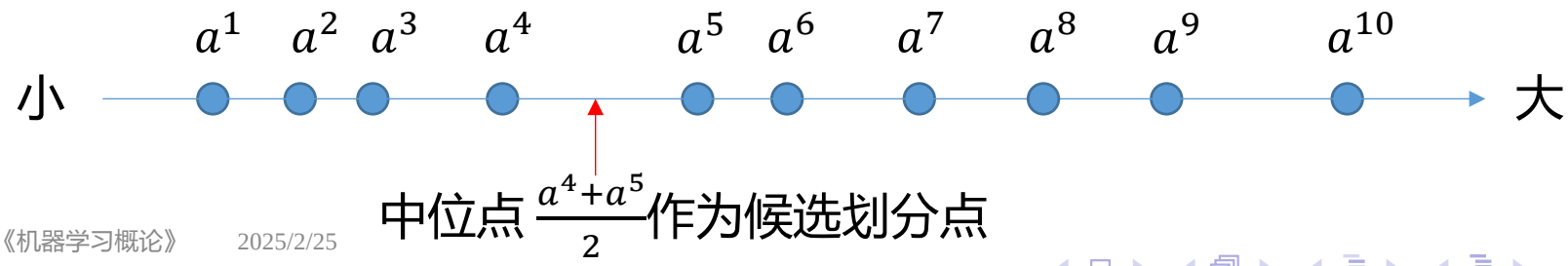
连续与缺失值—连续值处理

- 连续属性离散化(二分法)

第一步：
- 假定连续属性a在样本集 D上出现 n个不同的取值，从小到大排列，记为 $a^1, a^2, \cdots, a^n$。
- 基于划分点 t，可将 D 分为子集D_t^- 和D_t^+，
- D_t^-包含那些在属性a上取值不大于t的样本
- D_t^+包含那些在属性a上取值大于t的样本

考虑包含$n-1$ 个元素的候选划分点集合

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \middle| 1 \le i \le n-1 \right\}$$



连续与缺失值—连续值处理

- 连续属性离散化(二分法)

第二步：
- 采用离散属性值方法，考察这些划分点，选取最优的划分点进行样本集合的划分

$$\text{Gain}(D,a) = \max_{t\in T_a} \text{Gain}(D,a,t)$$

$$= \max_{t\in T_a} \left(\text{Ent}(D) - \sum_{\lambda\in\{-,+\}} \frac{D_t^\lambda}{|D|} Ent\left(D_t^\lambda\right) \right)$$

其中$\text{Gain}(D,a,t)$是样本集D基于划分点t二分后的信息增益
于是可选择使 $\text{Gain}(D,a,t)$ 最大化的划分点

连续与缺失值—连续值处理

- 连续值处理实例

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

对属性“密度”，其候选划分点集合包含17个候选值：

$T_{密度}$={0.244,0.294,0.351,0.381,0.420,0.459,0.518,0.574,0.600,0.621,0.636,0.648,0.661,0.681,0.708,0.746}

属性“密度”划分点为0.381

对属性“含糖量”进行同样处理

属性“含糖量”划分点为 0.126，信息增益为0.349

$$\text{Gain}(D,a,0.244)=0.998-\frac{16}{17}\times\left(-\frac{8}{16}\log_2\frac{8}{16}-\frac{8}{16}\log_2\frac{8}{16}\right)=0.0568$$

$$\vdots$$

$$\text{Gain}(D,a,0.381)=0.998-\frac{13}{17}\times\left(-\frac{8}{13}\log_2\frac{8}{13}-\frac{5}{13}\log_2\frac{5}{13}\right)=0.2629$$

$$\vdots$$

连续与缺失值—连续值处理

- 连续值处理实例

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

$$\mathrm{Gain}(D, \text{色泽}) = 0.109$$

$$\mathrm{Gain}(D, \text{根蒂}) = 0.143$$

$$\mathrm{Gain}(D, \text{敲声}) = 0.141$$

$$\mathrm{Gain}(D, \text{纹理}) = 0.381$$

$$\mathrm{Gain}(D, \text{脐部}) = 0.289$$

$$\mathrm{Gain}(D, \text{触感}) = 0.006$$

$$\mathrm{Gain}(D, \text{密度}) = 0.262$$

$$\mathrm{Gain}(D, \text{含糖率}) = 0.349$$

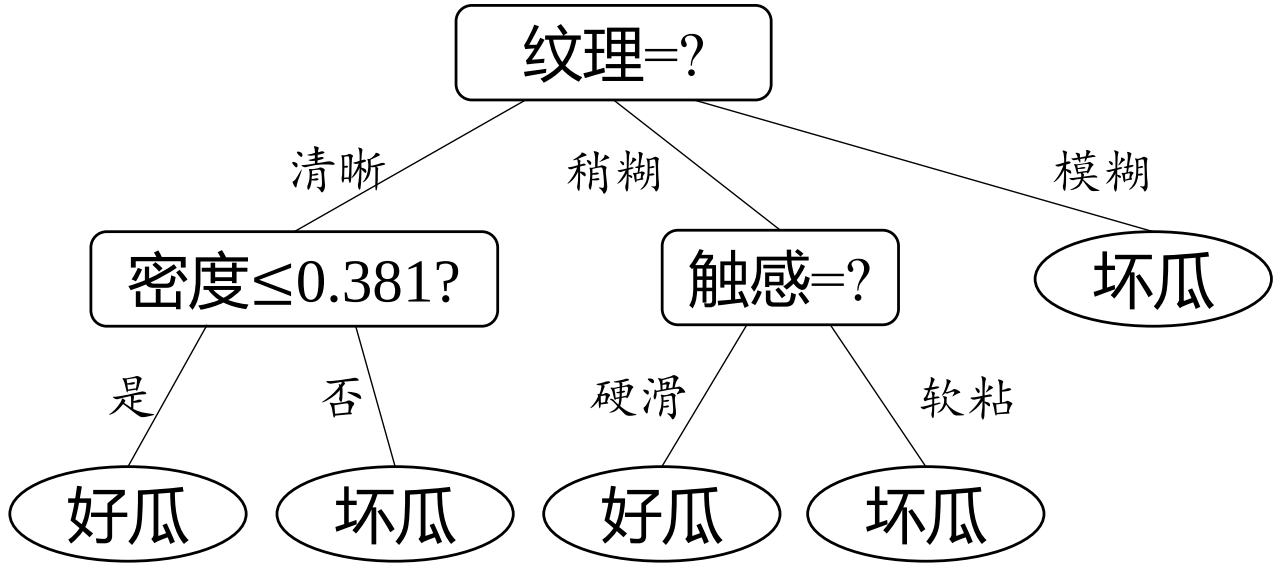
连续与缺失值—连续值处理

- 连续值处理实例

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响						
4	青绿	蜷缩	沉闷						
5	浅白	蜷缩	浊响						
6	青绿	稍蜷	浊响						
7	乌黑	稍蜷	浊响						
8	乌黑	稍蜷	浊响						
9	乌黑	稍蜷	沉闷						
10	青绿	硬挺	清脆						
11	浅白	硬挺	清脆						
12	浅白	蜷缩	浊响						
13	青绿	稍蜷	浊响						
14	浅白	稍蜷	沉闷						
15	乌黑	稍蜷	浊响						
16	浅白	蜷缩	浊响						
17	青绿	蜷缩	沉闷						

$$\text{Gain}(D, \text{色泽}) = 0.109$$

$$\text{Gain}(D, \text{根蒂}) = 0.143$$



与离散属性不同，若当前结点划分属性为连续属性，该属性还可作为其后代结点的划分属性

连续与缺失值—缺失值处理

- 不完整样本，即样本的属性值缺失
- 仅使用无缺失的样本进行学习?

对数据信息极大的浪费

- 使用有缺失值的样本，需要解决哪些问题?

Q1：如何在属性缺失的情况下进行划分属性选择?

Q2：给定划分属性,若样本在该属性上的值缺失，如何对样本进行划分?

连续与缺失值—缺失值处理

Q1：如何在属性缺失的情况下进行划分属性选择?

- $\tilde{D}$ 表示D中在属性a上没有缺失值的样本子集，$\tilde{D}^v$表示$\tilde{D}$中在属性a上取值为a^v的样本子集，$\tilde{D}_k$表示$\tilde{D}$中属于第k类的样本子集

$$\mathrm{Gain}(D,a) = \rho Gain(\tilde{D},a)$$

$$= \rho\left(\mathrm{Ent}(\tilde{D}) - \sum_{v=1}^{V}\tilde{r}_v \mathrm{Ent}(\tilde{D}^v)\right)$$

$$\rho = \frac{\sum_{x\in\tilde{D}} w_x}{\sum_{x\in D} w_x}$$

$$\mathrm{Ent}(\tilde{D}) = -\sum_i \tilde{p}_k \log_2 \tilde{p}_k$$

$$\tilde{r}_v = \frac{\sum_{x\in\tilde{D}^v} w_x}{\sum_{x\in\tilde{D}} w_x}$$

$$\tilde{p}_k = \frac{\sum_{x\in\tilde{D}_k} w_x}{\sum_{x\in\tilde{D}} w_x}$$

w_x为每个样本x的权重

连续与缺失值—缺失值处理

Q2：给定划分属性,若样本在该属性上的值缺失，如何对样本进行划分?

- 若样本x在划分属性a上的取值已知，则将x划入与其取值对应的子结点，且样本权值在子结点中保持为w_x

- 若样本x在划分属性 a上的取值未知，则将x同时划入所有子结点，且样本权值在与属性值a^v对应的子结点中调整为 $r_v \times w_x$

直观来看，相当于让同一个样本以不同概率划入不同的子结点中去

连续与缺失值—缺失值处理

- 缺失值处理实例

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	–	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	–	是
3	乌黑	蜷缩	–	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	–	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	–	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	–	稍凹	硬滑	是
9	乌黑	–	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	–	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	–	否
12	浅白	蜷缩	–	模糊	平坦	软粘	否
13	–	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	–	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	–	沉闷	稍糊	稍凹	硬滑	否

- 学习开始时，根结点包含样本集D中全部17个样例，各样例的权值均为1
- 以属性“色泽”为例，该属性上无缺失值的样例子集$\tilde{D}$包含 14个样例

$\tilde{D}$的信息熵 $\mathrm{Ent}(\tilde{D}) = -\left(\frac{6}{14}\log_2\frac{6}{14} + \frac{8}{14}\log_2\frac{8}{14}\right) = 0.985$

连续与缺失值—缺失值处理

- 缺失值处理实例

令 $\tilde{D}^1$，$\tilde{D}^2$，$\tilde{D}^3$分别表示在属性“色泽”上取值为“青绿”“乌黑”以及“浅白”的样本子集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	–	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	–	是
3	乌黑	蜷缩	–	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	–	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	–	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	–	稍凹	硬滑	是
9	乌黑	–	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	–	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	–	否
12	浅白	蜷缩	–	模糊	平坦	软粘	否
13	–	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	–	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	–	沉闷	稍糊	稍凹	硬滑	否

$$\mathrm{Ent}(\tilde{D}^1) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

$$\mathrm{Ent}(\tilde{D}^2) = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) = 0.918$$

$$\mathrm{Ent}(\tilde{D}^3) = -\left(\frac{0}{4}\log_2\frac{0}{4} + \frac{4}{4}\log_2\frac{4}{4}\right) = 0$$

$$\mathrm{Gain}(D, \text{色泽}) = \rho \times \mathrm{Gain}(\tilde{D}, \text{色泽}) = \frac{14}{17} \times 0.306 = 0.252$$

$\tilde{D}$上属性“色泽”的信息增益为 $\mathrm{Gain}(\tilde{D}, \text{色泽}) = \mathrm{Ent}(\tilde{D}) - \sum_v \tilde{r}_v \mathrm{Ent}(\tilde{D}^v)$

$$= 0.985 - \left(\frac{4}{14} \times 1 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0\right)$$

连续与缺失值—缺失值处理

- 类似地可计算出所有属性在数据集上的信息增益

$\text{Gain}(D, \text{色泽}) = 0.252$　$\text{Gain}(D, \text{敲声}) = 0.145$　$\text{Gain}(D, \text{脐部}) = 0.289$

$\text{Gain}(D, \text{根蒂}) = 0.171$　$\text{Gain}(D, \text{纹理}) = 0.424$　$\text{Gain}(D, \text{触感}) = 0.006$

进入“纹理=清晰”分支 7个样本

进入“纹理=稍糊”分支 5个样本

进入“纹理=模糊”分支 3个样本

样本权重在各子结点仍为1

在属性“纹理”上出现缺失值，样本8和10同时进入3个分支，调整8和10在3分分支权值分别为7/15，5/15，3/15

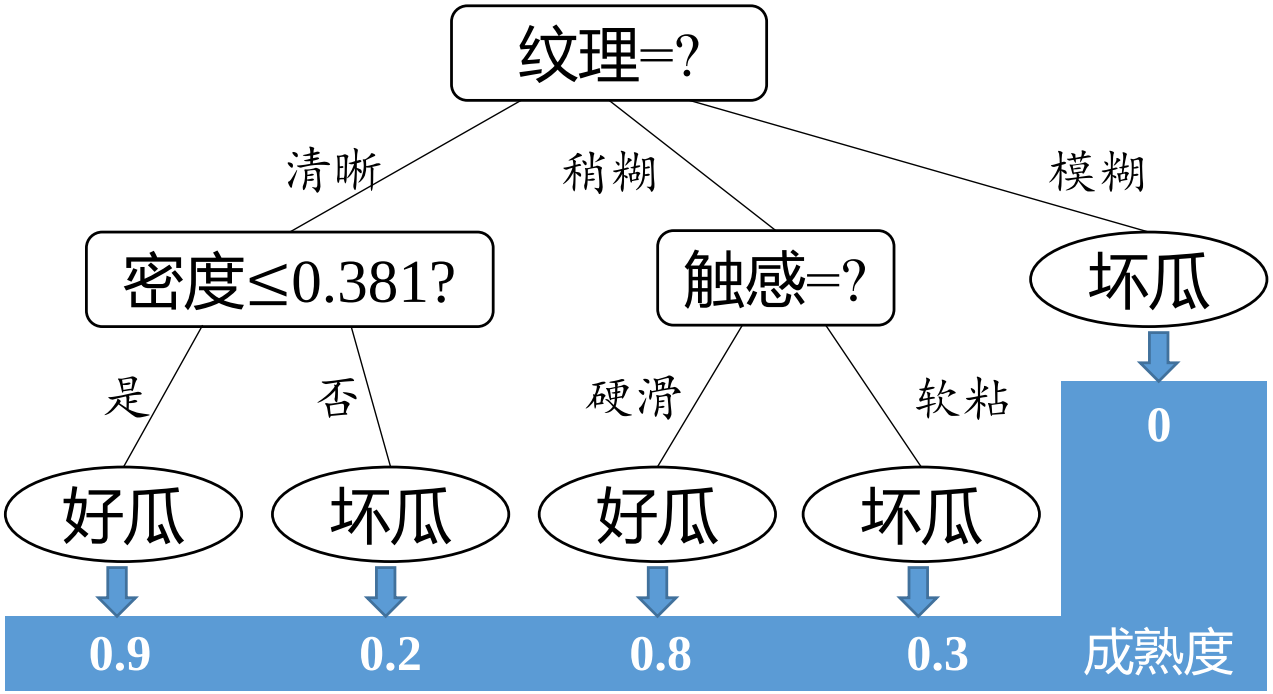
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	–	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	–	是
3	乌黑	蜷缩	–	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	–	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	–	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	–	稍凹	硬滑	是
9	乌黑	–	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	–	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	–	否
12	浅白	蜷缩	–	模糊	平坦	软粘	否
13	–	稍蜷	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	–	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	–	沉闷	稍糊	稍凹	硬滑	否

回归树

当标记为离散值时，叶子节点标记为数量最多的类别

若标记为连续值时，叶子节点应该如何标记？（均值？中位数？）

如何选择划分节点？



回归树

- 假设待划分节点中的数据为 $D = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_m, y_m)\}$
- 离散属性a有V个可能的取值$\{a^1, a^2, \cdots, a^V\}$
- 若用a来进行划分，则会产生V个分支结点；第v个分支结点包含D中所有在属性a上取值为a^v的样本，记为D^v

- 若优化均方误差，

划分前误差 $\min_{\mathrm{c}} \sum_{i} (y_i - c)^2 \Rightarrow c = \mathrm{avg}(\{y|(\boldsymbol{x}, y) \in D\})$

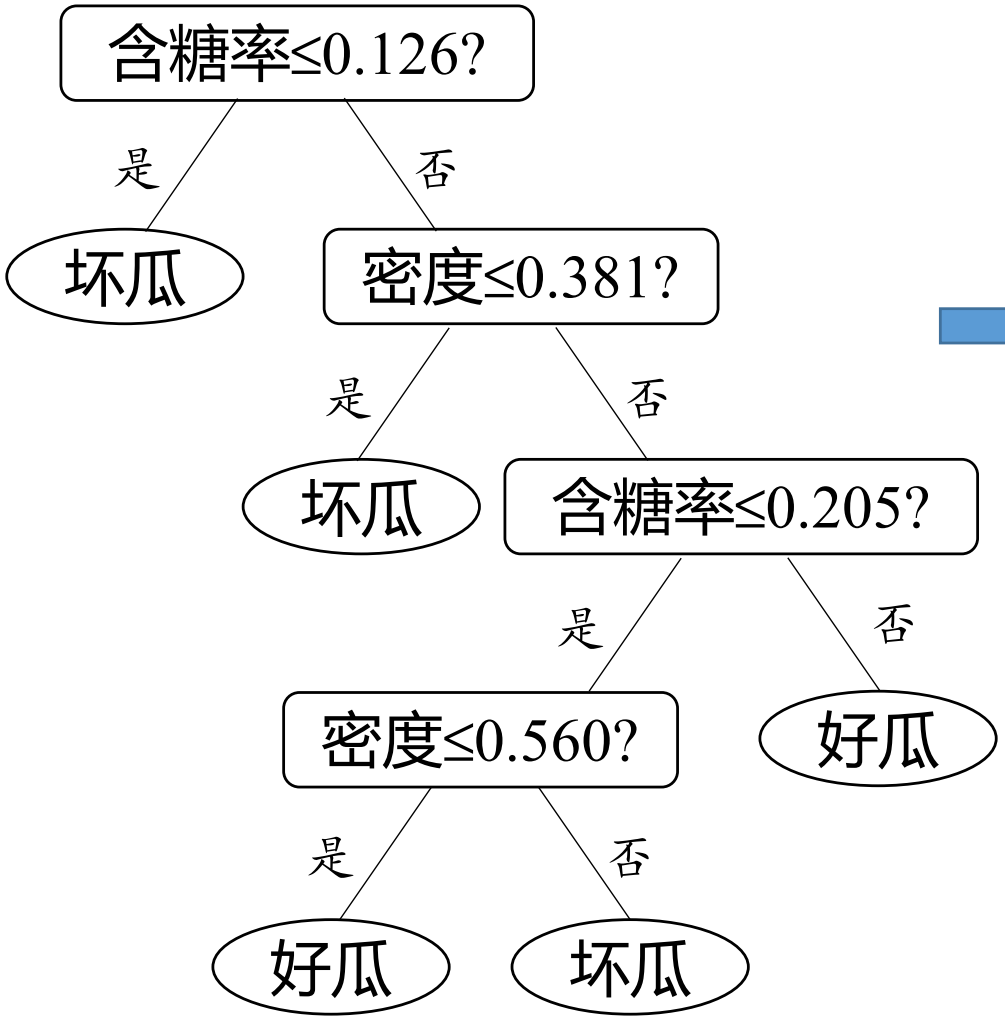
划分后误差 $\mathrm{error(D, a)} = \sum_{v=1}^{V} \min_{c^v} \sum_{(\boldsymbol{x}_i, y_i) \in D^v} (y_i - c^v)^2 \Rightarrow c^v = \mathrm{avg}(\{y|(\boldsymbol{x}, y) \in D^v\})$

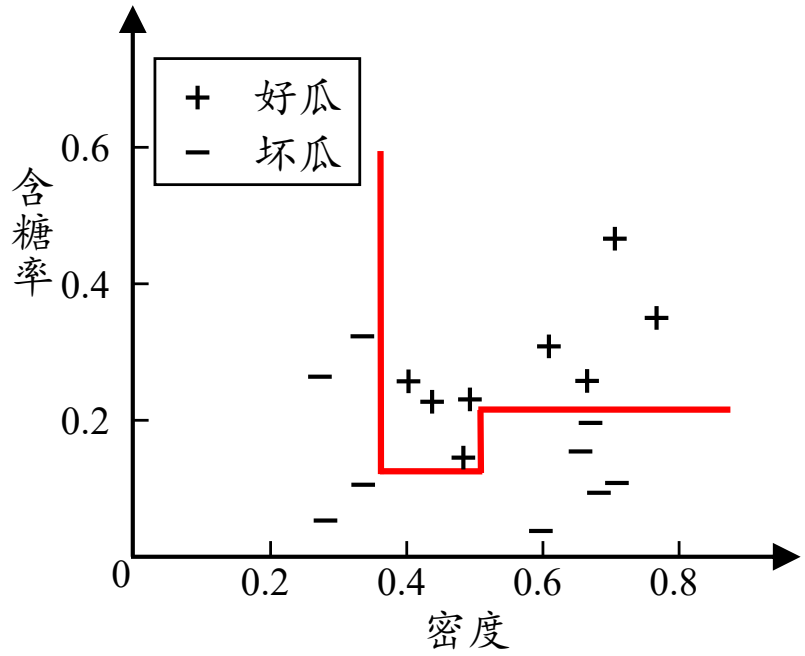
- 选择划分后误差最小的属性作为最优划分属性

$$a^\star = \arg\min_{a \in A} \mathrm{error}(D, a)$$

多变量决策树

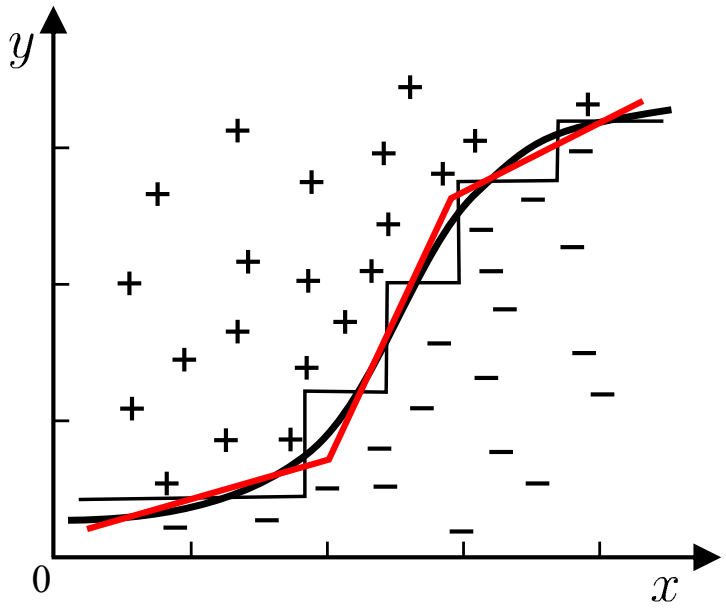
- 单变量决策树分类边界:轴平行





多变量决策树

- 单变量决策树分类边界:轴平行



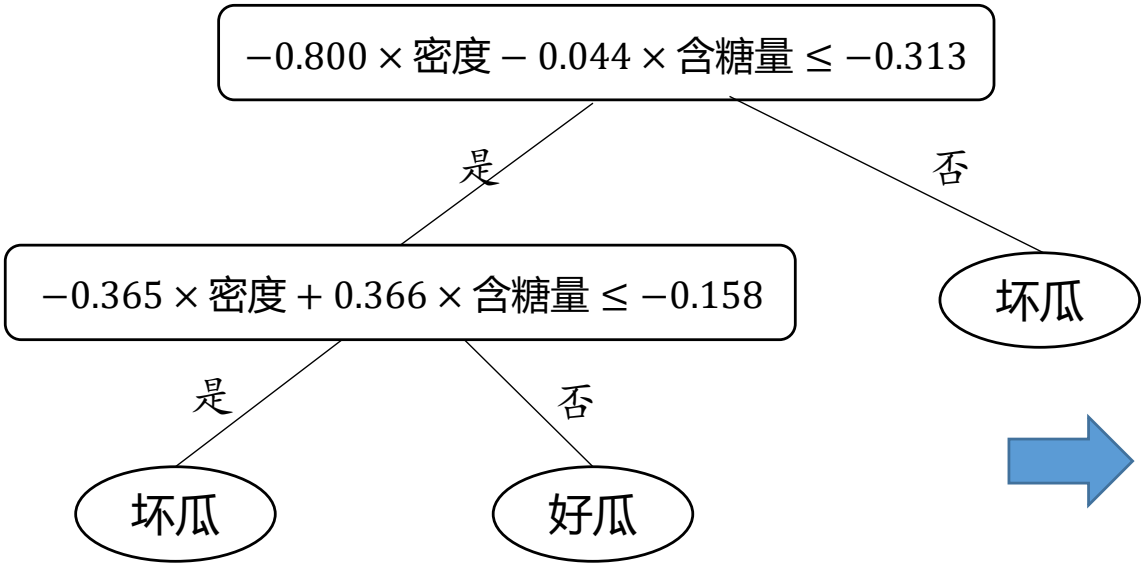
非叶节点不再是仅对某个属性，而是对属性的线性组合 进行测试

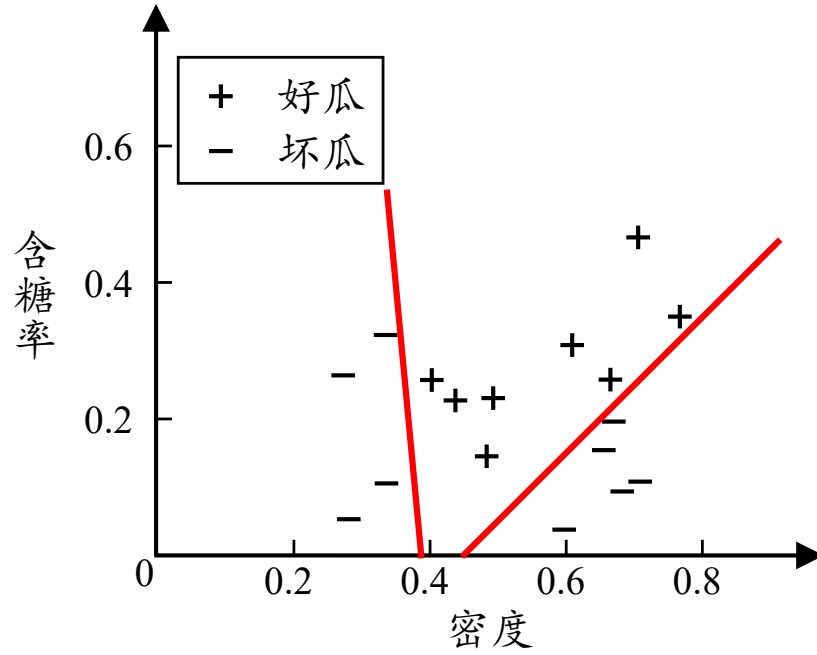
每个非叶结点是一个形如$\sum_{i=1}^{d} w_i a_i = t$的线性分类器，其中$w_i$是属性$a_i$的权值，$w_i$和$t$可在该结点所含的样本集和属性集上学得

- 多变量决策树

多变量决策树

- 多变量决策树





习题

- 4.1

- 4.9

- 假设离散随机变量$X \in \{1, \cdots, K\}$，其取值为k的概率$P(X = k) = p_k$，其熵为$H(\boldsymbol{p}) = -\sum_k p_k \log_2 p_k$，试用朗格朗日乘子法证明熵最大的分布为均匀分布

习题

- 下表表示的二分类数据集，具有三个属性A,B,C，样本标记为两类“+”，“-”。请运用你学过的知识完成如下问题：

实例	A	B	C	类别
1	T	T	1.0	+
2	T	T	6.0	+
3	T	F	5.0	-
4	F	F	4.0	+
5	F	T	7.0	-
6	F	T	3.0	-
7	F	F	8.0	-
8	T	F	7.0	+
9	F	T	5.0	-
10	F	F	2.0	+

(a) 整个训练样本关于类属性的熵是多少
(b) 数据集中A，B两个属性的信息增益各是多少
(c) 对于属性C，计算所有可能划分的信息增益
(d) 根据Gini指数，A和B两个属性哪个是最优划分
(e) 采用算法C4.5，构造决策树

第五章：神经网络

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

2016年AI技术重大突破

AlphaGo 以4:1比分赢得Lee Sedol

Versions	Hardware	Elo	Date	Results
AlphaGo Fan	176 GPUs, distributed	3,144	Oct-15	5:0 against Fan Hui
AlphaGo Lee	48 TPUs, distributed	3,739	Mar-16	4:1 against Lee Sedol
AlphaGo Master	4 TPUs, single machine	4,858	May-17	60:0 against professional players; Future of Go Summit
AlphaGo Zero (40 block)	4 TPUs, single machine	5,185	Oct-17	100:0 against AlphaGo Lee 89:11 against AlphaGo Master
AlphaZero (20 block)	4 TPUs, single machine	5,018	Dec-17	60:40 against AlphaGo Zero (20 block)

Rank	Name	♂♀	Flag	Elo
1	Shin Jinseo	♂		3800
2	Ke Jie	♂		3726
3	Park Junghwan	♂		3692
4	Gu Zihao	♂		3667
5	Lian Xiao	♂		3596
6	Fan Tingyu	♂		3589
7	Fan Yunruo	♂		3576
8	Jiang Weijie	♂		3574
9	Shin Minjun	♂		3572
10	Yang Dingxin	♂		3570
11	Xie Erhao	♂		3566
12	Mi Yuting	♂		3562
13	Xie Ke	♂		3557
14	Ding Hao	♂		3556
15	Tuo Jiaxi	♂		3555
16	Ichiriki Ryo	♂		3554
17	Chen Yaoye	♂		3550
18	Xu Jiayang	♂		3538
19	Iyama Yuta	♂		3538
20	Tong Mengcheng	♂		3530
21	Byun Sangil	♂		3527
22	Lee Donghoon	♂		3525
23	Tao Xinran	♂		3522
24	Zhao Chenyu	♂		3519
25	Li Weiqing	♂		3518
26	Kang Dongyun	♂		3504
27	Liao Yuanhe	♂		3501
28	Shi Yue	♂		3499

AlphaGo中的机器学习

- 策略网络

监督学习

- 预测人类如何下下一步棋

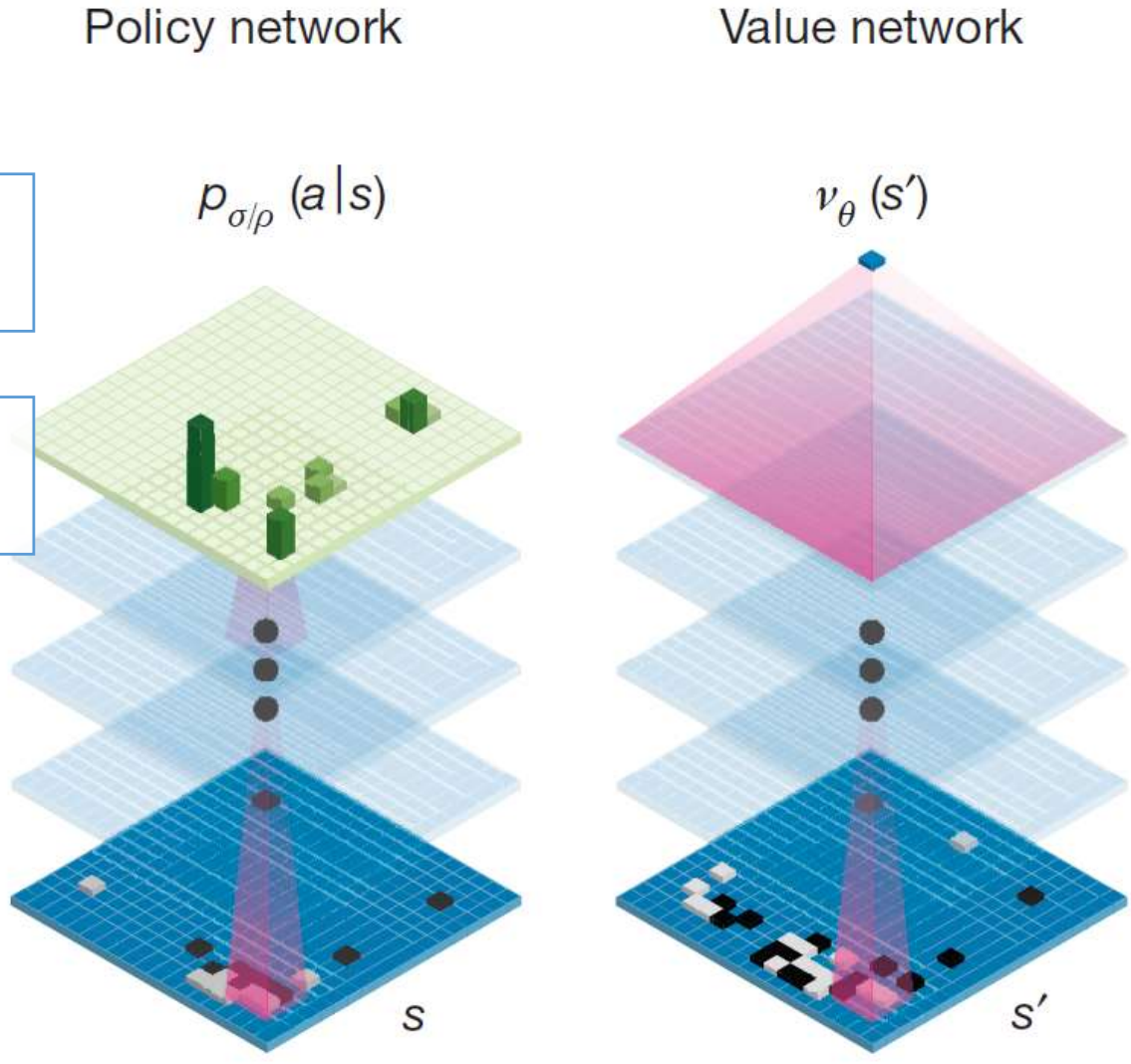
强化学习

- 学习如何下下一步棋以最大化赢率

- 值网络

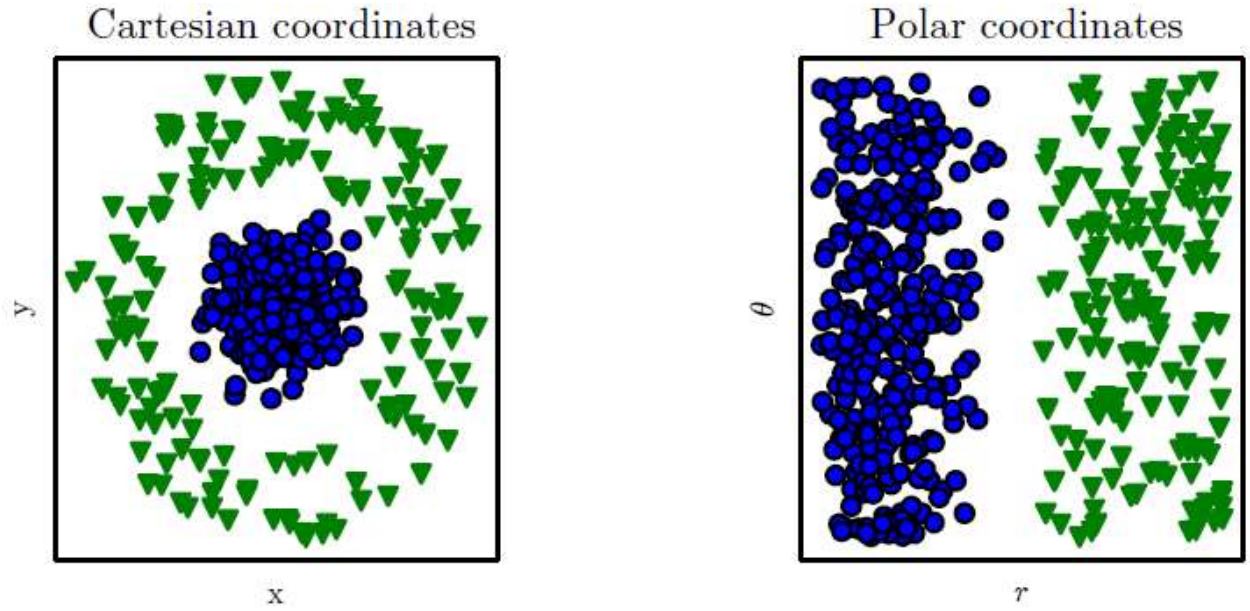
估计给定棋局的赢率

通过（深度）神经网络来实现



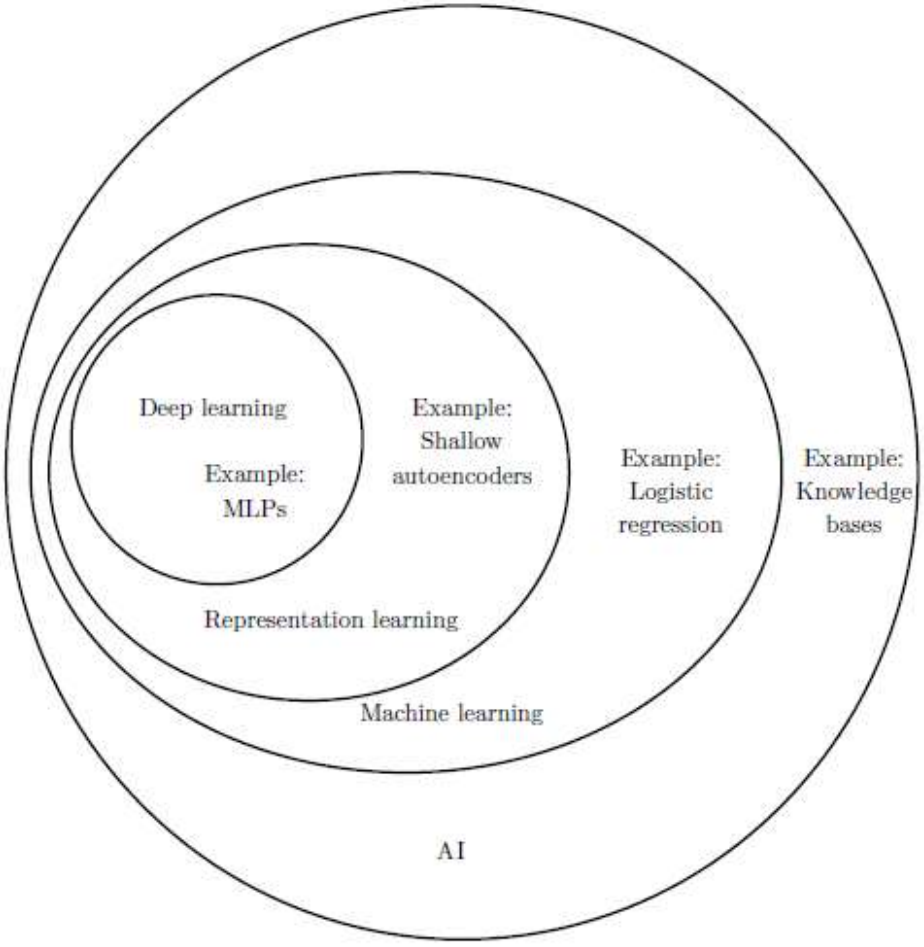
神经网络是什么?

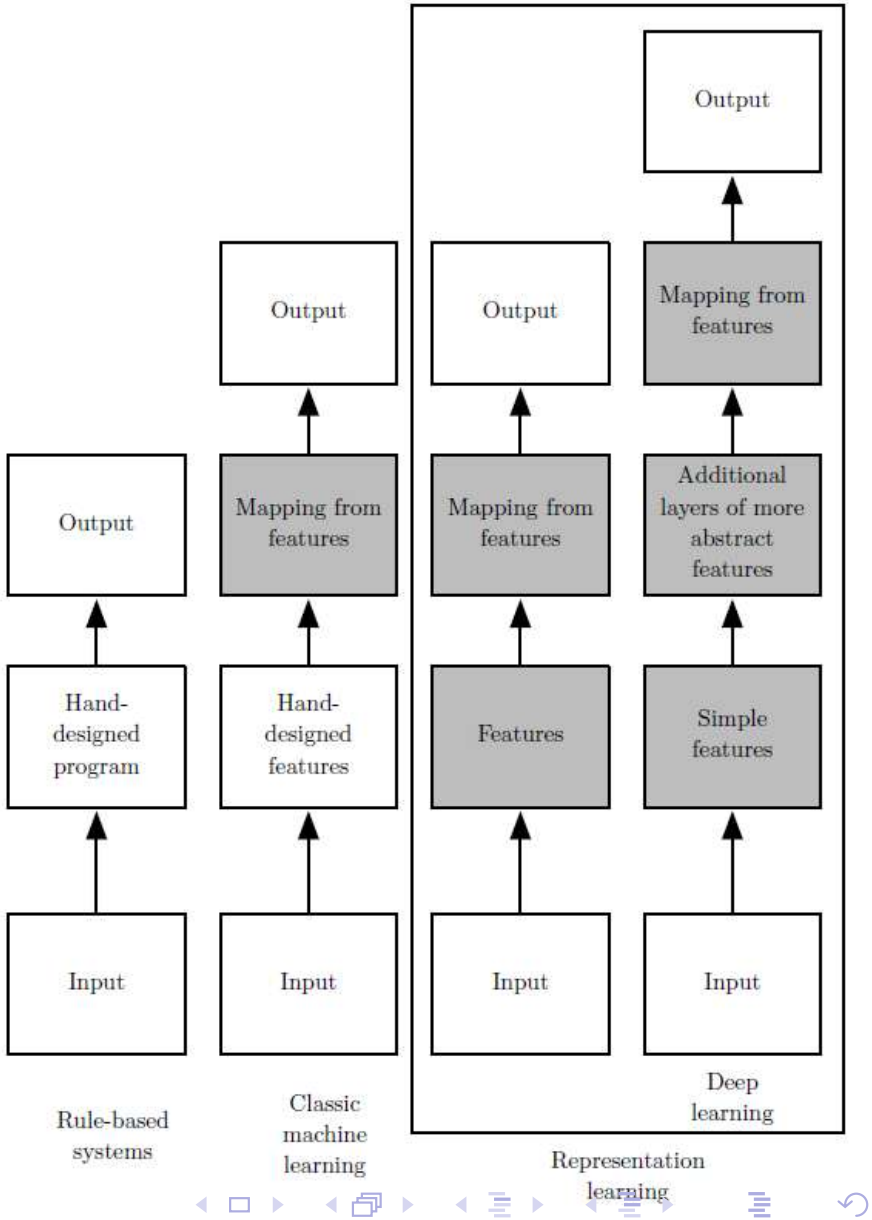
- 线性模型$y = \boldsymbol{w}^{\top}\boldsymbol{x}$：无法建模任何两个输入变量之间的相互作用
- 扩展线性模型来表示 $\boldsymbol{x}$ 的非线性函数，$y = \boldsymbol{\phi}(\boldsymbol{x})^{\top}\boldsymbol{w}$
- 简单机器学习方法常通过特征工程设计$\boldsymbol{\phi}(\boldsymbol{x})$



- （深度）神经网络直接学习特征表示，即 $y = f(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{w}) = \boldsymbol{\phi}(\boldsymbol{x};\boldsymbol{\theta})^{T}\boldsymbol{w}$

神经网络是什么?


Deep learning
Example: MLPs
Example: Shallow autoencoders
Representation learning
Example: Logistic regression
Machine learning
Example: Knowledge bases
AI


Output
Output
Output
Output
Mapping from features
Mapping from features
Mapping from features
Additional layers of more abstract features
Hand-designed program
Hand-designed features
Features
Simple features
Input
Input
Input
Input
Rule-based systems
Classic machine learning
Representation learning
Deep learning

神经网络发展史—第一阶段

- 1943年, McCulloch和Pitts 提出第一个神经元数学模型, 即M-P模型, 并从原理上证明了人工神经网络能够计算任何算数和逻辑函数
- 1949年, Hebb 发表《The Organization of Behavior》一书, 提出生物神经元学习的机理, 即Hebb学习规则
- 1958年, Rosenblatt 提出感知机网络（Perceptron）模型和其学习规则
- 1960年, Widrow和Hoff提出自适应线性神经元（Adaline）模型和最小均方学习算法
- 1969年, Minsky和Papert 发表《Perceptrons》一书, 指出单层神经网路不能解决非线性问题, 多层网络的训练算法尚无希望. 这个论断导致神经网络进入低谷

神经网络发展史—第二阶段

- 1982年, 物理学家Hopfield提出了一种具有联想记忆、优化计算能力的递归网络模型, 即Hopfield 网络
- 1986年, Rumelhart 等编辑的著作《Parallel Distributed Proceesing: Explorations in the Microstructures of Cognition》报告了反向传播算法
- 1987年, IEEE 在美国加州圣地亚哥召开第一届神经网络国际会议（ICNN）
- 90年代初, 伴随统计学习理论和SVM的兴起, 神经网络由于理论不够清楚, 试错性强, 难以训练, 再次进入低谷

神经网络发展史—第三阶段

- 2006年, Hinton提出了深度信念网络(DBN), 通过“预训练+微调”使得深度模型的最优化变得相对容易
- 2012年, Hinton 组参加ImageNet 竞赛, 使用 CNN 模型以超过第二名10个百分点的成绩夺得当年竞赛的冠军
- 伴随云计算、大数据时代的到来，计算能力的大幅提升，使得深度学习模型在计算机视觉、自然语言处理、语音识别等众多领域都取得了较大的成功

神经元模型

- 神经网络的定义

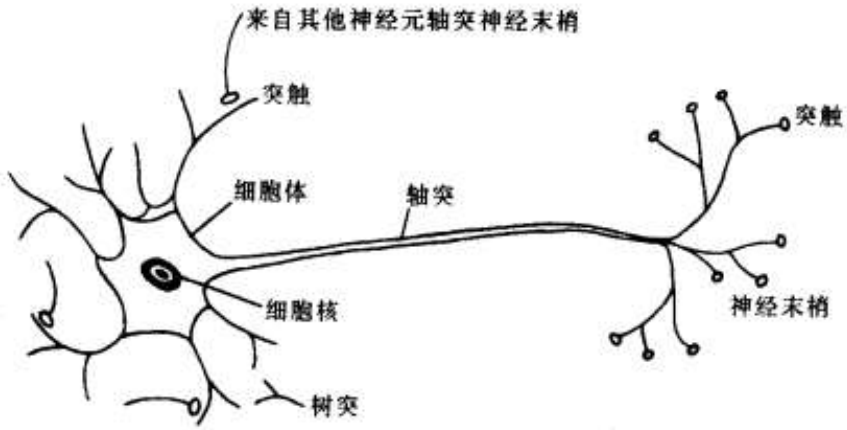
> “神经网络是由具有适应性的简单单元组成的广泛并行互联的网络, 它的组织能够模拟生物神经系统对真实世界物体所作出的反应”

[Kohonen, 1988]

- 机器学习中的神经网络通常是指“神经网络学习” 或者机器学习与神经网络两个学科的交叉部分
- 神经元模型即上述定义中的“简单单元”是神经网络的基本成分

生物神经网络

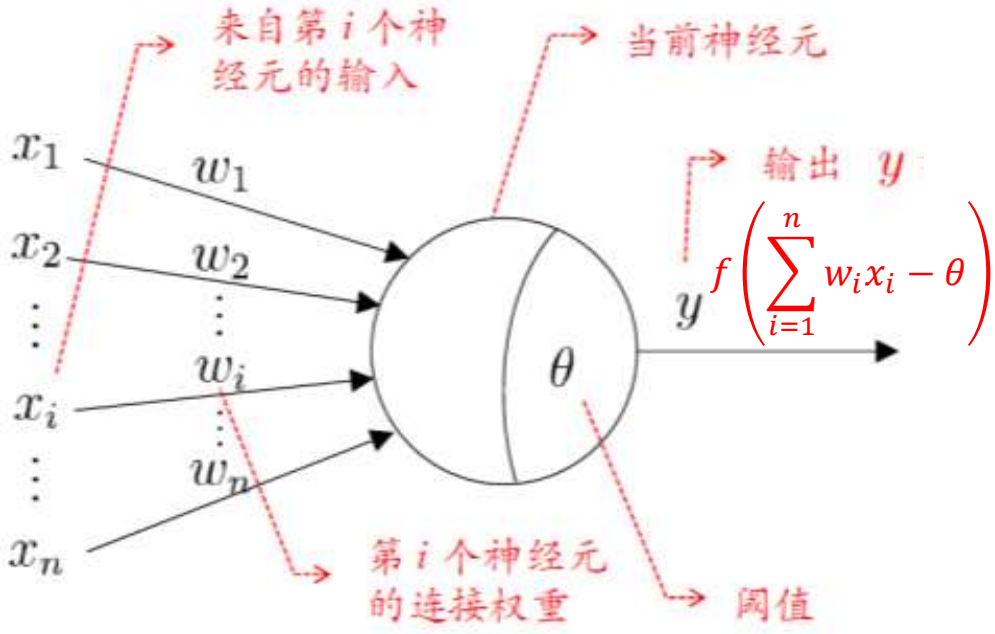
每个神经元与其他神经元相连, 当它“兴奋”时, 就会向相连的神经元发送化学物质, 从而改变这些神经元内的电位; 如果某神经元的电位超过一个“阈值”, 那么它就会被激活, 即“兴奋”起来, 向其它神经元发送化学物质



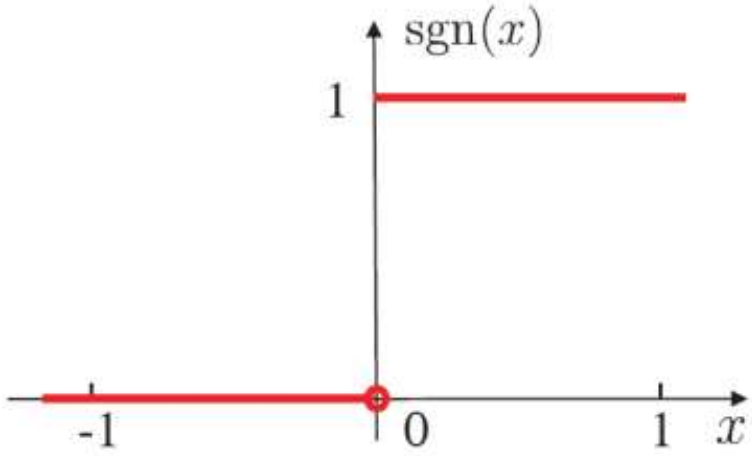
神经元模型

M-P 神经元模型 [McCulloch and Pitts, 1943]

- 输入：来自其他n个神经云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

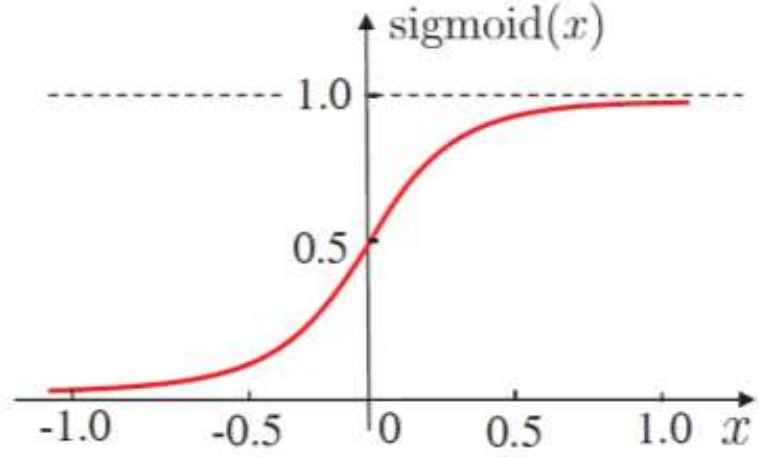


神经元模型—激活函数



$$\text{sgn}(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{if } x < 0. \end{cases}$$

(a) 阶跃函数



$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$$

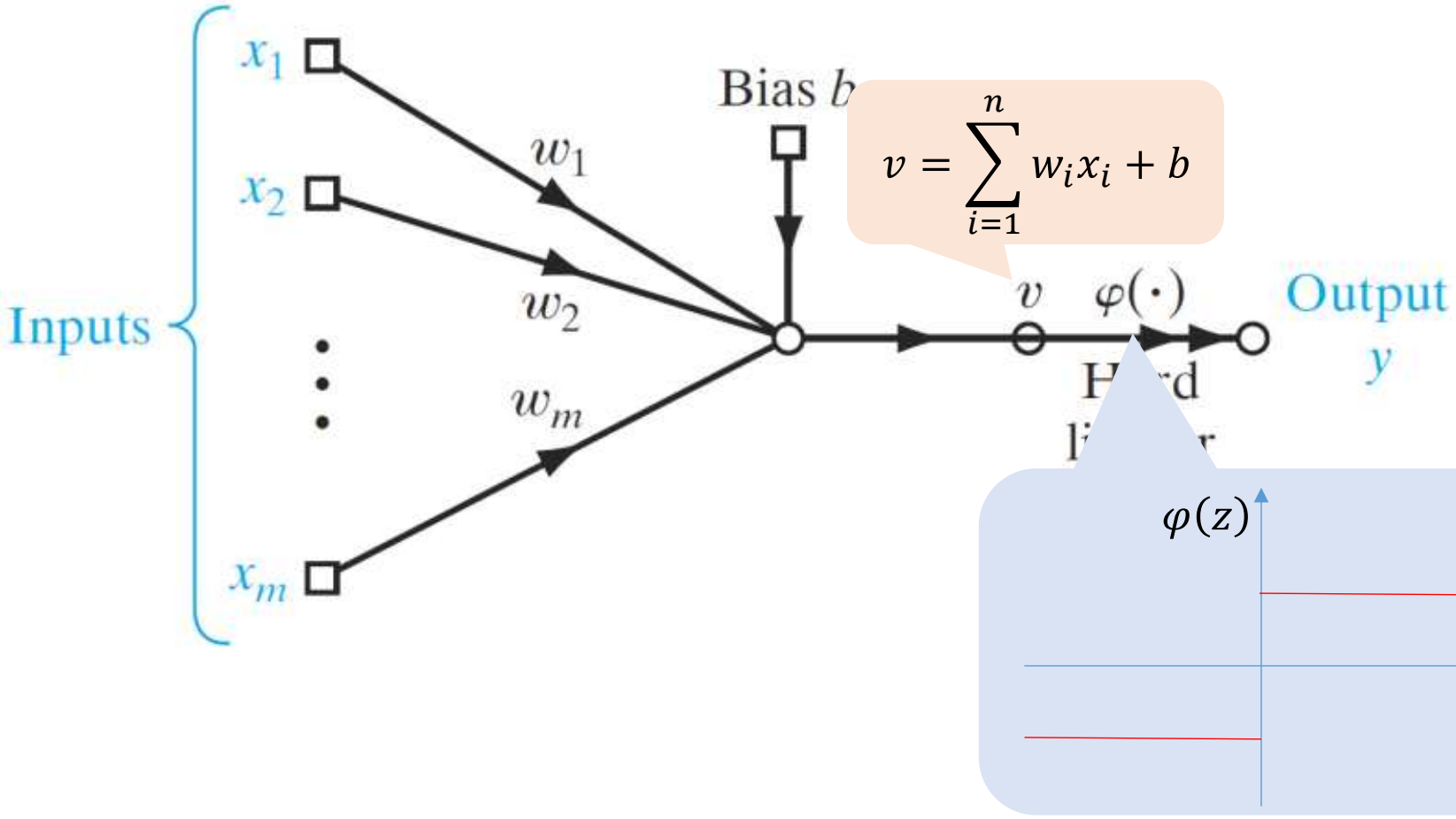
(b) Sigmoid 函数

理想激活函数是阶跃函数
0表示抑制神经元；1表示激活神经元

阶跃函数具有不连续、不光滑等不好的性质, 常用的是 Sigmoid 函数

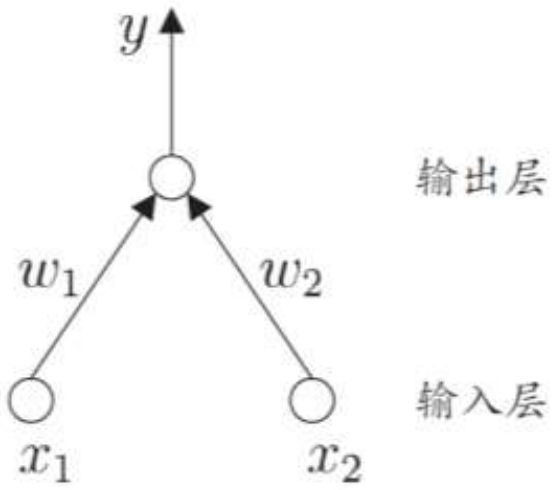
感知机

- 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）



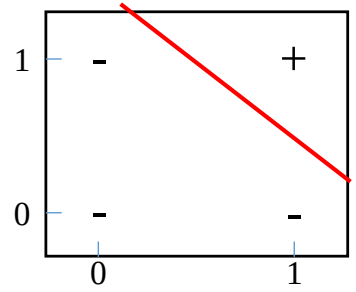
感知机

- 感知机能够容易地实现逻辑与、或、非运算



$x_1 \wedge x_2$：

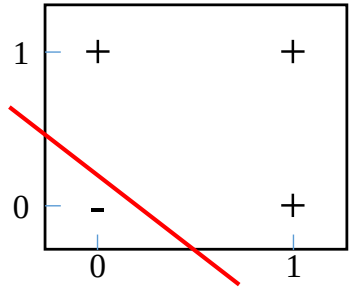
$y = f(1 \cdot x_1 + 1 \cdot x_2 - 1.5)$



与

$x_1 \vee x_2$：

$y = f(1 \cdot x_1 + 1 \cdot x_2 - 0.5)$



或

感知机学习

- 给定训练集, 权重$w_i(i = 1,2,\cdots,n)$与阈值θ可以通过学习得到

- 感知机学习规则

对训练样例$(\boldsymbol{x}, y)$, 若当前感知机的输出为$\hat{y}$, 则感知机权重调整规则为:

$$w_i \leftarrow w_i + \eta(y - \hat{y})x_i$$

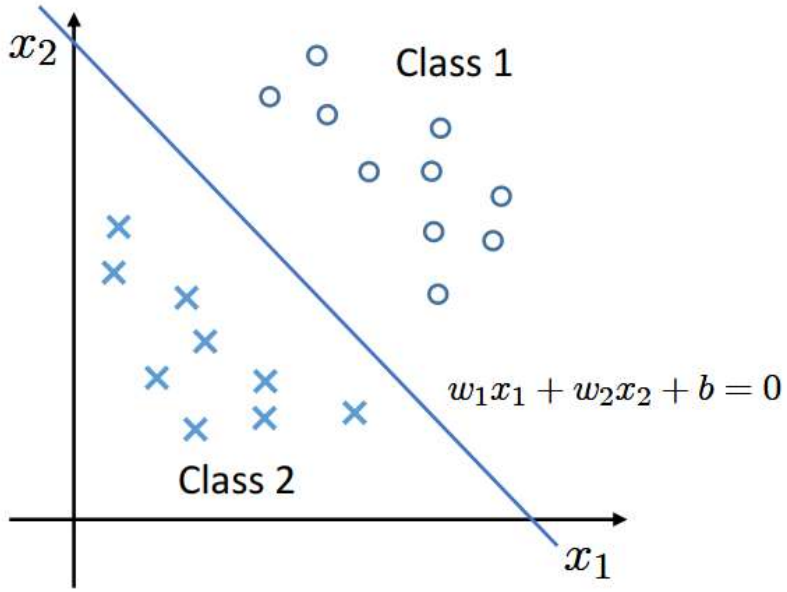
学习率

对应如下更新规则

若对训练样本$(\boldsymbol{x}, y)$预测正确，则感知机不发生变化;
若预测值更大，降低激活输入的权重;
若预测值更小，增加激活输入的权重

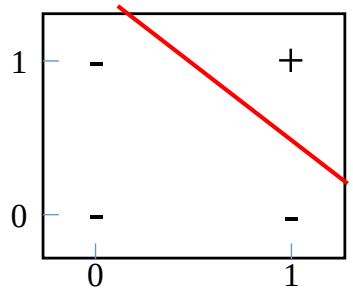
感知机学习

- 若两类模式线性可分, 则感知机的学习过程一定会收敛；否感知机的学习过程将会发生震荡 [Minsky and Papert, 1969]
- 单层感知机的学习能力非常有限, 只能解决线性可分问题

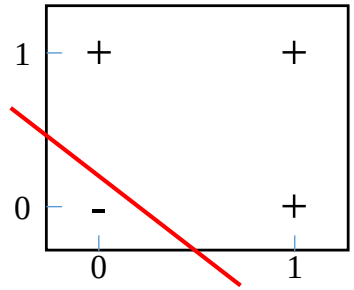


感知机学习

- 与、或、非问题是线性可分的, 因此感知机学习过程能够求得适当的权值向量

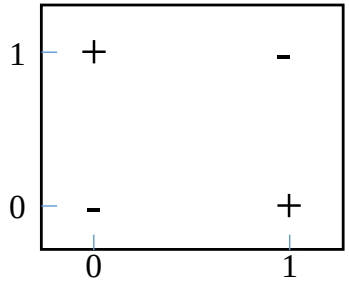


与



或

- 异或问题不是线性可分的, 感知机学习不能求得合适解



异或 $x_1 \oplus x_2$

对于非线性可分问题, 如何求解? 多层感知机

多层感知机

- 输出层与输入层之间的一层神经元, 被称之为隐层或隐含层, 隐含层和输出层神经元都是具有激活函数的功能神经元

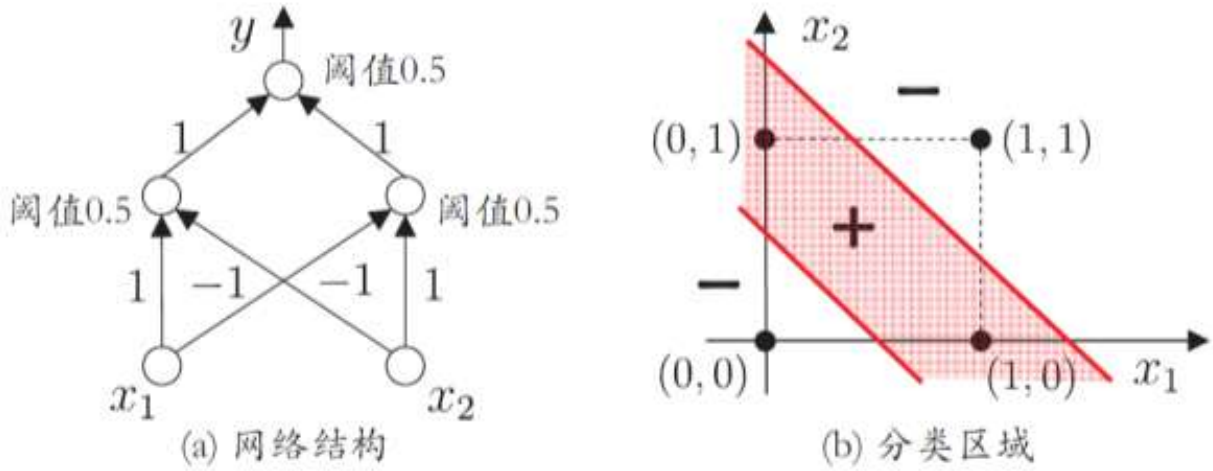
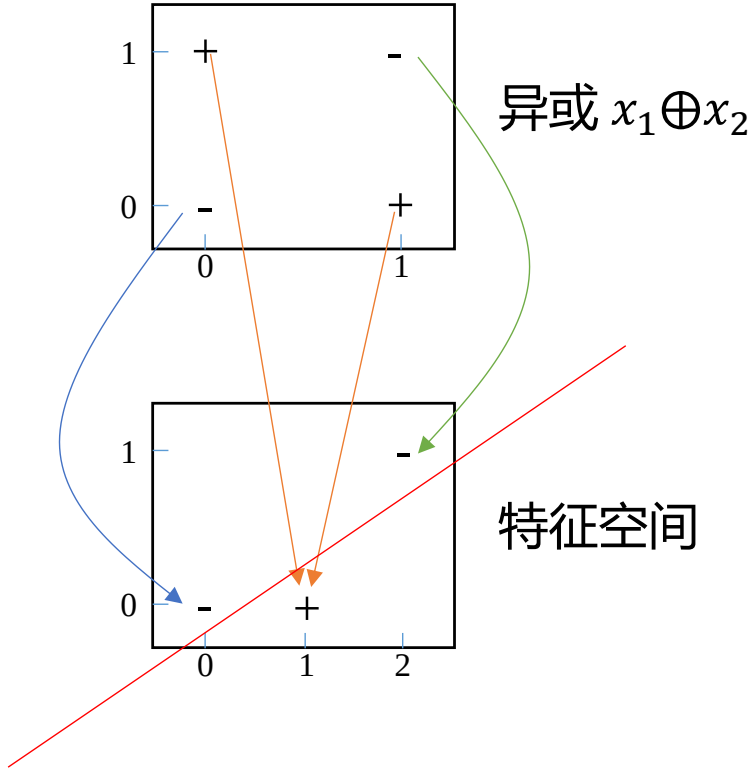


图 5.5 能解决异或问题的两层感知机

多层感知机

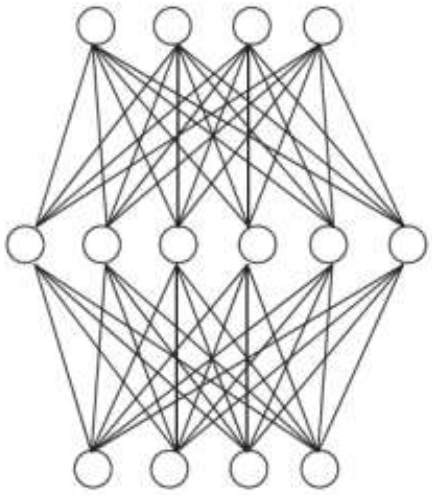
- $\boldsymbol{h} = max\left(0, \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ -1 \end{bmatrix}\right)$

- $y = [1, -2]\,\boldsymbol{h} - 0.5$

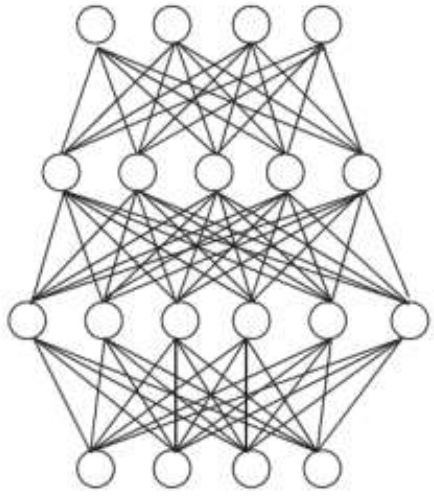


多层前馈神经网络

- 定义：每层神经元与下一层神经元全互联，神经元之间不存在同层连接也不存在跨层连接
- 前馈：输入层接受外界输入，隐含层与输出层神经元对信号进行加工，最终结果由输出层神经元输出
- 学习：根据训练数据来调整神经元之间的“连接权”以及每个功能神经元的“阈值”
- 多层网络：包含隐层的网络



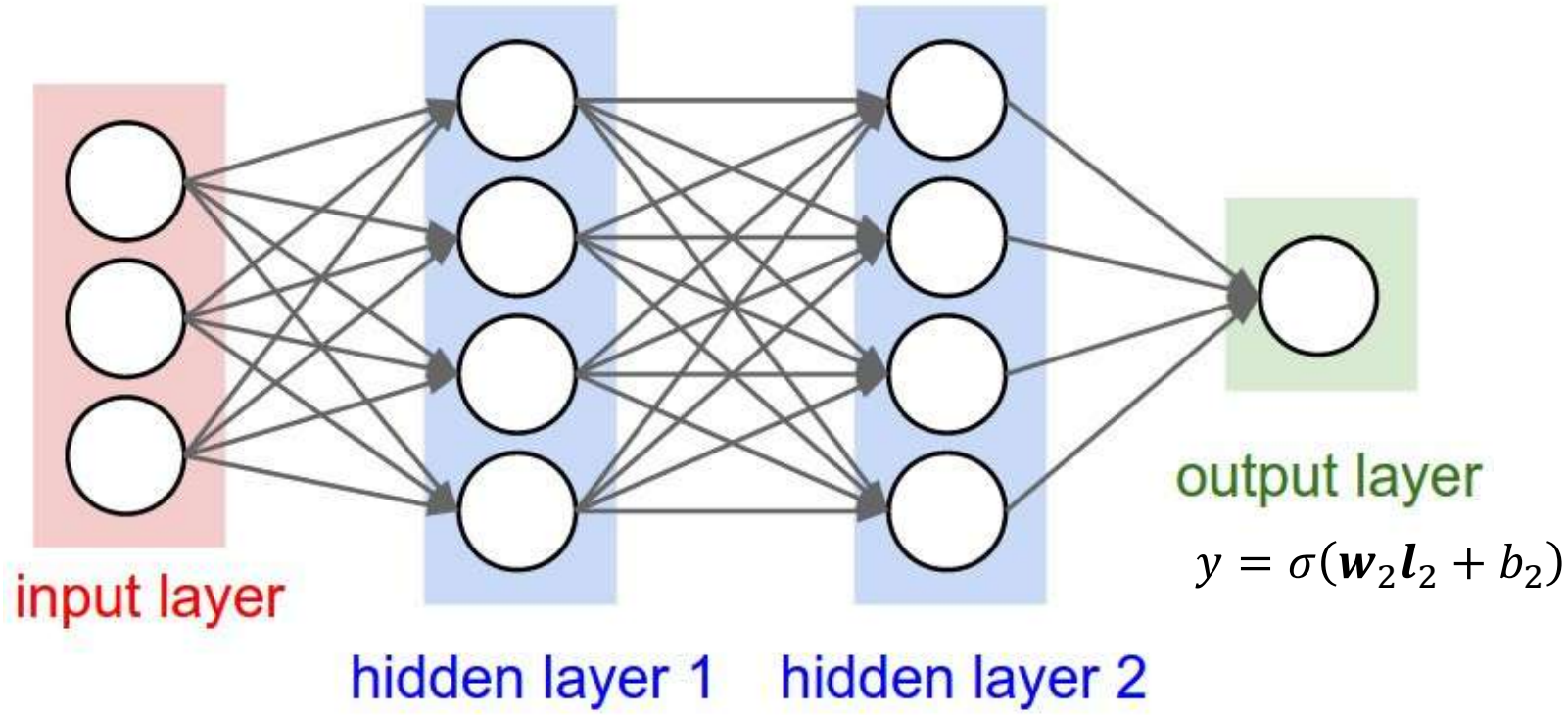
(a) 单隐层前馈网络



(b) 双隐层前馈网络

多层前馈神经网络—表示能力

- 只需要一个包含足够多神经元且具有任何 “挤压” 性质的激活函数的隐层, 多层前馈神经网络就能以任意精度逼近有界闭集上的任意连续函数



多层前馈神经网络

- 如何学习多层前馈神经网络的参数呢?

误差逆传播算法

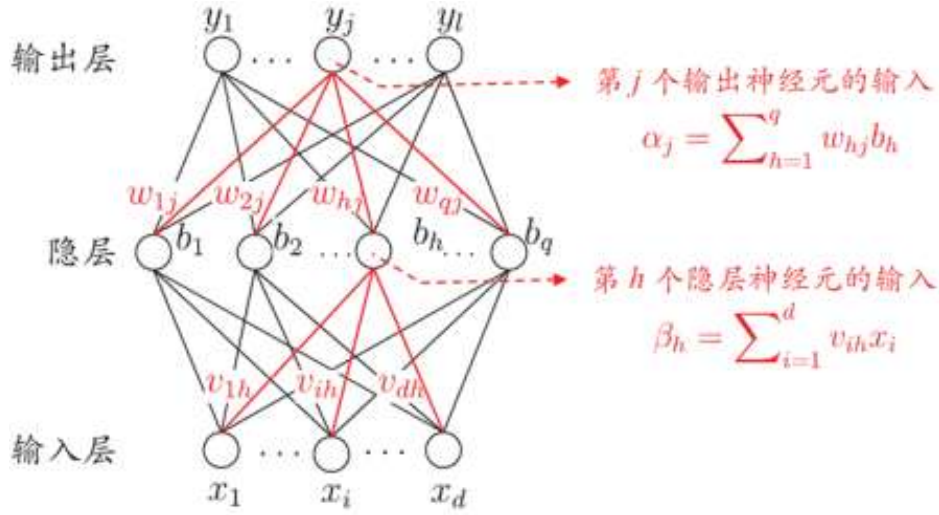
误差逆传播算法（Error BackPropagation, 简称BP）是最成功的训练多层前馈神经网络的学习算法

误差逆传播算法

输出l维实值向量$\boldsymbol{y}$

θ_j： 输出层第j个神经元阈值
w_{hj}：隐层与输出层神经元之间的连接权重

γ_h： 隐含层第h个神经元阈值
v_{ih}：输入层与隐层神经元之间的连接权重



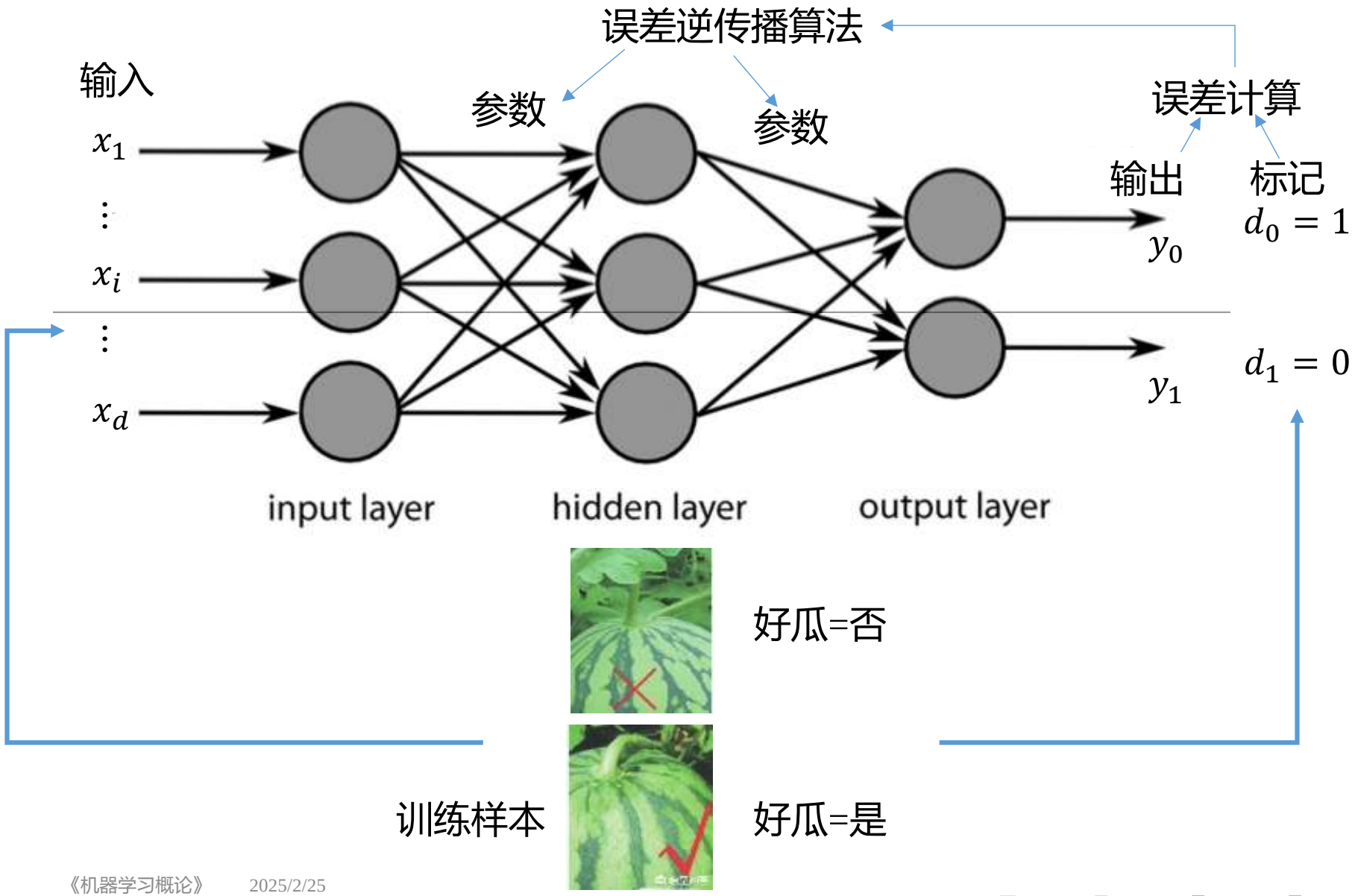
网络中需要 $(d+l+1)q+l$ 个参数需要优化

输入示例$\boldsymbol{x}$由d个属性描述

BP是一个迭代学习算法, 在迭代的每一轮中采用广义的感知机学习规则对参数进行更新估计, 任意的参数v的更新估计式为

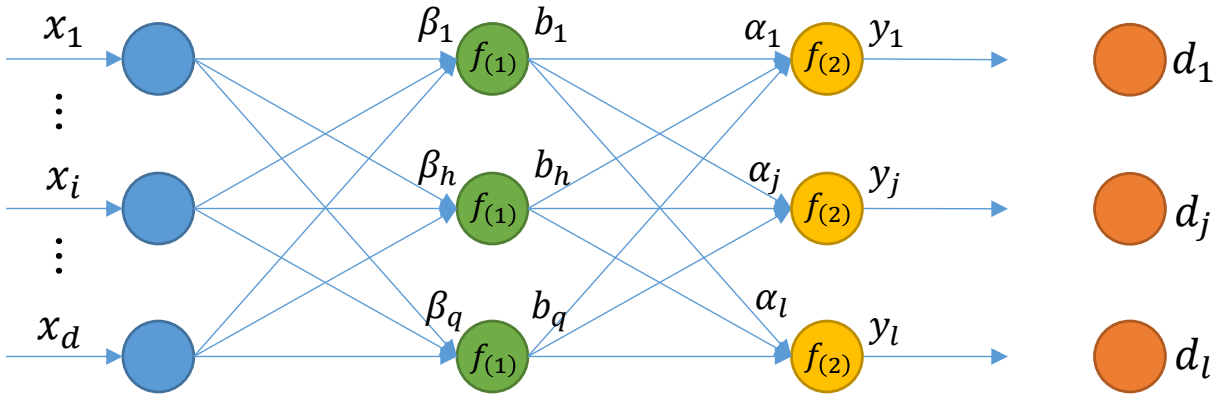
$$v \leftarrow v + \Delta v$$

误差逆传播算法


误差逆传播算法
输入
参数
参数
误差计算
输出
标记
$d_0 = 1$
x_1
x_i
x_d
y_0
y_1
$d_1 = 0$
input layer
hidden layer
output layer
好瓜=否
训练样本
好瓜=是

误差逆传播算法—前向

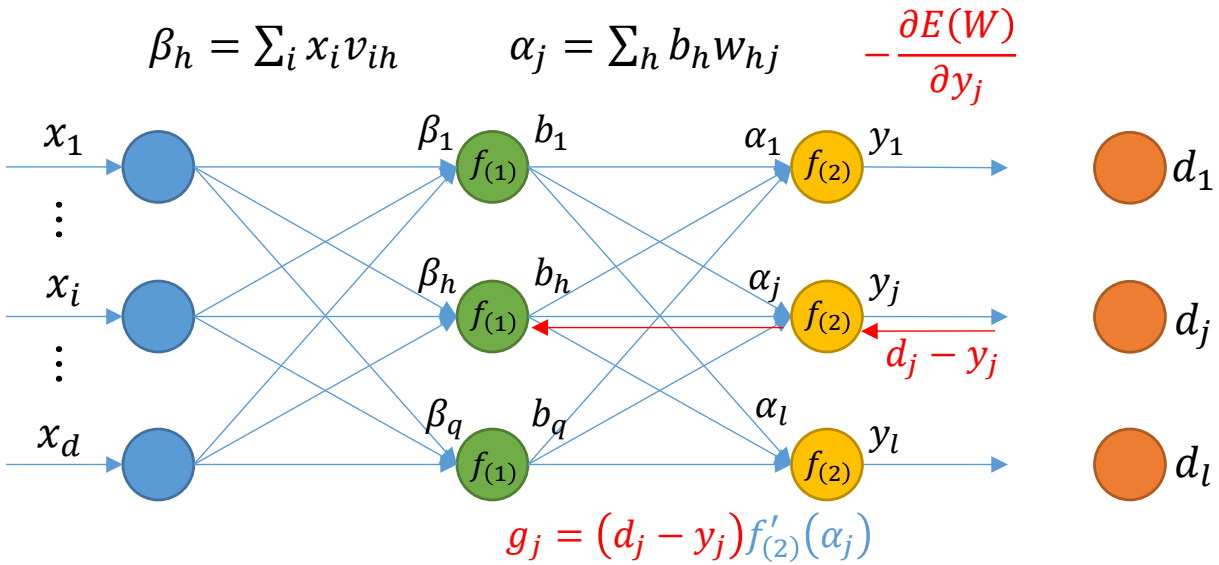
$$\beta_h = \sum_i x_i v_{ih} \qquad \alpha_j = \sum_h b_h w_{hj}$$



前向预测

$$\boldsymbol{x} \xrightarrow[f^{(1)}(\beta_h)]{\beta_h = \sum_i x_i v_{ih}} b_h \xrightarrow[f^{(2)}(\alpha_j)]{\alpha_j = \sum_h b_h w_{hj}} y_j$$

误差逆传播算法—后向

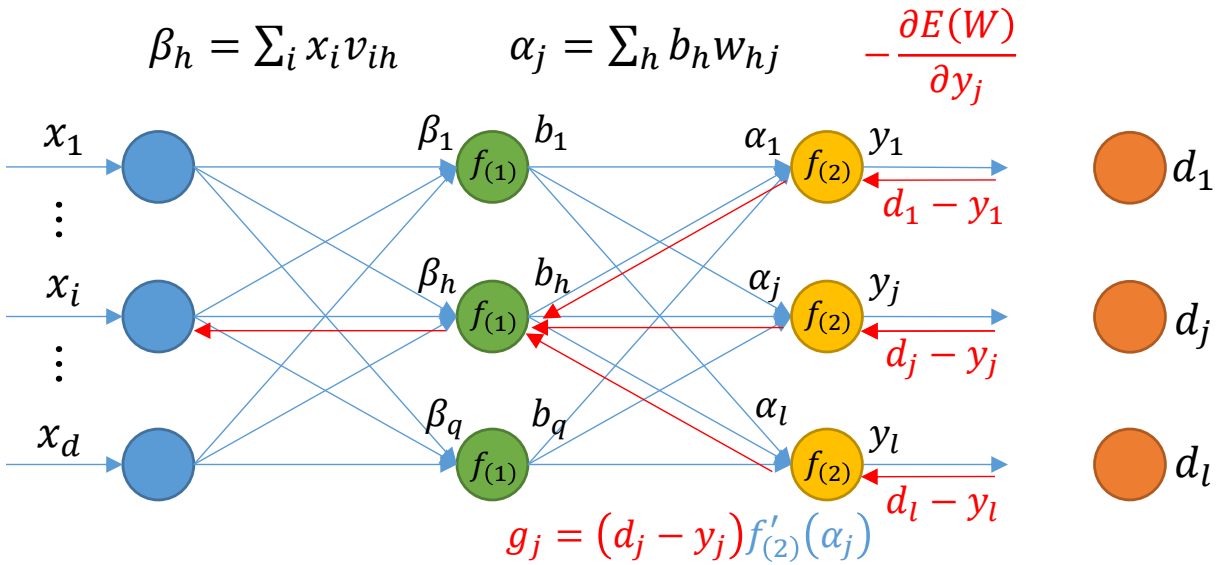


后向传播

$$w_{hj} = w_{hj} + \Delta w_{hj} \quad \Delta w_{hj} = \eta \text{Error}_j \text{Output}_h = \eta g_j b_h \quad E(\boldsymbol{W}) = \frac{1}{2}\sum_{j=1}^{l}(y_j - d_j)^2$$

$$\Delta w_{hj} = -\eta \frac{\partial E(W)}{\partial w_{hj}} = -\eta \frac{\partial E(W)}{\partial y_j}\frac{\partial y_j}{\partial \alpha_j}\frac{\partial \alpha_j}{\partial w_{hj}} = \eta (d_j - y_j) f'_{(2)}(\alpha_j) b_h = \eta g_j b_h$$

误差逆传播算法—后向



后向传播

$$v_{ih} = v_{ih} + \Delta v_{ih} \qquad \Delta v_{ih} = \eta \text{Error}_h \text{Output}_i = \eta e_h x_i \qquad E(\boldsymbol{W}) = \frac{1}{2}\sum_{j=1}^{l}(y_j - d_j)^2$$

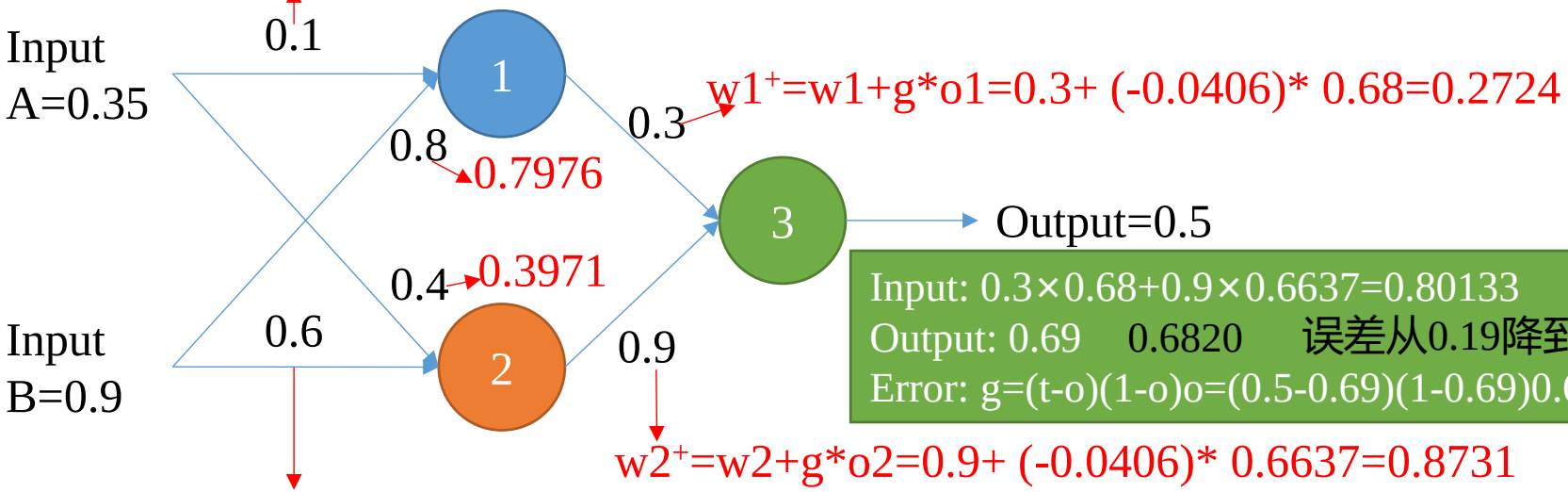
$$\Delta v_{ih} = -\eta \frac{\partial E(W)}{\partial v_{ih}} = -\eta \frac{\partial E(W)}{\partial b_h}\frac{\partial b_h}{\partial \beta_h}\frac{\partial \beta_h}{\partial v_{ih}} = \eta \sum_j g_j w_{hj} f'_{(1)}(\beta_h) x_i = \eta e_h x_i$$

BP算法：简单例子

- 考虑如下简单网络　假设激活函数为Sigmoid函数

Input: 0.35×0.1+0.9×0.8=0.755　0.7525
Output: 0.68　0.6797
Error: e1=g*w1*o*(1-o)=-0.0406*0.3*0.68*(1-0.68)=-2.650*10^{-3}

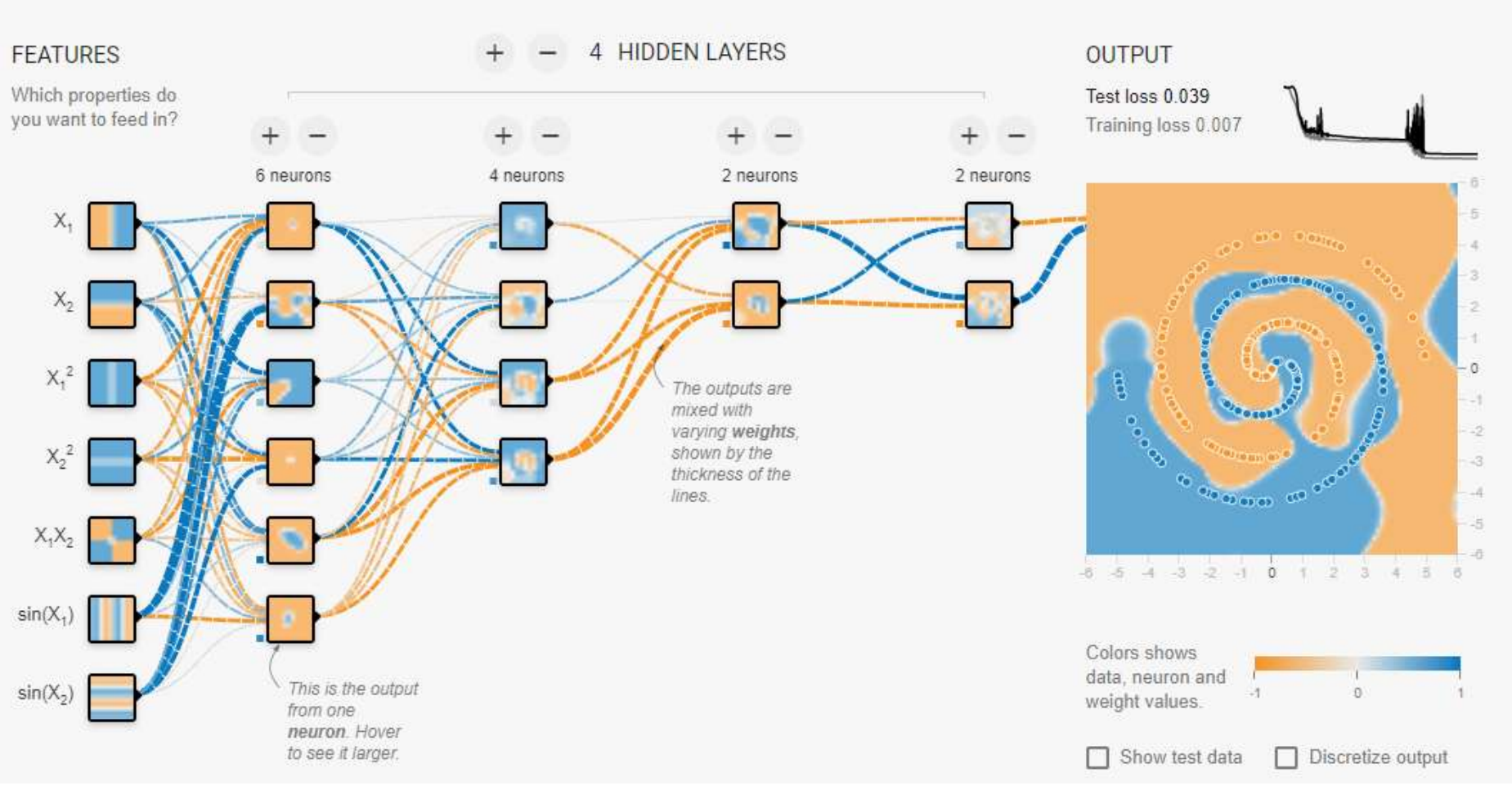
$w3^{+}=w3+e1*A=0.1+ (-2.650*10^{-3})*0.35=0.0991$



Input: 0.3×0.68+0.9×0.6637=0.80133　0.7631
Output: 0.69　0.6820　误差从0.19降到0.1820
Error: g=(t-o)(1-o)o=(0.5-0.69)(1-0.69)0.69=-0.0406

Input: 0.35×0.4+0.9×0.6=0.68　0.6724
Output: 0.6637　0.6620
Error: e2=g*w2*o*(1-o)=-0.0406*0.9*0.6637*(1-0.6637)=-8.156*10^{-3}

BP演示



http://playground.tensorflow.org/

误差逆传播算法

输入: 训练集 $D = \{(\boldsymbol{x}_k, \boldsymbol{y}_k)\}_{k=1}^{m}$;
学习率 η.

