

## 第2章 Agents

### 第3章 通过搜索进行问题求解

#### 3.1.1 良定义的问题及解

#### 3.1.2 问题的形式化

#### 3.2 问题实例

##### 3.2.1 玩具问题

##### 3.2.2 现实世界问题

#### 3.3 通过搜索求解

##### 3.3.1 搜索算法基础

##### 3.3.2 问题求解算法的性能

#### 3.4 无信息搜索策略

##### 3.4.1 宽度优先搜索

##### 3.4.2 一致代价搜索

##### 3.4.3 深度优先搜索

##### 3.4.4 深度受限搜索

##### 3.4.5 迭代加深的深度优先搜索 (IDS)

##### 3.4.6 双向搜索

##### 3.4.7 无信息搜索策略对比

##### 重复状态问题

##### 图搜索与树搜索 (ch4 ppt 70也有)

#### Ch4ppt

#### 3.5 有信息(启发式)的搜索策略

##### 3.5.1 贪婪最佳优先搜索

##### 3.5.2 A\*搜索: 缩小总评估代价

##### 3.5.3 存储受限的启发式搜索

#### 3.6 启发式函数 ppt 35

### 第4章 超越经典搜索

#### 4.1 局部搜索算法和最优化问题

##### 爬山法

##### 模拟退火搜索

##### 局部剪枝搜索

##### 遗传算法

### 第5章 对抗搜索 (Game Playing)

#### 5.1 博弈

#### 5.2 博弈中的优化决策

#### 5.4 不完美的实时决策

#### PPT: p39: AlphaGo的神经网络

#### 5.5 随机博弈

#### 5.6 部分可观察的博弈

### 第6章 约束满足问题 (CSP)

#### 6.1 定义约束满足问题

#### 6.2 约束传播: CSP中的推理

#### 6.3 CSP的回溯搜索

#### 6.4 CSP局部搜索

#### 6.5 问题的结构

### 第7章 逻辑Agent

#### 7.1 基于知识的Agent

#### 7.2 Wumpus世界

#### 7.3 逻辑 书202-204

#### 7.4 命题逻辑

#### 7.5 命题逻辑定理证明

##### 7.5.1, 7.5.2

##### 7.5.3

##### 7.5.4

##### PPT

第8章 一阶逻辑

- 8.1.2 结合形式语言和自然语言优势
- 8.2 一阶逻辑的语法和语义
- 8.3 运用一阶逻辑
- 8.4 一阶逻辑的知识工程

第9章 一阶逻辑的推理

- 9.1 命题推理与一阶推理
- 9.2 合一和提升
- 9.3 前向链接
- 9.4 反向链接
- 9.5 归结

第13章 不确定性

- 13.1 不确定性
- 13.2 基本概率符号
- 13.3 使用完全联合分布进行推理
- 13.4 独立性及贝叶斯规则

第14章 概率推理（贝叶斯网络）

- 14.1 不确定性问题域中的知识表示
- 14.2 贝叶斯网络的语义
- 14.4 贝叶斯网络中的精确推理

Ch18 Learning-1

- Introduction to ML

ch18 Learning-2

- 监督学习
  - 决策树
  - 最近邻模型
  - 线性预测
- 常用概念

ch20 SVM

ch21 Logistic regression

ch22 神经网络

ch23 无监督学习

- PCA
- 聚类

## 第2章 Agents

---

- 概念介绍：环境、传感器、动作、执行器
- 吸尘器案例
- 理性智能体：每次选择的动作都能最大化性能度量
  - rational => exploration、learning、autonomy
- PEAS：以出租车为例
- 任务环境的性质：书p39，例子见ppt16
- Agent=体系结构+程序
- 简单反射Agent
- 基于模型的反射Agent
- 基于目标的Agent
- 基于效用的Agent
- 学习Agent

## 第3章 通过搜索进行问题求解

---

### 3.1.1 良定义的问题及解

- 一个问题的5部分形式化
  - 初始状态
  - 可能的行动
  - 转移模型
  - 目标测试
  - 路径耗散

### 3.1.2 问题的形式化

## 3.2 问题实例

### 3.2.1 玩具问题

- 真空吸尘器世界、八数码、N皇后、数字生成问题
- 多用来作为算法性能的基准测试与比较

### 3.2.2 现实世界问题

## 3.3 通过搜索求解

- 搜索树、结点、扩展、叶结点结合(边缘/开结点表)、搜索策略
- 冗余状态(如有环路)的情况的讨论：有的可减少、有的不可避免
  - 考虑探索集(closed 表)：记录每个已扩展过的结点

