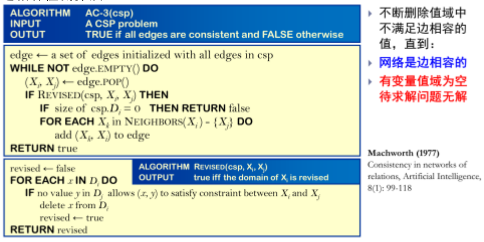
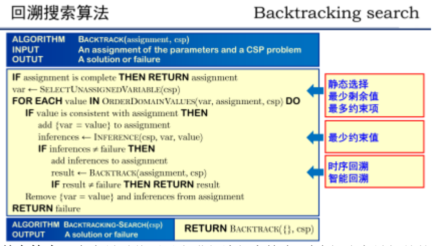
序言：如果这份 CheatSheet 流传到了十年后，那我和清华本科教育中一定有一个出了很大的问题.
高嘉伟 于 2023 年春

搜索

状态空间法(S,F,G)(初始状态 操作符 目标状态)
如食人生番:动作 op(b1,b2,b3)(人数,野人数,哪岸到哪岸)
状态(a1,a2,a3)(左岸人 num,左岸野人 num,船在左 or 右)
状态转换: $A \to B$，$OPERATOR$(操作对象,初值,终值)
（状态集，行动集，代价函数）描述了该问题
搜索:
宽度:Open 中 pop 出节点,将节点放入 closed 集,对每个子节点,首先判断它不在 closed 和 open 里,并判断是否达成目标,若不是目标,则将子节点放入 open
开节点表采用队列,先进先出
宽度优先搜索能找到解,宽度优先搜索能找到最优解.
设每个节点扩展出 b 个节点,目标位于第 d 层的最后一个节点,则一共扩展出的节点为$O(b^d)$,时间效率为指数复杂度,空间效率为指数复杂度.
一致代价搜索: $g(n_{i+1}) = g(n_i) + c(n_i, n_{i+1})$ 每次从 open 表中选 g 最小的节点，判断该点是否为目标，然后放进 closed 表.对该节点的每一个子节点,如果不在 open 或 closed,则放入 open 表;而如果该子节点在 open 表中,但是 cost 更大,则更新开节点表(replace that node with child).
开节点表为优先队列.
一致代价搜索算法能找到解:如果单步代价非负,且不存在代价为零的环路.一致代价搜索能找到最优解
Dijkstra 算法:权重不能为负.
设最优路径代价为C^*,每一步行动代价至少为ε.则一共扩展出的节点为$O(b^{1+\lfloor C^*/\varepsilon \rfloor})$.时间效率为指数复杂度,空间效率为指数复杂度.
深度优先:一般实现为树搜索,不使用闭节点表.首先扩展初始状态,开节点表实现为堆栈,Open 中 pop 出节点,判断是否是目标,若不是,则对每个子节点,将其放入 open 表.
递归深度优先搜索:对每个子节点使用深度优先搜索算法.
如果状态空间没有环路（如一棵树）深度优先搜索能找到解,加入环路检查可以避免循环.深度优先搜索不能保证找到最优解
设每个节点扩展出 b 个节点,搜索到第 m 层的最后一个节点时发现目标.时间效率为指数复杂度:共扩展出$O(b^m)$个节点.空间复杂度为线性复杂度,$O(bm)$
深度受限搜索:pop 出节点后检查是否为目标,是否达到了深度上限.
不论状态空间是否有环路,深度限制小于解的深度时找不到解,大于解的深度时能找到解.对于环路,深度限制避免了形成高环路环路.深度受限搜索不能保证找到最优解.
时间效率为指数复杂度，$O(b^l)$，空间效率为线性复杂度$O(bl)$
迭代加深搜索:不断增加深度限制（从 0 开始，依次加 1）.
迭代加深搜索能找到解,迭代加深搜索能找到最优解.
时间效率为指数复杂度 $mb + (m-1)b^2 + \cdots + b^m = O(b^m)$ 与宽度优先类似
空间效率为线性复杂度 $b + b + \cdots + b = O(bm)$ 与深度优先类似
启发函数:对问题定义之外信息的利用. 对当前状态最小代价的一个估计.
$h(n)$ 越小表示 n 越接近目标
贪婪最佳优先搜索：把一致代价搜索算法的代价函数换为启发函数.
具有完备性，不能保证找到问题的最优解.