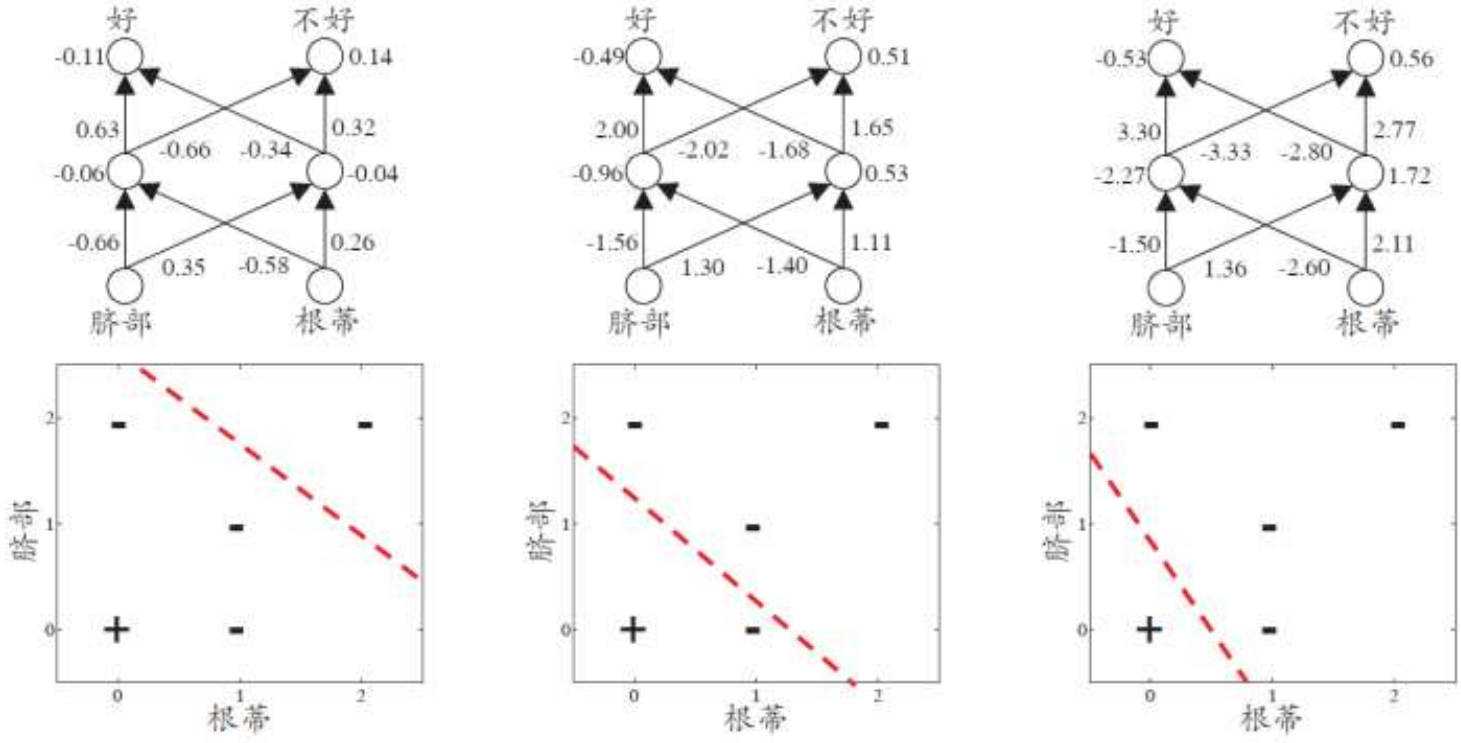
过程:

1: 在$(0,1)$范围内随机初始化网络中所有连接权和阈值
2: **repeat**
3: **for all** $(\boldsymbol{x}_k, \boldsymbol{y}_k) \in D$ **do**
4: 根据当前参数和式(5.3) 计算当前样本的输出 $\hat{\boldsymbol{y}}_k$;
5: 根据式(5.10) 计算输出层神经元的梯度项 g_j;
6: 根据式(5.15) 计算隐层神经元的梯度项 e_h;
7: 根据式(5.11)-(5.14) 更新连接权 w_{hj}, v_{ih} 与阈值 θ_j, γ_h
8: **end for**
9: **until** 达到停止条件

输出: 连接权与阈值确定的多层前馈神经网络

图 5.8 误差逆传播算法

误差逆传播算法



(a) 第 25 轮　(b) 第 50 轮　(c) 第 100 轮

图 5.9　在 2 个属性、5 个样本的西瓜数据上, BP网络参数更新和分类边界的变化情况

误差逆传播算法

- 标准 BP 算法
  - 每次针对单个训练样例更新权值与阈值
  - 参数更新频繁, 不同样例可能抵消, 需要多次迭代.

也称为随机梯度下降

- 累计 BP 算法
  - 优化的目标是最小化整个训练集上的累计误差

$$E = \frac{1}{m}\sum_{k=1}^{m} E_k$$

  - 读取整个训练集一遍才对参数进行更新, 参数更新频率较低.

但在很多任务中，累计误差下降到一定程度后，进一步下降会非常缓慢，这时标准BP算法往往会获得较好的解, 尤其当训练集非常大时效果更明显.

误差逆传播算法

- 标准 BP 算法 Stochastic Gradient Descent
  - 每次针对单个训练样例更新权值与阈值
- 累计 BP 算法
  - 优化的目标是最小化整个训练集上的累计误差 $E = \frac{1}{m}\sum_{k=1}^{m} E_k$

- 小批量随机梯度下降法 Mini-Batch Stochastic Gradient Descent

$$\frac{\partial E}{\partial \boldsymbol{\theta}} = \frac{1}{m}\sum_{k=1}^{m}\frac{\partial E_k}{\partial \boldsymbol{\theta}} \approx \frac{1}{n}\sum_{i=1}^{n}\frac{\partial E_i}{\partial \boldsymbol{\theta}}$$

$n \ll m$，称为batch size

误差逆传播算法

- 多层前馈网络局限
  - 神经网络由于强大的表示能力, 经常遭遇过拟合

    表现为：训练误差持续降低, 但测试误差却可能上升

早停

- 在训练过程中, 若训练误差降低, 但验证误差升高, 则停止训练

正则化

- 在误差目标函数中增加一项描述网络复杂程度的部分, 例如连接权值与阈值的平方和

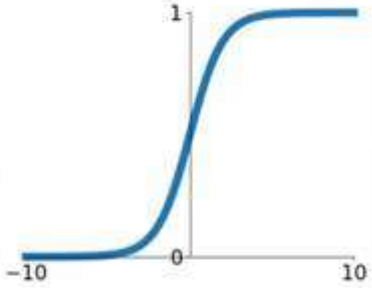
  - 如何设置隐层神经元的个数仍然是个未决问题

    实际应用中通常使用“试错法”调整

激活函数

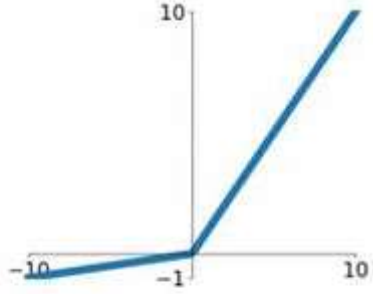
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



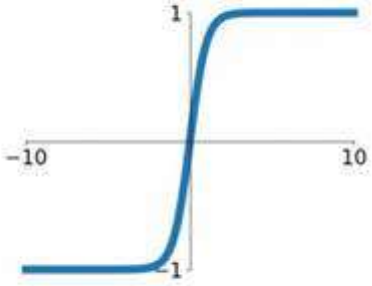
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

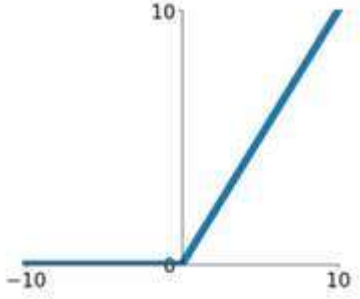


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

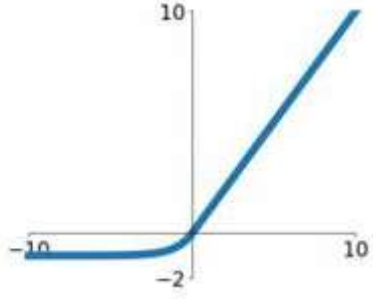
ReLU

$$\max(0, x)$$

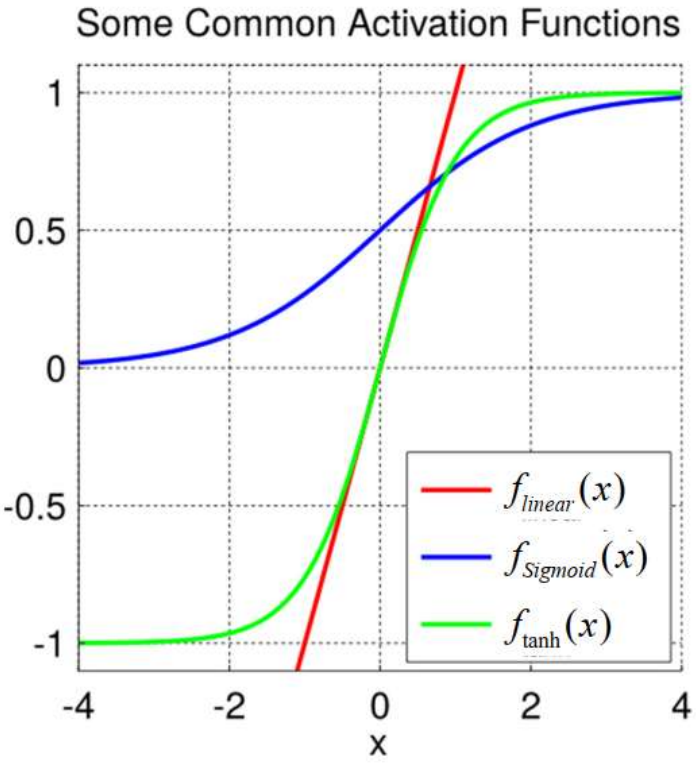


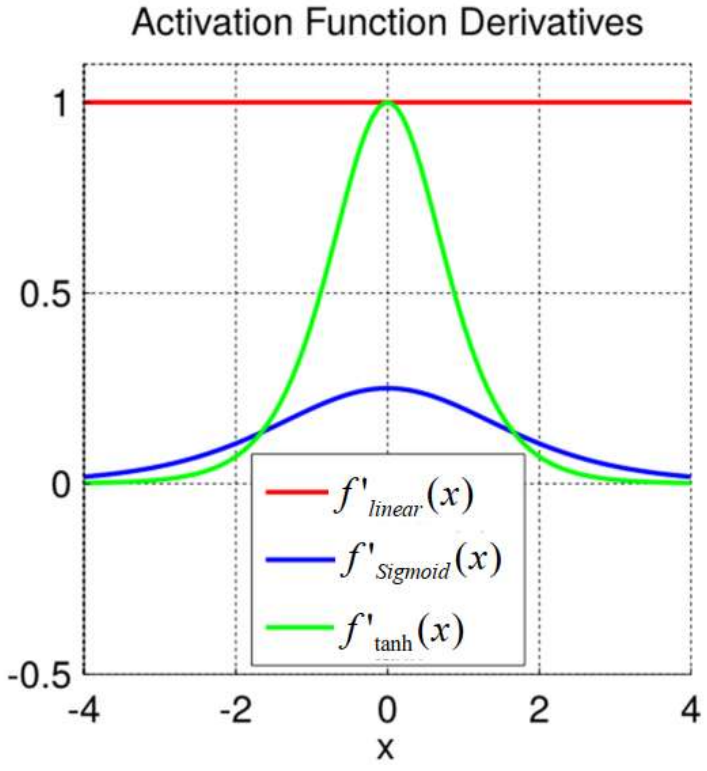
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



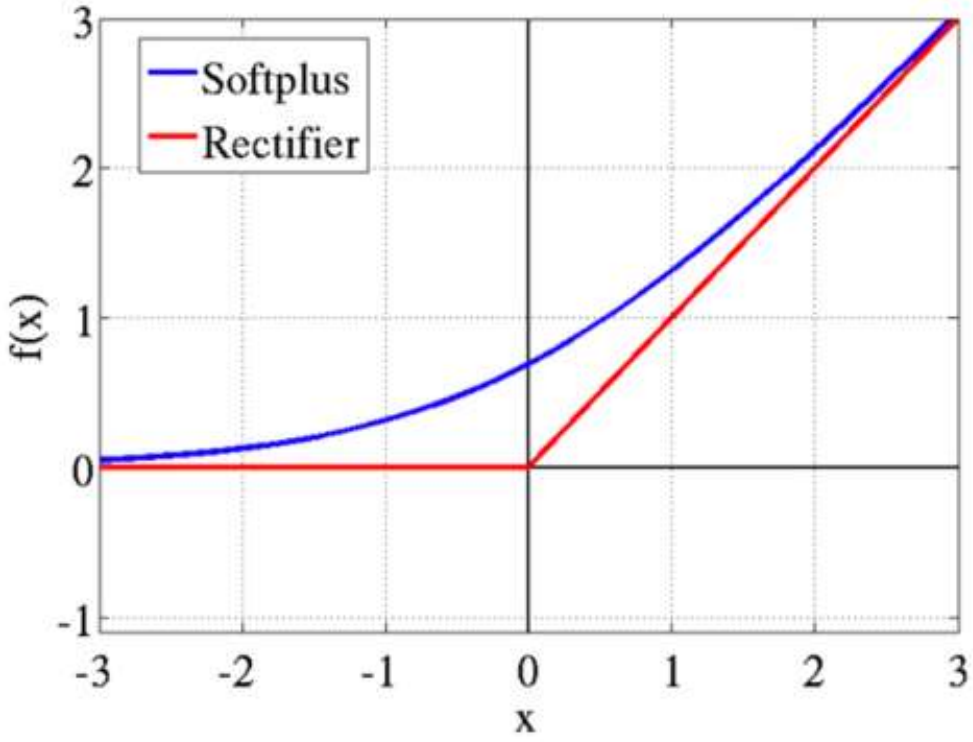
激活函数


Some Common Activation Functions
1
0.5
0
-0.5
-1
-4
-2
0
2
4
x
$f_{linear}(x)$
$f_{Sigmoid}(x)$
$f_{\tanh}(x)$


Activation Function Derivatives
1
0.5
0
-0.5
-4
-2
0
2
4
x
$f'_{linear}(x)$
$f'_{Sigmoid}(x)$
$f'_{\tanh}(x)$

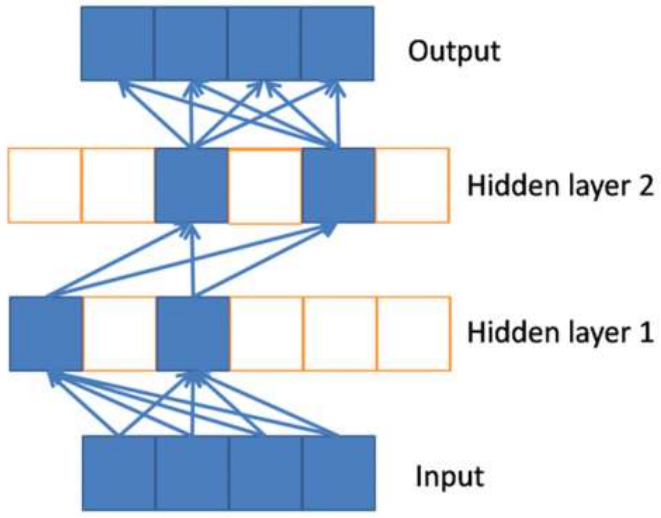
激活函数

- ReLU（整流线性单元） $f_{\mathrm{ReLU}}(x) = \max(0, x)$
- 与Softplus函数近似 $f_{\mathrm{Softplus}}(x) = \log(1 + e^x)$



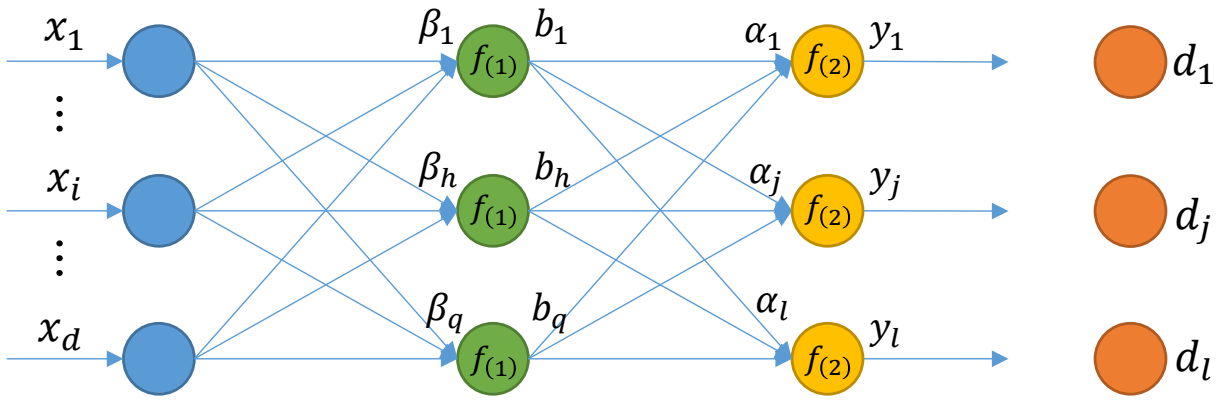
ReLU支持稀疏表示

只有一部分神经元被激活



损失函数

均方误差 $\frac{1}{2}\sum_j (y_j - d_j)^2$



多分类损失 Cross Entropy $-\sum_j d_j \log y_j$

one-hot向量

$$H(P,Q) = \sum_x P(x) \log Q(x)$$

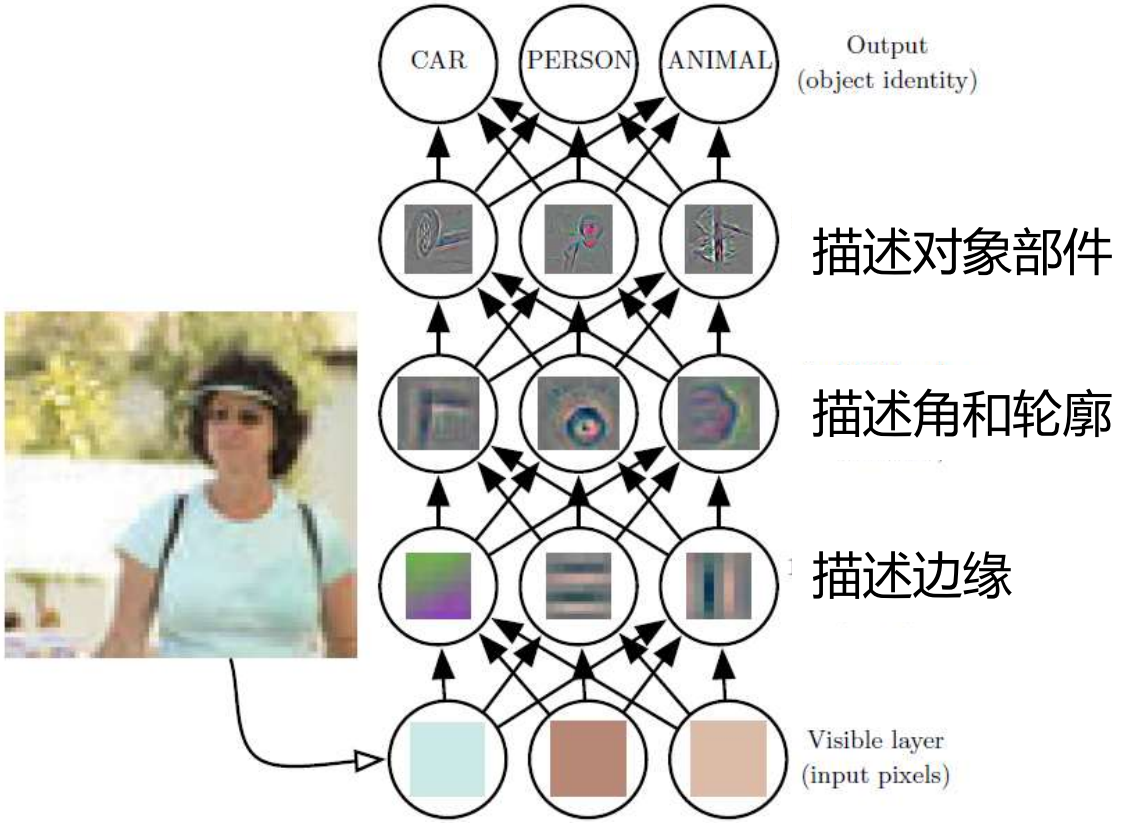
$$y_j = \frac{\exp(\alpha_j)}{\sum_{j'} \exp(\alpha_{j'})}$$

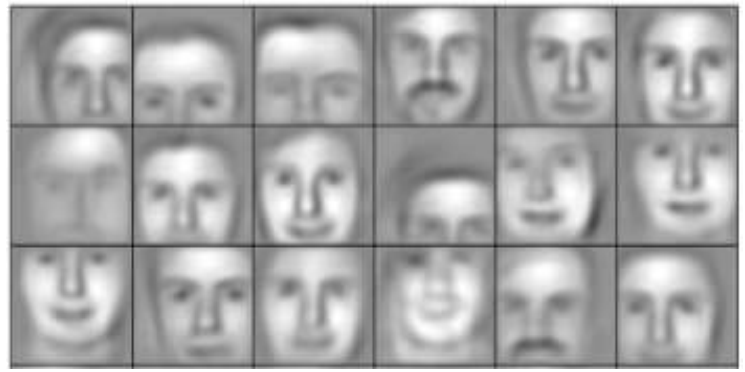
二分类损失 Cross Entropy $l = 1$ $-d \log y - (1-d) \log(1-y)$

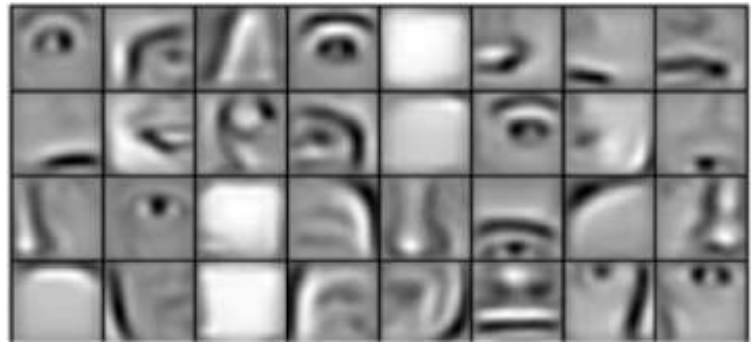
$$y = \frac{1}{1 + \exp(-\alpha)}$$

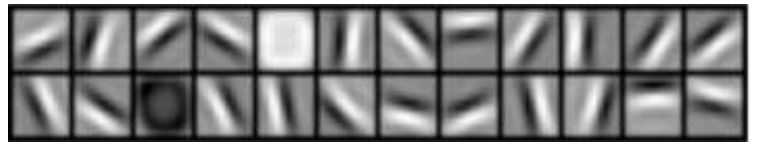
深度学习初探

- 深度学习将大千世界表示为嵌套的层次概念体系
  - 由较简单概念间的联系定义复杂概念
  - 从一般抽象概括到高级抽象表示

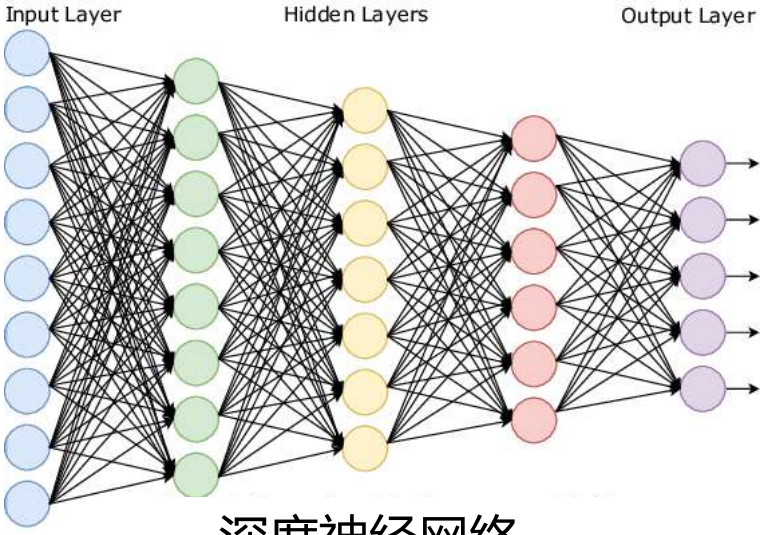




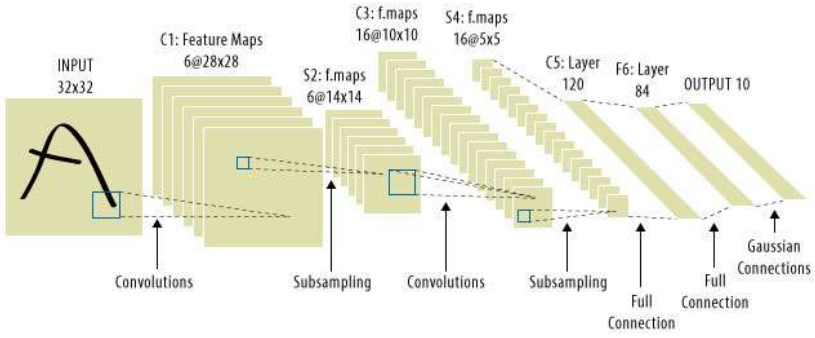




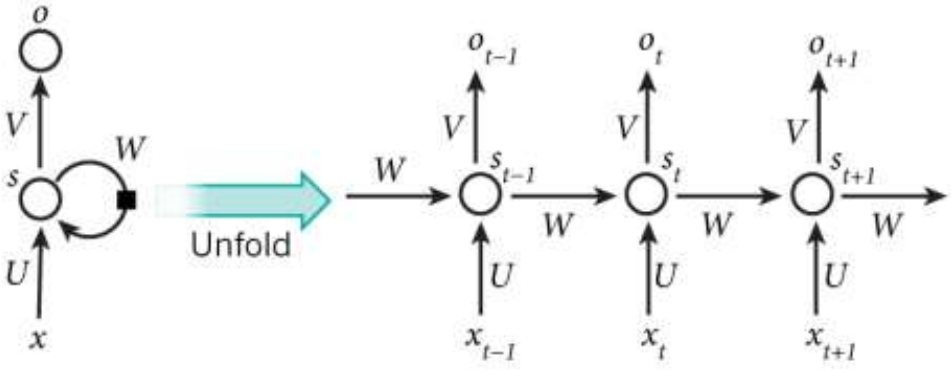
深度神经网络



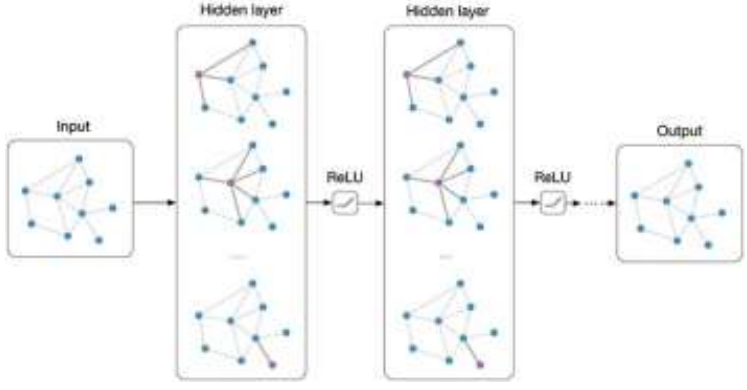
深度神经网络



深度卷积网络



循环神经网络



图卷积网络

深度卷积网络

- 一维卷积

$$s(t) = \int x(a)w(t-a)da$$

$$s(t) = \sum_a x(a)w(t-a)$$

输入

$$s(t) = (x * w)(t)$$

核函数

- 二维卷积

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, i-n)$$

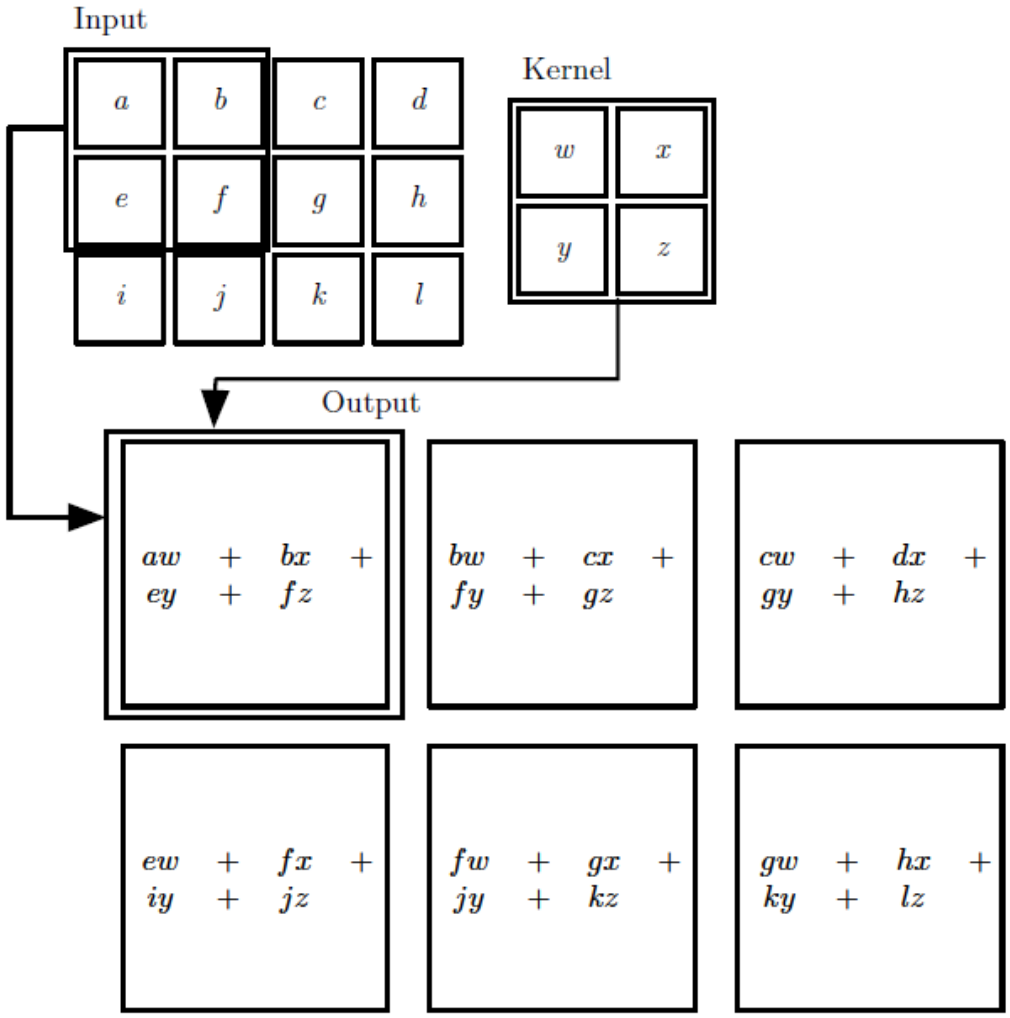
$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n)$$

- 互相关函数

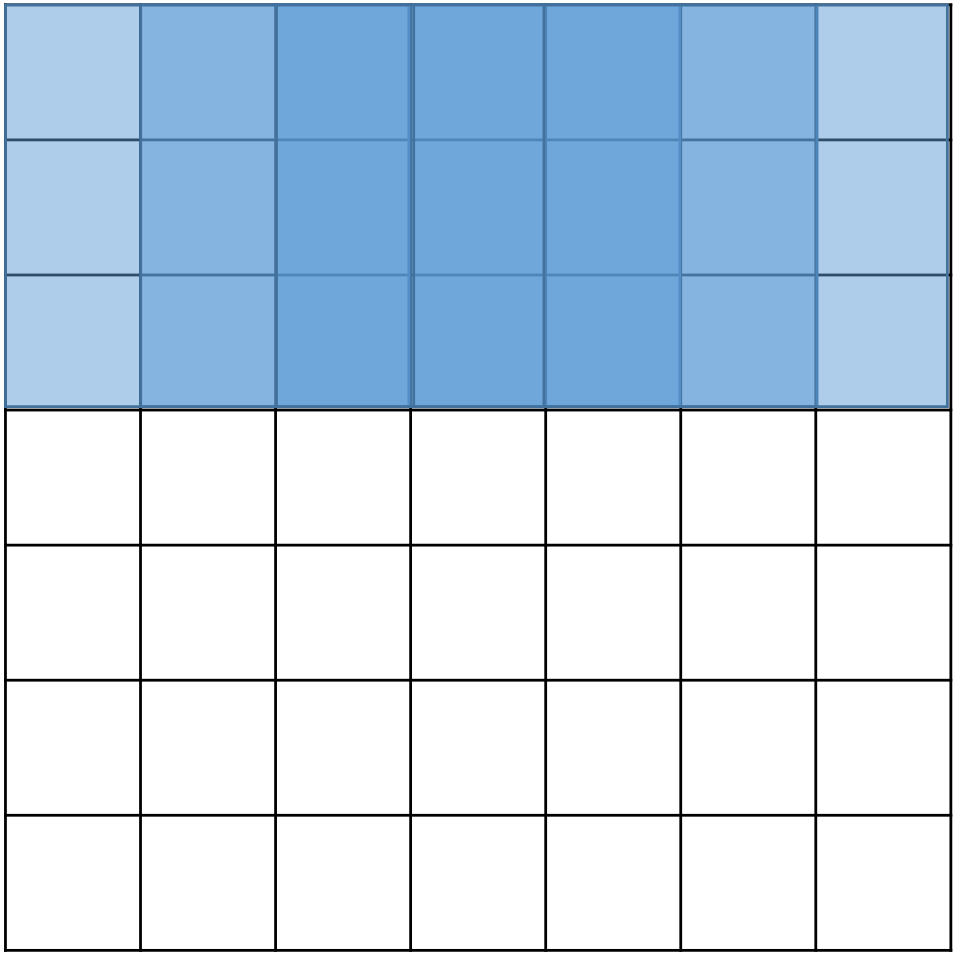
$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$$

卷积网络中的卷积

卷积的例子


Input
a b c d
e f g h
i j k l
Kernel
w x
y z
Output
aw + bx +
ey + fz
bw + cx +
fy + gz
cw + dx +
gy + hz
ew + fx +
iy + jz
fw + gx +
jy + kz
gw + hx +
ky + lz

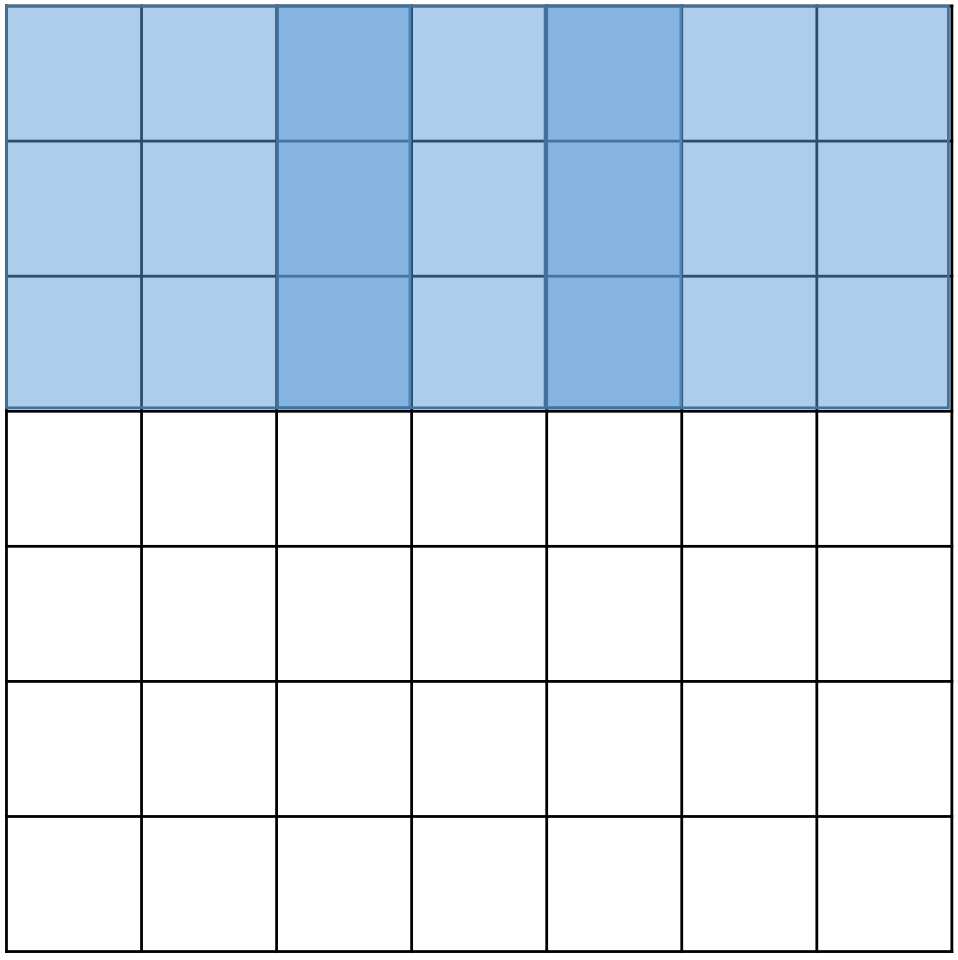
卷积的例子



7x7的输入
3x3的核

5x5的输出

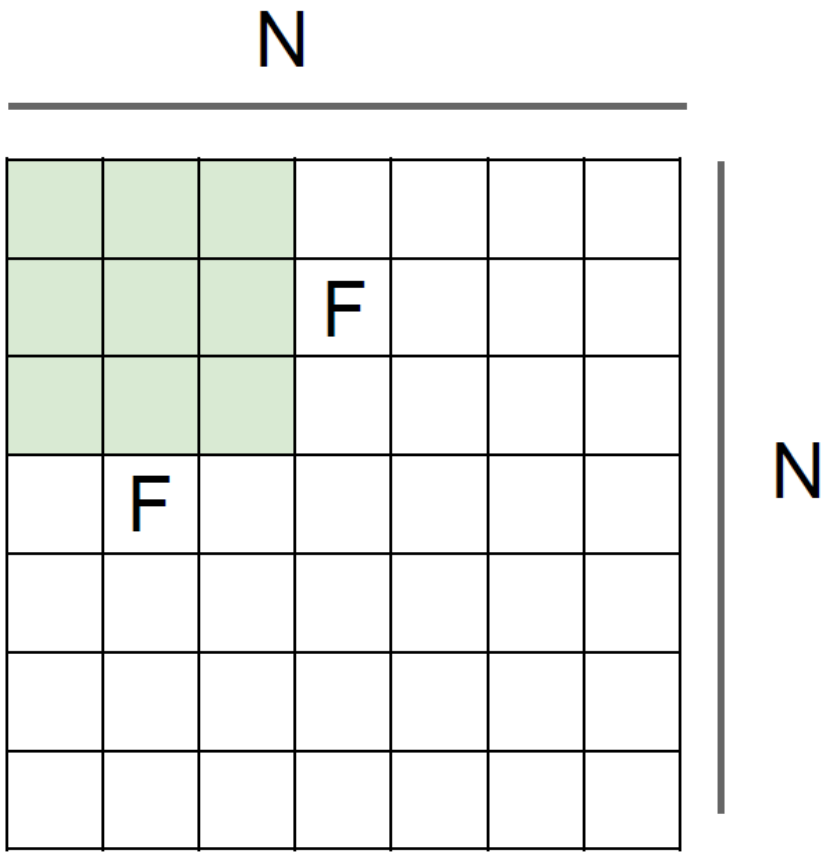
卷积的例子



7x7的输入
3x3的核
步幅为2

3x3的输出

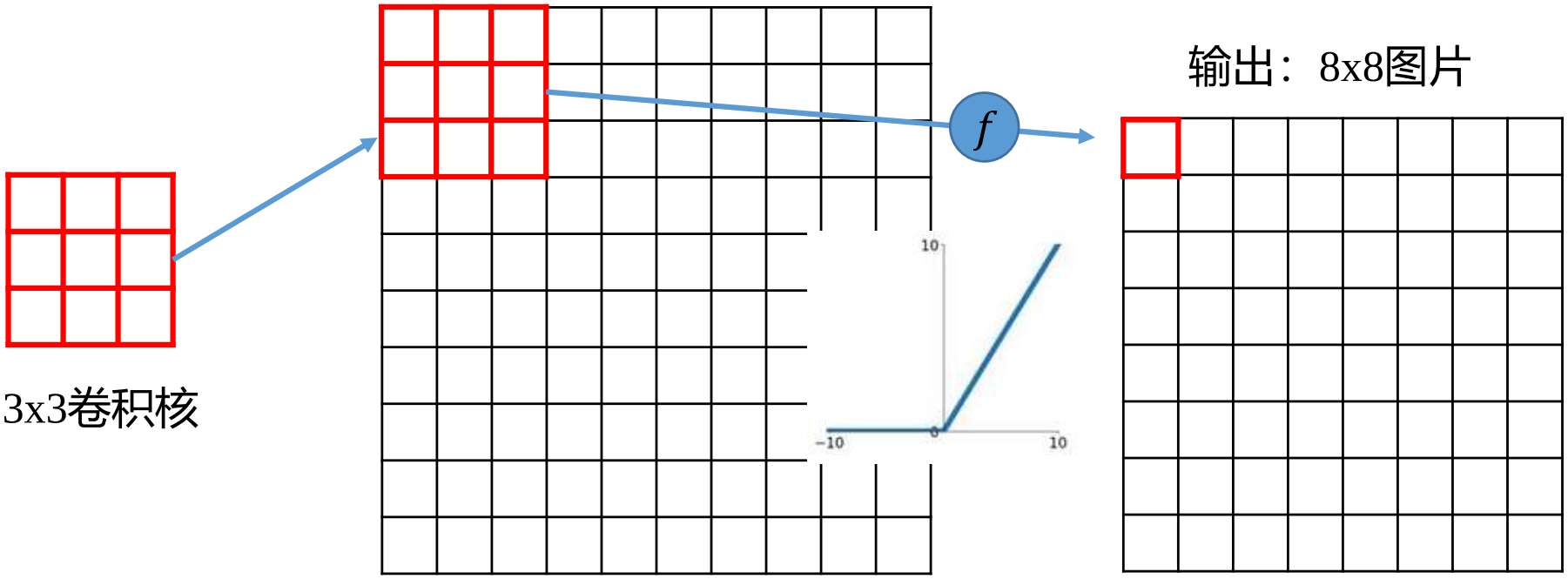
卷积

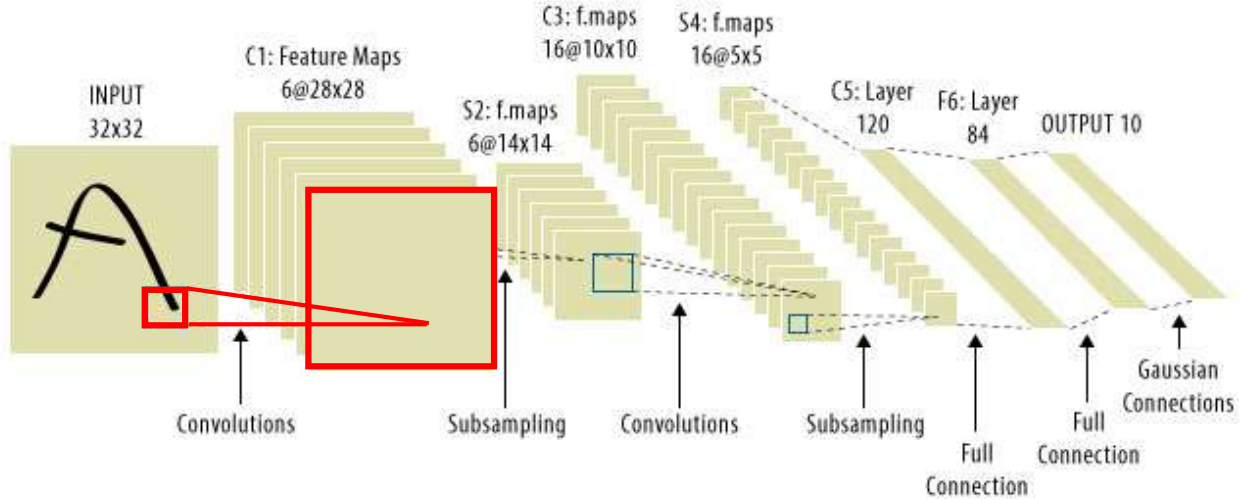


- 输出大小：(N-F)/S+1
  - S为步幅大小
- 比如N=7, F=3
  - 步幅为1，(7-3)/1+1=5
  - 步幅为2，(7-3)/2+1=3
  - 步幅为3，(7-3)/3+1=2.3..

卷积层

输入：10x10图片



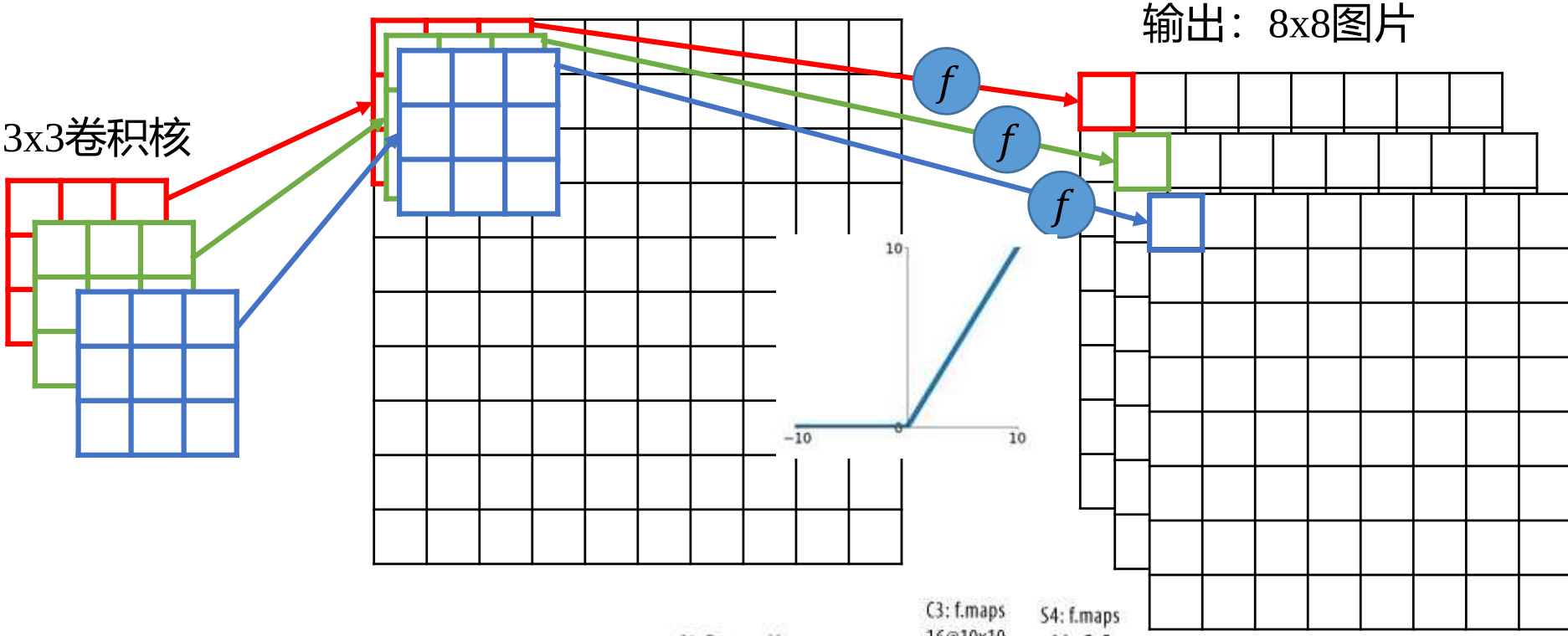


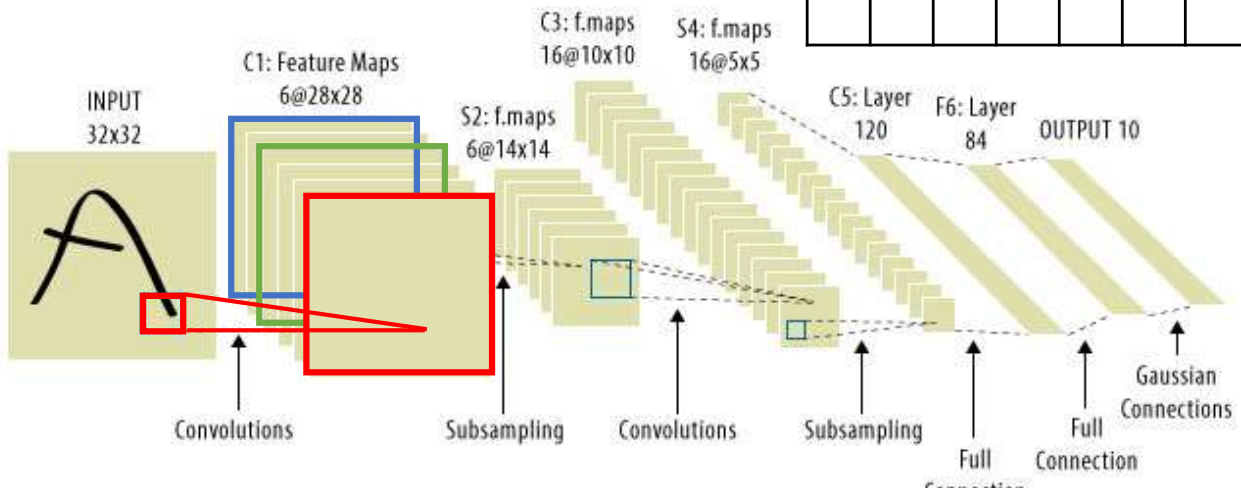
卷积层

输入：10x10图片

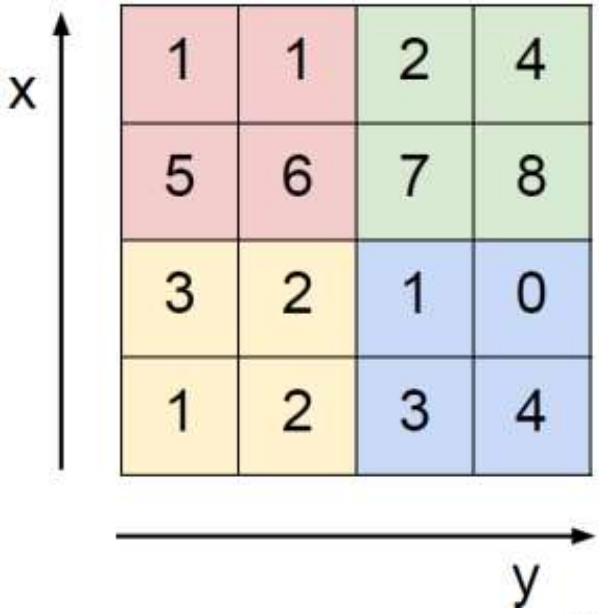
输出：8x8图片

3x3卷积核





池化层



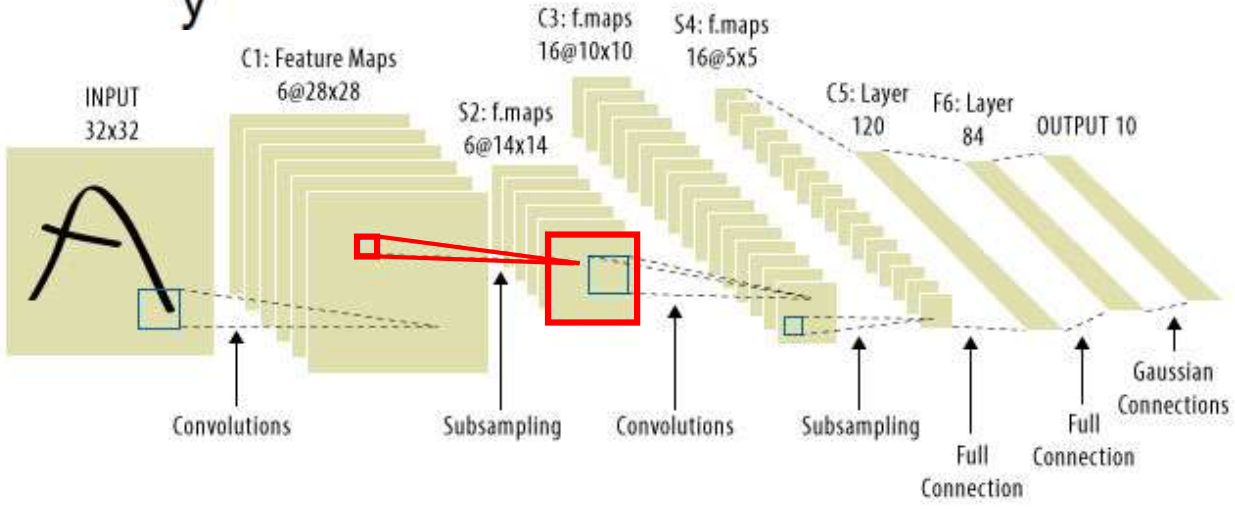
2x2的核，步幅为2

6	8
3	4

最大池化

3.25	5.25
2	2

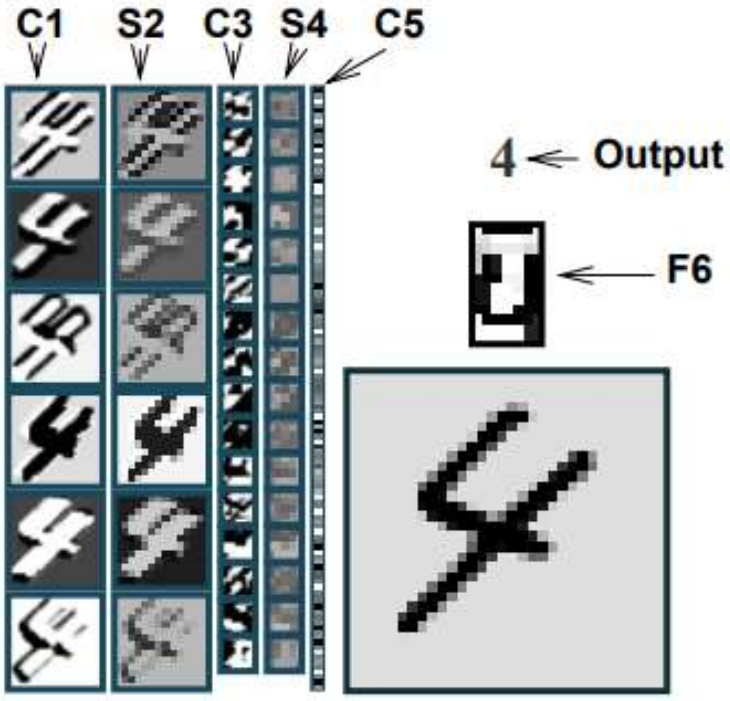
平均池化



手写字符识别

- MNIST (handwritten digits) 数据集 http://yann.lecun.com/exdb/mnist/

6万训练样本和1万测试样本

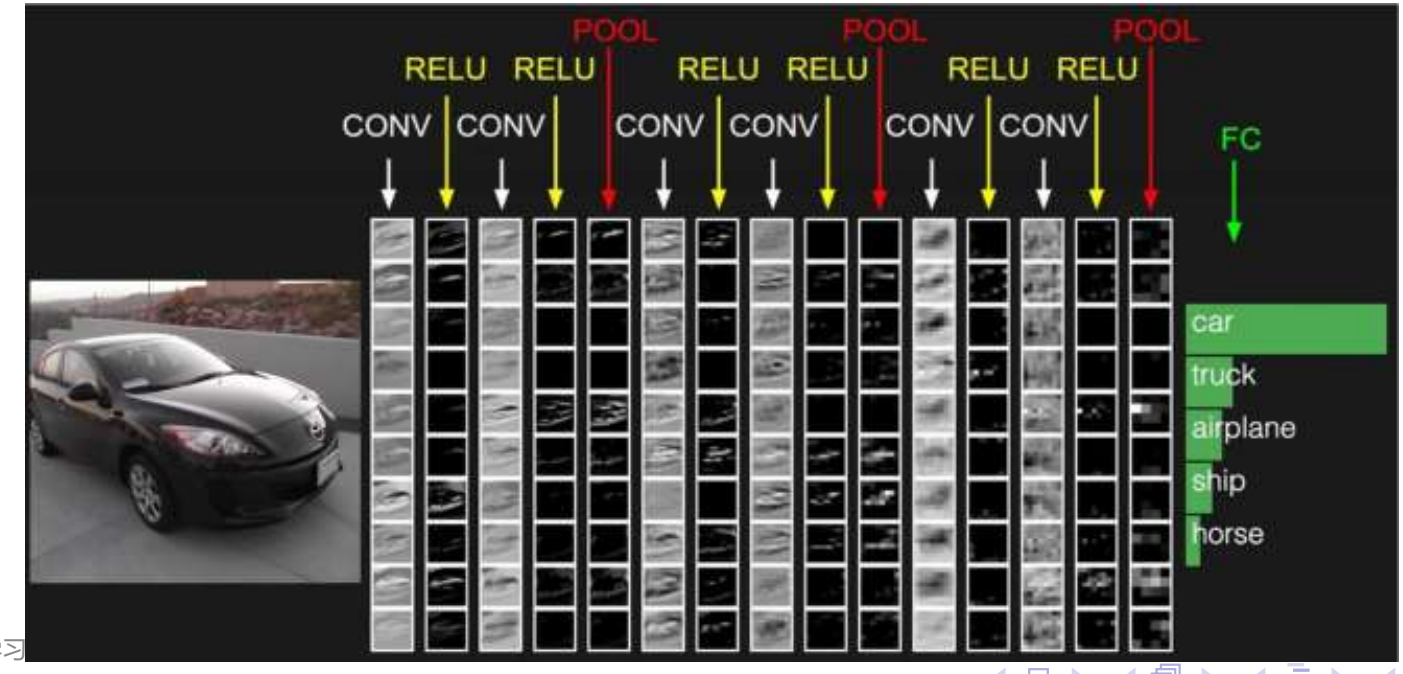


错误案例

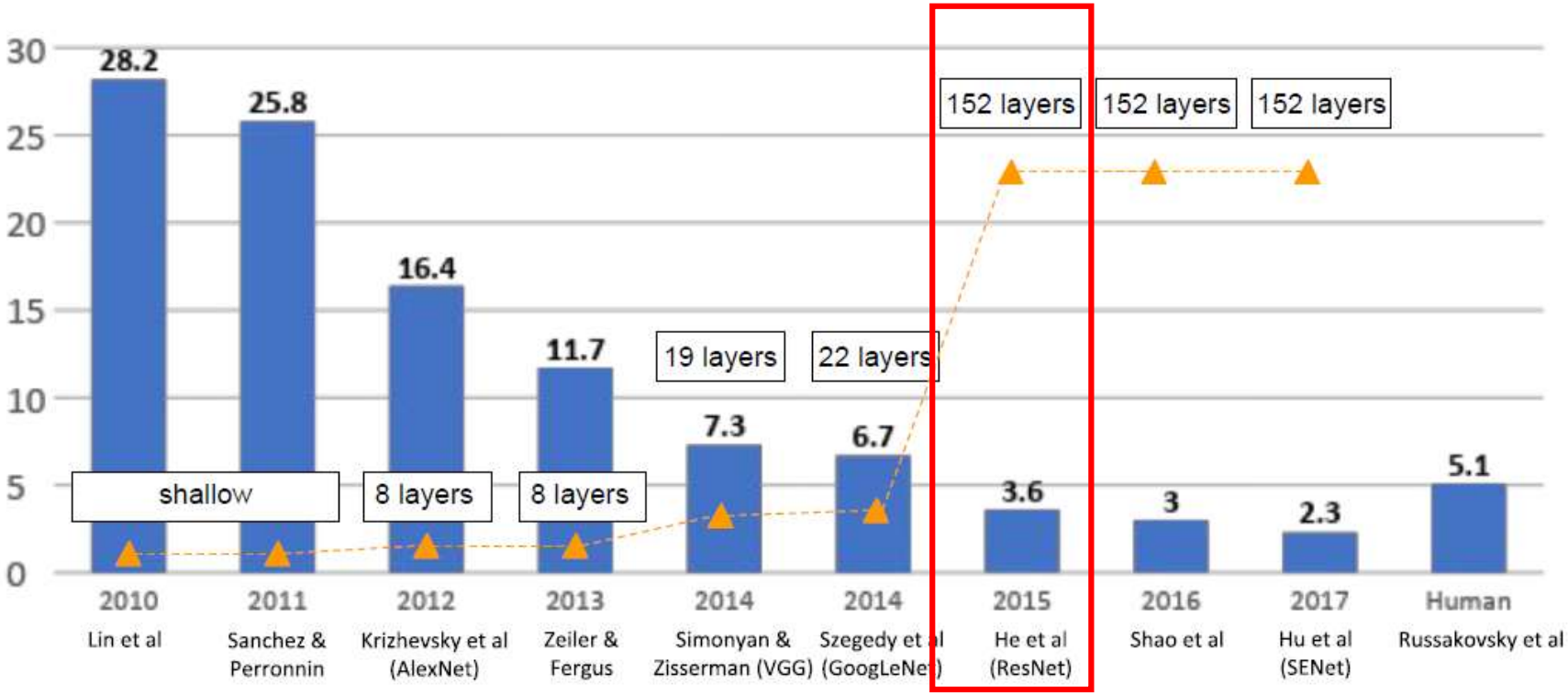
图片分类

ImageNet数据集

大约2万2千个类别，1千5百万张经过Amazon众包平台标注过的图片



图片分类

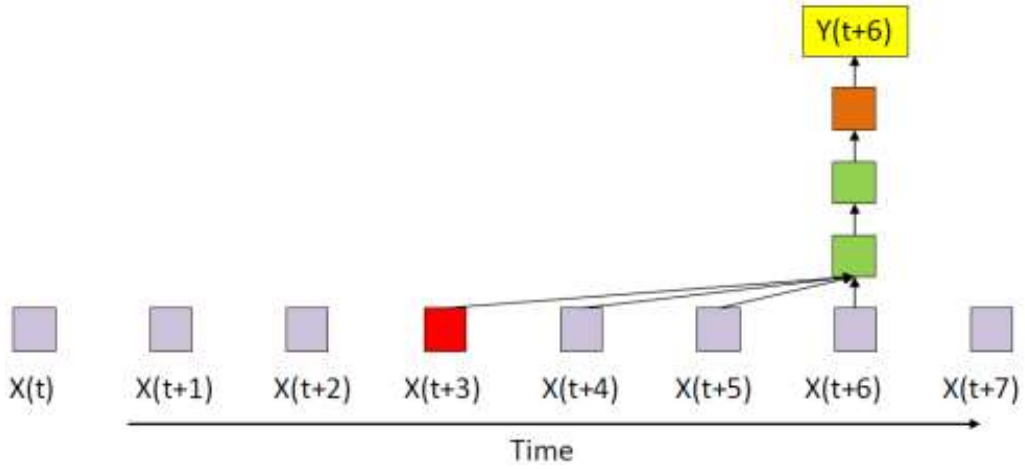


微软ResNet

在ImageNet图像数据库上，达到4.94%的错误率，低于人类5.1%的错误率

循环神经网络

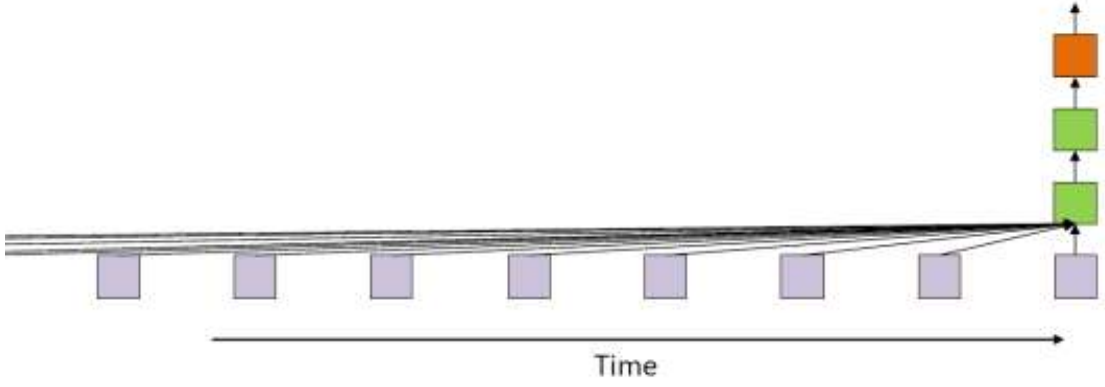
- 建模序列数据，比如文本、时间序列等



有限响应模型

今天的信息只会对未来N天内的预测有用

$$Y_t = f(X_t, X_{t-1}, \cdots, X_{t-N})$$



无限响应模型

今天的信息对未来任何时刻的预测都有用

$$Y_t = f(X_t, X_{t-1}, \cdots, X_{t-\infty})$$

循环神经网络

- 用隐状态变量（状态）$\boldsymbol{h}_t$ “存储” 直到t时刻的历史数据 $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t\}$，并用状态转移方程进行描述

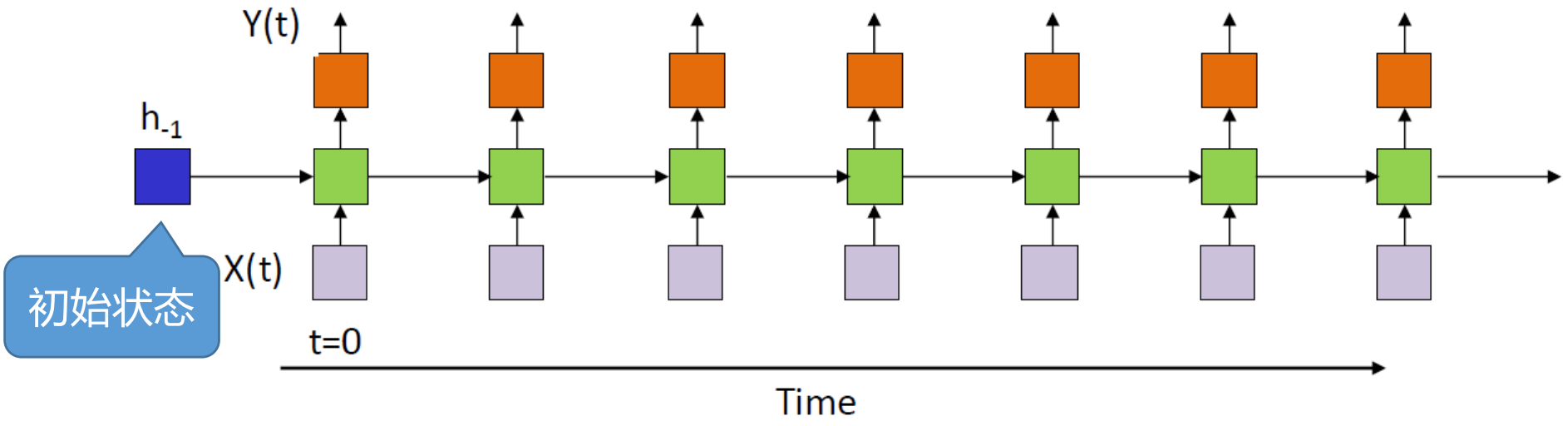
$$\boldsymbol{h}_t = f_W(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$$

每个时间步上的函数相同

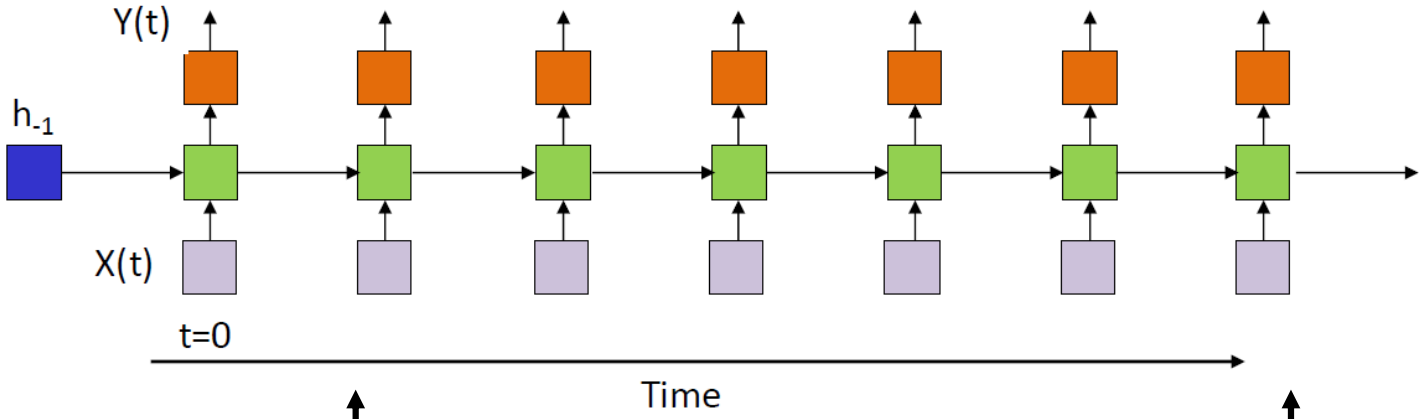
新状态 参数W 旧状态 t时刻的输入

$$\boldsymbol{h}_t = f_W(f_W(f_W(\boldsymbol{h}_{t-3}, \boldsymbol{x}_{t-2}), \boldsymbol{x}_{t-1}), \boldsymbol{x}_t)$$

$$y_t = g(\boldsymbol{h}_t)$$

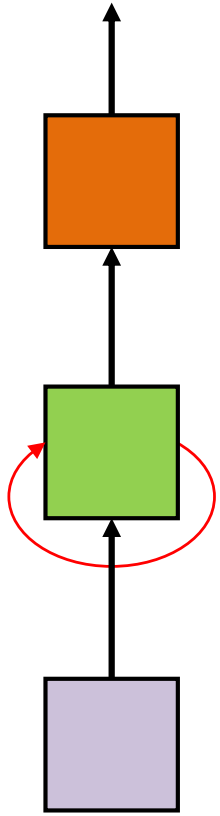


循环神经网络



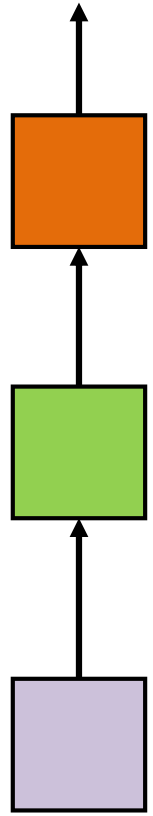
$$y_t = g(\boldsymbol{h}_t)$$

$$\boldsymbol{h}_t = f_W(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$$

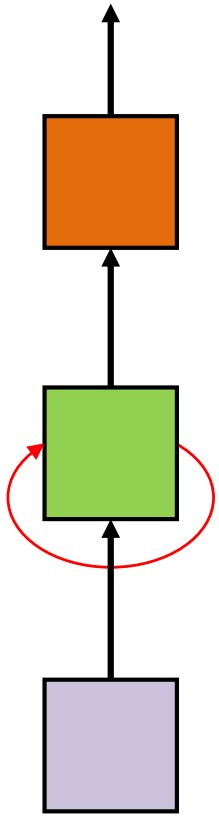


$$y_t = g(\boldsymbol{h}_t)$$

$$\boldsymbol{h}_t = f_W(\boldsymbol{x}_t)$$



循环神经网络

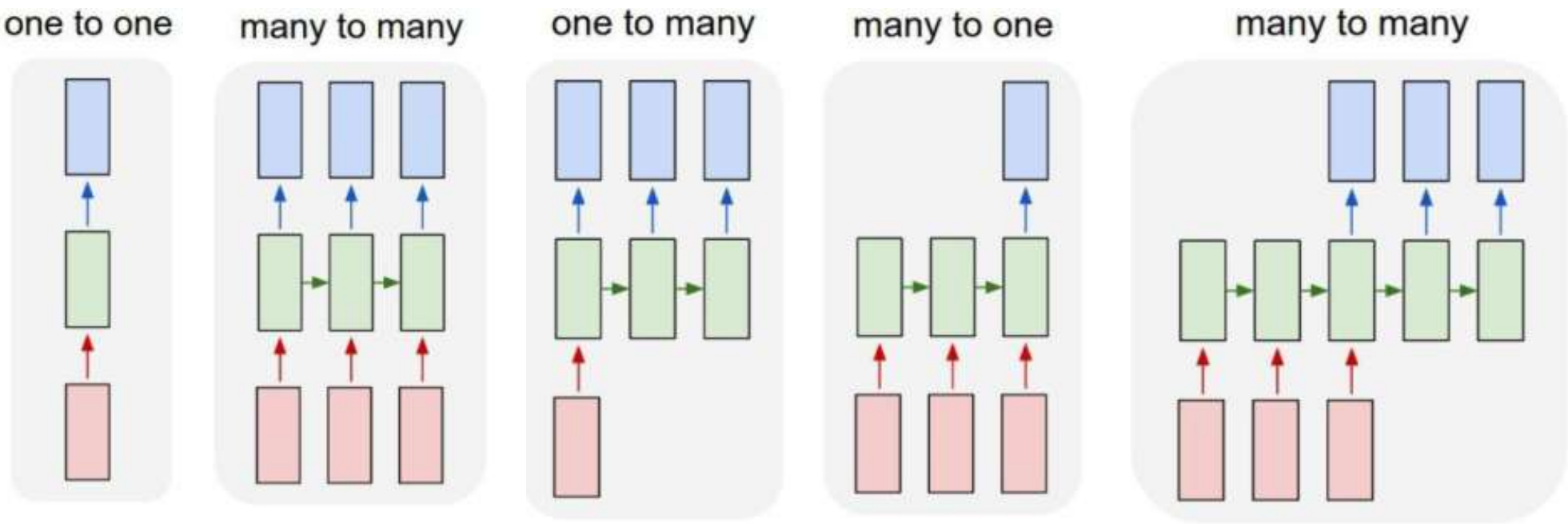


$$y_t = g(\boldsymbol{h}_t) \ = \boldsymbol{W}_{hy}\boldsymbol{h}_t$$

$$\boldsymbol{h}_t = f_{\boldsymbol{W}}(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$$
$$= \tanh(\boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{xh}\boldsymbol{x}_t)$$

称Vanilla RNN
或Elman RNN

循环神经网络



前向网络

例：视频帧分类
视频帧 -> 类别

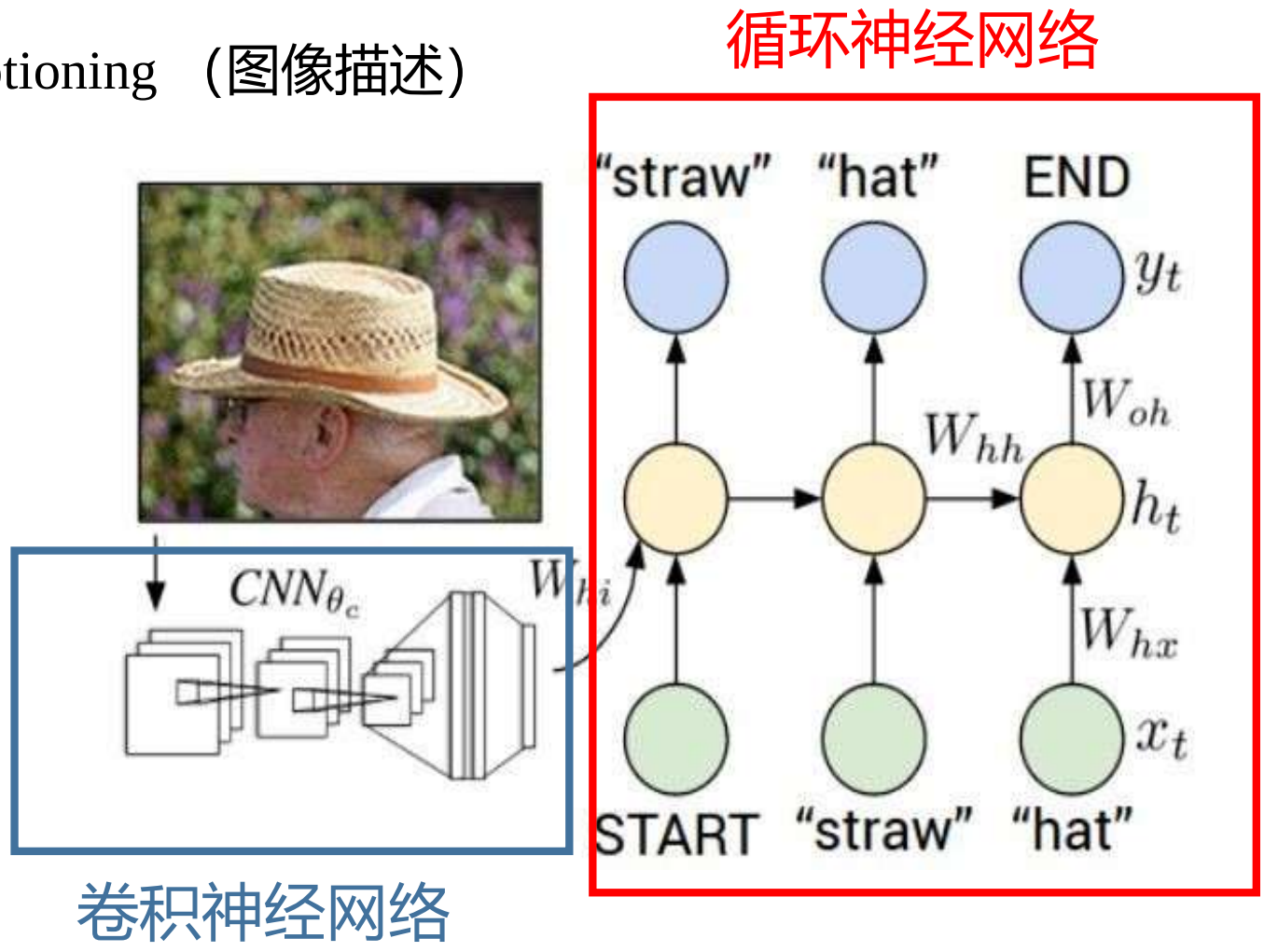
例：图像描述
图像 -> 句子

例：文本分类
文档 -> 类别

例：机器翻译
句子 -> 句子

循环神经网络

Image Captioning （图像描述）



循环神经网络

Image Captioning （图像描述）

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two people walking on the beach with surfboards

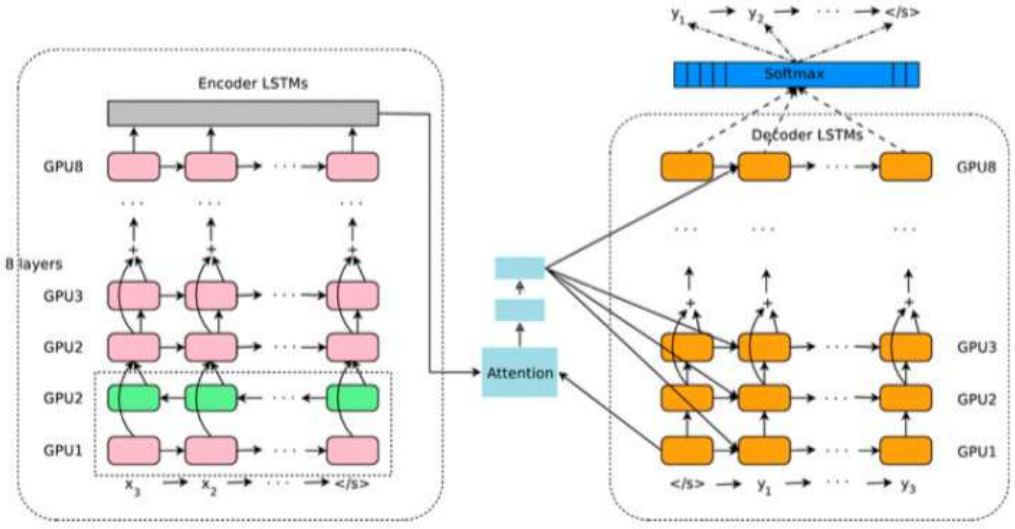
A tennis player in action on the court

Two giraffes standing in a grassy field

A man riding a dirt bike on a dirt track

循环神经网络

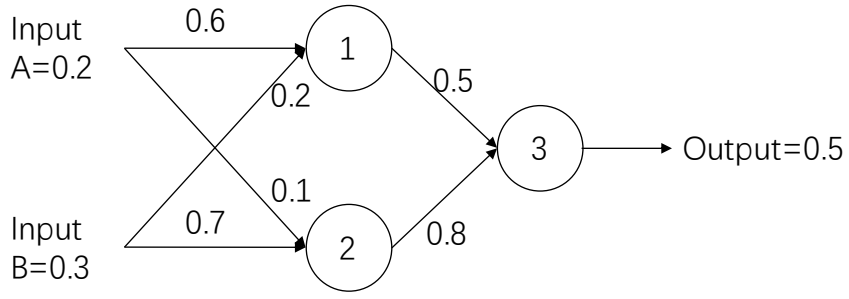
Google 机器翻译



2016 年，Google官方将全产品线的翻译算法换成了基于神经网络的机器翻译系统

作业

- 5.1
- 讨论$\frac{\exp(x_i)}{\sum_{j=1}^{C}\exp(x_j)}$和$\log\sum_{j=1}^{C}\exp(x_j)$的数值溢出问题
- 计算$\frac{\exp(x_i)}{\sum_{j=1}^{C}\exp(x_j)}$和$\log\frac{\exp(x_i)}{\sum_{j=1}^{C}\exp(x_j)}$关于向量$\boldsymbol{x}=[x_1,\cdots,x_C]$的梯度
- 考虑如下简单网络，假设激活函数为ReLU，用平方损失$\frac{1}{2}(y-\hat{y})^2$计算误差，请用BP算法更新一次所有参数（学习率为1），给出更新后的参数值（给出详细计算过程），并计算给定输入值x=(0.2,0.3)时初始时和更新后的输出值，检查参数更新是否降低了平方损失值.



第六章：支持向量机

主讲：连德富 特任教授 | 博士生导师

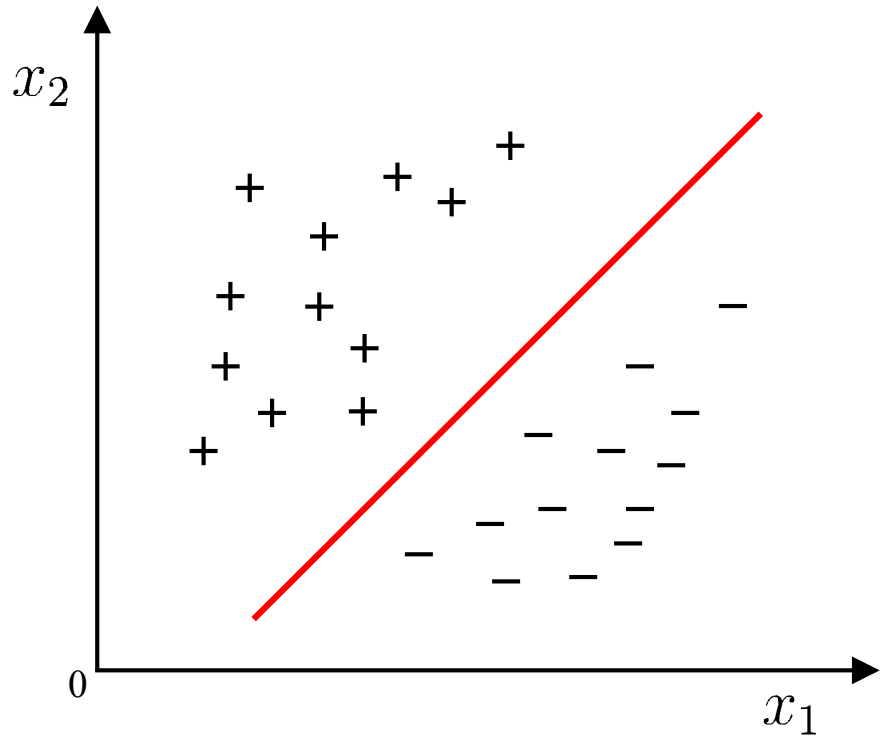
邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

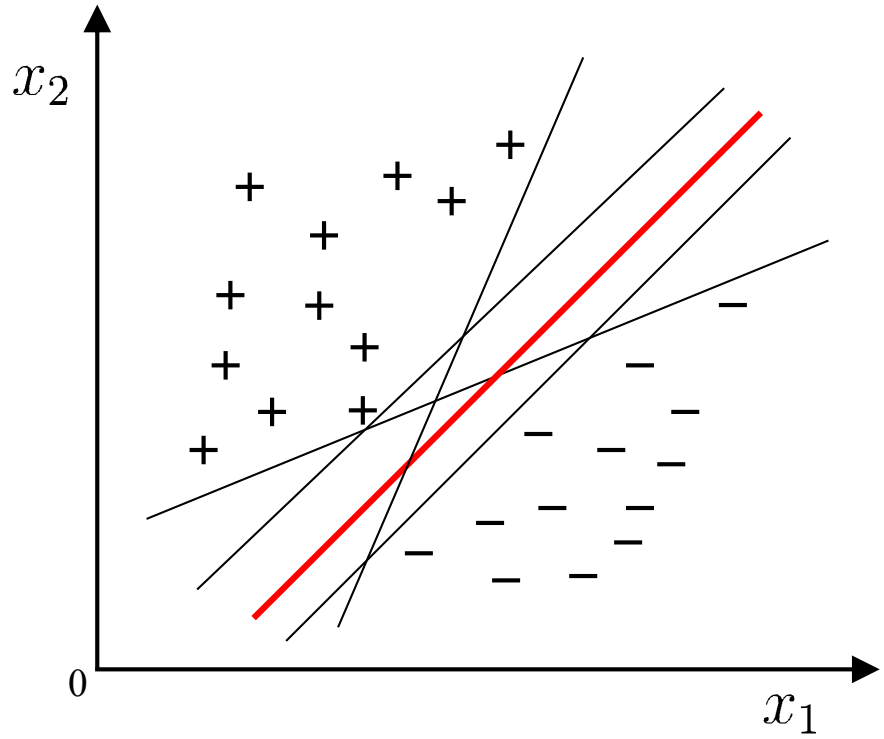
线性模型

在样本空间中寻找一个超平面, 将不同类别的样本分开



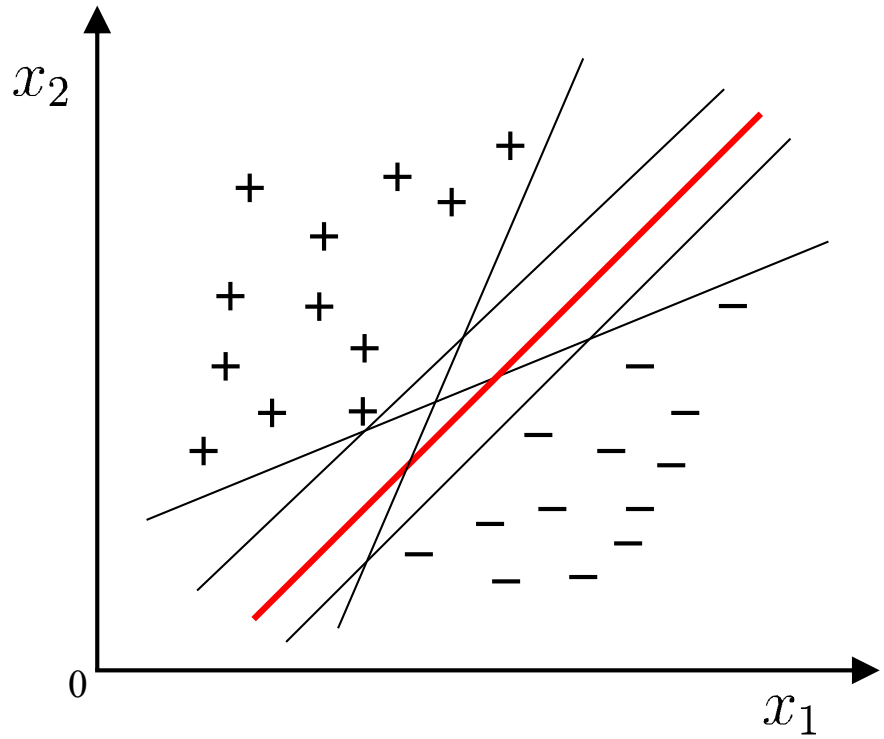
线性模型

- 将训练样本分开的超平面可能有很多, 哪一个好呢?



线性模型

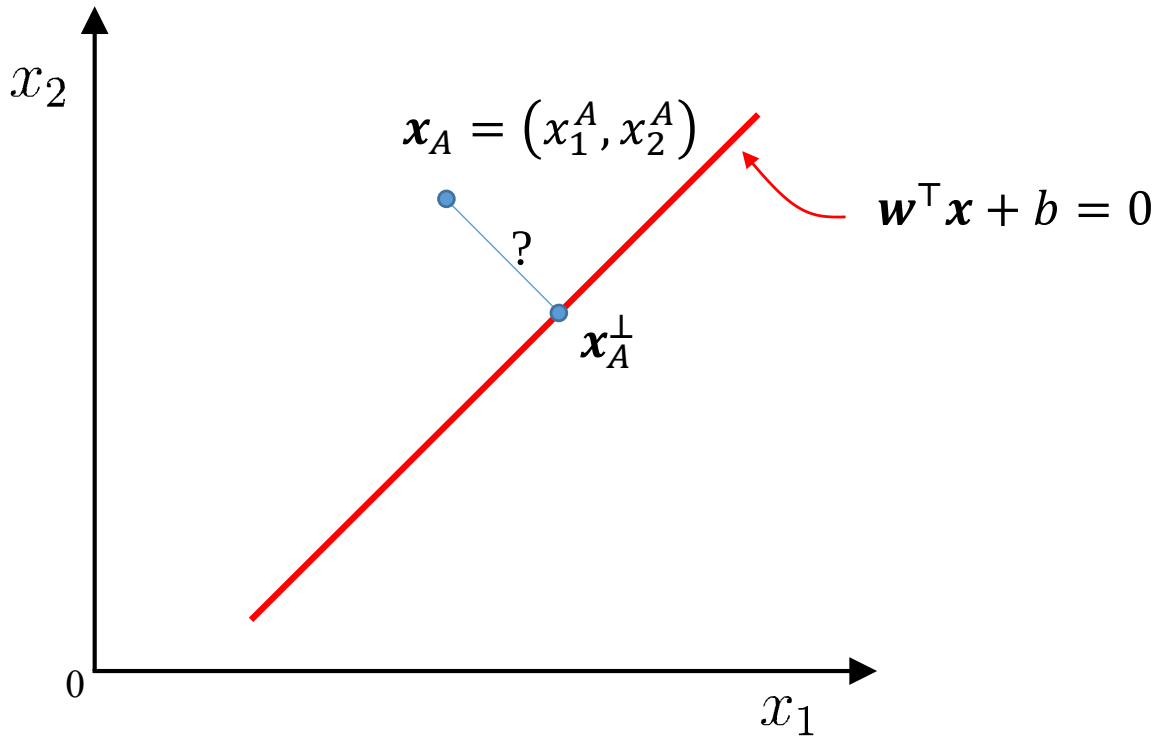
- 将训练样本分开的超平面可能有很多, 哪一个好呢?