### 3.3.1 搜索算法基础

- tree-search的基础概念
  - 树中每个结点都含有：State、Parent、Action、Path-Cost
- 算法见ppt p32-34

### 3.3.2 问题求解算法的性能

- 完备性:问题有解时，算法是否能保证找到解？
- 最优性：搜索策略是否能找到最优解？
- 时间复杂度：搜索过程中产生的节点数目来度量
- 空间复杂度：内存中存储的最多结点数来度量
- 由于状态空间图多为无限的(AI领域)，所以复杂度由
  - 分支因子b（任何结点的最多后继数）
  - d：目标结点所在的最浅深度
  - m：状态空间中任何路径的最大长度

来决定

## 3.4 无信息搜索策略

### 3.4.1 宽度优先搜索

- 性能：完备、不一定最优（仅当路径代价是基于结点深度的非递减函数），  
时间、空间： $O(b^d)$

### 3.4.2一致代价搜索

- 特点：目标检测应用于结点被选择扩展时
- 完备、最优
- 例子：[https://blog.csdn.net/weixin\\_45021364/article/details/109746913](https://blog.csdn.net/weixin_45021364/article/details/109746913)

### 3.4.3 深度优先搜索

- 性能参考p77-p78
- 回溯搜索
- 优势在于空间复杂度，但时间复杂度不行

### 3.4.4 深度受限搜索

- 状态空间的直径
- 深度的选择
- 为啥要额外有一个failure ?

### 3.4.5 迭代加深的深度优先搜索 (IDS)

- 不断递增深度界限，反复调用深度受限搜索
- 状态的重复生成问题：先BFS再对边缘集进行IDS
- 迭代加长搜索(3.4.2+DFS)：ILS，通过不断增加的路径代价界限来代替增加的深度界限

### 3.4.6 双向搜索

### 3.4.7 无信息搜索策略对比

- 图3.21

### 重复状态问题

### 图搜索与树搜索 (ch4 ppt 70也有)

- jugs问题 (ppt78-79)：最终在4L罐子里放2L水  
直接搜索即可，并注意去重

## Ch4ppt

## 3.5 有信息(启发式)的搜索策略

- 最佳优先搜索：结点基于评价函数 $f(n)$ 被选择扩展，对 $f$ 的选择决定了搜索策略  
如选择 $h(n)$ ：结点 $n$ 到目标结点的最小代价路径的代价估计值

### 3.5.1 贪婪最佳优先搜索

- $f(n) = h(n)$ ,  $h(n)$ 为直线距离启发式 (在罗马尼亚问题)
- 不完备(有限状态空间的图搜索版本是完备的，但无限的不是)、非最优、时空复杂度：书p83顶部

### 3.5.2 A\*搜索：缩小总评估代价

- $f(n) = g(n) + h(n)$ ,  $g(n)$ 为从开始结点到结点 $n$ 的路径代价
- 若 $h(n)$ 满足特定条件，则算法完备且最优
  - 保证最优性：可采纳性和一致性
    - $h(n)$ 为可采纳启发式：其从不会过高估计到达目标的代价
    - 一致性(单调性)：需满足三角不等式

- 结论:  $h(n)$ 可采纳, 则A\*的树搜索版本为最优; 若 $h(n)$ 一致, 则图搜索A\*算法最优
  - 一个简单的证明: ch4 ppt30, 即证明A\* 在可采纳式 $h(n)$ 下会选择最优
- A\*算法对于任何给定的一致启发式函数都是效率最优的

### 3.5.3 存储受限的启发式搜索

## 3.6 启发式函数 ppt 35

- 8数码问题里可采纳的启发式函数 $h(n)$ 的两种选择
- 启发式的精确度对性能的影响
  - 有效分支因子  $b^*$ , 越接近1越好
  - 一般选择值较大的启发式函数
  - 统治、占优的启发式函数
  - 设置启发式函数本身也有代价 (ppt38)
- 从松弛问题出发设计可采纳的启发式
  - 减少原问题的行动限制得到的新问题称为松弛问题
  - 松弛问题的最优解代价是原问题的可采纳、一致的启发式
  - 但生成的松弛问题本质上要不用搜索就可以求解, 即求解简单
  - $h(n)=\max\{\}$ 的方式来整合多个互不占优的松弛问题最优解
  - 这样的 $h(n)$ 也是可采纳、一致的