时间指数复杂度$O((b^*)^m)$，空间指数复杂度 $O((b^*)^m)$
A*算法:把一致代价搜索算法的代价函数换为路径代价与启发函数的和.
如果 h(n)是**可采纳**的,则树搜索 A 星算法是最优的.可采纳:启发函数不会高估到达目标的代价.A 星算法永远不会扩展次优目标节点.
如果 h(n)是**一致**的,则图搜索 A 星算法是最优的.对每个节点 n 及其任意后继节点n',满足$h(n) \le c(n,n') + h(n')$.一致的启发函数满足三角不等式.
图搜索 A 星实际上是在剪枝.时间效率：指数复杂度.空间效率：指数复杂度.
两个启发函数，如果对于任意节点 n 均满足$h_1(n) \le h_2(n)$,则后者占优
松弛问题:以放宽条件的问题的精确解作为原问题的启发函数
启发函数的合成$h(n) = max\{h_1(n), \dots, h_k(n)\}$
注意 h 是启发函数, g 是路径代价, f 是最终结果.
计算复杂性$f(x) = O(g(x)) \Leftrightarrow \exists A, B > 0, A \le |f/g| \le B$
P 类问题:有多项式时间算法解决的**判定问题**(有 $P \subseteq NP$)，后者尚不清楚.
NP 类问题:对猜想存在多项式时间算法来验证的判定问题. NP 类问题是指可以通过非确定性多项式算法求解的判定问题.
两个判定问题 A 和 B，如果 A 的每一个实例都能通过一个函数转化为 B 的一个实例，则称 A 可归约为 B.如果该函数能够在多项式时间完成归约，则称 A 多项式时间可归约为 B.A：求解一元一次方程.B：求解一元二次方程.
NP 完全问题:(NP 中最难的问题)如果任何一个 NP 类问题都能在多项式时间归约为一个特定的 NP 类问题，则称该问题是一个 NP 完全问题.该问题自己属于 NP 类.NP 类中任何一个问题都能在多项式时间归约为该问题.NP 类中任何一个问题都不比该问题难.所有 NP 完全问题之间可以进行多项式时间归约
NP 难题:如果任何一个 NP 类问题都能在多项式时间归约为一个特定的问题，则称该问题是一个 NP 难题.该问题自己不一定属于 NP 类.
约束满足：用一组变量来描述状态，变量还有一定的约束(取值范围+取值有约束).
八皇后问题：变量$\mathbf{X} = \{X_1, \dots, X_n\}, n = 8$. 值域：$\mathbf{D} = \{D_1, \dots, D_n\}, X_i \in D_i = \{1, \dots, n\}$. 约束：$\mathbf{C} = \left\{\left((x_i, x_j), x_i \ne x_j\right), \left((x_i, x_j), |x_i - x_j| \ne |i - j|\right)\right\}$.
一元约束，二元约束，全局约束.
回溯搜索：每次选择一个变量来赋值. 取合法值时从该状态继续为下一变量赋值，没有合法值时返回父状态尝试其新的赋值.
变量选择的顺序:最少剩余值：选择值域最小的变量；最多约束项：选择约束最多的变量. 失败优先原则
取值选择的顺序：最少约束值：选择给邻居留下更多可能的值
如果一个变量在其值域中的所有取值均满足其一元约束，则称该变量是节点相容的. 如果一个变量在其值域中的所有取值均满足与该变量相关的二元约束，则称该变量是边相容的.
边相容检测算法: AC-3



如果对任何 k-1 个变量的相容赋值，第 k 个变量总能被赋予一个和它们相容的值，则称这 k-1 个变量对于第 k 个变量是 k 相容的.
3-相容就是路径相容
搜索后推理:每次决定某变量的取值后，推理其邻居变量的值域.