应选择”正中间”, 容忍性好, 鲁棒性高, 泛化能力最强

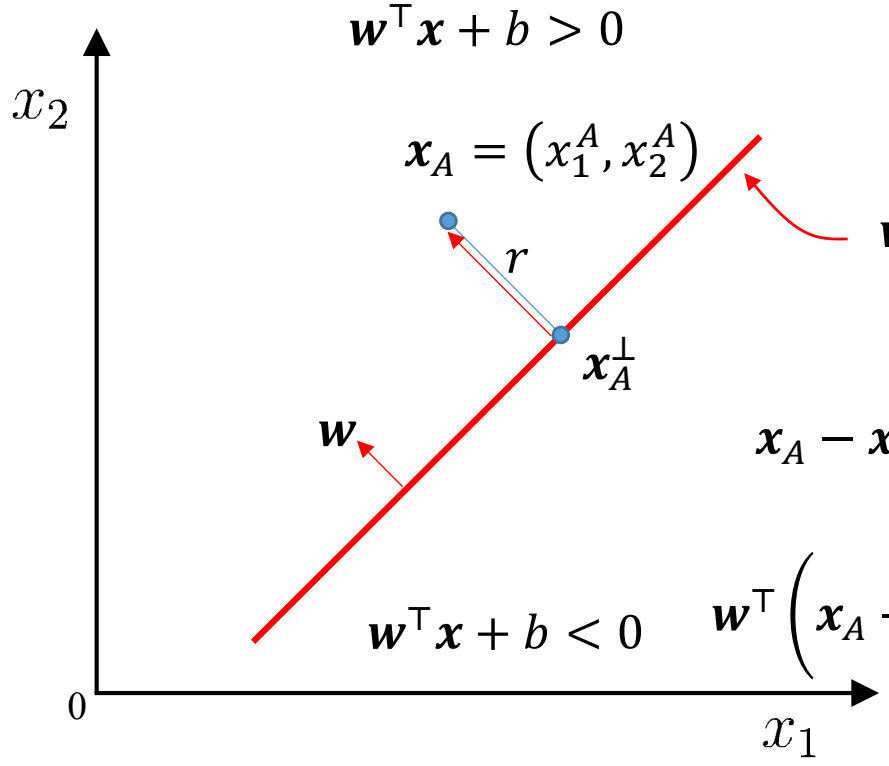
线性模型

- 超平面方程: $\boldsymbol{w}^{\top}\boldsymbol{x}+b=0$



线性模型

- 超平面方程: $\boldsymbol{w}^{\top}\boldsymbol{x}+b=0$



$$r=\frac{\left|\boldsymbol{w}^{\top}\boldsymbol{x}_A+b\right|}{\|\boldsymbol{w}\|}$$

线性模型（线性可分）

- 假设超平面能将训练样本正确分类，即对于$(x_i, y_i) \in D$
  - 若$y_i = +1$，则有$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b > 0$
  - 若$y_i = -1$，则有$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b < 0$

$$\Rightarrow \quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) > 0$$

- 分类平面对于向量$\boldsymbol{w}' = [\boldsymbol{w}, b]$的长度具有不变性

$$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) > 0 \quad \xRightarrow[\alpha > 0]{\boldsymbol{w}' \mapsto \alpha\boldsymbol{w}'} \quad y_i(\alpha\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + \alpha b) > 0$$

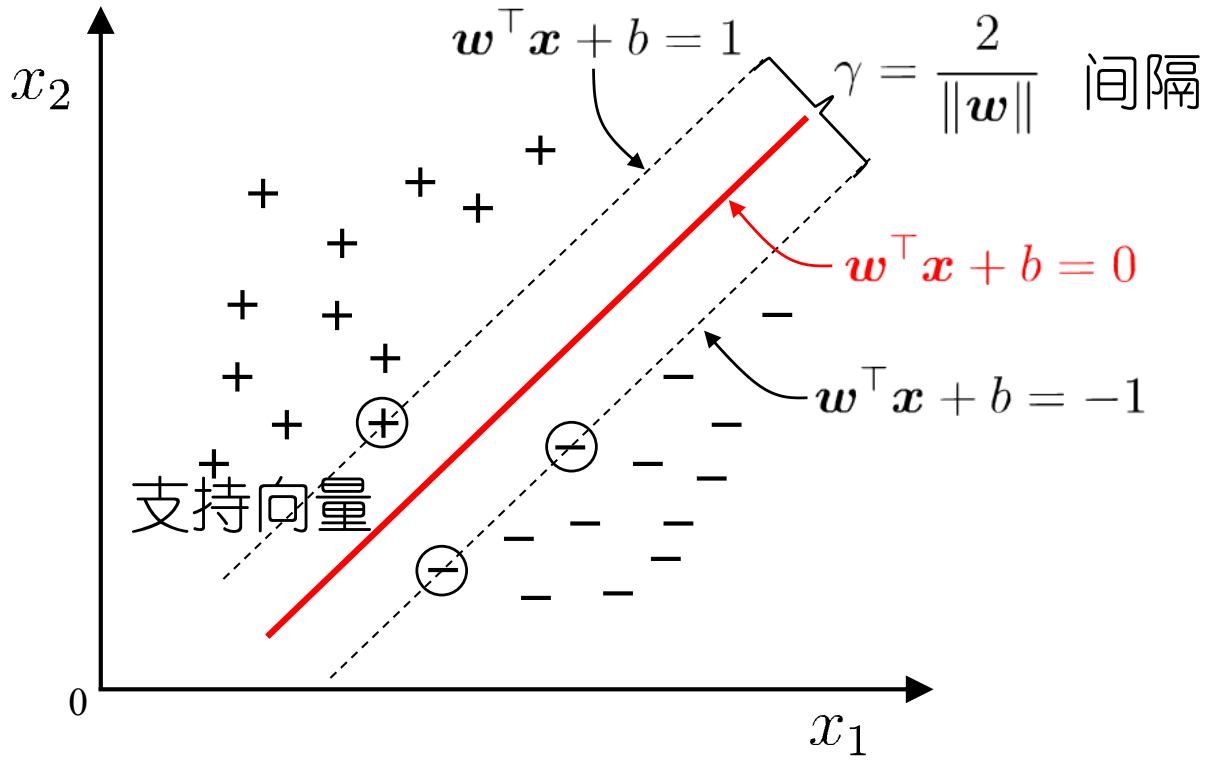
- 假设$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq \epsilon$，则$y_i\left(\frac{\boldsymbol{w}^{\mathrm{T}}}{\epsilon}\boldsymbol{x}_i + \frac{b}{\epsilon}\right) \geq 1$

$$y_i\left(\frac{\boldsymbol{w}^{\mathrm{T}}}{\epsilon}\boldsymbol{x}_i + \frac{b}{\epsilon}\right) \geq 1 \quad \xLeftrightarrow{\frac{1}{\epsilon}(\boldsymbol{w}, b) \mapsto (\boldsymbol{w}', b')} \quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq 1$$

线性模型（线性可分）

- $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$

若$y_i = +1$，则有$\boldsymbol{w}^\top \boldsymbol{x}_i + b \geq +1$
若$y_i = -1$，则有$\boldsymbol{w}^\top \boldsymbol{x}_i + b \leq -1$

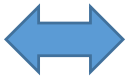


支持向量机基本型

- 最大间隔:寻找参数$\boldsymbol{w}$和b, 使得间隔r最大

$$\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|}$$

$$\text{s.t.}\ y_i\left(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}_i + b\right) \geq 1 \qquad i = 1, \cdots, m$$



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

$$\text{s.t.}\ y_i\left(\boldsymbol{w}^{\mathsf{T}}\boldsymbol{x}_i + b\right) \geq 1 \qquad i = 1, \cdots, m$$

对偶问题

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 \qquad \text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \qquad i = 1, \cdots, m$$

凸函数　　凸函数

凸优化问题

$$\min_{x} f(\boldsymbol{x})$$
$$s.t. \quad g_i(\boldsymbol{x}) \leq 0, i = 1,2, \ldots, m$$
$$\boldsymbol{a}_i^\top \boldsymbol{x} = b, j = 1,2, \cdots, n$$

其中$f(\boldsymbol{x}), g_i(\boldsymbol{x})$是凸函数

对偶问题

原问题 凸优化

$$\min_x f(\boldsymbol{x})$$
$$s.t.\quad g_i(\boldsymbol{x}) \le 0, i = 1,2,\dots,m$$
$$h_j(\boldsymbol{x}) = 0, j = 1,2,\cdots,n$$

广义拉格朗日函数

$$L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) = f(\boldsymbol{x}) + \sum_i u_i\, g_i(\boldsymbol{x}) + \sum_j v_j\, h_j(\boldsymbol{x})$$

其中$u_i \ge 0$

对偶问题 凸优化

$$\max_{\boldsymbol{u},\boldsymbol{v}} g(\boldsymbol{u},\boldsymbol{v})\ s.t.\ \ \boldsymbol{u} \geqslant 0$$

其中$g(\boldsymbol{u},\boldsymbol{v}) = \min_{\boldsymbol{x}} L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v})$

对偶问题

原问题 凸优化

$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2/2$$
$$s.t.\ \ y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, i = 1,2,\dots,m$$

广义拉格朗日函数

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1)$$

其中$\alpha_i \geq 0$

对偶问题 凸优化

$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha})\ s.t.\ \ \boldsymbol{\alpha} \geqslant 0$$

其中$g(\boldsymbol{\alpha}) = \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$

对偶问题

- $g(\boldsymbol{\alpha}) = \min\limits_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$

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

$L(\boldsymbol{w}, b, \boldsymbol{\alpha})$是关于$\boldsymbol{w}$和$b$的凸函数，　在其梯度等于0时取得最优值

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = 0 \quad \Rightarrow \quad \boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{b}} = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$

对偶问题

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1)$$

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$

$$g(\boldsymbol{\alpha}) = \frac{1}{2}\left\|\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i\right\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i \left(\left(\sum_{j=1}^{m} \alpha_j y_j \boldsymbol{x}_j \right)^\top \boldsymbol{x}_i \right) \right) + \sum_{i}^{m} \alpha_i$$

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

对偶问题

$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j$$
$$s.t. \ \ \boldsymbol{\alpha} \geqslant 0 \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0$$

对偶问题

原问题 凸优化

$$\min_{x} f(\boldsymbol{x})$$
$$s.t. \quad g_i(\boldsymbol{x}) \leq 0, i = 1,2,\ldots,m$$
$$h_j(\boldsymbol{x}) = 0, j = 1,2,\cdots,n$$

对偶问题 凸优化

$$\max_{\boldsymbol{u},\boldsymbol{v}} g(\boldsymbol{u},\boldsymbol{v}) \; s.t. \; \boldsymbol{u} \geqslant 0$$

其中$g(\boldsymbol{u},\boldsymbol{v}) = \min_{\boldsymbol{x}} L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v})$

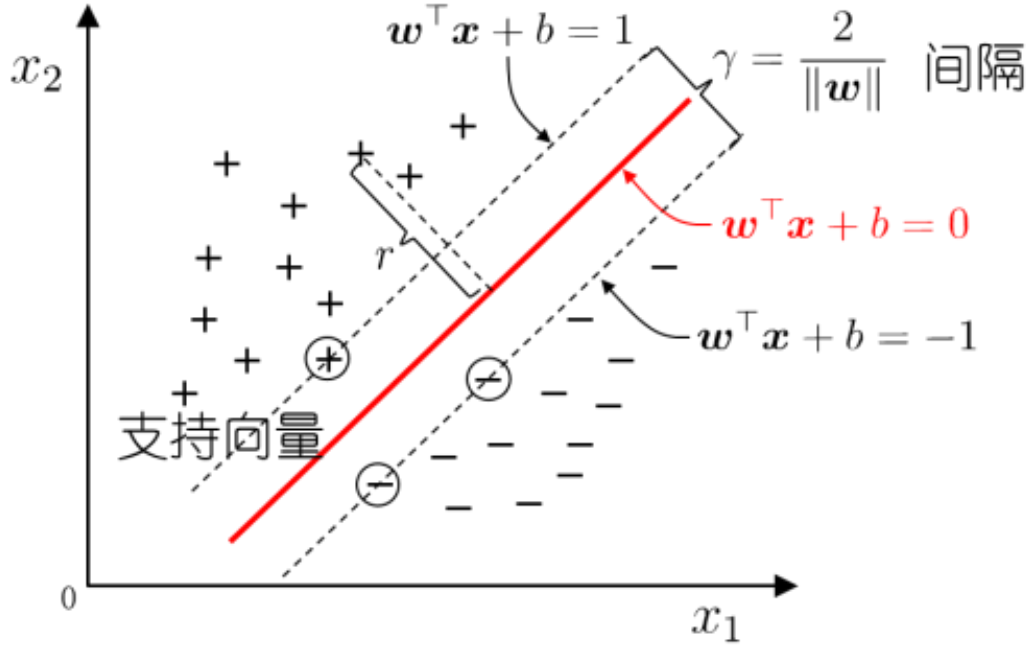
强对偶性

$$f^\star = g^\star = \max_{\boldsymbol{u},\boldsymbol{v}} g(\boldsymbol{u},\boldsymbol{v})$$

Slater条件

原问题为凸优化问题，且可行域中至少有一个点使得不等式约束严格成立

对偶问题



- 对于线性可分问题，一定存在 $(\boldsymbol{w}, b)$ 使得 $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$，

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

- 设$\epsilon < 1$，则$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) > \epsilon$ ⇒ $y_i\left(\frac{\boldsymbol{w}^\top}{\epsilon}\boldsymbol{x}_i + \frac{b}{\epsilon}\right) > 1$

则$\left(\frac{\boldsymbol{w}}{\epsilon}, \frac{b}{\epsilon}\right)$严格满足不等式。因此，该优化问题满足强对偶性条件

对偶问题

对偶问题

$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \underbrace{\boldsymbol{x}_i^{\top} \boldsymbol{x}_j}_{K_{ij}}$$
$$s.t.\ \ \boldsymbol{\alpha} \geqslant 0 \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0$$

通过序列最小优化（SMO）求解最优的$\boldsymbol{\alpha}$

基本思路：不断执行如下两个步骤直至收敛

第一步：选取一对需更新的变量α_i和α_j

第二步：固定α_i和α_j以外的参数，求解对偶问题更新α_i和α_j

$$g(\alpha_i, \alpha_j) = \alpha_i + \alpha_j - \frac{1}{2}\left(\alpha_i^2 K_{ii} + \alpha_j^2 K_{jj} + 2\alpha_i \alpha_j y_i y_j K_{ij} + \alpha_i y_i \sum_{k \neq i,j} \alpha_k y_k K_{ik} + \alpha_j y_j \sum_{k \neq i,j} \alpha_k y_k K_{jk}\right)$$

仅考虑α_i和α_j时，对偶问题的约束变为

$$\alpha_i y_i + \alpha_j y_j = -\sum_{k \neq i,j}^{m} \alpha_i y_i = \zeta \quad \Rightarrow \quad \alpha_i = (\zeta - \alpha_j y_j) y_i$$

对偶问题

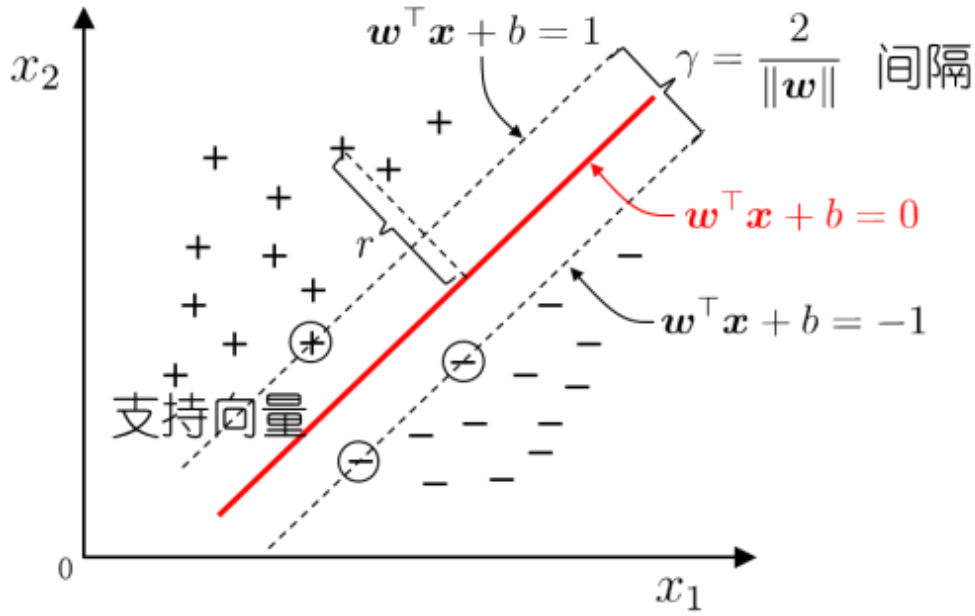
- 终止条件：KKT条件

$$\begin{cases} \alpha_i \geq 0 \\ y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \\ \alpha_i(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1) = 0 \end{cases}$$

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) > 1 \Rightarrow \alpha_i = 0$$

$$\alpha_i > 0 \Rightarrow y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 1$$

支持向量机解的稀疏性：训练完成后，大部分的训练样本都不需保留，最终模型仅与支持向量有关



对偶问题

- 如何选择两个变量?

$$\begin{cases} \alpha_i \geq 0 \\ y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \\ \alpha_i(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1) = 0 \end{cases}$$

若α_i和α_j有一个违反了KKT条件，目标函数就会在迭代后增大 Osuna et al. 1997

- KKT条件违背的程度越大，则变量更新后可能导致的目标函数值增幅越大

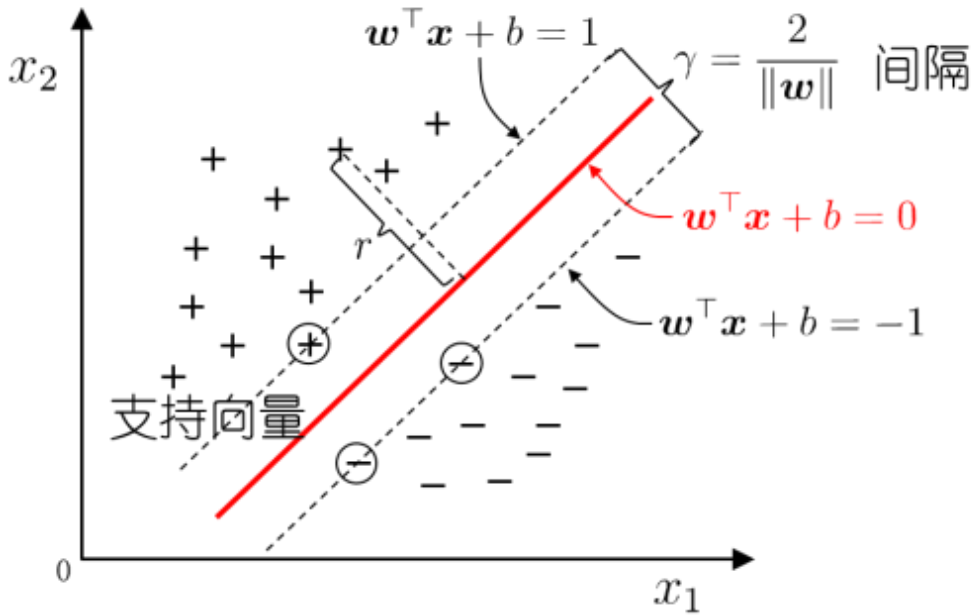
第一个变量：选取违背KKT条件程度最大的变量
第二个变量：与第一个变量的间隔最大的变量

对偶问题

给定最优$\boldsymbol{\alpha}$，求解$\boldsymbol{w}$和b

$$\boldsymbol{\alpha}^\star = \max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha})$$

$$\boldsymbol{w}^\star = \sum_{i=1}^{m} \alpha_i^\star y_i \boldsymbol{x}_i$$



对于任意的支持向量 $\boldsymbol{x}_i \in S$，均满足$\boldsymbol{w}^\top \boldsymbol{x}_i + b = y_i$，则

$$b^\star = \frac{1}{|S|} \sum_{i \in S} \left(y_i - \boldsymbol{x}_i^\top \boldsymbol{w}^\star \right)$$

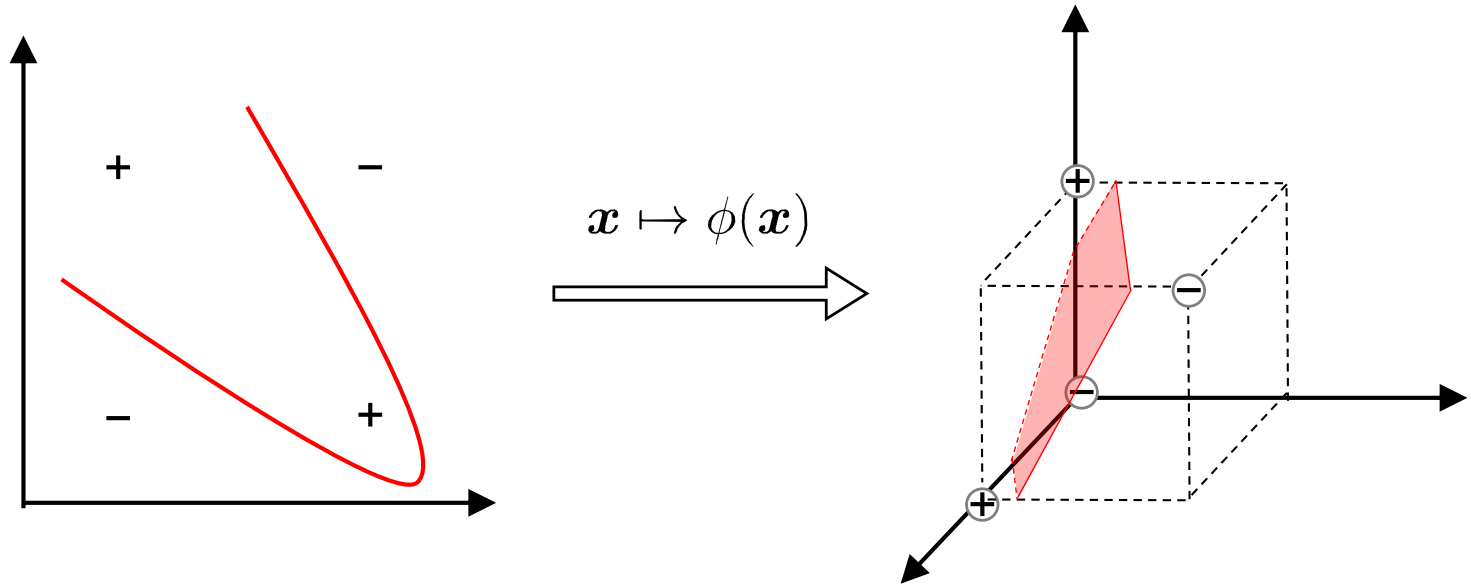
支持向量机解的稀疏性: 训练完成后, 大部分的训练样本都不需保留, 最终模型仅与支持向量有关

$$y = \boldsymbol{x}^\top \boldsymbol{w} + b = \sum_{i}^{m} \alpha_i^\star y_i \boldsymbol{x}^\top \boldsymbol{x}_i + b^\star$$

线性模型（线性不可分）

- 若不存在一个能正确划分两类样本的超平面, 怎么办

将样本从原始空间映射到一个更高维的特征空间, 使得样本在这个特征空间内线性可分



核支持向量机

- 设样本$\boldsymbol{x}$映射后的向量为$\phi(\boldsymbol{x})$, 划分超平面为$\boldsymbol{w}^{\top}\phi(\boldsymbol{x})+b=0$

原问题

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|^2/2$$
$$s.t.\ \ y_i(\boldsymbol{w}^{\top}\phi(\boldsymbol{x}_i)+b)\geq 1, i=1,2,\ldots,m$$

对偶问题

$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha})=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \phi(\boldsymbol{x}_i)^{\top}\phi(\boldsymbol{x}_j)$$
$$s.t.\ \ \boldsymbol{\alpha}\geqslant 0 \text{ and } \sum_{i=1}^{m}\alpha_i y_i=0$$

只以内积的形式出现

预测函数

$$y=\boldsymbol{x}^{\top}\boldsymbol{w}+b=\sum_{i}^{m}\alpha_i^{\star}y_i\phi(\boldsymbol{x})^{\top}\phi(\boldsymbol{x}_i)+b^{\star}$$

核支持向量机

- 由于特征空间维数可能很高，甚至是无穷维，因此直接计算$\phi(\boldsymbol{x})^{\mathrm{T}}\phi(\boldsymbol{x}_i)$通常是困难的。
- 可以设计函数$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_i)$

核函数

对偶问题

$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha})=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)$$
$$s.t.\ \ \boldsymbol{\alpha}\geqslant 0 \text{ and } \sum_{i=1}^{m}\alpha_i y_i=0$$

预测函数

$$y=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}+b=\sum_{i}^{m}\alpha_i^{\star} y_i \kappa(\boldsymbol{x}_i,\boldsymbol{x})+b^{\star}$$

核支持向量机

- 若已知$\phi(\cdot)$，则可写出核函数$\kappa(\cdot,\cdot)$；但现实任务中通常不知道$\phi(\cdot)$的形式
  - 核函数是否存在?
  - 什么样的函数可以作为核函数?

令$\mathcal{X}$为输入空间，$\mathcal{H}$为特征空间，如果存在$\phi(\cdot)$: $\mathcal{X} \mapsto \mathcal{H}$，使得对所有$\boldsymbol{x}, \boldsymbol{z} \in \mathcal{X}$，函数$\kappa(\cdot,\cdot)$满足条件

$$\kappa(\boldsymbol{x}, \boldsymbol{z}) = \phi(\boldsymbol{x})^{\mathrm{T}} \phi(\boldsymbol{z})$$

则称$\kappa(\cdot,\cdot)$为核函数

Mercer定理：令$\mathcal{X}$为输入空间，$\kappa(\cdot,\cdot)$是定义在$\mathcal{X} \times \mathcal{X}$上的对称函数，则$\kappa$是核函数当且仅当对于任意数据集$D = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$，核矩阵K总是半正定的

$$K = \begin{bmatrix} \kappa(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_1, \boldsymbol{x}_m) \\ \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_m, \boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_m, \boldsymbol{x}_m) \end{bmatrix}$$

核支持向量机

- 常用核函数：

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j)^d$	$d \geq 1$为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\delta^2}\right)$	$\delta > 0$为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\delta}\right)$	$\delta > 0$
Sigmoid核	$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \boldsymbol{x}_i^\top \boldsymbol{x}_j + \theta)$	tanh为双曲正切函数, $\beta > 0, \theta < 0$

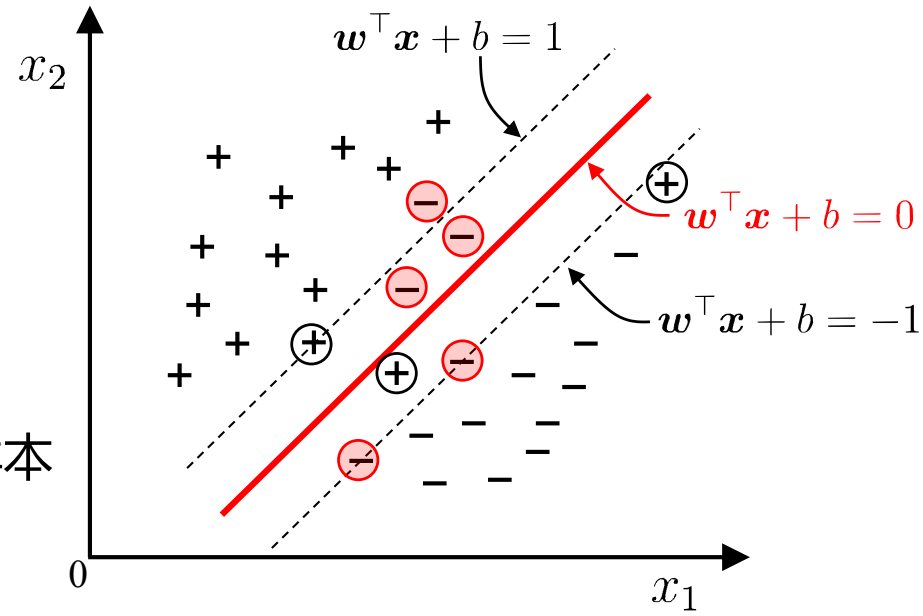
- 函数组合得到
  - 核函数线性组合$\gamma_1\kappa_1 + \gamma_2\kappa_2$
  - 核函数直积 $\kappa_1 \otimes \kappa_2(\boldsymbol{x}, \boldsymbol{z}) = \kappa_1(\boldsymbol{x}, \boldsymbol{z}) \otimes \kappa_1(\boldsymbol{x}, \boldsymbol{z})$
  - $g(\boldsymbol{x})\kappa(\boldsymbol{x}, \boldsymbol{z})g(\boldsymbol{z})$

线性模型（线性不可分）

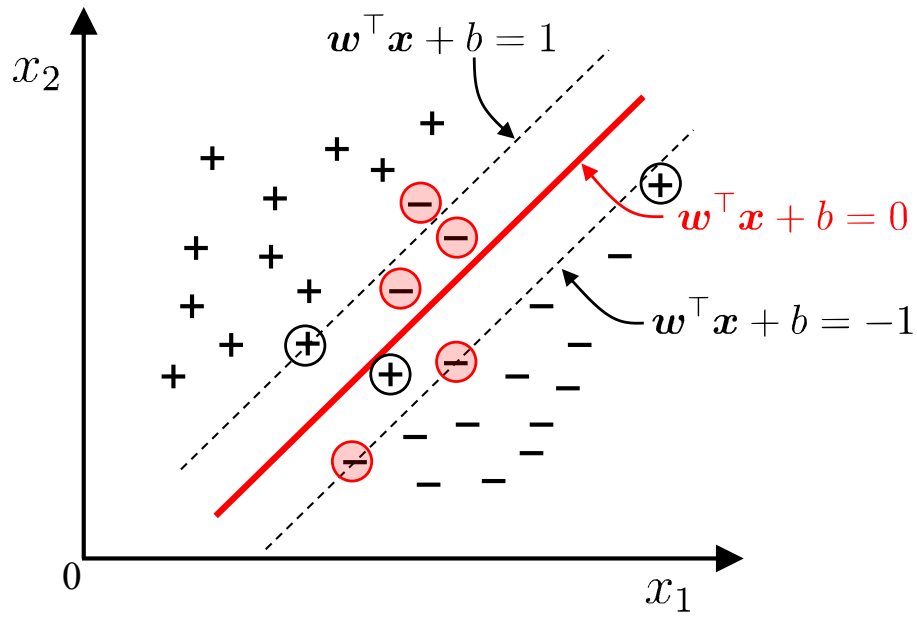
- 很难确定合适的核函数使得训练样本在特征空间中线性可分
- 一个线性可分的结果也很难断定是否是有过拟合造成的

引入软间隔的概念, 允许支持向量机在一些样本上不满足约束

不满足约束的样本



软间隔支持向量机



$$\min_{\boldsymbol{w},b,\xi} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

正则化

$$s.t.\ \ y_i(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, i = 1,2,\ldots,m$$

为每个样本$(\boldsymbol{x}_i, y_i)$引入松弛变量ξ_i　　$\xi_i \geq 0, i = 1,2,\ldots,m$

软间隔支持向量机

原问题 凸优化

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t.\ \ y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x_i}) + b) \geq 1 - \xi_i, i = 1,2,\dots,m$$

$$\xi_i \geq 0, i = 1,2,\dots,m$$

广义拉格朗日函数

$$L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i(y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x_i}) + b) - 1 + \xi_i) - \sum_{i}^{m}\mu_i\xi_i$$

$$\alpha_i, \mu_i \geq 0$$

对偶问题

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

$$s.t.\ C \succcurlyeq \boldsymbol{\alpha} \succcurlyeq 0 \text{ and } \sum_{i=1}^{m}\alpha_i y_i = 0$$

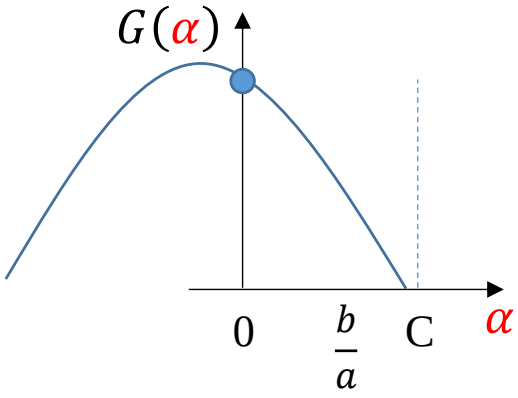
软间隔支持向量机

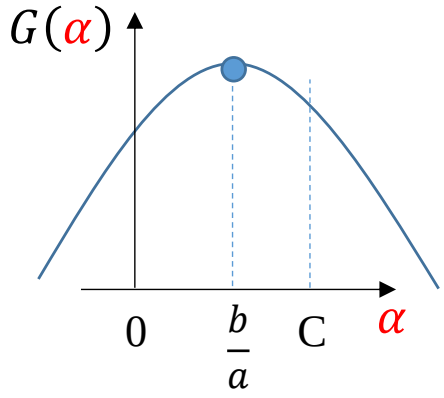
对偶问题

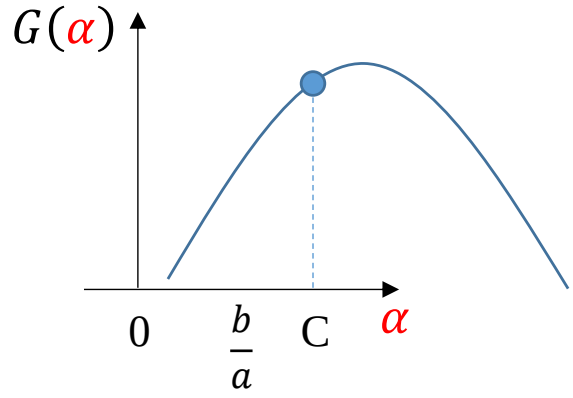
$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\top} \boldsymbol{x}_j$$
$$s.t.\ C \succcurlyeq \boldsymbol{\alpha} \succcurlyeq 0 \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0$$

SMO

$$\max_{\alpha} G(\alpha) = -a\alpha^2 + 2b\alpha + c$$
$$\text{s.t. } 0 \le \alpha \le C$$







软间隔支持向量机

- 终止条件：KKT条件

$$\begin{cases} \alpha_i \geq 0, \mu_i \geq 0 \\ y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq 1 - \xi_i \\ \alpha_i(y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - 1 + \xi_i) = 0 \\ \xi_i \geq 0, \mu_i\xi_i = 0 \end{cases}$$

对于样本$(\boldsymbol{x}_i, y_i)$，若$\alpha_i > 0$，则$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = 1 - \xi_i$.　该样本是支持向量

若$\alpha_i < C$，由$C = \alpha_i + \mu_i$得出$\mu_i > 0$，从而得出$\xi_i = 0$.　该样本在最大间隔边界上

若$\alpha_i = C$，由$C = \alpha_i + \mu_i$得出$\mu_i = 0$

若$\xi_i \leq 1$，　该样本在最大间隔内

若$\xi_i > 1$，　该样本被错误分类

软间隔支持向量机

- 目标函数视角

基本想法：最大化间隔的同时，让不满足约束的样本应尽可能少.

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \ell_{0/1}(y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + b) - 1)$$

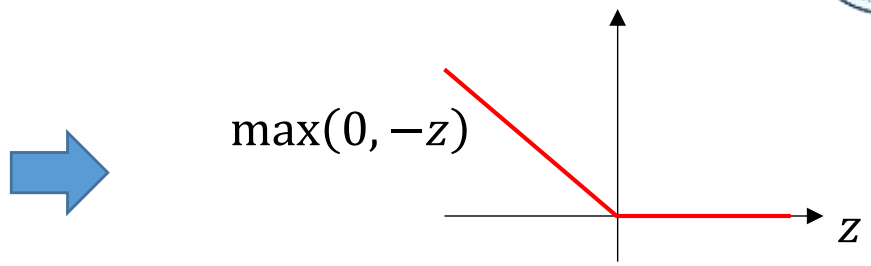
其中 $\ell_{0/1}$是0/1损失函数

$$\ell_{0/1}(z) = \begin{cases} 1, z < 0 \\ 0, otherwise \end{cases}$$

存在的问题：0/1损失函数非凸、非连续，不易优化！

软间隔支持向量机

$$\ell_{0/1}(z)=\begin{cases}1, z<0\\ 0, otherwise\end{cases}$$



$$\min_{\boldsymbol{w},b}\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{m}\max\left(0,1-y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b)\right)$$

$$\xi_i=\max\left(0,1-y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b)\right)$$

$$\min_{\boldsymbol{w},b,\xi}\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{m}\xi_i$$

$$\text{s.t.}\quad 1-y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b)\le\xi_i \quad\Rightarrow\quad y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b)\ge 1-\xi_i$$

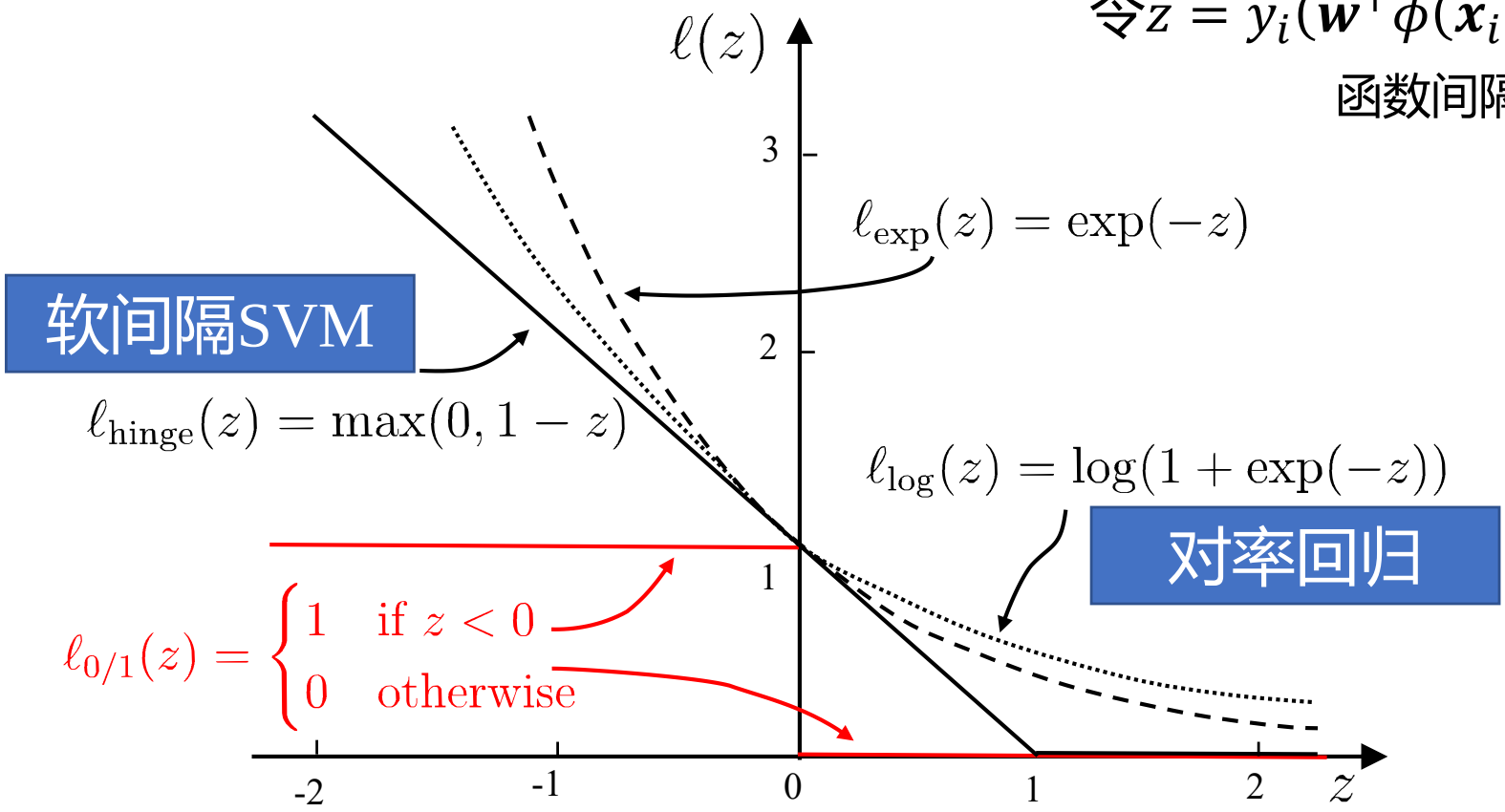
$$0\le\xi_i$$

替代损失函数

$y_i \in \{\pm 1\}$

令$z = y_i(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) + b)$

函数间隔



软间隔SVM

对率回归

替代损失函数数学性质较好，一般是0/1损失函数的上界

正则化

- 支持向量机学习模型的更一般形式

$$\min_{f} \Omega(f) + C \sum_{i}^{m} \ell(f(x_i), y_i)$$

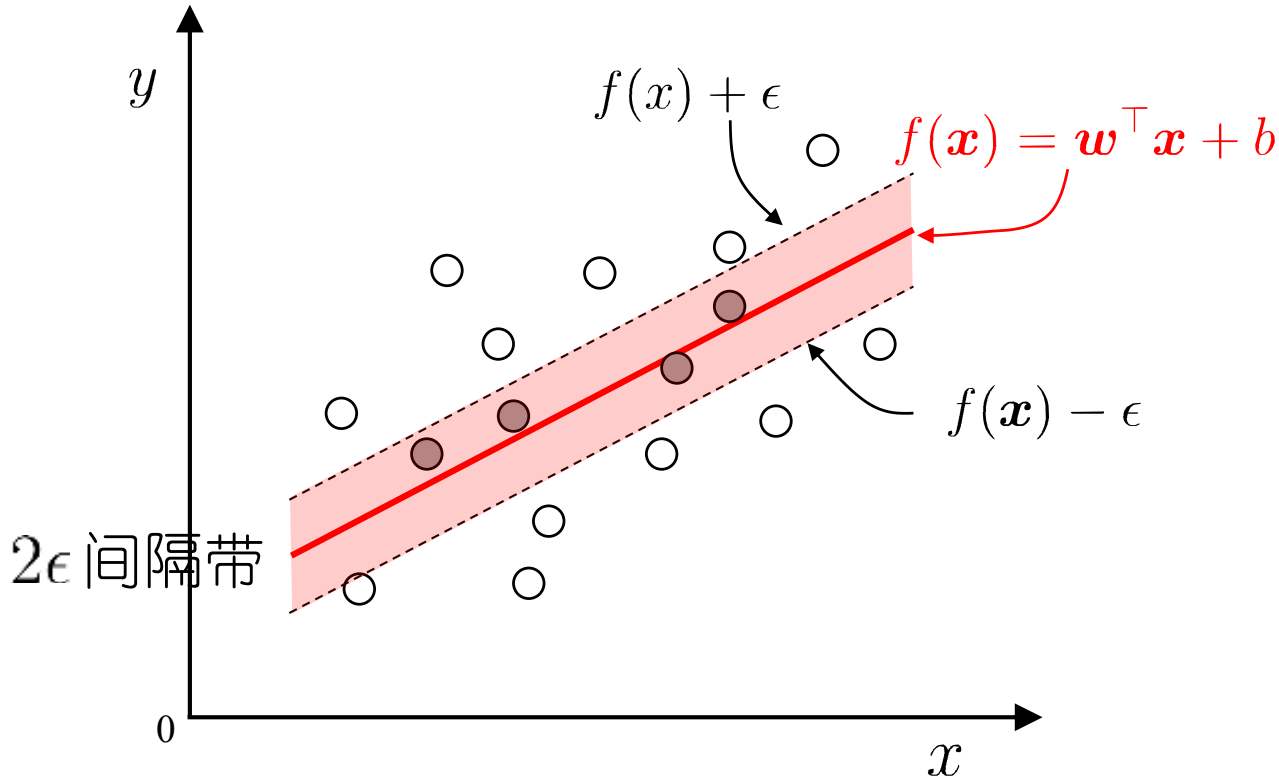
结构风险，描述模型的某些性质

经验风险，描述模型与训练数据的契合程度

- 通过替换上面两个部分，可以得到许多其他学习模型
  - 对数几率回归(Logistic Regression)
  - 最小绝对收缩选择算子(LASSO)
  - ……

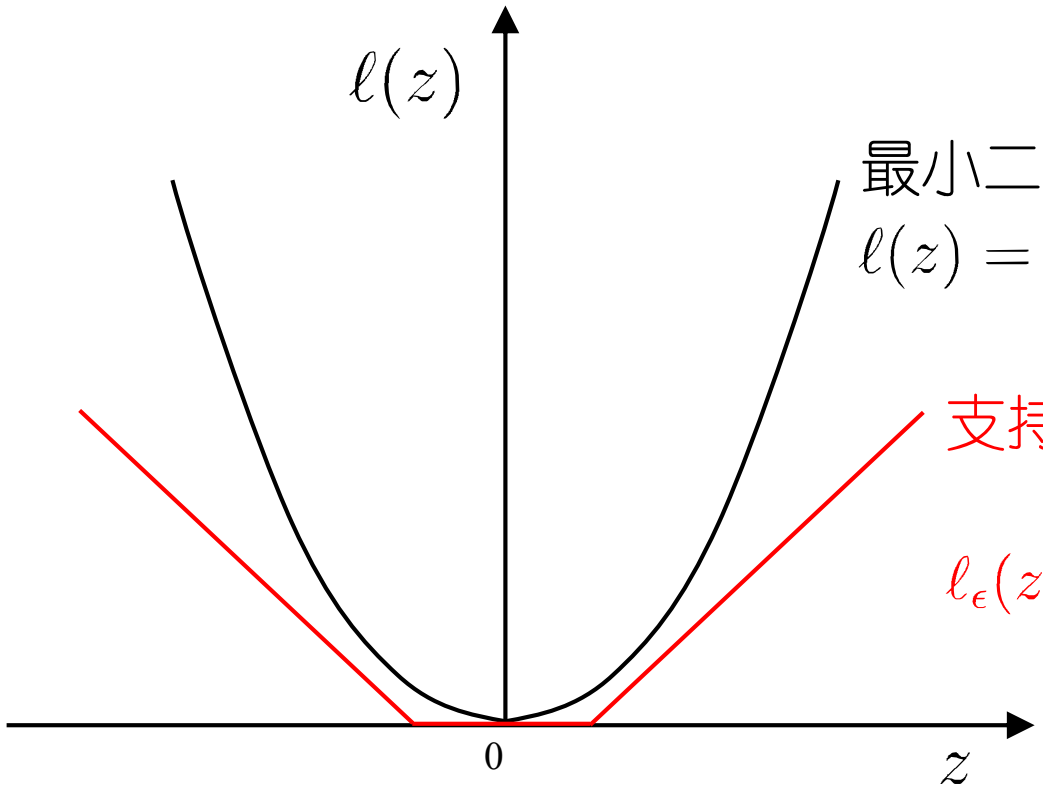
支持向量回归

- 允许模型输出和实际输出间存在 2ϵ的偏差.



损失函数

- 落入中间2ϵ 间隔带的样本不计算损失, 从而使得模型获得稀疏性



最小二乘损失函数

$$\ell(z) = z^2$$

支持向量回归损失函数

$$\ell_\epsilon(z) = \begin{cases} 0 & \text{if } |z| \leq \epsilon \\ |z| - \epsilon & \text{otherwise} \end{cases}$$

$$f(\boldsymbol{x}_i) = \boldsymbol{w}^\mathsf{T}\boldsymbol{x}_i + b$$

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \ell_\epsilon(f(\boldsymbol{x}_i) - y_i)$$

支持向量回归

原问题

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}(\xi_i+\hat{\xi}_i)$$

$$\text{s.t.} \quad f(\boldsymbol{x}_i) - y_i \le \epsilon + \xi_i,$$

$$y_i - f(\boldsymbol{x}_i) \le \epsilon + \hat{\xi}_i,$$

$$\xi_i \ge 0 \quad \hat{\xi}_i \ge 0, i = 1,2,\dots,m$$

对偶问题

$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i-\hat{\alpha}_i)(\alpha_j-\hat{\alpha}_j)\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) + \sum_{i=1}^{m}\left(y_i(\hat{\alpha}_i-\alpha_i) - \epsilon(\hat{\alpha}_i+\alpha_i)\right)$$

$$s.t.\ C \geqslant \boldsymbol{\alpha},\hat{\boldsymbol{\alpha}} \geqslant 0 \text{ and } \sum_{i=1}^{m}(\alpha_i-\hat{\alpha}_i) = 0$$

预测函数

$$y = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + b = \sum_{i}^{m}(\hat{\alpha}_i^{\star}-\alpha_i^{\star})y_i\kappa(\boldsymbol{x}_i,\boldsymbol{x}) + b^{\star}$$

核方法

- 支持向量机 $$y = \boldsymbol{w}^{\mathsf{T}}\phi(\boldsymbol{x}) + b = \sum_{i}^{m} \alpha_i^{\star} y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b^{\star}$$
- 支持向量回归 $$y = \boldsymbol{w}^{\mathsf{T}}\phi(\boldsymbol{x}) + b = \sum_{i}^{m} (\hat{\alpha}_i^{\star} - \alpha_i^{\star}) y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b^{\star}$$

无论是支持向量机还是支持向量回归, 学得的模型总可以表示成核函数的线性组合

更一般的结论（表示定理）：令$\mathbb{H}$为核函数κ对应的再生核希尔伯特空间，$\|h\|_{\mathbb{H}}$表示在$\mathbb{H}$空间中关于h的范数，对于任意单调增函数$\Omega\colon [0,\infty) \mapsto \mathbb{R}$和任意非负损失函数$\ell\colon \mathbb{R}^m \mapsto [0,\infty)$, 优化问题

$$\min_{\mathrm{h}\in H} F(h) = \Omega(\|h\|_{\mathbb{H}}) + \ell\big(h(\boldsymbol{x}_1), \cdots, h(\boldsymbol{x}_m)\big)$$

的解总可以写成$h^{\star} = \sum_i^m \alpha_i \kappa(\cdot, \boldsymbol{x}_i)$

核线性判别分析

- 通过表示定理可以得到很多线性模型的”核化”版本
  - 核SVM
  - 核LDA
  - 核PCA
  - ……
- 核LDA: 先将样本映射到高维特征空间, 然后在此特征空间中做线性判别分析

$$\boldsymbol{S}_b^{\phi} = \left(\boldsymbol{\mu}_1^{\phi} - \boldsymbol{\mu}_0^{\phi}\right)\left(\boldsymbol{\mu}_1^{\phi} - \boldsymbol{\mu}_0^{\phi}\right)^{\mathsf{T}}$$

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{\boldsymbol{w}^{\mathsf{T}}\boldsymbol{S}_b^{\phi}\boldsymbol{w}}{\boldsymbol{w}^{\mathsf{T}}\boldsymbol{S}_w^{\phi}\boldsymbol{w}} \qquad \boldsymbol{S}_w^{\phi} = \sum_{i=0}^{1}\sum_{x\in X_i}\left(\phi(\boldsymbol{x}) - \boldsymbol{\mu}_i^{\phi}\right)\left(\phi(\boldsymbol{x}) - \boldsymbol{\mu}_i^{\phi}\right)^{\mathsf{T}}$$

$$h(\boldsymbol{x}) = \boldsymbol{w}^{\mathsf{T}}\phi(\boldsymbol{x}) = \sum_{i}^{m}\alpha_i\kappa(\boldsymbol{x}_i, \boldsymbol{x})$$

$$\max_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{\alpha}}{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{N}\boldsymbol{\alpha}}$$

$$\boldsymbol{M} = (\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_0)(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_0)^{\mathsf{T}}$$

$$\boldsymbol{N} = \boldsymbol{K}\boldsymbol{K}^{\mathsf{T}} - \sum_{i=0}^{1} m_i\hat{\boldsymbol{\mu}}_i\hat{\boldsymbol{\mu}}_i^{\mathsf{T}}$$

核线性判别分析

由表示定理可得 $\boldsymbol{w}=\sum_{i=1}^{m}\alpha_i\phi(\boldsymbol{x}_i)$ $\boldsymbol{\mu}_i^{\phi}=\frac{1}{m_i}\sum_{\boldsymbol{x}\in X_i}\phi(\boldsymbol{x})$

$$\boldsymbol{S}_b^{\phi}=\left(\boldsymbol{\mu}_1^{\phi}-\boldsymbol{\mu}_0^{\phi}\right)\left(\boldsymbol{\mu}_1^{\phi}-\boldsymbol{\mu}_0^{\phi}\right)^{\mathsf{T}}$$

$$\boldsymbol{w}^{\mathsf{T}}\phi(\boldsymbol{x})=\sum_{i=1}^{m}\alpha_i\kappa(\boldsymbol{x}_i,\boldsymbol{x})=\boldsymbol{\alpha}^{\mathsf{T}}K(:,\boldsymbol{x})$$

$$\boldsymbol{w}^{\mathsf{T}}\boldsymbol{\mu}_i^{\phi}=\frac{1}{m_i}\sum_{\boldsymbol{x}\in X_i}\boldsymbol{w}^{\mathsf{T}}\phi(\boldsymbol{x})=\frac{1}{m_i}\sum_{\boldsymbol{x}\in X_i}\boldsymbol{\alpha}^{\mathsf{T}}K(:,\boldsymbol{x})=\boldsymbol{\alpha}^{\mathsf{T}}\underbrace{\frac{1}{m_i}\boldsymbol{K}\boldsymbol{1}_i}_{\widehat{\boldsymbol{\mu}}_i}$$

$$\boldsymbol{w}^{\mathsf{T}}\boldsymbol{S}_b^{\phi}\boldsymbol{w}=\boldsymbol{w}^{\mathsf{T}}\left(\boldsymbol{\mu}_1^{\phi}-\boldsymbol{\mu}_0^{\phi}\right)\left(\boldsymbol{\mu}_1^{\phi}-\boldsymbol{\mu}_0^{\phi}\right)^{\mathsf{T}}\boldsymbol{w}=\boldsymbol{\alpha}^{\mathsf{T}}\underbrace{(\widehat{\boldsymbol{\mu}}_1-\widehat{\boldsymbol{\mu}}_0)(\widehat{\boldsymbol{\mu}}_1-\widehat{\boldsymbol{\mu}}_0)^{\mathsf{T}}}_{M}\boldsymbol{\alpha}$$

核线性判别分析

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i \phi(\boldsymbol{x}_i) \qquad \boldsymbol{w}^{\mathsf{T}} \phi(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) = \boldsymbol{\alpha}^{\mathsf{T}} K(:, \boldsymbol{x})$$

$$\boldsymbol{w}^{\mathsf{T}} \boldsymbol{S}_w^{\phi} \boldsymbol{w} = \sum_{i=0}^{1} \sum_{x \in X_i} \boldsymbol{w}^{\mathsf{T}} \left(\phi(\boldsymbol{x}) - \boldsymbol{\mu}_i^{\phi} \right) \left(\phi(\boldsymbol{x}) - \boldsymbol{\mu}_i^{\phi} \right)^{\mathsf{T}} \boldsymbol{w}$$

$$= \sum_{i=1}^{m} \boldsymbol{w}^{\mathsf{T}} \phi(\boldsymbol{x}_i) \phi(\boldsymbol{x}_i)^{\mathsf{T}} \boldsymbol{w} - \sum_{i=0}^{1} m_i \boldsymbol{w}^{\mathsf{T}} \boldsymbol{\mu}_i^{\phi} \left(\boldsymbol{\mu}_i^{\phi} \right)^{\mathsf{T}} \boldsymbol{w}$$

$$= \boldsymbol{\alpha}^{\mathsf{T}} \left(\boldsymbol{K} \boldsymbol{K}^{\mathsf{T}} - \sum_{i=0}^{1} m_i \hat{\boldsymbol{\mu}}_i \hat{\boldsymbol{\mu}}_i^{\mathsf{T}} \right) \boldsymbol{\alpha}$$

$$\sum_{i=1}^{m} \boldsymbol{w}^{\mathsf{T}} \phi(\boldsymbol{x}_i) \phi(\boldsymbol{x}_i)^{\mathsf{T}} \boldsymbol{w} = \sum_{i=1}^{m} \boldsymbol{\alpha}^{\mathsf{T}} K(:, \boldsymbol{x}_i) K(:, \boldsymbol{x}_i)^{\mathsf{T}} \boldsymbol{\alpha} = \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{K} \boldsymbol{K}^{\mathsf{T}} \boldsymbol{\alpha}$$

$$\boldsymbol{w}^{\mathsf{T}} \boldsymbol{\mu}_i^{\phi} = \frac{1}{m_i} \sum_{\boldsymbol{x} \in X_i} \boldsymbol{w}^{\mathsf{T}} \phi(\boldsymbol{x}) = \frac{1}{m_i} \sum_{x \in X_i} \boldsymbol{\alpha}^{\mathsf{T}} K(:, \boldsymbol{x}) = \boldsymbol{\alpha}^{\mathsf{T}} \underbrace{\frac{1}{m_i}}_{\hat{\boldsymbol{\mu}}_i} \boldsymbol{K} \boldsymbol{1}_i$$

$$\sum_{i=0}^{1} m_i \boldsymbol{w}^{\mathsf{T}} \boldsymbol{\mu}_i^{\phi} \left(\boldsymbol{\mu}_i^{\phi} \right)^{\mathsf{T}} \boldsymbol{w} = \sum_{i=0}^{1} m_i \boldsymbol{\alpha}^{\mathsf{T}} \hat{\boldsymbol{\mu}}_i \hat{\boldsymbol{\mu}}_i^{\mathsf{T}} \boldsymbol{\alpha} = \boldsymbol{\alpha}^{\mathsf{T}} \left(\sum_{i=0}^{1} m_i \hat{\boldsymbol{\mu}}_i \hat{\boldsymbol{\mu}}_i^{\mathsf{T}} \right) \boldsymbol{\alpha}$$

作业

- 6.4
- 6.6
- 6.9

若$\kappa(x_i, x_j) = \phi(x_i)^T \phi(x_j) = (x_i^T x_j)^2$，求$\phi(x_i)$表达式。

支持向量回归的对偶问题如下，

$$\max_{\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}} g(\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i - \hat{\alpha}_i)(\alpha_j - \hat{\alpha}_j)\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{m}\left(y_i(\hat{\alpha}_i - \alpha_i) - \epsilon(\hat{\alpha}_i + \alpha_i)\right)$$

$$s.t.\ C \succcurlyeq \boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}} \succcurlyeq 0 \text{ and } \textstyle\sum_{i=1}^{m}(\alpha_i - \hat{\alpha}_i) = 0$$

请将该问题转化为类似于如下标准型的形式（$\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{K}$均已知），

$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{v} - \frac{1}{2}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{K}\boldsymbol{\alpha}$$

$$s.t.\ C \succcurlyeq \boldsymbol{\alpha} \succcurlyeq 0 \text{ and } \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{u} = 0$$

例如在软间隔SVM中$\boldsymbol{v} = \mathbf{1}, \boldsymbol{u} = \boldsymbol{y}, \boldsymbol{K}[i,j] = y_i y_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

第七章：贝叶斯网

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

贝叶斯决策论

- 贝叶斯决策论（Bayesian decision theory）是在概率框架下实施决策的基本方法。
  - 在分类问题情况下，在所有相关概率都已知的理想情形下，贝叶斯决策考虑如何基于这些概率和误判损失来选择最优的类别标记。

- 假设有N种可能的类别标记，即$\mathcal{Y} = \{c_1, \cdots, c_N\}$，$\lambda_{ij}$是将一个真实标记为$c_i$的样本误分类为$c_j$所产生的损失。基于后验概率$P(c_i|\boldsymbol{x})$可获得将样本$\boldsymbol{x}$分类为$c_i$所产生的期望损失（expected loss），即在样本上的“条件风险”（conditional risk）

$$R(c_i|\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_{ij} P(c_j|\boldsymbol{x})$$

- 贝叶斯决策论的任务是寻找一个判定准则 $h: X \mapsto Y$以最小化总体风险

$$R(h) = \mathbb{E}_x[R(h(\boldsymbol{x})|\boldsymbol{x})]$$

贝叶斯决策论

- 寻找一个判定准则 $h: X \mapsto Y$以最小化总体风险

$$R(h) = \mathbb{E}_x[R(h(\boldsymbol{x})|\boldsymbol{x})] \qquad R(c_i|\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_{ij} P(c_j|\boldsymbol{x})$$

- 显然，对每个样本$\boldsymbol{x}$，若$\boldsymbol{x}$能最小化条件风险$R(h(\boldsymbol{x})|\boldsymbol{x})$，则总体风险$R(h)$ 也将被最小化

- 这就产生了贝叶斯判定准则（Bayes decision rule）： 为最小化总体风险，只需在每个样本上选择那个能使条件风险 $R(c|\boldsymbol{x})$最小的类别标记，即

$$h^\star(\boldsymbol{x}) = \arg\min_{c \in \mathcal{Y}} R(c|\boldsymbol{x})$$

  - 被称为贝叶斯最优分类器(Bayes optimal classifier)，与之对应的总体风险 $R(h^\star)$称为贝叶斯风险 (Bayes risk)
  - 反映了分类起所能达到的最好性能，即通过机器学习所能产生的模型精度的理论上限。

贝叶斯决策论

- 若目标是最小化分类错误率，则误判损失λ_{ij}可写为

$$\lambda_{ij} = \begin{cases} 0, \text{if } i = j \\ 1, \text{otherwise} \end{cases}$$

- 此时条件风险

$$R(c_i|\boldsymbol{x}) = \sum_{j=1}^{N} \lambda_{ij} P(c_j|\boldsymbol{x}) = \sum_{j \neq i} P(c_j|\boldsymbol{x}) = 1 - P(c_i|\boldsymbol{x})$$

- 最小化分类错误率的贝叶斯最有分类器为

$$h^{\star}(\boldsymbol{x}) = \arg\min_{c \in \mathcal{Y}} R(c|\boldsymbol{x}) = \arg\max_{c \in \mathcal{Y}} P(c|\boldsymbol{x})$$

  - 即对每个样本$\boldsymbol{x}$，选择能使后验概率 $P(c|\boldsymbol{x})$最大的类别标记。

贝叶斯决策论

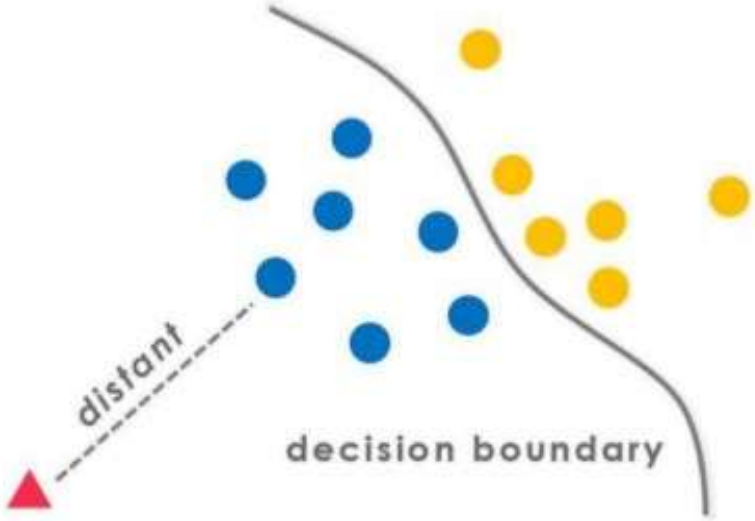
- 最小化分类错误率的贝叶斯最有分类器为

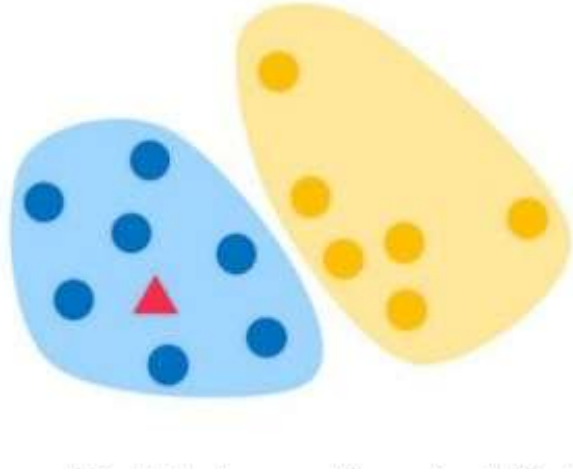
$$h^{\star}(\boldsymbol{x}) = \arg\min_{c\in\mathcal{Y}} R(c|\boldsymbol{x}) = \arg\max_{c\in\mathcal{Y}} P(c|\boldsymbol{x})$$

- 不难看出，使用贝叶斯判定准则来最小化决策风险，首先要获得后验概率$P(c|\boldsymbol{x})$
- 然而，在现实中通常难以直接获得。机器学习所要实现的是基于有限的训练样本尽可能准确地估计出后验概率 $P(c|\boldsymbol{x})$。

贝叶斯决策论

有监督学习目标：学习决策函数 $c = f(X)$ 或者条件概率 $P(c|X)$





- 判别模型
  - 直接学习$f(\boldsymbol{x})$或者$P(c|\boldsymbol{x})$
  - 直接面对预测，准确率较高，学习简单
  - 典型模型包括对率回归、SVM等

- 生成模型
  - 建模联合概率分布$P(\boldsymbol{x}, c)$，求出条件概率进行预测

$$P(c|\boldsymbol{x}) = \frac{P(\boldsymbol{x}, c)}{P(\boldsymbol{x})}$$

  - 典型模型包括朴素贝叶斯

贝叶斯决策论

- 生成式模型

$$P(c|\boldsymbol{x}) = \frac{P(\boldsymbol{x}, c)}{P(\boldsymbol{x})}$$

- 基于贝叶斯定理，$P(c|\boldsymbol{x})$可写成

$$P(c|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|c)P(c)}{P(\boldsymbol{x})}$$

先验概率
样本空间中各类样本所占的比例，可通过各类样本出现的频率估计（大数定理）

类标记c相对于样本$\boldsymbol{x}$的“类条件概率”（class-conditional probability），或称“似然”。

“证据”（evidence）因子，与类标记无关

极大似然估计

- 估计类条件概率的常用策略：先假定其具有某种确定的概率分布形式，再基于训练样本对概率分布参数估计。

- 记关于类别c的类条件概率为$P(\boldsymbol{x}|c)$，
  - 假设$P(\boldsymbol{x}|c)$具有确定的形式被参数$\boldsymbol{\theta}_c$唯一确定，我们的任务就是利用训练集D估计参数$\boldsymbol{\theta}_c$

- 概率模型的训练过程就是参数估计过程，统计学界的两个学派提供了不同的方案：
  - 频率主义学派 (frequentist)认为参数虽然未知，但却存在客观值，因此可通过优化似然函数等准则来确定参数值

  - 贝叶斯学派 (Bayesian)认为参数是未观察到的随机变量、其本身也可由分布，因此可假定参数服从一个先验分布，然后基于观测到的数据计算参数的后验分布。

极大似然估计

- 令D_c表示训练集中第c类样本的组合的集合，假设这些样本是独立的，则参数$\boldsymbol{\theta}_c$对于数据集D_c的似然是

$$P(D_c|\boldsymbol{\theta}_c) = \prod_{\boldsymbol{x}\in D_c} P(\boldsymbol{x}|\boldsymbol{\theta}_c)$$

- 对 $\boldsymbol{\theta}_c$进行极大似然估计，寻找能最大化似然$P(D_c|\boldsymbol{\theta}_c)$的参数值$\hat{\boldsymbol{\theta}}_c$。直观上看，极大似然估计是试图在$\boldsymbol{\theta}_c$所有可能的取值中，找到一个使数据出现的“可能性”最大值。
- 连乘操作易造成下溢，通常使用对数似然(log-likelihood)

$$LL(\theta_c) = \log P(D_c|\theta_c) = \sum_{x\in D_c} \log P(\boldsymbol{x}_c|\theta_c)$$

- 此时参数 $\boldsymbol{\theta}_c$的极大似然估计$\hat{\boldsymbol{\theta}}_c$为

$$\hat{\boldsymbol{\theta}}_c = \arg\max_{\boldsymbol{\theta}_c} LL(\boldsymbol{\theta}_c)$$

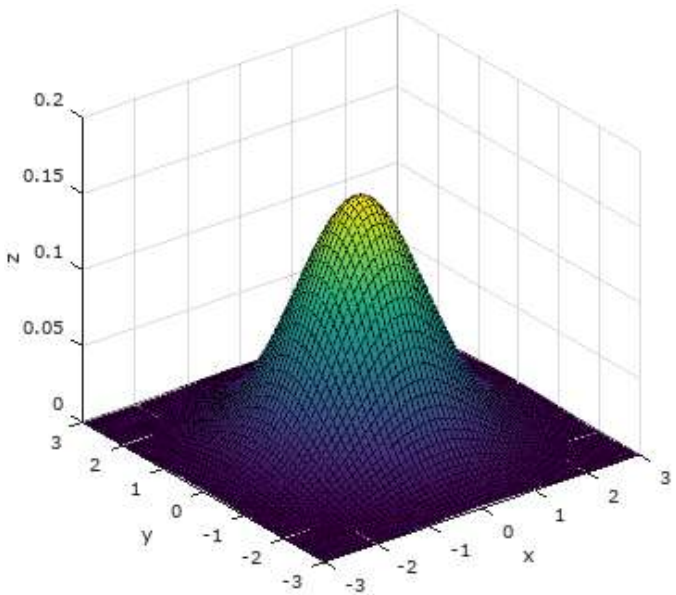
极大似然估计

- 在连续属性情形下，假设概率密度函数$P(\boldsymbol{x}|c)\sim\mathcal{N}(\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c)$，

  则参数$\boldsymbol{\mu}_c$和$\boldsymbol{\Sigma}_c$的极大似然估计为

$$\hat{\boldsymbol{\mu}}_c=\frac{1}{|D_c|}\sum_{\boldsymbol{x}\in D_c}\boldsymbol{x}$$

$$\hat{\boldsymbol{\Sigma}}_c=\frac{1}{|D_c|}\sum_{\boldsymbol{x}\in D_c}(\boldsymbol{x}-\boldsymbol{\mu}_c)(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\mathsf{T}}$$



$$P(\boldsymbol{x}|\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c)=\frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}_c|}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\mathsf{T}}\boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)\right)$$

极大似然估计

$$P(\boldsymbol{x}|\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c)=\frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}_c|}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\top}\boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)\right)$$

$$LL(\boldsymbol{\theta}_c)=\sum_{\boldsymbol{x}\in D_c}\log P(\boldsymbol{x}|\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c)$$

$$=-\sum_{\boldsymbol{x}\in D_c}\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\top}\boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)-\frac{|D_c|}{2}\log|\boldsymbol{\Sigma}_c|-\frac{d|D_c|}{2}\log(2\pi)$$

$$\frac{\partial LL(\boldsymbol{\theta}_c)}{\partial\boldsymbol{\mu}_c}=\sum_{\boldsymbol{x}\in D_c}\boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)=0 \quad\Rightarrow\quad \hat{\boldsymbol{\mu}}_c=\frac{1}{|D_c|}\sum_{\boldsymbol{x}\in D_c}\boldsymbol{x}$$

$$\frac{\partial LL(\boldsymbol{\theta}_c)}{\partial\boldsymbol{\Sigma}_c}=-\frac{\partial}{\partial\boldsymbol{\Sigma}_c}\sum_{\boldsymbol{x}\in D_c}\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\top}\boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c)-\frac{|D_c|}{2}\frac{\partial}{\partial\boldsymbol{\Sigma}_c}\log|\boldsymbol{\Sigma}_c|$$

极大似然估计

$$\hat{\boldsymbol{\Sigma}}_c = \frac{1}{|D_c|}\sum_{\boldsymbol{x}\in D_c}(\boldsymbol{x}-\boldsymbol{\mu}_c)(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\mathsf{T}}$$

$$\frac{\partial LL(\boldsymbol{\theta}_c)}{\partial \boldsymbol{\Sigma}_c} = -\frac{\partial}{\partial \boldsymbol{\Sigma}_c}\sum_{\boldsymbol{x}\in D_c}\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\mathsf{T}}\boldsymbol{\Sigma}_c^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_c) - \frac{|D_c|}{2}\frac{\partial}{\partial \boldsymbol{\Sigma}_c}\log|\boldsymbol{\Sigma}_c|$$

$$= -\frac{|D_c|}{2}\frac{\partial}{\partial \boldsymbol{\Sigma}_c}\mathrm{tr}\left(\boldsymbol{\Sigma}_c^{-1}\hat{\boldsymbol{\Sigma}}_c\right) - \frac{|D_c|}{2}\frac{\partial}{\partial \boldsymbol{\Sigma}_c}\log|\boldsymbol{\Sigma}_c| = \frac{|D_c|}{2}\boldsymbol{\Sigma}_c^{-1}\hat{\boldsymbol{\Sigma}}_c\boldsymbol{\Sigma}_c^{-1} - \frac{|D_c|}{2}\boldsymbol{\Sigma}_c^{-1}$$

$$\left[\frac{\partial}{\partial \boldsymbol{\Sigma}_c}\mathrm{tr}\left(\boldsymbol{\Sigma}_c^{-1}\hat{\boldsymbol{\Sigma}}_c\right)\right]_{ij} = \frac{\partial}{\partial \Sigma_{ij}^c}\mathrm{tr}\left(\boldsymbol{\Sigma}_c^{-1}\hat{\boldsymbol{\Sigma}}_c\right)$$

$$= \mathrm{tr}\left(\frac{\partial}{\partial \Sigma_{ij}^c}\boldsymbol{\Sigma}_c^{-1}\hat{\boldsymbol{\Sigma}}_c\right)$$

$$= -\mathrm{tr}\left(\boldsymbol{\Sigma}_c^{-1}\frac{\partial \boldsymbol{\Sigma}_c}{\partial \Sigma_{ij}^c}\boldsymbol{\Sigma}_c^{-1}\hat{\boldsymbol{\Sigma}}_c\right)$$

$$= -\mathrm{tr}\left(\frac{\partial \boldsymbol{\Sigma}_c}{\partial \Sigma_{ij}^c}\boldsymbol{\Sigma}_c^{-1}\hat{\boldsymbol{\Sigma}}_c\boldsymbol{\Sigma}_c^{-1}\right)$$

$$= -\left[\boldsymbol{\Sigma}_c^{-1}\hat{\boldsymbol{\Sigma}}_c\boldsymbol{\Sigma}_c^{-1}\right]_{ij}$$

$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$

$$\frac{\partial \boldsymbol{A}^{-1}\boldsymbol{A}}{\partial x} = \frac{\partial \boldsymbol{A}^{-1}}{\partial x}\boldsymbol{A} + \boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x} = \boldsymbol{0}$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_c}\log|\boldsymbol{\Sigma}_c| = (\boldsymbol{\Sigma}_c^{-1})^{\mathsf{T}} = \boldsymbol{\Sigma}_c^{-1}$$

$$\frac{\partial LL(\boldsymbol{\theta}_c)}{\partial \boldsymbol{\Sigma}_c} = 0 \Rightarrow \hat{\boldsymbol{\Sigma}}_c\boldsymbol{\Sigma}_c^{-1} = I$$

$$\boldsymbol{\Sigma}_c = \hat{\boldsymbol{\Sigma}}_c$$

极大似然估计

- 在连续属性情形下，假设概率密度函数$P(\boldsymbol{x}|c)\sim\mathcal{N}(\boldsymbol{\mu}_c,\boldsymbol{\Sigma}_c)$，则参数$\boldsymbol{\mu}_c$和$\boldsymbol{\Sigma}_c$的极大似然估计为

$$\hat{\boldsymbol{\mu}}_c=\frac{1}{|D_c|}\sum_{\boldsymbol{x}\in D_c}\boldsymbol{x}$$

$$\hat{\boldsymbol{\Sigma}}_c=\frac{1}{|D_c|}\sum_{\boldsymbol{x}\in D_c}(\boldsymbol{x}-\boldsymbol{\mu}_c)(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\mathrm{T}}$$

- 通过极大似然法得到的正态分布均值就是样本均值，方差就是$(\boldsymbol{x}-\boldsymbol{\mu}_c)(\boldsymbol{x}-\boldsymbol{\mu}_c)^{\mathrm{T}}$的均值，这显然是一个符合直觉的结果。

- 需注意的是，这种参数化的方法虽能使类条件概率估计变得相对简单，但估计结果的准确性严重依赖于所假设的概率分布形式是否符合潜在的真实数据分布。

朴素贝叶斯分类器

- 估计后验概率$P(c|\boldsymbol{x})$主要困难：类条件概率$P(\boldsymbol{x}|c)$是所有属性上的联合概率难以从有限的训练样本估计获得。

- 朴素贝叶斯分类器(Naïve Bayes Classifier)采用了“属性条件独立性假设”(attribute conditional independence assumption)：每个属性独立地对分类结果发生影响。

- 基于属性条件独立性假设，

$$P(c|\boldsymbol{x}) = \frac{P(c)P(\boldsymbol{x}|c)}{P(\boldsymbol{x})} = \frac{P(c)}{P(\boldsymbol{x})}\prod_{i=1}^{d} P(x_i|c)$$

  - 其中d为属性数目，x_i为$\boldsymbol{x}$在第i个属性上的取值。

朴素贝叶斯分类器

$$P(c|\boldsymbol{x}) = \frac{P(c)P(\boldsymbol{x}|c)}{P(\boldsymbol{x})} = \frac{P(c)}{P(\boldsymbol{x})}\prod_{i=1}^{d} P(x_i|c)$$

由于对所有类别来说 $P(\boldsymbol{x})$相同，因此基于贝叶斯判定准则有

$$h_{nb}(\boldsymbol{x}) = \arg\max_{c\in\mathcal{Y}} P(c)\prod_{i=1}^{d} P(x_i|c)$$

这就是朴素贝叶斯分类的表达式子

朴素贝叶斯分类器

- 朴素贝叶斯分类器的训练器的训练过程就是基于训练集D
  - 估计类先验概率$P(c)$

令 D_c表示训练集D中第c类样本组成的集合，若有充足的独立同分布样本，则可容易地估计出类先验概率

$$P(c) = \frac{|D_c|}{|D|}$$

  - 为每个属性估计条件概率$P(x_i|c)$

对离散属性，令D_{c,x_i}表示D中在第i个属性上取值为x_i的样本组成的集合，则条件概率 $P(x_i|c)$可估计为

$$P(x_i|c) = \frac{|D_{c,x_i}|}{|D_c|}$$

对连续属性，设$p(x_i|c) = \mathcal{N}\left(x_i|\mu_{c,i}, \sigma_{c,i}^2\right)$其中 $\mu_{c,i}$和$\sigma_{c,i}^2$分别是第c类样本在第i个属性上取值的均值和方差

$$P(x_i|c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i-\mu_{c,i})^2}{2\sigma_{c,i}^2}\right)$$

朴素贝叶斯分类器

- 用西瓜数据集3.0训练朴素贝叶斯分类器，对测试例“测1”进行分类

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
测 1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	?

朴素贝叶斯分类器

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
测 1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	?

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

- 估计类先验概率$P(c)$,

$$P(\text{好瓜} = \text{是}) = \frac{8}{17}$$

- 为每个属性估计条件概率

$$P(\text{色泽} = \text{青绿}|\text{好瓜} = \text{是}) = 3/8$$

$$P(\text{色泽} = \text{青绿}|\text{好瓜} = \text{否}) = 3/9$$

$$P(\text{根蒂} = \text{蜷缩}|\text{好瓜} = \text{是}) = 5/8$$

$$P(\text{根蒂} = \text{蜷缩}|\text{好瓜} = \text{否}) = 3/9$$

$$P(\text{敲声} = \text{浊响}|\text{好瓜} = \text{是}) = 6/8$$

$$P(\text{敲声} = \text{浊响}|\text{好瓜} = \text{否}) = 4/9$$

$$P(\text{纹理} = \text{清晰}|\text{好瓜} = \text{是}) = 7/8$$

$$P(\text{纹理} = \text{清晰}|\text{好瓜} = \text{否}) = 2/9$$

$$P(\text{脐部} = \text{凹陷}|\text{好瓜} = \text{是}) = 5/8$$

$$P(\text{脐部} = \text{凹陷}|\text{好瓜} = \text{否}) = 2/9$$

$$P(\text{密度} = 0.697|\text{好瓜} = \text{是}) = \frac{1}{\sqrt{2\pi} \times 0.129} \exp\left(-\frac{(0.697 - 0.574)^2}{2 \times 0.129^2}\right) \approx 1.959$$

$$P(\text{密度} = 0.697|\text{好瓜} = \text{否}) = \frac{1}{\sqrt{2\pi} \times 0.195} \exp\left(-\frac{(0.697 - 0.496)^2}{2 \times 0.195^2}\right) \approx 1.203$$

朴素贝叶斯分类器

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
测 1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	?

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.460	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.360	0.370	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

- 估计类先验概率$P(c)$,

$$P(\text{好瓜}=\text{是})=\frac{8}{17}$$

属性估计条件概率

$P(\text{青绿}|\text{好瓜}=\text{是})=3/8$

$P(\text{青绿}|\text{好瓜}=\text{否})=3/9$

$P(\text{蜷缩}|\text{好瓜}=\text{是})=5/8$

$P(\text{蜷缩}|\text{好瓜}=\text{否})=3/9$

$P(\text{浊响}|\text{好瓜}=\text{是})=6/8$

$P(\text{敲声}=\text{浊响}|\text{好瓜}=\text{否})=4/9$

$P(\text{纹理}=\text{清晰}|\text{好瓜}=\text{是})=7/8$

$P(\text{清晰}|\text{好瓜}=\text{否})=2/9$

$P(\text{脐部}=\text{凹陷}|\text{好瓜}=\text{是})=5/8$

$P(\text{脐部}=\text{凹陷}|\text{好瓜}=\text{否})=2/9$

$$P(\text{好瓜}=\text{是}|\boldsymbol{x})\propto P(\text{好瓜}=\text{是})\times P_{\text{青绿}|\text{是}}\times P_{\text{蜷缩}|\text{是}}\times P_{\text{浊响}|\text{是}}\times P_{\text{清晰}|\text{是}}\times P_{\text{凹陷}|\text{是}}\times P_{\text{硬滑}|\text{是}}\approx 0.052$$

$$P(\text{好瓜}=\text{否}|\boldsymbol{x})\propto P(\text{好瓜}=\text{否})\times P_{\text{青绿}|\text{否}}\times P_{\text{蜷缩}|\text{否}}\times P_{\text{浊响}|\text{否}}\times P_{\text{清晰}|\text{否}}\times P_{\text{凹陷}|\text{否}}\times P_{\text{硬滑}|\text{否}}\approx 6.80\times 10^{-5}$$

$$P(\text{密度}=0.697|\text{好瓜}=\text{是})=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(0.697-0.574)^2}{\cdots}\right)\approx 1.959$$

测试例“测1”判别为好瓜

$$P(\text{密度}=0.697|\text{好瓜}=\text{是})=\frac{1}{\sqrt{2\pi}\times 0.195}\exp\left(-\frac{(0.697-0.496)^2}{2\times 0.195^2}\right)\approx 1.203$$

拉普拉斯修正

- 若某个属性值在训练集中没有与某个类同时出现过，则直接计算会出现问题。
  - 比如“敲声=清脆”测试例，训练集中没有该样例，因此连乘式计算的概率值为0，无论其他属性上明显像好瓜，分类结果都是“好瓜=否”。

这显然不合理

- 为了避免其他属性携带的信息被训练集中未出现的属性值“抹去”，在估计概率值时通常要进行“拉普拉斯修正”
  - 令N表示训练集D中可能的类别数，N_i表示第i个属性可能的取值数

$$P(c)=\frac{|D_c|}{|D|} \Rightarrow \frac{|D_c|+1}{|D|+N} \qquad P(x_i|c)=\frac{|D_{c,x_i}|}{|D_c|} \Rightarrow \frac{|D_{c,x_i}|+1}{|D_c|+N_i}$$

- 现实任务中，朴素贝叶斯分类器的使用：
  - 速度要求高，“查表”；
  - 任务数据更替频繁，“懒惰学习” (lazy learning)；
  - 数据不断增加，增量学习。

半朴素贝叶斯分类器

- 为了降低贝叶斯公式中估计后验概率的困难，朴素贝叶斯分类器采用的属性条件独立性假设；对属性条件独立假设记性一定程度的放松，由此产生了一类称为“半朴素贝叶斯分类器” (semi-naïve Bayes classifiers)
- 半朴素贝叶斯分类器最常用的一种策略： “独依赖估计”(One-Dependent Estimator,简称ODE)，假设每个属性在类别之外最多仅依赖一个其他属性，即

$$P(c|\boldsymbol{x}) \propto P(c)\prod_{i}^{d} P(x_i|c,pa_i)$$

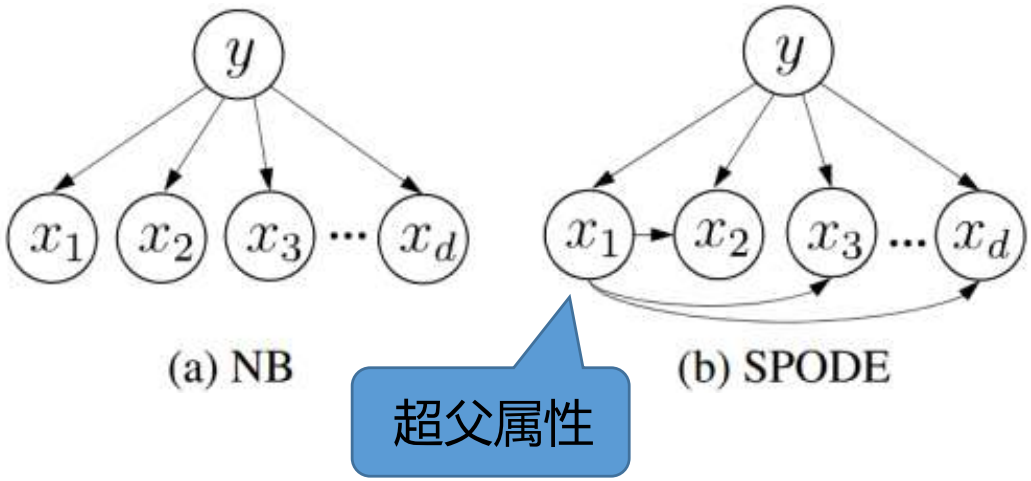
  - 其中pa_i为属性x_i所依赖的属性，称为x_i的父属性

- 对每个属性x_i，若其父属性 pa_i 已知，则可估计概值$P(x_i|c,pa_i)$，

问题的关键转化为如何确定每个属性的父属性

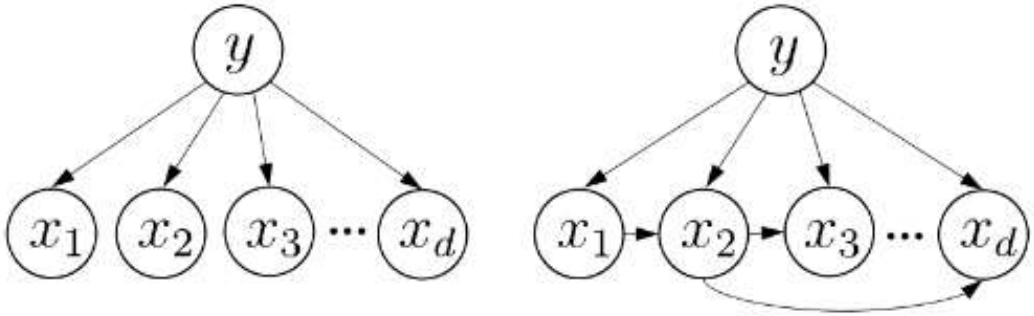
半朴素贝叶斯分类器

- 最直接的做法是假设所有属性都依赖于同一属性，称为“超父” (super-parenet)，然后通过交叉验证等模型选择方法来确定超父属性，由此形成了SPODE (Super-Parent ODE)方法。



半朴素贝叶斯分类器

- TAN (Tree augmented Naïve Bayes) [Friedman et al., 1997] 则在最大带权生成树 (Maximum weighted spanning tree) 算法 [Chow and Liu, 1968] 的基础上，通过以下步骤将属性间依赖关系简约为



  - 计算任意两个属性之间的条件互信息

$$I(x_i, x_j|y) = \sum_{x_i,x_j;c\in\mathcal{Y}} P(x_i, x_j, c) \log \frac{P(x_i, x_j|c)}{P(x_i|c)P(x_i|c)}$$

  - 以属性为结点构建完全图，任意两个结点之间边的权重设为互信息
  - 构建此完全图的最大带权生成树，挑选根变量，将边设为有向；
  - 加入类别节点y，增加从y到每个属性的有向边。

半朴素贝叶斯分类器

- AODE (Averaged One-Dependent Estimator) [Webb et al. 2005] 是一种基于集成学习机制、更为强大的分类器。
  - 尝试将每个属性作为超父构建 SPODE
  - 将具有足够训练数据支撑的SPODE集群起来作为最终结果

$$P(c|\boldsymbol{x}) \propto \sum_{i;\left|D_{x_i}\right|\geq m'}^{d} P(c,x_i)\prod_{j}^{d} P\left(x_j|c,x_i\right)$$

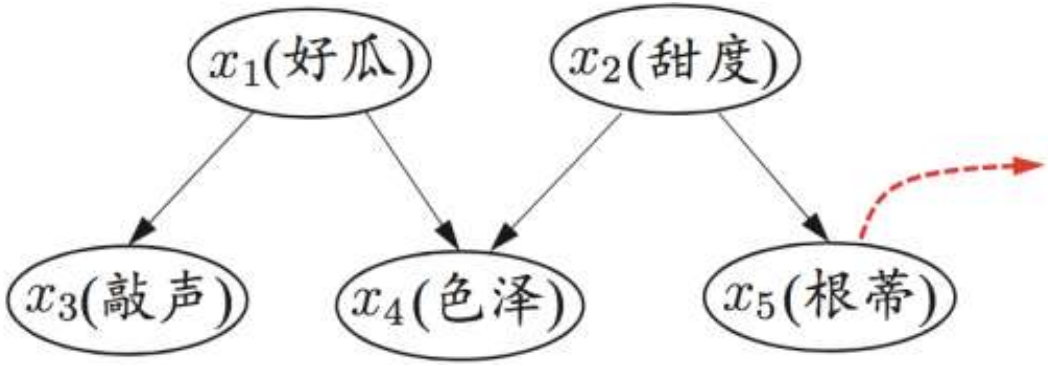
其中，D_{x_i}是在第i个属性上取值x_i的样本的集合，m'为阈值常数

$$P(x_i,c)=\frac{\left|D_{c,x_i}\right|+1}{|D|+N\times N_i} \qquad P\left(x_j|c,x_i\right)=\frac{\left|D_{c,x_i,x_j}\right|+1}{\left|D_{c,x_i}\right|+N_j}$$

N_i是在第i个属性上取值数，D_{c,x_i}是类别为 c且在第i个属性上取值为 x_i 的样本集合，D_{c,x_i,x_j}是类别为 c且在第i和第j个属性上取值分别为 x_i和x_j的样本集合

贝叶斯网

- 贝叶斯网 (Bayesian network)亦称“信念网”(brief network), 它借助有向无环图 (Directed Acyclic Graph, DAG)来刻画属性间的依赖关系，并使用条件概率表 (Conditional Probability Table, CPT)来表述属性的联合概率分布。



根蒂

甜度	硬挺	蜷缩
高	0.1	0.9
低	0.7	0.3

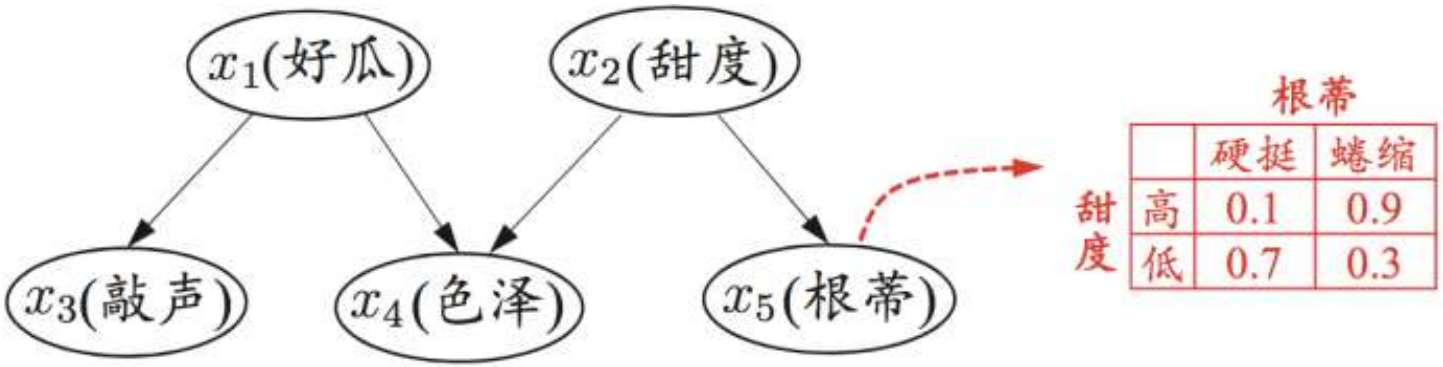
西瓜问题的一种贝叶斯网结构以及属性“根蒂”的条件概率表

- 从网络图结构可以看出 -> “色泽”直接依赖于“好瓜”和“甜度”
- 从条件概率表可以得到 -> “根蒂”对“甜度”的量化依赖关系
  P(根蒂 = 硬挺|甜度 = 高) = 0.1

贝叶斯网

- 贝叶斯网有效地表达了属性间的条件独立性。给定父结集，贝叶斯网假设每个属性与他的非后裔属性独立。
- $B = \langle G, \Theta \rangle$将属性 $x_1, x_2, \cdots, x_d$的联合概率分布定义为

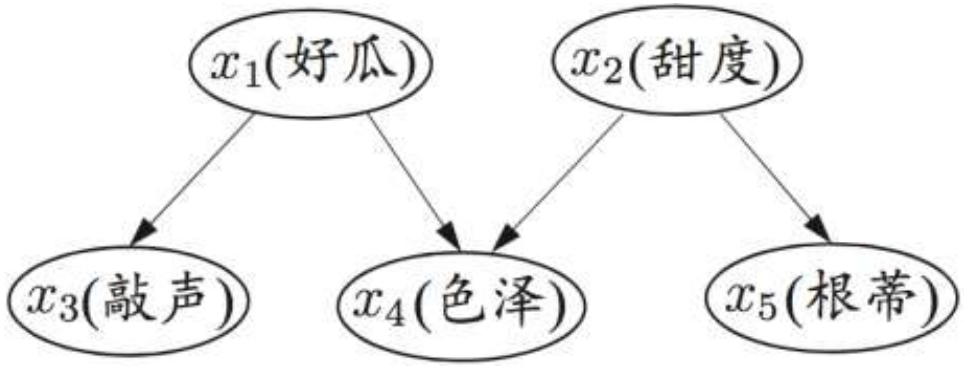
$$P_B(x_1, x_2, \cdots, x_d) = \prod_{i=1}^{d} P_B(x_i|\pi_i) = \prod_{i=1}^{d} \theta_{x_i|\pi_i}$$



$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)$$

贝叶斯网

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)$$



x_4和x_5在给定x_2的取值时独立

$$x_5 \perp x_4|x_2$$

$$P(x_3, x_4|x_1) = \frac{\sum_{x_2,x_5} P(x_1, x_2, x_3, x_4, x_5)}{\sum_{x_2,x_3,x_4,x_5} P(x_1, x_2, x_3, x_4, x_5)}$$

$$= \frac{\sum_{x_2,x_5} P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)}{\sum_{x_2,x_3,x_4,x_5} P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)}$$

$$= \frac{\sum_{x_2} P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)}{P(x_1)}$$

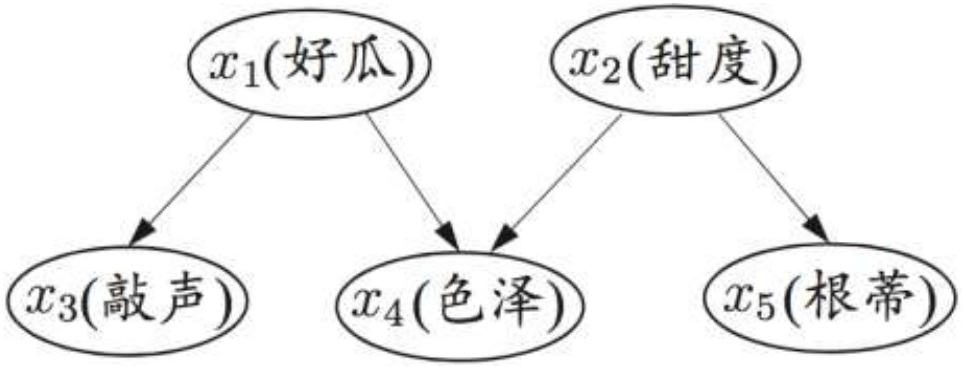
$$P(x_4|x_1, x_2)P(x_2) = P(x_4, x_2|x_1)$$

$$= \frac{P(x_1)P(x_3|x_1)P(x_4|x_1)}{P(x_1)} = P(x_3|x_1)P(x_4|x_1)$$

x_3和x_4在给定x_1的取值时独立 $x_3 \perp x_4|x_1$

贝叶斯网

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)$$



$$P(x_1, x_2) = \sum_{x_3, x_4, x_5} P(x_1, x_2, x_3, x_4, x_5)$$

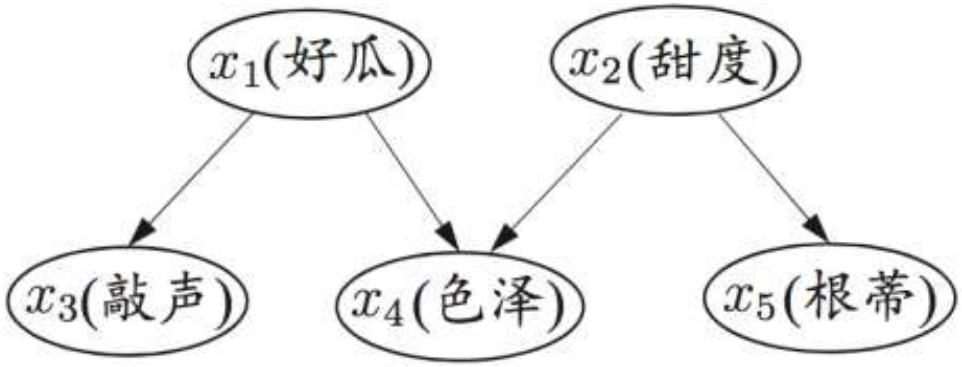
$$= \sum_{x_3, x_4, x_5} P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)$$

$$= P(x_1)P(x_2)$$

x_1和x_2相互独立　　$x_1 \perp x_2$　　称为边际独立性

贝叶斯网

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)$$



$$P(x_1, x_2|x_4) = \frac{\sum_{x_3,x_5} P(x_1, x_2, x_3, x_4, x_5)}{\sum_{x_1,x_2,x_3,x_5} P(x_1, x_2, x_3, x_4, x_5)}$$

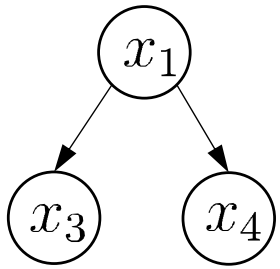
$$= \frac{\sum_{x_3,x_5} P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)}{\sum_{x_1,x_2,x_3,x_5} P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_1, x_2)P(x_5|x_2)}$$

$$= \frac{P(x_1)P(x_2)P(x_4|x_1, x_2)}{\sum_{x_1,x_2} P(x_1)P(x_2)P(x_4|x_1, x_2)} = \frac{P(x_1, x_2, x_4)}{P(x_4)}$$

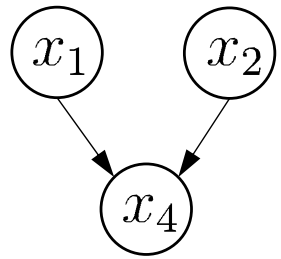
x_1和x_2在给定x_4的取值时不独立　　$x_1 \not\perp x_2|x_4$

贝叶斯网

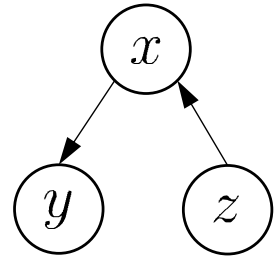
- 贝叶斯网中三个变量之间的典型依赖关系：



同父结构



V型结构



顺序结构

$$P(y,z|x) = \frac{P(x,y,z)}{P(x)}$$

$$= \frac{P(z)P(x|z)P(y|x)}{P(x)}$$

$$= \frac{P(x)P(z|x)P(y|x)}{P(x)}$$

$$= P(z|x)P(y|x)$$

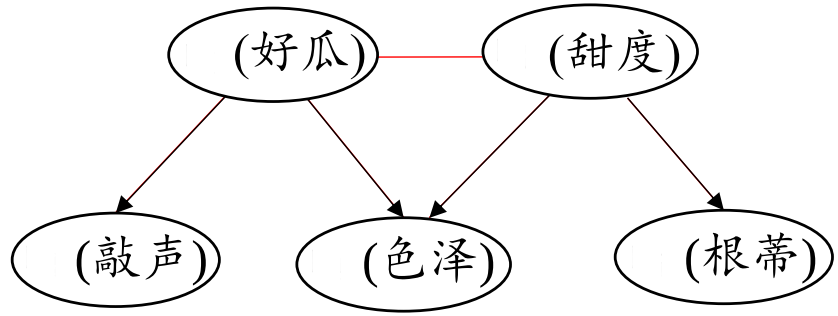
y和z在给定x的取值时独立

$$y \perp z|x$$

贝叶斯网

- 分析有向图中变量间的条件独立性，可使用“有向分离”(D-separation)

  - V型结构父结点相连
  - 有向边变成无向边



由此产生的图称为道德图(moral graph)

贝叶斯网：学习

- 贝叶斯网络首要任务：根据训练集找出结构最“恰当”贝叶斯网
- 我们用评分函数评估贝叶斯网与训练数据的契合程度。
  - “最小描述长度”（Minimal Description Length, MDL）综合编码长度（包括描述网路和编码数据）最短

- 给定训练集$D = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$，贝叶斯网$B = \langle G, \Theta \rangle$在$D$上的评价函数可以写为

$$s(B|D) = f(\theta)|B| - LL(B|D)$$

其中，$|B|$是贝叶斯网的参数个数；$f(\theta)$表示描述每个参数θ所需的字节数

$$LL(B|D) = \sum_{i}^{m} \log P_B(\boldsymbol{x}_i)$$

是贝叶斯网的对数似然

贝叶斯网：学习

$$s(B|D) = f(\theta)|B| - LL(B|D)$$

- 若$f(\theta) = 1$，即每个参数用1位编码描述，则得到AIC (Akaike Information Criterion)

$$AIC(B|D) = |B| - LL(B|D)$$

- 若$f(\theta) = \frac{1}{2}\log m$，则得到BIC (Bayesian Information Criterion)

$$BIC(B|D) = \frac{\log m}{2}|B| - LL(B|D)$$

若贝叶斯网$B = \langle G, \Theta \rangle$网络结构$G$给定，$s(B|D)$最小化等价于参数$\Theta$的极大似然估计

参数$\theta_{x_i|\pi_i}$能直接在训练数据D上通过经验估计　　$\theta_{x_i|\pi_i} = \hat{P}_D(x_i|\pi_i)$

因此为了最小化评分函数$s(B|D)$，只需要对网络结构进行搜索

该问题是一个NP难问题

1. 贪心法：从某个网络结构出发，每次调整一条边（增加、删除、调整方向），直到评分函数不再降低为止
2. 通过给网络结构施加约束来消减搜索空间，例如将网络结构限定为树形结构

贝叶斯网：推断

- 通过已知变量观测值来推测待推测查询变量的过程称为“推断”(inference），已知变量观测值称为“证据” (evidence)。
- 最理想的是根据贝叶斯网络定义的联合概率分布来精确计算后验概率，在现实应用中，贝叶斯网的近似推断常使用吉布斯采样(Gibbs sampling)来完成。

输入： 贝叶斯网 $B = \langle G, \Theta \rangle$;
采样次数 T;
证据变量 $\mathbf{E}$ 及其取值 $\mathbf{e}$;
待查询变量 $\mathbf{Q}$ 及其取值 $\mathbf{q}$.