## 第4章 超越经典搜索

---

### 4.1 局部搜索算法和最优化问题

- 对于状态空间很大的情况, A\*搜索可能不太适用
- 局部搜索算法一般只用很少的内存, 并且能在很大的状态空间里找到合理的解; 当然这类问题只关心解, 而不关心路径 (如最优化问题)

#### 爬山法

- 不完备
- 以n皇后问题为例
- 爬山法也叫贪婪局部搜索, 存在的问题:
  - 局部极大值
  - 山脊
  - 高原、山肩
- 解决方式
  - 随机爬山法、首选爬山法、随机重启爬山法

#### 模拟退火搜索

- 即允许下山的随机爬山算法

## 局部剪枝搜索

- k个随机生成的初始状态，分别往下走一步（生成各自的后继节点集），从整个大集合里再选k个最佳后继作为新的k个状态
- 改进：随机束搜索

## 遗传算法

- 书p110-112

## 第5章 对抗搜索（Game Playing）

---

### 5.1 博弈

- 零和游戏
- 游戏分类：四类，是否能观察到完整信息、是否是确定性 ppt7

### 5.2 博弈中的优化决策

- 极小极大算法
  - 考虑双方都做最优决策的情况
  - MINIMAX函数：书p139，注意例子(图5.2)
  - 算法：书p140，时间、空间复杂度分析

算法主要就是自上而下前进到叶结点，然后再递归回溯得到各个结点的极小极大值，相当于做了DFS
  - 算法的性能ppt18：
- $\alpha - \beta$ 剪枝
  - 不需要遍历所有结点就可计算出正确的极小极大值
  - 例子见书p142； $\alpha$ 和 $\beta$ 的含义见书p143
  - 有效性分析：ppt27, 28

### 5.4 不完美的实时决策

- 加入截断测试和启发式评估函数(代替棋局效用值)
- 如何设计评估函数：加权线性函数，把多个棋局特征按重要程度整合起来；缺陷：实际上也存在非线性特征组合
  - 根据领域知识设计特征和权重
  - 已知特征，也可用机器学习来学习好的权重
- 棋局效用值满足单调性即可，不用准确估计，所以评估函数没那么严格的限制：ppt33
- 当存在时间限制时：先跑一个baseline，有多余时间再接着跑更高级的，ppt34-35

## PPT：p39: AlphaGo的神经网络

### 5.5 随机博弈

- 如加入掷骰子，所以博弈树里除了MAX和MIN结点外还应该包括机会结点
- 因为存在概率，所以只能计算棋局的期望值，无明确的极小极大值
- 计算公式见书p150
- 此时lookahead的深度反而不宜过深，应做限制； $\alpha - \beta$ 剪枝也不能用，分析见书p151
- 评价函数应该是棋局的期望效用值的正线性变换

## 5.6 部分可观察的博弈

# 第6章 约束满足问题 (CSP)

- 第3,4章讨论搜索状态空间进行问题求解，其中每个状态都是一个黑盒子，不会考虑内部结构。现在CSP问题用成分来描述状态：每个状态对应一组变量的一个特定取值，问题得到解决==每个变量的赋值同时满足所有关于变量的约束
- CSP里利用了状态结构的优势，使用的是通用策略而非问题的专用启发式

## 6.1 定义约束满足问题

- 变量集合X、值域集合D、约束集合C
- 相容、合法的赋值；完整赋值；部分赋值
- 地图着色问题：约束图；为何将问题形式化为CSP：书p170倒数第2段(6.1.2前)
- CSP的形式化：
  - 变量离散
    - 有限值域：可枚举
    - 无限值域：需要约束语言；当为线性约束时问题可解，否则很难
  - 变量连续
    - 采用线性规划方法可在多项式时间里求解
  - CSP里约束的类型
    - 一元约束、二元约束、高阶约束、全局约束
    - 绝对约束与偏好约束, 约束优化问题COP
- 密码算术谜题：书p172
- 转换为二元约束的方法：书p173