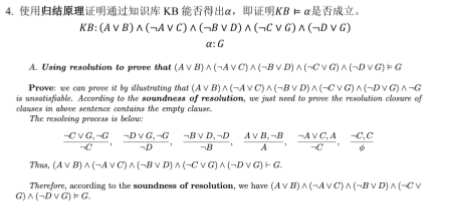
前向检查：在变量赋值后对它进行边相容检查,删除与该变量相关的未赋值变量不相容的值.
时序回溯：需要回溯时，退回到前一个变量，尝试其下一个取值.
智能回溯：需要回溯时，退回到产生冲突的变量,
对抗搜索:搜索的主体多于一个
极大极小搜索： 生成树：通过搜索算法生成完整的博弈树. 倒推值：自底向上逐级计算非叶节点的效用值.
棋手：MAX 就是自己，MIN 就是对手.一层 MAX，一层 MIN，构成一个回合.MAX 层节点的子节点之间是或的关系.MIN 层节点的子节点之间是与的关系.效用：终止状态下对局者的收益.
截断的极小极大搜索:通过搜索算法生成一定深度的博弈树
效率:检查$O(b^m)$个节点.
Alpha-Beta 剪枝:α为最大下界,β最小上界
极大节点的 Alpha 剪枝:MAX $(x_1, x_n, \dots, x_n, \boldsymbol{\alpha})$则小于等于$\boldsymbol{\alpha}$的 x 值都不用考虑.a 剪枝用于极大层，裁剪某节点的后继兄弟节点.当该节点的估值小于等于父节点的 a 值时可以裁剪该节点的后继兄弟节点
极小节点的 Beta 剪枝:MIN $(x_1, x_2, \dots, x_n, \boldsymbol{\beta})$.b 剪枝用于极小层，裁剪某节点的后继兄弟节点.当该节点的估值大于等于父节点的 b 值时可以裁剪该节点的后继兄弟节点.
效率:检查$O(b^{m/2})$个节点.
蒙特卡洛树搜索:
Upper confidence bounds applied to trees :U(n)为节点 n 的总仿真效用,N(n)为节点 n 的总仿真次数,C 为平衡利用与开发的常数.

$$UCB1(n) = \frac{U(n)}{N(n)} + C \times \sqrt{\frac{\log N(\text{PARENT}(n))}{N(n)}}$$