过程：

$n_q = 0$
$\mathbf{q}^0 =$ 对 $\mathbf{Q}$ 随机赋初值
for $t = 1, 2, \ldots, T$ **do**
  for $Q_i \in \mathbf{Q}$ **do**
    $\mathbf{Z} = \mathbf{E} \cup \mathbf{Q} \setminus \{Q_i\}$;
    $\mathbf{z} = \mathbf{e} \cup \mathbf{q}^{t-1} \setminus \{q_i^{t-1}\}$;
    根据 B 计算分布 $P_B(Q_i \mid \mathbf{Z} = \mathbf{z})$;
    $q_i^t =$ 根据 $P_B(Q_i \mid \mathbf{Z} = \mathbf{z})$ 采样所获 Q_i 取值;
    $\mathbf{q}^t =$ 将 $\mathbf{q}^{t-1}$ 中的 q_i^{t-1} 用 q_i^t 替换
  end for
  if $\mathbf{q}^t = \mathbf{q}$ **then**
    $n_q = n_q + 1$
  end if
end for

输出： $P(\mathbf{Q} = \mathbf{q} \mid \mathbf{E} = \mathbf{e}) \simeq \frac{n_q}{T}$

$$P(Q = q|E = e)$$

{好瓜 = 是, 甜度 = 高}

{色泽 = 青绿，敲声 = 浊响，根蒂 = 蜷缩}

吉布斯采样随机产生一个与证据 $E = e$ 一致的样本q^0作为初始点，然后每步从当前样本出发产生下一个样本。假定经过T次采样的得到与q一致的样本共有n_q个，则可近似估算出后验概率

$$P(Q = q|E = e) \simeq \frac{n_q}{T}$$

贝叶斯网：推断

- 吉布斯采样是在所有变量的联合状态空间与证据$E = e$一致的子空间中的随机游走，即每一步仅依赖于前一步的状态，这是一个“马尔可夫链”(Markov Chain)

- 在一定的条件下，无论从什么初始状态开始，马尔科夫链第t步的状态分布在$t \to \infty$时比收敛于一个平稳分布，即$P(Q|E = e)$

- 当T很大时，吉布斯采样相当于根据$P(Q|E = e)$采样

EM算法

- “不完整”的样本：西瓜已经脱落的根蒂，无法看出是“蜷缩”还是“坚挺”,则训练样本的“根蒂”属性变量值未知，如何对模型参数进行估计?

- 未观测的变量称为“隐变量”(latent variable)。令 $\boldsymbol{X}$表示已观测变量集，$\boldsymbol{Z}$表示隐变量集，若预对模型参数$\boldsymbol{\Theta}$做极大似然估计，则应最大化对数似然函数

$$LL(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{Z}) = \ln P(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\Theta})$$

- 由于Z是隐变量，上式无法直接求解。此时我们可以通过对$\boldsymbol{Z}$边际化，来最大化已观测数据的对数“边际似然”(marginal likelihood)

$$LL(\boldsymbol{\Theta}|\boldsymbol{X}) = \ln P(\boldsymbol{X}|\boldsymbol{\Theta}) = \ln \sum_{Z} P(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\Theta})$$

EM算法

- EM (Expectation-Maximization)算法 [Dempster et al., 1977] 是常用的估计参数隐变量的利器。

当参数Θ已知　　根据训练数据推断出最优隐变量Z的值(E步)

当Z已知　　对Θ做极大似然估计(M步)

以初始值Θ^0为起点，可迭代执行以下步骤直至收敛：

E　基于Θ^t推断隐变量Z的期望，记为 Z^t;

M　基于已观测到变量X和Z^t对参数Θ做极大似然估计，记Θ^{t+1};

EM算法

若不是取$\boldsymbol{Z}$的期望，而是基于$\boldsymbol{\Theta}^t$计算隐变量Z的概率分布$P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)$，则EM算法的两个步骤

以初始值$\boldsymbol{\Theta}^0$为起点，可迭代执行以下步骤直至收敛：

E 基于$\boldsymbol{\Theta}^t$推断隐变量$\boldsymbol{Z}$的分布$P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)$，并计算对数似然$LL(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{Z})$关于$\boldsymbol{Z}$的期望;

$$Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)=\mathbb{E}_{\boldsymbol{Z}\sim P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)}LL(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{Z})$$

M 寻找参数最大化期望似然;

$$\boldsymbol{\Theta}^{t+1}=\arg\max_{\boldsymbol{\Theta}}Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$$

EM算法

最大化已观测数据的对数“边际似然”

$$\arg\max_{\Theta} LL(\Theta|X) = \ln P(X|\Theta) = \ln \sum_{Z} P(X, Z|\Theta)$$

$$\ln P(X|\Theta) = \ln \sum_{Z} Q(Z) \frac{P(X, Z|\Theta)}{Q(Z)}$$

$$\geq \sum_{Z} Q(Z) \ln \frac{P(X, Z|\Theta)}{Q(Z)} \qquad = \sum_{Z} Q(Z) \ln \frac{P(Z|X, \Theta) P(X|\Theta)}{Q(Z)}$$

$$\mathcal{L}(Q, \Theta)$$

$$= \ln P(X|\Theta) + \sum_{Z} Q(Z) \ln \frac{P(Z|X, \Theta)}{Q(Z)}$$

$$= \ln P(X|\Theta) - KL(Q||P)$$

$$\ln P(X|\Theta) = \mathcal{L}(Q, \Theta) + KL(Q||P)$$

EM算法

$$\max_{\boldsymbol{\Theta}} \ln P(\boldsymbol{X}|\boldsymbol{\Theta}) = \max_{Q,\boldsymbol{\Theta}} \big(\mathcal{L}(Q,\boldsymbol{\Theta}) + KL(Q||P)\big)$$

因为$KL(Q||P) \geq 0$, 所以$\mathcal{L}(Q,\boldsymbol{\Theta}) \leq \ln P(\boldsymbol{X}|\boldsymbol{\Theta})$

$$\mathcal{L}(Q,\boldsymbol{\Theta}) = \sum_{\boldsymbol{Z}} Q(\boldsymbol{Z}) \ln \frac{P(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\Theta})}{Q(\boldsymbol{Z})}$$

当$KL(Q||P) = 0$，即$Q(\boldsymbol{Z}) = P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\Theta})$时，$\mathcal{L}(Q,\boldsymbol{\Theta})$取得最大值 $= \ln P(\boldsymbol{X}|\boldsymbol{\Theta})$

E 基于$\boldsymbol{\Theta}^t$推断隐变量$\boldsymbol{Z}$的分布$P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)$，并计算对数似然$LL(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{Z})$关于$\boldsymbol{Z}$的期望;

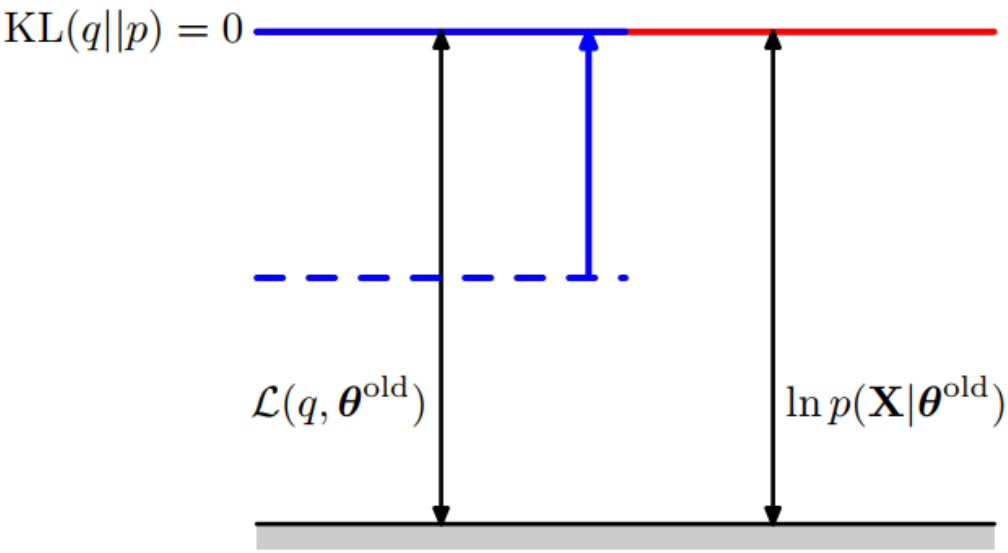
$$\mathcal{L}(P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\Theta}),\boldsymbol{\Theta}) = \mathbb{E}_{\boldsymbol{Z}\sim P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)} LL(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{Z}) = Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$$

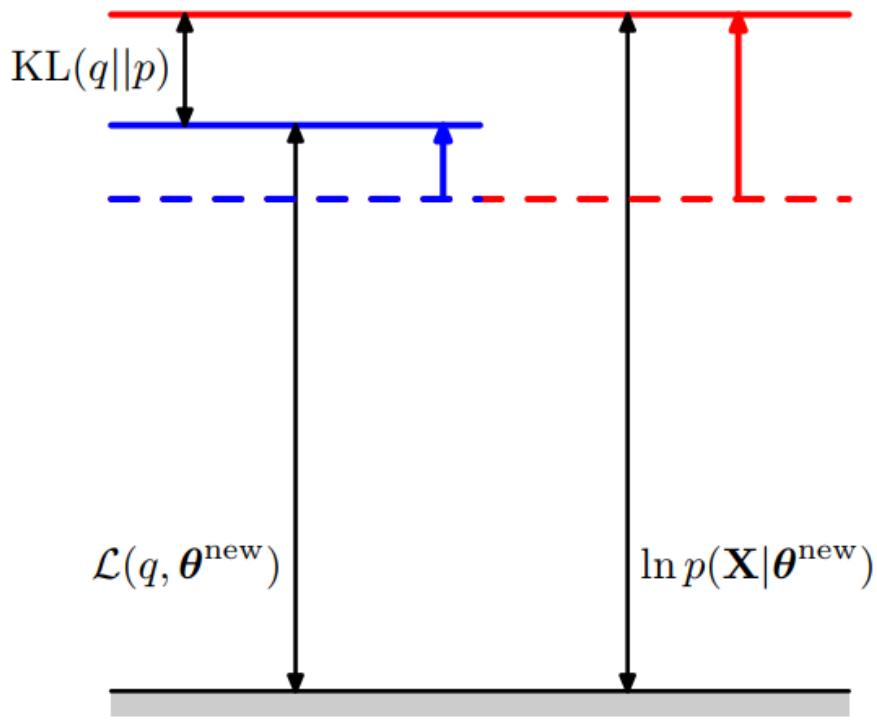
M 寻找参数最大化期望似然;

$$\boldsymbol{\Theta}^{t+1} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$$

EM算法图解

$$\max_{\boldsymbol{\Theta}} \ln P(\boldsymbol{X}|\boldsymbol{\Theta}) = \max_{Q,\boldsymbol{\Theta}} \big(\mathcal{L}(Q,\boldsymbol{\Theta}) + KL(Q||P)\big)$$





EM算法的应用—高斯混合模型

- 高斯混合分布的定义

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

该分布由k个混合分布组成，每个分布对应一个高斯分布。其中，$\boldsymbol{\mu}_c$与$\boldsymbol{\Sigma}_c$是第c个高斯混合成分的参数。而α_c为相应的“混合系数”，$\sum_c \alpha_c = 1$。

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i}^{m} \ln p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{m} \ln \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

EM算法的应用—高斯混合模型

假设样本的生成过程由高斯混合分布给出：

(1) 根据$\alpha_1, \cdots, \alpha_k$定义的先验分布选择高斯混合成分，$\alpha_i$为选择第$i$个成分的概率；

(2) 根据被选择的混合成分的概率密度函数进行采样，从而生成相应的样本

- 数据集$D = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$由上述过程生成
- 令随机变量$z_i \in \{1, \cdots, k\}$表示生成样本的高斯混合成分，则z_j的先验概率$P(z_i = c) = \alpha_c$

$$P(\boldsymbol{x}_i, z_i) = P(\boldsymbol{x}_i | z_i) P(z_i) = \prod_{c=1}^{k} [P(\boldsymbol{x}_i | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \alpha_c]^{\mathbb{I}(z_i = c)}$$

$$\sum_{z_i} P(\boldsymbol{x}_i, z_i) = \sum_{z_i} \prod_{c=1}^{k} [P(\boldsymbol{x}_i | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \alpha_c]^{\mathbb{I}(z_i = c)} = \sum_{c=1}^{k} P(\boldsymbol{x}_i | \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \alpha_c$$

EM算法的应用—高斯混合模型

E 基于$\boldsymbol{\Theta}^t$推断隐变量$\boldsymbol{Z}$的分布$P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)$，并计算对数似然$LL(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{Z})$关于$\boldsymbol{Z}$的期望;

$$\mathcal{L}(P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\Theta}),\boldsymbol{\Theta}) = \mathbb{E}_{\boldsymbol{Z}\sim P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)} LL(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{Z}) = Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$$

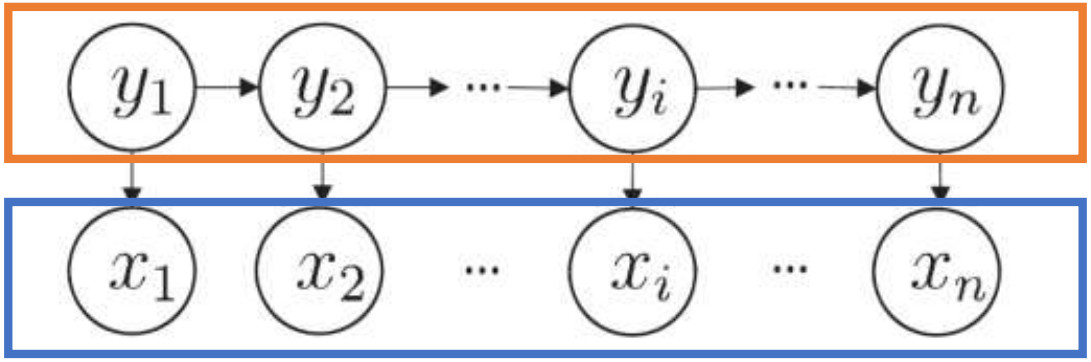
$$P(z_i = c|\boldsymbol{x}_i) = \frac{P(\boldsymbol{x}_i, z_i = c)}{\sum_{z_i} P(\boldsymbol{x}_i, z_i)} = \frac{\alpha_c P(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c'} \alpha_{c'} P(\boldsymbol{x}_i|\boldsymbol{\mu}_{c'}, \boldsymbol{\Sigma}_{c'})} = \gamma_{ic}$$

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{Z}\sim P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)} LL(\boldsymbol{\Theta}|\boldsymbol{X},\boldsymbol{Z}) &= \mathbb{E}_{\boldsymbol{Z}\sim P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)} \log \prod_{i=1}^{m} \prod_{c=1}^{k} [P(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)\alpha_c]^{\mathbb{I}(z_i=c)} \\
&= \mathbb{E}_{\boldsymbol{Z}\sim P(\boldsymbol{Z}\,|\,\boldsymbol{X},\boldsymbol{\Theta}^t)} \sum_{i}^{m} \sum_{c=1}^{k} \mathbb{I}(z_i = c) \log P(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)\alpha_c \\
&= \sum_{i}^{m} \sum_{c=1}^{k} \gamma_{ic} [\log P(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) + \log \alpha_c]
\end{aligned}$$

M 寻找参数最大化期望似然; $\boldsymbol{\Theta}^{t+1} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$

EM算法的应用—隐马尔可夫模型（HMM）

- 隐马尔可夫模型（Hidden Markov Model，HMM）
- 组成
  - 状态变量: $\{y_1, y_2, \ldots, y_n\}$ 通常假定是**隐藏**的，不可被观测的
    - 取值范围为$\mathcal{Y}$， 通常有N个可能取值的离散空间
  - 观测变量: $\{x_1, x_2, \ldots, x_n\}$ 表示第i时刻的观测值集合
    - 观测变量可以为离散或连续型，本章中只讨论离散型观测变量，取值范围$\mathcal{X}$为$\{o_1, o_2, \ldots, o_M\}$



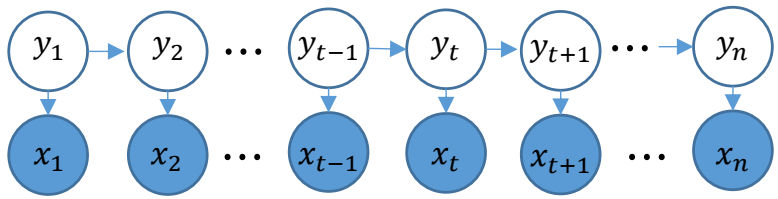
EM算法的应用—隐马尔可夫模型（HMM）

- t时刻的状态x_t仅依赖于$t-1$时刻状态x_{t-1}，与其余n − 2个状态无关

马尔可夫链：

系统下一时刻状态仅由当前状态决定，不依赖于以往的任何状态

$$P(x_1, y_1, \ldots, x_n, y_n) = P(y_1)P(x_1|y_1)\prod_{t=2}^{n} P(y_t|y_{t-1})P(x_t|y_t)$$



联合概率

HMM的基本组成

- 确定一个HMM需要**三组参数**
  - 状态**转移概率**：模型在各个状态间转换的概率
  - 表示在任意时刻t，若状态为s_i，下一状态为s_j的概率

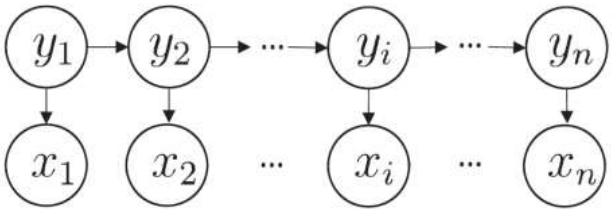
$$A = [a_{i,j}]_{N\times N} \qquad a_{ij} = P(y_{t+1} = s_j | y_t = s_i) \qquad 1 \le i, j \le N$$

  - 输出**观测概率**：模型根据当前状态获 得各个观测值的概率
  - 在任意时刻t，若状态为s_i，则在观测值为o_j的概率

$$B = [b_{i,j}]_{N\times M} \qquad b_{ij} = P(x_t = o_j | y_t = s_i) \qquad 1 \le i \le N \quad 1 \le j \le M$$

  - **初始**状态概率：模型在初始时刻各个状态出现的概率

$$\pi = [\pi_1, \dots, \pi_n] \qquad \pi_i = P(y_1 = s_i) \qquad 1 \le i \le N$$

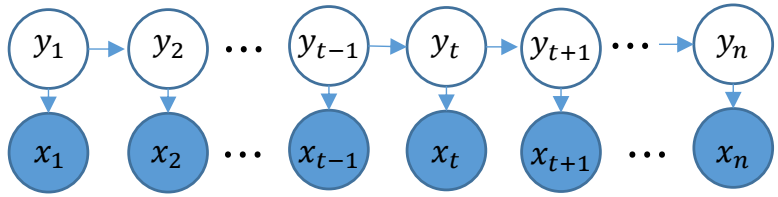


$$P(x_1, y_1, \dots, x_n, y_n) = P(y_1)P(x_1|y_1)\prod_{i=2}^{n} P(y_i|y_{i-1})P(x_i|y_i)$$

HMM的生成过程

- 通过指定状态空间$\mathcal{Y}$，观测空间$\mathcal{X}$和上述三组参数，就能确定一个隐马尔可夫模型。给定$\lambda = [A, B, \pi]$，它按如下过程**生成观察序列**

1. 设置t = 1, 并根据初始状态π选择初始状态y_1
2. 根据状态y_t和输出观测概率B选择观测变量取值x_t
3. 根据状态y_t和状态转移矩阵A转移模型状态，即确定y_{t+1}
4. 若$t < n$, 设置$t = t + 1$，并转到（2）步，否则停止



HMM的基本问题

对于模型 $\lambda = [A, B, \pi]$，给定观测序列 $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$

基本问题	具体应用
• **概率计算问题**：评估模型和观测序列间的匹配程度：有效计算观测序列产生概率$P(\boldsymbol{x}\|\lambda)$	• 根据以往的观序列$\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$预测下一时刻最有可能的观测值$x_{n+1}$
• **预测问题**：根据观测序列“推测”隐藏的模型状态 $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$	• 语音识别：根据观测的语音信号推测最有可能的状态序列（即：对应的文字）
• **学习问题**：如何调整模型参数 $\lambda = [A, B, \pi]$以使得该序列出现的概率$P(\boldsymbol{x}\|\lambda)$最大	• 通过数据学习参数（模型训练）

学习问题（Baum-Welch算法）

- 如何调整模型参数 $\lambda = [A, B, \pi]$以使得该序列出现的概率$P(\boldsymbol{x}|\lambda)$最大
- 使用EM算法

E 基于$\boldsymbol{\Theta}^t$推断隐变量$\boldsymbol{Z}$的分布$P(\boldsymbol{Z}\,|\,\boldsymbol{X}, \boldsymbol{\Theta}^t)$，并计算对数似然$LL(\boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{Z})$关于$\boldsymbol{Z}$的期望;

$$\mathcal{L}(P(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\Theta}), \boldsymbol{\Theta}) = \mathbb{E}_{\boldsymbol{Z}\sim P(\boldsymbol{Z}\,|\,\boldsymbol{X}, \boldsymbol{\Theta}^t)} LL(\boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{Z}) = Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$$

M 寻找参数最大化期望似然;

$$\boldsymbol{\Theta}^{t+1} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$$

学习问题（Baum-Welch算法）

$$Q(\lambda|\lambda^t) = \mathbb{E}_{\boldsymbol{y}\sim P(Y|\boldsymbol{x},\boldsymbol{\lambda}^t)} LL(\lambda|\boldsymbol{x},\boldsymbol{y})$$

$$= \mathbb{E}_{\boldsymbol{y}\sim P(Y|\boldsymbol{x},\boldsymbol{\lambda}^t)} \log P(\boldsymbol{x},\boldsymbol{y}|\lambda)$$

$$= \mathbb{E}_{\boldsymbol{y}\sim P(Y|\boldsymbol{x},\boldsymbol{\lambda}^t)} \log P(y_1)P(x_1|y_1)\prod_{t=2}^{n} P(y_t|y_{t-1})P(x_t|y_t)$$

$$= \mathbb{E}_{\boldsymbol{y}\sim P(Y|\boldsymbol{x},\boldsymbol{\lambda}^t)} \log P(y_1) + \log P(x_1|y_1) + \sum_{t=2}^{n} \log P(y_t|y_{t-1}) + \log P(x_t|y_t)$$

$$= \mathbb{E}_{y_1}[\log P(y_1) + \log P(x_1|y_1)] + \sum_{t=2}^{n} \mathbb{E}_{y_{t-1},y_t}[\log P(y_t|y_{t-1})] + \mathbb{E}_{y_t}[\log P(x_t|y_t)]$$

$$= \sum_i \gamma_1(i)\left(\log \pi_i + b_{i,x_1}\right) + \sum_{t=2}^{n}\left(\sum_{i,j} \xi_{t-1}(i,j)\log a_{ij} + \sum_i \gamma_t(i) b_{i,x_t}\right)$$

学习问题（Baum-Welch算法）

$$Q(\lambda|\lambda^t)=\sum_i \gamma_1(i)\left(\log \pi_i + b_{i,x_1}\right)+\sum_{t=2}^{n}\left(\sum_{i,j}\xi_{t-1}(i,j)\log a_{ij}+\sum_i \gamma_t(i)\log b_{i,x_t}\right)$$

- 求解初始状态概率

  针对约束条件$\sum_i \pi_i = 1$，基于拉格朗日乘子法，求解 $\pi_i = \gamma_1(i)$

- 求解转移概率

  针对约束条件$\sum_j a_{i,j} = 1$，基于拉格朗日乘子法，求解 $a_{i,j} = \frac{\sum_{t=2}^{n}\xi_{t-1}(i,j)}{\sum_j \sum_{t=2}^{n}\xi_{t-1}(i,j)}$

- 求解观测概率

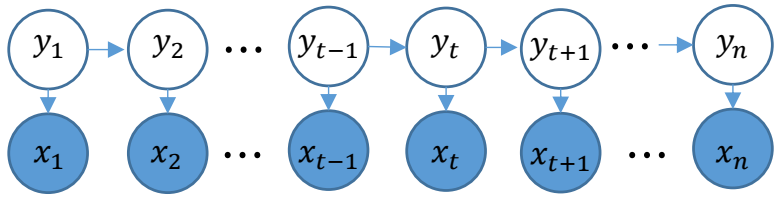
  针对约束条件$\sum_j b_{i,j} = 1$，基于拉格朗日乘子法，求解 $b_{i,j} = \frac{\sum_{t=1}^{n}\gamma_t(i)\,\mathbb{I}(x_t = o_j)}{\sum_{t=1}^{n}\gamma_t(i)}$

概率计算问题

- 对于模型 $\lambda = [A, B, \pi]$，给定观测序列 $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$
- 评估模型和观测序列之间的匹配程度: 有效计算观测序列其产生的概率

$$P(\boldsymbol{y}|\lambda) = \pi_{y_1} a_{y_1,y_2} a_{y_2,y_3} \cdots a_{y_{n-1},y_n}$$

$$P(\boldsymbol{x}|\boldsymbol{y},\lambda) = b_{y_1,x_1} b_{y_2,x_2} \cdots b_{y_n,x_n}$$

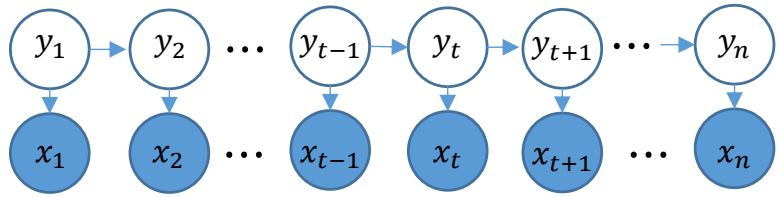


$$P(\boldsymbol{x}|\lambda) = \sum_{\boldsymbol{y}} P(\boldsymbol{x}|\boldsymbol{y},\lambda)\, P(\boldsymbol{y}|\lambda)$$

$$= \sum_{y_1,y_2,\ldots,y_n} \pi_{y_1} b_{y_1,x_1} a_{y_1,y_2} b_{y_2,x_2} a_{y_2,y_3} \cdots a_{y_{n-1},y_n} b_{y_n,x_n}$$

直接计算开销很大，$O(nN^n)$，不可行

概率计算问题



$$P(\boldsymbol{x}|\lambda) = \sum_{y_1,y_2,\ldots,y_n} \pi_{y_1} b_{y_1,x_1} a_{y_1,y_2} b_{y_2,x_2} a_{y_2,y_3} \cdots a_{y_{n-1},y_n} b_{y_n,x_n}$$

$$= \sum_{y_1} \pi_{y_1} b_{y_1,x_1} \sum_{y_2} a_{y_1,y_2} b_{y_2,x_2} \sum_{y_3} a_{y_2,y_3} \cdots \sum_{y_n} a_{y_{n-1},y_n} b_{y_n,x_n}$$

$$= \sum_{y_1} \pi_{y_1} b_{y_1,x_1} \cdots \sum_{y_t} a_{y_{t-1},y_t} b_{y_t,x_t} \sum_{y_{t+1}} a_{y_t,y_{t+1}} b_{y_{t+1},x_{t+1}} \cdots \sum_{y_n} a_{y_{n-1},y_n} b_{y_n,x_n}$$

$$= \sum_{y_t} \left(\sum_{y_1} \pi_{y_1} b_{y_1,x_1} \cdots a_{y_{t-1},y_t} b_{y_t,x_t} \sum_{y_{t+1}} a_{y_t,y_{t+1}} b_{y_{t+1},x_{t+1}} \cdots \sum_{y_n} a_{y_{n-1},y_n} b_{y_n,x_n} \right)$$

$$= \sum_{y_t} P(x_1, x_2, \ldots, x_t, y_t|\lambda) \, P(x_{t+1}, x_{t+2}, \ldots, x_n|y_t, \lambda)$$

$$= \sum_{y_t} \alpha(y_t) \beta(y_t)$$

概率计算问题

$$\alpha(y_t) = P(x_1, x_2, \dots, x_t, y_t|\lambda)$$

$$= \sum_{y_{t-1}} P(x_1, x_2, \dots, x_{t-1}, y_{t-1}|\lambda) P(y_t|y_{t-1}, A) P(x_t|y_t, B)$$

$$= \sum_{y_{t-1}} \alpha(y_{t-1}) P(y_t|y_{t-1}, A) P(x_t|y_t, B)$$

记$\alpha_t(i) = \alpha(y_t = s_i)$，则 $\alpha_t(i) = \sum_{j=1}^{N} \alpha_{t-1}(j) a_{j,i} b_{i,x_t}$

(1) 初值 $\alpha_1(i) = \pi_i b_{i,x_1}$

(2) 递推 $\alpha_t(i) = \sum_{j=1}^{N} \alpha_{t-1}(j) a_{j,i} b_{i,x_t} \quad t = 1,2, \dots, n$

(3) 终止 $P(\boldsymbol{x}|\lambda) = \sum_{i=1}^{N} \alpha_n(i)$

前向算法

概率计算问题

$$\beta(y_t) = P(x_{t+1}, x_{t+2}, \dots, x_n | y_t, \lambda)$$
$$= \sum_{y_{t+1}} P(x_{t+1}, x_{t+2}, \dots, x_n, y_{t+1} | y_t, \lambda)$$
$$= \sum_{y_{t+1}} P(x_{t+2}, \dots, x_n, y_{t+2} | y_{t+1}, \lambda) P(x_{t+1} | y_{t+1}, B) P(y_{t+1} | y_t, A)$$

记$\beta_t(i) = \beta(y_t = s_i)$，则 $\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{i,j} b_{j,x_{t+1}}$

(1) 初值 $\beta_n(i) = 1$
(2) 递推 $\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{i,j} b_{j,x_{t+1}} \quad t = n-1, n-2, \dots, 1$
(3) 终止 $P(\boldsymbol{x}|\lambda) = \sum_{i=1}^{N} \beta_1(i)\, \pi_i b_{i,x_1}$

后向算法

概率计算问题

- 一些概率和期望值计算

给定模型λ和观测序列$\boldsymbol{x}$，在t时刻处于状态s_i的概率

$$\gamma_t(i) = P(y_t = s_i|\boldsymbol{x},\lambda) = \frac{P(y_t = s_i,\boldsymbol{x}|\lambda)}{P(\boldsymbol{x}|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}$$

$$\alpha(y_t) = P(x_1, x_2, \dots, x_t, y_t|\lambda)$$

$$\beta(y_t) = P(x_{t+1}, x_{t+2}, \dots, x_n|y_t, \lambda)$$

$$\alpha(y_t)\beta(y_t) = P(y_t, \boldsymbol{x}|\lambda)$$

给定模型λ和观测序列$\boldsymbol{x}$，在t时刻处于状态s_i且t+1时刻处于状态s_j的概率

$$\xi_t(i,j) = P\left(y_t = s_i, y_{t+1} = s_j\middle|\boldsymbol{x},\lambda\right) = \frac{P\left(y_t = s_i, y_{t+1} = s_j,\boldsymbol{x}\middle|\lambda\right)}{P(\boldsymbol{x}|\lambda)} = \frac{\alpha_t(i)a_{ij}b_{j,x_{t+1}}\beta_{t+1}(j)}{\sum_{i,j}\alpha_t(i)a_{ij}b_{j,x_{t+1}}\beta_{t+1}(j)}$$

概率计算问题

- 一些概率和期望值计算

给定模型λ和观测序列$\boldsymbol{x}$，在t时刻处于状态s_i的概率 $\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)}$

给定模型λ和观测序列$\boldsymbol{x}$，在t时刻处于状态s_i且t+1时刻处于状态s_j的概率

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_{j,x_{t+1}}\beta_{t+1}(j)}{\sum_{i,j}\alpha_t(i)a_{ij}b_{j,x_{t+1}}\beta_{t+1}(j)}$$

给定模型λ和观测序列$\boldsymbol{x}$，状态s_i出现的概率 $\sum_{t=1}^{n}\gamma_t(i)$

给定模型λ和观测序列$\boldsymbol{x}$，状态s_i转移到状态s_j的概率 $\sum_{t=1}^{n-1}\xi_t(i,j)$

预测问题（Viterbi Algorithm）

- 根据观测序列“推测”隐藏的模型状态 $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$

$$\operatorname{argmax}_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}, \lambda) = \operatorname{argmax}_{\boldsymbol{y}} P(\boldsymbol{y}, \boldsymbol{x}|\lambda)$$

$$\max_{\boldsymbol{y}} P(\boldsymbol{y}, \boldsymbol{x}|\lambda) = \max_{y_1,\ldots,y_n} P(y_1)P(x_1|y_1)\prod_{i=2}^{n} P(y_i|y_{i-1})P(x_i|y_i)$$

$$\max(ab, ac) = a \cdot \max(b, c)$$

$$= \max_{y_1} P(y_1)P(x_1|y_1) \max_{y_2} P(y_2|y_1)P(x_2|y_2) \cdots \max_{y_n} P(y_n|y_{n-1})P(x_n|y_n)$$

前t项最大化

$$\delta(y_{t+1}) = \max_{y_1,\ldots,y_t} P(y_{t+1}, y_1, \ldots, y_t, x_1, \ldots, x_{t+1})$$

$$\delta(y_2) = \max_{y_1} P(y_2, y_1, x_1, x_2)$$

$$= \max_{y_t} P(y_{t+1}|y_t)P(x_{t+1}|y_{t+1}) \max_{y_1,\ldots,y_{t-1}} P(y_t, y_1, \ldots, y_{t-1}, x_1, \ldots, x_t)$$

$$= \max_{y_t} P(y_{t+1}|y_t)P(x_{t+1}|y_{t+1})\delta(y_t)$$

预测问题（Viterbi 算法）

概率连乘容易导致溢出，因此引入对数函数

$$\delta(y_{t+1}) = \max_{y_1,\cdots,y_t} \log P(y_{t+1}, y_1, \cdots, y_t, x_1, \cdots, x_{t+1})$$

$$= \log P(x_{t+1}|y_{t+1}) + \max_{y_t} \log P(y_{t+1}|y_t) + \max_{y_1,\cdots,y_{t-1}} \log P(y_t, y_1, \cdots, y_{t-1}, x_1, \cdots, x_t)$$

$$= \log P(x_{t+1}|y_{t+1}) + \max_{y_t} \log P(y_{t+1}|y_t) + \delta(y_t)$$

记$\delta_t(i) = \delta(y_t = s_i)$，则 $\delta_{t+1}(i) = \log b_{i,x_{t+1}} + \max_{1\le j\le N}\left[\log a_{j,i} + \delta_t(j)\right]$

$$\delta_1(i) = \log \pi_i + \log b_{i,x_1}$$

作业

- 习题7.4
- 习题7.5
- 证明EM算法的收敛性
- 在HMM中，求解概率$P(x_{n+1}|x_1, x_2, \ldots, x_n)$

第八章：集成学习

主讲：连德富 特任教授 | 博士生导师

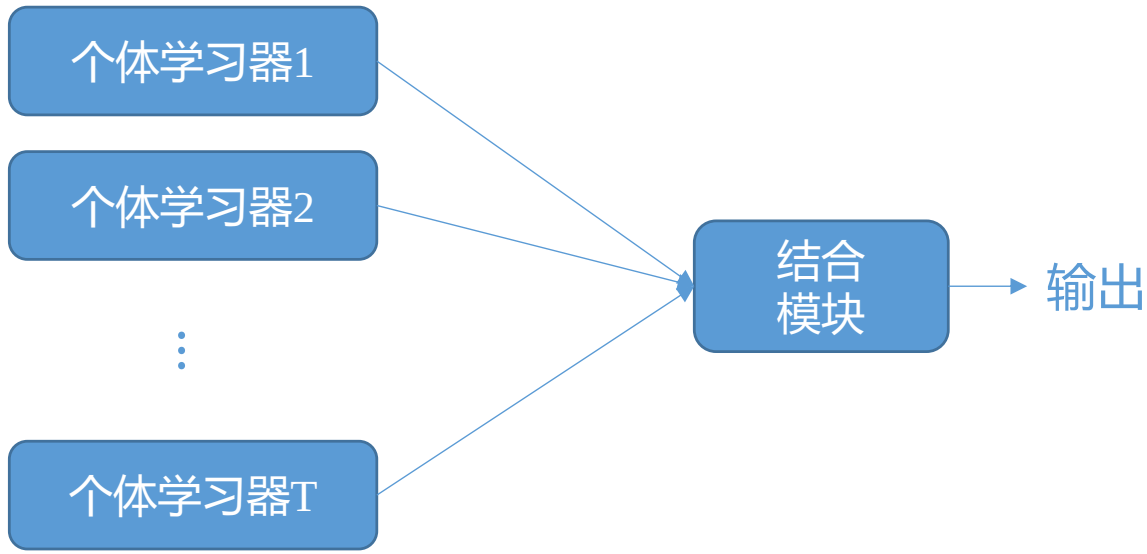
邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

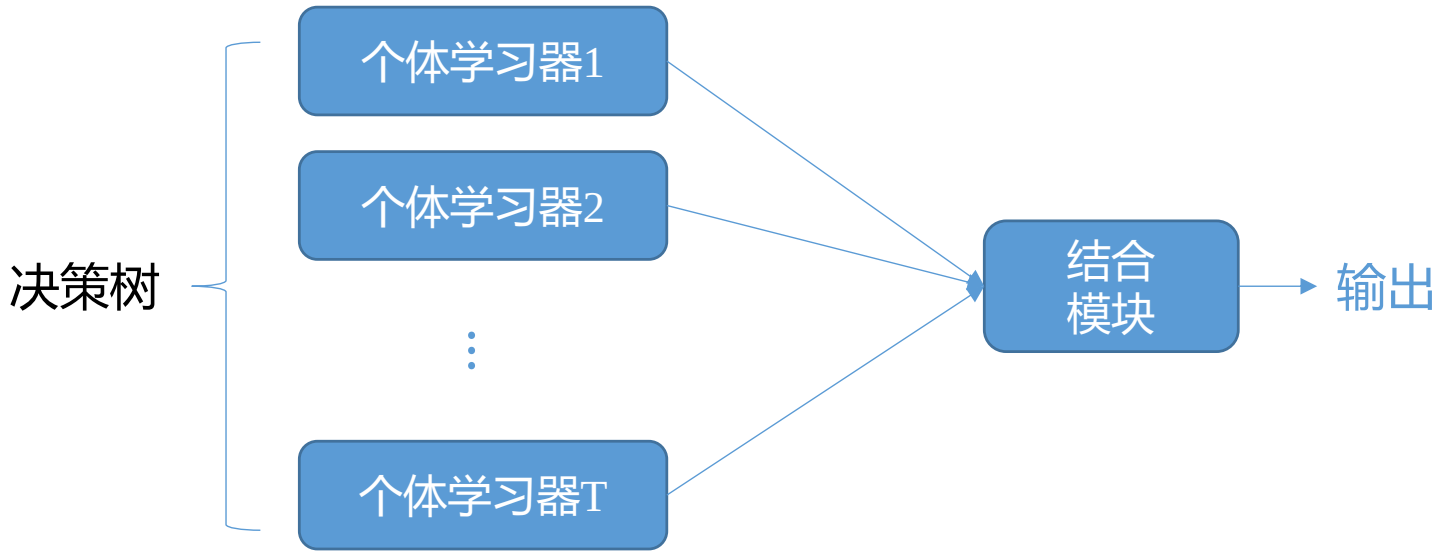
个体与集成

- 集成学习(ensemble learning)通过构建并结合多个学习器来提升性能



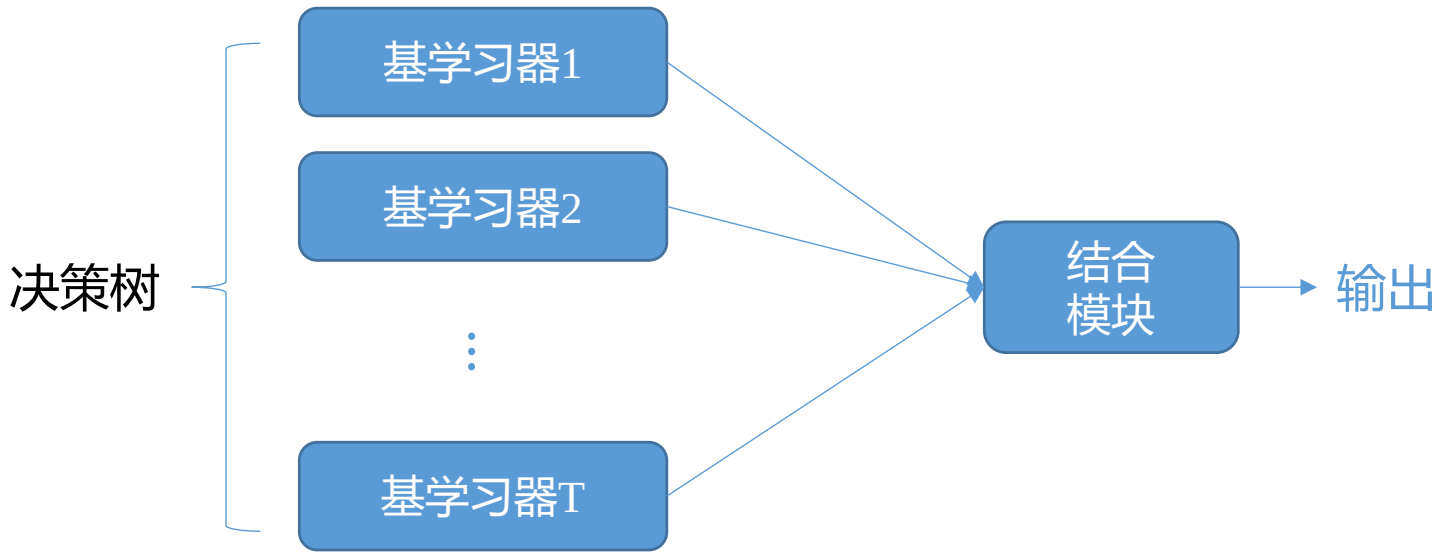
个体与集成

- 集成学习(ensemble learning)通过构建并结合多个学习器来提升性能



个体与集成

- 集成学习(ensemble learning)通过构建并结合多个学习器来提升性能

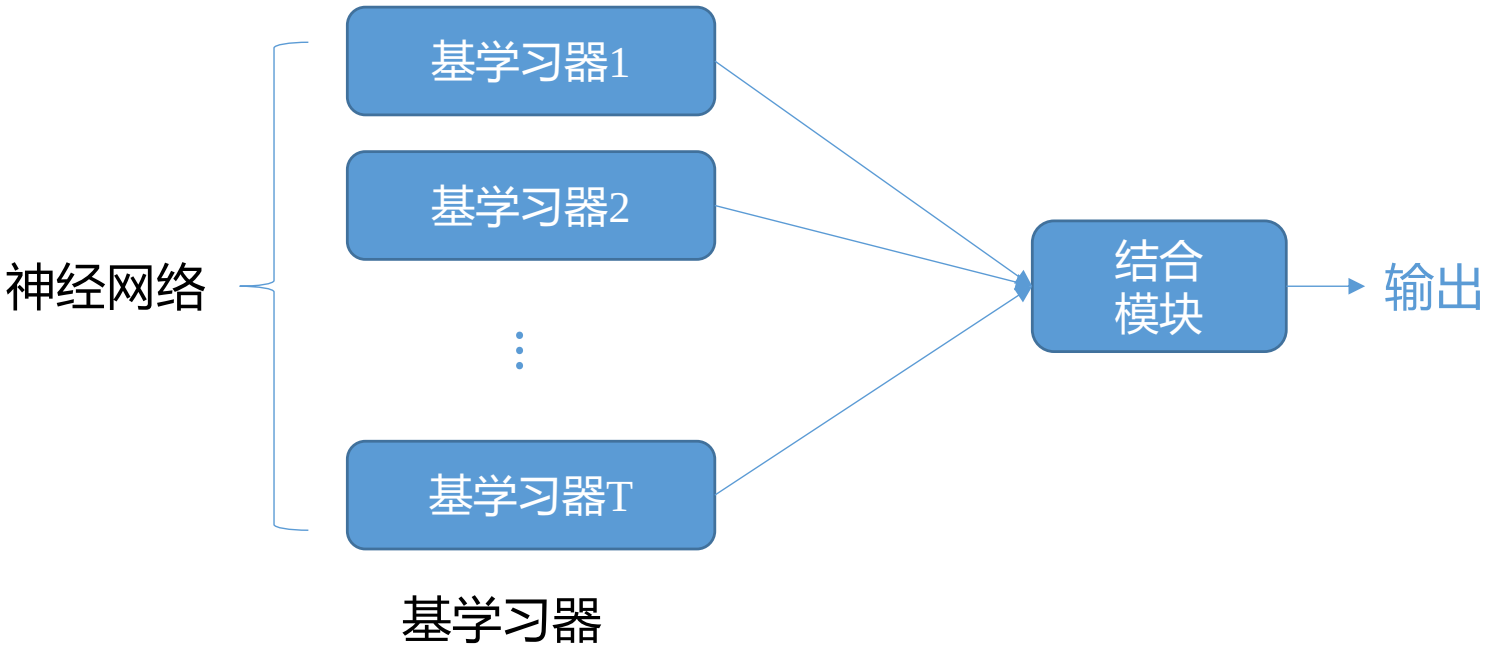


同质集成

决策树集成

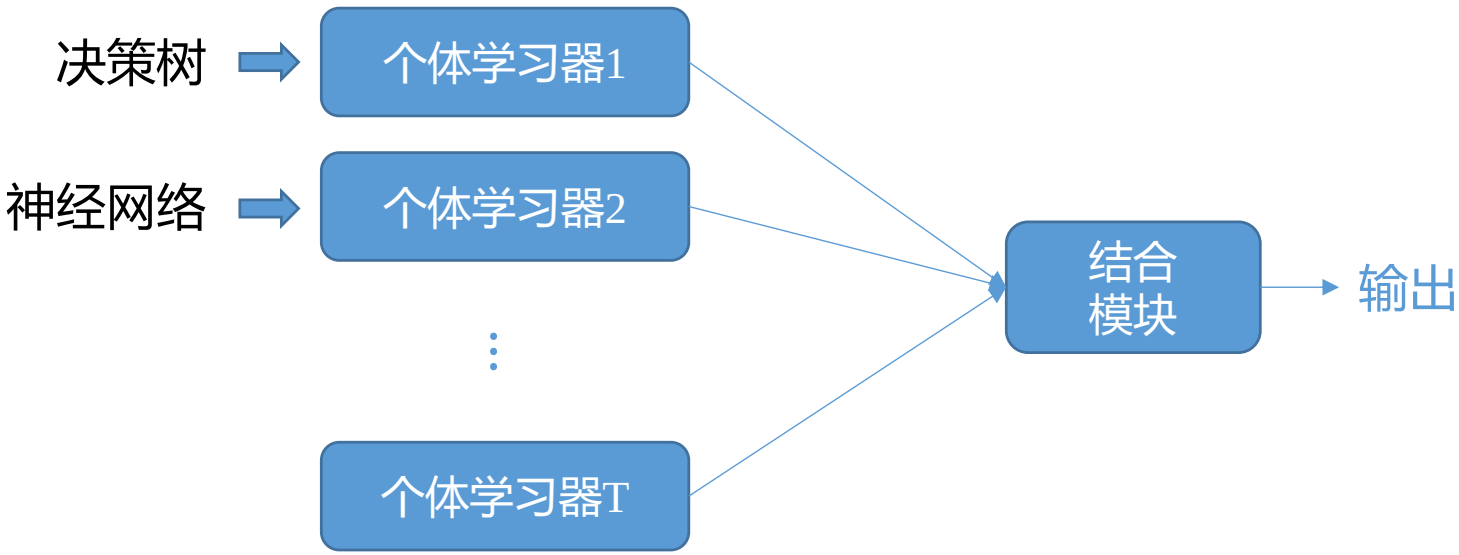
个体与集成

- 集成学习(ensemble learning)通过构建并结合多个学习器来提升性能



个体与集成

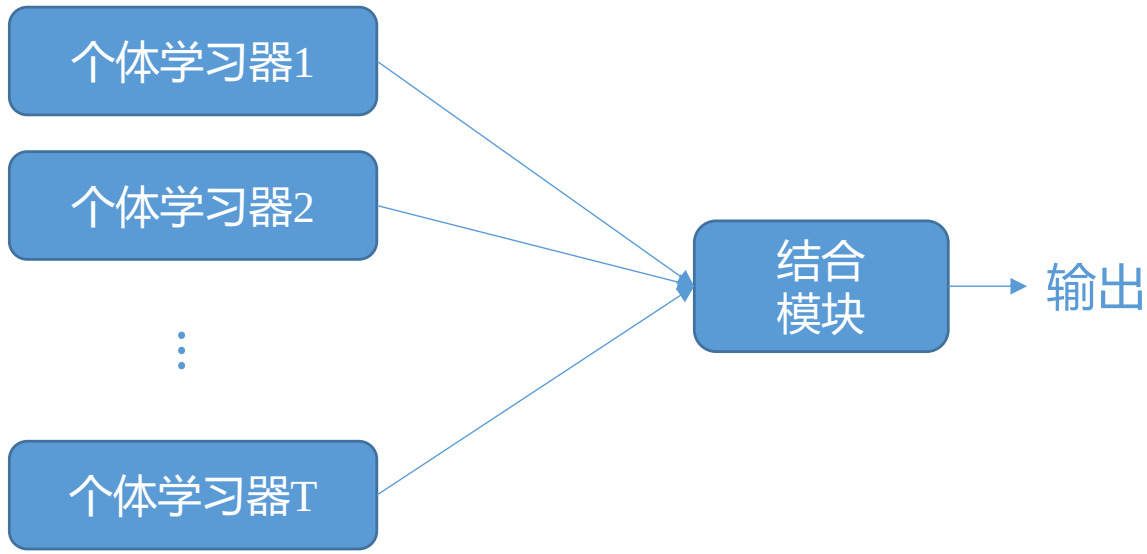
- 集成学习(ensemble learning)通过构建并结合多个学习器来提升性能



异质集成

个体与集成

- 集成学习(ensemble learning)通过构建并结合多个学习器来提升性能



集成学习获得比单一学习器显著优越的泛化性能，对弱学习器尤为明显。

集成学习很多理论研究都是针对弱学习器进行的

个体与集成

- 考虑一个简单的例子，在二分类问题中，假定3个分类器在三个样本中的表现如下图所示，其中√ 表示分类正确，X 号表示分类错误，集成的结果通过投票产生。

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	×	√	√
h_3	√	×	√
集群	√	√	√

集成提升性能

	测试例1	测试例2	测试例3
h_1	√	√	×
h_2	√	√	×
h_3	√	√	×
集群	√	√	×

集成不起作用

	测试例1	测试例2	测试例3
h_1	√	×	×
h_2	×	√	×
h_3	×	×	√
集群	×	×	×

集成起负作用

- 集成个体应：好而不同

个体与集成 – 简单分析

- 考虑二分类问题，假设基分类器的错误率为：

$$P\big(h(\boldsymbol{x}) \neq f(\boldsymbol{x})\big) = \epsilon$$

- 假设集成通过简单投票法结合T个分类器

$$H(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{i}^{T} h_i(\boldsymbol{x})\right)$$

- 假设基分类器的错误率相互独立，令$n(T)$表示T个基分类器中对$\boldsymbol{x}$预测正确的个数

若有超过半数的基分类器正确则分类就正确

$$P\big(H(\boldsymbol{x}) \neq f(\boldsymbol{x})\big) = P(n(T) \leq \lfloor T/2 \rfloor) = \sum_{k=0}^{\lfloor T/2 \rfloor} \binom{T}{k} (1-\epsilon)^k \epsilon^{T-k}$$

个体与集成 – 简单分析

- Hoeffding's inequality: $P\left(\frac{n(T)}{T} \leq \mathbb{E}\left[\frac{n(T)}{T}\right] - \delta\right) \leq \exp\left(-2\delta^2 T\right)$

$$P(n(T) \leq \mathbb{E}[n(T)] - T\delta) \leq \exp\left(-2\delta^2 T\right)$$

$$\mathbb{E}[n(T)] = T(1-\epsilon)$$

$$k = \mathbb{E}[n(T)] - T\delta$$

$$\Rightarrow \quad P(n(T) \leq k) \leq \exp\left(-2\frac{(\mathbb{E}[n(T)] - k)^2}{T}\right)$$

$$P\left(H(\boldsymbol{x}) \neq f(\boldsymbol{x})\right) = P(n(T) \leq \lfloor T/2 \rfloor)$$

$$\leq P(n(T) \leq T/2)$$

$$\leq \exp\left(-2\frac{(\mathbb{E}[n(T)] - T/2)^2}{T}\right)$$

$$\leq \exp\left(-2\frac{(T - T\epsilon - T/2)^2}{T}\right) \leq \exp\left(-\frac{1}{2}T(1-2\epsilon)^2\right)$$

在一定条件下，随着集成分类器数目的增加，集成的错误率将指数级下降，最终趋向于0

个体与集成 – 简单分析

- 上面的分析有一个关键假设： 基学习器的误差相互独立
- 现实任务中，个体学习器是为解决同一个问题训练出来的，显然不可能互相独立！
- 事实上，个体学习器的“准确性”和“多样性”本身就存在冲突

如何产生“好而不同”的个体学习器是集成学习研究的核心

集成学习

集成学习大致可分为两大类

个体学习器之间存在强依赖关系，必须串行生成的序列化方法

- 代表性方法是Boosting

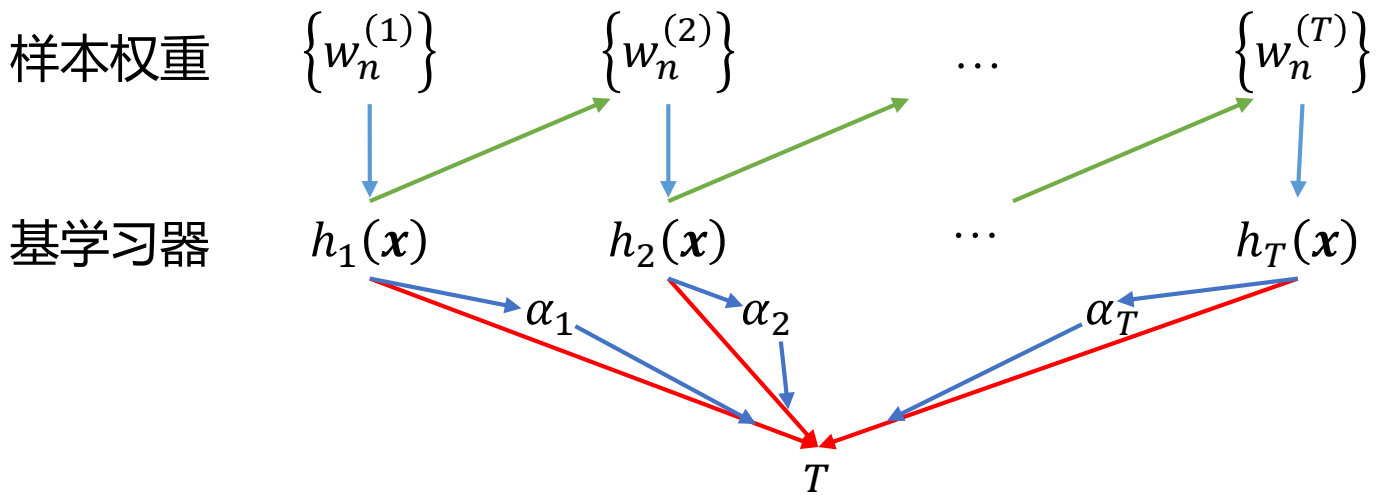
个体学习器不存在强依赖关系，可同时生成的并行化方法

- 代表性方法是Bagging和随机森林

Boosting

- 个体学习器存在强依赖关系，串行生成
- 每次调整训练数据的样本分布

$$D = \{(\boldsymbol{x}_1, y_1, w_1), (\boldsymbol{x}_2, y_2, w_2) \cdots, (\boldsymbol{x}_m, y_m, w_m)\}$$



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

AdaBoost

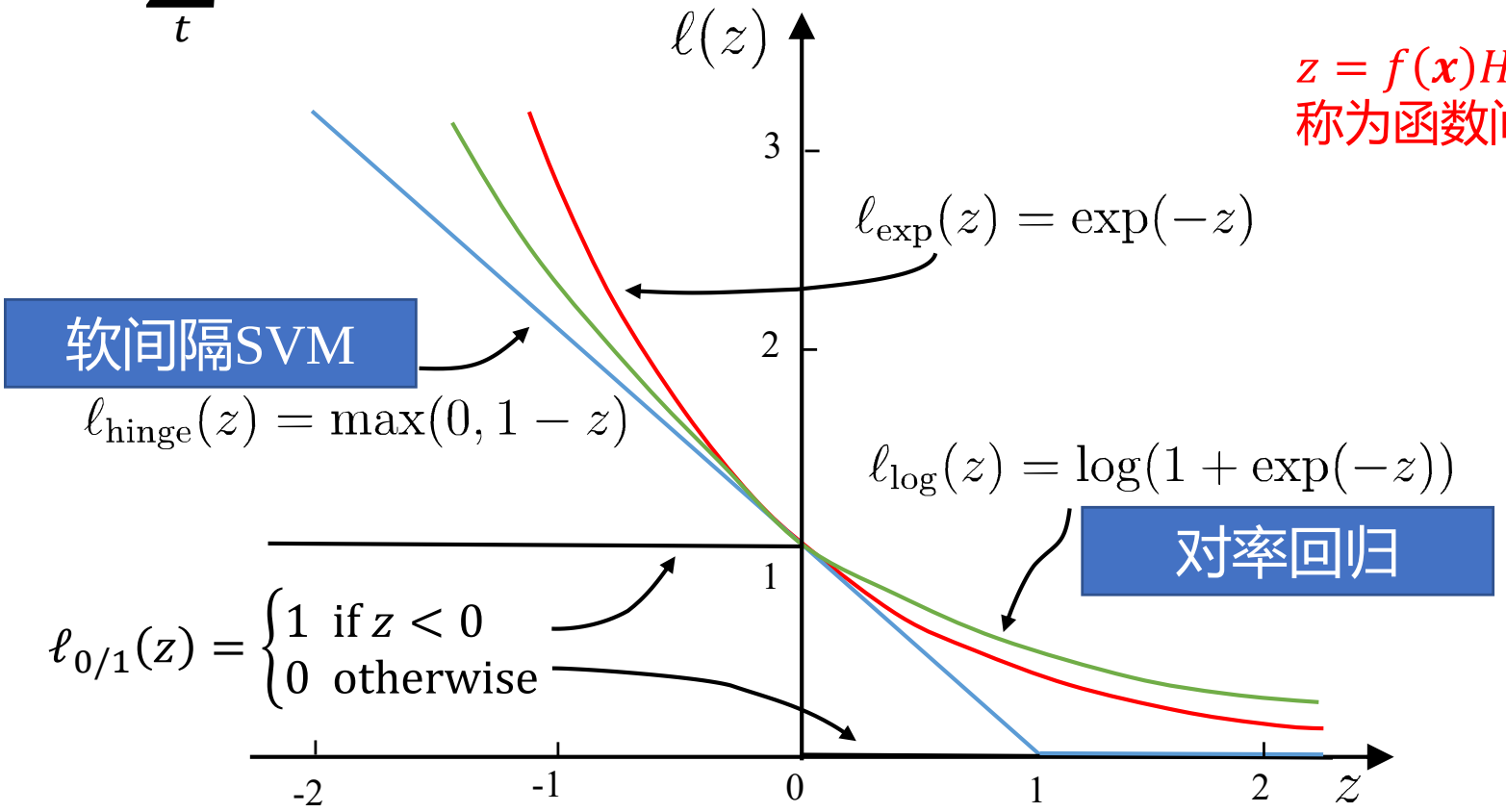
- 基学习器的线性组合

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

- 最小化指数损失函数

$$\ell(H|D) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\exp\left(-f(\boldsymbol{x})H(\boldsymbol{x})\right)\right]$$

$z = f(\boldsymbol{x})H(\boldsymbol{x})$
称为函数间隔



指数损失函数的一致性

- 假设$f(\boldsymbol{x})$具不确定性，即$y = f(\boldsymbol{x})$随机变量，则损失写成

$$\begin{aligned}\ell(H|D) &= \mathbb{E}_{\boldsymbol{x},y}\left[\exp\left(-yH(\boldsymbol{x})\right)\right] \\ &= \mathbb{E}_{\boldsymbol{x},y}\left[\mathbb{I}(y=1)\exp\left(-H(\boldsymbol{x})\right)+\mathbb{I}(y=-1)\exp\left(H(\boldsymbol{x})\right)\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{y|\boldsymbol{x}}\mathbb{I}(y=1)\exp\left(-H(\boldsymbol{x})\right)+\mathbb{E}_{y}\mathbb{I}(y=-1)\exp\left(H(\boldsymbol{x})\right)\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\left[P(y=1|\boldsymbol{x})\exp\left(-H(\boldsymbol{x})\right)+P(y=-1|\boldsymbol{x})\exp\left(H(\boldsymbol{x})\right)\right]\end{aligned}$$

- 若$H(\boldsymbol{x})$能令指数损失函数最小化，则上式对$H(\boldsymbol{x})$的偏导值为0

$$\frac{\partial\ell(H|D)}{\partial H(\boldsymbol{x})} = P(\boldsymbol{x})\left(-P(y=1|\boldsymbol{x})\exp\left(-H(\boldsymbol{x})\right)+P(y=-1|\boldsymbol{x})\exp\left(H(\boldsymbol{x})\right)\right) = 0$$

$$\Rightarrow\quad P(y=1|\boldsymbol{x})\exp\left(-H(\boldsymbol{x})\right) = P(y=-1|\boldsymbol{x})\exp\left(H(\boldsymbol{x})\right)$$

$$\Rightarrow\quad H(\boldsymbol{x}) = \frac{1}{2}\ln\frac{P(y=1|\boldsymbol{x})}{P(y=-1|\boldsymbol{x})}$$

指数损失函数的一致性

$$H(\boldsymbol{x}) = \frac{1}{2}\ln\frac{P(y=1|\boldsymbol{x})}{P(y=-1|\boldsymbol{x})} \quad \Rightarrow \quad P(y=1|\boldsymbol{x}) = \frac{1}{1+\exp -2\cdot H(\boldsymbol{x})}$$

$$\operatorname{sign}\big(H(\boldsymbol{x})\big) = \begin{cases} +1, \text{if } P(y=1|\boldsymbol{x}) > P(y=-1|\boldsymbol{x}) \\ -1, \text{if } P(y=1|\boldsymbol{x}) < P(y=-1|\boldsymbol{x}) \end{cases}$$

$$\Rightarrow \quad \operatorname{sign}\big(H(\boldsymbol{x})\big) = \arg\max_{y\in\{\pm 1\}} P(y|\boldsymbol{x})$$

$\operatorname{sign}\big(H(\boldsymbol{x})\big)$达到了贝叶斯最优错误率

说明指数损失函数是分类任务原来0/1损失函数的一致的替代函数

AdaBoost

- 假设$f(\boldsymbol{x})$是确定性的，在第t步，优化以下损失函数

$$\ell(H_{t-1}+\alpha_t h_t|D)=\mathbb{E}_{\boldsymbol{x}}\left[\exp\left(-f(\boldsymbol{x})\left(H_{t-1}(\boldsymbol{x})+\alpha_t h_t(\boldsymbol{x})\right)\right)\right]$$

$$=\mathbb{E}_{\boldsymbol{x}}\left[\exp\left(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})\right)\exp\left(-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})\right)\right]$$

$$=\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\exp\left(-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})\right)\right]$$

$$=\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\exp(-\alpha_t)\,\mathbb{I}\left(f(\boldsymbol{x})=h_t(\boldsymbol{x})\right)+\bar{w}_t(\boldsymbol{x})\exp(\alpha_t)\,\mathbb{I}\left(f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right)\right]$$

$$=\exp(-\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x})=h_t(\boldsymbol{x})\right)\right]+\exp(\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right)\right]$$

$$D=\{(\boldsymbol{x}_1,y_1,w_1),\cdots,(\boldsymbol{x}_m,y_m,w_m)\}$$

样本权重 $\left\{w_n^{(t)}\right\}$

基学习器 $h_t(\boldsymbol{x})$

α_t

$$H_t(x)=\alpha_t h_t(\boldsymbol{x})+H_{t-1}(x)$$

AdaBoost

- 假设$f(\boldsymbol{x})$是确定性的，在第t步，优化以下损失函数（求解h_t）

$$\ell(H_{t-1}+\alpha_t h_t|D)=\exp(-\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x})=h_t(\boldsymbol{x})\right)\right]+\exp(\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right)\right]$$

$$\min_{h_t}\ell(H_{t-1}+\alpha_t h_t|D)=\exp(-\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\left(1-\mathbb{I}\left(f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right)\right)\right]+\exp(\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right)\right]$$

$$D=\{(\boldsymbol{x}_1,y_1,w_1),\cdots,(\boldsymbol{x}_m,y_m,w_m)\}$$

$$=\left(\exp(\alpha_t)-\exp(-\alpha_t)\right)\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right)\right]+\exp(-\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\right]$$

样本权重 $\left\{w_n^{(t)}\right\}$

当$\alpha_t>0$，$\exp(\alpha_t)-\exp(-\alpha_t)>0$

基学习器 $h_t(\boldsymbol{x})$

$\min_{h_t}\ell(H_{t-1}+\alpha_t h_t|D)$ 等价于 $\min_{h_t}\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x})\neq h_t(\boldsymbol{x})\right)\right]$

α_t

$$\bar{w}_t(\boldsymbol{x})=\exp\left(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})\right)$$

$$H_t(x)=\alpha_t h_t(\boldsymbol{x})+H_{t-1}(x)$$

AdaBoost

$$h_t^\star(\boldsymbol{x}) = \underset{h_t}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\overline{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x})\right)\right]$$

$$= \underset{h_t}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\frac{\overline{w}_t(\boldsymbol{x})}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\overline{w}_t(\boldsymbol{x})]}\mathbb{I}\left(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x})\right)\right] = \underset{h_t}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[\mathbb{I}\left(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x})\right)\right]$$

$$\mathcal{D}_t(\boldsymbol{x}) = \mathcal{D}(\boldsymbol{x})\frac{\overline{w}_t(\boldsymbol{x})}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\overline{w}_t(\boldsymbol{x})]} = \mathcal{D}(\boldsymbol{x})\frac{\exp\left(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})\right)}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\exp\left(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})\right)\right]}$$

$$= \mathcal{D}(\boldsymbol{x})\frac{\exp\left(-f(\boldsymbol{x})H_{t-2}(\boldsymbol{x})\right)\exp\left(-\alpha_{t-1}f(\boldsymbol{x})h_{t-1}(\boldsymbol{x})\right)}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\exp\left(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})\right)\right]}$$

$$= \mathcal{D}_{t-1}(\boldsymbol{x})\exp\left(-\alpha_{t-1}f(\boldsymbol{x})h_{t-1}(\boldsymbol{x})\right)\frac{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\exp\left(-f(\boldsymbol{x})H_{t-2}(\boldsymbol{x})\right)\right]}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\exp\left(-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})\right)\right]}$$

$$= \frac{\mathcal{D}_{t-1}(\boldsymbol{x})\exp\left(-\alpha_{t-1}f(\boldsymbol{x})h_{t-1}(\boldsymbol{x})\right)}{Z_{t-1}}$$

AdaBoost

$$h_t^\star(\boldsymbol{x}) = \underset{h_t}{\operatorname{argmin}}\, \mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x})\right)\right]$$

- 假设$f(\boldsymbol{x})$是确定性的，在第t步，优化以下损失函数（求解α_t）

$$\ell(H_{t-1} + \alpha_t h_t^\star | D) = (\exp(\alpha_t) - \exp(-\alpha_t))\mathbb{E}_{\boldsymbol{x}}\left[\bar{w}_t(\boldsymbol{x})\mathbb{I}\left(f(\boldsymbol{x}) \neq h_t^\star(\boldsymbol{x})\right)\right] + \exp(-\alpha_t)\,\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})]$$

$$\mathbb{E}_{\boldsymbol{x}}\left[\frac{\bar{w}_t(\boldsymbol{x})}{\mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})]}\mathbb{I}\left(f(\boldsymbol{x}) \neq h_t^\star(\boldsymbol{x})\right)\right] = \epsilon_t$$

$$= \mathbb{E}_{\boldsymbol{x}}[\bar{w}_t(\boldsymbol{x})]\left((\exp(\alpha_t) - \exp(-\alpha_t))\epsilon_t + \exp(-\alpha_t)\right)$$

$$D = \{(\boldsymbol{x}_1, y_1, w_1), \cdots, (\boldsymbol{x}_m, y_m, w_m)\}$$

样本权重 $\left\{w_n^{(t)}\right\}$

基学习器 $h_t(\boldsymbol{x})$

$$\frac{\partial \ell(H_{t-1} + \alpha_t h_t^\star | D)}{\partial \alpha_t} = 0$$

$$\Rightarrow (\exp(\alpha_t) + \exp(-\alpha_t))\epsilon_t - \exp(-\alpha_t) = 0$$

$$\Rightarrow (\exp(2\alpha_t) + 1)\epsilon_t - 1 = 0$$

$$\Rightarrow \exp(2\alpha_t) = \frac{1-\epsilon_t}{\epsilon_t} \Rightarrow \alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$$

$$H_t(x) = \alpha_t h_t(\boldsymbol{x}) + H_{t-1}(x)$$

AdaBoost

输入: 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程:

1: $\mathcal{D}_1(\boldsymbol{x}) = 1/m$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: $\quad h_t = \mathfrak{L}(D, \mathcal{D}_t)$;
4: $\quad \epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$;
5: $\quad$ **if** $\epsilon_t > 0.5$ **then break**
6: $\quad \alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$;
7: $\quad \mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases}$
$= \frac{\mathcal{D}_t(\boldsymbol{x}) \exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$
8: **end for**

输出: $H(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \right)$

AdaBoost

- Boosting算法要求基学习器能对特定的数据分布进行学习

重赋权法：在每一轮训练，根据样本分布为每个样本赋一个权重

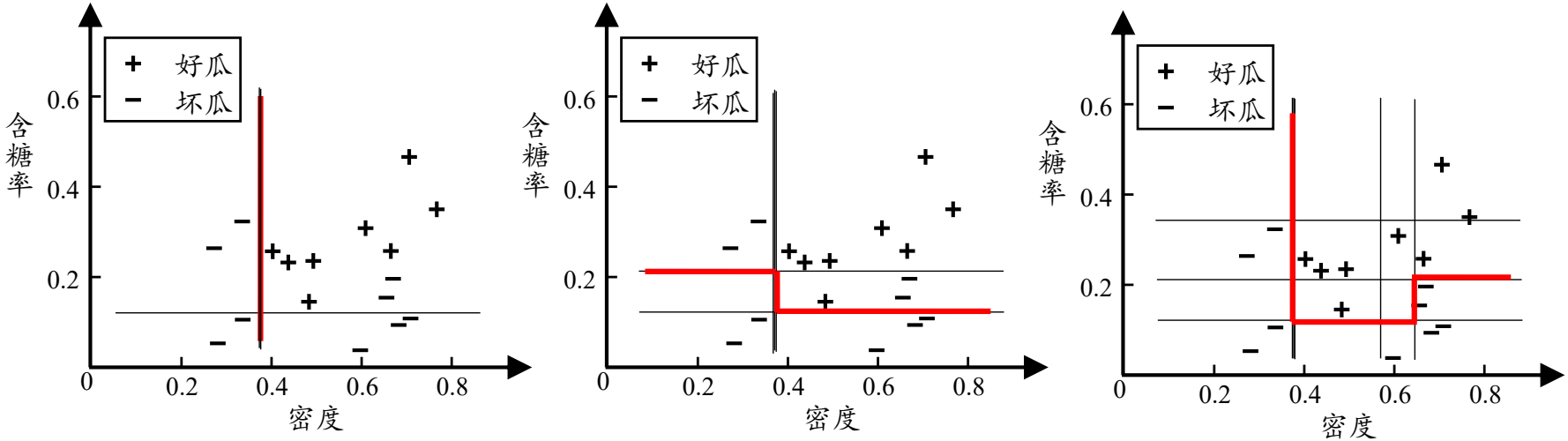
- 检查满足泛化误差大于0.5，条件不满足，就会被放弃，且学习会停止

重采样法：在每一轮学习，根据样本分布对训练集进行重采样，再用重采样而得的样本集对基学习器进行训练

- 获得重启动机会一避免训练过程过早停止：在抛弃不满足条件的当前基学习器之后，可根据当前分布重新对训练样本进行采样，再基于新的采样结果重新训练处基学习器。

AdaBoost

- 从偏差-方差的角度：降低偏差，可对泛化性能相当弱的学习器构造出很强的集成

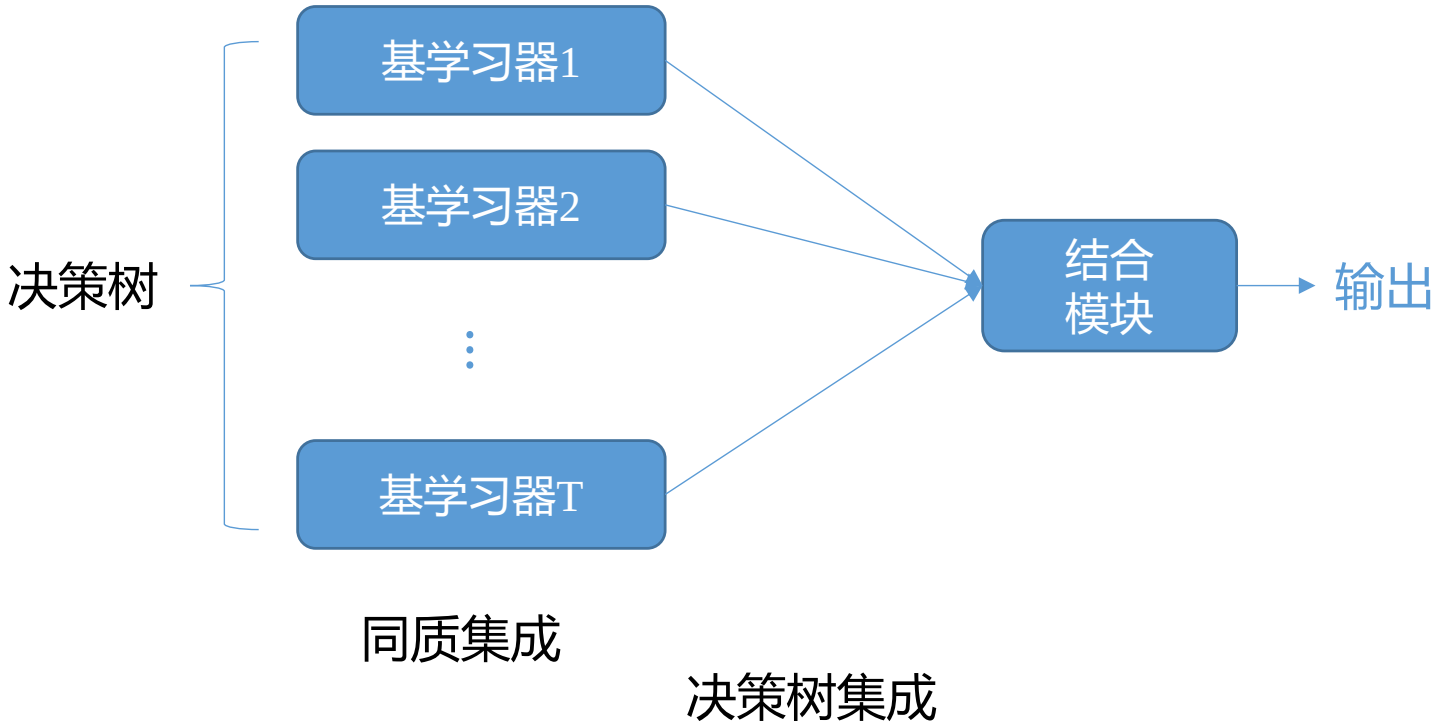


(a) 3个基学习器　(b) 5个基学习器　(c) 11个基学习器

集成模型（红色）和基学习器（黑色）的分类边界

梯度提升树

- 基于决策树的Boosting集成



梯度提升树

Adaboost

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

先训练学习器再学习权重

$$\min_{H(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{x},y}\left[\exp\left(-yH(\boldsymbol{x})\right)\right]$$

适用于分类

Gradient Boosting

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

同时学习权重和学习器

$$\min_{H(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{x},y}\left[\ell\left(y, H(\boldsymbol{x})\right)\right]$$

适用于回归和分类

梯度提升树

- 与Adaboost一样，通过前向分布法优化 $\min_{H(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{x},y}\left[\ell\left(y, H(\boldsymbol{x})\right)\right]$

将$H(\boldsymbol{x})$看过一个参数（泛函），$H(\boldsymbol{x})$的学习过程看成是一个梯度下降过程

$$H_t(\boldsymbol{x}) = H_{t-1}(\boldsymbol{x}) + h_t(\boldsymbol{x})$$

负梯度

$$h_t(\boldsymbol{x}) \approx -\frac{d\ell(y,\hat{y})}{d\hat{y}}\Big|_{\hat{y}=H_{t-1}(\boldsymbol{x})} = -\ell'\left(y, H_{t-1}(\boldsymbol{x})\right)$$

$$\ell\left(y, H(\boldsymbol{x})\right) = \left(y - H(\boldsymbol{x})\right)^2 \qquad -\ell'\left(y, H_{t-1}(\boldsymbol{x})\right) = y - H_{t-1}(\boldsymbol{x})$$

残差

梯度提升树

- 梯度提升树的算法流程

(1) 初始化 $h_0(\boldsymbol{x}) = \text{argmin}_\gamma \sum_{i=1}^{m} \ell(y_i, \gamma)$

(2) 对 t=1 to T

(a) 计算负梯度:$\tilde{y}_i = -\ell'\big(y_i, H_{t-1}(\boldsymbol{x}_i)\big)$， $i \in \{1,2,\dots m\}$

(b) 通过最小化平方误差，用基学习器$h_t(\boldsymbol{x})$根据$\boldsymbol{x}_i$拟合$\tilde{y}_i$

(c) 使用线搜索确定步长ρ_t，$\rho_t = \text{argmin}_{\rho_t} \sum_{i=1}^{m} \ell\big(y_i, H_{t-1}(\boldsymbol{x}) + \rho_t h_t(\boldsymbol{x})\big)$

(d) 更新$H_t(\boldsymbol{x}) = H_{t-1}(\boldsymbol{x}) + \rho_t h_t(\boldsymbol{x})$

(3) 输出$H_T(\boldsymbol{x})$

梯度提升树

- 回忆：回归树将空间划分为M个区域$R_1, R_2, \cdots, R_M$，那么回归树的决策函数为

$$h(\boldsymbol{x}) = \sum_{c=1}^{C} w_c I(\boldsymbol{x} \in R_c)$$

- 如果优化平方损失，在给定区域划分情况下，最优的c_m值是对应区域内的平均值

$$f_i = \ell'\big(y_i, H_{t-1}(\boldsymbol{x}_i)\big) \qquad s_i = \ell''\big(y_i, H_{t-1}(\boldsymbol{x}_i)\big)$$

- 提升树每次优化

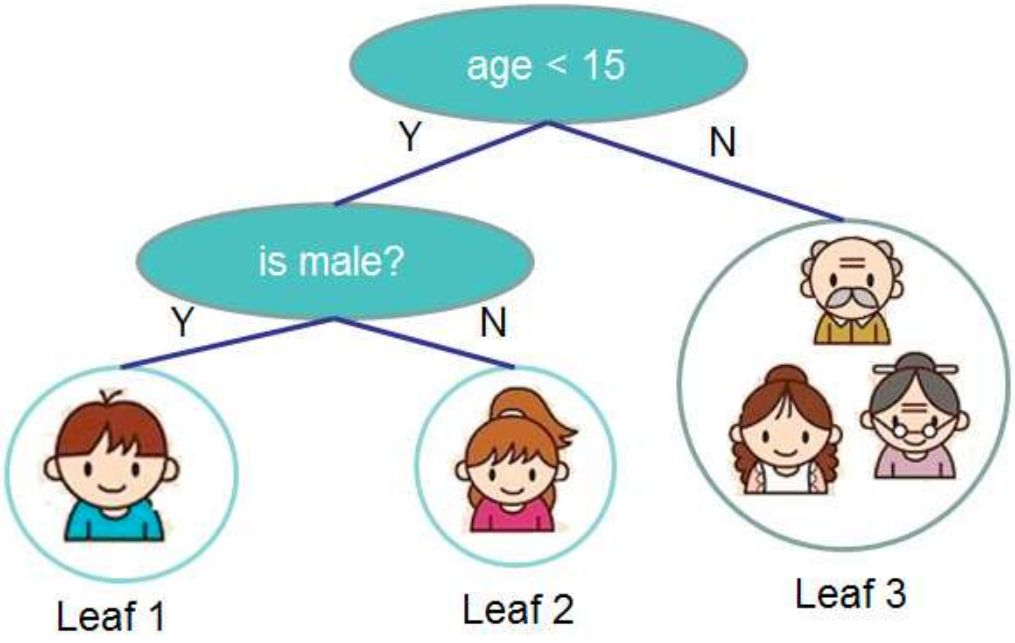
$$\begin{aligned} J^{(t)}\big(h_t(\boldsymbol{x}_i)\big) &= \sum_{i=1}^{m} \ell\big(y_i, H_{t-1}(\boldsymbol{x}_i) + h_t(\boldsymbol{x}_i)\big) + \Omega(h_t) \\ &\simeq \sum_{i=1}^{m} \left[\ell\big(y_i, H_{t-1}(\boldsymbol{x}_i)\big) + f_i h_t(\boldsymbol{x}_i) + \frac{1}{2} s_i h_t^2(\boldsymbol{x}_i)\right] + \Omega(h_t) \\ &= \sum_{i=1}^{m} \left[f_i h_t(\boldsymbol{x}_i) + \frac{1}{2} s_i h_t^2(\boldsymbol{x}_i)\right] + \Omega(h_t) + \text{const} \end{aligned}$$

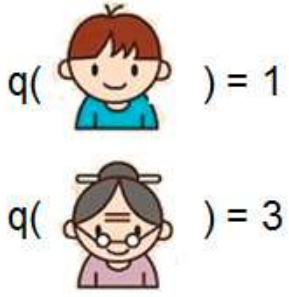
梯度提升树

$$h(\boldsymbol{x}) = \sum_{c=1}^{C} w_c I(\boldsymbol{x} \in R_c)$$

树的复杂度可以定义为

$$\Omega(h) = \gamma C + \frac{1}{2}\lambda \sum_{c=1}^{C} w_c^2$$





w_1 w_2 w_3

梯度提升树

$$h(\boldsymbol{x}) = \sum_{c=1}^{C} w_c I(\boldsymbol{x} \in R_c)$$

$$J^{(t)}\big(h_t(\boldsymbol{x}_i)\big) \simeq \sum_{i=1}^{m}\left[f_i h_t(\boldsymbol{x}_i) + \frac{1}{2} s_i h_t^2(\boldsymbol{x}_i)\right] + \Omega(h_t) + \text{const}$$

$$\sum_{i=1}^{m} f_i \sum_{c=1}^{C} w_c I(\boldsymbol{x}_i \in R_c) = \sum_{i=1}^{m}\left[f_i h_t(\boldsymbol{x}_i) + \frac{1}{2} s_i h_t^2(\boldsymbol{x}_i)\right] + \gamma C + \frac{1}{2}\lambda \sum_{c=1}^{C} w_c^2 + \text{const}$$

$$\sum_{c=1}^{C} w_c \sum_{i=1}^{m} f_i\, I(\boldsymbol{x}_i \in R_c) = \sum_{c=1}^{C}\left[\left(\sum_{i \in R_c} f_i\right) w_c + \frac{1}{2}\left(\sum_{i \in R_c} s_i + \lambda\right) w_c^2\right] + \gamma C + \text{const}$$

$$= \sum_{c=1}^{C}\left[F_c w_c + \frac{1}{2}(S_c + \lambda) w_c^2\right] + \gamma C + \text{const}$$

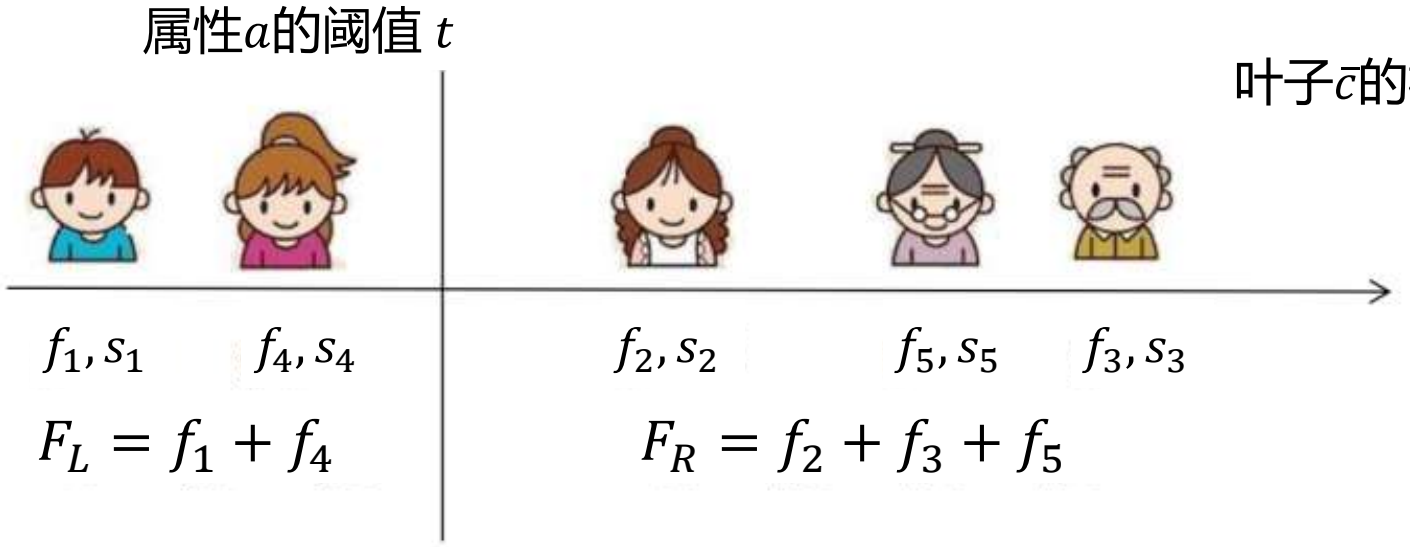
给定在给定区域划分情况下

$$w_c^{\star} = -\frac{F_c}{S_c + \lambda}$$

$$J^{(t)}\big(h_t(\boldsymbol{x}_i)\big) = -\frac{1}{2}\sum_{c=1}^{C}\frac{F_c^2}{S_c + \lambda} + \gamma C$$

该树与训练数据的适合度度量，可用于指导分类树的构建

梯度提升树



- 划分点选择准则

$\frac{(F_L+F_R)^2}{S_L+S_r+\lambda}$

- 划分前：$-\frac{1}{2}\sum_{c=1,c\neq\bar{c}}^{C}\frac{F_c^2}{S_c+\lambda}-\frac{1}{2}\frac{F_{\bar{c}}^2}{S_{\bar{c}}+\lambda}+\gamma C$（假设划分叶子为$\bar{c}$）
- 划分后：$-\frac{1}{2}\sum_{c=1,c\neq\bar{c}}^{C}\frac{F_c^2}{S_c+\lambda}-\frac{1}{2}\frac{F_L^2}{S_L+\lambda}-\frac{1}{2}\frac{F_R^2}{S_R+\lambda}+\gamma(C+1)$

$$\text{gain}(D,a,t)=\frac{1}{2}\left(\frac{F_L^2}{S_L+\lambda}+\frac{F_R^2}{S_R+\lambda}-\frac{(F_L+F_R)^2}{S_L+S_r+\lambda}\right)-\gamma$$

划分为左子树L和右子树R

梯度提升树

- GBDT的知名算法库

- XGBoost

dmlc XGBoost eXtreme Gradient Boosting

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

https://github.com/dmlc/xgboost

- LightGBM

LightGBM

Light Gradient Boosting Machine

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems* (pp. 3146-3154).

https://github.com/microsoft/LightGBM

Bagging

- 个体学习器不存在强依赖关系、并行化生成
- 基学习器尽可能具有较大的差异

对训练样本进行采样，产生出不同的子集，再从每个子集中训练出一个基学习器

若采样出完全不同的子集，每个基学习器只用到一小部分训练数据，可能不足以进行有效学习，无法确保基学习器的性能

Bagging

- 自助采样法

对数据集 D 有放回采样 m 次得到训练集D'

样本在m次采样中始终不被采样到的概率 $= \lim_{m\to\infty}\left(1-\frac{1}{m}\right)^{m} = \frac{1}{e} \approx 0.368$

初始训练集中约有63.2%样本出现在采样集中

Bagging

输入： 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程：

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad h_t = \mathfrak{L}(D, \mathcal{D}_{bs})$
3: **end for**

输出： $H(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{t=1}^{T} \mathbb{I}(h_t(\boldsymbol{x}) = y)$

对分类任务使用简单投票法 **结合策略** 对回归任务使用简单平均法

Bagging

- 时间复杂度低
  - 假定个体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))
  - 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

- 由于个体学习器并未使用全部训练数据，因此其余数据可用作验证集对泛化性能进行包外估计（out-of-bag estimate）

- $H^{oob}(\boldsymbol{x})$表示对样本$\boldsymbol{x}$的包外预测，即仅考虑那些未使用样本$\boldsymbol{x}$训练的基学习器在$\boldsymbol{x}$上的预测

$$H^{oob}(\boldsymbol{x}) = \operatorname{argmax}_{y \in \mathcal{Y}} \sum_{i}^{T} \mathbb{I}(h_t(\boldsymbol{x}) = y) \cdot \mathbb{I}(\boldsymbol{x} \notin D_t)$$

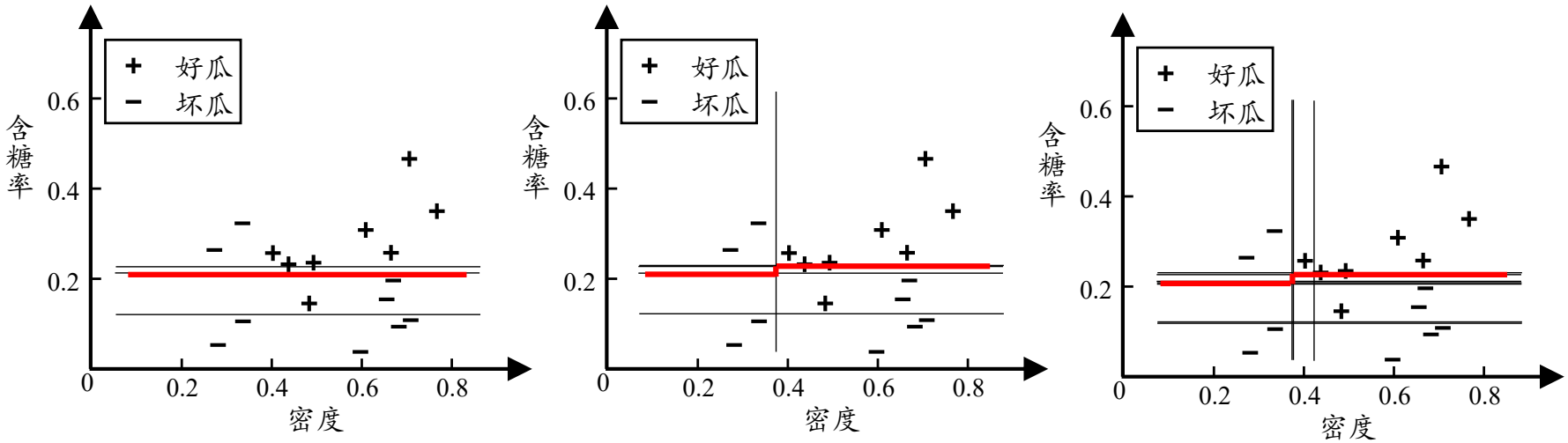
- Bagging泛化误差的包外估计为：

$$\epsilon^{oob} = \frac{1}{|D|} \sum_{(x,y) \in D} \mathbb{I}\left(H^{oob}(\boldsymbol{x}) = y\right)$$

- 包外样本还可以用于决策树剪枝、神经网络早停等

Bagging

从偏差-方差的角度：降低方差，在不剪枝的决策树、神经网络等易受样本影响的学习器上效果更好



(a) 3个基学习器　(b) 5个基学习器　(c) 11个基学习器

随机森林

- 随机森林(Random Forest，简称RF)是bagging的一个扩展变种
- 在采样的随机性基础上，进一步引入了属性选择的随机性

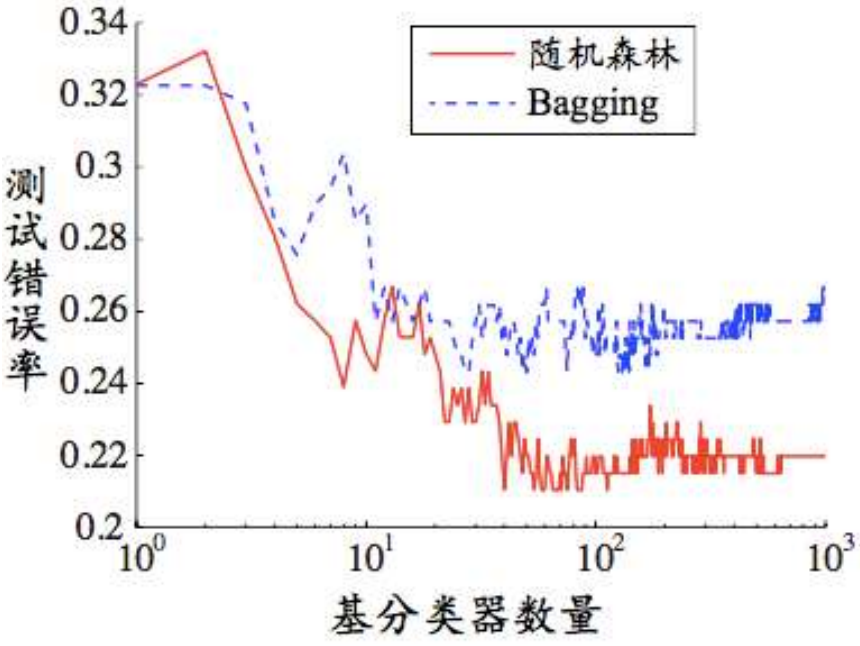
传统决策树在选择划分属性时是在当前节点的属性集合（假定d个属性）中选择一个最优属性

对基决策树的每个节点，先从该节点的属性集合中随机选择一个包含k个属性的子集，然后再从这个子集中选择一个最优属性用于划分。 $k = \log_2 d$

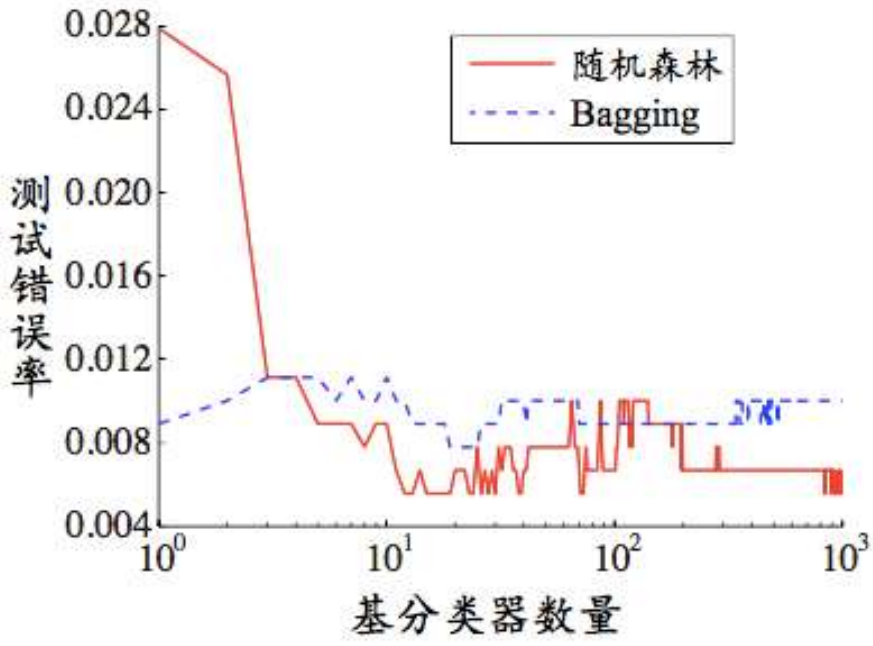
属性扰动使得个体学习器相关性进一步减弱，提升了泛化性能

随机森林简单、易实现、计算开销小，很多任务展现强大性能，被誉为“代表集成学习水平的方法”

随机森林



(a) glass 数据集

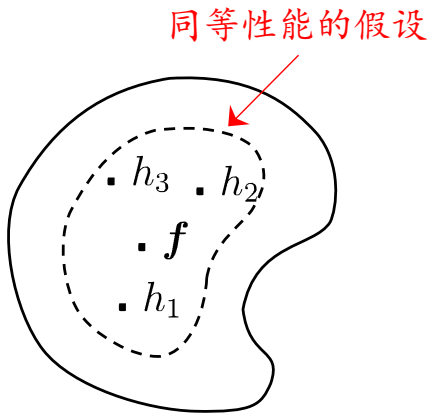


(b) auto-mpg 数据集

随着基分类器数目增加，随机森林通常会收敛到更低的泛化误差

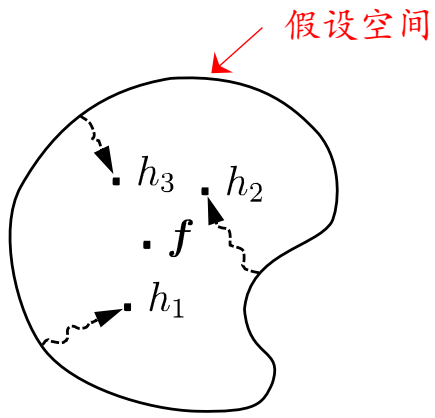
结合策略

- 学习器的组合可以从三个方面带来好处



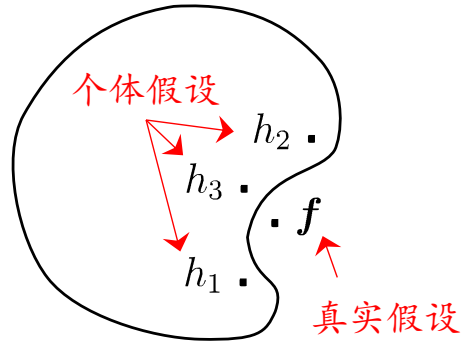
(a) 统计的原因

学习任务假设空间大，多个假设在训练集上可能达到同等性能，结合多个学习器会减小方差



(b) 计算的原因

学习算法往往会陷入局部极小值，有的局部极小点对应的泛化性能可能差，通过多次运行后进行结合，减低陷入差极小点的风险



(c) 表示的原因

某些学习任务的真实假设可能不在当前学习算法的假设空间中，结合多个学习器，可以扩大假设空间，学得更好的近似

结合策略—回归

- 简单平均法

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

- 加权平均法

$$H(\boldsymbol{x}) = \sum_{i}^{T} w_i h_i(\boldsymbol{x}) \qquad w_i \geq 0 \text{ and } \sum_{i} w_i = 1$$

- 加权平均不一定优于简单平均

结合策略—分类

- 绝对多数投票法（majority voting）假设N个类别

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

- 相对多数投票法（plurality voting）

$$H(\boldsymbol{x}) = c_{\text{argmax}_j \sum_{i=1}^{T} h_i^j(\boldsymbol{x})}$$

- 加权投票法（weighted voting）

$$H(\boldsymbol{x}) = c_{\text{argmax}_j \sum_{i=1}^{T} w_i h_i^j(\boldsymbol{x})} \qquad w_i \geq 0 \text{ and } \sum_i w_i = 1$$

结合策略—学习法

- Stacking是学习法的典型代表

先从训练数据集训练出初级学习器，然后生成一个新数据集用于训练次级学习器

Input: Data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
First-level learning algorithms $\mathfrak{L}_1, \ldots, \mathfrak{L}_T$;
Second-level learning algorithm $\mathfrak{L}$.