## 6.2 约束传播：CSP中的推理

- CSP里两种动作：搜索、约束传播；核心思想是局部相容性
- 弧相容：p174  
最流行的弧相容算法 AC-3,复杂度分析p175顶  
通用弧相容：二元约束扩充到n元约束

## 6.3 CSP的回溯搜索

- 部分赋值的回溯搜索算法
- 一般的DFS经过CSP的可交换性可优化为 $d^n$
- 回溯搜索用于DFS中，算法见p179图
- 对于算法的优化
  - 总体准则：减少搜索空间，提高效率
  - 下一步给哪个变量赋值？
    - **最少剩余值(MRV)**启发式：选择合法取值最少的变量
    - 度启发式：选择与其他未赋值变量约束最多的变量来进行赋值，多用于初始选择/打破僵局
    - 如此来有效剪枝
  - 按什么顺序尝试要赋的值？

- **最少约束值**启发式：优先选择的值能给邻居留下更多选择
 

此法仅在 只需找到一个解时有用；当要求出所有解时则没有意义，因为所有可能都要尝试
- 能检测到不可避免的失败吗，即每步搜索应做怎样的推理inference
  - **前向检验**(一种推理形式)：对被赋值的变量X作弧相容检查(针对未赋值变量Y)
 

但问题在于向前看得不够远（相比于AC-3），所以无法检测出所有不相容
  - 所以换为MAC方法（本质是用AC-3）
- 如何利用问题的结构：6.5节

## 6.4 CSP局部搜索

- 初始状态是给每个变量赋一个值，搜索过程是一次改变一个变量的取值
- 4.1节里所有局部搜索技术都可应用于此
- 每次改变的启发式：最少冲突
- 但这种方式容易造成一系列高原,解决方式
  - 高原搜索
  - 禁忌搜索
  - 模拟退火：ppt50
- ppt48
- 约束加权技术:ppt51

## 6.5 问题的结构

- 利用图的结构来简化问题：如在约束图里寻找连通子图来把原问题分解为多个子问题
- 当约束图形成树时：拓扑排序后在 $O(n)$ 步内将其改为直接弧相容( $n \rightarrow 2$ )，此后从父节点开始依次往后赋值（ $1 \rightarrow n$ ），无序回溯
- 对更一般的约束图，尝试简化为树
  - 基于删除节点：对部分变量先赋值，使得剩下的变量形成一棵树  $\rightarrow$  割集调整法(一般算法见p186)
  - 基于合并结点

# 第7章 逻辑Agent

---

## 7.1 基于知识的Agent

- 知识库KB，公理，背景知识
- 三步走：TELL-》ASK-》TELL，代码见书p198
- 知识层、实现层

## 7.2 Wumpus世界

- 一个具体实例
- 实例性质分析：ppt20



## 7.3 逻辑书202-204

- 语法、语义、真值、模型
- $m$ 是 $\alpha$ 的一个模型，表示语句 $\alpha$ 在模型 $m$ 中为真 或  $m$ 满足 $\alpha$   
 $M(\alpha)$ : 表示所有满足 $\alpha$ 的模型
- 蕴含:  $\alpha \models \beta$ 当且仅当在使 $\alpha$ 为真的每个模型里,  $\beta$ 也为真
- 逻辑推理
- 模型检验: 通过枚举所有可能的模型来检验KB为真的条件下 $\alpha$ 是否都为真
- 推理算法的 (ppt 41)
  - 可靠性(真值保持的): 只导出语义蕴含句
  - 完备性: 可以生成任一蕴含句
- 命题逻辑不满足完备性???

## 7.4 命题逻辑

- 语法:
  - 原子语句、命题词、复合句
  - 5种逻辑连接词, 运算优先级
- 语义:
  - 真值、真值表、模型
  - 复合语句的真值
- 一个简单的知识库
- 简单推理过程
  - 枚举: 列举所有模型, 判断在KB为真的模型里, 语句 $\alpha$ 是不是都为真, 从而得到蕴含关系  $\rightarrow$  模型检验的方法
  - 算法书p208: 此算法可靠且完备