Process:

1. **for** $t = 1, \ldots, T$:
2. $h_t = \mathfrak{L}_t(D)$; ← 个体学习器称为初级学习器
3. **end**
4. $D' = \emptyset$;
5. **for** $i = 1, \ldots, m$:
6. **for** $t = 1, \ldots, T$:
7. $z_{it} = h_t(\boldsymbol{x}_i)$;
8. **end**
9. $D' = D' \cup ((z_{i1}, \ldots, z_{iT}), y_i)$; ← 生成一个新数据集
10. **end**
11. $h' = \mathfrak{L}(D')$; ← 用于结合的学习器称为元学习器

Output: $H(\boldsymbol{x}) = h'(h_1(\boldsymbol{x}), \ldots, h_T(\boldsymbol{x}))$

误差-分歧分解

- 对集成学习进行简单理论分析

- 定义学习器h_i的分歧

$$A(h_i|\boldsymbol{x}) = \left(h_i(\boldsymbol{x}) - H(\boldsymbol{x})\right)^2 \qquad H(\boldsymbol{x}) = \sum_i^T w_i h_i(\boldsymbol{x})$$

$$w_i \geq 0 \text{ and } \sum_i w_i = 1$$

- 集成的分歧

$$\bar{A}(h|x) = \sum_i^T w_i A(h_i|x) \quad = \sum_i^T w_i \left(h_i(\boldsymbol{x}) - H(\boldsymbol{x})\right)^2$$

分歧项代表了个体学习器在样本$\boldsymbol{x}$上的不一致性，即在一定程度上反映了个体学习器的多样性

误差-分歧分解

$$\bar{A}(h|\boldsymbol{x}) = \sum_i^T w_i\big(h_i(\boldsymbol{x}) - H(\boldsymbol{x})\big)^2 = \sum_i^T w_i\big(h_i(\boldsymbol{x}) - f(\boldsymbol{x}) + f(\boldsymbol{x}) - H(\boldsymbol{x})\big)^2$$

$$= \sum_i^T w_i\big(h_i(\boldsymbol{x}) - f(\boldsymbol{x})\big)^2 + \sum_i^T w_i\big(f(\boldsymbol{x}) - H(\boldsymbol{x})\big)^2 + 2\sum_i^T w_i\big(h_i(\boldsymbol{x}) - f(\boldsymbol{x})\big)\big(f(\boldsymbol{x}) - H(\boldsymbol{x})\big)$$

$$= \sum_i^T w_i\big(h_i(\boldsymbol{x}) - f(\boldsymbol{x})\big)^2 + \big(f(\boldsymbol{x}) - H(\boldsymbol{x})\big)^2 + 2\big(H(\boldsymbol{x}) - f(\boldsymbol{x})\big)\big(f(\boldsymbol{x}) - H(\boldsymbol{x})\big)$$

$$= \sum_i^T w_i\big(h_i(\boldsymbol{x}) - f(\boldsymbol{x})\big)^2 - \big(f(\boldsymbol{x}) - H(\boldsymbol{x})\big)^2$$

$$= \sum_i^T w_i E(h_i|\boldsymbol{x}) - E(H|\boldsymbol{x}) = \bar{E}(h|\boldsymbol{x}) - E(H|\boldsymbol{x})$$

$E(h_i|\boldsymbol{x})$集成h_i的平方误差

$E(H|\boldsymbol{x})$集成H的平方误差

$\bar{E}(h|\boldsymbol{x})$个体学习器误差的加权均值

误差-分歧分解

$$\bar{A}(h|\boldsymbol{x}) = \bar{E}(h|\boldsymbol{x}) - E(H|\boldsymbol{x})$$

- 上式对所有样本$\boldsymbol{x}$均成立，令$p(\boldsymbol{x})$表示样本的概率密度，则在全样本上有

$$\sum_{i=1}^{T} w_i \int A(h_i|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \sum_{i=1}^{T} w_i \int \bar{E}(h_i|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} - \int E(H|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$$

$$\sum_{i=1}^{T} w_i A_i = \sum_{i=1}^{T} w_i E_i - E$$

$$\boldsymbol{E} = \overline{\boldsymbol{E}} - \overline{\boldsymbol{A}}$$

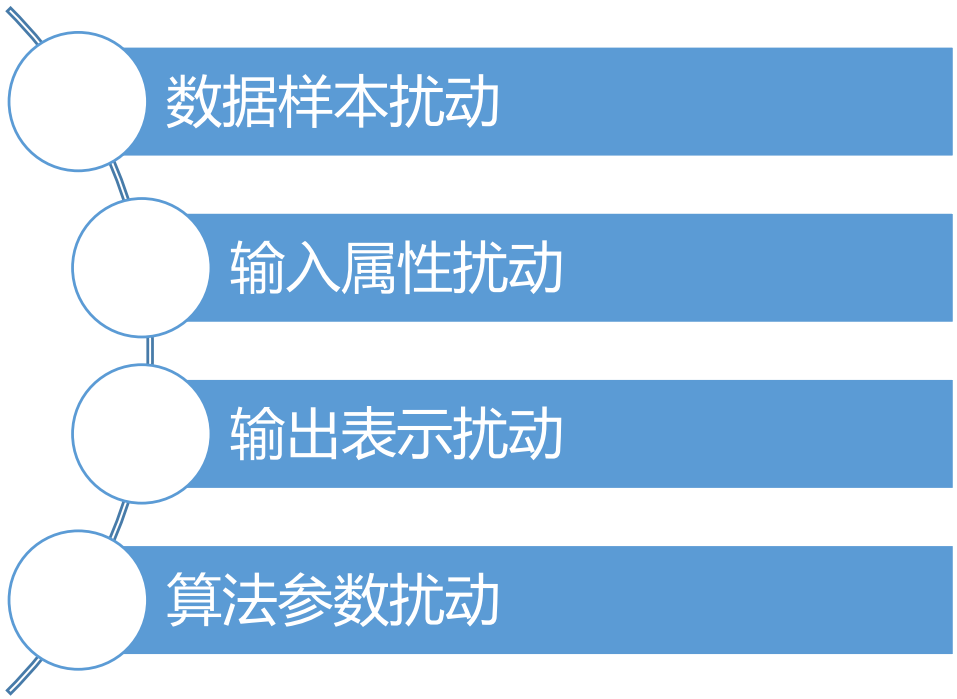
个体学习器精确性越高($\overline{\boldsymbol{E}}$小)、多样性越大($\overline{\boldsymbol{A}}$大)，则集成效果越好。

$$\overline{\boldsymbol{A}} = \overline{\boldsymbol{E}} - \boldsymbol{E}$$

个体学习器的加权分歧　　个体学习器的加权泛化误差　　集成的泛化误差

多样性增强

- 常见的增强个体学习器的多样性的方法



多样性增强—数据样本扰动

- 数据样本扰动通常是基于采样法
  - Bagging中的自助采样法
  - Adaboost中的序列采样

数据样本扰动对“不稳定基学习器”很有效

- 对数据样本的扰动敏感的基学习器(不稳定基学习器)
  - 决策树，神经网络等

- 对数据样本的扰动不敏感的基学习器(稳定基学习器)
  - 线性学习器，支持向量机，朴素贝叶斯，k近邻等

多样性增强—属性扰动

- 随机子空间算法(random subspace)
- 从初始属性集中抽取出若干个属性子集，在基于每个属性子集训练一个基学习器

输入： 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
基学习器数 T;
子空间属性数 d'.

过程：

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \mathcal{F}_t = \mathrm{RS}(D, d')$
3: $\quad D_t = \mathrm{Map}_{\mathcal{F}_t}(D)$
4: $\quad h_t = \mathfrak{L}(D_t)$
5: **end for**

输出： $H(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{t=1}^{T} \mathbb{I}\left(h_t\left(\mathrm{Map}_{\mathcal{F}_t}(\boldsymbol{x})\right) = y\right)$

多样性增强

- 输出扰动
  - 翻转法(Flipping Output)
  - 输出调剂法(Output Smearing)
  - ECOC法
- 负相关法
  - 显式地通过正则化来强制个体神经网络使用不同的参数

作业

- 习题8.2
- 习题8.6
- 习题8.8

第九章：聚类

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

机器学习分类

- 按照标记区分

- 分类：标记为离散值（二分类、多分类）
- 回归：标记为连续值（瓜的成熟度）

监督学习
Supervised Learning

- 聚类：没有标记

无监督学习
Unsupervised Learning

聚类：将数据集中的样本划分为若干个通常是不相交的子集

每个子集称为一个簇（cluster）

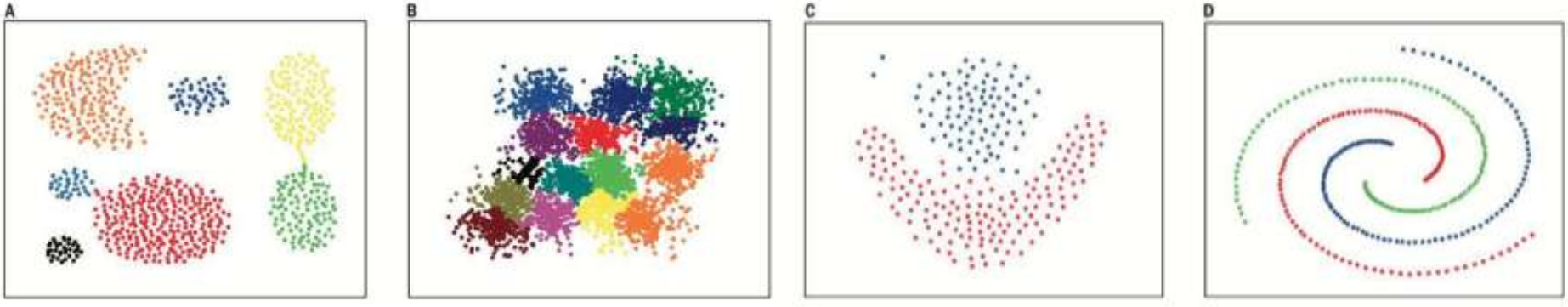
浅色瓜、深色瓜
有籽瓜、无籽瓜
本地瓜、外地瓜

聚类

- 聚类：将数据集中的样本划分为若干个通常是不相交的子集

- 形式化地说，假定样本集$D = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$包含m个无标记样本，聚类算法将样本集$D$划分为k个不相交的簇$\{C_l | l = 1,2, \ldots, k\}$，其中$C_{l'} \cap C_l = \emptyset, \forall l' \neq l$ 且$D = \cup_{i=1}^{k} C_l$

- 用$\lambda_j \in \{1,2, \ldots, k\}$表示样本$\boldsymbol{x}_j$的簇标记（cluster label），即$\boldsymbol{x}_j \in C_{\lambda_j}$



性能度量

- 聚类性能度量，亦称为聚类“有效性指标”（validity index）
- 直观来讲：

希望“物以类聚”，即同一簇的样本尽可能彼此相似，不同簇的样本尽可能不同。换言之，聚类结果的“簇内相似度”（intra-cluster similarity）高，且“簇间相似度”（inter-cluster similarity）低，这样的聚类效果较好

性能度量

- 聚类性能度量：

  - 外部指标 (external index)
    将聚类结果与某个“参考模型” (reference model)进行比较

  - 内部指标 (internal index)
    直接考察聚类结果而不用任何参考模型

性能度量—外部指标

- 对$D = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$，通过聚类给出的簇划分为$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$
- 参考模型给出的簇划分为$\mathcal{C}^\star = \{C_1^\star, C_2^\star, \ldots, C_k^\star\}$
- 令λ与$\lambda^\star$分别表示与C和$C^\star$对应的簇标记向量

性能度量—外部指标

- 将样本两两配对

$$SS = \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^\star = \lambda_j^\star, i < j\}, \quad a = |SS|$$
$$SD = \{(x_i, x_j) | \lambda_i = \lambda_j, \lambda_i^\star \neq \lambda_j^\star, i < j\}, \quad b = |SS|$$
$$DS = \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^\star = \lambda_j^\star, i < j\}, \quad c = |SS|$$
$$DD = \{(x_i, x_j) | \lambda_i \neq \lambda_j, \lambda_i^\star \neq \lambda_j^\star, i < j\}, \quad d = |SS|$$

- Jaccard系数（Jaccard Coefficient, JC）

$$JC = \frac{a}{a+b+c}$$

- FM指数（Fowlkes and Mallows Index, FMI）

$$FMI = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$$

- Rand指数（Rand Index, RI）

$$RI = \frac{2(a+b)}{m(m-1)}$$

上述性能度量的结果值均在[0,1]之间，值越大越好

性能度量—内部指标

- 考虑聚类结果的簇划分$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

簇内样本平均距离

$$avg(C) = \frac{1}{|C|}\sum_{i=1}^{|C|} dist(\boldsymbol{\mu}, \boldsymbol{x_i})$$

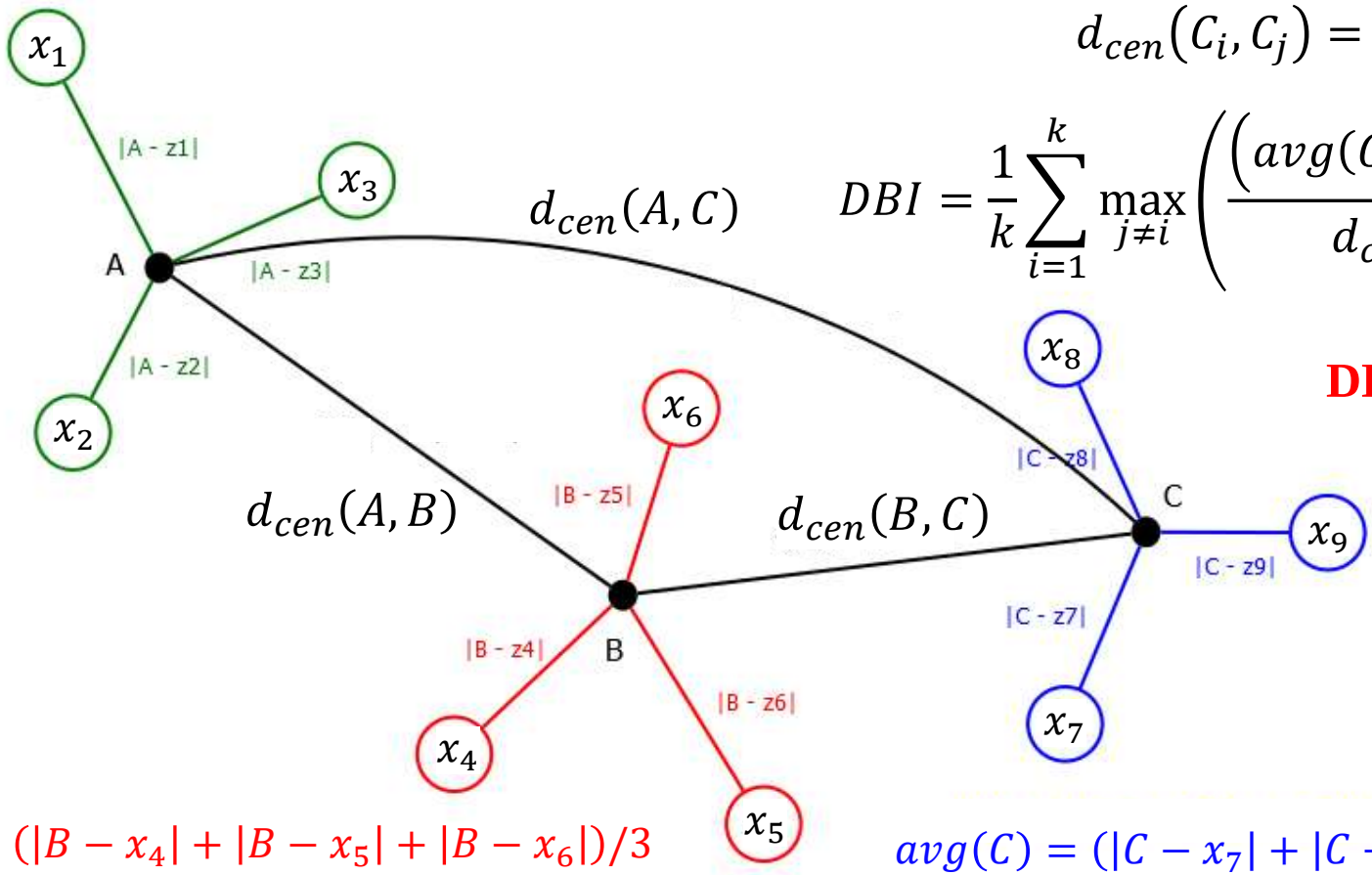
簇中心点的距离

$$d_{cen}(C_i, C_j) = dist(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$$

$$DBI = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\left(\frac{\left(avg(C_i) + avg(C_j)\right)}{d_{cen}(C_i, C_j)}\right)$$

DBI越小越好

$avg(A) = (|A - x_1| + |A - x_2| + |A - x_3|)/3$



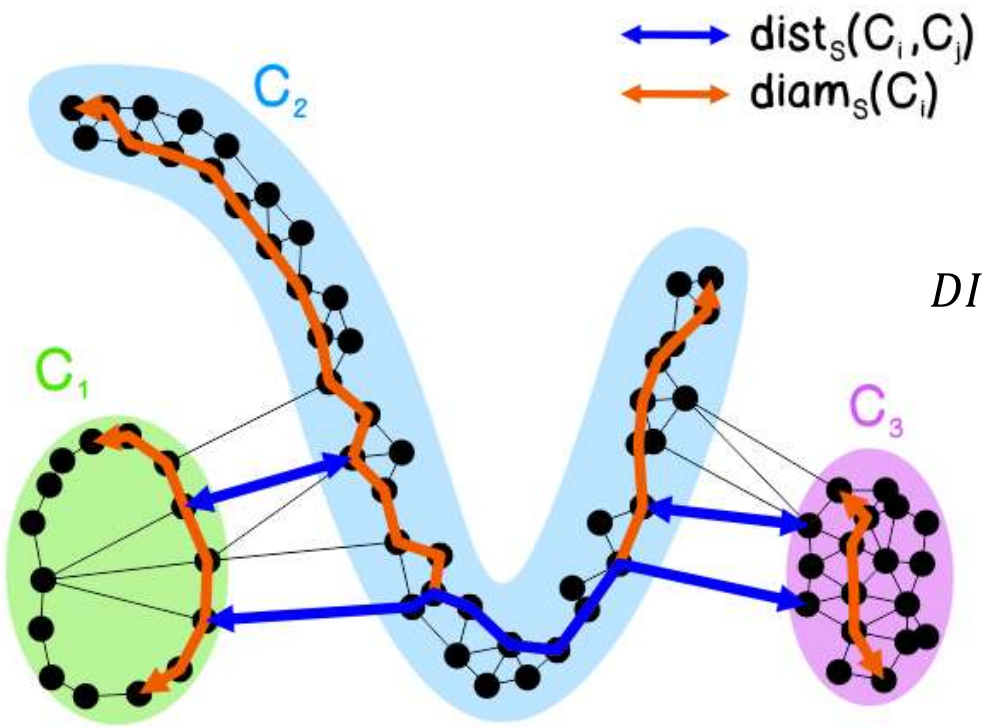
$avg(B) = (|B - x_4| + |B - x_5| + |B - x_6|)/3$

$avg(C) = (|C - x_7| + |C - x_8| + |C - x_9|)/3$

性能度量—内部指标

- 考虑聚类结果的簇划分$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

$$dist(C_i, C_j) = \min_{\boldsymbol{x}_i \in C_i, \boldsymbol{x}_j \in C_j} dist(\boldsymbol{x}_i, \boldsymbol{x}_j) \qquad \text{diam}(C) = \max_{1 \le i < j \le |C|} dist(\boldsymbol{x}_i, \boldsymbol{x}_j)$$



$$DI = \min_{1 \le i \le k} \left\{ \min_{j \ne i} \left(\frac{dist(C_i, C_j)}{\max_{1 \le l \le k} (\text{diam}(C_l))} \right) \right\}$$

DI越大越好

距离计算

- 距离度量的性质：

非负性：$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$

同一性：$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$当且仅当$\boldsymbol{x}_i = \boldsymbol{x}_j$

对称性：$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) = dist(\boldsymbol{x}_j, \boldsymbol{x}_i)$

传递性：$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq dist(\boldsymbol{x}_i, \boldsymbol{x}_k) + dist(\boldsymbol{x}_k, \boldsymbol{x}_j)$

- 常用距离：

闵可夫斯基距离（Minkowski distance）：

$$dist(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{u}^{n} |x_{iu} - x_{ju}|^p \right)^{1/p}$$

主要应用连续属性上

$p = 2$：欧式距离

$p = 1$：曼哈顿距离，也称街区距离

距离计算

- 处理离散属性
  - 如果属性取值可比较，比如定义域{少年、中年、老年}。那么可以用{1,2,3}数值来表示，而1与2比较接近，与3比较远，可以直接计算距离。
  - 如果属性取值不可比，比如定义域为{飞机、火车、轮船}。那么无法计算距离
- 采样VDM (Value Difference Metric) 来处理无序属性
  - 令$m_{u,a}$表示属性u上取值为a的样本数，$m_{u,a,i}$表示在第i个样本簇中在属性u上取值为a的样本数。则属性u在两个离散值a和b的VDM距离为

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

距离计算

- 处理混合属性

$$\mathrm{MinkovDM}_p = \left(\sum_{u=1}^{n_c} \left| x_{iu} - x_{ju} \right|^p + \sum_{u=n_c+1}^{n} VDM_p(x_{iu}, x_{ju}) \right)$$

- 加权距离（样本中不同属性的重要性不同时）：
  - 加权闵可夫斯基距离

$$dist_{wmk}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(w_1 \left| x_{i1} - y_{j1} \right|^p + \cdots + w_n \left| x_{in} - x_{jn} \right|^p \right)^{\frac{1}{p}}$$

$$\text{满足} w_i \geq 0, \sum_i^n w_i = 1$$

原型聚类

- 原型聚类

　　也称为“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。

- 介绍几种著名的原型聚类算法

　　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

算法流程（迭代优化）：

初始化每个簇的均值向量

repeat

1. 将每个样本分配给最近的簇；
2. 计算每个簇的均值向量

until 当前均值向量均未更新

原型聚类—K均值

输入： 样本集$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
聚类簇数k.

过程：

1: 从D中随机选择k个样本作为初始均值向量$\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k\}$

2: **repeat**

3: 令$C_i = \emptyset \ (1 \leq i \leq k)$

4: **for** $j = 1, \ldots, m$ **do**

5: 计算样本$\boldsymbol{x}_j$与各均值向量$\boldsymbol{\mu}_i \ (1 \leq i \leq k)$的距离：$d_{ji} = ||\boldsymbol{x}_j - \boldsymbol{\mu}_i||_2$;

6: 根据距离最近的均值向量确定$\boldsymbol{x}_j$的簇标记：$\lambda_j = \arg\min_{i \in \{1,2,\ldots,k\}} d_{ji}$;

7: 将样本$\boldsymbol{x}_j$划入相应的簇：$C_{\lambda_j} = C_{\lambda_j} \bigcup \{\boldsymbol{x}_j\}$;

8: **end for**

9: **for** $i = 1, \ldots, k$ **do**

10: 计算新均值向量：$\boldsymbol{\mu}'_i = \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{x}$;

11: **if** $\boldsymbol{\mu}'_i \neq \boldsymbol{\mu}_i$ **then**

12: 将当前均值向量$\boldsymbol{\mu}_i$更新为$\boldsymbol{\mu}'_i$

13: **else**

14: 保持当前均值向量不变

15: **end if**

16: **end for**

17: **until** 当前均值向量均未更新

18: **return** 簇划分结果

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

初始化每个簇的均值向量

将每个样本分配给最近的簇

计算每个簇的均值向量

原型聚类—K均值

- 给定数据集 $D = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$，K均值算法针对聚类所得簇划分 $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$最小化平方误差

$$E = \sum_{c=1}^{k} \sum_{\boldsymbol{x} \in C_c} \|\boldsymbol{x} - \boldsymbol{\mu}_c\|_2^2 \qquad \boldsymbol{\mu}_c = \frac{1}{|C_c|} \sum_{\boldsymbol{x} \in C_c} \boldsymbol{x}$$

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。

$$E(\boldsymbol{T}, \boldsymbol{\mu}) = \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \boldsymbol{t}_i^{\mathsf{T}} \boldsymbol{\mu} \right\|_2^2 = \|\boldsymbol{X} - \boldsymbol{T}\boldsymbol{\mu}\|_F^2 \qquad \boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^{\mathsf{T}} \\ \boldsymbol{t}_2^{\mathsf{T}} \\ \vdots \\ \boldsymbol{t}_m^{\mathsf{T}} \end{bmatrix}$$

$$\boldsymbol{t}_i^{\mathsf{T}} = [0, \ldots, \underset{c_i}{1}, \ldots, 0] \qquad \boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k] \in \mathbb{R}^{k \times d}$$

1 c_i k

原型聚类—K均值

- 优化问题：$\min_{\mathbf{T},\boldsymbol{\mu}} E(\boldsymbol{T},\boldsymbol{\mu}) = \|\boldsymbol{X} - \boldsymbol{T}\boldsymbol{\mu}\|_F^2$

算法流程（迭代优化）：

初始化每个簇的均值向量

repeat

1. 将每个样本分配给最近的簇；$\boldsymbol{T}^{(t)} \leftarrow \min_{\boldsymbol{T}} E\left(\boldsymbol{T}, \boldsymbol{\mu}^{(t-1)}\right)$

2. 计算每个簇的均值向量；$\boldsymbol{\mu}^{(t)} \leftarrow \min_{\boldsymbol{\mu}} E\left(\boldsymbol{T}^{(t)}, \boldsymbol{\mu}\right)$

until 当前均值向量均未更新

原型聚类—K均值

- 考虑$\boldsymbol{T}^{(t)} \leftarrow \min_{\boldsymbol{T}} E\left(\boldsymbol{T}, \boldsymbol{\mu}^{(t-1)}\right)$

- 由于样本间互不依赖，考虑求解第i个样本的$\boldsymbol{t}_i$

$$\min_{\boldsymbol{t}_i} \left\|\boldsymbol{x}_i - \boldsymbol{t}_i^{\top}\boldsymbol{\mu}\right\|_2^2 = \left\|\boldsymbol{x}_i - \sum_{c} t_{ic}\,\boldsymbol{\mu}_c\right\|_2^2 \Leftrightarrow \min_{\mathrm{c}} \|\boldsymbol{x}_i - \boldsymbol{\mu}_c\|$$

将样本i划分给距离最近的簇

t_{ic}在所有的c中只能有一个地方取值为1，其余均为0，即$\sum_c t_{ic} = 1$

原型聚类—K均值

- 进一步考虑$\boldsymbol{\mu}^{(t)} \leftarrow \min_{\boldsymbol{\mu}} E\left(\boldsymbol{T}^{(t)}, \boldsymbol{\mu}\right)$
- 求$E(\boldsymbol{T}, \boldsymbol{\mu})$关于$\boldsymbol{\mu}$的梯度，令其等于0，则

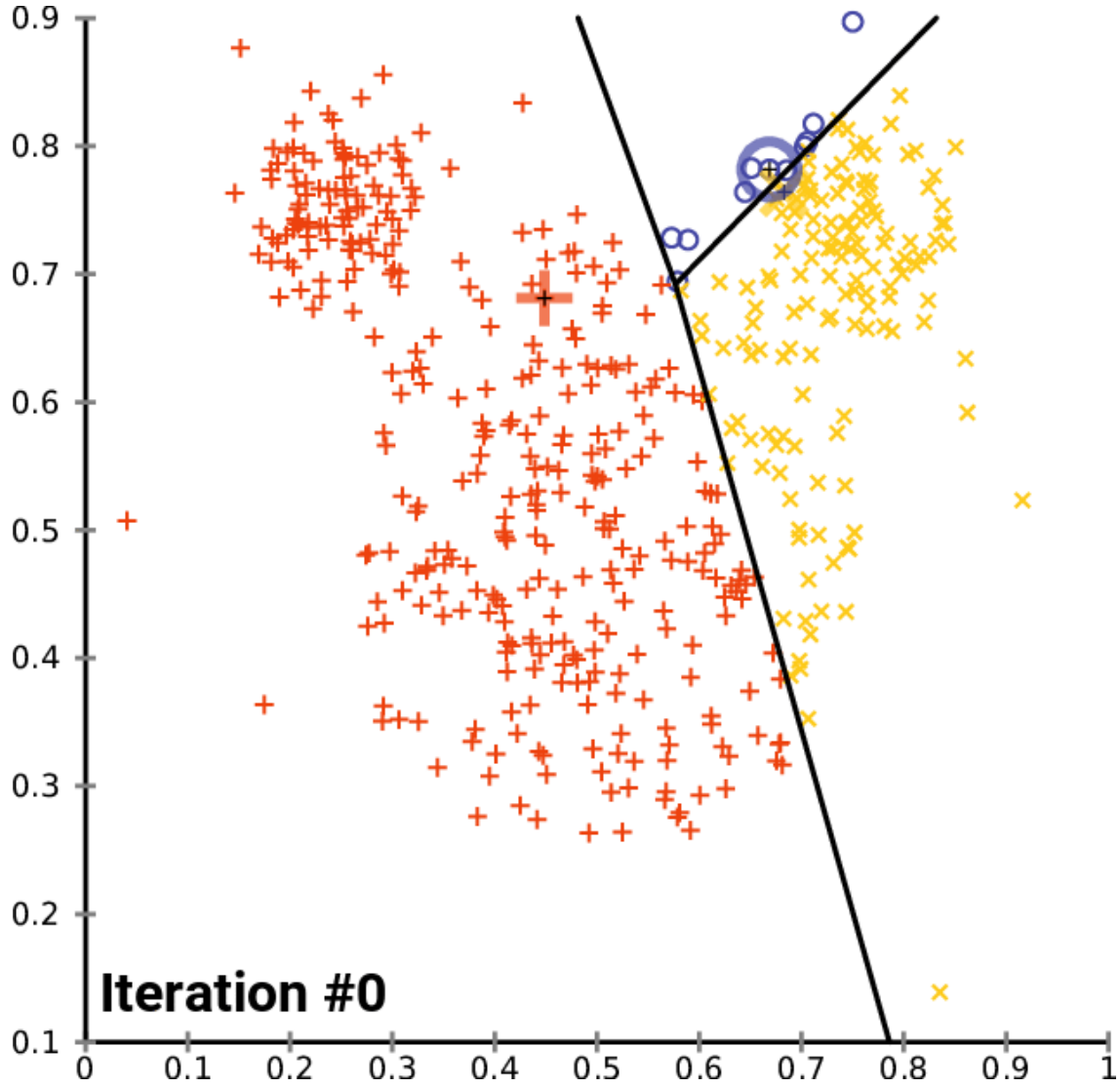
$$\boldsymbol{\mu} = (\boldsymbol{T}^{\top}\boldsymbol{T})^{-1}\boldsymbol{T}^{\top}\boldsymbol{X}$$

$$\boldsymbol{T}^{\top}\boldsymbol{T} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_m]\begin{bmatrix}\boldsymbol{t}_1^{\top}\\ \boldsymbol{t}_2^{\top}\\ \vdots\\ \boldsymbol{t}_m^{\top}\end{bmatrix} = \begin{bmatrix}\tilde{\boldsymbol{t}}_1^{\top}\\ \tilde{\boldsymbol{t}}_2^{\top}\\ \vdots\\ \tilde{\boldsymbol{t}}_k^{\top}\end{bmatrix}[\tilde{\boldsymbol{t}}_1, \tilde{\boldsymbol{t}}_2, \ldots, \tilde{\boldsymbol{t}}_k] = \begin{bmatrix}\tilde{\boldsymbol{t}}_1^{\top}\tilde{\boldsymbol{t}}_1 & \tilde{\boldsymbol{t}}_1^{\top}\tilde{\boldsymbol{t}}_2 & \cdots & \tilde{\boldsymbol{t}}_1^{\top}\tilde{\boldsymbol{t}}_k\\ \tilde{\boldsymbol{t}}_2^{\top}\tilde{\boldsymbol{t}}_1 & \tilde{\boldsymbol{t}}_2^{\top}\tilde{\boldsymbol{t}}_2 & \cdots & \tilde{\boldsymbol{t}}_2^{\top}\tilde{\boldsymbol{t}}_k\\ \vdots & \vdots & \ddots & \vdots\\ \tilde{\boldsymbol{t}}_k^{\top}\tilde{\boldsymbol{t}}_1 & \tilde{\boldsymbol{t}}_k^{\top}\tilde{\boldsymbol{t}}_2 & \cdots & \tilde{\boldsymbol{t}}_k^{\top}\tilde{\boldsymbol{t}}_k\end{bmatrix}$$

$$\boldsymbol{T}^{\top}\boldsymbol{X} = \begin{bmatrix}\tilde{\boldsymbol{t}}_1^{\top}\boldsymbol{X}\\ \tilde{\boldsymbol{t}}_2^{\top}\boldsymbol{X}\\ \vdots\\ \tilde{\boldsymbol{t}}_k^{\top}\boldsymbol{X}\end{bmatrix} = \begin{bmatrix}\sum_{\boldsymbol{x}\in C_1}\boldsymbol{x}\\ \sum_{\boldsymbol{x}\in C_2}\boldsymbol{x}\\ \vdots\\ \sum_{\boldsymbol{x}\in C_k}\boldsymbol{x}\end{bmatrix} = \begin{bmatrix}|c_1| & 0 & \cdots & 0\\ 0 & |c_2| & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & |c_k|\end{bmatrix}$$

$$\boldsymbol{\mu} = (\boldsymbol{T}^{\top}\boldsymbol{T})^{-1}\boldsymbol{T}^{\top}\boldsymbol{X} = \begin{bmatrix}\frac{1}{|c_1|}\sum_{\boldsymbol{x}\in C_1}\boldsymbol{x}\\ \frac{1}{|c_1|}\sum_{\boldsymbol{x}\in C_2}\boldsymbol{x}\\ \vdots\\ \frac{1}{|c_1|}\sum_{\boldsymbol{x}\in C_k}\boldsymbol{x}\end{bmatrix}$$

原型聚类—K均值



k均值算法实例

- 接下来以表9-1的西瓜数据集4.0为例，来演示k均值算法的学习过程。将编号为i的样本称为$\boldsymbol{x}_i$ 。

编号	密度	含糖率	编号	密度	含糖率	编号	密度	含糖率
1	0.697	0.460	11	0.245	0.057	21	0.748	0.232
2	0.774	0.376	12	0.343	0.099	22	0.714	0.346
3	0.634	0.264	13	0.639	0.161	23	0.483	0.312
4	0.608	0.318	14	0.657	0.198	24	0.478	0.437
5	0.556	0.215	15	0.360	0.370	25	0.525	0.369
6	0.403	0.237	16	0.593	0.042	26	0.751	0.489
7	0.481	0.149	17	0.719	0.103	27	0.532	0.472
8	0.437	0.211	18	0.359	0.188	28	0.473	0.376
9	0.666	0.091	19	0.339	0.241	29	0.725	0.445
10	0.243	0.267	20	0.282	0.257	30	0.446	0.459

k均值算法实例

假定聚类簇数$k=3$，算法开始时，随机选择3个样本x_6, x_{12}, x_{27}作为初始均值向量，即$\mu_1 = (0.403, 0.237), \mu_2 = (0.343, 0.099), \mu_3 = (0.533, 0.472)$

考察样本$\boldsymbol{x}_1 = (0.697, 0.460)$，它与当前均值向量 μ_1, μ_2, μ_3的距离分别为0.369, 0.506, 0.166，因此 $\boldsymbol{x}_1$将被划入簇C_3中。类似的，对数据集中的所有样本考察一遍后，可得当前簇划分为

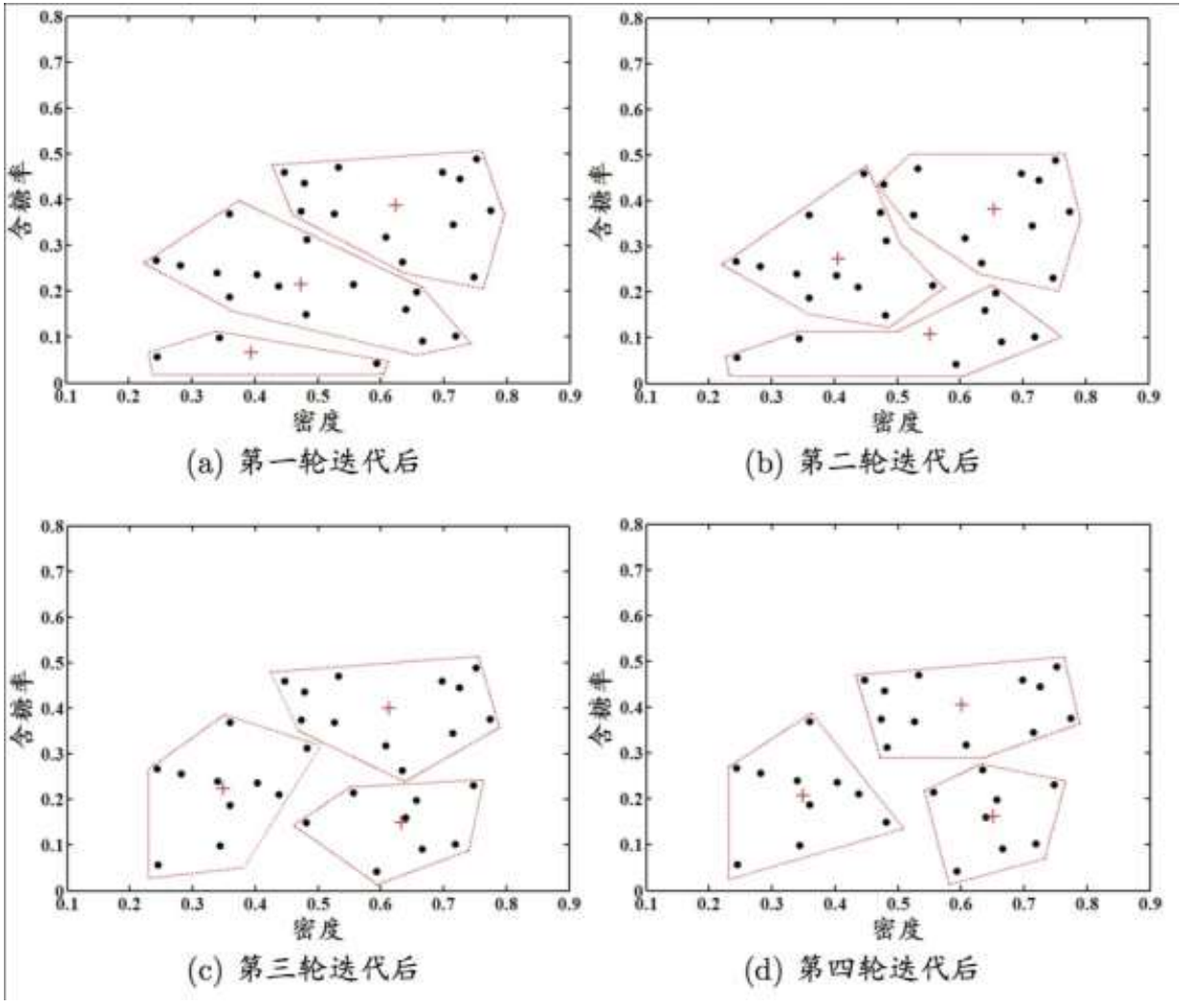
$$C_1 = \{x_5, x_6, x_7, x_8, x_9, x_{10}, x_{13}, x_{14}, x_{15}, x_{17}, x_{18}, x_{19}, x_{20}, x_{23}\}$$
$$C_2 = \{x_{11}, x_{12}, x_{16}\}$$
$$C_3 = \{x_1, x_2, x_3, x_4, x_{21}, x_{22}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{30}\}$$

于是，可以从分别求得新的均值向量

$\mu_1' = (0.473, 0.214), \mu_2' = (0.394, 0.066), \mu_3' = (0.623, 0.388)$

不断重复上述过程，如下图所示。

k均值算法实例



原型聚类—学习向量量化

- 学习向量量化（Learning Vector Quantization, LVQ）

- 与一般聚类算法不同的是，LVQ假设数据样本带有类别标记，学习过程中利用样本的这些监督信息来辅助聚类

- 给定样本集$D = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_m, y_m)\}$，$y_i \in \mathcal{Y}$，LVQ的目标是学得一组$d$维原型向量$\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_q$，每个原型向量代表一个聚类簇，簇标记$t_i \in \mathcal{Y}$

原型聚类—学习向量量化

输入: 样本集$D = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$;
原型向量个数q, 各原型向量预设的类别标记$\{t_1, t_2, \ldots, t_q\}$;
学习率$\eta \in (0, 1)$.

过程:

1: 初始化一组原型向量$\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_q\}$
2: **repeat**
3: 从样本集D随机选取样本$(\boldsymbol{x}_j, y_j)$;
4: 计算样本$\boldsymbol{x}_j$与$\boldsymbol{p}_i$ $(1 \leq i \leq q)$的距离: $d_{ji} = ||\boldsymbol{x}_j - \boldsymbol{p}_i||_2$;
5: 找出与$\boldsymbol{x}_j$距离最近的原型向量; $i^* = \arg\min_{i \in \{1,2,\ldots,q\}} d_{ji}$;
6: **if** $y_j = t_{i^*}$ **then**
7: $\boldsymbol{p}' = \boldsymbol{p}_{i^*} + \eta \cdot (\boldsymbol{x}_j - \boldsymbol{p}_{i^*})$
8: **else**
9: $\boldsymbol{p}' = \boldsymbol{p}_{i^*} - \eta \cdot (\boldsymbol{x}_j - \boldsymbol{p}_{i^*})$
10: **end if**
11: 将原型向量$\boldsymbol{p}_{i^*}$更新为$\boldsymbol{p}'$
12: **until** 满足停止条件
13: **return** 当前原型向量

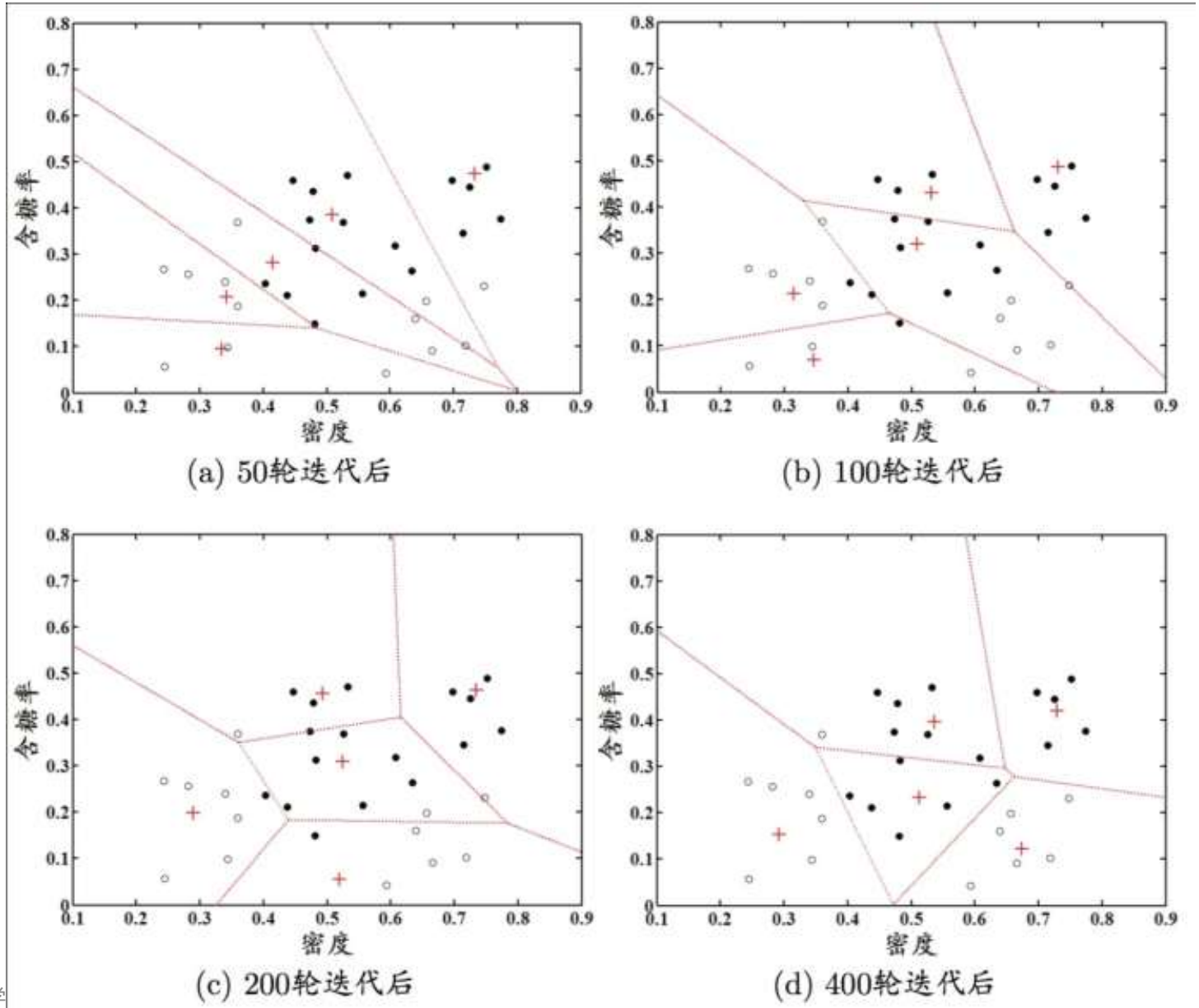
输出: 原型向量$\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_q\}$

根据两者的类别标记是否一致来对原型向量进行更新

对样本$\boldsymbol{x}_j$，若最近的原型向量$\boldsymbol{p}_{i^*}$和$\boldsymbol{x}_j$的类别标记相同，则令$\boldsymbol{p}_{i^*}$向$\boldsymbol{x}_j$的方向靠拢

$$\begin{aligned}\|\boldsymbol{p}' - \boldsymbol{x}_j\| &= \|\boldsymbol{p}_{i^*} + \eta(\boldsymbol{x}_j - \boldsymbol{p}_{i^*}) - \boldsymbol{x}_j\| \\ &= (1-\eta)\|\boldsymbol{p}_{i^*} - \boldsymbol{x}_j\| \\ &< \|\boldsymbol{p}_{i^*} - \boldsymbol{x}_j\|\end{aligned}$$

原型聚类—学习向量量化


含糖率
密度
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
(a) 50轮迭代后
(b) 100轮迭代后
(c) 200轮迭代后
(d) 400轮迭代后

原型聚类 – 高斯混合聚类

- 与k均值、LVQ用原型向量来刻画聚类结构不同，高斯混合聚类（Mixture-of-Gaussian）采用概率模型来表达聚类原型：

- 对d维样本空间中的随机向量$\boldsymbol{x}$，若$\boldsymbol{x}$服从多元高斯分布，其概率密度函数为

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 其中$\boldsymbol{\mu}$是d维均值向量，$\boldsymbol{\Sigma}$是$d \times d$的协方差矩阵。也可将概率密度函数记作$p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$。

原型聚类 – 高斯混合聚类

- 高斯混合分布的定义

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

该分布由k个混合分布组成，每个分布对应一个高斯分布。其中，$\boldsymbol{\mu}_c$与$\boldsymbol{\Sigma}_c$是第c个高斯混合成分的参数。而α_c为相应的“混合系数”，$\sum_c \alpha_c = 1$。

假设样本的生成过程由高斯混合分布给出：

(1) 根据$\alpha_1, \cdots, \alpha_k$定义的先验分布选择高斯混合成分，$\alpha_i$为选择第$i$个成分的概率；

(2) 根据被选择的混合成分的概率密度函数进行采样，从而生成相应的样本

原型聚类 – 高斯混合聚类

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \qquad p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 模型求解：最大化（对数）似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i}^{m} \ln p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{m} \ln \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

- 求$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$关于$\boldsymbol{\mu_c}$的偏导数

$$\frac{\partial LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu_c}} = \sum_{i=1}^{m} \frac{\alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu_c}) = 0$$

$$\Rightarrow \sum_{i=1}^{m} \gamma_{ic} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu_c}) = 0$$

$$\Rightarrow \boldsymbol{\mu_c} = \frac{\sum_{i=1}^{m} \gamma_{ic} \boldsymbol{x}_i}{\sum_{i=1}^{m} \gamma_{ic}}$$

各混合成分的均值可通过样本加权平均来估计

原型聚类 – 高斯混合聚类

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \qquad p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 模型求解：最大化（对数）似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i}^{m} \ln p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{m} \ln \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

- 令$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$关于$\boldsymbol{\Sigma}_c$的偏导数等于0，可得

$$\boldsymbol{\Sigma}_c = \frac{\sum_i^m \gamma_{ic} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^{\mathsf{T}}}{\sum_i^m \gamma_{ic}}$$

原型聚类 – 高斯混合聚类

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \qquad p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 模型求解：最大化（对数）似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i}^{m} \ln p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{m} \ln \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

- 考虑$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$的拉格朗日形式

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda\left(\sum_{c} \alpha_c - 1\right)$$

- 求其关于α_c的导数，令其等于0

$$\frac{\partial LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \alpha_c} = \sum_{i=1}^{m} \frac{p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} + \lambda = 0 \Rightarrow \sum_{i=1}^{m} \frac{\alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} = -\lambda\alpha_c$$

$$\Rightarrow \sum_{c=1}^{k}\sum_{i=1}^{m} \frac{\alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} = \sum_{c=1}^{k} -\lambda\alpha_c \Rightarrow \lambda = -m$$

原型聚类 – 高斯混合聚类

$$p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \qquad p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

- 模型求解：最大化（对数）似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i}^{m} \ln p_{\mathcal{M}}(\boldsymbol{x}) = \sum_{i}^{m} \ln \sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

- 考虑$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$的拉格朗日形式

$$LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda\left(\sum_{c} \alpha_c - 1\right)$$

- 求其关于α_c的导数，令其等于0

$$\frac{\partial LL(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \alpha_c} = \sum_{i=1}^{m} \frac{p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} + \lambda = 0 \Rightarrow \sum_{i=1}^{m} \frac{1}{\alpha_c} \frac{\alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_{c=1}^{k} \alpha_c p(\boldsymbol{x}_i|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)} = m$$

$$\Rightarrow \sum_{i=1}^{m} \frac{1}{\alpha_c} \gamma_{ic} = m \Rightarrow \alpha_c = \frac{1}{m} \sum_{i=1}^{m} \gamma_{ic}$$

原型聚类 – 高斯混合聚类

输入: 样本集$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
高斯混合成分个数k.

过程:

初始化高斯混合分布的模型参数$\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leq i \leq k\}$
repeat
  for $j = 1, \ldots, m$ **do**
    根据(9.30)计算$\boldsymbol{x}_j$由各混合成分生成的后验概率, 即
    $\gamma_{ji} = p_{\mathcal{M}}(z_j = i \mid \boldsymbol{x}_j)\ (1 \leq i \leq k)$
  end for
  for $i = 1, \ldots, k$ **do**
    计算新均值向量: $\boldsymbol{\mu}_i' = \frac{\sum_{j=1}^{m} \gamma_{ji} \boldsymbol{x}_j}{\sum_{j=1}^{m} \gamma_{ji}}$;
    计算新协方差矩阵: $\boldsymbol{\Sigma}_i' = \frac{\sum_{j=1}^{m} \gamma_{ji} (\boldsymbol{x}_j - \boldsymbol{\mu}_i')(\boldsymbol{x}_j - \boldsymbol{\mu}_i')^{\top}}{\sum_{j=1}^{m} \gamma_{ji}}$;
    计算新混合系数: $\alpha_i' = \frac{\sum_{j=1}^{m} \gamma_{ji}}{m}$;
  end for
  将模型参数$\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leq i \leq k\}$更新为$\{(\alpha_i', \boldsymbol{\mu}_i', \boldsymbol{\Sigma}_i') \mid 1 \leq i \leq k\}$
until 满足停止条件
$C_i = \emptyset\ (1 \leq i \leq k)$
for $j = 1, \ldots, m$ **do**
  根据(9.31)确定$\boldsymbol{x}_j$的簇标记λ_j;
  将$\boldsymbol{x}_j$划入相应的簇: $C_{\lambda_j} = C_{\lambda_j} \bigcup \{\boldsymbol{x}_j\}$
end for
return 簇划分结果

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

密度聚类

- 密度聚类也称为“基于密度的聚类” (density-based clustering)。
- 此类算法假设聚类结构能通过样本分布的紧密程度来确定。
- 通常情况下，密度聚类算法从样本密度的角度来考察样本之间的可连接性，并基于可连接样本不断扩展聚类簇来获得最终的聚类结果

接下来介绍DBSCAN这一密度聚类算法。

密度聚类

- DBSCAN算法：基于一组“邻域”参数$(\epsilon, \text{MinPts})$来刻画样本分布的紧密程度。
- 基本概念：

- ϵ邻域：对样本$\boldsymbol{x}_i \in D$，其ϵ邻域包含样本集D中与$\boldsymbol{x}_i$的距离不大于ϵ的样本；
- 核心对象：若样本$\boldsymbol{x}_i$的ϵ邻域至少包含MinPts个样本，则该样本点为核心对象；
- 密度直达：若样本$\boldsymbol{x}_j$位于样本$\boldsymbol{x}_i$的ϵ邻域中，且$\boldsymbol{x}_i$是一个核心对象，则称样本$\boldsymbol{x}_j$由$\boldsymbol{x}_i$密度直达；
- 密度可达：对样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$，若存在样本序列$\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n$，其中$\boldsymbol{p}_1 = \boldsymbol{x}_i, \boldsymbol{p}_n = \boldsymbol{x}_j$且$\boldsymbol{p}_{i+1}$由$\boldsymbol{p}_i$密度直达，则该两样本密度可达；
- 密度相连：对样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$，若存在样本$\boldsymbol{x}_k$使得两样本均由$\boldsymbol{x}_k$密度可达，则称该两样本密度相连。

密度聚类

- 一个例子

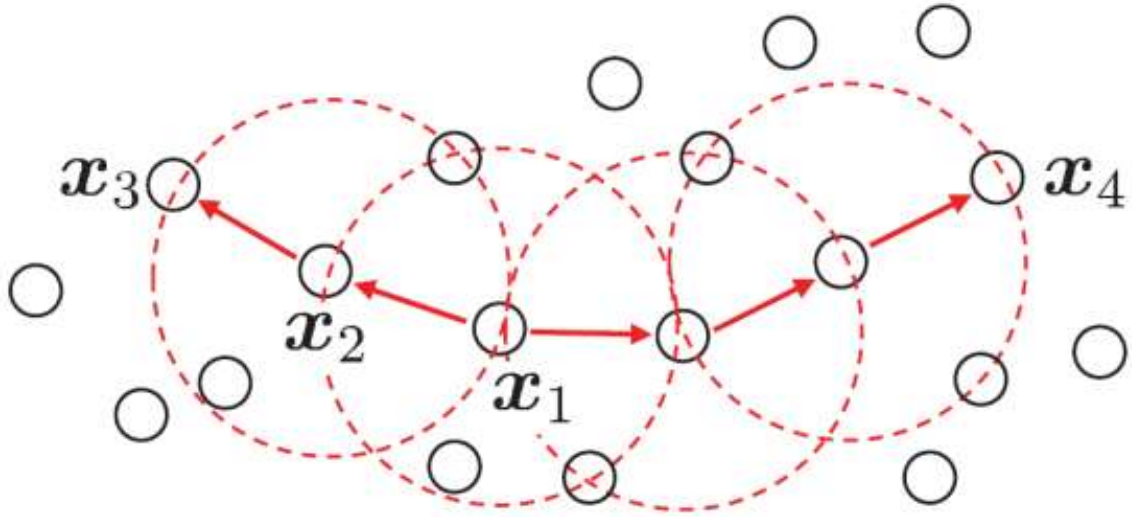
  令$MinPts = 3$，则虚线显示出ϵ领域。

  $\boldsymbol{x}_1$是核心对象。

  $\boldsymbol{x}_2$由$\boldsymbol{x}_1$密度直达。

  $\boldsymbol{x}_3$由$\boldsymbol{x}_1$密度可达。

  $\boldsymbol{x}_3$与$\boldsymbol{x}_4$密度相连。



密度聚类

- 对“簇”的定义

  由密度可达关系导出的最大密度相连样本集合。

- 对“簇”的形式化描述

  给定领域参数，簇是满足以下性质的非空样本子集：

  连接性：$\boldsymbol{x}_i \in C, \boldsymbol{x}_j \in C \Rightarrow \boldsymbol{x}_i$与$\boldsymbol{x}_j$密度相连

  最大性：$\boldsymbol{x}_i \in C$，$\boldsymbol{x}_i$与$\boldsymbol{x}_j$密度可达$\Rightarrow \boldsymbol{x}_j \in C$

实际上，若$\boldsymbol{x}$为核心对象，由$\boldsymbol{x}$密度可达的所有样本组成的集合记为$X = \{\boldsymbol{x}' \in D \mid \boldsymbol{x}'$由$\boldsymbol{x}$密度可达$\}$，则$X$为满足连接性与最大性的簇。

密度聚类

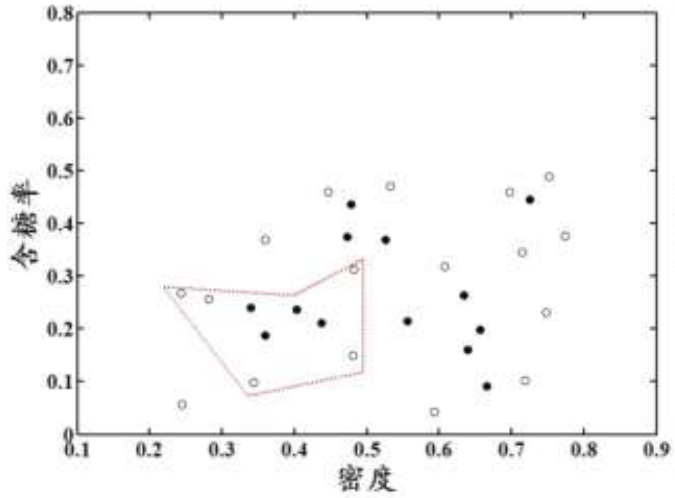
输入: 样本集$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
邻域参数$(\epsilon, MinPts)$.

过程:

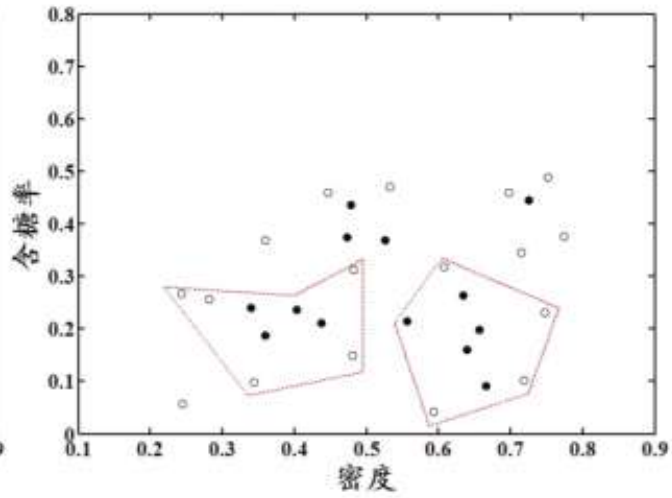
初始化核心对象集合：$\Omega = \emptyset$
for $j = 1, \ldots, m$ **do**
  确定样本$\boldsymbol{x}_j$的ϵ-邻域$N_\epsilon(\boldsymbol{x}_j)$;
  if $|N_\epsilon(\boldsymbol{x}_j)| \geq MinPts$ **then**
    将样本$\boldsymbol{x}_j$加入核心对象集合：$\Omega = \Omega \bigcup \{\boldsymbol{x}_j\}$
  end if
end for
初始化聚类簇数：$k = 0$
初始化未访问样本集合：$\Gamma = D$
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$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

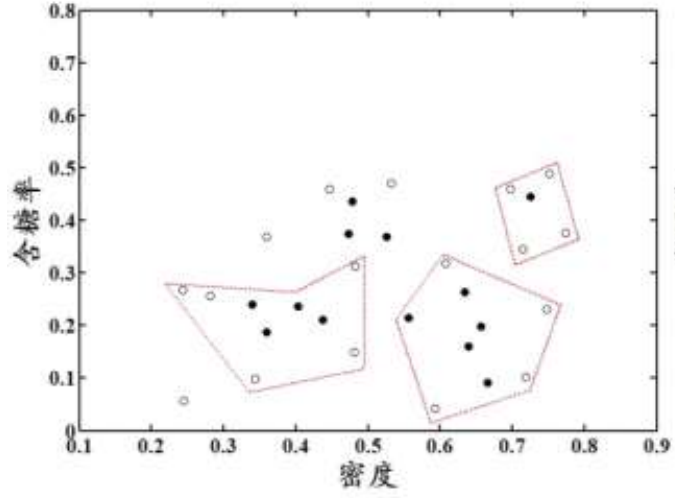
密度聚类



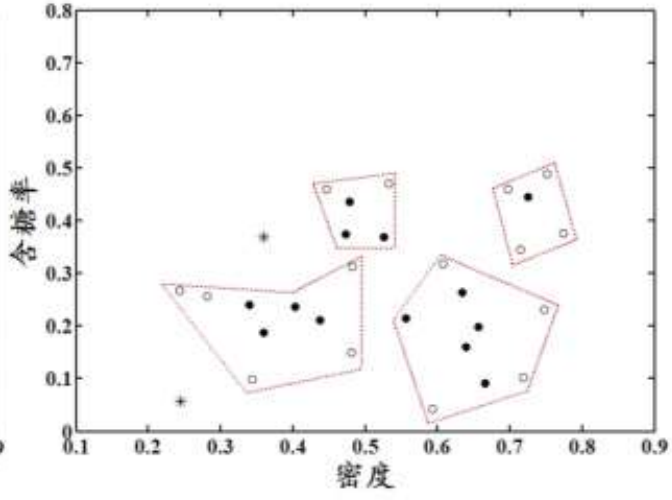
(a) 生成聚类簇C_1



(b) 生成聚类簇C_2



(c) 生成聚类簇C_3



(d) 生成聚类簇C_4

层次聚类

- 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。

层次聚类

- AGNES算法（自底向上的层次聚类算法）

1 将样本中的每一个样本看做一个初始聚类簇

未到预设聚类簇数

2 在算法运行的每一步中找出距离最近的两个聚类簇进行合并

层次聚类

- 这里两个聚类簇C_i和C_j的距离，可以有3种度量方式。

最小距离：$d_{\min}(C_i, C_j) = \min_{\boldsymbol{x}\in C_i, \boldsymbol{y}\in C_j} dist(\boldsymbol{x}, \boldsymbol{y})$

最大距离：$d_{\max}(C_i, C_j) = \max_{\boldsymbol{x}\in C_i, \boldsymbol{y}\in C_j} dist(\boldsymbol{x}, \boldsymbol{y})$

平均距离：$d_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|}\sum_{\boldsymbol{x}\in C_i}\sum_{\boldsymbol{y}\in C_j} dist(\boldsymbol{x}, \boldsymbol{y})$

层次聚类

输入： 样本集$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
聚类簇距离度量函数$d \in \{d_{\min}, d_{\max}, d_{\text{avg}}\}$;
聚类簇数k.

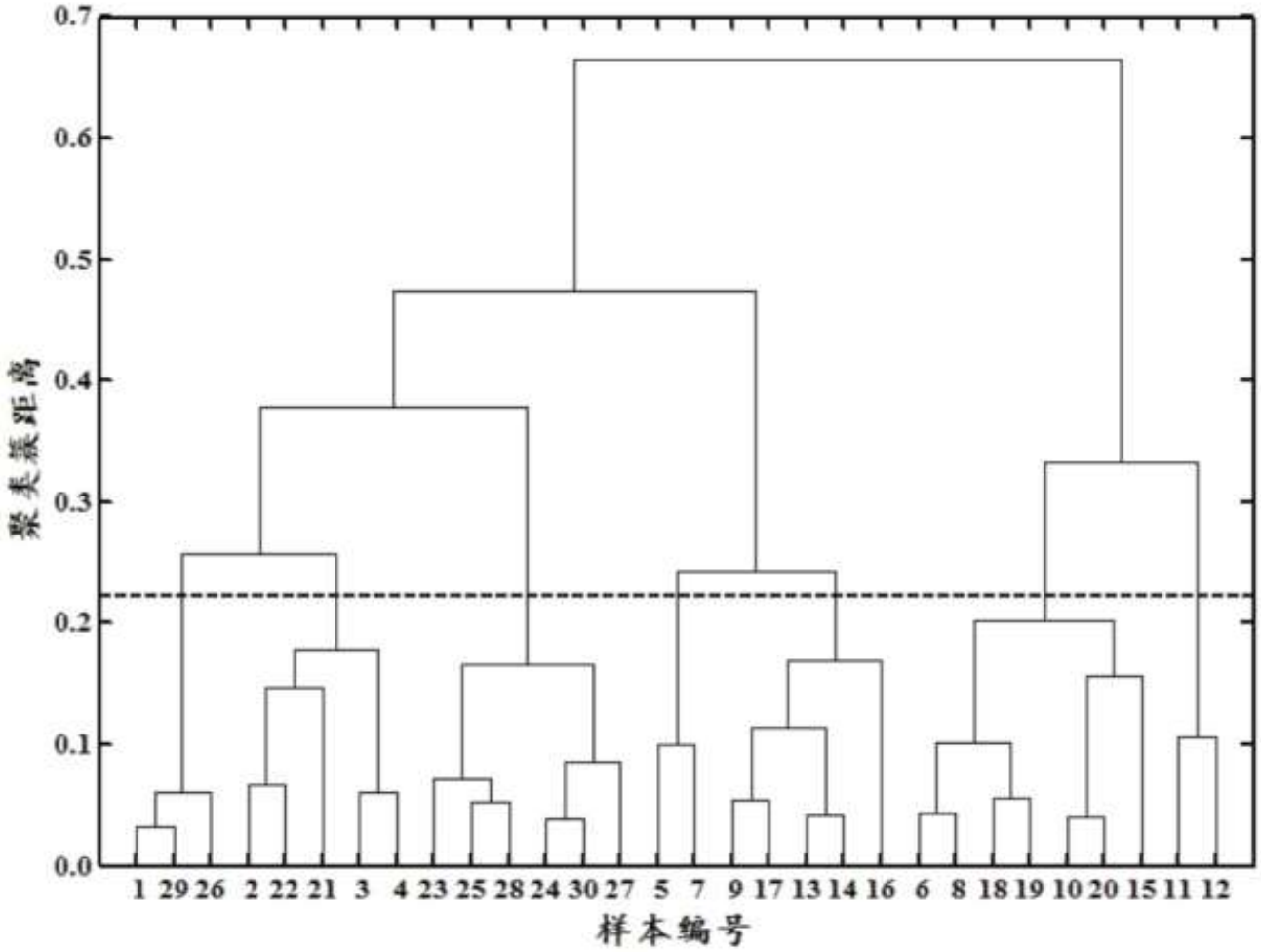
过程：

1: **for** $j = 1, \ldots, m$ **do**
2: $C_j = \{\boldsymbol{x}_j\}$
3: **end for**
4: **for** $i = 1, \ldots, m$ **do**
5: **for** $j = i, \ldots, m$ **do**
6: $M(i,j) = d(C_i, C_j)$;
7: $M(j,i) = M(i,j)$
8: **end for**
9: **end for**
10: 设置当前聚类簇个数：$q = m$
11: **while** $q > k$ **do**
12: 找出距离最近的两个聚类簇(C_{i^*}, C_{j^*});
13: 合并(C_{i^*}, C_{j^*})：$C_{i^*} = C_{i^*} \bigcup C_{j^*}$;
14: **for** $j = j^* + 1, \ldots, q$ **do**
15: 将聚类簇C_j重编号为C_{j-1}
16: **end for**
17: 删除距离矩阵M的第j^*行与第j^*列;
18: **for** $j = 1, \ldots, q-1$ **do**
19: $M(i^*, j) = d(C_{i^*}, C_j)$;
20: $M(j, i^*) = M(i^*, j)$
21: **end for**
22: $q = q - 1$
23: **end while**
24: **return** 簇划分结果

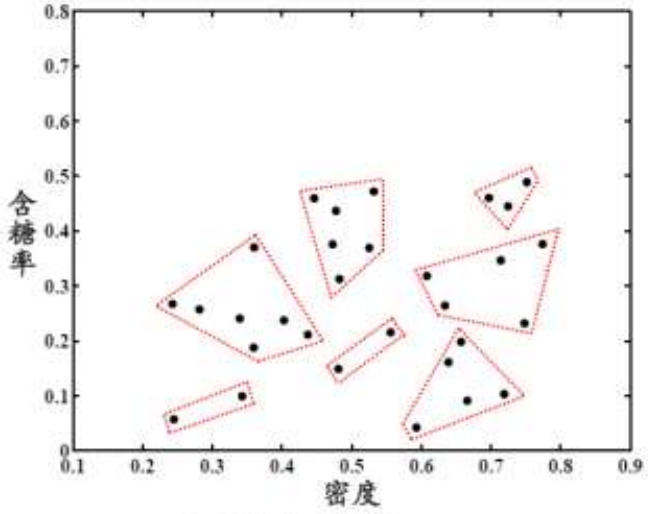
输出： 簇划分$\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$

层次聚类

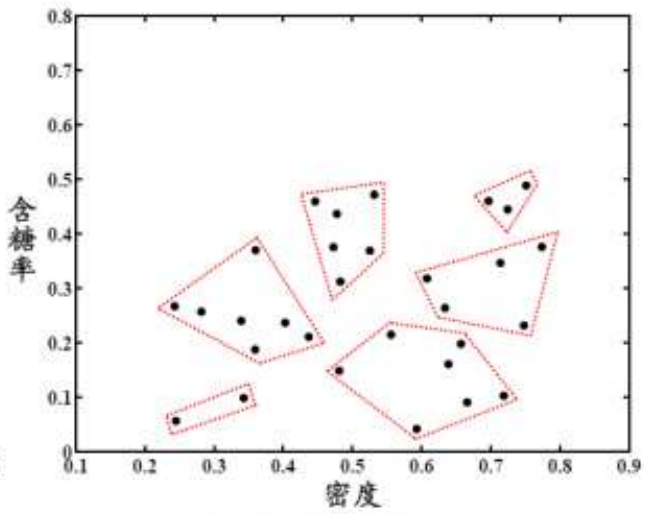
- AGNES算法树状图:



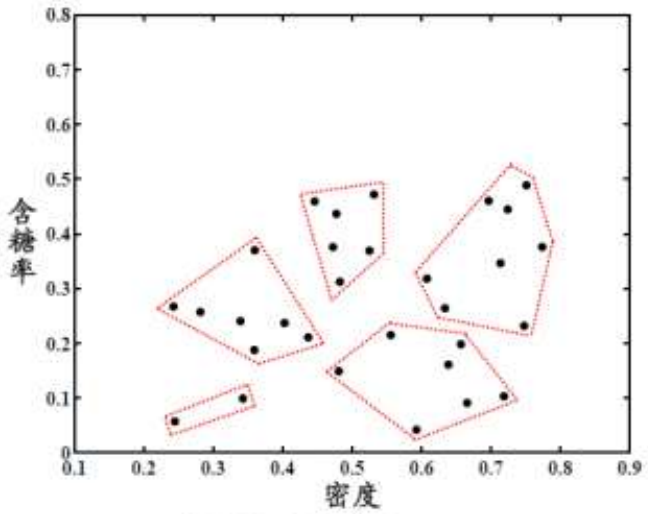
层次聚类



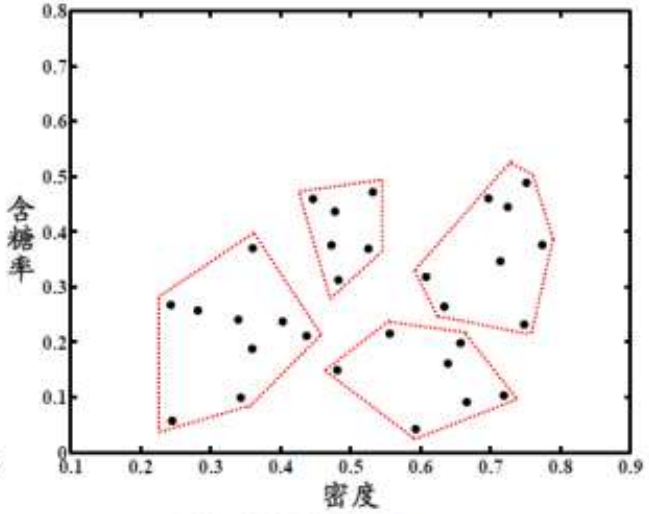
(a) 聚类簇数 $k = 7$



(b) 聚类簇数 $k = 6$



(c) 聚类簇数 $k = 5$



(d) 聚类簇数 $k = 4$

作业

- 给定任意的两个相同长度向量$\boldsymbol{x}, \boldsymbol{y}$，其余弦距离为$1 - \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}}{|\boldsymbol{x}||\boldsymbol{y}|}$，证明余弦距离不满足传递性，而余弦夹角$\arccos\left(\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}}{|\boldsymbol{x}||\boldsymbol{y}|}\right)$满足

- 证明k-means算法的收敛性

- 在k-means算法中替换欧式距离为其他任意的度量，请问“聚类簇”中心如何计算？

第十章：降维与度量学习

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

k近邻学习

- k近邻(k-Nearest Neighbor, kNN)学习是一种常用的监督学习方法:
  - 确定训练样本，以及某种距离度量。
  - 对于某个给定的测试样本，找到训练集中距离最近的k个样本
  - 对于分类问题使用“投票法”获得预测结果

    投票法：选择这k个样本中出现最多的类别标记作为预测结果。
  - 对于回归问题使用“平均法”获得预测结果。

    平均法：将这k个样本的实值输出标记的平均值作为预测结果。
  - 还可基于距离远近进行加权平均或加权投票，距离越近的样本权重越大。

“懒惰学习”与“急切学习”

- K近邻学习没有显式的训练过程，属于“懒惰学习”

“懒惰学习”(lazy learning):

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。

“急切学习”(eager learning):

在训练阶段就对样本进行学习处理的方法。

k近邻分类示意图

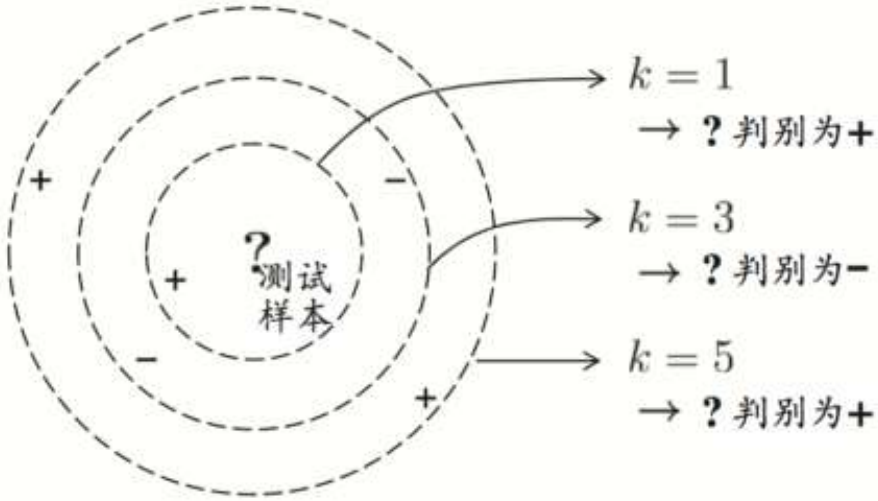


图 10.1 k 近邻分类器示意图. 虚线显示出等距线; 测试样本在 $k = 1$ 或 $k = 5$ 时被判别为正例, $k = 3$ 时被判别为反例.

- k近邻分类器中的k是一个重要参数，当k取不同值时，分类结果会有显著不同。
- 另一方面，若采用不同的距离计算方式，则找出的“近邻”可能有显著差别，从而也会导致分类结果有显著不同。

k近邻学习—1NN二分类错误率$P(err)$

- 暂且假设距离计算是“恰当”的，即能够恰当地找出k个近邻
- 我们来对“最近邻分类器”（1NN，即k=1）在二分类问题上的性能做一个简单的讨论。
- 给定测试样本$\boldsymbol{x}$，若其最近邻样本为$\boldsymbol{z}$，则最近邻出错的概率就是$\boldsymbol{x}$与$\boldsymbol{z}$类别标记不同的概率，即

$$P(err) = 1 - \sum_{c \in \mathcal{Y}} P(c|\boldsymbol{x}) P(c|\boldsymbol{z})$$

k近邻学习—1NN二分类错误率$P(err)$

- 假设样本独立同分布，且对任意$\boldsymbol{x}$和任意小正整数δ，在$\boldsymbol{x}$附近δ距离范围内总能找到一个训练样本

- 换言之，对任意测试样本，总能在任意近的范围内找到$P(err) = 1 - \sum_{c \in \mathcal{Y}} P(c|\boldsymbol{x})P(c|\boldsymbol{z})$中的训练样本$\boldsymbol{z}$

- 令 $c^\star = \operatorname{argmax}_{c \in \mathcal{Y}} P(c|\boldsymbol{x})$表示贝叶斯最优分类器的结果，有

$$
\begin{aligned}
P(err) &= 1 - \sum_{c \in \mathcal{Y}} P(c|\boldsymbol{x})P(c|\boldsymbol{z}) \\
&\simeq 1 - \sum_{c \in \mathcal{Y}} P^2(c|\boldsymbol{x}) \\
&\leq 1 - P^2(c^\star|\boldsymbol{x}) \\
&= \big(1 + P(c^\star|\boldsymbol{x})\big)\big(1 - P(c^\star|\boldsymbol{x})\big) \\
&\leq 2\big(1 - P(c^\star|\boldsymbol{x})\big)
\end{aligned}
$$

最近邻分类虽简单，但它的泛化错误率不超过贝叶斯最优分类器错误率的两倍！

维数灾难 (curse of dimensionality)

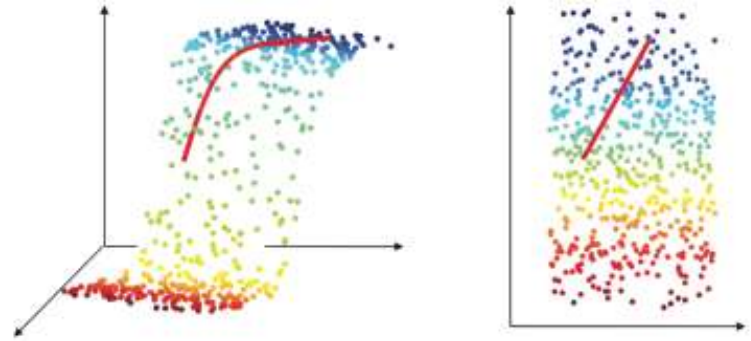
- 上述讨论基于一个重要的假设：任意测试样本 附近的任意小的距离范围内总能找到一个训练样本，即训练样本的采样密度足够大，或称为“密采样”。
- 然而，这个假设在现实任务中通常很难满足
  - 若属性维数为1，当$\delta = 0.001$，仅考虑单个属性，则仅需1000个样本点平均分布在归一化后的属性取值范围内，即可使得任意测试样本在其附近0.001距离范围内总能找到一个训练样本，此时最近邻分类器的错误率不超过贝叶斯最优分类器的错误率的两倍。若属性维数为20，若样本满足密采样条件，则至少需要1000^{20}个样本
  - 现实应用中属性维数经常成千上万，要满足密采样条件所需的样本数目是无法达到的天文数字
  - 许多学习方法都涉及距离计算，而高维空间会给距离计算带来很大的麻烦，例如当维数很高时甚至连计算内积都不再容易

在高维情形下出现的数据样本稀疏、距离计算困难等问题，是所有机器学习方法共同面临的严重障碍，被称为“维数灾难”

低维嵌入

- 缓解维数灾难的一个重要途径是降维(dimension reduction)
  - 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。

- 为什么能进行降维?
  - 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)，因而可以对数据进行有效的降维。



(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

多维缩放（MDS）

- 若要求原始空间中样本之间的距离在低维空间中得以保持，即得到“多维缩放”(Multiple Dimensional Scaling, MDS)：

- 假定有m个样本，在原始空间中的距离矩阵为$\boldsymbol{D} \in \mathbb{R}^{m \times m}$，其第$i$行$j$列的元素$dist_{ij}$为样本$i$到$j$的距离

- 目标是获得样本在$d'(\ll d)$维空间中的欧氏距离等于原始空间中的距离，即$\|\boldsymbol{z}_i - \boldsymbol{z}_j\| = \text{dist}_{ij}$

- 令$\boldsymbol{B} = \boldsymbol{Z}^{\top}\boldsymbol{Z}$，其中$\boldsymbol{B}$为降维后的内积矩阵，$b_{ij} = \boldsymbol{z}_i^{\top}\boldsymbol{z}_j$，有

$$\text{dist}_{ij}^2 = \|\boldsymbol{z}_i\|^2 + \|\boldsymbol{z}_j\|^2 - 2\boldsymbol{z}_i^{\top}\boldsymbol{z}_j$$

$$= b_{ii} + b_{jj} - 2b_{ij}$$

多维缩放（MDS）

- 为便于讨论，令降维后的样本$\boldsymbol{Z}$被中心化，即$\sum_i \boldsymbol{z}_i = 0$。显然，矩阵$\boldsymbol{B}$的行与列之和均为零，即

$$\sum_i b_{ij} = \sum_i \boldsymbol{z}_i^{\mathrm{T}} \boldsymbol{z}_j = \boldsymbol{z}_i^{\mathrm{T}} \sum_i \boldsymbol{z}_j = \boldsymbol{z}_i^{\mathrm{T}} 0 = 0 \qquad \sum_j b_{ij} = 0$$

$$\sum_{j=1}^{m} \mathrm{dist}_{ij}^2 = \sum_{j=1}^{m} \left(b_{ii} + b_{jj} - 2b_{ij}\right) = m b_{ii} + \mathrm{tr}(\boldsymbol{B})$$

$$\sum_{i=1}^{m} \mathrm{dist}_{ij}^2 = \sum_{i=1}^{m} \left(b_{ii} + b_{jj} - 2b_{ij}\right) = m b_{jj} + \mathrm{tr}(\boldsymbol{B})$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} dist_{ij}^2 = \sum_{i} \left(m b_{ii} + \mathrm{trace}(\boldsymbol{B})\right) = 2\, m\, \mathrm{tr}(\boldsymbol{B})$$

多维缩放（MDS）

$$\begin{aligned}\mathrm{dist}_{i\cdot}^2 &= \frac{1}{m}\sum_{j=1}^{m}\mathrm{dist}_{ij}^2 \\ &= \frac{1}{m}\left(mb_{ii}+\mathrm{tr}(\boldsymbol{B})\right) \\ &= b_{ii}+\frac{1}{m}\mathrm{tr}(\boldsymbol{B})\end{aligned}$$

$$\begin{aligned}\mathrm{dist}_{\cdot j}^2 &= \frac{1}{m}\sum_{i=1}^{m}\mathrm{dist}_{ij}^2 \\ &= \frac{1}{m}\left(mb_{jj}+\mathrm{tr}(\boldsymbol{B})\right) \\ &= b_{jj}+\frac{1}{m}\mathrm{tr}(\boldsymbol{B})\end{aligned}$$

$$\mathrm{dist}_{\cdot\cdot}^2 = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}dist_{ij}^2 \quad = \frac{1}{m^2}2\,m\,\mathrm{tr}(\boldsymbol{B}) \qquad = 2\frac{1}{m}\,\mathrm{tr}(\boldsymbol{B})$$

$$\mathrm{dist}_{ij}^2 = b_{ii}+b_{jj}-2b_{ij}$$

$$\mathrm{dist}_{i\cdot}^2+\mathrm{dist}_{\cdot j}^2-\mathrm{dist}_{\cdot\cdot}^2 = b_{ii}+b_{jj} \quad = \mathrm{dist}_{ij}^2+2b_{ij}$$

$$b_{ij} = \frac{1}{2}\left(\mathrm{dist}_{i\cdot}^2+\mathrm{dist}_{\cdot j}^2-\mathrm{dist}_{\cdot\cdot}^2-\mathrm{dist}_{ij}^2\right)$$

多维缩放（MDS）

$$b_{ij} = \frac{1}{2}\left(\mathrm{dist}_{i\cdot}^2 + \mathrm{dist}_{\cdot j}^2 - \mathrm{dist}_{\cdot\cdot}^2 - \mathrm{dist}_{ij}^2\right)$$

原空间中距离矩阵

$\boldsymbol{D} \in \mathbb{R}^{m\times m}$ ⇒ $\boldsymbol{B} = \boldsymbol{Z}^{\top}\boldsymbol{Z}$

降维后的内积矩阵

- 对矩阵$\boldsymbol{B}$做特征值分解(eigenvalue decomposition) $\boldsymbol{B} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\top}$，其中$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_d)$为特征值构成的对角矩阵，$\boldsymbol{V}$为特征向量矩阵
- $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$为特征向量矩阵，假定其中有$d^{\star}$个非零特征值， 它们构成对角矩阵$\boldsymbol{\Lambda}_{\star} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{d^{\star}})$。令$\boldsymbol{V}_{\star}$表示相应的特征矩阵，

$$\boldsymbol{Z} = \boldsymbol{\Lambda}_{\star}^{1/2}\boldsymbol{V}_{\star} \in \mathbb{R}^{d^{\star}\times m}$$

- 在现实应用中为了有效降维，往往仅需降维后的距离与原始空间中的距离尽可能接近，而不必严格相等。
- 此时可取$d' \ll d$个最大特征值构成对角矩阵$\tilde{\boldsymbol{\Lambda}} = \mathrm{diag}\left(\lambda_1, \lambda_2, \cdots, \lambda_{d'}\right)$，令$\tilde{\boldsymbol{V}}$表示相应的特征向量矩阵，

$$\boldsymbol{Z} = \tilde{\boldsymbol{\Lambda}}^{1/2}\tilde{\boldsymbol{V}}$$

多维缩放（MDS）

输入： 距离矩阵 $\mathbf{D} \in \mathbb{R}^{m \times m}$，其元素 $dist_{ij}$ 为样本 $\boldsymbol{x}_i$ 到 $\boldsymbol{x}_j$ 的距离；
低维空间维数 d'.

过程：

1: 根据式(10.7)–(10.9)计算 $dist_{i\cdot}^2$, $dist_{\cdot j}^2$, $dist_{\cdot\cdot}^2$;
2: 根据式(10.10)计算矩阵 $\mathbf{B}$;
3: 对矩阵 $\mathbf{B}$ 做特征值分解;
4: 取 $\tilde{\mathbf{\Lambda}}$ 为 d' 个最大特征值所构成的对角矩阵，$\tilde{\mathbf{V}}$ 为相应的特征向量矩阵.

输出： 矩阵 $\tilde{\mathbf{V}}\tilde{\mathbf{\Lambda}}^{1/2} \in \mathbb{R}^{m \times d'}$，每行是一个样本的低维坐标

图 10.3　MDS 算法

线性降维方法

- 一般来说，欲获得低维子空间，最简单的是对原始高维空间进行线性变换。给定d维空间中的样本 $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m) \in \mathbb{R}^{d \times m}$，变换之后得到$d' \leq d$维空间中的样本

$$\boldsymbol{Z} = \boldsymbol{W}^{\top}\boldsymbol{X},$$

其中$\boldsymbol{W}$是变换矩阵，$\boldsymbol{Z}$是样本在新空间中的表达

- 变换矩阵$\boldsymbol{W}$可视为d'个d维属性向量。换言之，z_i是原属性向量x_i在新坐标系$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}\}$中的坐标向量。若$\boldsymbol{w}_i$与$\boldsymbol{w}_j (i \neq j)$正交，则新坐标系是一个正交坐标系，此时$\boldsymbol{W}$为正交变换。

- 显然，新空间中的属性是原空间中的属性的线性组合。

- 基于线性变换来进行降维的方法称为线性降维方法，对低维子空间性质的不同要求可通过对$\boldsymbol{W}$施加不同的约束来实现。

主成分分析

- 对于正交属性空间中的样本点，如何用一个超平面对所有样本进行恰当的表达?

- 容易想到，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

- 基于最近重构性和最大可分性，能分别得到主成分分析的两种等价推导。

主成分分析—最近重构性

- 对样本进行中心化，即 $\sum_{i=1}^{m} \boldsymbol{x}_i = 0$

- 再假定投影变换后得到的新坐标系为$\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d\}$，其中$\boldsymbol{w}_i$是标准正交基向量，

$$\|\boldsymbol{w}_i\|_2 = 1, \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_j = 0 \ (i \neq j)$$

- 若丢弃新坐标系中的部分坐标，即将维度降低到$d' < d$，则样本点在低维坐标系中的投影是$\boldsymbol{z}_i = \left(z_{i1}, z_{i2}, \ldots, z_{id'}\right)$，$z_{ij}$是$\boldsymbol{x}_i$在低维坐标下第$i$维的坐标。基于$\boldsymbol{z}_i$来重构$\boldsymbol{x}_i$，则会得到

$$\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d'} z_{ij} \boldsymbol{w}_j = \left(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}\right) \boldsymbol{z}_i$$

$$= \boldsymbol{W} \boldsymbol{z}_i$$

$$\boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} = \boldsymbol{I}_{d'}$$

主成分分析—最近重构性

- 考虑整个训练集，原样本点$\boldsymbol{x}_i$与基于投影重构的样本点$\hat{\boldsymbol{x}}_i$之间的距离为

$$\sum_{i=1}^{m}\|\hat{\boldsymbol{x}}_i-\boldsymbol{x}_i\|_2^2=\sum_{i=1}^{m}\|\boldsymbol{W}\boldsymbol{z}_i-\boldsymbol{x}_i\|_2^2=\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_i-2\sum_{i}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i+\sum_{i}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i$$

$$\min_{\boldsymbol{W}}\sum_{i=1}^{m}\|\hat{\boldsymbol{x}}_i-\boldsymbol{x}_i\|_2^2=\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_i-2\sum_{i}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i+\sum_{i}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i$$

$$\Leftrightarrow\ \max_{\boldsymbol{W}}\sum_{i=1}^{m}\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i=\sum_{i=1}^{m}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i=\sum_{i=1}^{m}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i=\mathrm{tr}\left(\boldsymbol{W}^{\mathrm{T}}\sum_{i=1}^{m}\left(\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right)\boldsymbol{W}\right)$$

$$=\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W})$$

$$\boldsymbol{Z}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\ \Rightarrow\ \boldsymbol{z}_i=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i$$

主成分分析—最近重构性

$$\min_{\boldsymbol{W}} \sum_{i=1}^{m} \|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2^2 \quad \Leftrightarrow \quad \max_{\boldsymbol{W}} \operatorname{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}) \quad \text{s.t. } \boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{I}_{d'}$$

- 这就是主成分分析的优化目标

- 由于$\sum_{i=1}^{m} \boldsymbol{x}_i = 0$，$\frac{1}{m-1}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} = \frac{1}{m-1}\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$ 是协方差矩阵

主成分分析—最大可分性

- 样本点$\boldsymbol{x}_i$在新空间中超平面上的投影是$\boldsymbol{z}_i = \boldsymbol{W}^\top \boldsymbol{x}_i$，若所有样本点的投影能尽可能分开，则应该使得投影后样本点的方差最大化。

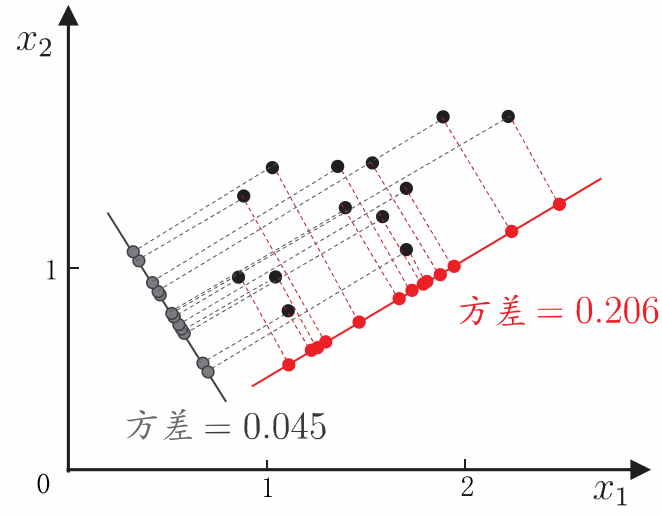
- 由于$\sum_{i=1}^{m} \boldsymbol{x}_i = 0$，$\sum_{i=1}^{m} \boldsymbol{z}_i = \boldsymbol{W}^\top \sum_{i=0}^{m} \boldsymbol{x}_i = 0$

- 投影后样本点的方差是$\frac{1}{m-1}\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^\top$，优化目标为

$$\max_{\boldsymbol{W}} \operatorname{tr}\left(\sum_{i=1}^{m} \boldsymbol{z}_i \boldsymbol{z}_i^\top\right) = \operatorname{tr}\left(\sum_{i=1}^{m} \boldsymbol{W}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{W}\right)$$

$$\Leftrightarrow \quad \max_{\boldsymbol{W}} \operatorname{tr}\left(\boldsymbol{W}^\top \left(\sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}_i^\top\right) \boldsymbol{W}\right)$$

$$\Leftrightarrow \quad \max_{\boldsymbol{W}} \operatorname{tr}(\boldsymbol{W}^\top \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{W})$$



主成分分析—求解

$$\max_{\boldsymbol{W}} \operatorname{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}) \quad \text{s.t. } \boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{I}_{d'}$$

- 使用拉格朗日乘子法可得

$$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W} = \boldsymbol{W}\boldsymbol{\Lambda}$$

只需对协方差矩阵$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$进行特征值分解，并将求得的特征值排序：$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$，再取前$d'$个特征值对应的特征向量构成$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$

这就是主成分分析的解。

主成分分析—算法

输入： 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
低维空间维数 d'.

过程：

1: 对所有样本进行中心化: $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$;
2: 计算样本的协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$;
3: 对协方差矩阵 $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ 做特征值分解;
4: 取最大的 d' 个特征值所对应的特征向量 $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}$.

输出： 投影矩阵 $\mathbf{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'})$.

图 10.5　PCA 算法

主成分分析—维度选择

- 降维后低维空间的维数d'通常是由用户事先指定，或通过在d'值不同的低维空间中对k近邻分类器（或其它开销较小的学习器）进行交叉验证来选取较好的d'值。
- 对PCA，还可从重构的角度设置一个重构阈值，例如$t = 95\%$，然后选取使下式成立的最小d'值：

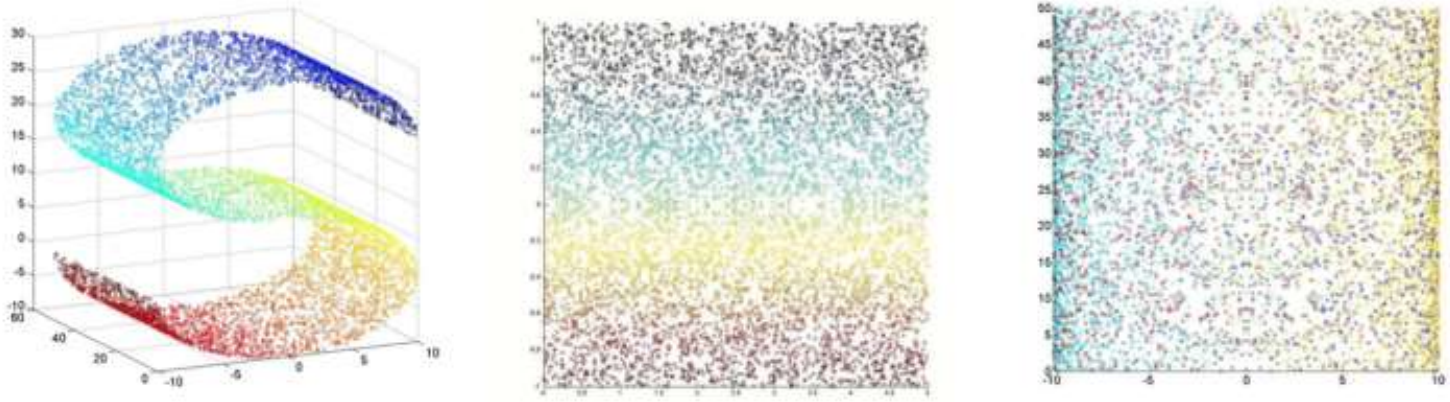
$$\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq t$$

- PCA仅需保留$\boldsymbol{W}$与样本的均值向量即可通过简单的向量减法和矩阵-向量乘法将新样本投影至低维空间中

- 降维虽会导致信息的损失，但一方面舍弃这些信息后能使得样本的采样密度增大，另一方面，当数据受到噪声影响时，最小的特征值所对应的特征向量往往与噪声有关，舍弃可以起到去噪效果

核化线性降维

- 线性降维方法假设从高维空间到低维空间的函数映射是线性的，然而，在不少现实任务中，可能需要非线性映射才能找到恰当的低维嵌入：



(a) 三维空间中的观察　(b) 本真二维结构　(c) PCA 降维结果

图 10.6　三维空间中观察到的 3000 个样本点，是从本真二维空间中矩形区域采样后以 S 形曲面嵌入，此情形下线性降维会丢失低维结构. 图中数据点的染色显示出低维空间的结构.

核化主成分分析 (KPCA)

- 非线性降维的一种常用方法，是基于核技巧对线性降维方法进行“核化” (kernelized)。

- 假定我们将在高维特征空间中把数据$\phi(\boldsymbol{x})$投影到由$\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_d)$确定的超平面上，即PCA欲求解

$$\left(\sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^{\mathrm{T}}\right)\boldsymbol{w}_j = \lambda_j \boldsymbol{w}_j \quad \Rightarrow \quad \sum_{l=1}^{m} K_{il}\alpha_l^j = \lambda_j \alpha_i^j \quad \Rightarrow \quad \boldsymbol{K}\boldsymbol{\alpha}^j = \lambda_j \boldsymbol{\alpha}^j$$

$$\phi(\boldsymbol{x}_i)^{\mathrm{T}}\left(\sum_{l=1}^{m}\phi(\boldsymbol{x}_l)\phi(\boldsymbol{x}_l)^{\mathrm{T}}\right)\boldsymbol{w}_j = \lambda_j \phi(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{w}_j \qquad \sum_{l=1}^{m} K_{il}\phi(\boldsymbol{x}_l)^{\mathrm{T}}\boldsymbol{w}_j = \lambda_j \phi(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{w}_j$$

$$\boldsymbol{w}_j = \frac{1}{\lambda_j}\left(\sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)^{\mathrm{T}}\right)\boldsymbol{w}_j = \sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\frac{\phi(\boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{w}_j}{\lambda_j} = \sum_{i=1}^{m}\phi(\boldsymbol{x}_i)\alpha_i^j$$

核化主成分分析 (KPCA)

KPCA欲求解 $\boldsymbol{K}\boldsymbol{\alpha}^j = \lambda_j \boldsymbol{\alpha}^j$

- 对新样本$\boldsymbol{x}$，其投影后的第$j(j = 1,2,\ldots,d')$维坐标为

$$z_j = \boldsymbol{w}_j^{\mathrm{T}} \phi(\boldsymbol{x}) = \sum_{i=1}^{m} \phi(\boldsymbol{x}_i)^{\mathrm{T}} \alpha_i^j \, \phi(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i^j \, \kappa(\boldsymbol{x}_i, \boldsymbol{x})$$

$$\boldsymbol{w}_j = \sum_{i=1}^{m} \phi(\boldsymbol{x}_i) \alpha_i^j$$

由该式可知，为获得投影后的坐标，KPCA需对所有样本求和，因此它的计算开销较大。

流形学习

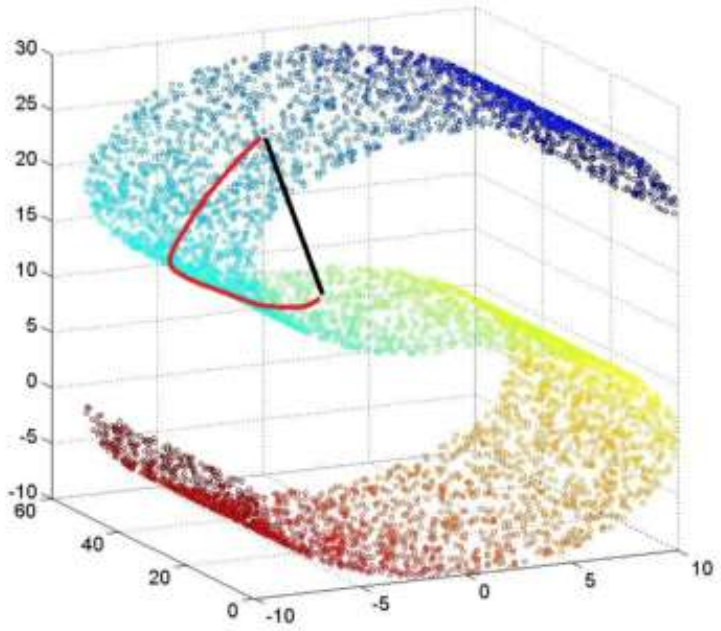
- 流形学习(manifold learning)是一类借鉴了拓扑流形概念的降维方法。“流形”是在局部与欧氏空间同胚的空间，换言之，它在局部具有欧氏空间的性质，能用欧氏距离来进行距离计算。

- 若低维流形嵌入到高维空间中，则数据样本在高维空间的分布虽然看上去非常复杂，但在局部上仍具有欧氏空间的性质，因此，可以容易地在局部建立降维映射关系，然后再设法将局部映射关系推广到全局。

- 当维数被降至二维或三维时，能对数据进行可视化展示，因此流形学习也可被用于可视化。

等度量映射(Isometric Mapping，Isomap)

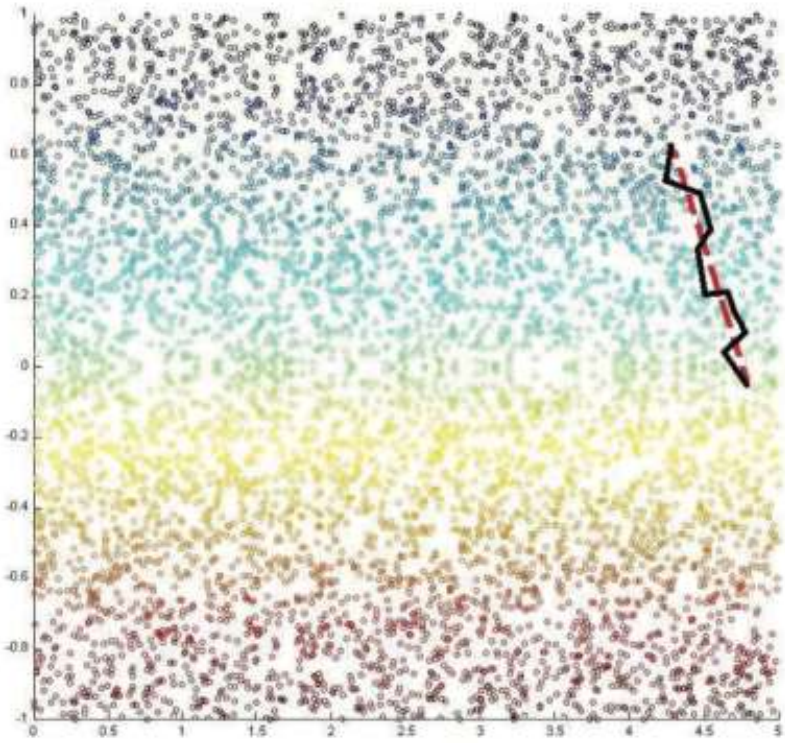
- 低维流形嵌入到高维空间之后，直接在高维空间中计算直线距离具有误导性，因为高维空间中的直线距离在低维嵌入流形上不可达。而低维嵌入流形上两点间的本真距离是“测地线”(geodesic)距离。



(a) 测地线距离与高维直线距离

等度量映射(Isometric Mapping，Isomap)

- 测地线距离的计算：利用流形在局部上与欧氏空间同胚这个性质，对每个点基于欧氏距离找出其近邻点，然后就能建立一个近邻连接图，图中近邻点之间存在连接，而非近邻点之间不存在连接，
- 于是，计算两点之间测地线距离的问题，就转变为计算近邻连接图上两点之间的最短路径问题。
- 最短路径的计算可通过Dijkstra算法或Floyd算法实现。得到距离后可通过多维缩放方法获得样本点在低维空间中的坐标。



(b) 测地线距离与近邻距离

等度量映射(Isometric Mapping，Isomap)

输入: 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
近邻参数 k;
低维空间维数 d'.

过程:

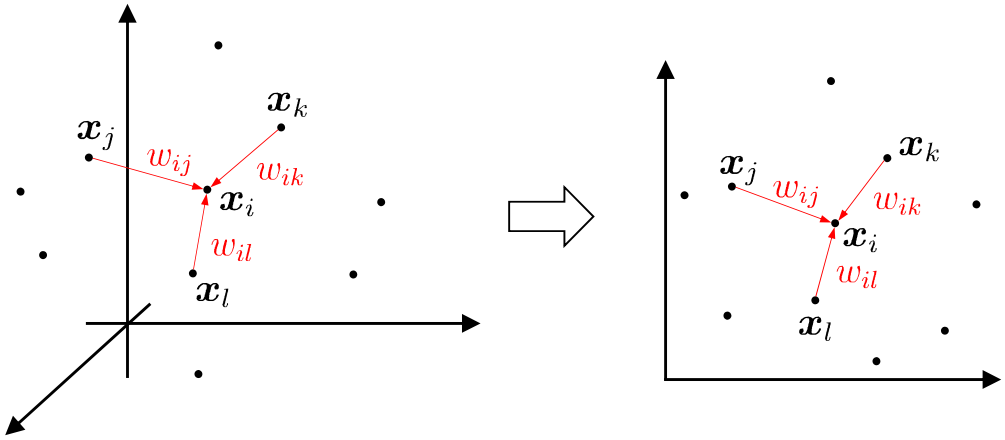
1: **for** $i = 1, 2, \ldots, m$ **do**
2: 确定 $\boldsymbol{x}_i$ 的 k 近邻;
3: $\boldsymbol{x}_i$ 与 k 近邻点之间的距离设置为欧氏距离, 与其他点的距离设置为无穷大;
4: **end for**
5: 调用最短路径算法计算任意两样本点之间的距离 $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
6: 将 $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 作为 MDS 算法的输入;
7: **return** MDS 算法的输出

输出: 样本集 D 在低维空间的投影 $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$.

图 10.8 Isomap 算法

局部线性嵌入 (Locally Linear Embedding，LLE)

- 局部线性嵌入试图保持邻域内的线性关系，并使得该线性关系在降维后的空间中继续保持。



$$\boldsymbol{x}_i = w_{ij}\boldsymbol{x}_j + w_{ik}\boldsymbol{x}_k + w_{il}\boldsymbol{x}_l$$

局部线性嵌入 (Locally Linear Embedding，LLE)

- LLE先为每个样本$\boldsymbol{x}_i$找到其近邻下标集合Q_i，然后计算出基于Q_i的中的样本点对$\boldsymbol{x}_i$进行线性重构的系数$\boldsymbol{w}_i$

$$\min_{\boldsymbol{w}_i} \left\| \boldsymbol{x}_i - \sum_{j \in Q_i} w_{ij} \boldsymbol{x}_j \right\|_2^2 \quad \text{s.t.} \sum_{j \in Q_i} w_{ij} = 1 \Rightarrow \boldsymbol{w}_i^{\mathsf{T}} \mathbf{1} = 1$$

$$\Rightarrow \min_{\boldsymbol{w}_i} \left\| \boldsymbol{x}_i \mathbf{1}^{\mathsf{T}} \boldsymbol{w}_i - \boldsymbol{X}_i^{\mathsf{T}} \boldsymbol{w}_i \right\|_2^2 \qquad \boldsymbol{w}_i = \left[w_{i_1}, w_{i_1}, \ldots, w_{i_{|Q_i|}} \right]^{\mathsf{T}}$$

$$\Rightarrow \min_{\boldsymbol{w}_i} \left\| \left(\boldsymbol{x}_i \mathbf{1}^{\mathsf{T}} - \boldsymbol{X}_i^{\mathsf{T}} \right) \boldsymbol{w}_i \right\|_2^2 \qquad \boldsymbol{X}_i = \left[\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_{|Q_i|}} \right]^{\mathsf{T}}$$

- 引入拉格朗日乘子，求梯度，并令其梯度等于0可得

$$\left(\boldsymbol{x}_i \mathbf{1}^{\mathsf{T}} - \boldsymbol{X}_i^{\mathsf{T}} \right)^{\mathsf{T}} \left(\boldsymbol{x}_i \mathbf{1}^{\mathsf{T}} - \boldsymbol{X}_i^{\mathsf{T}} \right) \boldsymbol{w}_i = \lambda \mathbf{1}$$

局部线性嵌入 (Locally Linear Embedding，LLE)

$$\left(\boldsymbol{x}_i\mathbf{1}^{\mathsf{T}}-\boldsymbol{X}_i^{\mathsf{T}}\right)^{\mathsf{T}}\left(\boldsymbol{x}_i\mathbf{1}^{\mathsf{T}}-\boldsymbol{X}_i^{\mathsf{T}}\right)\boldsymbol{w}_i=\lambda\mathbf{1} \quad \Rightarrow \quad \boldsymbol{C}\boldsymbol{w}_i=\lambda\mathbf{1}$$

令$C_{jk}=\left(\boldsymbol{x}_i-\boldsymbol{x}_j\right)^{\mathsf{T}}\left(\boldsymbol{x}_i-\boldsymbol{x}_j\right)$ $\Rightarrow$ $\boldsymbol{w}_i=\lambda\boldsymbol{C}^{-1}\mathbf{1}$

- w_{ij}的最终闭形式解为 $$w_{ij}=\frac{\sum_{k\in Q_i}C_{jk}^{-1}}{\sum_{l,s\in Q_i}C_{ls}^{-1}}$$

局部线性嵌入 (Locally Linear Embedding，LLE)

- LLE在低维空间中保持$\boldsymbol{w}_i$不变，于是$\boldsymbol{x}_i$对应的低维空间坐标$\boldsymbol{z}_i$可通过下式求解：

$$\min_{\boldsymbol{z}_1,\boldsymbol{z}_2,\ldots,\boldsymbol{z}_m} \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \sum_{j\in Q_i} w_{ij}\boldsymbol{z}_j \right\|_2^2 = \sum_{i=1}^{m} \left\| \boldsymbol{z}_i - \boldsymbol{Z}^{\top}\boldsymbol{W}_i \right\|_2^2$$

$$(\boldsymbol{W})_{ij} = w_{ij}$$

$$= \left\| \boldsymbol{Z}^{\top} - \boldsymbol{Z}^{\top}\boldsymbol{W} \right\|_F^2$$

$$\min_{\boldsymbol{W}} \left\| \boldsymbol{Z}^{\top} - \boldsymbol{Z}^{\top}\boldsymbol{W} \right\|_F^2 = \mathrm{tr}(\boldsymbol{Z}^{\top}\underbrace{(\boldsymbol{I}-\boldsymbol{W})(\boldsymbol{I}-\boldsymbol{W})^{\top}}_{\boldsymbol{M}}Z)$$

$$\text{s.t.}\ \boldsymbol{Z}^{\top}\boldsymbol{Z} = \boldsymbol{I}$$

该优化可以通过特征值分解得以求解

局部线性嵌入 (Locally Linear Embedding，LLE)

输入: 样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
近邻参数 k;
低维空间维数 d'.

过程:

1: **for** $i = 1, 2, \ldots, m$ **do**
2: 确定 $\boldsymbol{x}_i$ 的 k 近邻;
3: 从式(10.27)求得 w_{ij}, $j \in Q_i$;
4: 对于 $j \notin Q_i$, 令 $w_{ij} = 0$;
5: **end for**
6: 从式(10.30)得到 $\mathbf{M}$;
7: 对 $\mathbf{M}$ 进行特征值分解;
8: **return** $\mathbf{M}$ 的最小 d' 个特征值对应的特征向量

输出: 样本集 D 在低维空间的投影 $Z = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$.

图 10.10 LLE 算法

度量学习—研究动机

- 在机器学习中，对高维数据进行降维的主要目的是希望找到一个合适的低维空间，在此空间中进行学习能比原始空间性能更好。事实上，每个空间对应了在样本属性上定义的一个距离度量，而寻找合适的空间，实质上就是在寻找一个合适的距离度量。那么，为何不直接尝试“学习”出一个合适的距离度量呢?

度量学习

- 欲对距离度量进行学习，必须有一个便于学习的距离度量表达形式。对两个d维样本$\boldsymbol{x}_i$和$\boldsymbol{x}_j$，它们之间的平方欧氏距离可写为

$$dist_{ed}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = dist_{ij,1}^2 + dist_{ij,2}^2 + \cdots + dist_{ij,d}^2$$

- 其中$dist_{ij,k}^2$表示 $\boldsymbol{x}_i$和$\boldsymbol{x}_j$在第k维上的距离。若假定不同属性的重要性不同，则可引入属性权重$\boldsymbol{w}$，得到

$$dist_{ed}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = w_1 \cdot dist_{ij,1}^2 + w_2 \cdot dist_{ij,2}^2 + \cdots + w_d \cdot dist_{ij,d}^2$$

$$= (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \boldsymbol{W} (\boldsymbol{x}_i - \boldsymbol{x}_j)$$

- 其中$w_i \geq 0, \boldsymbol{W} = \mathrm{diag}(\boldsymbol{w})$是一个对角矩阵，$(\boldsymbol{W})_{ii} = w_{ii}$，可通过学习确定。

度量学习

- $\boldsymbol{W}$的非对角元素均为零，意味着坐标轴正交，即属性之间无关
- 但现实问题中往往不是这样，例如考虑西瓜的“重量”和“体积”这两个属性，它们显然是正相关的，其对应的坐标轴不再正交。
- 为此将$\boldsymbol{W}$替换为一个普通的半正定对称矩阵$\boldsymbol{M}$，于是就得到了马氏距离(Mahalanobis distance)

$$dist_{ed}^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \boldsymbol{M} (\boldsymbol{x}_i - \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\boldsymbol{M}}^2$$

- 其中$\boldsymbol{M}$亦称“度量矩阵”，而度量学习则是对$\boldsymbol{M}$进行学习。注意到为了保持距离非负且对称，$\boldsymbol{M}$必须是（半）正定对称矩阵，即必有矩阵$\boldsymbol{P}$使得$\boldsymbol{M}$能写为$\boldsymbol{M} = \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}$

- 对$\boldsymbol{M}$进行学习当然要设置一个目标。假定我们是希望提高近邻分类器的性能，则可将$\boldsymbol{M}$直接嵌入到近邻分类器的评价指标中去，通过优化该性能指标相应地求得$\boldsymbol{M}$

近邻成分分析(Neighbourhood Component Analysis, NCA)

- 近邻成分分析在进行判别时通常使用多数投票法，邻域中的每个样本投1票，邻域外的样本投0票。不妨将其替换为概率投票法。对于任意样本$\boldsymbol{x}_j$，它对$\boldsymbol{x}_i$分类结果影响的概率为

$$p_{ij} = \frac{\exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_M^2\right)}{\sum_l \exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_l\right\|_M^2\right)}$$

- 当$i = j$时，p_{ij}最大。显然，$\boldsymbol{x}_i$对$\boldsymbol{x}_j$的影响随着它们之间距离的增大而减小。

近邻成分分析(Neighbourhood Component Analysis, NCA)

- 若以留一法(LOO)正确率的最大化为目标，则可计算$\boldsymbol{x}_i$的留一法正确率，即它被自身之外的所有样本正确分类的概率为

$$p_i = \sum_{j \in \Omega_i} p_{ij} \qquad p_{ij} = \frac{\exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_M^2\right)}{\sum_l \exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_l\right\|_M^2\right)}$$

其中Ω_i表示与$\boldsymbol{x}_i$属于相同类别的样本的下标集合

- 整个样本集上的留一法正确率为 $\sum_{i=1}^{m} p_i = \sum_{i}^{m} \sum_{j \in \Omega_i} p_{ij}$

- 由于$\boldsymbol{M} = \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}}$，则NCA的优化目标为

$$\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^{\mathrm{T}} \boldsymbol{M} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)$$

$$\Rightarrow \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^{\mathrm{T}} \boldsymbol{P}\boldsymbol{P}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)$$

$$\min_{\boldsymbol{P}} 1 - \sum_{i=1}^{m} \sum_{j \in \Omega_i} \frac{\exp\left(-\left\|\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_j\right\|_2^2\right)}{\sum_l \exp\left(-\left\|\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_l\right\|_2^2\right)}$$

求解即可得到最大化近邻分类器LOO正确率的距离度量矩阵$\boldsymbol{M}$

度量学习

- 实际上，我们不仅能把错误率这样的监督学习目标作为度量学习的优化目标，还能在度量学习中引入领域知识。
- 若已知某些样本相似、某些样本不相似，则可定义“必连”(must-link) 约束集合$\mathcal{M}$ 与“勿连”(cannot-link)约束集合$\mathcal{C}$：
  - $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M}$，表示$\boldsymbol{x}_i$与$\boldsymbol{x}_j$相似
  - $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}$，表示$\boldsymbol{x}_i$与$\boldsymbol{x}_j$不相似

- 显然，我们希望相似的样本之间距离较小，不相似的样本之间距离较大，可通过求解下面这个凸优化问题获得适当的度量矩阵$\boldsymbol{M}$

$$\min_{\mathrm{M}} \sum_{(x_i,x_j)\in\mathcal{M}} \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_{\boldsymbol{M}}^2 \quad \text{s. t.} \sum_{(x_i,x_j)\in\mathcal{C}} \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_{\boldsymbol{M}}^2 \geq 1 \text{ and } \boldsymbol{M} \succcurlyeq 0$$

$\boldsymbol{M}$必须是半正定的

- 上式要求在不相似样本间的距离不小于1的前提下，使相似样本间的距离尽可能小

度量学习

- 不同的度量学习方法针对不同目标获得“好”的半正定对称距离度量矩阵$\boldsymbol{M}$，若$\boldsymbol{M}$是一个低秩矩阵，则通过对$\boldsymbol{M}$进行特征值分解，总能找到一组正交基，其正交基数目为矩阵$\boldsymbol{M}$的秩$\mathrm{rank}(\boldsymbol{M})$，小于原属性数$d$。
- 于是，度量学习学得的结果可衍生出一个降维矩阵$\boldsymbol{P} \in \mathbb{R}^{d \times \mathrm{rank}(\boldsymbol{M})}$

习题

- 记$\mathrm{err}^{\star}(\boldsymbol{x})=1-\max_{c\in\mathcal{Y}}P(c|\boldsymbol{x})$，$\mathrm{err}(\boldsymbol{x})=1-\sum_{c}P(c|\boldsymbol{x})P(c|\boldsymbol{z})$，其中$\boldsymbol{z}$为$\boldsymbol{x}$的最近邻，试证明在样本无穷多时

$$\mathrm{err}^{\star}(\boldsymbol{x})\leq\mathrm{err}(\boldsymbol{x})\leq\mathrm{err}^{\star}(\boldsymbol{x})\left(2-\frac{|\mathcal{Y}|}{|\mathcal{Y}|-1}\times\mathrm{err}^{\star}(\boldsymbol{x})\right)$$

  提示：柯西-施瓦兹不等式$(\sum_i a_i)^2\leq n\left(\sum_i a_i^2\right)$

- 10.4

- 求解20页的优化问题

附加题：令$\mathbf{M}=PP^T$，那么下列问题还是凸优化问题吗？试证明之。

$$\min_{P}\sum_{(x_i,x_j)\in\mathcal{M}}\|x_i-x_j\|_{\mathbf{M}}^2\quad\text{s. t.}\quad\sum_{(x_i,x_j)\in\mathcal{C}}\|x_i-x_j\|_{\mathbf{M}}^2\geq 1$$

第十一章：特征选择与稀疏学习

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

特征

- 特征
  - 描述物体的属性

- 特征的分类
  - 相关特征: 对**当前学习任务**有用的属性
  - 无关特征: 与**当前学习任务**无关的属性
  - 冗余特征: 其所包含信息能由其他特征推演出来

例子：西瓜的特征

西瓜的**特征**

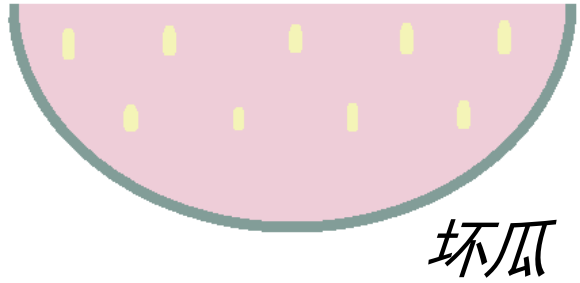
- 颜色
- 纹理
- 触感

无关特征

- 根蒂
- 声音

相关特征

当前任务：西瓜是否是好瓜



坏瓜

好瓜

特征选择

- 特征选择
  - 从给定的特征集合中选出**任务相关**特征子集
  - 必须确保不丢失重要特征

- 原因
  - 减轻维度灾难：在少量属性上构建模型
  - 降低学习难度：留下关键信息

例子：判断是否好瓜时的特征选择

西瓜的**特征**

- 颜色
- 纹理
- 触感

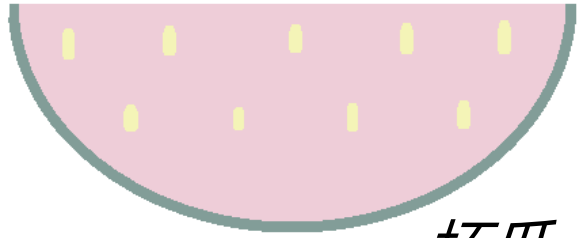
无关特征

- 根蒂
- 声音

相关特征

特征选择：选择当前任务相关特征

当前任务：西瓜是否是好瓜



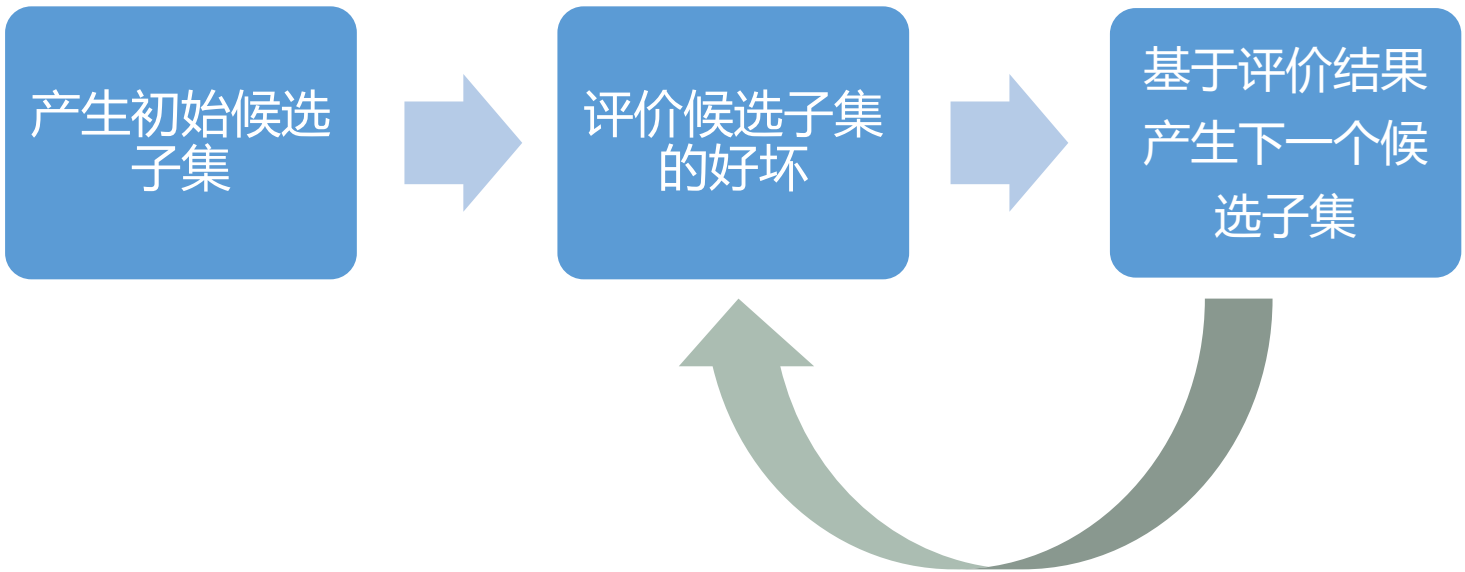
坏瓜

好瓜

特征选择的一般方法

- 遍历所有可能的子集
  - 计算上遭遇组合爆炸，**不可行**

- 可行方法



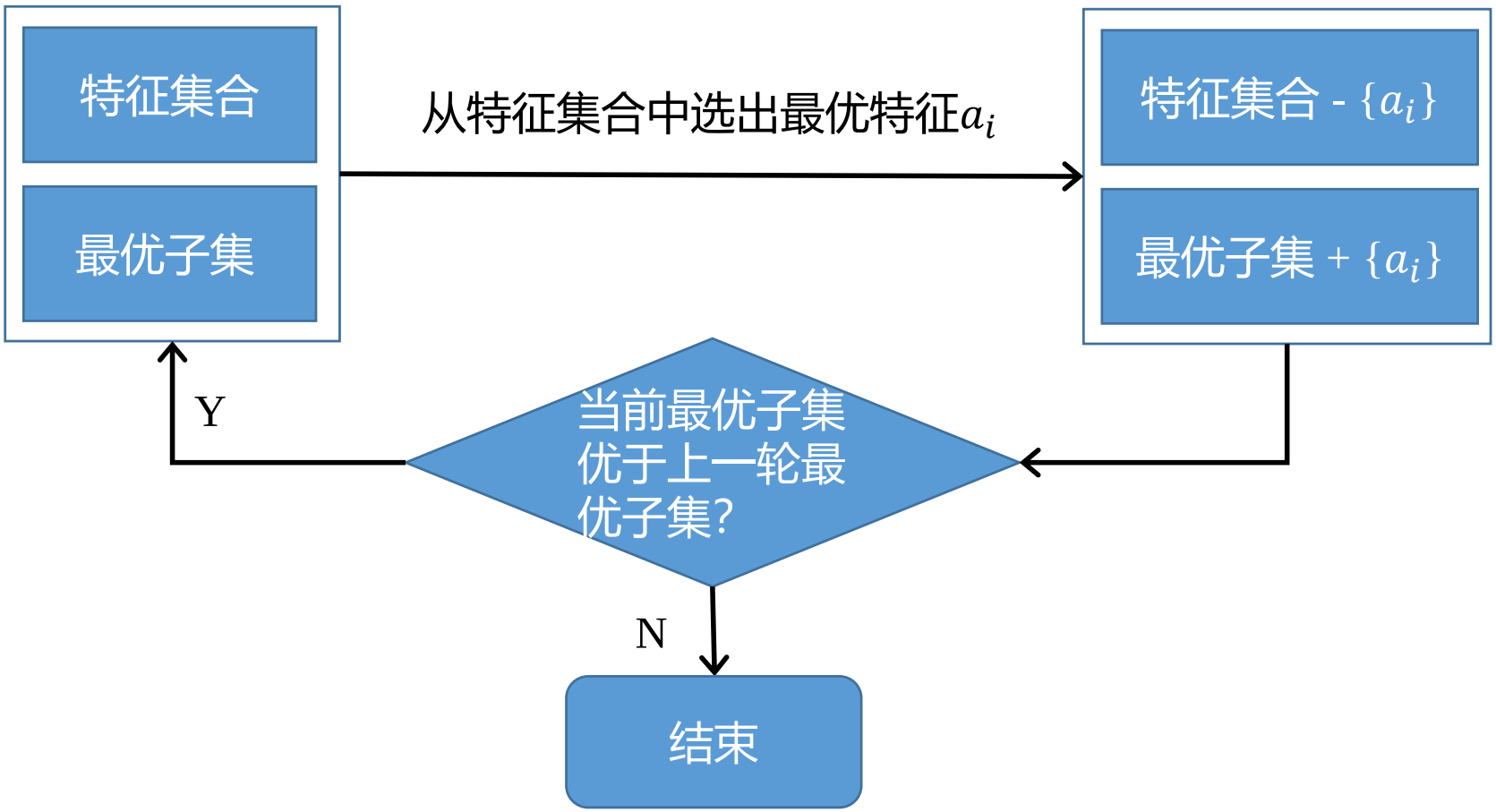
两个关键环节：子集搜索和子集评价

子集搜索

- 用贪心策略选择包含重要信息的特征子集
  - 前向搜索：逐渐增加相关特征
  - 后向搜索：从完整的特征集合开始，逐渐减少特征
  - 双向搜索：每一轮逐渐增加相关特征，同时减少无关特征

前向搜索

- 最优子集初始为空集，特征集合初始时包括所有给定特征



子集评价

- 特征子集确定了对数据集的一个划分
  - 每个划分区域对应着特征子集的某种取值

- 样本标记对应着对数据集的真实划分

通过估算这两个划分的差异，就能对特征子集进行评价；与样本标记对应的划分的差异越小，则说明当前特征子集越好

用信息熵进行子集评价

- 特征子集A确定了对数据集D的一个划分
  - A上的取值将数据集D分为V份，每一份用D^v表示
  - $\mathrm{Ent}(D^v)$表示D^v上的信息熵

- 样本标记Y对应着对数据集D的真实划分
  - $\mathrm{Ent}(D)$表示D上的信息熵

D上的信息熵定义为

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|y|} p_k \log_2 p_k$$

第i类样本所占比例为p_i

特征子集A的信息增益为：

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

常见的特征选择方法

- 将特征子集搜索机制与子集评价机制相结合，即可得到特征选择方法

常见的特征选择方法大致分为如下三类：

- 过滤式
- 包裹式
- 嵌入式

过滤式选择

先用特征选择过程过滤原始数据，再用过滤后的特征来训练模型；特征选择过程与后续学习器无关

- Relief (Relevant Features) 方法 [Kira and Rendell, 1992]
  - 为每个初始特征赋予一个“**相关统计量**”，度量特征的重要性
  - 特征子集的重要性由子集中每个特征所对应的相关统计量之和决定
  - 设计一个阈值，然后选择比阈值大的相关统计量分量所对应的特征
  - 或者指定欲选取的特征个数，然后选择相关统计量分量最大的指定个数特征

如何确定相关统计量？

Relief方法中相关统计量的确定

- 猜对近邻（near-hit）：$\boldsymbol{x}_i$的同类样本中的最近邻$\boldsymbol{x}_{i,nh}$
- 猜错近邻（near-miss）：$\boldsymbol{x}_i$的异类样本中的最近邻$\boldsymbol{x}_{i,nm}$
- 相关统计量对应于属性j的分量为

$$\delta^j = \sum_i -\mathrm{diff}\left(x_i^j, x_{i,nh}^j\right)^2 + \mathrm{diff}\left(x_i^j, x_{i,nm}^j\right)^2$$

若j为离散型，则$x_a^j = x_b^j$时$\mathrm{diff}\left(x_a^j, x_b^j\right) = 0$，否则为1；
若j为连续型，则$\mathrm{diff}\left(x_a^j, x_b^j\right) = |x_a^j - x_b^j|$，注意$x_a^j$，$x_b^j$已规范化到[0,1]区间

- 相关统计量越大，属性j上，猜对近邻比猜错近邻越近，即属性j对区分对错越有用
- Relief方法的时间开销随采样次数以及原始特征数线性增长，运行效率很高

Relief方法的多类拓展

- Relief方法是为二分类问题设计的，其扩展变体Relief-F [Kononenko, 1994]能处理多分类问题
- 数据集中的样本来自$|\mathcal{Y}|$个类别，其中$\boldsymbol{x}_i$属于第k类
- 猜中近邻：第k类中$\boldsymbol{x}_i$的最近邻$\boldsymbol{x}_{i,nh}$
- 猜错近邻：第k类之外的每个类中找到一个$\boldsymbol{x}_i$的最近邻作为猜错近邻，记为$\boldsymbol{x}_{i,l,nm}(l = 1,2,\ldots, \quad ; l \neq k)$
- 相关统计量对应于属性的分量为

$$\delta^j = \sum_i \left(-\mathrm{diff}\left(x_i^j, x_{i,nh}^j\right)^2 + \sum_{l \neq k} p_l \times \mathrm{diff}\left(x_i^j, x_{i,l,nm}^j\right)^2 \right)$$

p_l为第l类样本在数据集D中所占的比例

包裹式选择

- 包裹式选择直接把最终将要使用的学习器的性能作为特征子集的评价准则
  - 包裹式特征选择的目的就是为给定学习器选择最有利于其性能、“量身定做”的特征子集
  - 包裹式选择方法直接针对给定学习器进行优化，因此从最终学习器性能来看，包裹式特征选择比过滤式特征选择更好
  - 包裹式特征选择过程中需多次训练学习器，计算开销通常比过滤式特征选择大得多

LVW包裹式特征选择方法

- LVW（Las Vegas Wrapper）[Liu and Setiono, 1996] 在拉斯维加斯方法框架下使用随机策略来进行子集搜索，并以最终分类器的误差作为特征子集评价准则

- 基本步骤
  - 在循环的每一轮随机产生一个特征子集
  - 在随机产生的特征子集上通过交叉验证推断当前特征子集的误差
  - 进行多次循环，在多个随机产生的特征子集中选择误差最小的特征子集作为最终解*

*若有运行时间限制，则该算法有可能给不出解

嵌入式选择

- 嵌入式特征选择是将特征选择过程与学习器训练过程融为一体，两者在同一个优化过程中完成，在学习器训练过程中自动地进行特征选择

- 考虑最简单的线性回归模型，以平方误差为损失函数，并引入L_2范数正则化项防止过拟合，则有

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^{\mathsf{T}} \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|_2^2$$

岭回归 (ridge regression)
[Tikhonov and Arsenin, 1977]

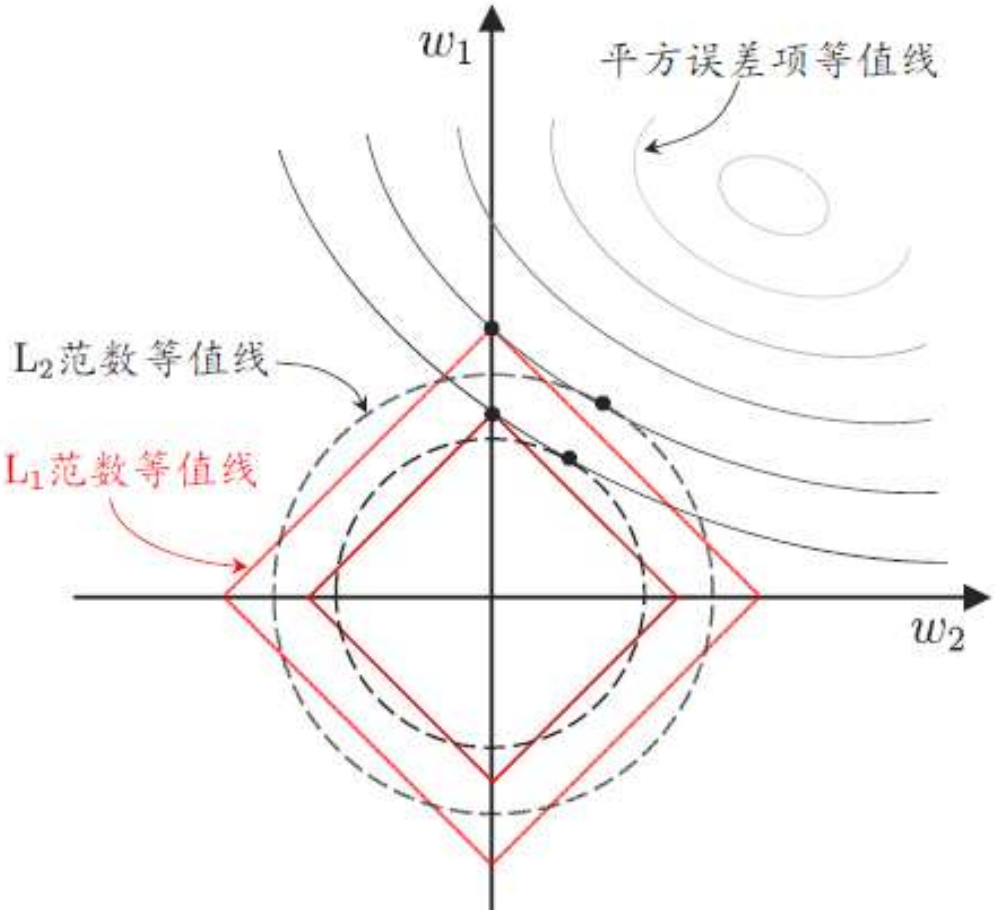
- 将L_2范数替换为L_1范数，则有**LASSO** [Tibshirani, 1996]

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} (y_i - \boldsymbol{w}^{\mathsf{T}} \boldsymbol{x}_i)^2 + \lambda \|\boldsymbol{w}\|_1$$

$$\|\boldsymbol{w}\|_1 = \sum_{d} |w_d|$$

易获得稀疏解，是一种嵌入式特征选择方法

使用L_1范数正则化易获得稀疏解



等值线即取值相同的点的连线

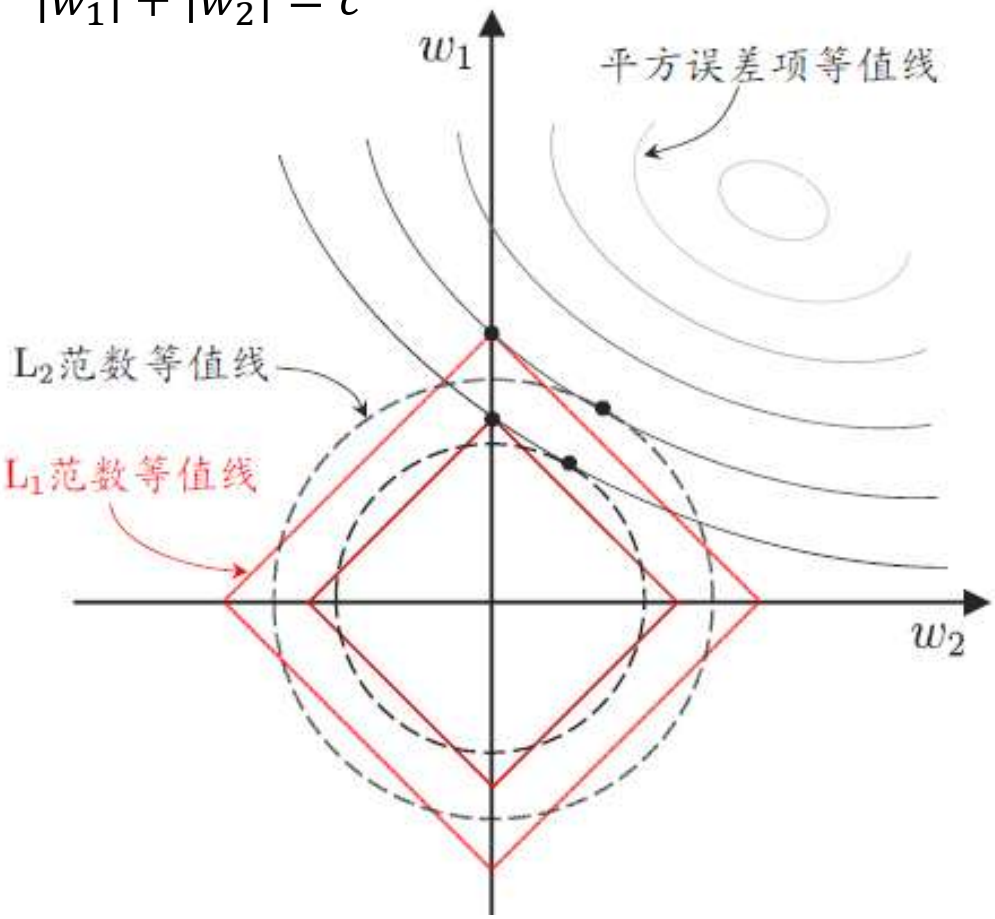
$$\|\boldsymbol{w}\|_1 = c$$

$$\Rightarrow \quad |w_1| + |w_2| = c$$

$$\begin{cases} w_1 + w_2 = c, \text{if } w_1 \geq 0, w_2 \geq 0 \\ w_1 - w_2 = c, \text{if } w_1 \geq 0, w_2 < 0 \\ -w_1 + w_2 = c, \text{if } w_1 < 0, w_2 \geq 0 \\ -w_1 - w_2 = c, \text{if } w_1 < 0, w_2 < 0 \end{cases}$$

使用L_1范数正则化易获得稀疏解

$|w_1| + |w_2| = c$



等值线即取值相同的点的连线

- 假设$\boldsymbol{x}$仅有两个属性，那么$\boldsymbol{w}$有两个分量w_1和w_2. 那么目标优化的解要在平方误差项与正则化项之间折中,即出现在图中平方误差项等值线与正则化等值线相交处.

从图中看出,采用L_1范数时交点常出现在坐标轴上,即产生w_1或者w_2为0的稀疏解

L_1正则化问题的求解(1)

- 可使用近端梯度下降(Proximal Gradient Descend，简称PGD) 求解

[Boyd and Vandenberghe, 2004]

- 考虑更一般的问题

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1$$

- 假设$f(\boldsymbol{w})$为凸函数，且$\nabla f(\boldsymbol{w})$满足L-Lipschitz条件， 即存在常数$L > 0$, 使得

$$\forall \boldsymbol{w}, \boldsymbol{w}' \quad \|\nabla f(\boldsymbol{w}') - \nabla f(\boldsymbol{w})\| \leq L\|\boldsymbol{w}' - \boldsymbol{w}\|$$

L_1正则化问题的求解(1)

- 假设$f(\boldsymbol{x})$为凸函数，且$\nabla f(\boldsymbol{x})$满足L-Lipschitz条件， 即存在常数$L > 0$, 使得

$$\forall \boldsymbol{w}, \boldsymbol{w}' \quad \|\nabla f(\boldsymbol{w}') - \nabla f(\boldsymbol{w})\| \leq L\|\boldsymbol{w}' - \boldsymbol{w}\|$$

- 等价于$g(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|^2 - f(\boldsymbol{w})$为凸函数

$$f(\boldsymbol{w}') \geq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathsf{T}}(\boldsymbol{w}' - \boldsymbol{w}) \quad + \quad f(\boldsymbol{w}) \geq f(\boldsymbol{w}') + \nabla f(\boldsymbol{w}')^{\mathsf{T}}(\boldsymbol{w} - \boldsymbol{w}')$$

$$\Rightarrow \left(\nabla f(\boldsymbol{w}) - \nabla f(\boldsymbol{w}')\right)^{\mathsf{T}}(\boldsymbol{w} - \boldsymbol{w}') \geq 0$$

$$\Rightarrow \left(\nabla f(\boldsymbol{w}) - \nabla f(\boldsymbol{w}')\right)^{\mathsf{T}}(\boldsymbol{w} - \boldsymbol{w}') \leq L\|\boldsymbol{w} - \boldsymbol{w}'\|^2$$

$$\left(\nabla g(\boldsymbol{w}') - \nabla g(\boldsymbol{w})\right)^{\mathsf{T}}(\boldsymbol{w}' - \boldsymbol{w}) = \left(L(\boldsymbol{w}' - \boldsymbol{w}) - \left(\nabla f(\boldsymbol{w}') - \nabla f(\boldsymbol{w})\right)\right)(\boldsymbol{w}' - \boldsymbol{w})$$
$$\geq 0$$

L_1正则化问题的求解(1)

- 假设$f(\boldsymbol{x})$为凸函数，且$\nabla f(\boldsymbol{x})$满足L-Lipschitz条件， 即存在常数$L > 0$, 使得

$$\forall \boldsymbol{w}, \boldsymbol{w}' \quad \|\nabla f(\boldsymbol{w}') - \nabla f(\boldsymbol{w})\| \leq L\|\boldsymbol{w}' - \boldsymbol{w}\|$$

- 等价于$g(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|^2 - f(\boldsymbol{w})$为凸函数
- 也等价于$f(\boldsymbol{w}') \leq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})(\boldsymbol{w}' - \boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{w}' - \boldsymbol{w}\|^2$

$$g(\boldsymbol{w}') \geq g(\boldsymbol{w}) + \nabla g(\boldsymbol{w})(\boldsymbol{w}' - \boldsymbol{w})$$

$$\frac{L}{2}\|\boldsymbol{w}'\|^2 - f(\boldsymbol{w}') \geq \frac{L}{2}\|\boldsymbol{w}\|^2 - f(\boldsymbol{w}) + \left(L\boldsymbol{w} - \nabla f(\boldsymbol{w})\right)^{\mathsf{T}}(\boldsymbol{w}' - \boldsymbol{w})$$

$$f(\boldsymbol{w}') \leq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathsf{T}}(\boldsymbol{w}' - \boldsymbol{w}) - L\boldsymbol{w}(\boldsymbol{w}' - \boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{w}'\|^2 - \frac{L}{2}\|\boldsymbol{w}\|^2$$

$$f(\boldsymbol{w}') \leq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})(\boldsymbol{w}' - \boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{w}' - \boldsymbol{w}\|^2$$

L_1正则化问题的求解(1)

- 假设$f(\boldsymbol{x})$为凸函数，且$\nabla f(\boldsymbol{x})$满足L-Lipschitz条件， 即存在常数$L > 0$, 使得

$$\forall \boldsymbol{w}, \boldsymbol{w}' \quad \|\nabla f(\boldsymbol{w}') - \nabla f(\boldsymbol{w})\| \leq L\|\boldsymbol{w}' - \boldsymbol{w}\|$$

- 等价于$g(\boldsymbol{w}) = \frac{L}{2}\|\boldsymbol{w}\|^2 - f(\boldsymbol{w})$为凸函数
- 也等价于$f(\boldsymbol{w}') \leq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})(\boldsymbol{w}' - \boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{w}' - \boldsymbol{w}\|^2$

$$= \frac{L}{2}\left\|\boldsymbol{w}' - \left(\boldsymbol{w} - \frac{1}{L}\nabla f(\boldsymbol{w})\right)\right\|_2^2 + \text{const}$$

$$f(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_1 \leq \frac{L}{2}\left\|\boldsymbol{w} - \left(\boldsymbol{w}^k - \frac{1}{L}\nabla f(\boldsymbol{w}^k)\right)\right\|_2^2 + \lambda\|\boldsymbol{w}\|_1 + \text{const}$$

L_1正则化问题的求解(1)

- $f(\boldsymbol{w})+\lambda\|\boldsymbol{w}\|_1 \le \frac{L}{2}\left\|\boldsymbol{w}-\left(\boldsymbol{w}^k-\frac{1}{L}\nabla f(\boldsymbol{w}^k)\right)\right\|_2^2+\lambda\|\boldsymbol{w}\|_1+\text{const}$

$$\boldsymbol{w}^{k+1}=\arg\min_{\boldsymbol{w}}\frac{L}{2}\left\|\boldsymbol{w}-\left(\boldsymbol{w}^k-\frac{1}{L}\nabla f(\boldsymbol{w}^k)\right)\right\|_2^2+\lambda\|\boldsymbol{w}\|_1$$

$$\boldsymbol{w}^{k+1}=\arg\min_{\boldsymbol{w}}\frac{L}{2}\left\|\boldsymbol{w}-\boldsymbol{z}^k\right\|_2^2+\lambda\|\boldsymbol{w}\|_1$$

$$\left[\boldsymbol{w}^{k+1}\right]_i=\arg\min_{w_i}\frac{L}{2}\left(w_i-z_i^k\right)^2+\lambda|w_i|$$

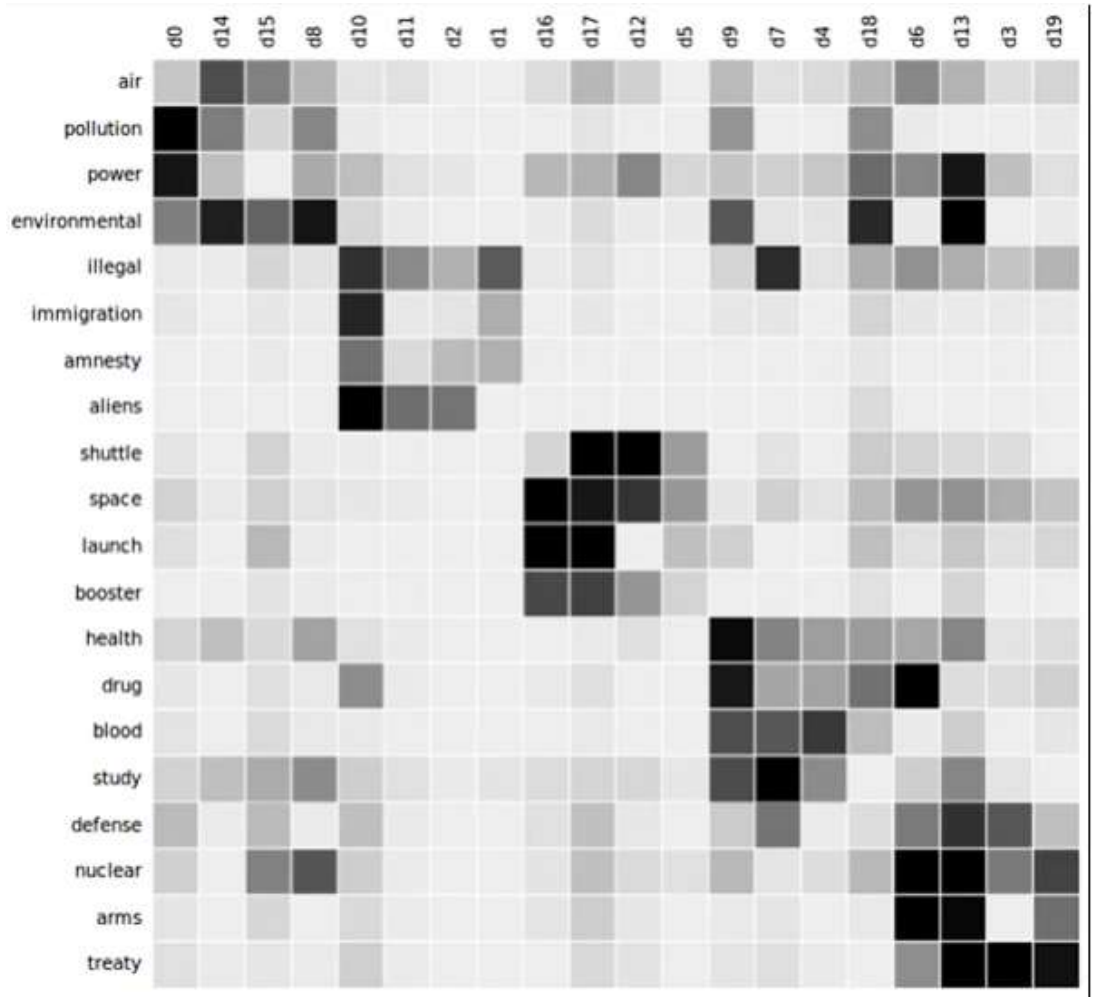
$$\frac{L}{2}\left(w_i-z_i^k\right)^2+\lambda|w_i|$$

$$=\begin{cases}\frac{L}{2}\left(w_i-z_i^k+\frac{\lambda}{L}\right)^2+const,\text{if } w_i\ge 0\\ \frac{L}{2}\left(w_i-z_i^k-\frac{\lambda}{L}\right)^2+const,\text{if } w_i<0\end{cases}$$

$$w_i=\begin{cases}z_i^k-\frac{\lambda}{L},\text{if } \frac{\lambda}{L}<z_i^k\\ 0,\text{if } \frac{\lambda}{L}\ge\left|z_i^k\right|\\ z_i^k+\frac{\lambda}{L},\text{if} z_i^k<-\frac{\lambda}{L}\end{cases}$$

稀疏表示与字典学习

- 将数据集考虑成一个矩阵，每行对应一个样本，每列对应一个特征
- 矩阵中有很多零元素，且非整行整列出现
- 稀疏表达的优势：
  - 文本数据线性可分
  - 存储高效



能否将稠密表示的数据集转化为“稀疏表示”，使其享受稀疏表达的优势？

稀疏表示与字典学习

- 一般的学习任务中，并没有字典可用，需学习出这样一个字典

- 为普通稠密表达的样本找到合适的字典，将样本转化为稀疏表示，这一过程称为字典学习

- 给定数据集$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}, \boldsymbol{x}_i \in \mathbb{R}^d$
- 最简单的字典学习的优化形式为

$$\min_{\boldsymbol{B}, \boldsymbol{\alpha}_i} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1$$

采用交替迭代优化进行求解

$\boldsymbol{B} \in \mathbb{R}^{d \times k}$为字典矩阵、$\boldsymbol{\alpha}_i \in \mathbb{R}^k$为样本的稀疏表示

k称为字典的词汇量，通常由用户指定

字典学习的解法(1)

- 固定字典**B**，参考LASSO的方法求解

$$\min_{\boldsymbol{\alpha}_i} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{\alpha}_i\|_2^2 + \lambda \sum_{i=1}^{m} \|\boldsymbol{\alpha}_i\|_1$$

- 以$\boldsymbol{\alpha}_i$为初值更新字典$\boldsymbol{B}$

$$\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m] \in \mathbb{R}^{d \times m}$$

$$\min_{\boldsymbol{B}} L = \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{B}\boldsymbol{\alpha}_i\|_2^2 = \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{A}\|_F^2$$

$$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_m] \in \mathbb{R}^{k \times m}$$

$$\nabla_{\boldsymbol{B}} L = -2(\boldsymbol{X} - \boldsymbol{B}\boldsymbol{A})\boldsymbol{A}^{\mathsf{T}} = 0 \quad \Rightarrow \quad \boldsymbol{B} = \boldsymbol{X}\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}})^{-1}$$

字典学习的解法(1)

- 当k很大的时候，求逆计算代价高，可以通过逐列更新

$$L = \|\boldsymbol{X} - \boldsymbol{B}\boldsymbol{A}\|_F^2 = \left\|\boldsymbol{X} - \sum_{i=1}^{k} \boldsymbol{b}_i \boldsymbol{a}_{(i)}^{\top}\right\|$$

$\boldsymbol{b}_i$表示字典$\boldsymbol{B}$的第i列

$\boldsymbol{a}_{(i)}$表示矩阵$\boldsymbol{A}$的第i行

$$= \left\|\left(\boldsymbol{X} - \sum_{j \neq i} \boldsymbol{b}_i \boldsymbol{a}_{(i)}^{\top}\right) - \boldsymbol{b}_i \boldsymbol{a}_{(i)}^{\top}\right\|_F^2$$

$$= \left\|\boldsymbol{E}_i - \boldsymbol{b}_i \boldsymbol{a}_{(i)}^{\top}\right\|_F^2$$

$$\nabla_{\boldsymbol{b}_i} L = -2\left(\boldsymbol{E}_i - \boldsymbol{b}_i \boldsymbol{a}_{(i)}^{\top}\right)\boldsymbol{a}_{(i)} = 0 \quad \Rightarrow \quad \boldsymbol{b}_i = \frac{\boldsymbol{E}_i \boldsymbol{a}_{(i)}}{\boldsymbol{a}_{(i)}^{\top} \boldsymbol{a}_{(i)}}$$

反复迭代以获得最优解

压缩感知

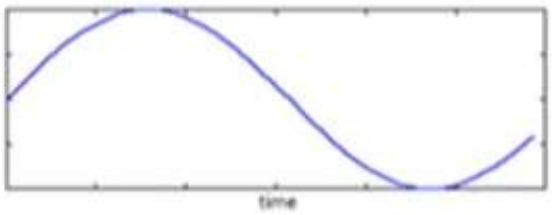
- 能否利用部分数据恢复全部数据?

- 数据传输中，能否利用接收到的压缩、丢包后的数字信号，精确重构出原信号?

- 压缩感知 (compressive sensing) [Cándes et al., 2006, Donoho, 2006] 为解决此类问题提供了新的思路.

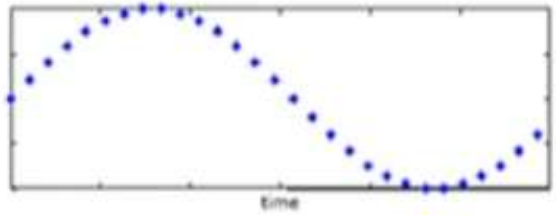
针对信号采样的技术，它通过一些手段，实现了“压缩的采样”，准确说是**在采样过程中完成了数据压缩的过程。**

模拟信号采样

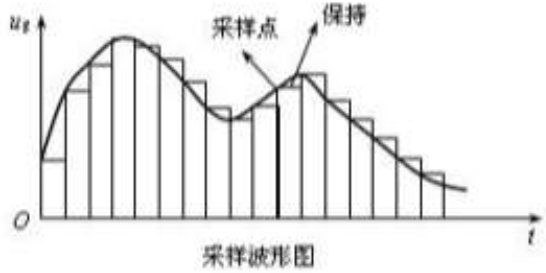


模拟信号

采样



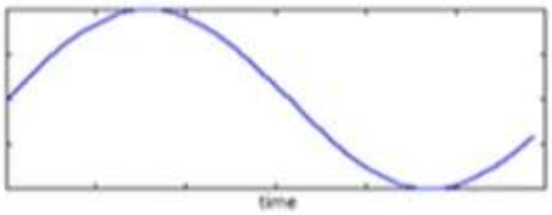
数字信号



采样波形图

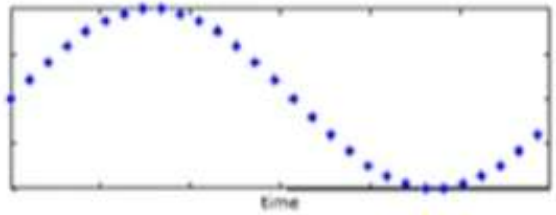
用多大的采样频率，才能完全恢复模拟信号？

模拟信号采样

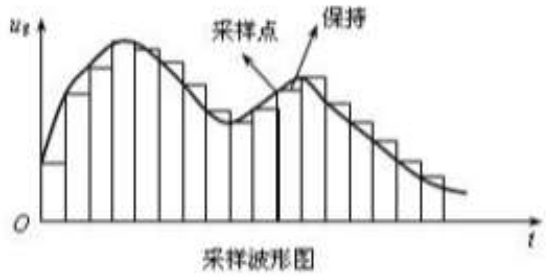


模拟信号

采样



数字信号



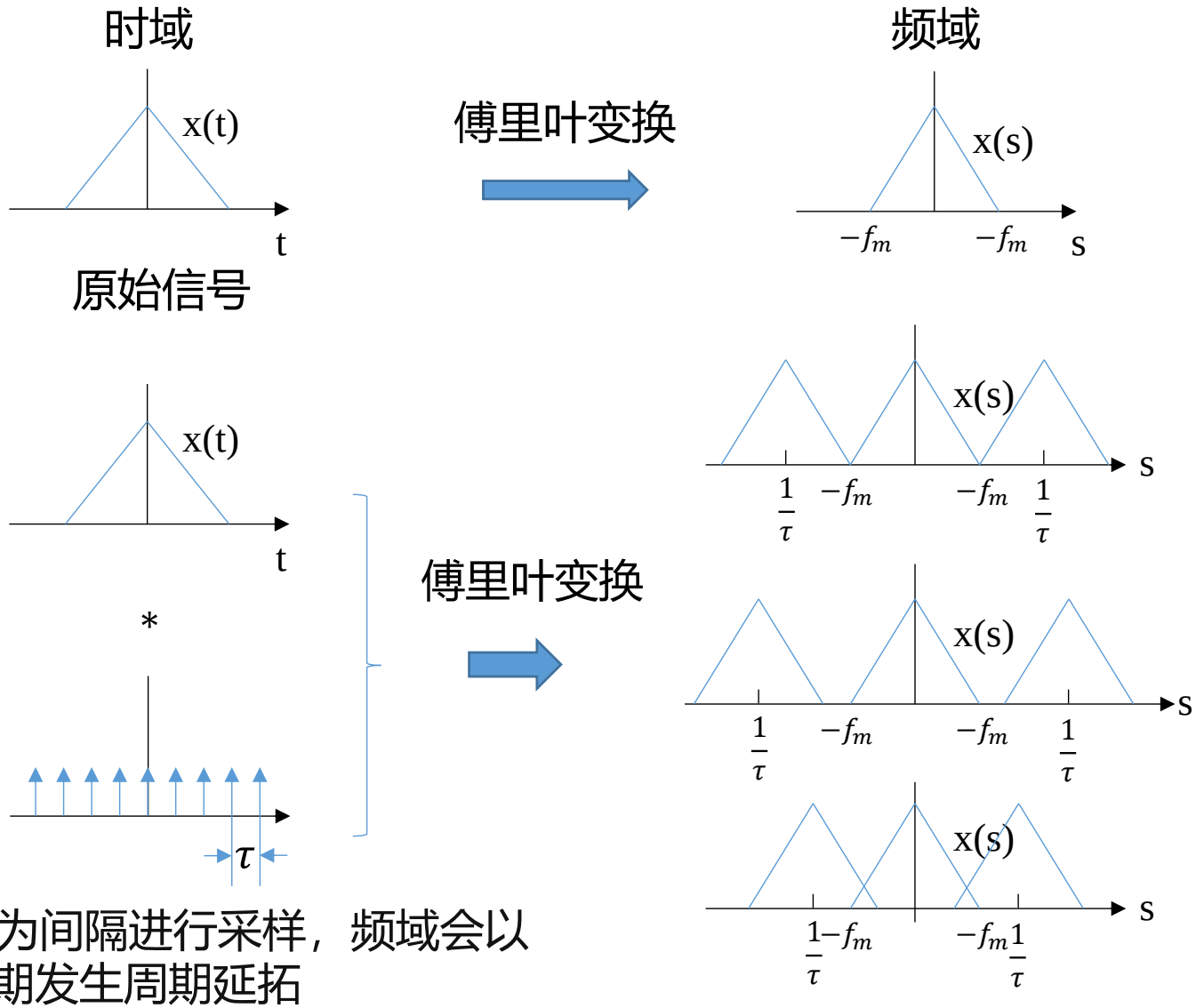
采样波形图

奈奎斯特采样定理

为了不失真地恢复模拟信号，采样频率应该大于模拟信号频谱中最高频率的2倍。

奈奎斯特
1889-1976

模拟信号采样



混叠

时域以τ为间隔进行采样，频域会以1/τ为周期发生周期延拓

压缩感知

- 2004年，陶哲轩等人证明，如果信号是稀疏的，那么可以由远低于采样定理要求的采样点重建恢复，并与2007年正式提出压缩感知这个概念

陶哲轩

Emmanuel Candes

David Donoho

压缩感知

- 采样频率应该大于模拟信号频谱中最高频率的2倍

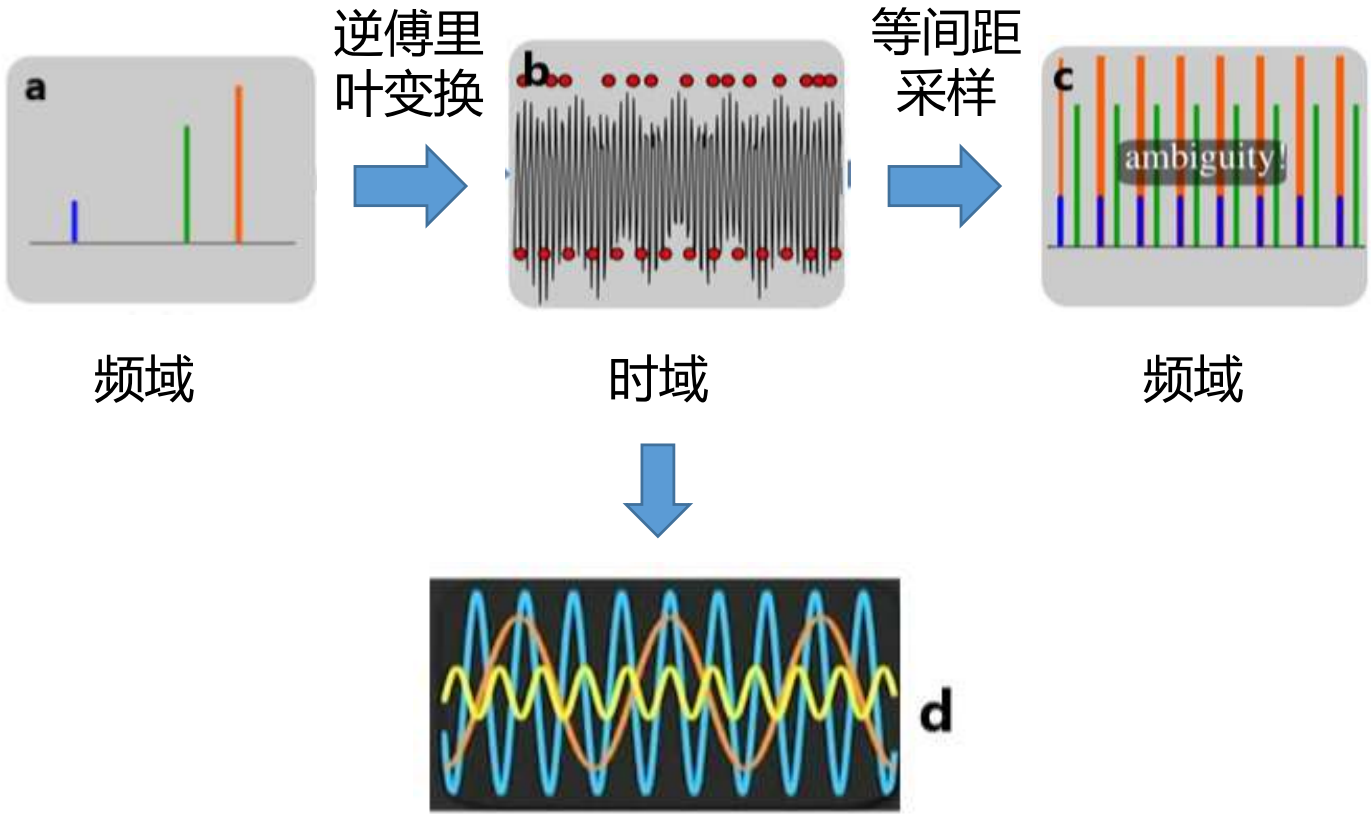
奈奎斯特采样定理

- 给定采样频率采样，意味着等间距采样

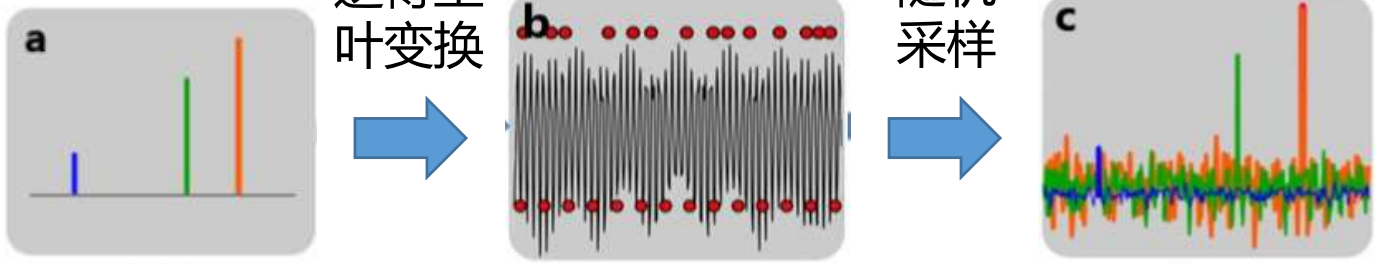
- 等间距采样，频域将以以1/τ为周期延拓，采样频率低的时候会一起混叠

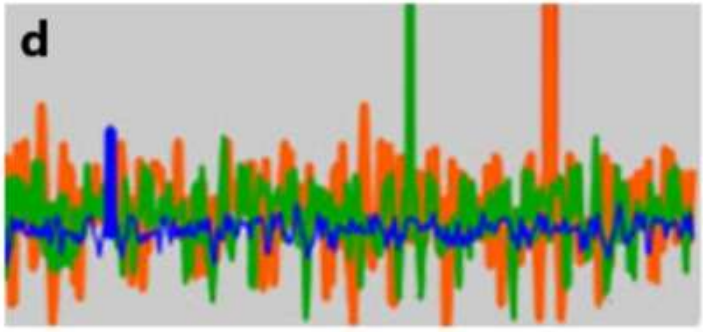
- 如果是不等间距采样或者随机采样呢?

压缩感知



压缩感知

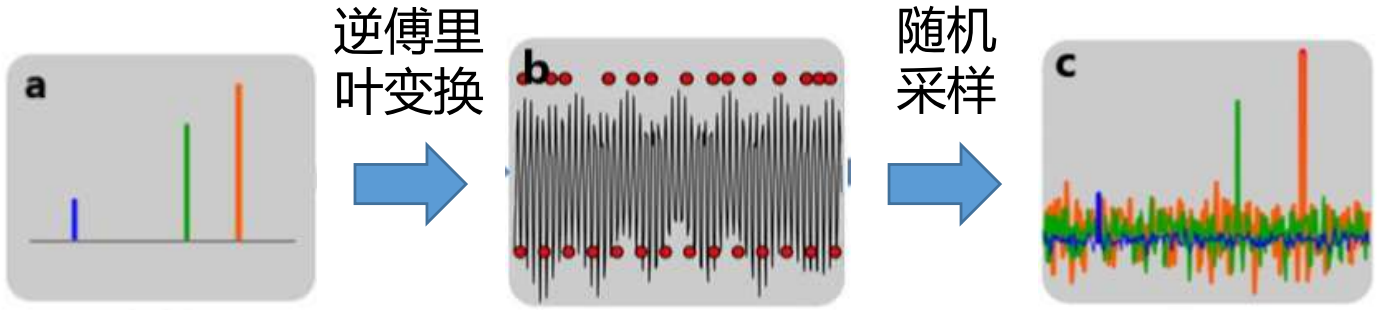




频域就不再是以固定周期进行延拓了，而是会产生大量不相关（incoherent）的干扰值

随机采样使得频谱不再是整齐地搬移，而是一小部分一小部分胡乱地搬移

压缩感知—前提条件



信号在频域稀疏

随机采样

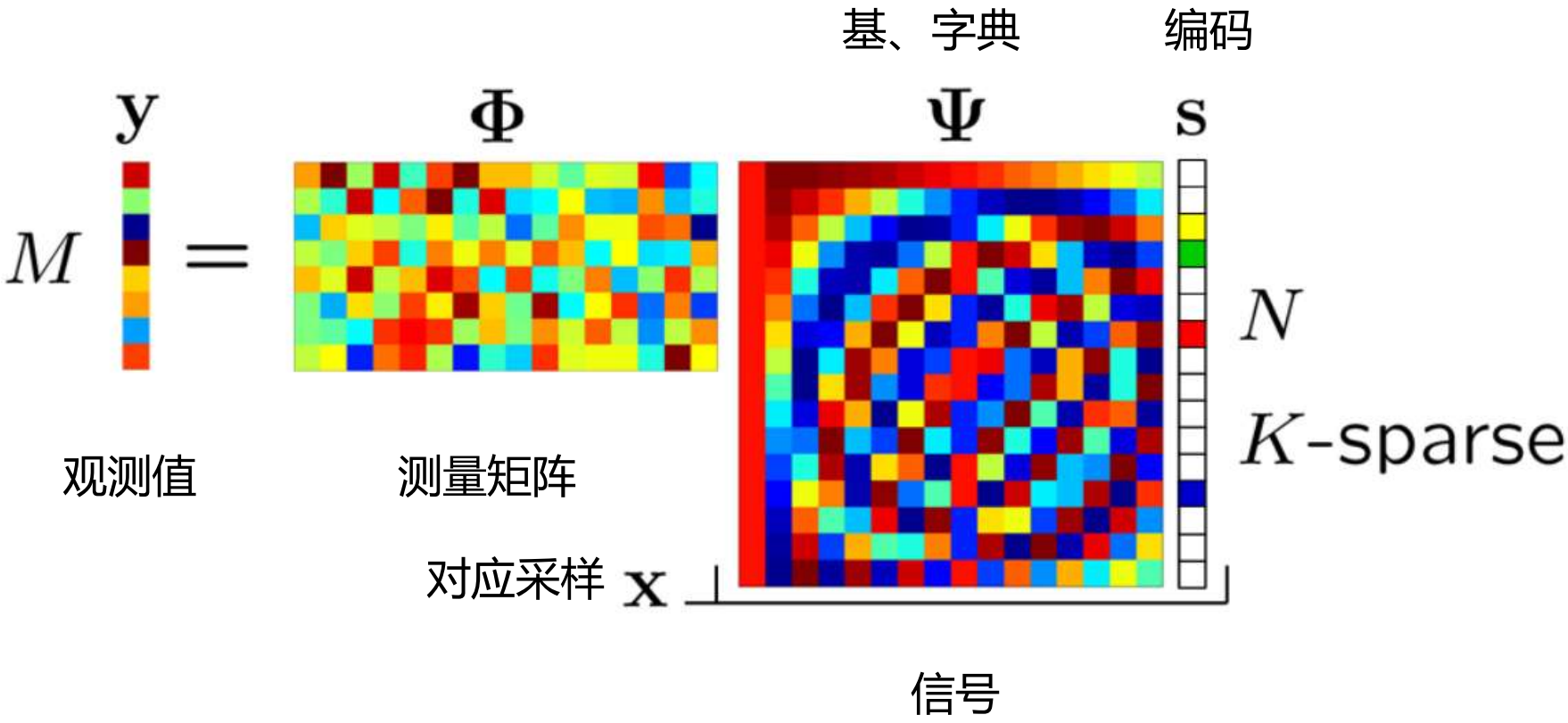
稀疏性（sparsity）

不相关性（incoherence）

压缩感知

一般的自然信号x本身并不是稀疏的，需要在某种稀疏基上进行稀疏表示

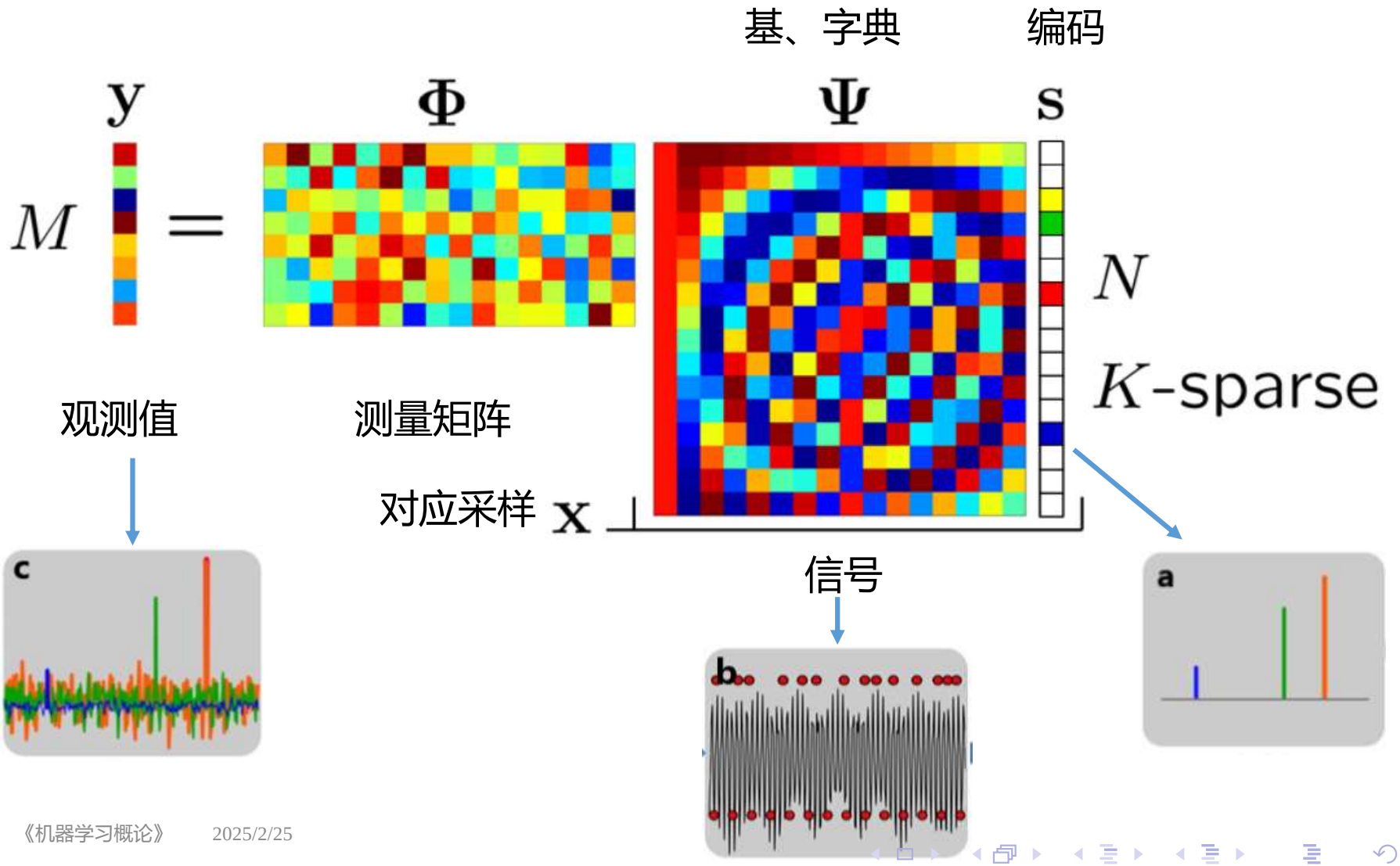
对x在基Ψ上进行稀疏表示，$x = \Psi s$



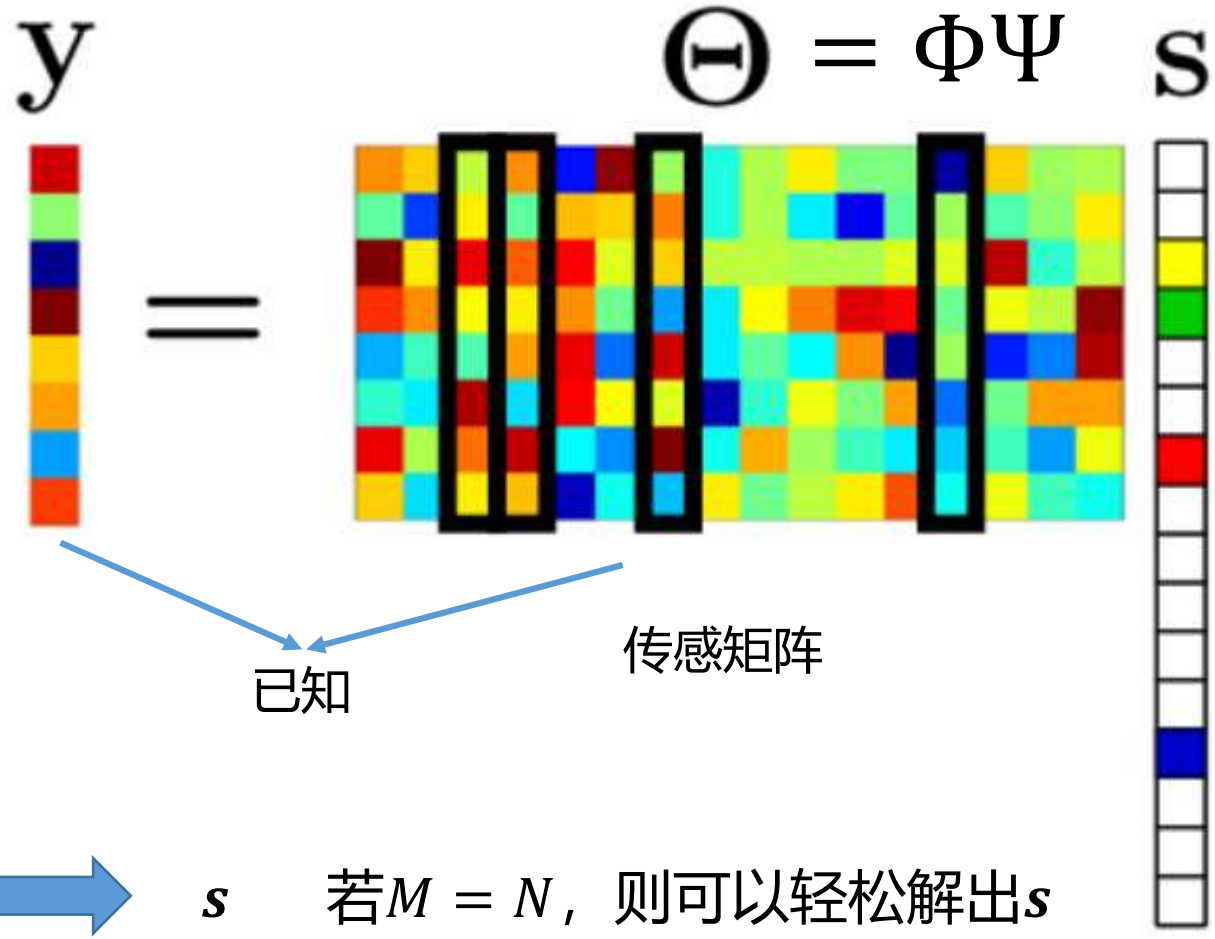
测量矩阵将高维信号投影到低纬空间

压缩感知问题就是在已知测量值y和测量矩阵Φ的基础上，求解欠定方程组$y = \Phi x$得到原信号x

压缩感知



压缩感知



y、Θ ⇒ s

若$M = N$，则可以轻松解出s

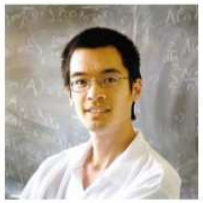
若$M < N$，则根据Θ的RIP特性重构

RIP特性—k-RIP

- 存在某个$\epsilon > 0$，对任意含有k个非零元素的向量v,

满足 $1-\epsilon \leq \frac{\|\boldsymbol{\Theta v}\|_2}{\|\boldsymbol{v}\|_2} \leq 1+\epsilon$,

矩阵$\boldsymbol{\Theta}$必须维持向量$\boldsymbol{v}$的长度只发生小量变化



- Baraniuk证明：RIP的等价条件是测量矩阵和稀疏基的不相关性

Φ 不相关性 Ψ

陶哲轩和Candès又证明：

独立同分布的高斯随机测量矩阵可以成为普适的压缩感知测量矩阵

压缩感知的优化目标和解法

- 若$\mathbf{A}$满足k限定等距性，则可通过下面的优化问题近乎完美地从$\boldsymbol{y}$中恢复出稀疏信号$\boldsymbol{s}$，进而恢复出$\boldsymbol{x}$：

$$\min_{\mathbf{s}} \|\boldsymbol{s}\|_0 \ \ s.t. \boldsymbol{y} = \boldsymbol{\Theta s}$$

- L_0范数最小化是NP难问题。不过，L1范数最小化在一定条件下与L0范数最小化问题共解 [Cándes et al., 2006]：将上式转化为共解的L_1范数最小化问题

$$\min_{\mathbf{s}} \|\boldsymbol{s}\|_1 \ \ s.t. \boldsymbol{y} = \boldsymbol{\Theta s}$$

- 转化为LASSO的等价形式再通过近端梯度下降法求解，即使用“基寻踪去噪”（Basis Pursuit De-Noising）[Chen et al., 1998]

矩阵补全

客户对书籍的喜好程度的评分

	《笑傲江湖》	《万历十五年》	《人间词话》	《云海玉弓缘》	《人类的故事》
赵大	5	?	?	3	2
钱二	?	5	3	?	5
孙三	5	3	?	?	?
李四	3	?	5	4	?