## 7.5 命题逻辑定理证明

### 7.5.1, 7.5.2

- 逻辑等价
- 图7.11: 各种逻辑等价的式子
- 有效性(重言式)、演绎定理(对于任意语句 $\alpha$ 和 $\beta$ ,  $\alpha \models \beta$ 当且仅当语句 $\alpha \Rightarrow \beta$ 是有效的)、可满足性(语句在某些模型里为真)
  - $\alpha$ 有效当且仅当 非 $\alpha$ 不可满足
  - p210顶部公式(归谬、反证、矛盾法)
- 推导和证明
  - 假言推理规则
  - 消去合取词
  - 分离规则: ppt54  
**Modus Ponens (分离规则, 肯定前件的假言推理) (for Horn Form):**  
complete for Horn KBs

$$\frac{\alpha_1, \dots, \alpha_n, \quad \alpha_1 \wedge \dots \wedge \alpha_n \Rightarrow \beta}{\beta}$$

- 所有逻辑等价都可以作为推理规则

- 证明问题的四个定义：初始状态、行动、结果、目标
  - 逻辑系统的单调性
- 归结证明
  - 单元归结、全归结、单元语句、互补文字、归并
  - 子句：文字的析取式；
 

文字：一个命题符号，或加否定的命题符号，(正文字、负文字)
  - 命题逻辑里的归结规则是可靠和完备的，见书p212底部两段
- 合取范式 (CNF)
- 归结算法：书p213-214
- 归结的完备性：基本归结定理（如果子句集是不可满足的，那么这些子句的归结闭包包含空子句）

### 7.5.3

- Horn子句：至多只有一个正文字的析取式；
  - 目标子句
  - Horn子句在归结下是封闭的
  - Horn子句判定蕴含需要的时间与知识库大小成线性关系
  - ppt p54也有定义
  - 使用Horn子句的推理可以使用前向链接和反向链接算法
- 限定子句：恰好只含一个正文字的析取式
  - 每个限定子句都可以写成蕴含式

### 7.5.4

- 前向连接：书p216算法
  - 运行时间为线性
  - 与或图例子
  - 可靠性：每个推理本质上是假言推理规则的一个应用
  - 完备性：每个被蕴含的原子语句都将得以生成（证明见ppt66）
- 反向链接：ppt67起
- 前向与反向链接的比较：ppt p79

### PPT

- 前向/反向链接算法和归结算法的比较以及时间复杂度：ppt85-88

## 第8章 一阶逻辑

---

### 8.1.2 结合形式语言和自然语言优势

- 参考ppt
- 命题逻辑的优点和缺点：ppt 4-6
- 一阶逻辑的新东西：对象、关系、函数
 

谓词：用来描述个体之间的关系或属性
- 各种逻辑的比较

## 8.2 一阶逻辑的语法和语义

- 一阶逻辑的模型
  - 模型的域是它所包含的对象或域元素的集合
  - 一阶逻辑中的模型要求全函数：即每个输入元组必须有一个结果值
- 符号和解释：
  - 常量符号(对象)、谓词符号(关系)、函词(函数)、变量、联结词、等词、量词
  - 元数：确定参数个数
  - 解释：常量、谓词、函词对应现实里对象、关系、函数的一个映射
- 由于可能模型的数量是无限的，通过枚举所有可能模型以检验蕴含在一阶逻辑中不可行
- 项：指代对象的逻辑表达式
  - 常量符号是项
  - 复合项由函词及紧随其后的参数、被括号括起来的列表项组成
  - 变量
  - 无变量的项称为基项
- 原子语句：由谓词符号及随后被括号括起来的列表项(包括复合项)组成

如果谓词所指代的关系在参数所指代的对象中成立，那么原子语句在给定模型、给定解释下为真
- 复合语句：原子语句+逻辑联结词
- 量词：
  - 全称量词：多与蕴含词一起用
  - 存在量词：多与合取词连用
  - 嵌套量词
  - 全称和存在连词间的互相转换
- 等词：声明两个项指代同一个对象
- 另一种语义

## 8.3 运用一阶逻辑

- 一阶逻辑的断言和查询：置换和绑定表
- 亲属关系论域：定理与公理；定义与基本谓词集合
- 数、集合和表
- Wumpus世界：感知、反射行为、环境等的表示

## 8.4 一阶逻辑的知识工程

- 介绍 知识库构造的一般过程：称为知识工程。
- 具体过程：
  - 确定任务
  - 搜集相关知识
  - 确定词汇表、包括谓词、函词和常量：本体论
  - 对领域通用知识编码：写出公理
  - 对特定问题实例描述编码
  - 把查询提交给推理过程并获取答案
  - 知识库调试
- 电路领域具体实例