- 能否将表中已经通过读者评价得到的数据当作部分信号，基于压缩感知的思想恢复出完整信号从而进行书籍推荐呢？从题材、作者、装帧等角度看（相似题材的书籍有相似的读者），表中反映的信号是稀疏的，能通过类似压缩感知的思想加以处理。

“矩阵补全”技术解决此类问题

矩阵补全的优化问题和解法

- 矩阵补全（matrix completion）技术的优化形式为

$$\min_{\boldsymbol{X}} \operatorname{rank}(\boldsymbol{X})$$
$$\text{s.t.}\, X_{ij} = A_{ij}, (i,j) \in \Omega$$

- **X**: 需要恢复的稀疏信号
- rank(**X**): **X**的秩
- **A**: 已观测信号
- Ω: **A**中已观测到的元素的位置下标的集合

约束表明，恢复出的矩阵中$\mathbf{X}_{ij}$应当与已观测到的对应元素相同

- NP难问题. 将rank(**X**)转化为其凸包“核范数”（nuclear norm）

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_*$$
$$\text{s.t.}\, X_{ij} = A_{ij}, (i,j) \in \Omega$$

$$\|\boldsymbol{X}\|_* = \sum_{j=1}^{\min\{m,n\}} \sigma_j(\boldsymbol{X})$$

$\sigma_j(\boldsymbol{X})$ 为**X**的奇异值

$\|\boldsymbol{X}\|_*$矩阵的核范数为矩阵的奇异值之和

- 凸优化问题，通过半正定规划求解（SDP，Semi-Definite Programming）

满足一定条件时，只需观察到$O(mr\log^2 m)$个元素就能完美恢复**A**
[Recht, 2011]

作业

- 11.5
- 11.7
- PPT 20页：证明回归和对率回归的损失函数的梯度是否满足L-Lipschitz条件，并求出L

阅读材料

- Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788-791.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507.
- Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, *344*(6191), 1492-1496.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), 484-489.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Chen, Y. (2017). Mastering the game of go without human knowledge. *nature*, *550*(7676), 354-359.
- Zhu, J., Wen, C., Zhu, J., Zhang, H., & Wang, X. (2020). A polynomial algorithm for best-subset selection problem. *Proceedings of the National Academy of Sciences*.

第十三章：半监督学习

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

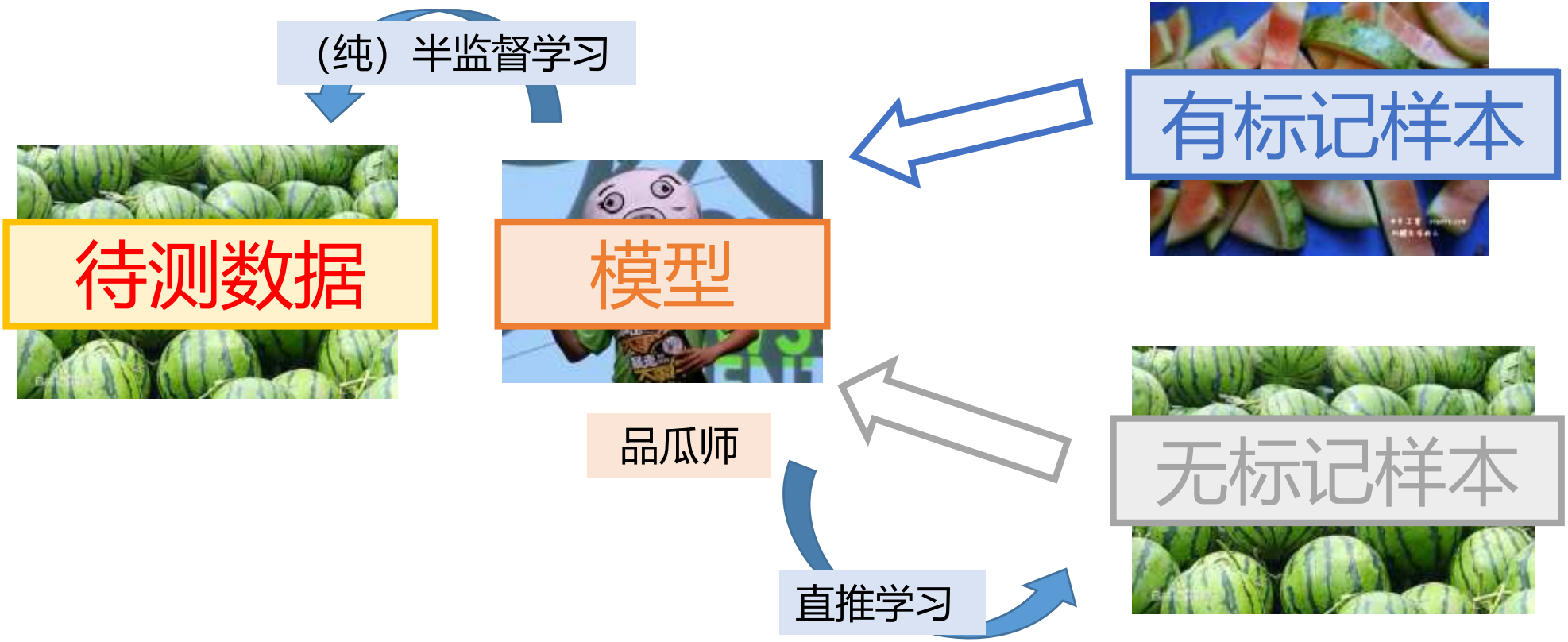
主页：http://staff.ustc.edu.cn/~liandefu

背景（半监督学习）

品瓜师

背景（半监督学习）

训练数据中的未标记样本并非待预测数据

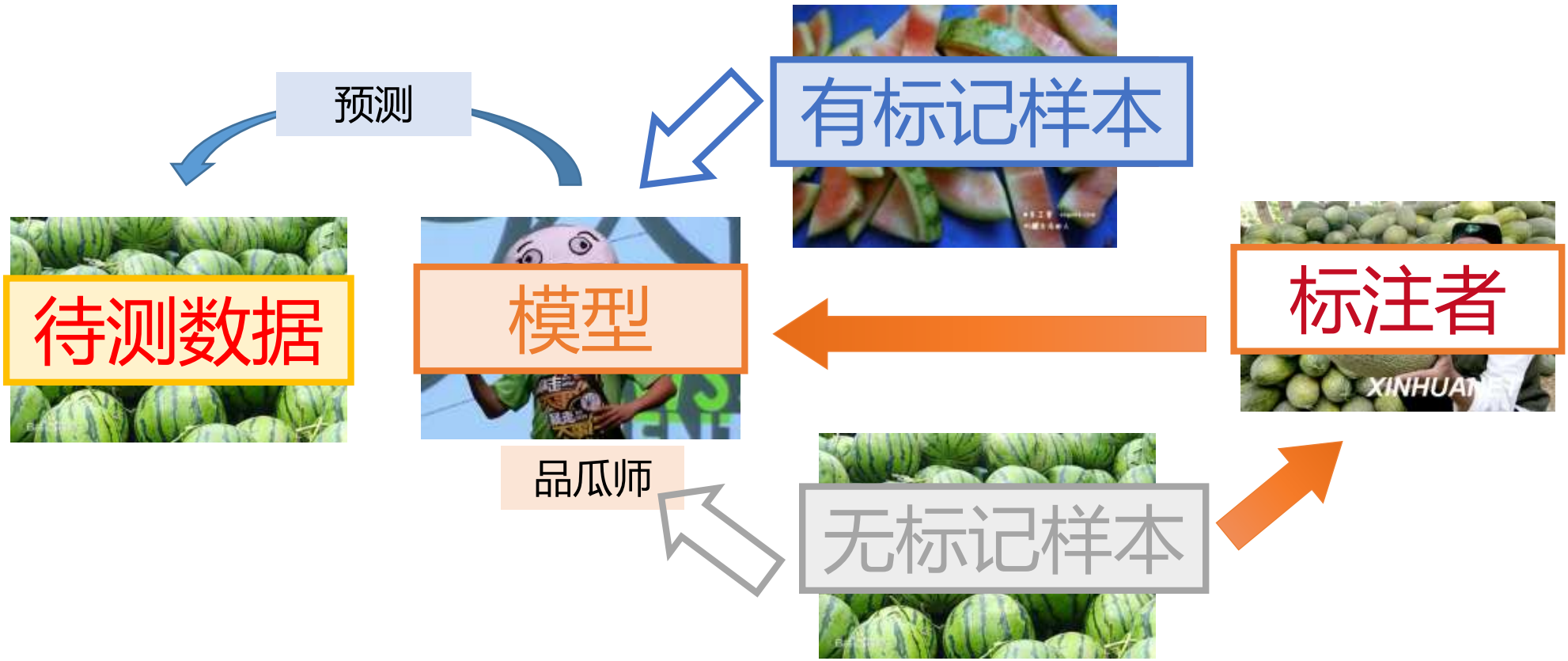


假定学习过程中所考虑的未标记样本恰是待预测数据

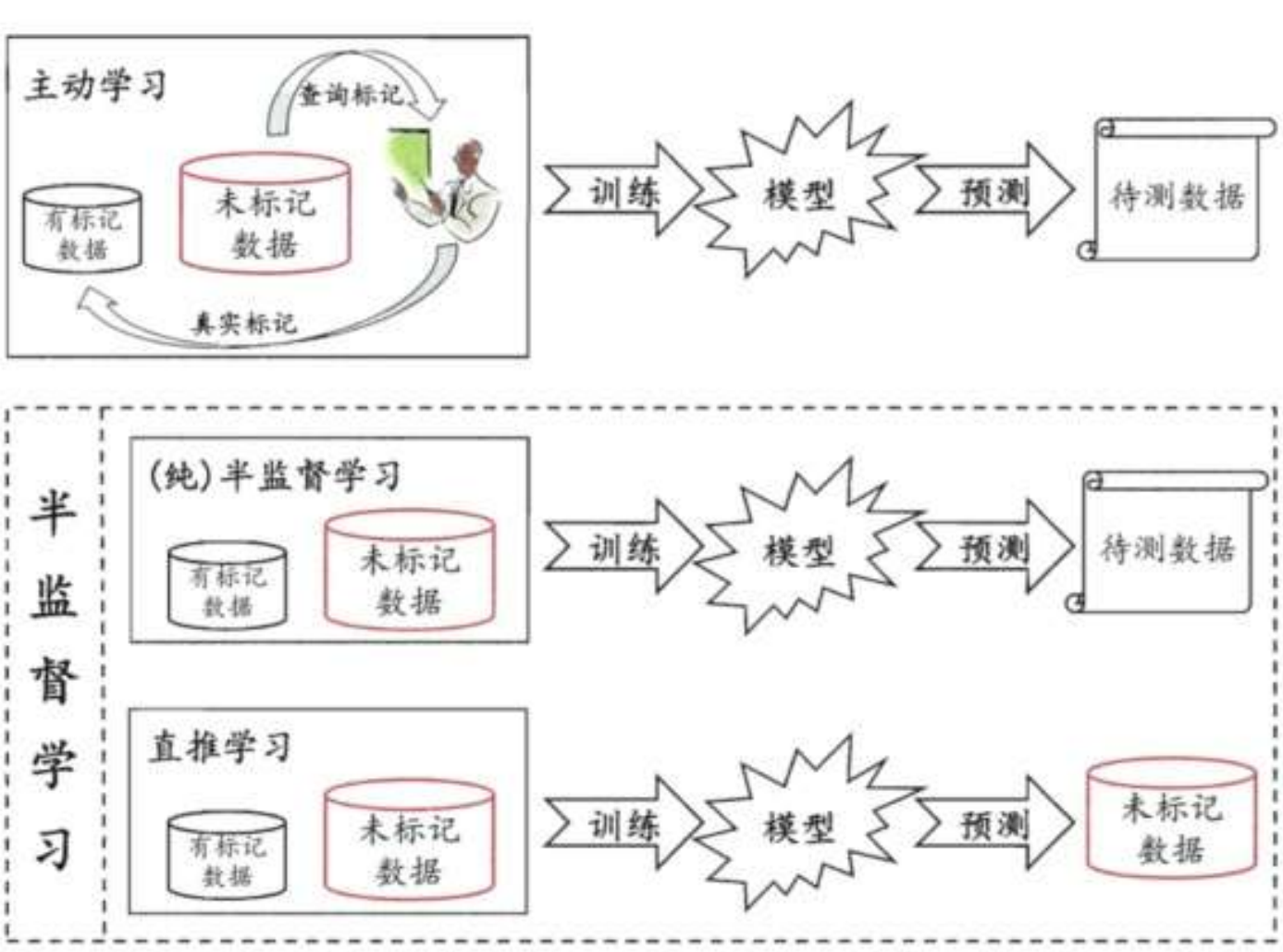
背景（主动学习）

品瓜师

背景（主动学习）



背景（半监督学习）


主动学习
查询标记
有标记数据
未标记数据
真实标记
训练
模型
预测
待测数据
半监督学习
(纯)半监督学习
有标记数据
未标记数据
训练
模型
预测
待测数据
直推学习
有标记数据
未标记数据
训练
模型
预测
未标记数据

未标记样本的效用

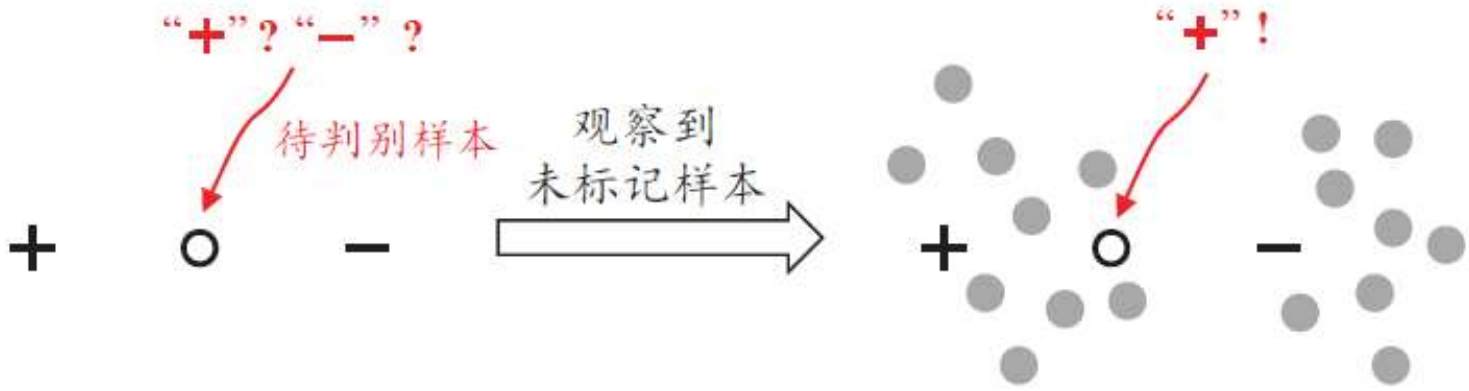


图 13.1　未标记样本效用的例示. 右边的灰色点表示未标记样本

未标记样本的假设

- 要利用未标记样本，必然要做一些将未标记样本所揭示的数据分布信息与类别标记相联系的假设，其中有两种常见的假设。

- 聚类假设（clustering assumption）
  假设数据存在簇结构，同一簇的样本属于同一类别

- 流形假设（manifold assumption）
  假设数据分布在一个流形结构上，邻近的样本具有相似的输出值

流形假设可看做聚类假设的推广

半监督算法

- 生成式方法
- 半监督SVM
- 图半监督学习
- 基于分歧的方法
- 半监督聚类

生成式方法

- 生成式方法是直接基于生成式模型的方法。
- 假设所有数据都是由同一个潜在模型生成的
- 假设样本由混合高斯混合模型生成, 且每个类别对应一个高斯混合成分：

$$p(\boldsymbol{x}) = \sum_{i=1}^{k} \alpha_i p(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

其中 $\alpha_i \geq 0, \sum_{i=1}^{k} \alpha_i = 1$

$$p(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}_i|}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i)\right)$$

生成式方法

- 由最大化后验概率可知：

$$f(\boldsymbol{x}) = \arg\max_{j \in \mathcal{Y}} P(y = j|\boldsymbol{x})$$

$$= \arg\max_{j \in \mathcal{Y}} \sum_{i=1}^{k} P(y = j, \mathrm{z} = i|\boldsymbol{x})$$

$$= \arg\max_{j \in \mathcal{Y}} \sum_{i=1}^{k} P(y = j|\mathrm{z} = i, \boldsymbol{x}) P(\mathrm{z} = i|\boldsymbol{x})$$

$$P(y = j|\mathrm{z} = i) = \begin{cases} 1, \text{if } i = j \\ 0, \text{otherise} \end{cases}$$

不涉及标记

$$P(\mathrm{z} = i|\boldsymbol{x}) = \frac{\alpha_i p(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{i=1}^{k} \alpha_i p(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}$$

生成式方法

$$D_l = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_l, y_l)\}$$

$$D_u = \{\boldsymbol{x}_{1+l}, \boldsymbol{x}_{2+l}, \boldsymbol{x}_{u+l}\}$$

- 假设样本独立同分布，且由同一个高斯混合模型生成，对数似然函数：

$$\ln P(D_l \cup D_u) = \sum_{(\boldsymbol{x}_j, y_j) \in D_l} \ln P(y_j, z_j, \boldsymbol{x}_j) + \sum_{\boldsymbol{x}_j \in D_u} \ln P(z_j, \boldsymbol{x}_j)$$

$$= \sum_{(\boldsymbol{x}_j, y_j) \in D_l} \ln P(z_j, \boldsymbol{x}_j) P(y_j | z_j, \boldsymbol{x}_j) + \sum_{\boldsymbol{x}_j \in D_u} \ln P(z_j, \boldsymbol{x}_j)$$

$$= \sum_{(\boldsymbol{x}_j, y_j) \in D_l} \ln \sum_{i=1}^{K} P(z_j = i)\, P(\boldsymbol{x}_j | z_j = i) P(y_j | z_j = i, \boldsymbol{x}_j) + \sum_{\boldsymbol{x}_j \in D_u} \ln P(z_j, \boldsymbol{x}_j)$$

$$= \sum_{(\boldsymbol{x}_j, y_j) \in D_l} \ln \left(\sum_{i=1}^{k} \alpha_i p(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\, P(y_j | z_j = i) \right)$$

$$+ \sum_{\boldsymbol{x}_j \in D_u} \ln \left(\sum_{i=1}^{k} \alpha_i p(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right)$$

生成式方法

E 基于Θ^t推断隐变量z的分布$p(z \mid x, \Theta^t)$，并计算对数似然$LL(\Theta|x,z)$关于z的期望;

$$\mathcal{L}(p(z|x,\Theta),\Theta) = \mathbb{E}_{z\sim p(z\mid x,\Theta^t)} LL(\Theta|x,z) \quad = Q(\Theta|\Theta^t)$$

根据当前模型参数计算未标记样本属于各高斯混合成分的概率

$$r_{ji} = \frac{\alpha_i p(x_j|\mu_i,\Sigma_i)}{\sum_{i=1}^{k} \alpha_i p(x_j|\mu_i,\Sigma_i)}$$

M 寻找参数最大化期望似然;

$$\Theta^{t+1} = \arg\max_{\Theta} Q(\Theta|\Theta^t)$$

$$\mu_i = \frac{1}{\sum_{x_j\in D_u} r_{ji} + l_i}\left(\sum_{x_j\in D_u} r_{ji}x_j + \sum_{(x_j,y_j)\in D_l \wedge y_j=i} x_j\right) \qquad \alpha_i = \frac{1}{m}\sum_{x_j\in D_u} r_{ji} + l_i$$

$$\Sigma_i = \frac{1}{\sum_{x_j\in D_u} r_{ji} + l_i}\left(\sum_{x_j\in D_u} r_{ji}(x_j-\mu_i)(x_j-\mu_i)^{\mathsf{T}} + \sum_{(x_j,y_j)\in D_l \wedge y_j=i} (x_j-\mu_i)(x_j-\mu_i)^{\mathsf{T}}\right)$$

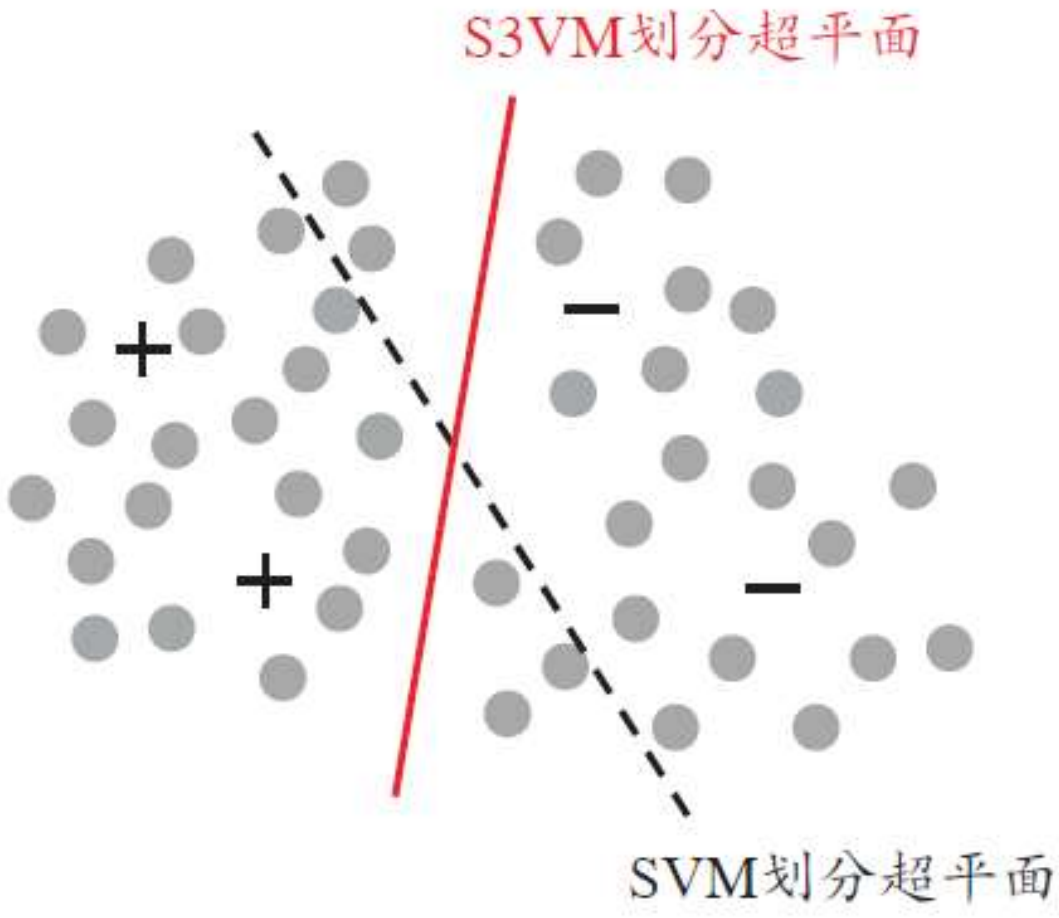
生成式方法

- 将上述过程中的高斯混合模型换成混合专家模型，朴素贝叶斯模型等即可推导出其他的生成式半监督学习算法。

- 此类方法简单、易于实现, 在有标记数据极少的情形下往往比其他方法性能更好。

- 然而, 此类方法有一个关键: 模型假设必须准确, 即假设的生成式模型必须与真实数据分布吻合; 否则利用未标记数据反而会显著降低泛化性能。

半监督SVM



半监督SVM

- 半监督支持向量机中最著名的是TSVM(Transductive Support Vector Machine)

$$\min_{\boldsymbol{w},b,\hat{\boldsymbol{y}},\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C_l \sum_{i=1}^{l} \xi_i + C_u \sum_{i=l+1}^{m} \xi_i$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l,$$

$$\hat{y}_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad i = l+1, \ldots, m,$$

$$\xi_i \geq 0, \quad i = 1, \ldots, m,$$

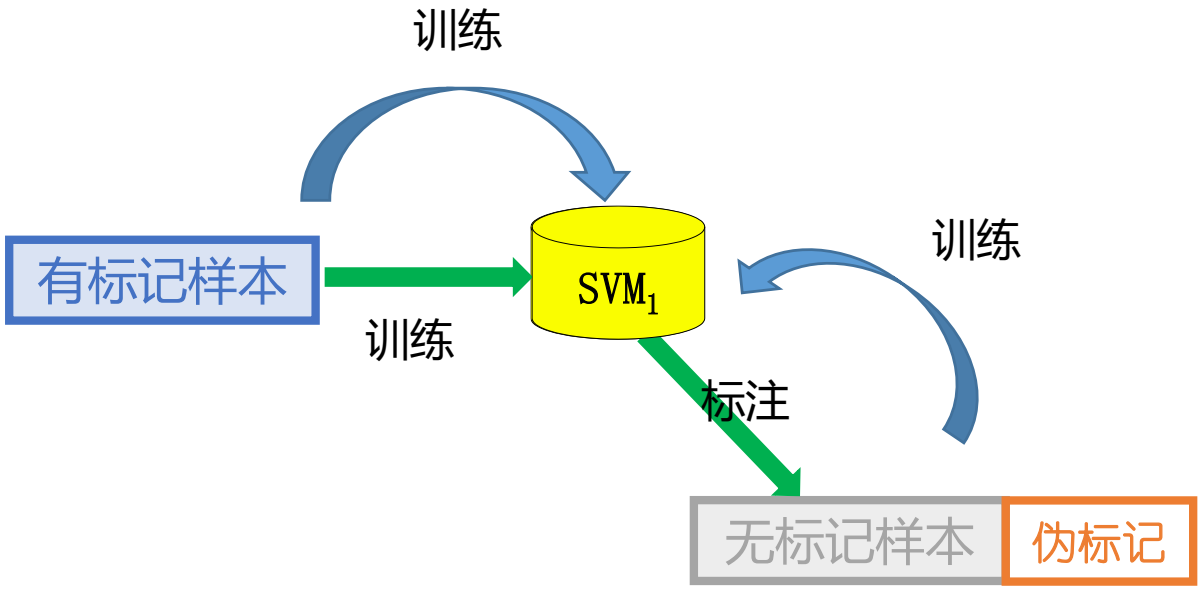
试图考虑对未标记样本进行各种可能的标记指派

即尝试将每个未标记样本分别作为正例或者负例，然后在所有这些结果中，寻求一个在所有样本上间隔最大化的划分超平面

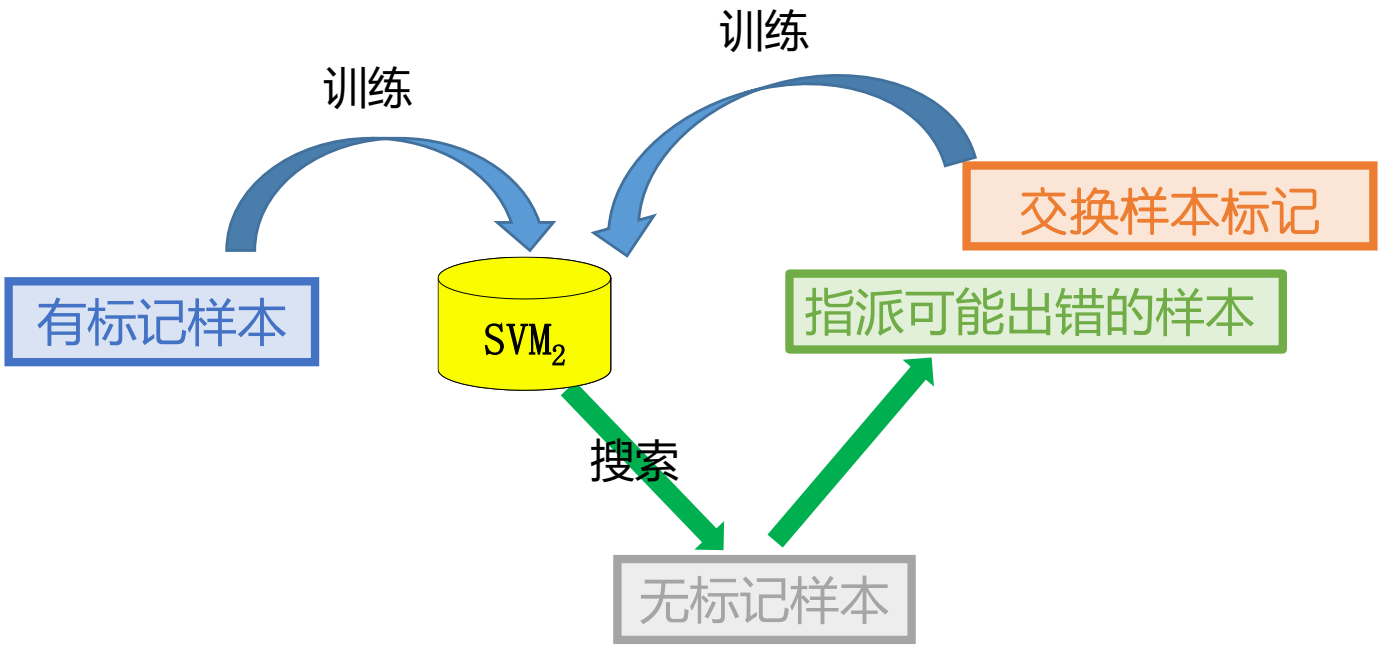
半监督SVM

- 尝试未标记样本的各种标记指派是一个穷举过程，仅当为标记样本很少是才有可能直接求解

- TSVM采用局部搜索来迭代地寻找近似解



半监督SVM



半监督SVM

输入： 有标记样本集 $D_l = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_l, y_l)\}$;
未标记样本集 $D_u = \{\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_{l+u}\}$;
折中参数 C_l, C_u.

过程：

用 D_l 训练一个 SVM_l;
用 SVM_l 对 D_u 中样本进行预测, 得到 $\hat{\boldsymbol{y}} = (\hat{y}_{l+1}, \hat{y}_{l+2}, \ldots, \hat{y}_{l+u})$;
初始化 $C_u \ll C_l$;
while $C_u < C_l$ **do**
  基于 $D_l, D_u, \hat{\boldsymbol{y}}, C_l, C_u$ 求解式(13.9), 得到 $(\boldsymbol{w}, b)$, $\boldsymbol{\xi}$;
  while $\exists\{i, j \mid (\hat{y}_i \hat{y}_j < 0) \wedge (\xi_i > 0) \wedge (\xi_j > 0) \wedge (\xi_i + \xi_j > 2)\}$ **do**
    $\hat{y}_i = -\hat{y}_i$;
    $\hat{y}_j = -\hat{y}_j$;
    基于 $D_l, D_u, \hat{\boldsymbol{y}}, C_l, C_u$ 重新求解式(13.9), 得到 $(\boldsymbol{w}, b)$, $\boldsymbol{\xi}$
  end while
  $C_u = \min\{2C_u, C_l\}$
end while

输出： 未标记样本的预测结果: $\hat{\boldsymbol{y}} = (\hat{y}_{l+1}, \hat{y}_{l+2}, \ldots, \hat{y}_{l+u})$

未标记样本的伪标记不准确

可使得每轮迭代后目标函数值下降

图 13.4 TSVM 算法

半监督SVM

- 未标记样本进行标记指派及调整的过程中, 有可能出现类别不平衡问题,即某类的样本远多于另一类。

- 为了减轻类别不平衡性所造成的不利影响, 可对算法稍加改进:
  将优化目标中的C_u项拆分为C_u^+与C_u^-两项,并在初始化时令:

$$C_u^+ = \frac{u_-}{u_+} C_u^-$$

u_+和 u_-为基于伪标记而当做正、反例使用的未标记样本

半监督SVM

- 显然, 搜寻标记指派可能出错的每一对未标记样本进行调整, 仍是一个涉及巨大计算开销的大规模优化问题。

- 因此, 半监督SVM研究的一个重点是如何设计出高效的优化求解策略

- 例如基于图核(graph kernel)函数梯度下降的Laplacian SVM [Chapelle and Zien, 2005]、基于标记均值估计的meanS3VM[Li et al., 2009]等.

图半监督学习

- 给定一个数据集, 我们可将其映射为一个图, 数据集中每个样本对应于图中一个结点, 若两个样本之间的相似度很高(或相关性很强), 则对应的结点之间存在一条边, 边的“强度”(strength)正比于样本之间的相似度(或相关性)

- 将有标记样本所对应的结点想象为染过色, 而未标记样本所对应的结点则尚未染色。半监督学习就对应于“颜色”在图上扩散或传播的过程

- 由于一个图对应了一个矩阵, 这就使得我们能基于矩阵运算来进行半监督学习算法的推导与分析

图半监督学习

- 先基于$D_l \cup D_u$构建图$G = (V, E)$，其中节点集

$$V = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l, \boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+u}\}$$

- 边集E可表示为一个相似度矩阵，常基于高斯函数定义为

$$w_{ij} = \begin{cases} \exp\dfrac{-\|x_i - x_j\|^2}{2\sigma^2}, \text{if } i \neq j \\ 0, \text{ otherwise} \end{cases}$$

图半监督学习

- 假定从图$G=(V,E)$将学得一个实值函数$f: V \to \mathbb{R}$

- 直观上讲相似的样本应具有相似的标记,于是可定义关于f的“能量函数”(energy function)[Zhu et al., 2003]:

$$E(f)=\frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}w_{ij}\left(f(\boldsymbol{x}_i)-f(\boldsymbol{x}_j)\right)^2$$

$$=\frac{1}{2}\sum_{i=1}^{m}\sum_{i=1}^{m}w_{ij}\left(f(\boldsymbol{x}_i)^2-2f(\boldsymbol{x}_i)f(\boldsymbol{x}_j)+f(\boldsymbol{x}_j)^2\right)$$

$$d_i=\sum_{j=1}^{m}w_{ij} \qquad =\sum_{i=1}^{m}d_i f(\boldsymbol{x}_i)^2-\sum_{i=1}^{m}\sum_{i=1}^{m}w_{ij}f(\boldsymbol{x}_i)f(\boldsymbol{x}_j)$$

$$\boldsymbol{D}=\mathrm{diag}(d_1,d_2,\ldots,d_m) \qquad =\boldsymbol{f}^{\mathsf{T}}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{f}$$

$$\boldsymbol{f}=\left(f(\boldsymbol{x}_1),\ldots,f(\boldsymbol{x}_l),f(\boldsymbol{x}_{l+1}),\ldots,f(\boldsymbol{x}_{l+u})\right)$$

监督学习

- 能量函数$E(f) = \boldsymbol{f}^{\top}(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{f}$
- 能量函数最小化时即得到最优结果
- 具有最小能量的函数f
  - 在有标记样本上满足$f(\boldsymbol{x}_i) = y_i \ (i = 1,2, \ldots, l)$
  - 在未标记样本上满足 $\boldsymbol{\Delta f} = 0$

$\boldsymbol{\Delta} = \boldsymbol{D} - \boldsymbol{W}$ 称为拉普拉斯矩阵

图半监督学习

- 采用分块矩阵表示方式:

$$E(f) = \boldsymbol{f}^{\top}(\boldsymbol{D}-\boldsymbol{W})\boldsymbol{f} = \left(\boldsymbol{f}_l^{\top}\ \boldsymbol{f}_u^{\top}\right)\left(\begin{bmatrix}\boldsymbol{D}_{ll} & 0\\ 0 & \boldsymbol{D}_{uu}\end{bmatrix}-\begin{bmatrix}\boldsymbol{W}_{ll} & \boldsymbol{W}_{lu}\\ \boldsymbol{W}_{ul} & \boldsymbol{W}_{uu}\end{bmatrix}\right)\begin{pmatrix}\boldsymbol{f}_l\\ \boldsymbol{f}_u\end{pmatrix}$$

$$= \boldsymbol{f}_l^{\top}(\boldsymbol{D}_{ll}-\boldsymbol{W}_{ll})\boldsymbol{f}_l - 2\boldsymbol{f}_u^{\top}\boldsymbol{W}_{ul}\boldsymbol{f}_l + \boldsymbol{f}_u^{\top}(\boldsymbol{D}_{uu}-\boldsymbol{W}_{uu})\boldsymbol{f}_u$$

- 令$\frac{\partial E(f)}{\boldsymbol{f}_u} = 0$可得

$$\boldsymbol{f}_u = (\boldsymbol{D}_{uu}-\boldsymbol{W}_{uu})^{-1}\boldsymbol{W}_{ul}\boldsymbol{f}_l$$

图半监督学习

- 若对$\boldsymbol{W}$矩阵归一化 $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}$

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{D}_{ll}^{-1} & 0 \\ 0 & \boldsymbol{D}_{uu}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{W}_{ll} & \boldsymbol{W}_{lu} \\ \boldsymbol{W}_{ul} & \boldsymbol{W}_{uu} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}_{ll}^{-1}\boldsymbol{W}_{ll} & \boldsymbol{D}_{ll}^{-1}\boldsymbol{W}_{lu} \\ \boldsymbol{D}_{uu}^{-1}\boldsymbol{W}_{ul} & \boldsymbol{D}_{uu}^{-1}\boldsymbol{W}_{uu} \end{bmatrix}$$

令$\frac{\partial E(f)}{\boldsymbol{f}_u} = 0$可得

$$\begin{aligned} \boldsymbol{f}_u &= (\boldsymbol{D}_{uu} - \boldsymbol{W}_{uu})^{-1}\boldsymbol{W}_{ul}\boldsymbol{f}_l \\ &= (\boldsymbol{I} - \boldsymbol{D}_{uu}^{-1}\boldsymbol{W}_{uu})^{-1}\boldsymbol{D}_{uu}^{-1}\boldsymbol{W}_{ul}\boldsymbol{f}_l \\ &= (\boldsymbol{I} - \boldsymbol{P}_{uu})^{-1}\boldsymbol{P}_{ul}\boldsymbol{f}_l \end{aligned}$$

图半监督学习

- 上面描述的是一个针对二分类问题的“单步式”标记传播(label propagation)方法, 下面我们来看一个适用于多分类问题的“迭代式”标记传播方法[Zhou et al., 2004].

- 先基于$D_l \cup D_u$构建图$G = (V, E)$，其中节点集 $V = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l, \boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+u}\}$

- 定义一个$(l+u) \times |\mathcal{Y}|$的非负标记矩阵 $\boldsymbol{F} = \left(\boldsymbol{F}_1^\top, \ldots, \boldsymbol{F}_m^\top\right)^\top$，其第$i$行$\boldsymbol{F}_i = \left(F_{i1}, \ldots, F_{i|\mathcal{Y}|}\right)$为示例$\boldsymbol{x}_i$的标记向量, 相应的分类规则为

$$y_i = \underset{1 \le j \le |\mathcal{Y}|}{\operatorname{argmax}} F_{ij}$$

- 将$\boldsymbol{F}$ 初始化为

$$\boldsymbol{F}(0) = \boldsymbol{Y} \qquad Y_{ij} = \begin{cases} 1, \text{if } (1 \le i \le l) \wedge (y_i = j) \\ 0, \text{otherwise} \end{cases}$$

$\boldsymbol{Y}$的前l行 为l个有标记样本的标记向量

图半监督学习

- 基于$\boldsymbol{W}$构造一个标记传播矩阵$\boldsymbol{S} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}$

  其中$\boldsymbol{D}^{-\frac{1}{2}} = \mathrm{diag}\left(\frac{1}{\sqrt{d_1}}, \ldots, \frac{1}{\sqrt{d_m}}\right)$

- 有迭代计算式:

$$\boldsymbol{F}(t+1) = \alpha\, \boldsymbol{S}\, \boldsymbol{F}(t) + (1-\alpha)\, \boldsymbol{Y}$$

- 基于迭代至收敛可得:

$$\boldsymbol{F}^* = \lim_{t\to\infty} \boldsymbol{F}(t) = (1-\alpha)(\boldsymbol{I} - \alpha\boldsymbol{S})^{-1}\boldsymbol{Y}$$

图半监督学习

输入: 有标记样本集 $D_l = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_l, y_l)\}$;
未标记样本集 $D_u = \{\boldsymbol{x}_{l+1}, \boldsymbol{x}_{l+2}, \ldots, \boldsymbol{x}_{l+u}\}$;
构图参数 σ;
折中参数 α.

过程:

基于式(13.11)和参数 σ 得到 $\mathbf{W}$;
基于 $\mathbf{W}$ 构造标记传播矩阵 $\mathbf{S} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$;
根据式(13.18)初始化 $\mathbf{F}(0)$;
$t = 0$;
repeat
  $\mathbf{F}(t+1) = \alpha\mathbf{S}\mathbf{F}(t) + (1-\alpha)\mathbf{Y}$;
  $t = t + 1$
until 迭代收敛至 $\mathbf{F}^*$
for $i = l+1, l+2, \ldots, l+u$ **do**
  $y_i = \arg\max_{1 \le j \le |\mathcal{Y}|} (\mathbf{F}^*)_{ij}$
end for

输出: 未标记样本的预测结果: $\hat{\boldsymbol{y}} = (\hat{y}_{l+1}, \hat{y}_{l+2}, \ldots, \hat{y}_{l+u})$

图 13.5 迭代式标记传播算法

图半监督学习

- 事实上, 算法对应于正则化框架[Zhou et al., 2004]:

$$\min_{\boldsymbol{F}} \frac{1}{2}\left(\sum_{i,j=1}^{m} w_{ij}\left\|\frac{1}{\sqrt{d_i}}\boldsymbol{F}_i - \frac{1}{\sqrt{d_j}}\boldsymbol{F}_j\right\|^2\right) + \mu\sum_{i=1}^{l}\|\boldsymbol{F}_i - \boldsymbol{Y}_i\|^2$$

$$\Rightarrow \min_{\boldsymbol{F}} \operatorname{tr}(\boldsymbol{F}^\top(\boldsymbol{I}-\boldsymbol{S})\boldsymbol{F}) + \mu\|\boldsymbol{F}-\boldsymbol{Y}\|_F^2$$

- 当$\mu = \frac{1-\alpha}{\alpha}$时，最优解恰为迭代算法的收敛解 $\boldsymbol{F}^*$

$$\frac{1}{2}\left(\sum_{i,j=1}^{m}\frac{w_{ij}}{d_i}\|\boldsymbol{F}_i\|^2 + \frac{w_{ij}}{d_j}\|\boldsymbol{F}_j\|^2 - \frac{2w_{ij}}{\sqrt{d_i d_j}}\boldsymbol{F}_i^\top\boldsymbol{F}_j\right)$$

$$= \sum_{i}^{m}\|\boldsymbol{F}_i\|^2 - \sum_{i,j=1}^{m}\frac{w_{ij}}{\sqrt{d_i d_j}}\boldsymbol{F}_i^\top\boldsymbol{F}_j \quad = \|\boldsymbol{F}\|_F^2 - \operatorname{tr}(\boldsymbol{F}^\top\boldsymbol{S}\boldsymbol{F})$$

图半监督学习

- 图半监督学习方法在概念上相当清晰, 且易于通过对所涉矩阵运算的分析来探索算法性质。

- 但此类算法的缺陷也相当明显

  - 首先，存储开销高

  - 其次, 由于构图过程仅能考虑训练样本集, 难以判知新样本在图中的位置, 因此, 在接收到新样本时, 或是将其加入原数据集对图进行重构并重新进行标记传播, 或是需引入额外的预测机制

基于分歧的方法

- 生成式、半监督SVM、图半监督学习等基于单学习器
- 基于分歧的方法(disagreement-based methods)使用多学习器，disagreement亦称diversity
- 学习器之间的“分歧”(disagreement)对未标记数据的利用至关重要
- 协同训练(co-training)[Blum and Mitchell, 1998]是基于分歧的方法的重要代表, 它最初是针对“多视图”(multi-view)数据设计的, 因此也被看作“多视图学习”(multi-view learning)的代表.

基于分歧的方法

文字视图

从目前的情况来看，1月6日华盛顿特区的大游行，充其量只能为特朗普积攒人气，对于改变选举结果没有丝毫意义。

图片视图

作者 | 南风窗常务副主编 谢奕秋

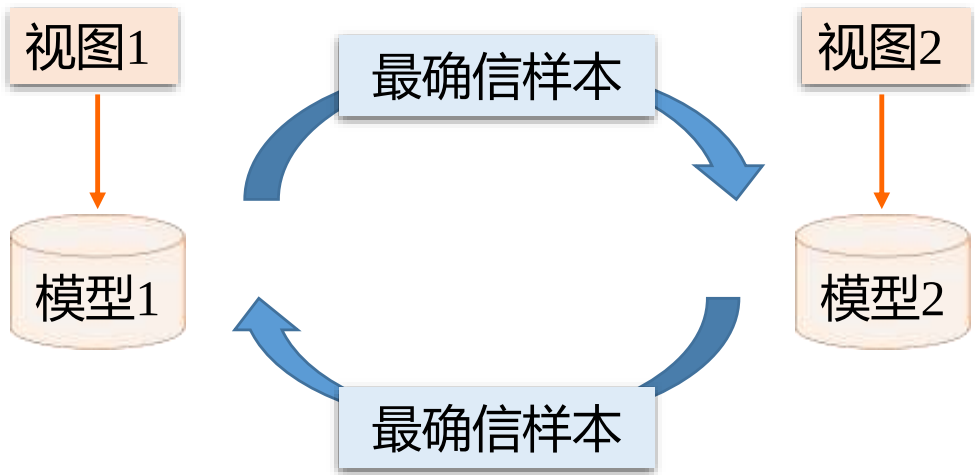
1月6日，美国首都所在地华盛顿特区，会发生大事。

去年12月28日一大早，特朗普发了一条推特，内容是这样的：“1月6日华盛顿特区见，不见不散。且待后续！”12月31日，他提前结束在佛罗里达的度假，回到白宫。

新闻分类任务中的双视图

基于分歧的方法

- 协同训练正是很好地利用了多视图的“相容互补性”。 假设数据拥有两个“充分”(sufficient)且“条件独立”视图。



基于分歧的方法

x_i 的上标仅用于指代两个视图，不表示序关系，即 $\langle x_i^1, x_i^2\rangle$ 与 $\langle x_i^2, x_i^1\rangle$ 表示的是同一个样本.

输入: 有标记样本集 $D_l = \{(\langle x_1^1, x_1^2\rangle, y_1), \ldots, (\langle x_l^1, x_l^2\rangle, y_l)\}$;
未标记样本集 $D_u = \{\langle x_{l+1}^1, x_{l+1}^2\rangle, \ldots, \langle x_{l+u}^1, x_{l+u}^2\rangle\}$;
缓冲池大小 s;
每轮挑选的正例数 p;

在每个视图上基于有标记样本分别训练出一个分类器，然后让每个分类器分别去挑选自己最有把握的未标记样本赋予伪标记

在视图 j 上用有标记样本训练 h_j.

```
for j = 1, 2 do
    h_j ← 𝔏(D_l^j);
    考察 h_j 在 D_s^j = {x_i^j | ⟨x_i^j, x_i^{3-j}⟩ ∈ D_s} 上的分类置信度, 挑选 p 个正例置信度最高的样本 D_p ⊂ D_s、n 个反例置信度最高的样本 D_n ⊂ D_s;
    由 D_p^j 生成伪标记正例 D̃_p^{3-j} = {(x_i^{3-j}, +1) | x_i^j ∈ D_p^j};
    由 D_n^j 生成伪标记反例 D̃_n^{3-j} = {(x_i^{3-j}, −1) | x_i^j ∈ D_n^j};
    D_s = D_s \ (D_p ⋃ D_n);
end for
```

14: if h_1, h_2 均未发生改变 then

扩充有标记数据集.

```
for j = 1, 2 do
    D_l^j = D_l^j ⋃ (D̃_p^j ⋃ D̃_n^j);
end for
```

将伪标记样本提供给另外一个分类器作为新增的有标记样本用于训练更新

输出: 分类器 h_1, h_2

图 13.6 协同训练算法

基于分歧的方法

- 协同训练过程虽简单, 但令人惊讶的是, 理论证明显示出, 若两个视图充分且条件独立, 则可利用未标记样本通过协同训练将弱分类器的泛化性能提升到任意高[Blum and Mitchell, 1998]

- 视图的条件独立性在现实任务中通常很难满足,不会是条件独立的，性能提升幅度不会那么大

- 研究表明, 即使在更弱的条件下,协同训练仍可有效地提升弱分类器的性能[周志华, 2013]

基于分歧的方法

- 协同训练算法本身是为多视图数据而设计的, 但此后出现了一些能在单视图数据上使用的变体算法

- 它们或是使用不同的学习算法[Goldman and Zhou,2000]、或使用不同的数据采样[Zhou and Li, 2005b]、甚至使用不同的参数设置[Zhou and Li, 2005a]来产生不同的学习器, 也能有效地利用未标记数据来提升性能

- 后续理论研究发现, 此类算法事实上无需数据拥有多视图, 仅需弱学习器之间具有显著的分歧(或差异), 即可通过相互提供伪标记样本的方式来提高泛化性能[周志华, 2013]

基于分歧的方法

- 基于分歧的方法只需采用合适的基学习器, 就较少受到模型假设、损失函数非凸性和数据规模问题的影响, 学习方法简单有效、理论基础相对坚实、适用范围较为广泛。

- 为了使用此类方法, 需能生成具有显著分歧、性能尚可的多个学习器, 但当有标记样本很少、尤其是数据不具有多视图时, 要做到这一点并不容易。

半监督聚类

- 聚类是一种典型的无监督学习任务

- 在现实聚类任务中我们往往能获得一些额外的监督信息, 于是可通过“半监督聚类”(semi-supervised clustering)来利用监督信息以获得更好的聚类效果

- 聚类任务中获得的监督信息大致有两种类型:

  - 第一种类型是“必连”(must-link)与“勿连”(cannot-link)约束, 前者是指样本必属于同一个簇, 后者则是指样本必不属于同一个簇;

  - 第二种类型的监督信息则是少量的有标记样本

半监督聚类

- 约束k均值(Constrained k-means)算法[Wagstaff et al., 2001]是利用第一类监督信息的代表。

- 该算法是k均值算法的扩展,它在聚类过程中要确保“必连”关系集合与“勿连”关系集合中的约束得以满足, 否则将返回错误提示。

半监督聚类

输入：样本集 $D = \{x_1, x_2, \ldots, x_m\}$；
必连约束集合 $\mathcal{M}$；
勿连约束集合 $\mathcal{C}$；
聚类簇数 k.

```
while ¬ is_merged do
    基于 K 找出与样本 x_i 距离最近的簇: r = arg min_{j∈K} d_ij ;
    检测将 x_i 划入聚类簇 C_r 是否会违背 M 与 C 中的约束;
    if ¬ is_voilated then
        C_r = C_r ⋃ {x_i};
        is_merged=true
    else
        K = K \ {r};
        if K = ∅ then
            break并返回错误提示
        end if
    end if
end while
```

不冲突，选择最近的簇

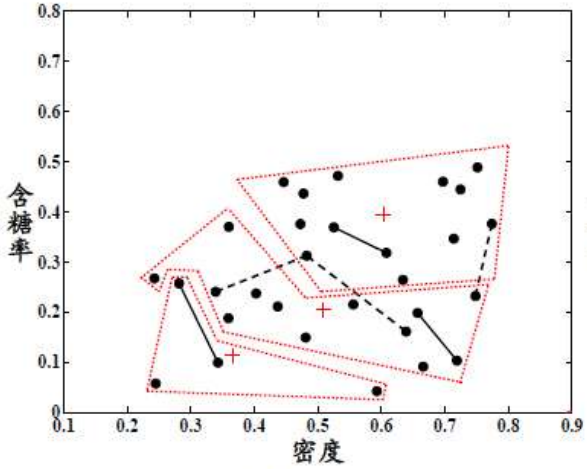
冲突，尝试次近的簇

找不到满足条件的簇，报错

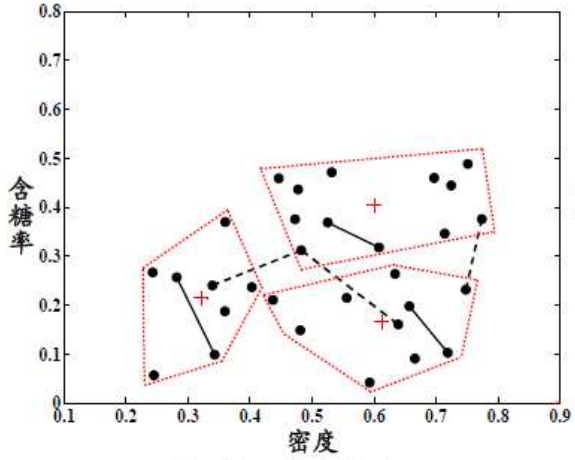
24: end for
25: until 均值向量均未更新
输出：簇划分 $\{C_1, C_2, \ldots, C_k\}$

图 13.7 约束 k 均值算法

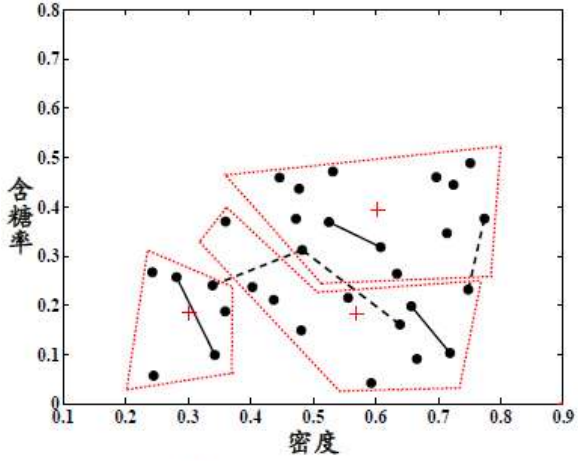
半监督聚类



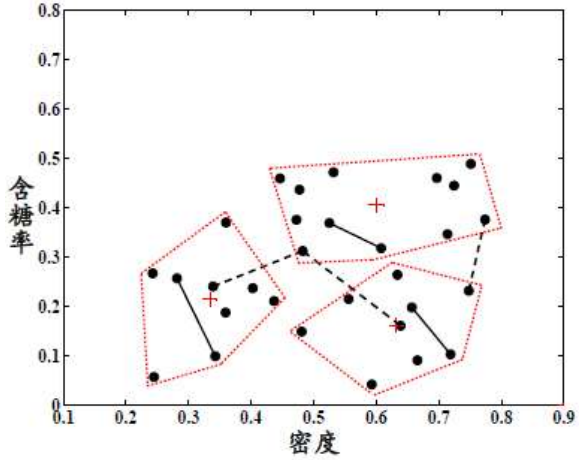
(a) 第 1 轮迭代后



(c) 第 3 轮迭代后



(b) 第 2 轮迭代后



(d) 第 4 轮迭代后

在西瓜数据集4.0上运行结果

半监督聚类

- 第二种监督信息是少量有标记样本。即假设少量有标记样本属于k个聚类簇。

- 将它们作为“种子”, 用它们初始化k均值算法的k个聚类中心, 并且在聚类簇迭代更新过程中不改变种子样本的簇隶属关系

- 这样就得到了约束种子k均值(Constrained Seed k-means)算法[Basu et al., 2002]。

半监督聚类

输入：样本集 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$；

用有标记样本初始化簇中心

```
for j = 1, 2, ..., k do
    μ_j = (1/|S_j|) Σ_{x∈S_j} x
end for
```

用有标记样本初始化k个簇

```
for j = 1, 2, ..., k do
    for all x ∈ S_j do
        C_j = C_j ⋃ {x}
    end for
```

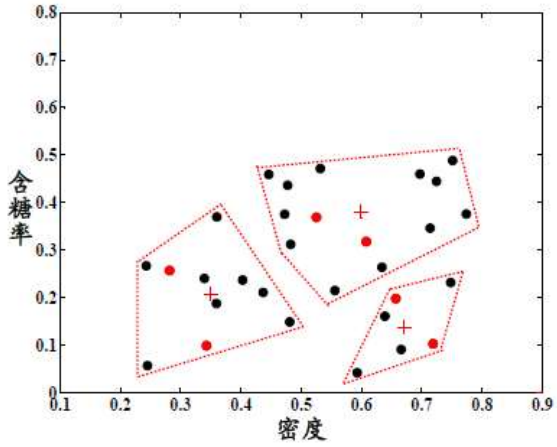
```
        找出与样本 x_i 距离最近的簇：r = arg min_{j∈{1,2,...,k}} d_ij ；
        将样本 x_i 划入相应的簇：C_r = C_r ⋃ {x_i}
    end for
    for j = 1, 2, ..., k do
        μ_j = (1/|C_j|) Σ_{x∈C_j} x;
    end for
until 均值向量均未更新
```

更新均值向量.

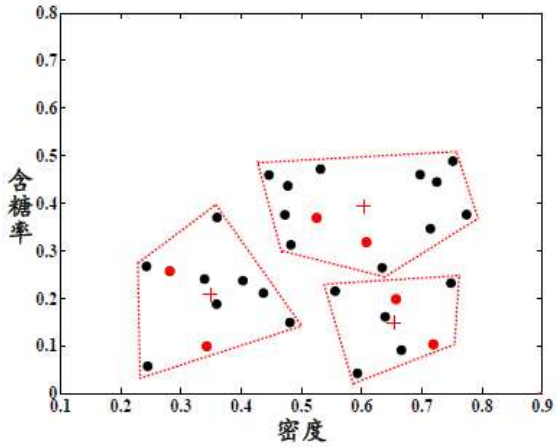
输出：簇划分 $\{C_1, C_2, \ldots, C_k\}$

图 13.9　约束种子 k 均值算法

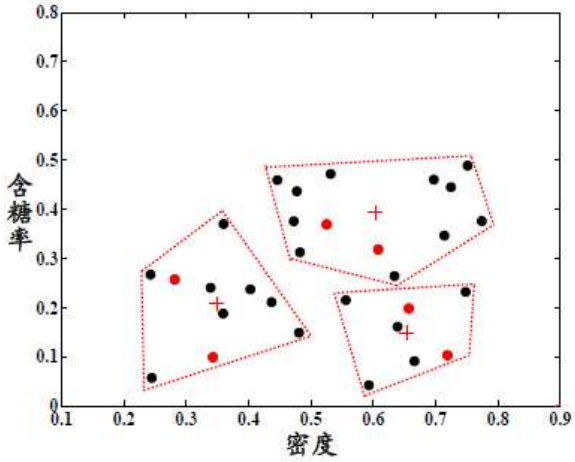
半监督聚类



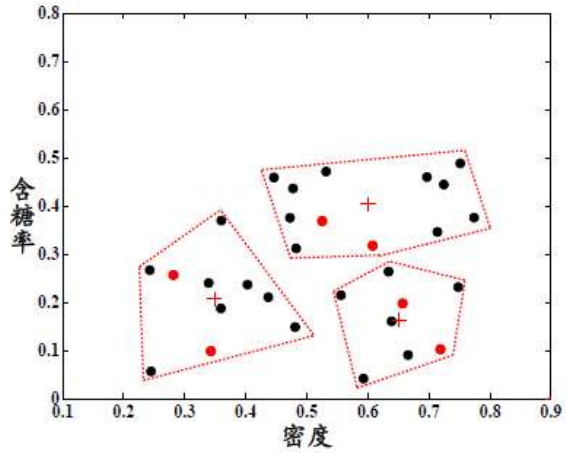
(a) 第 1 轮迭代后



(b) 第 2 轮迭代后



(b) 第 2 轮迭代后



(d) 第 4 轮迭代后

在西瓜数据集4.0上运行结果

第十四章：概率图模型

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

概率模型

- 机器学习最重要的任务是根据**已观察**到的证据（例如训练样本）对感兴趣的**未知**变量（例如类别标记）进行估计和推测。

- **概率模型**（probabilistic model）提供了一种描述框架，将描述任务归结为**计算变量的概率分布**，在概率模型中，利用**已知**的变量推测**未知**变量的分布称为“推断（inference）”，其**核心**在于基于可观测的变量推测出未知变量的**条件分布**
  - 生成式：计算联合分布 $P(Y,R,O)$
  - 判别式：计算条件分布 $P(Y,R|O)$

- 符号约定
  - Y为关心的变量的集合，O为可观测变量集合，R为其他变量集合

概率模型

直接利用概率求和规则消去变量R的时间和空间复杂度为**指数级别**$O(2^{|Y|+|R|})$，需要一种能够简洁紧凑表达变量间关系的工具

概率图模型

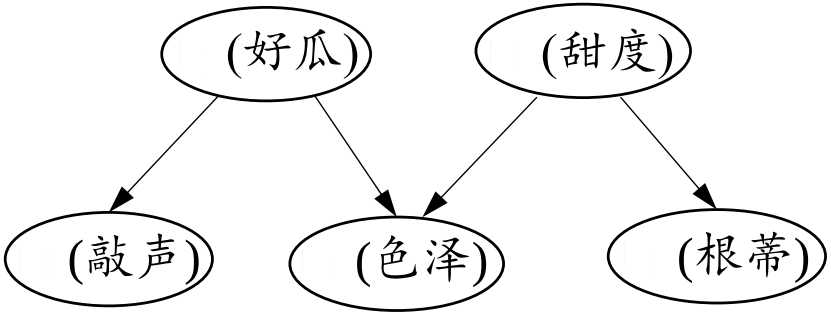
概率图模型

- 概率图模型(probabilistic graphical model)是一类用图来表达**变量相关关系**的概率模型
- 图模型提供了一种**描述框架**，
  - 结点：随机变量（集合）
  - 边：变量之间的依赖关系

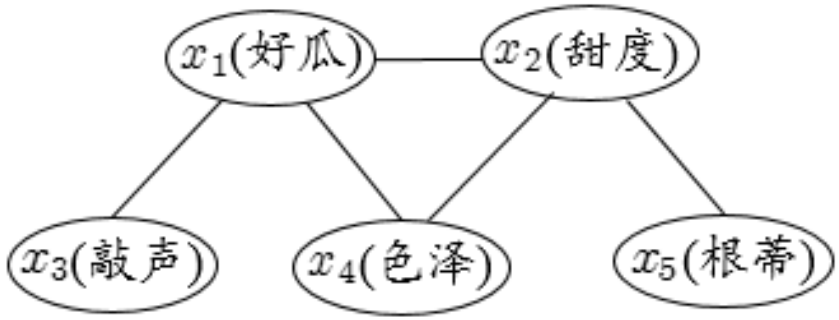
- 分类：
  - **有向图**：贝叶斯网
    - 使用有向无环图表示变量之间的依赖关系
  - **无向图**：马尔可夫网
    - 使用无向图表示变量间的相关关系

概率图模型

- **概率图模型**分类：
  - 有向图：贝叶斯网
  - 无向图：马尔可夫网



有向图



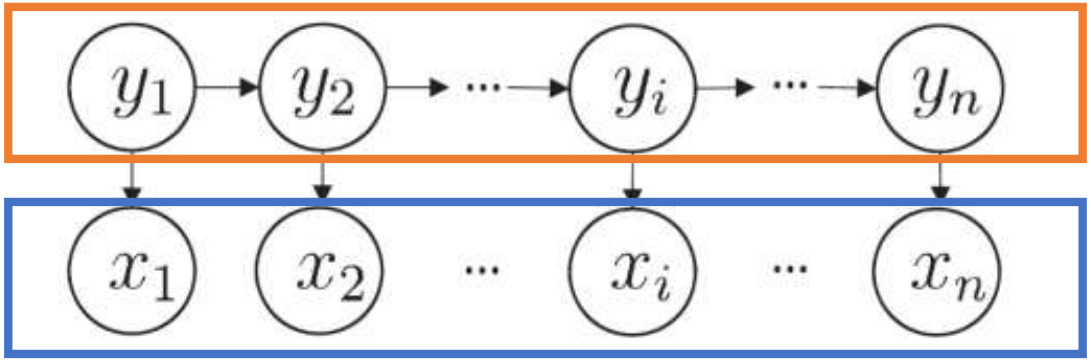
无向图

本章概要

- 隐马尔可夫模型（动态贝叶斯网）
- 马尔可夫随机场 /条件随机场
- 主题模型

隐马尔可夫模型

- 隐马尔可夫模型（Hidden Markov Model，HMM）
- 组成
  - 状态变量: $\{y_1, y_2, \ldots, y_n\}$ 通常假定是**隐藏**的，不可被观测的
    - 取值范围为$\mathcal{Y}$， 通常有N个可能取值的离散空间
  - 观测变量: $\{x_1, x_2, \ldots, x_n\}$ 表示第i时刻的观测值集合
    - 观测变量可以为离散或连续型，本章中只讨论离散型观测变量，取值范围$\mathcal{X}$为$\{o_1, o_2, \ldots, o_M\}$



隐马尔可夫模型

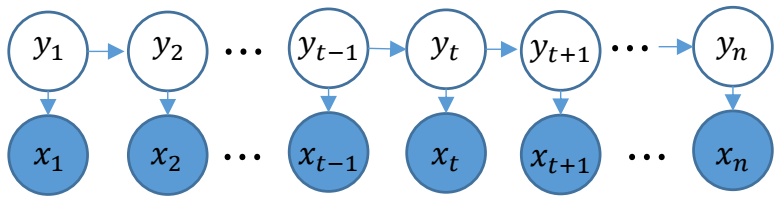
- t时刻的状态x_t仅依赖于$t-1$时刻状态x_{t-1}，与其余$n-2$个状态无关

马尔可夫链：

系统下一时刻状态仅由当前状态决定，不依赖于以往的任何状态

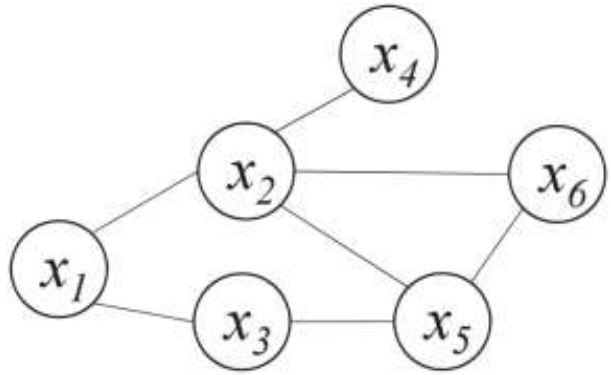
$$P(x_1, y_1, \ldots, x_n, y_n) = P(y_1)P(x_1|y_1)\prod_{t=2}^{n} P(y_t|y_{t-1})P(x_t|y_t)$$

联合概率



马尔可夫随机场

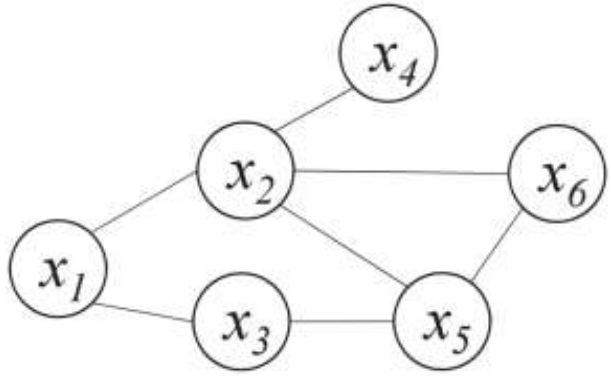
- 马尔可夫随机场（Markov Random Field，MRF）
  - 是典型的马尔可夫网
  - 著名的**无向图**模型



- 图模型表示：
  - 结点表示变量（集），边表示依赖关系
  - 有一组势函数（Potential Functions)，亦称“因子”（factor），这是定义在变量子集上的非负实函数，主要用于定义概率分布函数

马尔可夫随机场—分布形式化：

- 使用基于**极大团**的势函数（因子）
  - 对于图中结点的一个子集，若其中任意两结点间都有边连接，则称该结点子集为一个“团”（clique）。若一个团中加入另外任何一个结点都不再形成团，则称该团为“极大团”（maximal clique)
  - 图中$\{x_1, x_2\}, \{x_2, x_6\}, \{x_2, x_5, x_6\}$等为团
  - 图中$\{x_2, x_6\}$不是极大团
  - 每个结点至少出现在一个极大团中



马尔可夫随机场

- 使用基于**极大团**的势函数（因子）
  - 对于n个变量$\boldsymbol{x} = \{x_1, x_2, \ldots x_n\}$，所有团构成的集合为$\mathcal{C}$，与团$Q \in \mathcal{C}$对应的变量集合记为$\boldsymbol{x}_Q$，则联合概率定义为：

$$P(\boldsymbol{x}) = \frac{1}{Z}\prod_{Q\in\mathcal{C}}\psi_Q(\boldsymbol{x}_Q)$$

  - 其中，ψ_Q是于团Q对应的势函数，Z为概率的规范化因子
  - 在实际应用中，Z往往很难精确计算。但很多任务中，不需要对Z进行精确计算
  - 若变量问题较多，则团的数目过多，上式的乘积项过多，会给计算带来负担，所以需要考虑极大团

马尔可夫随机场

- 联合概率分布可以使用极大团定义，假设所有极大团构成的集合为$\mathcal{C}^*$

$$P(\boldsymbol{x}) = \frac{1}{Z^*}\prod_{Q\in\mathcal{C}^*}\psi_Q(\boldsymbol{x}_Q)$$

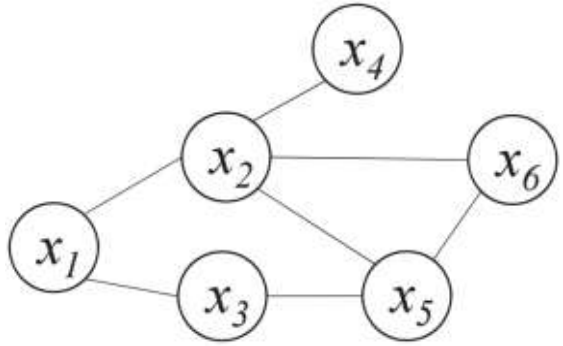
  - 其中，Z^*是规范化因子 $Z^* = \sum_{\boldsymbol{x}}\prod_{Q\in\mathcal{C}^*}\psi_Q(\boldsymbol{x}_Q)$

马尔可夫随机场

- 使用基于**极大团**的势函数（因子）
  - 联合概率分布可以使用极大团定义，假设所有极大团构成的集合为$\mathcal{C}^*$

$$P(\boldsymbol{x}) = \frac{1}{Z^*} \prod_{Q \in \mathcal{C}^*} \psi_Q(\boldsymbol{x}_Q)$$

- 图模型



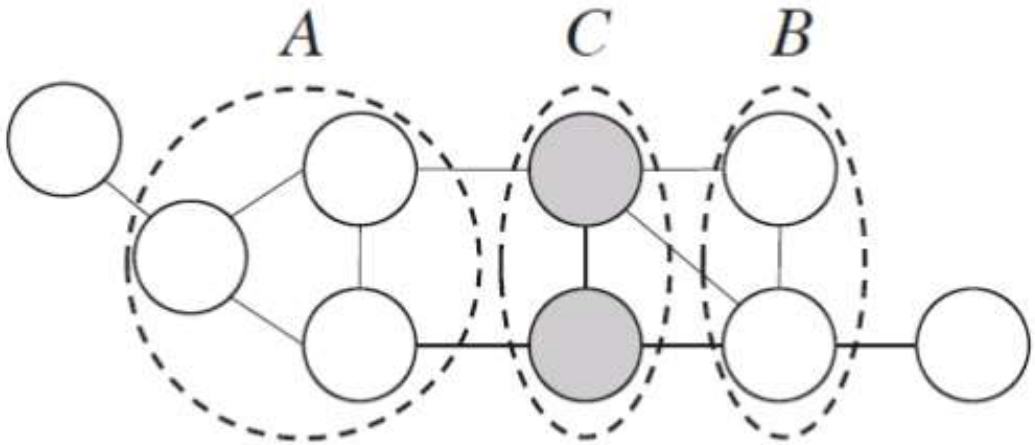
- 联合概率分布

$$P(\boldsymbol{x}) = \frac{1}{Z}\psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{24}(x_2, x_4)\psi_{35}(x_3, x_5)\psi_{256}(x_2, x_5, x_6)$$

马尔可夫随机场中的分离集

- 马尔可夫随机场中得到“**条件独立性**”

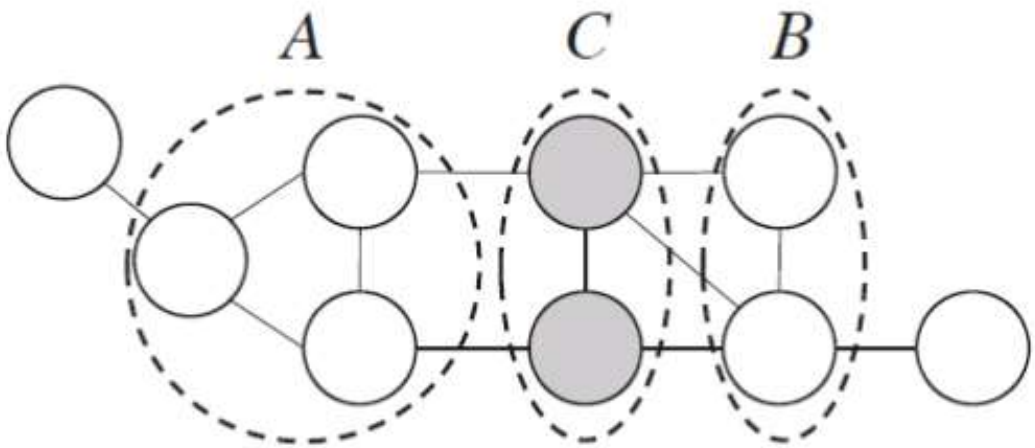
- 借助“**分离**”的概念，若从结点集A中的结点到B中的结点都必须经过结点集C中的结点，则称结点集A，B被结点集C分离，称C为分离集（separating set）



全局马尔可夫性

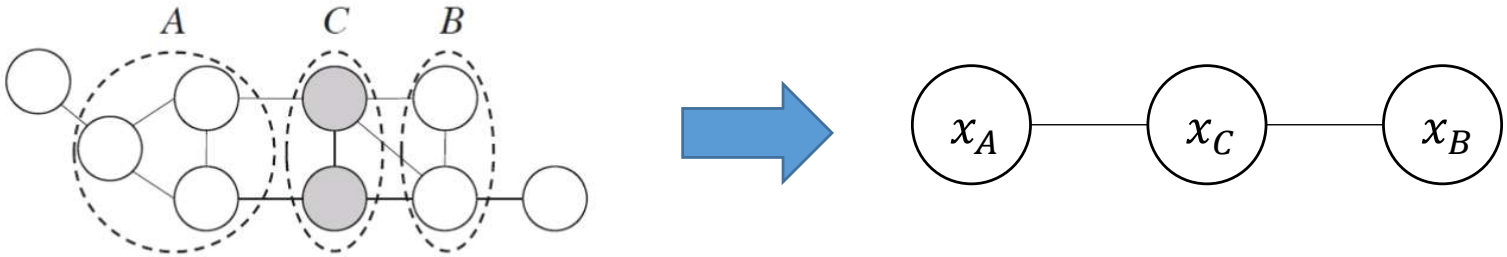
- 借助“分离”的概念，可以得到：
  - **全局**马尔可夫性（global Markov property）：在给定**分离集**的条件下，两个变量子集条件独立

  - 若令A,B,C对应的变量集分别为x_A, x_B, x_C，则x_A和x_B在x_C给定的条件下独立，记为$x_A \perp x_B \mid x_C$



全局马尔可夫性的验证

- 图模型简化：



- 得到图模型的联合概率为：$P(x_A, x_B, x_C) = \frac{1}{Z}\psi_{AC}(x_A, x_C)\psi_{BC}(x_B, x_C)$

条件概率

$$P(x_A, x_B|x_C) = \frac{P(x_A, x_B, x_C)}{\sum_{x_A,x_B} P(x_A, x_B, x_C)}$$

$$= \frac{\frac{1}{Z}\psi_{AC}(x_A, x_C)\psi_{BC}(x_B, x_C)}{\sum_{x_A,x_B} \frac{1}{Z}\psi_{AC}(x_A, x_C)\psi_{BC}(x_B, x_C)}$$

$$= \frac{\psi_{AC}(x_A, x_C)}{\sum_{x_A} \psi_{AC}(x_A, x_C)} \frac{\psi_{BC}(x_B, x_C)}{\sum_{x_B} \psi_{BC}(x_B, x_C)}$$

$$= P(x_A|x_C)P(x_B|x_C)$$

$$P(x_A|x_C) = \frac{\sum_{x_B} P(x_A, x_B, x_C)}{\sum_{x_A,x_B} P(x_A, x_B, x_C)}$$

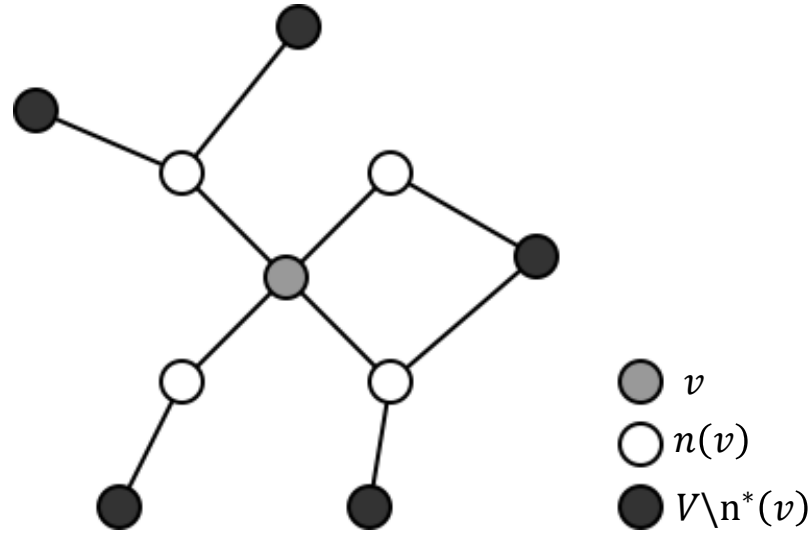
$$= \frac{\sum_{x_B} \frac{1}{Z}\psi_{AC}(x_A, x_C)\psi_{BC}(x_B, x_C)}{\sum_{x_A,x_B} \frac{1}{Z}\psi_{AC}(x_A, x_C)\psi_{BC}(x_B, x_C)}$$

$$= \frac{\psi_{AC}(x_A, x_C)}{\sum_{x_A,x_B} \psi_{AC}(x_A, x_C)}$$

马尔可夫随机场中的条件独立性

全局马尔可夫性（global Markov property）：在给定**分离集**的条件下，两个变量子集条件独立

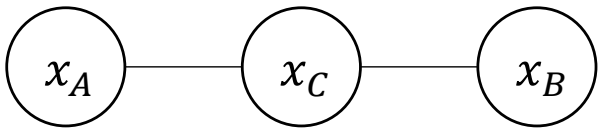
- **局部**马尔可夫性（local Markov property）：在给定**邻接变量**的情况下，一个变量条件独立于其它所有变量
  - 令V为图的结点集，$n(v)$为结点v在图上的邻接节点，$n^*(v) = n(v) \cup \{v\}$, 有$\mathrm{x}_v \perp \mathrm{x}_{V \backslash \mathrm{n}^*(v)} \mid \mathrm{x}_{n(v)}$



- **成对**马尔可夫性（pairwise Markov property）：在给定**所有其它变量**的情况下，两个非邻接变量条件独立
  - 令V为图的结点集，边集为E，对图中的两个结点u, v，若$\langle u, v \rangle \notin E$，有$\mathrm{x}_u \perp \mathrm{x}_v \mid \mathrm{x}_{V \backslash \langle u,v \rangle}$

马尔可夫随机场中的势函数

- 势函数$\psi_Q(\boldsymbol{x}_Q)$的作用是定量刻画变量集$\boldsymbol{x}_Q$中变量的相关关系，应为非负函数，且在所偏好的变量取值上有较大的函数值



- 上图中，假定变量均为二值变量，定义势函数：

$$\psi_{AC}(x_A, x_C) = \begin{cases} 1.5, \text{if } x_A = x_C \\ 0.1, \text{otherwise} \end{cases}$$

$$\psi_{BC}(x_B, x_C) = \begin{cases} 0.2, \text{if } x_B = x_C \\ 1.3, \text{otherwise} \end{cases}$$

模型偏好x_A与x_C有相同的取值
x_A与x_C正相关

模型偏好x_B与x_C有不同的取值
x_B与x_C负相关

令x_A与x_C相同且x_B与x_C不同的变量值指派将有较高的联合概率

马尔可夫随机场中的势函数

- 势函数$\psi_Q(\boldsymbol{x}_Q)$的作用是定量刻画变量集$\boldsymbol{x}_Q$中变量的相关关系，应为非负函数，且在所偏好的变量取值上有较大的函数值

- 为了满足**非负性**，指数函数常被用于定义势函数，即：

$$\psi_Q(\boldsymbol{x}_Q) = e^{-H_Q(x_Q)}$$

- 其中，$H_Q(\boldsymbol{x}_Q)$是一个 定义在变量 $\boldsymbol{x}_Q$ 上的实值函数，常见形式为：

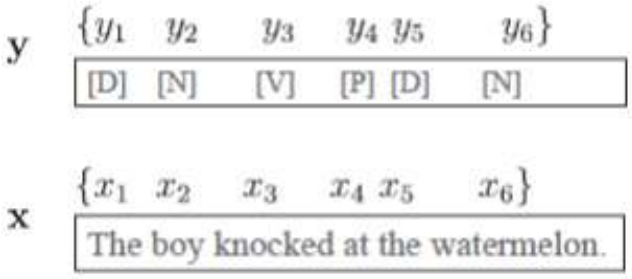
$$H_Q(\boldsymbol{x}_Q) = \sum_{u,v\in Q,u\neq v} \alpha_{uv} x_u\, x_v + \sum_{v\in Q} \beta_v x_v$$

- 其中，α_{uv}和β_v是参数，上式第一项考虑每一对结点的关系，第二项考虑单结点

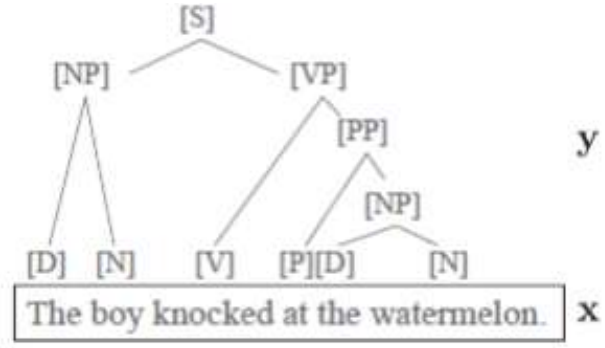
条件随机场

- 条件随机场（Conditional Random Field，CRF)是一种**判别式**无向图模型（可看作给定观测值的MRF），条件随机场对多个变量给定相应**观测值**后的条件概率进行建模
- 若令$\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$为观测序列，$\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$为对应的标记序列，CRF的目标是构建条件概率模型$P(\boldsymbol{y}|\boldsymbol{x})$

- 标记变量 $\boldsymbol{Y}$ 可以是结构型变量，它各个分量之间具有某种相关性。
  - 自然语言处理的词性标注任务中，观测数据为语句（单词序列），标记为相应的词性序列，具有线性序列结构
  - 在语法分析任务中，输出标记是语法树，具有树形结构



(a) 词性标注



(b) 语法分析

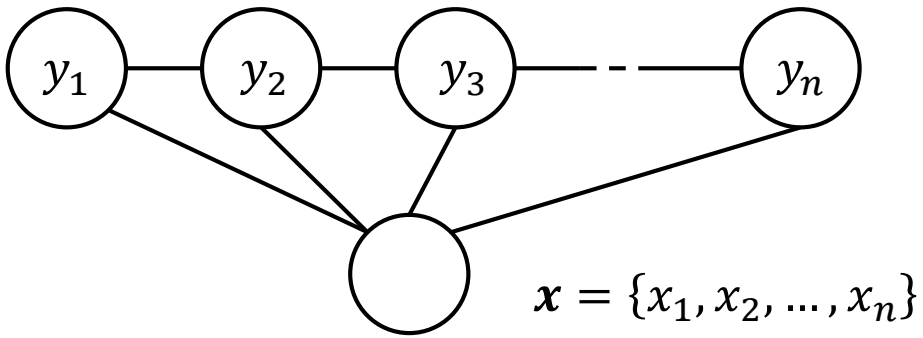
条件随机场

- 令$G = \langle V, E \rangle$表示结点与标记变量$\boldsymbol{y}$中元素一一对应的无向图。无向图中，Y_v表示与节点v对应的标记变量，$n(v)$表示结点v的邻接结点，若图中的每个结点都满足马尔可夫性，

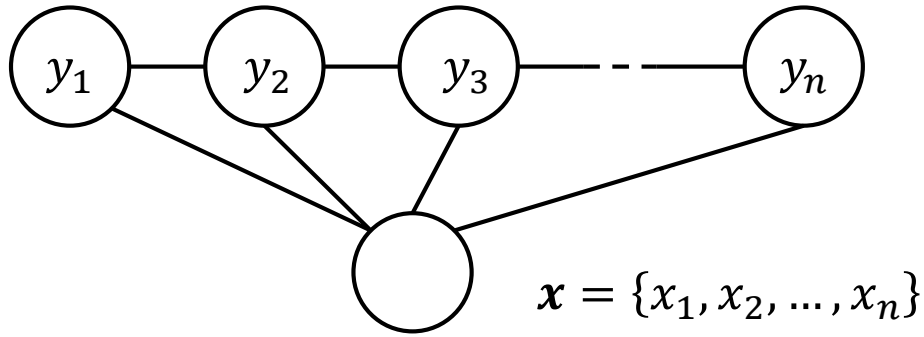
$$P\left(Y_v | X, Y_{V\setminus\{v\}}\right) = P\left(Y_v | X, Y_{n(v)}\right)$$

则(Y, X)构成条件随机场

- CRF使用势函数和图结构上的团来定义$P(\boldsymbol{y}|\boldsymbol{x})$
- 接下来仅考虑**链式**条件随机场（chain-structured CRF）



链式条件随机场



- 包含两种关于标记变量的团：
  - 相邻的标记变量，即 $\{y_{i-1}, y_i\}$；
  - 单个标记变量， $\{y_i\}$；
- 条件概率可被定义为：

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z}\exp\left(\sum_{j=1}^{K_1}\sum_{i=2}^{n}\lambda_j t_j(y_{i-1}, y_i, \boldsymbol{x}, i) + \sum_{l=1}^{K_2}\sum_{i=1}^{n}\mu_l s_l(y_i, \boldsymbol{x}, i)\right)$$

  - $t_j(y_{i+1}, y_i, \boldsymbol{x}, i)$是定义在观测序列的两个相邻标记位置上的转移特征函数（transition feature function），用于刻画相邻标记变量之间的相关关系以及观测序列对它们的影响
  - $s_k(y_i, \boldsymbol{x}, i)$是定义在观测序列的标记位置$i$上的状态特征函数（status feature function），用于刻画观测序列对标记变量的影响
  - λ_j, μ_k为参数，Z为规范化因子

CRF特征函数举例

- **特征函数**通常是实值函数，以刻画数据的一些**很可能成立**或者**期望成立**的经验特性，以词性标注任务为例：

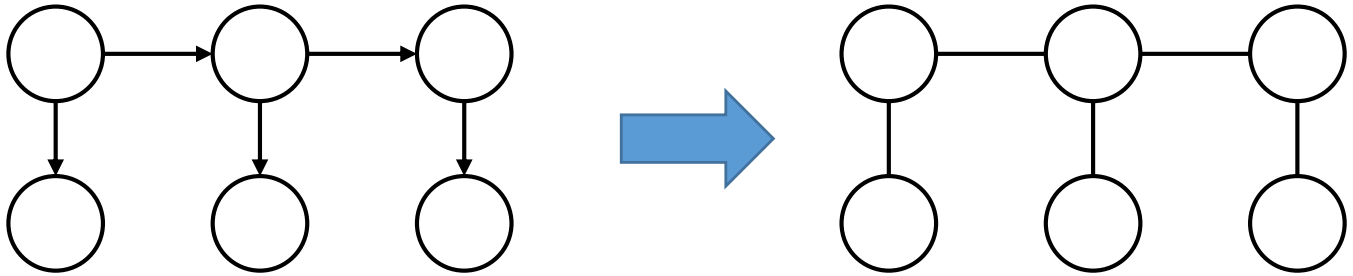
$\mathbf{y}$	$\{y_1$	y_2	y_3	y_4	y_5	$y_6\}$
	[D]	[N]	[V]	[P]	[D]	[N]
$\mathbf{x}$	$\{x_1$	x_2	x_3	x_4	x_5	$x_6\}$
	The	boy	knocked	at	the	watermelon.

- 采用特征函数：

$$t_j(y_{i+1}, y_i, \boldsymbol{x}, i) = \begin{cases} 1, \text{if } y_{i+1} = [P], y_i = [V], \text{and } x_i = \text{"knock"} \\ 0, \text{ otherwise} \end{cases}$$

  - 表示第i个观测值x_i为单词’knock’时，相应的标记y_i, y_{i+1}很可能分别为$[V], [P]$.

HMM to CRF



HMM　　CRF

$$P(\boldsymbol{x},\boldsymbol{y}) = P(y_1)P(x_1|y_1)\prod_{t=2}^{n} P(y_t|y_{t-1})P(x_t|y_t)$$

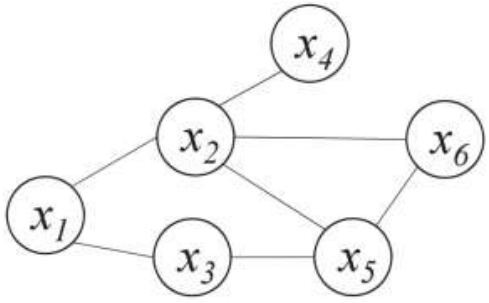
$$= \pi_{y_1} b_{y_1,x_1} a_{y_1,y_2} b_{y_2,x_2} a_{y_2,y_3} \cdots a_{y_{n-1},y_n} b_{y_n,x_n}$$

$$= \exp\left(\sum_{t=1}^{n-1}\sum_{i,j} \mathbb{I}(y_t = s_i, y_{t+1} = s_j)\log a_{ij} + \sum_{i} \mathbb{I}(y_1 = s_i)\log \pi_i \right.$$

$$\left. + \sum_{t=1}^{n}\sum_{j,k} \mathbb{I}(y_t = s_j, x_t = o_k)\log b_{jk}\right)$$

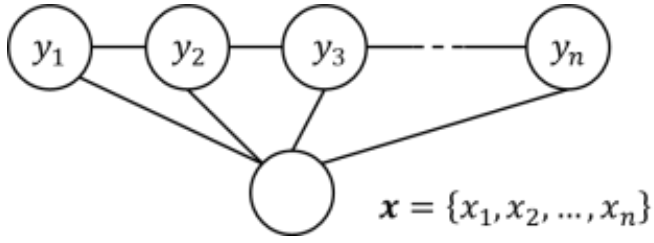
MRF与CRF的对比

- 使用团上的势函数定义概率
- 对**联合概率**建模



$$P(\boldsymbol{x}) = \frac{1}{Z}\psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{24}(x_2, x_4)$$
$$\psi_{35}(x_3, x_5)\psi_{256}(x_2, x_5, x_6)$$

- 使用团上的势函数定义概率
- 有观测变量，对**条件概率**建模



$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z}\exp\left(\sum_{j=1}^{K_1}\sum_{i=2}^{n}\lambda_j t_j(y_{i-1}, y_i, \boldsymbol{x}, i) + \sum_{l=1}^{K_2}\sum_{i=1}^{n}\mu_l s_l(y_i, \boldsymbol{x}, i)\right)$$

CRF基本问题

- **概率计算问题**：评估模型和观测序列间的匹配程度:
  有效计算观测序列产生概率$P(\boldsymbol{x}|\lambda)$

- **预测问题**：根据观测序列“推测”隐藏的模型状态
  $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$

- **学习问题**：如何调整模型参数 $\lambda = [A, B, \pi]$以使得
  该序列出现的概率$P(\boldsymbol{x}|\lambda)$最大

概率计算问题

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z}\exp\left(\sum_{j=1}^{K_1}\sum_{i=2}^{n}\lambda_j t_j(y_{i-1}, y_i, \boldsymbol{x}, i) + \sum_{l=1}^{K_2}\sum_{i=1}^{n}\mu_l s_l(y_i, \boldsymbol{x}, i)\right)$$

假设$y_0 = \text{"start"}$且$y_{n+1} = \text{"end"}$

记 $$f_k(y_{i-1}, y_i, \boldsymbol{x}, i) = \begin{cases} t_k(y_{i-1}, y_i, \boldsymbol{x}, i), & k = 1, \dots, K_1 \\ s_l(y_i, \boldsymbol{x}, i), & l = K_1 + l; l = 1, \dots, K_2 \end{cases}$$

定义 $M_i(\boldsymbol{x}) = [M_i(y_{i-1}, y_i|\boldsymbol{x})] \rightarrow \exp(W_i(y_{i-1}, y_i|\boldsymbol{x})) \rightarrow \sum_{k=1}^{K_1+K_2} w_k f_k(y_{i-1}, y_i, \boldsymbol{x}, i)$

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z}\prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x}) \qquad Z = \sum_{y_1,\dots,y_n}\prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x})$$

概率计算问题

$$Z = \sum_{y_1,\dots,y_n} \prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x}) = \alpha_{n+1}(y_{n+1}|\boldsymbol{x}) = [M_1(\boldsymbol{x})M_2(\boldsymbol{x})\cdots M_{n+1}(\boldsymbol{x})]_{start,stop}$$

$$= \sum_{y_1} M_1(y_0, y_1|\boldsymbol{x}) \sum_{y_2} M_2(y_1, y_2|\boldsymbol{x}) \sum_{y_3} M_3(y_2, y_3|\boldsymbol{x}) \dots \sum_{y_n} M_n(y_{n-1}, y_n|\boldsymbol{x}) M_{n+1}(y_n, y_{n+1}|\boldsymbol{x})$$

前t项求和

$$\alpha_{t+1}(y_{t+1}|\boldsymbol{x}) = \sum_{y_1,\dots,y_t} \prod_{i=1}^{t+1} M_i(y_{i-1}, y_i|\boldsymbol{x})$$

$$= \sum_{y_t} M_{t+1}(y_t, y_{t+1}|\boldsymbol{x}) \left(\sum_{y_1,\dots,y_{t-1}} \prod_{i=1}^{t} M_i(y_{i-1}, y_i|\boldsymbol{x}) \right)$$

$$= \sum_{y_t} M_{t+1}(y_t, y_{t+1}|\boldsymbol{x})\, \alpha_t(y_t|\boldsymbol{x})$$

记 $\alpha_0(y_0|\boldsymbol{x}) = \begin{cases} 1, y_0 = start \\ 0, otherwise \end{cases}$

$$\Rightarrow \boldsymbol{\alpha}_{t+1}(\boldsymbol{x}) = \boldsymbol{\alpha}_t(\boldsymbol{x}) M_{t+1}(\boldsymbol{x}) \Rightarrow \boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_0(\boldsymbol{x}) M_1(\boldsymbol{x}) M_2(\boldsymbol{x}) \cdots M_{n+1}(\boldsymbol{x})$$

概率计算问题

$$\beta_t(y_t|\boldsymbol{x}) = \sum_{y_{t+1},\ldots,y_n} \prod_{i=t+1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x})$$

$$= \sum_{y_{t+1}} M_{t+1}(y_t, y_{t+1}|\boldsymbol{x}) \left(\sum_{y_{t+2},\ldots,y_{n+1}} \prod_{i=t+2}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x}) \right)$$

$$= \sum_{y_{t+1}} M_{t+1}(y_t, y_{t+1}|\boldsymbol{x})\, \beta_{t+1}(y_{t+1}|\boldsymbol{x})$$

$$\Rightarrow \quad \boldsymbol{\beta}_t(\boldsymbol{x}) = M_{t+1}(\boldsymbol{x})\boldsymbol{\beta}_{t+1}(\boldsymbol{x}) \quad \Rightarrow \quad \boldsymbol{\beta}_t(\boldsymbol{x}) = M_{t+1}(\boldsymbol{x}) \cdots M_{n+1}(\boldsymbol{x})\boldsymbol{\beta}_{n+1}(\boldsymbol{x})$$

记 $\beta_{n+1}(y_{n+1}|\boldsymbol{x}) = \begin{cases} 1, y_{n+1} = \text{end} \\ 0, \text{otherwise} \end{cases}$

概率计算问题

$$P(y_t|\boldsymbol{x}) = \sum_{\boldsymbol{y}_{-t}} P(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{y}_{-t}} \frac{1}{Z} \prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x})$$

$$= \sum_{\boldsymbol{y}_{1:t-1}} \sum_{y_{t+1:n}} \frac{1}{Z} \prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x})$$

$$= \frac{1}{Z} \sum_{\boldsymbol{y}_{1:t-1}} \prod_{i=1}^{t} M_i(y_{i-1}, y_i|\boldsymbol{x}) \sum_{\boldsymbol{y}_{t+1:n}} \prod_{i=t+1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x})$$

$$= \frac{1}{Z} \alpha_t(y_t|\boldsymbol{x}) \beta_t(y_t|\boldsymbol{x})$$

概率计算问题

$$P(y_{t-1}, y_t|\boldsymbol{x}) = \sum_{\boldsymbol{y}_{-\{t-1,t\}}} P(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{y}_{-\{t-1,t\}}} \frac{1}{Z} \prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x})$$

$$= \sum_{\boldsymbol{y}_{1:t-2}} \sum_{\boldsymbol{y}_{t+1:n}} \frac{1}{Z} \prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x})$$

$$= \frac{1}{Z} \sum_{\boldsymbol{y}_{1:t-2}} \prod_{i=1}^{t-1} M_i(y_{i-1}, y_i|\boldsymbol{x})\, M_t(y_{t-1}, y_t|\boldsymbol{x}) \sum_{\boldsymbol{y}_{t+1:n}} \prod_{i=t+1}^{n+1} M_i(y_{i-1}, y_i|\boldsymbol{x})$$

$$= \frac{1}{Z} \alpha_{t-1}(y_t|\boldsymbol{x}) M_t(y_{t-1}, y_t|\boldsymbol{x}) \beta_t(y_t|\boldsymbol{x})$$

预测问题（Viterbi Algorithm）

- 作为习题

$$f_k(y_{i-1}, y_i, \boldsymbol{x}, i) = \begin{cases} t_k(y_{i-1}, y_i, \boldsymbol{x}, i), & k = 1, \dots, K_1 \\ s_l(y_i, \boldsymbol{x}, i), & l = K_1 + l; l = 1, \dots, K_2 \end{cases}$$

记 $F_t(y_{t-1}, y_t, \boldsymbol{x}) = \left(f_1(y_{i-1}, y_i, \boldsymbol{x}, i), f_2(y_{i-1}, y_i, \boldsymbol{x}, i), \dots, f_K(y_{i-1}, y_i, \boldsymbol{x}, i)\right)$

$K = K_1 + K_2$

$$\boldsymbol{w} = \left(\lambda_1, \dots, \lambda_{K_1}, \mu_1, \dots, \mu_{K_2}\right)$$

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z} \exp\left(\sum_{j=1}^{K_1}\sum_{i=2}^{n} \lambda_j t_j(y_{i-1}, y_i, \boldsymbol{x}, i) + \sum_{l=1}^{K_2}\sum_{i=1}^{n} \mu_l s_l(y_i, \boldsymbol{x}, i)\right)$$

$$\Rightarrow \quad P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z} \exp\left(\sum_{t=1}^{n} \boldsymbol{w}^{\mathrm{T}} F_t(y_{t-1}, y_t, \boldsymbol{x})\right)$$

学习问题（梯度下降法）

$$f_k(y_{i-1}, y_i, \boldsymbol{x}, i) = \begin{cases} t_k(y_{i-1}, y_i, \boldsymbol{x}, i), & k = 1, \dots, K_1 \\ s_l(y_i, \boldsymbol{x}, i), & l = K_1 + l; l = 1, \dots, K_2 \end{cases} \qquad K = K_1 + K_2$$

$$f_k(\boldsymbol{y}, \boldsymbol{x}) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, \boldsymbol{x}, i) \qquad \boldsymbol{w} = (\lambda_1, \dots, \lambda_{K_1}, \mu_1, \dots, \mu_{K_2})$$

$$P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z} \exp\left(\sum_{j=1}^{K_1} \sum_{i=2}^{n} \lambda_j t_j(y_{i-1}, y_i, \boldsymbol{x}, i) + \sum_{l=1}^{K_2} \sum_{i=1}^{n} \mu_l s_l(y_i, \boldsymbol{x}, i) \right)$$

$$= \frac{1}{Z_w(\boldsymbol{x})} \exp\left(\sum_{k=1}^{K} w_k f_k(\boldsymbol{y}, \boldsymbol{x}) \right) \qquad Z_w(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \exp\left(\sum_{k=1}^{K} w_k f_k(\boldsymbol{y}, \boldsymbol{x}) \right)$$

$$\log P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) = \sum_{k=1}^{K} w_k f_k(\boldsymbol{y}, \boldsymbol{x}) - \log Z_w(\boldsymbol{x})$$

学习问题（梯度下降法）

记 $F(\boldsymbol{y},\boldsymbol{x})=\left(f_1(\boldsymbol{y},\boldsymbol{x}),f_2(\boldsymbol{y},\boldsymbol{x}),\ldots,f_K(\boldsymbol{y},\boldsymbol{x})\right)$

$$f_k(\boldsymbol{y},\boldsymbol{x})=\sum_{i=1}^{n} f_k(y_{i-1},y_i,\boldsymbol{x},i)$$

$$\log P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})=\sum_{k=1}^{K} w_k\, f_k(\boldsymbol{y},\boldsymbol{x})-\log Z_w(\boldsymbol{x})$$

$$\nabla_{\boldsymbol{w}}\log P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{w})=F(\boldsymbol{y},\boldsymbol{x})-\nabla_w\log Z_w(\boldsymbol{x})=F(\boldsymbol{y},\boldsymbol{x})-\mathbb{E}_{\tilde{\boldsymbol{y}}\sim P(\tilde{\boldsymbol{y}}|\boldsymbol{x},\boldsymbol{w})}F(\tilde{\boldsymbol{y}},\boldsymbol{x})$$

$$Z_w(\boldsymbol{x})=\sum_{\boldsymbol{y}}\exp\left(\sum_{k=1}^{K} w_k\, f_k(\boldsymbol{y},\boldsymbol{x})\right)$$

$$=\frac{1}{Z_w(\boldsymbol{x})}\nabla_w Z_w(\boldsymbol{x})$$

$$=\frac{1}{Z_w(\boldsymbol{x})}\sum_{\boldsymbol{y}}\exp\left(\sum_{k=1}^{K} w_k\, f_k(\boldsymbol{y},\boldsymbol{x})\right)F(\boldsymbol{y},\boldsymbol{x})$$

$$=\sum_{\tilde{\boldsymbol{y}}} P(\tilde{\boldsymbol{y}}|\boldsymbol{x},\boldsymbol{w})\,F(\tilde{\boldsymbol{y}},\boldsymbol{x})=\mathbb{E}_{\tilde{\boldsymbol{y}}\sim P(\tilde{\boldsymbol{y}}|\boldsymbol{x},\boldsymbol{w})}F(\tilde{\boldsymbol{y}},\boldsymbol{x})$$

学习问题（梯度下降法）

Repeat

从数据集$D = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$随机采样一个样本对

计算梯度信息$\nabla_{\boldsymbol{w}} \log P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) = F(\boldsymbol{y}, \boldsymbol{x}) - \mathbb{E}_{\tilde{\boldsymbol{y}} \sim P(\tilde{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{w})} F(\tilde{\boldsymbol{y}}, \boldsymbol{x})$

权重更新 $w^{t+1} = w^t + \nabla_{\boldsymbol{w}} \log P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w})$

Until 收敛

计算代价高，如何提升计算效率呢?

主题模型

- 主题模型（topic model）是一类生成式**有向图模型**，主要用来处理离散型的数据集合（如文本集合）
- 有效利用海量数据发现文档集合中**隐含的语义**
- 隐狄里克雷分配模型（Latent Dirichlet Allocation, **LDA**）是话题模型的典型代表

隐狄里克雷分配 LDA

- LDA的基本单元
  - **词（word）**：待处理数据中的基本离散单元
  - **文档（document）**：待处理的数据对象，由词组成，词在文档中**不计顺序**。数据对象只要能用“词袋”（bag-of-words）表示就可以使用话题模型
  - **话题（topic）**：表示一个概念，具体表示为一系列相关的词，以及它们在该概念下出现的概率

The MNIST database of handwritten digits, a test set of 10,000 examples. It is a su normalized and centered in a fixed-size i

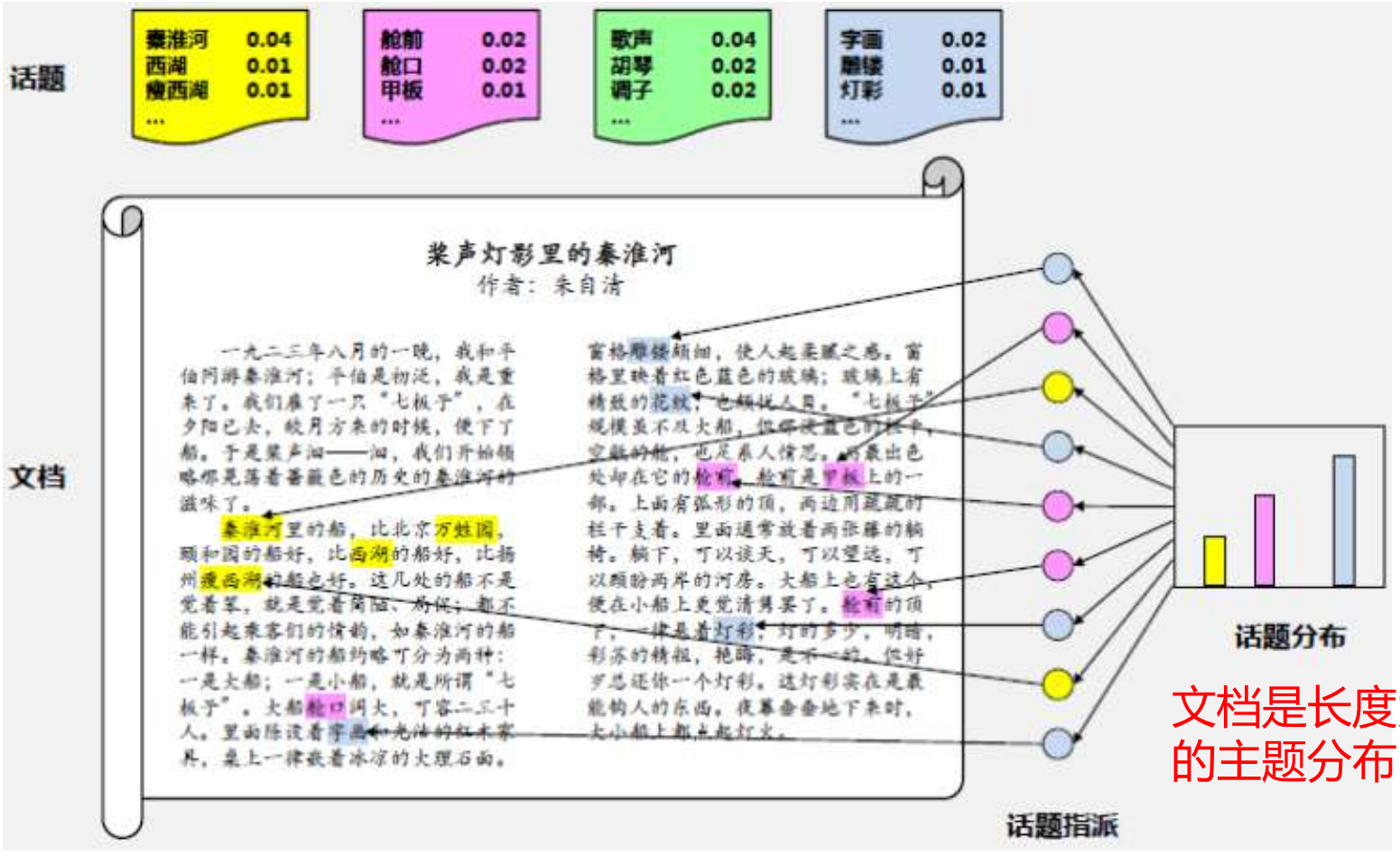
数据　计算机　生物　新闻

建筑　植物　天空

隐狄里克雷分配 LDA

设文档中的词来自一个包含V个词的字典

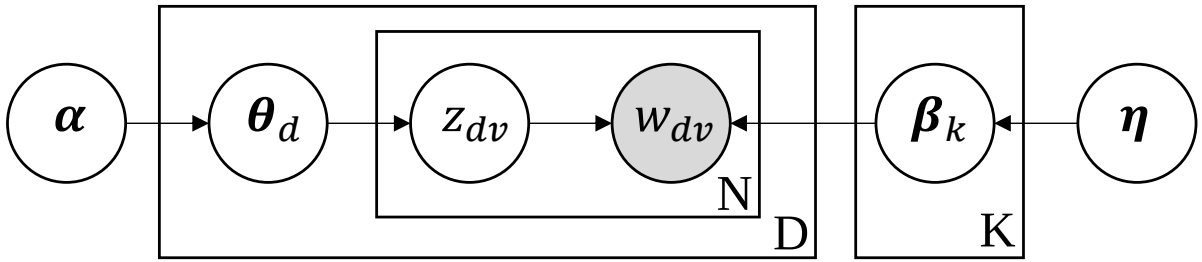
主题是V维概率词向量



隐狄里克雷分配 LDA

- 假定数据集中共含K个话题和D篇文档，词来自含V个词的字典
- 观测数据：D篇文档，每篇文档用长度为V的词频向量表示
  - $\mathcal{D} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \dots, \boldsymbol{w}_D\}$ 其中$\boldsymbol{w}_d = (w_{d1}, \dots, w_{dN_d})$为单词序列
  - 隐变量：K个话题$Z = \{z_1, \dots, z_K\}$，每个话题用长度为N的概率词向量表示
  - 第k话题 $\beta_k \in [0,1]^V$ $\beta_{kv} = P(w_v|z_k)$表示在第k个话题中单词w_v的概率
- 文档表示为话题的分布，由参数$\boldsymbol{\Theta}$ 确定
  - $\boldsymbol{\theta}_t \in [0,1]^K$ $\theta_{mk} = P(z_k|\boldsymbol{w}_m)$

隐狄里克雷分配 LDA

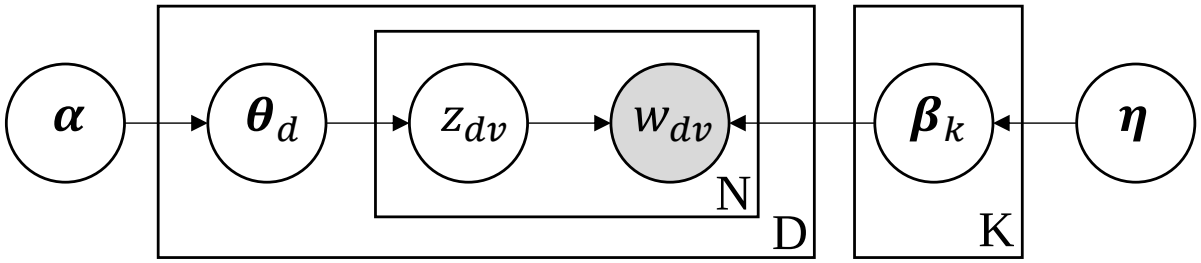


生成文档d过程

- 从以$\boldsymbol{\alpha}$为参数的狄利克雷分布中随机采样一个话题分布$\boldsymbol{\theta}_d$；
- 按如下步骤产生文档中的N_d个词
  - 根据$\boldsymbol{\theta}_d$进行话题指派，得到文档d中词v的话题$z_{d,v}$；
  - 根据指派的话题$z_{d,v}$所对应的的词分布β_k随机采样生成词w_{dv}

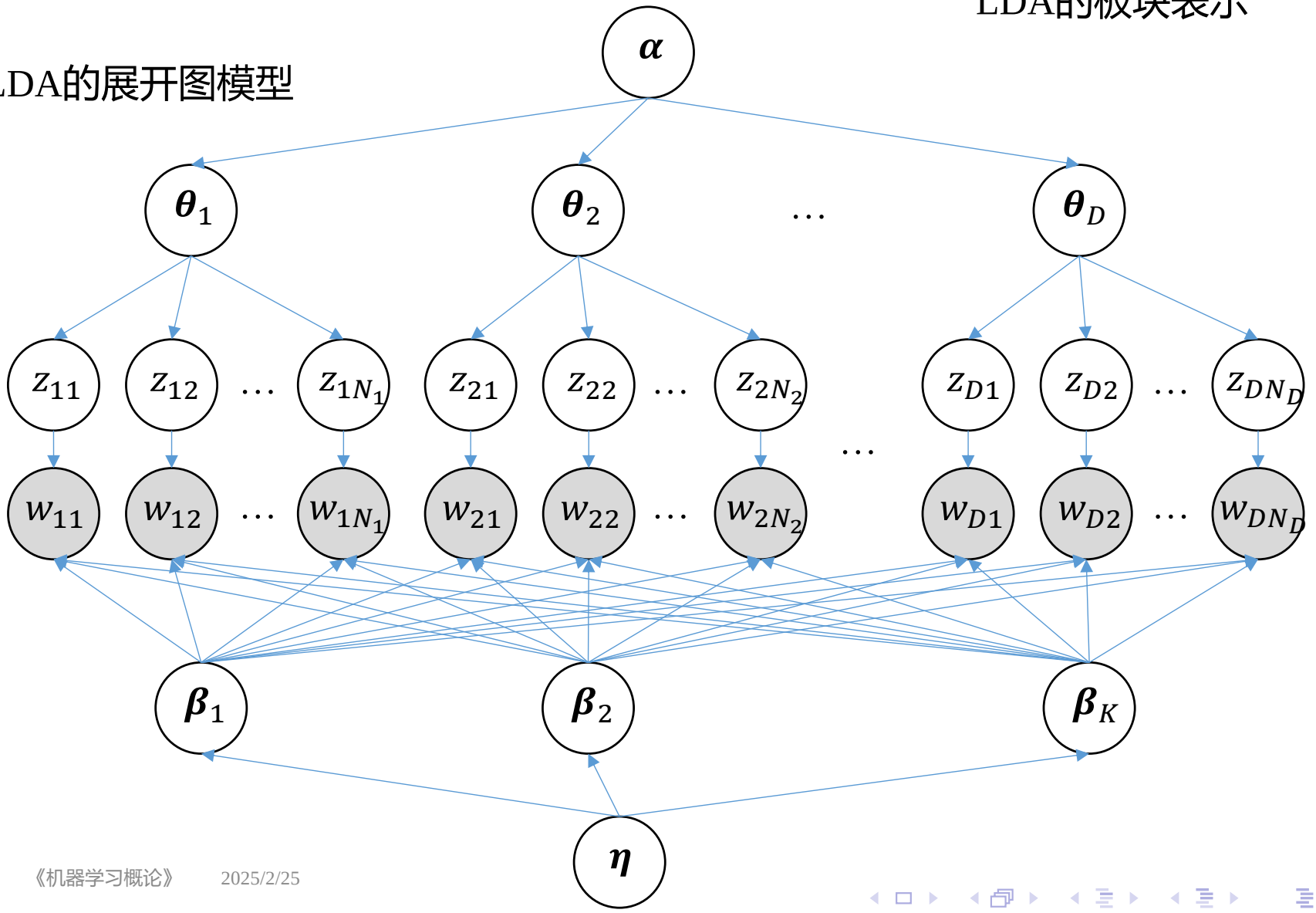
生成主题k过程

- 从以$\boldsymbol{\eta}$为参数的狄利克雷分布中随机采样一个话题分布$\boldsymbol{\beta}_k$；



LDA的板块表示

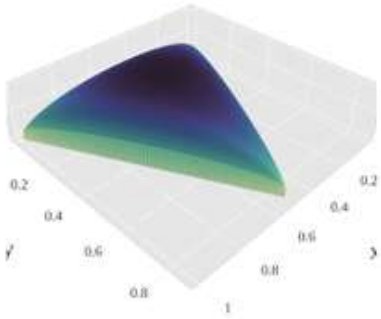
LDA的展开图模型



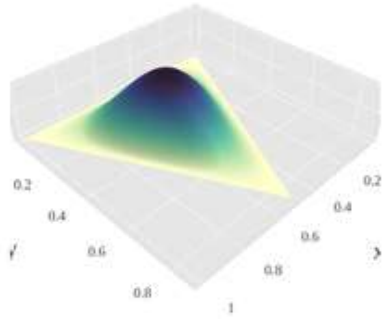
狄里克雷分布

$$\frac{1}{B(\boldsymbol{\alpha})}$$

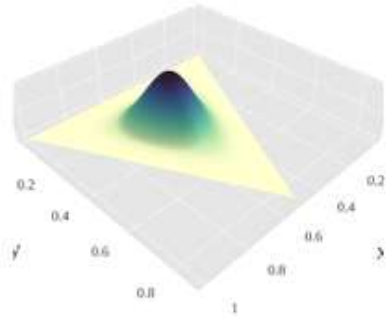
$$f(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_k^{\alpha_k-1} \qquad \sum_k \theta_k = 1$$



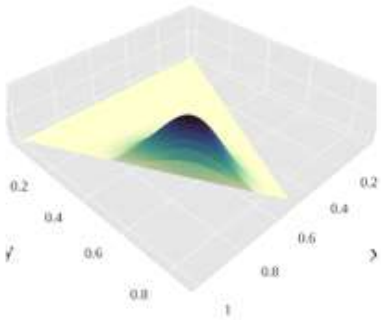
$\boldsymbol{\alpha} = (1.3, 1.3, 1.3)$



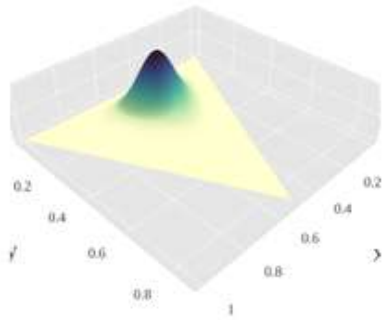
$\boldsymbol{\alpha} = (3,3,3)$



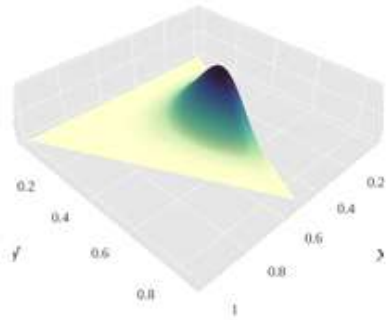
$\boldsymbol{\alpha} = (7,7,7)$



$\boldsymbol{\alpha} = (2,6,11)$

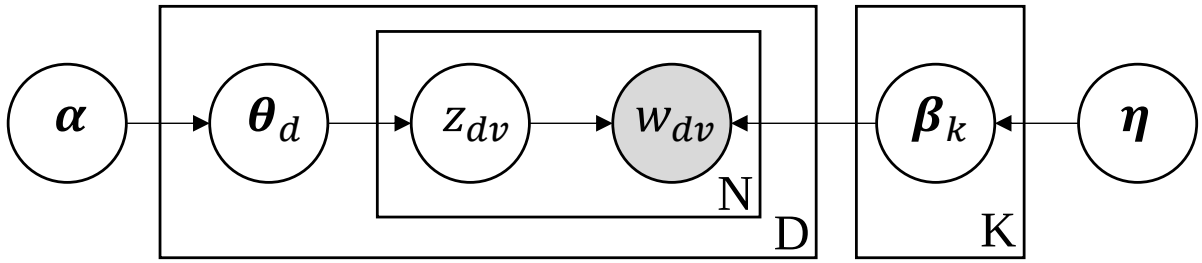


$\boldsymbol{\alpha} = (14, 9, 5)$



$\boldsymbol{\alpha} = (6,2,6)$

隐狄里克雷分配 LDA



$$p(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\beta},\boldsymbol{\Theta}|\boldsymbol{\alpha},\boldsymbol{\eta}) = \prod_{d}^{D} p(\boldsymbol{\theta}_d|\alpha) \prod_{k}^{K} p(\boldsymbol{\beta}_k|\eta) \left(\prod_{v=1}^{N_d} P(w_{dv}|z_{dv},\boldsymbol{\beta}_k) P(z_{dv}|\boldsymbol{\theta}_d) \right)$$

狄里克雷分布

LDA模型的**参数估计**

- 给定训练数据$\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$, 参数通过极大似然法估计，寻找$\boldsymbol{\alpha}$和$\boldsymbol{\eta}$以最大化对数似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\eta}) = \sum_{d=1}^{D} \ln p(\boldsymbol{w}_d | \boldsymbol{\alpha}, \boldsymbol{\eta})$$

- 求解算法
  - 可以通过吉布斯采样求解
  - 可以通过变分法求解

LDA模型的**参数估计**

- 给定训练数据$\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$, 参数通过极大似然法估计，寻找$\boldsymbol{\alpha}$和$\boldsymbol{\eta}$以最大化对数似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\eta}) = \sum_{d=1}^{D} \ln p(\boldsymbol{w}_d | \boldsymbol{\alpha}, \boldsymbol{\eta})$$

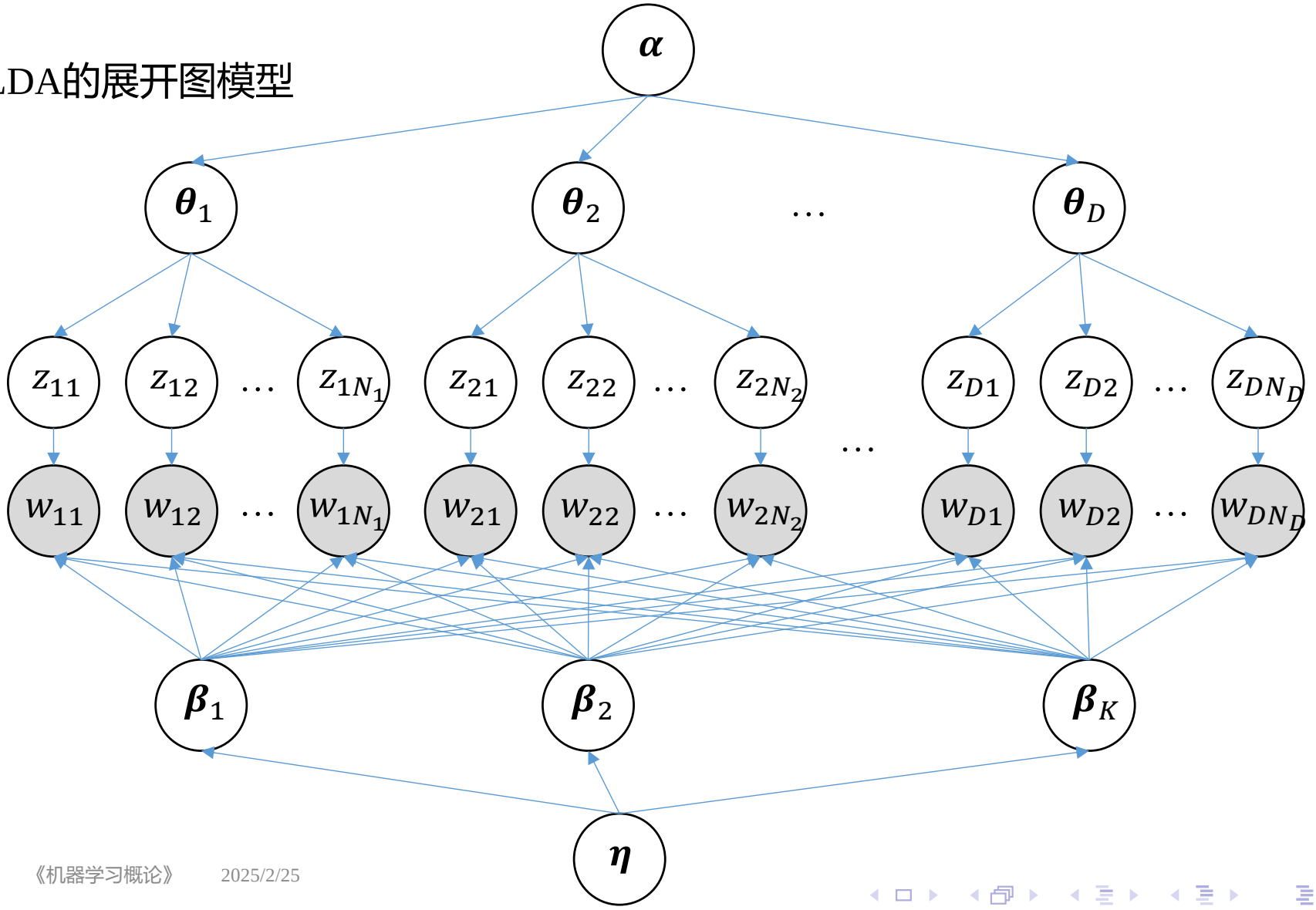
- 能否进行模型推断后用EM算法呢?

$$p(\boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \frac{p(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{\alpha}, \boldsymbol{\eta})}{p(\boldsymbol{w} | \boldsymbol{\alpha}, \boldsymbol{\eta})}$$

模型推断的挑战

$$p(\boldsymbol{z},\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\eta}) = \frac{p(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\beta},\boldsymbol{\theta}|\boldsymbol{\alpha},\boldsymbol{\eta})}{p(\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\eta})}$$

LDA的展开图模型



LDA模型的参数估计

- 给定训练数据$\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$, 参数通过极大似然法估计，寻找$\boldsymbol{\alpha}$和$\boldsymbol{\eta}$以最大化对数似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\eta}) = \sum_{d=1}^{D} \ln p(\boldsymbol{w}_d | \boldsymbol{\alpha}, \boldsymbol{\eta})$$

- 能否进行模型推断后用EM算法呢?

$$p(\boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \frac{p(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{\alpha}, \boldsymbol{\eta})}{p(\boldsymbol{w} | \boldsymbol{\alpha}, \boldsymbol{\eta})}$$

无法进行

- 求解算法
  - 可以通过吉布斯采样求解：通过使用随机化方法完成近似
  - 可以通过变分法求解：使用确定性近似完成推断

近似推断：采样法

- 核心思想：用 一组样本 近似 分布

  - 设某个计算机程序可以产生正态分布的样本，但是参数未知。那么可以不断调用该程序，产生一组样本，从而通过样本来估计均值和方差

  - 考虑计算$\mathbb{E}_p[f(x)] = \int p(x)f(x)dx$ 或者 $\mathbb{E}_p[f(x)] = \sum_x p(x)f(x)$，可以通过从分布$p(x)$中抽样n个样本$x^{(1)}, \cdots, x^{(n)}$

$$\mathbb{E}_p[f(x)] = \frac{1}{n}\sum_i f\left(x^{(i)}\right)$$

关键难题：那应该如何从$p(x)$从采样呢?

标准分布采样

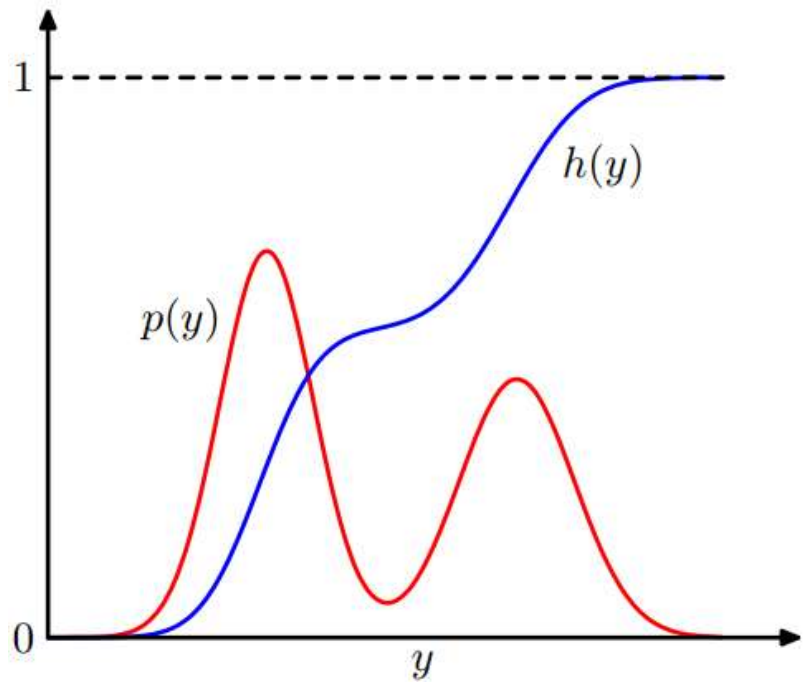
- 伯努利分布采样：$P(x) = p^x(1-p)^{1-x}$

$$z \sim U(0,1), \qquad x = \begin{cases} 1, z \le p \\ 0, othewise \end{cases}$$

$$P(z \le p) = p$$

标准分布采样

- 假设z满足均匀分布，$z \sim U(0,1)$, 通过对z进行变换可以求得相应分布$p(y)$
- $p(y) = p(z)\left|\frac{dz}{dy}\right| = \left|\frac{dz}{dy}\right|$
- 左右两边积分后，可得$z = h(y) = \int_{-\infty}^{y} p(\hat{y})d\hat{y}$
- 若$y = h^{-1}(z)$，那么$y \sim p(y)$



标准分布采样

- 指数分布采样：$p(y) = \lambda \exp(-\lambda y)$
  - $h(y) = 1 - \exp(-\lambda y)$
  - $y = h^{-1}(z) = -\lambda^{-1} \ln(1 - z)$
- 标准柯西分布采样：$p(y) = \frac{1}{\pi} \frac{1}{1+y^2}$
  - $h(y) = \frac{1}{\pi} \arctan(y) + \frac{1}{2}$
  - $y = h^{-1}(z) = \tan\left(z\pi - \frac{\pi}{2}\right)$

标准分布采样

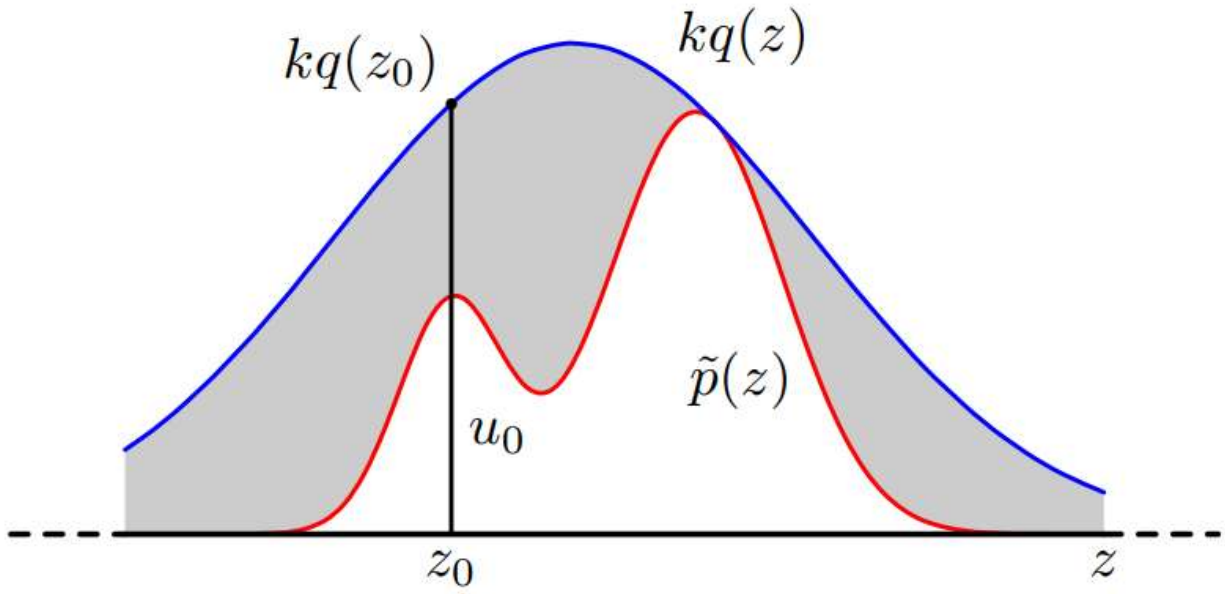
- 标准正态分布
  - $z_1, z_2 \sim U(-1,1)$
  - 如果$z_1^2 + z_2^2 > 1$，则丢掉这对样本，那么 $p(z_1, z_2) = \frac{1}{\pi}$
  - 令$y_1 = z_1 \left(\frac{-2\ln z_1}{z_1^2+z_2^2}\right)^{\frac{1}{2}}, y_2 = z_2 \left(\frac{-2\ln z_2}{z_1^2+z_2^2}\right)^{\frac{1}{2}}$, 则有
  - $p(y_1, y_2) = p(z_1, z_2)\left|\frac{\partial(z_1,z_2)}{\partial(y_1,y_2)}\right| = \left[\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}y_1^2\right)\right]\left[\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}y_2^2\right)\right]$
  - y_1和y_2 独立，且均值为0，方差为1

非标准分布采样

- 难以计算分布的累计概率分布并求逆函数
- 借助于两种重要而且通用的方法
  - 拒绝采样 Rejection sampling
  - 重要性重采样 Importance resampling

拒绝采样

- 从$p(z)$采样很难，但是计算$p(z)$很容易，可以不包括归一化项
  - $p(z) = \frac{1}{Z_p}\tilde{p}(z)$
  - $\tilde{p}(z)$很容易计算，但Z_p未知
- 提议分布 (proposal distribution): $q(z)$，可以容易从中采样
- 寻找常数k，满足$kq(z) \geq \tilde{p}(z)$



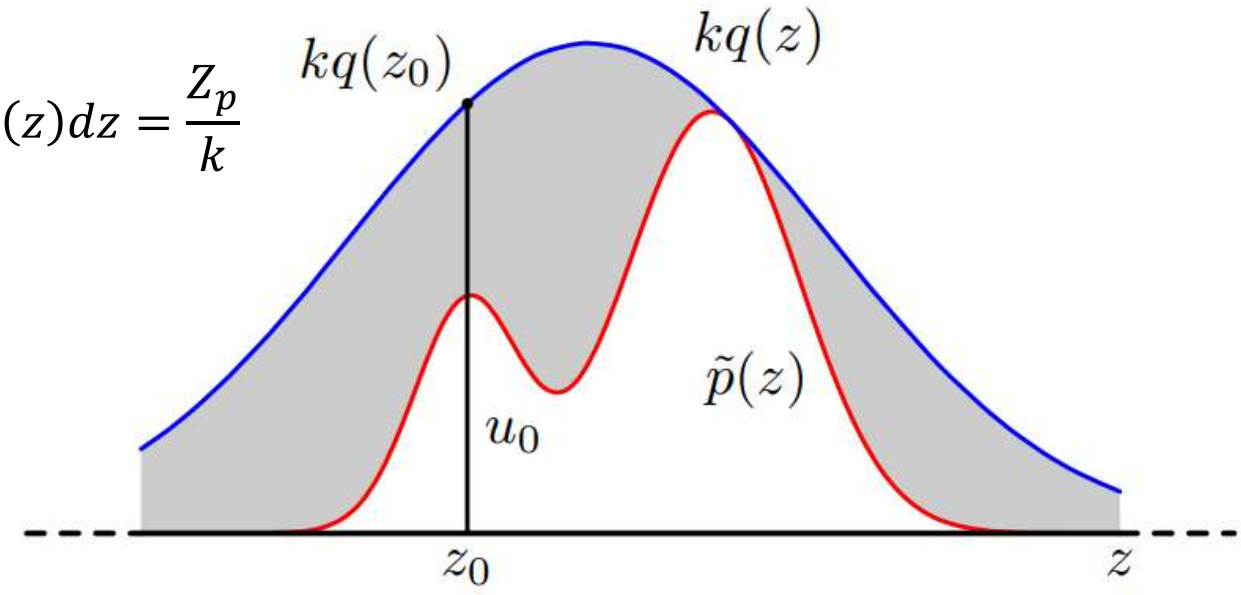
拒绝采样

采样过程

- 从$q(z)$从采样z_0
- 从$U(0, kq(z_0))$采样u_0
- 如果$u_0 > \tilde{p}(z_0)$，样本被拒绝，否则被保留

- 图中阴影部分的样本被拒绝

$$p(Accept) = \int \left\{ \frac{\tilde{p}(z)}{kq(z)} \right\} q(z)dz = \frac{Z_p}{k}$$



拒绝采样在高维分布的问题

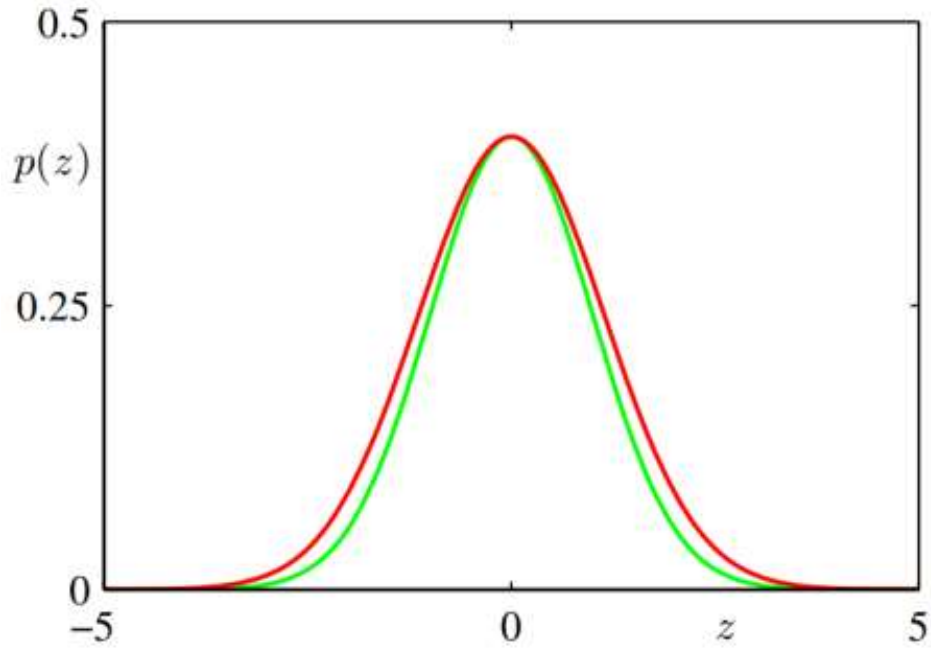
- $p(z)$是多维高斯分布，均值为0，协方差矩阵为$\sigma_p I$
- $q(z)$是多维高斯分布，均值为0，协方差矩阵为$\sigma_q I$

- 满足$\sigma_q \geq \sigma_p, k^\star = \left(\frac{\sigma_q}{\sigma_q}\right)^D$

- $P(Accept) = \frac{1}{k}$

维度越高，接收概率指数级减小

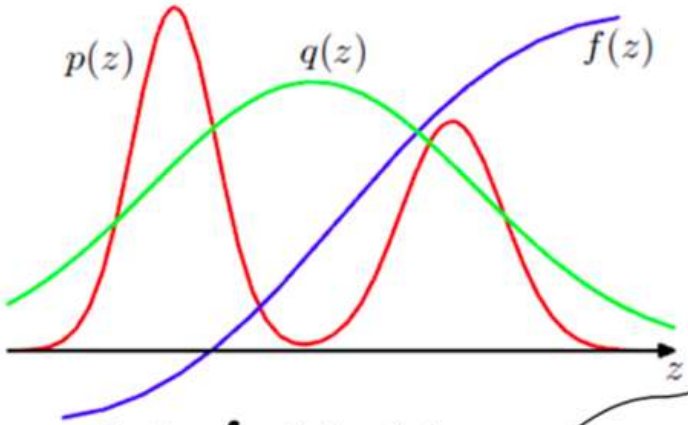
在高维情况下效率极低



重要性重采样 Importance resampling

- 并不直接丢弃样本，因此采样效率提升
- 会根据提议分布和采样分布上概率的差别给每个样本加权
  - 如果采样分布$p(z)$概率小，而提议分布$q(z)$概率大，权重越小
  - 如果采样分布$p(z)$概率大，而提议分布$q(z)$概率小，权重越大
- 有一个前提条件：在$p(z)$显著的地方$q(z)$不能太小
- 同样难以解决从高维概率分布采样的问题

重要性重采样 Importance resampling



重要性权重，用来矫正从“错误”分布采样带来的偏差

$$\mathrm{E}[f]=\int f(z)p(z)dz$$

$$=\int f(z)\frac{p(z)}{q(z)}q(z)dz$$

$$\approx\frac{1}{L}\sum_{l=1}^{L}\frac{p\left(z^{(l)}\right)}{q\left(z^{(l)}\right)}f\left(z^{(l)}\right)$$

$$\omega\left(z^{(l)}\right)=p\left(z^{(l)}\right)/q\left(z^{(l)}\right)$$

Importance Sampling

- 产生样本 $z_1,\ldots,z_L\sim q(z)$
- 计算

$$\hat{I}=\frac{1}{L}\sum_{l=1}^{L}f\left(z^{(l)}\right)\omega\left(z^{(l)}\right)$$

马尔可夫链蒙特卡罗方法（MCMC）

- 拒绝采样和重要性采样在高维情形下效率很低，通过马尔可夫链蒙特卡罗方法可以更好地扩展到高维情形

- **MCMC算法的关键在于通过“构造平稳分布为p的马尔可夫链”来产生样本**：当马尔可夫链运行足够长的时间（收敛到平稳状态），则产出的样本x近似服从p分布；并且通过多次重复运行、遍历马尔可夫链就可以取得多个服从该分布的独立同分布样本。

- 前提
  - 难以从$p(z) = \tilde{p}(z)/Z_p$中采样，但是很容易计算$\tilde{p}(z)$
  - 提议分布$q(z|z^\tau)$, 容易采样：z^τ为当前状态

- $z_1, z_2, \cdots$形成一个马尔可夫链。算法的每一轮，从提议分布中采样一个样本，按照某种规则来决定是否接受这个样本

关于MCMC的一些知识

- 一阶马尔可夫链有初始概率$p^{(0)}(z)$和转移概率$p(z'|z)$确定
- 在t时刻的状态采样概率为 $P^{(t)}(z)$，用向量$\boldsymbol{\pi}^{(t)}$来表示所有状态的概率$P^{(t)}(z)$
- 转移概率用转移矩阵表示： $A_{i,j} = p(z' = i|z = j)$
- 马尔科夫链的动态性用如下方式表示

$$P^{(t)}(z = i) = \sum_j P^{(t-1)}(z = j)P(z' = i|z = j)$$

$$\boldsymbol{\pi}^{(t)} = \boldsymbol{A}\boldsymbol{\pi}^{(t-1)} = \boldsymbol{A}^{(t)}\boldsymbol{\pi}^{(0)}$$

关于MCMC的一些知识

- 马尔科夫链的动态性用如下方式表示

$$\boldsymbol{\pi}^{(t)} = A\boldsymbol{\pi}^{(t-1)} = A^{(t)}\boldsymbol{\pi}^{(0)}$$

- A的每一列代表一个概率分布，A称为随机矩阵
- 任意状态之间在可达的情况下，A的最大特征值等于1；
- 当$t \to \infty$时，不等于1的特征值都衰减到0
- 在一些额外宽松条件下，A只有一个特征值为1的特征向量
  - MCMC会收敛到平稳分布 $\boldsymbol{\pi} = \boldsymbol{A\pi}$， 即$\boldsymbol{\pi}$为对应的特征向量

关于MCMC的一些知识

- 遍历马尔可夫链有唯一的平稳分布 $p^{\star}(z)$
  - 平稳分布满足：$p^{\star}(z)=\sum_{z'} p(z|z')p^{\star}(z')$
  - 遍历性 $\lim_{\tau\to\infty} p^{(\tau)}(z)=p^{\star}(z)$，不管初始概率分布的选择
- $p^{\star}(z)$是平稳分布的充分条件：$p^{\star}(z')p(z|z')=p(z'|z)p^{\star}(z)$

Metropolis 算法

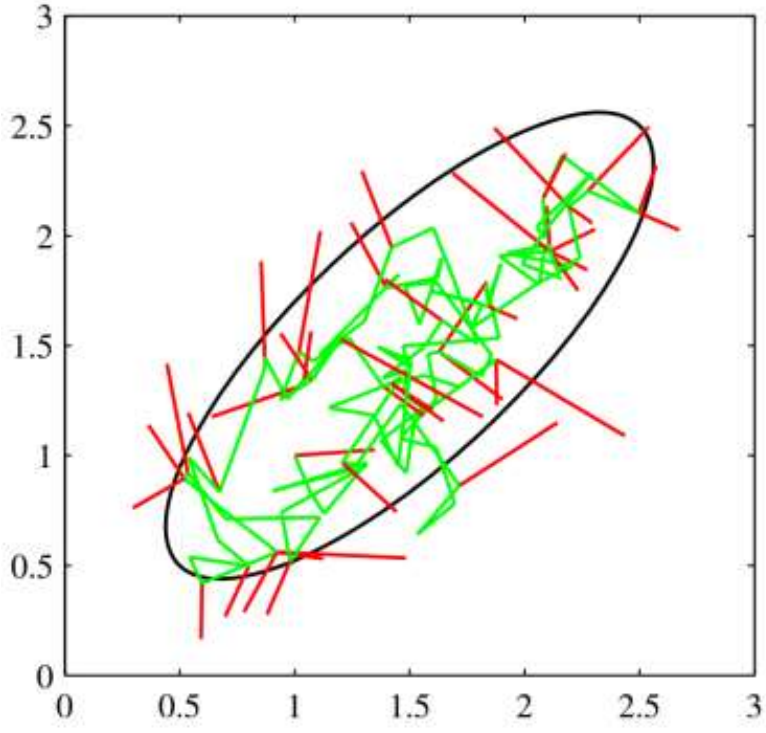
- 提议分布是对称的，满足$q(\boldsymbol{z}_A|\boldsymbol{z}_B) = q(\boldsymbol{z}_B|\boldsymbol{z}_A)$

- 从提议分布中采样的候选样本被接受的概率为

$$A(\boldsymbol{z}^\star, \boldsymbol{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\boldsymbol{z}^\star)}{\tilde{p}(\boldsymbol{z}^{(\tau)})}\right)$$

  - 如果$\tilde{p}(\boldsymbol{z}^\star) > \tilde{p}(\boldsymbol{z}^{(\tau)})$，那么$\boldsymbol{z}^\star$一定被接收

- 如果候选样本被接受，$\boldsymbol{z}^{(\tau+1)} = \boldsymbol{z}^\star$，否则$\boldsymbol{z}^{(\tau+1)} = \boldsymbol{z}^{(\tau)}$
  - 如果$\boldsymbol{z}^\star$被拒绝，$\boldsymbol{z}^{(\tau)}$被拷贝一次



状态$\boldsymbol{z}^{(\tau)}$到状态$\boldsymbol{z}^\star$的转移概率为$q(\boldsymbol{z}^\star|\boldsymbol{z}^{(\tau)})A(\boldsymbol{z}^\star, \boldsymbol{z}^{(\tau)})$

只要$q(z_A|z_B)$是正的，$z^{(\tau)}$的分布趋近于$p(z)$

Metropolis-Hastings 算法

Algorithm 24.2: Metropolis Hastings algorithm

Initialize x^0 ;

for $s = 0, 1, 2, \ldots$ **do**

  Define $x = x^s$;

  Sample $x' \sim q(x'|x)$;

  Compute acceptance probability

$$\alpha = \frac{\tilde{p}(x')q(x|x')}{\tilde{p}(x)q(x'|x)}$$

  Compute $r = \min(1, \alpha)$;

  Sample $u \sim U(0, 1)$;

  Set new sample to

$$x^{s+1} = \begin{cases} x' & \text{if } u < r \\ x^s & \text{if } u \geq r \end{cases}$$

连续变量提议分布一般选择以当前状态为均值的高斯分布，方差要做权衡

- 推广到非对称的提议分布，只需要修改接受概率

$$A\left(\boldsymbol{z}^{\star}, \boldsymbol{z}^{(\tau)}\right) = \min\left(1, \frac{\tilde{p}(\boldsymbol{z}^{\star})q\left(\boldsymbol{z}^{(\tau)}|\boldsymbol{z}^{\star}\right)}{\tilde{p}\left(\boldsymbol{z}^{(\tau)}\right)q\left(\boldsymbol{z}^{\star}|\boldsymbol{z}^{(\tau)}\right)}\right)$$

状态$\boldsymbol{z}^{(\tau)}$到状态$\boldsymbol{z}^{\star}$的转移概率为 $q\left(\boldsymbol{z}^{\star}|\boldsymbol{z}^{(\tau)}\right)A\left(\boldsymbol{z}^{\star}, \boldsymbol{z}^{(\tau)}\right)$

Metropolis-Hastings 算法

- $p(z)$是该算法定义的马尔科夫链的平稳分布

$$A(z^\star, z^{(\tau)}) = \min\left(1, \frac{\tilde{p}(z^\star)q(z^{(\tau)}|z^\star)}{\tilde{p}(z^{(\tau)})q(z^\star|z^{(\tau)})}\right)$$

$$\begin{aligned} p(z^{(\tau)})q(z^\star|z^{(\tau)})A(z^\star, z^{(\tau)}) &= \min\left(p(z^{(\tau)})q(z^\star|z^{(\tau)}), p(z^\star)q(z^{(\tau)}|z^\star)\right) \\ &= \min\left(p(z^\star)q(z^{(\tau)}|z^\star), p(z^{(\tau)})q(z^\star|z^{(\tau)})\right) \\ &= p(z^\star)q(z^{(\tau)}|z^\star)A(z^{(\tau)}, z^\star) \end{aligned}$$

Gibbs 采样

- Metropolis-Hastings 算法一个特列

状态$\boldsymbol{z}^{(\tau)}$到状态$\boldsymbol{z}^{\star}$的转移概率$q_k\left(\boldsymbol{z}^{\star}|\boldsymbol{z}^{(\tau)}\right) = p(z_k^{\star}|\boldsymbol{z}_{-\mathrm{k}})\mathrm{I}(\boldsymbol{z}_{-\mathrm{k}}^{\star}=\boldsymbol{z}_{-\mathrm{k}})$

1. Initialize $\{z_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
   - Sample $z_1^{(\tau+1)} \sim p(z_1|z_2^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - Sample $z_2^{(\tau+1)} \sim p(z_2|z_1^{(\tau+1)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - $\vdots$
   - Sample $z_j^{(\tau+1)} \sim p(z_j|z_1^{(\tau+1)}, \ldots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - $\vdots$
   - Sample $z_M^{(\tau+1)} \sim p(z_M|z_1^{(\tau+1)}, z_2^{(\tau+1)}, \ldots, z_{M-1}^{(\tau+1)})$.

可以证明：
Metropolis-Hastings步总是被接受

Gibbs 采样

- 可以证明：Metropolis-Hastings步总是被接受
  - $q_k(\boldsymbol{z}^\star|\boldsymbol{z}) = p(z_k^\star|\boldsymbol{z}_{-\mathrm{k}})\mathrm{I}(\boldsymbol{z}_{-\mathrm{k}}^\star=\boldsymbol{z}_{-\mathrm{k}})$
  - $p(\boldsymbol{z}) = p(z_k|\boldsymbol{z}_{-\mathrm{k}})p(\boldsymbol{z}_{-\mathrm{k}})$
  - $\boldsymbol{z}_{-\mathrm{k}}^\star=\boldsymbol{z}_{-\mathrm{k}}$
  - $A(\boldsymbol{z}^\star,\boldsymbol{z}) = \min\left(1,\frac{p(\boldsymbol{z}^\star)q_k(\boldsymbol{z}|\boldsymbol{z}^\star)}{p(\boldsymbol{z})q_k(\boldsymbol{z}^\star|\boldsymbol{z})}\right)$

$$= \min\left(1,\frac{p(z_k^\star|\boldsymbol{z}_{-\mathrm{k}}^\star)p(\boldsymbol{z}_{-\mathrm{k}}^\star)p(z_k|\boldsymbol{z}_{-\mathrm{k}}^\star)\mathrm{I}(\boldsymbol{z}_{-\mathrm{k}}^\star=\boldsymbol{z}_{-\mathrm{k}})}{p(z_k|\boldsymbol{z}_{-\mathrm{k}})p(\boldsymbol{z}_{-\mathrm{k}})p(z_k^\star|\boldsymbol{z}_{-\mathrm{k}})\mathrm{I}(\boldsymbol{z}_{-\mathrm{k}}^\star=\boldsymbol{z}_{-\mathrm{k}})}\right)$$

$$= 1$$

LDA模型的参数估计

- 给定训练数据$\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$, 参数通过极大似然法估计，寻找$\boldsymbol{\alpha}$和$\boldsymbol{\eta}$以最大化对数似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\eta}) = \sum_{d=1}^{D} \ln p(\boldsymbol{w}_d | \boldsymbol{\alpha}, \boldsymbol{\eta})$$

- 能否进行模型推断后用EM算法呢?

$$p(\boldsymbol{z}_d, \boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{w}_d, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \frac{p(\boldsymbol{w}_d, \boldsymbol{z}_d, \boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{\alpha}, \boldsymbol{\eta})}{p(\boldsymbol{w}_d | \boldsymbol{\alpha}, \boldsymbol{\eta})}$$

- 求解算法
  - 可以通过收缩的吉布斯采样求解：通过使用随机化方法完成近似

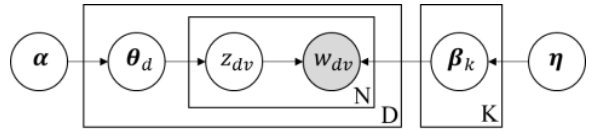
LDA模型的**参数估计**—吉布斯采样

- 基本思想是通过对隐变量θ和β积分，得到边缘概率 $P(\boldsymbol{z}_d|\boldsymbol{w}_d,\boldsymbol{\alpha},\boldsymbol{\eta})$
- 对后验概率进行吉布斯抽样，得到分布$P(\boldsymbol{z}_d|\boldsymbol{w}_d,\boldsymbol{\alpha},\boldsymbol{\eta})$的样本集合
- 利用这个样本集合对参数$\boldsymbol{\alpha}$ 和$\boldsymbol{\eta}$ 进行参数估计

LDA模型的**参数估计**—吉布斯采样

- 基本思想是通过对隐变量θ和β积分，得到边缘概率$P(\boldsymbol{z}|\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\eta})$

$$P(\boldsymbol{z}|\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\eta}) = \frac{P(\boldsymbol{z},\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\eta})}{P(\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\eta})}$$



$$\propto P(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\alpha},\boldsymbol{\eta})P(\boldsymbol{z}|\boldsymbol{\alpha},\boldsymbol{\eta}) = P(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\eta})P(\boldsymbol{z}|\boldsymbol{\alpha})$$

$$P(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\eta}) = \int p(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\beta})p(\boldsymbol{\beta}|\boldsymbol{\eta})d\boldsymbol{\beta} = \int \prod_{v=1}^{V}\prod_{k=1}^{K}\beta_{kv}^{n_{kv}} \prod_{k=1}^{K}\left(\frac{1}{B(\boldsymbol{\eta})}\prod_{v=1}^{V}\beta_{kv}^{\eta_v-1}\right)d\boldsymbol{\beta}$$

$$p(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\beta}) = \prod_{d=1}^{D}\prod_{w=1}^{N_d}\beta_{z_{dw}w} = \prod_{d=1}^{D}\prod_{w=1}^{N_d}\prod_{k=1}^{K}\beta_{kw}^{\mathbb{I}(z_{dw}=k)}$$

$$= \prod_{v=1}^{V}\prod_{k=1}^{K}\beta_{kv}^{\sum_d\sum_{w=v}\mathbb{I}(z_{dw}=k)}$$

$$= \prod_{v=1}^{V}\prod_{k=1}^{K}\beta_{kv}^{n_{kv}}$$

数据中第k个话题生成第v个单词的次数

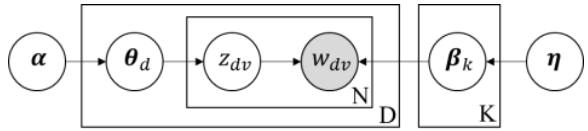
$$= \prod_{k=1}^{K}\frac{1}{B(\boldsymbol{\eta})}\left(\int\prod_{v=1}^{V}\beta_{kv}^{n_{kv}+\eta_v-1}d\boldsymbol{\beta}_k\right)$$

$$= \prod_{k=1}^{K}\frac{B(\boldsymbol{\eta}+\boldsymbol{n}_k)}{B(\boldsymbol{\eta})}$$

LDA模型的**参数估计**—吉布斯采样

- 基本思想是通过对隐变量θ和β积分，得到边缘概率$P(\boldsymbol{z}|\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\eta})$

$$P(\boldsymbol{z}|\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\eta}) = \frac{P(\boldsymbol{z},\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\eta})}{P(\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\eta})}$$



$$\propto P(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\alpha},\boldsymbol{\eta})P(\boldsymbol{z}|\boldsymbol{\alpha},\boldsymbol{\eta}) = P(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\eta})P(\boldsymbol{z}|\boldsymbol{\alpha})$$

$$P(\boldsymbol{z}|\boldsymbol{\alpha}) = \int p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})d\boldsymbol{\theta} = \int \prod_{d=1}^{D}\prod_{k=1}^{K}\theta_{dk}^{n_{dk}} \prod_{d=1}^{D}\left(\frac{1}{B(\boldsymbol{\alpha})}\prod_{k=1}^{K}\theta_{dk}^{\alpha_k-1}\right)d\boldsymbol{\theta}$$

$$p(\boldsymbol{z}|\boldsymbol{\theta}) = \prod_{d=1}^{D} p(\boldsymbol{z}_d|\boldsymbol{\theta}_d) = \prod_{d=1}^{D}\prod_{w=1}^{V}\theta_{d\,z_{dw}}$$

$$= \prod_{d=1}^{D}\prod_{k=1}^{K}\theta_{dk}^{\sum_w \mathbb{I}(z_{dw}=k)}$$

$$= \prod_{d=1}^{D}\prod_{k=1}^{K}\theta_{dk}^{n_{dk}}$$

数据中第d个文档生成第k个单词的次数

$$= \prod_{d=1}^{D}\frac{1}{B(\boldsymbol{\alpha})}\left(\int \prod_{k=1}^{K}\theta_{dk}^{n_{dk}+\alpha_k-1}\,d\boldsymbol{\theta}_d\right)$$

$$= \prod_{d=1}^{D}\frac{B(\boldsymbol{\alpha}+\boldsymbol{n}_d)}{B(\boldsymbol{\alpha})}$$

LDA模型的**参数估计**—吉布斯采样

- 基本思想是通过对隐变量θ和β积分，得到边缘概率$P(\boldsymbol{z}|\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\eta})$

$$P(\boldsymbol{z}|\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\eta}) = \frac{P(\boldsymbol{z},\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\eta})}{P(\boldsymbol{w}|\boldsymbol{\alpha},\boldsymbol{\eta})}$$

$$\propto P(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\alpha},\boldsymbol{\eta})P(\boldsymbol{z}|\boldsymbol{\alpha},\boldsymbol{\eta}) = P(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\eta})P(\boldsymbol{z}|\boldsymbol{\alpha})$$

$$P(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\eta}) = \prod_{k=1}^{K}\frac{B(\boldsymbol{\eta}+\boldsymbol{n}_k)}{B(\boldsymbol{\eta})} \qquad P(\boldsymbol{z}|\boldsymbol{\alpha}) = \prod_{d=1}^{D}\frac{B(\boldsymbol{\alpha}+\boldsymbol{n}_d)}{B(\boldsymbol{\alpha})}$$

$$\Rightarrow \quad P(\boldsymbol{z}|\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\eta}) \propto \prod_{k=1}^{K}\frac{B(\boldsymbol{\eta}+\boldsymbol{n}_k)}{B(\boldsymbol{\eta})}\prod_{d=1}^{D}\frac{B(\boldsymbol{\alpha}+\boldsymbol{n}_d)}{B(\boldsymbol{\alpha})}$$

LDA模型的**参数估计**—吉布斯采样

即第d'文档中的第w个词

记所有文本的单词序列的第i个单词为$w_i = v'$，对应话题为$z_i = \mathrm{k}'$的条件概率为

$$P(z_i = k'|\boldsymbol{z}_{-i}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\eta}) \propto \frac{P(\boldsymbol{z}|\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\eta})}{P(\boldsymbol{z}_{-i}|\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\eta})} = \frac{\prod_{k=1}^{K} \frac{B(\boldsymbol{\eta}+\boldsymbol{n}_k)}{B(\boldsymbol{\eta})} \prod_{d=1}^{D} \frac{B(\boldsymbol{\alpha}+\boldsymbol{n}_d)}{B(\boldsymbol{\alpha})}}{\prod_{k=1}^{K} \frac{B\left(\boldsymbol{\eta}+\boldsymbol{n}_k^{(-i)}\right)}{B(\boldsymbol{\eta})} \prod_{d=1}^{D} \frac{B\left(\boldsymbol{\alpha}+\boldsymbol{n}_d^{(-i)}\right)}{B(\boldsymbol{\alpha})}}$$

$$= \frac{\prod_{k=1}^{K} B(\boldsymbol{\eta}+\boldsymbol{n}_k)}{\prod_{k=1}^{K} B\left(\boldsymbol{\eta}+\boldsymbol{n}_k^{(-i)}\right)} \cdot \frac{\prod_{d=1}^{D} B(\boldsymbol{\alpha}+\boldsymbol{n}_d)}{\prod_{d=1}^{D} B\left(\boldsymbol{\alpha}+\boldsymbol{n}_d^{(-i)}\right)}$$

$$\frac{1}{B(\boldsymbol{\alpha})} = \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)}$$

$$= \frac{\frac{\prod_{k,v} \Gamma(\eta_v+n_{kv})}{\prod_k \Gamma(\sum_v \eta_v+n_{kv})}}{\frac{\Gamma\left(\eta_{v'}+n_{k'v'}-1\right) \prod_{k\neq k', v\neq v'} \Gamma(\eta_v+n_{kv})}{\Gamma\left(-1+\sum_v \eta_v+n_{k'v}\right) \prod_{k\neq k'} \Gamma(\sum_v \eta_v+n_{kv})}} \cdot \frac{\frac{\prod_{d,k} \Gamma(\alpha_k+n_{dk})}{\prod_d \Gamma\left(\sum_k \alpha_k+n_{dk}\right)}}{\frac{\Gamma\left(\alpha_{k'}+n_{d'k'}-1\right) \prod_{d=d', k\neq k'} \Gamma(\alpha_k+n_{dk})}{\Gamma\left(-1+\sum_k \alpha_k+n_{d'k}\right) \prod_{d'\neq d} \Gamma\left(\sum_k \alpha_k+n_{dk}\right)}}$$

$$= \frac{\frac{\Gamma\left(\eta_{v'}+n_{k'v'}\right)}{\Gamma\left(\sum_v \eta_v+n_{k'v}\right)}}{\frac{\Gamma\left(\eta_{v'}+n_{k'v'}-1\right)}{\Gamma\left(-1+\sum_v \eta_v+n_{k'v}\right)}} \cdot \frac{\frac{\Gamma\left(\alpha_{k'}+n_{d'k'}\right)}{\Gamma\left(\sum_k \alpha_k+n_{d'k}\right)}}{\frac{\Gamma\left(\alpha_{k'}+n_{d'k'}-1\right)}{\Gamma\left(-1+\sum_k \alpha_k+n_{d'k}\right)}} = \frac{\eta_{v'}+n_{k'v'}-1}{-1+\sum_v \eta_v+n_{k'v}} \cdot \frac{\alpha_{k'}+n_{d'k'}-1}{-1+\sum_k \alpha_k+n_{d'k}}$$

LDA模型的**参数估计**—吉布斯采样

- 估计参数$\boldsymbol{\theta}_d$

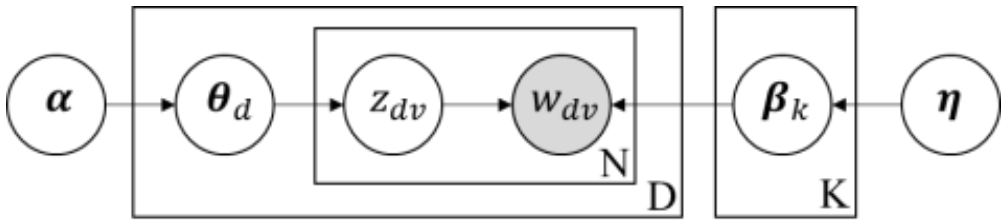
$$p(\boldsymbol{\theta}_d|\boldsymbol{z}_d,\boldsymbol{\alpha}) \propto p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})P(\boldsymbol{z}_d|\boldsymbol{\theta}_d) \propto \prod_{k=1}^{K} \theta_{dk}^{\alpha_k-1+n_{dk}}$$

$$p(\boldsymbol{\theta}_d|\boldsymbol{z}_d,\boldsymbol{\alpha}) = Dir(\boldsymbol{\theta}_d|\boldsymbol{n}_d+\boldsymbol{\alpha}) \Rightarrow \theta_{dk} = \frac{n_{dk}+\alpha_k}{\sum_k(n_{dk}+\alpha_k)}$$

- 估计参数$\boldsymbol{\beta}_k$

$$p(\boldsymbol{\beta}_k|\boldsymbol{w},\boldsymbol{\eta}) \propto P(\boldsymbol{w}|\boldsymbol{\beta}_k)p(\boldsymbol{\beta}_k|\boldsymbol{\eta}) \propto \prod_{v=1}^{V} \beta_{kv}^{n_{kv}+\eta_v-1}$$

$$p(\boldsymbol{\beta}_k|\boldsymbol{w},\boldsymbol{\eta}) = Dir(\boldsymbol{\beta}_k|\boldsymbol{n}_k+\boldsymbol{\eta}) \Rightarrow \beta_{kv} = \frac{n_{kv}+\eta_v}{\sum_v(n_{kv}+\eta_v)}$$



LDA模型的参数估计

- 给定训练数据$\boldsymbol{w} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_D\}$, 参数通过极大似然法估计，寻找$\boldsymbol{\alpha}$和$\boldsymbol{\eta}$以最大化对数似然

$$LL(\boldsymbol{\alpha}, \boldsymbol{\eta}) = \sum_{d=1}^{D} \ln p(\boldsymbol{w}_d | \boldsymbol{\alpha}, \boldsymbol{\eta})$$

- 能否进行模型推断后用EM算法呢?

$$p(\boldsymbol{z}_d, \boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{w}_d, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \frac{p(\boldsymbol{w}_d, \boldsymbol{z}_d, \boldsymbol{\beta}, \boldsymbol{\theta} | \boldsymbol{\alpha}, \boldsymbol{\eta})}{p(\boldsymbol{w}_d | \boldsymbol{\alpha}, \boldsymbol{\eta})}$$

- 求解算法
  - 可以通过收缩的吉布斯采样求解：通过使用随机化方法完成近似
  - 可以通过变分法求解：使用确定性近似完成推断

变分推断—EM算法回顾

$$\mathcal{L}(q,\boldsymbol{\Theta}) = \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln \frac{p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\Theta})}{q(\boldsymbol{z})}$$

$$\max_{\boldsymbol{\Theta}} \ln p(\boldsymbol{x}|\boldsymbol{\Theta}) = \max_{\boldsymbol{\Theta}} \max_{q} \mathcal{L}(q,\boldsymbol{\Theta})$$

E 基于$\boldsymbol{\Theta}^t$推断隐变量$\boldsymbol{z}$的分布$p(\boldsymbol{z}\,|\,\boldsymbol{x},\boldsymbol{\Theta}^t)$，并计算对数似然$LL(\boldsymbol{\Theta}|\boldsymbol{x},\boldsymbol{z})$关于$\boldsymbol{z}$的期望;

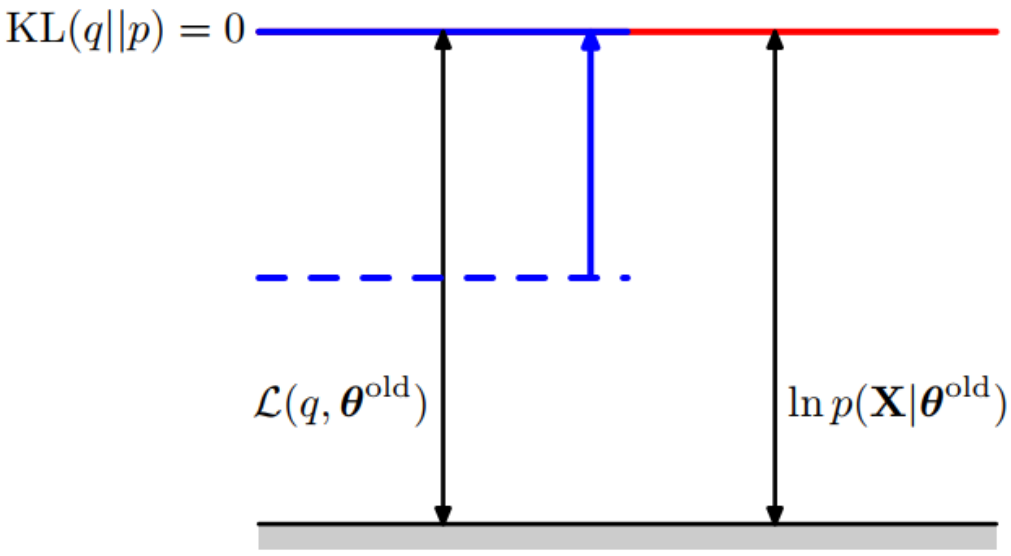
$$\mathcal{L}(p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\Theta}),\boldsymbol{\Theta}) = \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}\,|\,\boldsymbol{x},\boldsymbol{\Theta}^t)} LL(\boldsymbol{\Theta}|\boldsymbol{x},\boldsymbol{z}) \quad = Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$$

M 寻找参数最大化期望似然;

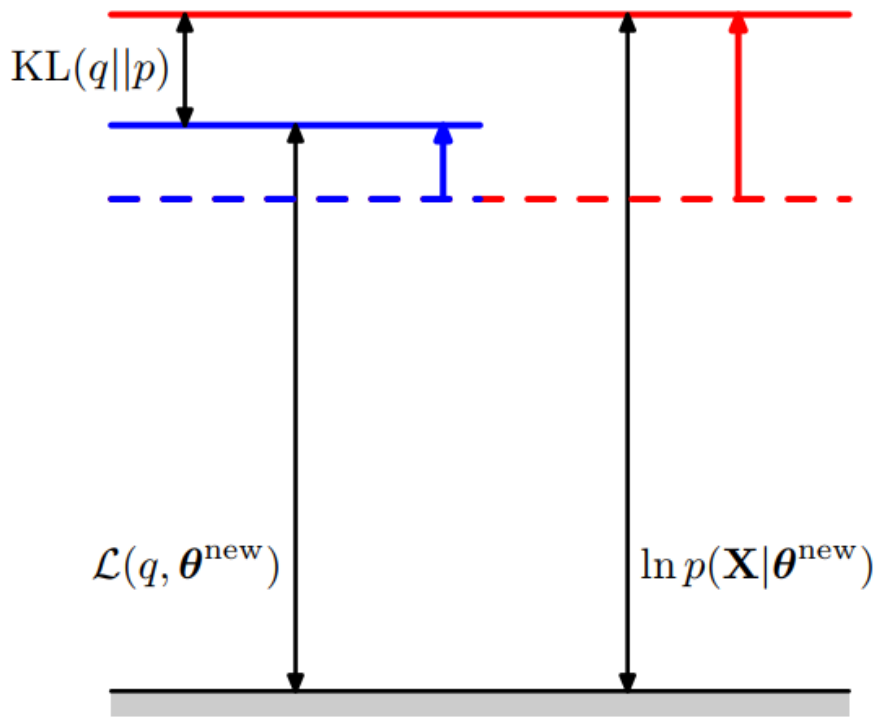
$$\boldsymbol{\Theta}^{t+1} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^t)$$

变分推断—EM算法回顾

$$\max_{\boldsymbol{\Theta}} \ln p(\boldsymbol{x}|\boldsymbol{\Theta}) = \max_{\boldsymbol{\Theta}} \max_{q} \mathcal{L}(q, \boldsymbol{\Theta})$$



EM算法的E步要求能求出最优的后验分布，若无法求解，如何是好?



答：近似后验分布

变分推断

- 平均场近似假设： $q(\boldsymbol{z}) = \prod_{i=1}^{M} q_i(\boldsymbol{z}_i)$

复杂的多变量$\boldsymbol{z}$可以拆解为一系列相互独立的多变量$\boldsymbol{z}_i$

- 此时最优解第i组随机变量的概率分布最优值为

$$q_i^*(\boldsymbol{z}_i) \propto \exp \mathbb{E}_{\boldsymbol{z}_{-i}}[\ln p(\boldsymbol{x}, \boldsymbol{z})]$$

证明

$$\mathcal{L}(q, \boldsymbol{\Theta}) = \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \ln p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\Theta}) - H(q)$$

$$= \sum_{\boldsymbol{z}_i} q_i(\boldsymbol{z}_i) \sum_{\boldsymbol{z}_{-i}} q_{-i}(\boldsymbol{z}_{-i}) \ln p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\Theta}) - \sum_{\boldsymbol{z}_i} q_i(\boldsymbol{z}_i) \ln q_i(\boldsymbol{z}_i) + \text{const}$$

$$= \sum_{\boldsymbol{z}_i} q_i(\boldsymbol{z}_i) \mathbb{E}_{\boldsymbol{z}_{-i}}[\ln p(\boldsymbol{x}, \boldsymbol{z})] - \sum_{\boldsymbol{z}_i} q_i(\boldsymbol{z}_i) \ln q_i(\boldsymbol{z}_i) + \text{const}$$

对$q_i(\boldsymbol{z}_i)$ 求导数，并令其等于0，便可得到上述最优解

变分推断

- 平均场近似假设： $q(\boldsymbol{z}) = \prod_{i=1}^{M} q_i(\boldsymbol{z}_i)$

复杂的多变量$\boldsymbol{z}$可以拆解为一系列相互独立的多变量$\boldsymbol{z}_i$

- 此时最优解第i组随机变量的概率分布最优值为

$$q_i^*(\boldsymbol{z}_i) \propto \exp \mathbb{E}_{\boldsymbol{z}_{-i}}[\ln p(\boldsymbol{x}, \boldsymbol{z})]$$

由于在对$\boldsymbol{z}_i$所服从的分布q_i^*估计时融合了$\boldsymbol{z}_i$之外的其它$\boldsymbol{z}_{-i}$的信息，即通过联合似然函数$\ln p(\mathrm{x}, \mathrm{z})$在$z_j$之外的隐变量分布上求期望得到的，因此亦称为“平均场” （mean field）方法

LDA模型的**参数估计**—变分推断

- 为简单起见，只考虑一个文档，记为$\boldsymbol{w} = (w_1, \dots, w_N)$；对应话题为$\boldsymbol{z} = (z_1, \dots, z_N)$

- 联合概率分布 $p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{n=1}^{N} p(z_n|\boldsymbol{\theta}) p(w_n|z_n, \boldsymbol{\beta})$

- 平均场假设 $q(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{\gamma}, \boldsymbol{\phi}) = q(\boldsymbol{\theta}|\boldsymbol{\gamma}) \prod_{n=1}^{N} q(z_n|\boldsymbol{\phi}_n)$

- 证据下界

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\phi}) = \mathbb{E}_q[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha})] + \sum_{n=1}^{N} \left(\mathbb{E}_q[p(z_n|\boldsymbol{\theta})] + \mathbb{E}_q[\log p(w_n|z_n, \boldsymbol{\beta})] \right)$$
$$- \mathbb{E}_q[q(\boldsymbol{\theta}|\boldsymbol{\gamma})] - \sum_{n=1}^{N} \mathbb{E}_q[\log q(z_n|\boldsymbol{\phi}_n)]$$

LDA模型的**参数估计**—变分推断

$$\mathcal{L}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\phi}) = \mathbb{E}_q[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha})] + \sum_{n=1}^{N}\left(\mathbb{E}_q[p(z_n|\boldsymbol{\theta})] + \mathbb{E}_q[\log p(w_n|z_n,\boldsymbol{\eta})]\right) - \mathbb{E}_q[q(\boldsymbol{\theta}|\boldsymbol{\gamma})] - \sum_{n=1}^{N}\mathbb{E}_q[\log q(z_n|\boldsymbol{\phi}_n)]$$

$$\mathbb{E}_q[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha})] = \mathbb{E}_{\boldsymbol{q}}\left[\log\frac{1}{B(\boldsymbol{\alpha})}\prod_{k=1}^{K}\theta_k^{\alpha_k-1}\right] = \mathbb{E}_{\boldsymbol{q}}\left[\log\frac{1}{B(\boldsymbol{\alpha})}\prod_{k=1}^{K}\theta_k^{\alpha_k-1}\right]$$

$$= \sum_{k}^{K}(\alpha_k-1)\mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\gamma})}[\log\theta_k] + \log\Gamma\left(\sum_{k=1}^{K}\alpha_k\right) - \sum_{k=1}^{K}\log\Gamma(\alpha_k)$$

$$= \sum_{k}^{K}(\alpha_k-1)\left(\Psi(\gamma_k) - \Psi\left(\sum_{k}^{K}\gamma_k\right)\right) + \log\Gamma\left(\sum_{k=1}^{K}\alpha_k\right) - \sum_{k=1}^{K}\log\Gamma(\alpha_k)$$

$$\Psi(\gamma_k) = \frac{d}{d\gamma_k}\Gamma(\gamma_k)$$

LDA模型的**参数估计**—变分推断

$$\mathcal{L}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\phi}) = \mathbb{E}_q[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha})] + \sum_{n=1}^{N}\left(\mathbb{E}_q[p(z_n|\boldsymbol{\theta})] + \mathbb{E}_q[\log p(w_n|z_n,\boldsymbol{\beta})]\right) - \mathbb{E}_q[q(\boldsymbol{\theta}|\boldsymbol{\gamma})] - \sum_{n=1}^{N}\mathbb{E}_q[\log q(z_n|\boldsymbol{\phi}_n)]$$

$$\mathbb{E}_q[p(z_n|\boldsymbol{\theta})] = \sum_{k=1}^{K}\phi_{nk}\left(\Psi(\gamma_k) - \Psi\left(\sum_{k=1}^{K}\gamma_k\right)\right)$$

$$\mathbb{E}_q[\log p(w_n|z_n,\boldsymbol{\beta})] = \sum_{k=1}^{K}\sum_{v=1}^{V}\phi_{nk}\mathbb{I}(w_n = v)\log\beta_{kv}$$

$$\mathbb{E}_q[q(\boldsymbol{\theta}|\boldsymbol{\gamma})] = \sum_{k=1}^{K}(\gamma_k - 1)\left(\Psi(\gamma_k) - \Psi\left(\sum_{k}^{K}\gamma_k\right)\right) + \log\Gamma\left(\sum_{k=1}^{K}\gamma_k\right) - \sum_{k=1}^{K}\log\Gamma(\gamma_k)$$

$$\mathbb{E}_q[\log q(z_n|\boldsymbol{\phi})] = \sum_{k=1}^{K}\phi_{nk}\log\phi_{nk}$$

LDA模型的**参数估计**—变分推断

$$\mathcal{L}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\phi}) = \mathbb{E}_q[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha})] + \sum_{n=1}^{N}\left(\mathbb{E}_q[p(z_n|\boldsymbol{\theta})] + \mathbb{E}_q[\log p(w_n|z_n,\boldsymbol{\beta})]\right) - \mathbb{E}_q[q(\boldsymbol{\theta}|\boldsymbol{\gamma})] - \sum_{n=1}^{N}\mathbb{E}_q[\log q(z_n|\boldsymbol{\phi}_n)]$$

对ϕ_{nk}求偏导数，令其等于0，

$$\Psi(\gamma_k) - \Psi\left(\sum_{k=1}^{K}\gamma_k\right) + \sum_{v=1}^{V}\mathbb{I}(w_n = v)\log\beta_{kv} - 1 - \log\phi_{nk} = 0$$

$$\Rightarrow \quad \phi_{nk} \propto \beta_{k\,w_n}\exp\left(\Psi(\gamma_k) - \Psi\left(\sum_{k=1}^{K}\gamma_k\right)\right)$$

LDA模型的**参数估计**—变分推断

$$\mathcal{L}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\phi}) = \mathbb{E}_q[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha})] + \sum_{n=1}^{N}\left(\mathbb{E}_q[p(z_n|\boldsymbol{\theta})] + \mathbb{E}_q[\log p(w_n|z_n,\boldsymbol{\beta})]\right) - \mathbb{E}_q[q(\boldsymbol{\theta}|\boldsymbol{\gamma})] - \sum_{n=1}^{N}\mathbb{E}_q[\log q(z_n|\boldsymbol{\phi}_n)]$$

$$L[\gamma_k] = \sum_{k=1}^{K}(\alpha_k - \gamma_k + \sum_n \phi_{nk})\left(\Psi(\gamma_k) - \Psi\left(\sum_{k=1}^{K}\gamma_k\right)\right) - \log\Gamma\left(\sum_{k=1}^{K}\gamma_k\right) + \log\Gamma(\gamma_k)$$

对γ_k求偏导数，令其等于0，

$$\frac{\partial L}{\partial \gamma_k} = (\alpha_k - \gamma_k + \sum_n \phi_{nk})\left(\Psi'(\gamma_k) - \Psi'\left(\sum_{l=1}^{K}\gamma_l\right)\right) = 0$$

$$\gamma_k = \alpha_k + \sum_n \phi_{nk}$$

LDA模型的**参数估计**—变分推断

- 估计参数β
  - 写出全数据集上的对数似然，在等式约束下最大化似然

$$L[\boldsymbol{\beta}] = \sum_{d=1}^{D}\sum_{n=1}^{N_d}\sum_{k=1}^{K}\sum_{v=1}^{V}\phi_{dnk}\mathbb{I}(w_{dn}=v)\log\beta_{kv} \qquad \sum_{v}\beta_{kv}=1$$

$$\Rightarrow \quad \beta_{kv} = \sum_{d=1}^{D}\sum_{n=1}^{N_d}\eta_{dnk}\mathbb{I}(w_{dn}=v)$$

- 估计参数$\boldsymbol{\alpha}$

$$L(\boldsymbol{\alpha}) = \sum_{d=1}^{D}\left(\sum_{k=1}^{K}(\alpha_k-1)\left(\Psi(\gamma_{dk})-\Psi\left(\sum_{k}^{K}\gamma_{dk}\right)\right)+\log\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)-\sum_{k=1}^{K}\log\Gamma(\alpha_k)\right)$$

用梯度上升法进行优化求解

作业

- 0. *详述除第一项外其余四项的化简过程(已删)

1. 假设数据集$D = \{x_1, x_2, \ldots, x_m\}$, 任意$x_i$是从均值为μ、方差为$\lambda^{-1}$的正态分布$\mathcal{N}(\mu, \lambda^{-1})$中独立采样而得到。假设$\mu$和$\lambda$的先验分布为$p(\mu, \lambda) = \mathcal{N}(\mu|\mu_0, (\kappa_0\lambda)^{-1})\mathrm{Gam}(\lambda|a_0, b_0)$其中

$$\mathrm{Gam}(\lambda|a_0, b_0) = \frac{1}{\Gamma(a_0)} b_0^{a_0} \lambda^{a_0-1} \exp(-b_0\lambda)$$

(1) 请写出联合概率分布$p(D, \mu, \lambda)$

(2) 请写出证据下界（即变分推断的优化目标），并证明其为观测数据边际似然$\sum_{i=1}^{m} \log p(x_i)$的下界

(3) 请用变分推断法近似推断后验概率$p(\mu, \lambda|D)$

2. PPT 32，给出CRF的预测问题的解法