# 第9章 一阶逻辑的推理

---

## 9.1 命题推理与一阶推理

- 全称量词实例化(UI): 置换的概念; 可多次应用而获得不同结果
- 存在量词实例化: 用到新的常量符号, 称为Skolem常数; 只可应用一次
- 由上述两种实例化, 可以从存在量词的语句推导出含量词的语句, 这样可以简化到命题逻辑, 然后利用已知的命题推理 (但缺陷: 知识库包含函词时, 可能的基项置换集是无限的, 书270顶部)
- 但一阶逻辑的蕴涵问题是半可判定的: ppt 8-9

## 9.2 合一和提升

- 按9.1节而言, 确实推导很有效, 但是可能会产生大量无关的命题
- 一般化假言推理规则(GMP): 书p271,ppt19
  - 只能用在只含限定子句的知识库里 (即每个句子只有一个正文字)
  - 充要性证明: ppt 21-22
- 最一般合一置换; 标准化分离: 书p270

## 9.3 前向链接

- 一阶确定子句: 书p274-275 (会省略全称量词)
  - 此知识库不包含函词: 是数据日志类知识库的一个实例
- 前向链接算法: ppt31, 例子见ppt33
- 性质: ppt 34
- 效率
- 对只含Horn子句的知识库是完备的

## 9.4 反向链接

- 算法: ppt36
- 性质: ppt45, 算是一种深度优先搜索
- 可用于逻辑程序设计
- 对只含Horn子句的知识库是完备的

## 9.5 归结

- 一阶逻辑的合取范式: 一阶逻辑里语句转换为合取范式(CNF)的方法: 书p287
- 归结推理规则
  - 归结算法:  $KB \models \alpha \iff KB \cap \neg \alpha$ 是不可满足的  
具体只需把 $KB \cap \neg \alpha$ 转换为合取范式, 然后不断应用归结规则, 若推出空集, 则说明是不可满足, 从而证明 $KB \models \alpha$
  - 归结规则: 书p288

# 第13章 不确定性

---

## 13.1 不确定性

- 决策理论=概率理论+效用理论

## 13.2 基本概率符号

- 随机变量、事件、命题
  - 原子事件：ppt 16
- 联合概率分布、边缘概率分布、概率分布、概率密度函数
- 无条件概率/先验概率；条件概率/后验概率（乘法规则、链式法则：ppt 23）

## 13.3 使用完全联合分布进行推理

- 归一化常数
- 查询变量、证据变量、未观测变量（ppt 30）

## 13.4 独立性及贝叶斯规则

- 定义
- 贝叶斯规则：书p414 式13.12, 13.13
  - 简单应用：书p415（涉及归一化常数）
- 条件独立性：ppt34, 书p416
- 朴素贝叶斯：书p417

# 第14章 概率推理（贝叶斯网络）

---

## 14.1 不确定性问题域中的知识表示

- 贝叶斯网络的定义：拓扑结构
- CPT（条件概率表）：例子：书p428
- 紧致性：ppt18

## 14.2 贝叶斯网络的语义

- 全局语义：书p429中间计算例子
- 局部语义：给定父节点，一个结点与它的非后代节点是条件独立的
- 贝叶斯常用结构：ppt22-24
- 将贝叶斯网络视为对联合概率分布的表示，用以构造贝叶斯网络：链式规则、偏序一致；构造出来的网络是无环的，且无冗余概率值。条件概率表CPT
  - 贝叶斯网络的紧致性与结点排序：ppt26 **添加节点的正确次序是首先添加“根本原因”节点，然后加入受它们直接影响的变量，以此类推**
  - 贝叶斯网络不一定反应因果，而是反应关联：ppt33
- 将其视为对一组条件依赖性语句的编码，用以设计推理过程
  - 条件独立性基本结构示例：ppt22-24
  - 因果关系：ppt33

## 14.4 贝叶斯网络中的精确推理

- 给定一组证据变量（已观测），计算一组查询变量的后验概率分布；隐藏变量
- 通过枚举进行推理：存在重复计算。是通过深度优先递归树，复杂度：ppt37-38
- 变量消元算法：逐点相乘；
  - 因子上的操作
  - 变量顺序和变量相关性
- 精确推理的复杂度：

- 单连通网络/多形树：时间、空间都为线性规模
  - 多连通网络：最坏为指数量级
- 朴素贝叶斯模型
  - 垃圾邮件检测
  - 文本分类
  - 数字识别

## Ch18 Learning-1

---

### Introduction to ML

- ML分类：监督学习、无监督学习、半监督学习
- 文本表示:vector
- 分类问题、回归问题

## ch18 Learning-2

---

### 监督学习

#### 决策树

- 关键：每次选择最有区分度的属性来划分
  - 从信息论角度出发，计算参考西瓜书
- 优点和缺点：ppt17

#### 最近邻模型

- 算法的影响因素：K值；距离指标

#### 线性预测

- 分类或回归问题
- 目标函数：最小化损失，如最小二乘损失
- **最小二乘分类及其损失函数计算**：ppt28-30
  - 多分类问题：ppt47-48
 

此时的计算实际类似于二分类
  - 最小平方损失的优缺点：ppt49
- 回归问题：曲线拟合（n次函数）

### 常用概念

- 训练误差、测试误差、泛化误差
- 过拟合、欠拟合
- 模型复杂度：奥卡姆剃刀原则
- regularization(规范化、正则化)：ppt45-46
  - L1正则：多为稀疏解，可用于过滤无关特征
  - L2正则：易优化，多为闭式解
- 交叉验证：ppt50及之后

- 机器学习完整过程：ppt56

## ch20 SVM

---

- 计算均参考西瓜书
- 哪个分类平面最优？
- 间距（margin）、支持向量、超平面
  - 最大间距分类
  - 问题定义
- 凸二次规划问题—> 对偶问题
  - 拉格朗日函数
  - SVM的分割平面只由支持向量决定，只有支持向量对应的 $\alpha_i \neq 0$
- **SVM总结：ppt23**
- 软间隔SVM
- 另一个角度看SVM：从最小化损失看（前面是从最大化间隔看）
- 非线性SVM：核函数
  - 核矩阵的性质：ppt48

## ch21 Logistic regression

---

- sigmoid函数
- 极大似然法：目标即最大化ppt6-7的函数
  - 梯度下降法

## ch22 神经网络

---

- MP神经元模型
- 激活函数
- 神经网络例子
- 学习网络参数：采样->前向计算->反向传播->更新参数
  - 反向传播的计算：ppt20
- 多层前馈网络：ppt24
  - 表示能力
  - 局限
  - 缓解过拟合的策略
- 预训练+微调
- 共享参数
- CNN
  - 一般要多个卷积核来提取不同特征
  - CNN与全连接网络的优缺点比较：ppt38, ppt42
  - max pooling:目的是减少参数或者数据
  - 激活函数：sigmoid、Tanh、ReLU

- 一些评论: ppt55-56
- DL和ML的区别: ppt57-58

## ch23 无监督学习

---

- 常见方法: PCA、流形学习、聚类、密度估计

### PCA

- 降维的原因: ppt15-17
  - 维度灾难: 准确度和效率
  - 只有少部分维度实际有用
  - visualization、DataCompression、Noise removal
- PCA目标: 降维, 并尽可能保留原数据的大部分信息。新的变量各个维度是相互独立的(PCs), 并且根据重要性排序
- 投影后方差最大/投影后垂直距离之和最小

寻找的准则: 与前面所找主元正交, 且该方向投影后能保留尽可能多的信息
- 数学推导: ppt26-29
  - 即是找协方差矩阵的特征向量和特征值
  - 具体求解方法: ppt30
- 若P是正交阵, 那么投影后并没降维, 只是换了投影方向, 此时无信息损失
- 若P不是full, 那么从y复原x会丢失部分信息
- 从另一个角度的数学推导: ppt33
  - 最小化信息损失
- 非线性PCA: 对式子变式后引入核函数

### 聚类

- K-means: 最小化距离平方和

目标函数: ppt43-44

需要同时确定每个簇的中心以及每个样本属于哪个簇

  - 交替优化策略
- 算法为何可以收敛: ppt45
- 数学形式: ppt46
- k-means的优缺点: ppt48