流程:选择-扩展-仿真-回传
随机博弈:除极小极大层外，还需增加随机层.不确定性使得需要考虑的可能性急剧增多

推理

知识库是语句的集合.
命题逻辑研究事实是否成立. **原子命题**：对单个事实的陈述.
5 种连接符: ¬: 否定∧:合取(与)∨:析取(或)→:蕴含↔:等价
$p \to q$ 为假当且仅当 $p = T, q = F$
$\alpha \vDash \beta$: α蕴含β，如果在所有使语句 a 为真的模型中，语句 b 也为真.
通过模型检查进行逻辑推理：本质上是一个回溯搜索算法.时间复杂度：指数.空间复杂度：线性.
永真/重言式:真；永假/矛盾式:假；可满足式:至少一个成真赋值；非重言式的可满足式:至少一个成真一个成假
原子公式:不含任何联结词的公式//子句:任何文字的析取式;文字:原子或~原子//合取**范式**:简单析取式构成的合取式//析取范式:简单合取式构成的析取式
等价演算
吸收律:$A \lor (A \land B) \Leftrightarrow A; A \land (A \lor B) \Leftrightarrow A$
分配律: $A \lor (B \land C) \Leftrightarrow (A \lor B) \land (A \lor C)$; $A \land (B \lor C) \Leftrightarrow (A \land B) \lor (A \land C)$
摩根律:$\neg(A \land B) \equiv \neg A \lor \neg B$; $\neg(A \lor B) \equiv \neg A \land \neg B$
$A \to B \Leftrightarrow \sim A \lor B$
归谬论: $(A \to B) \land (A \to \sim B) \Leftrightarrow \sim A$
假言移位: $A \Rightarrow B \equiv \neg B \Rightarrow \neg A$
等价等值：$A \Leftrightarrow B \equiv (A \Rightarrow B) \land (B \Rightarrow A)$
等价否定：$A \Leftrightarrow B \equiv \neg A \Leftrightarrow \neg B$
有效性：如果一个语句对所有模型都为真，则称该语句是有效的
演绎定理：$\alpha \vDash \beta$ if and only if $\alpha \Rightarrow \beta$ is valid
命题逻辑的推理规则
附加:$A \Rightarrow (A \lor B)$;假言推理:$((A \to B) \land A) \Rightarrow B$;
简化:$(A \land B) \Rightarrow A$;拒取式:$((A \to B) \land \sim B) \Rightarrow \sim A$;
析取三段论:$((A \lor B) \land \sim A) \Rightarrow B$;
假言三段论:$((A \to B) \land (B \to C)) \Rightarrow (A \to C)$;
等价三段论:$((A \leftrightarrow B) \land (B \leftrightarrow C)) \Rightarrow (A \leftrightarrow C)$
构造性二难:$(A \to B) \lor (C \to D) \land (A \lor C) \Rightarrow (B \lor D)$
可做:前提引入/结论引入/置换规则比如$\sim p \lor q$换$p \to q$
命题逻辑的归结原理：基于归谬法
子句:简单析取式,项是一个变量或者其否定。子句集:合取范式中所有子句的集合,∧换成“,”。
限定子句：恰好只有一个正文字的析取式.$(\neg A \lor \neg B \lor C) \equiv (A \land B) \Rightarrow C$
目标子句：不包含正文字的析取式.
仅包含限定子句的知识库:可解释性强,推理方法好,推理效率高.
前向链接：在限定子句上使用假言推理，从事实出发，匹配规则，产生事实，直到结论
后向链接：在限定子句上使用假言推理，从查询出发，匹配规则，倒推事实，直到满足.
语句α有效，当且仅当$\neg\boldsymbol{\alpha}$不可满足：语句α有效,则α总为真,$\neg\alpha$总为假,所以$\neg\boldsymbol{\alpha}$不可满足.
语句α可满足，当且仅当$\neg\boldsymbol{\alpha}$不是有效的：语句α可满足，则不会总为假,则$\neg\boldsymbol{\alpha}$不会总为真,故不是有效的.
归结原理:如 $p \lor q$ 和 $\sim p \lor r$ 都为真，那么 $p \lor r$ 为真。
归结式:$C_1 = P \lor C_1', C_2 = \sim P \lor C_2'$ 归结 $C_{12} = C_1' \lor C_2'$
合取范式：若干子句的合取等值.


4. 使用**归结原理**证明通过知识库 KB 能否得出α，即证明$KB \vDash \alpha$是否成立。

$$KB: (A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G)$$
$$\alpha: G$$

A. ***Using resolution to prove that*** $(A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G) \vDash G$

Prove: *we can prove it by illustrating that* $(A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G) \land \neg G$ *is unsatisfiable. According to the* ***soundness of resolution****, we just need to prove the resolution closure of clauses in above sentence contains the empty clause.*
The resolving process is below:

$$\frac{\neg C \lor G, \neg G}{\neg C}, \quad \frac{\neg D \lor G, \neg G}{\neg D}, \quad \frac{\neg B \lor D, \neg D}{\neg B}, \quad \frac{A \lor B, \neg B}{A}, \quad \frac{\neg A \lor C, A}{\neg C}, \quad \frac{\neg C, C}{\phi}$$

Thus, $(A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G) \vdash G$.

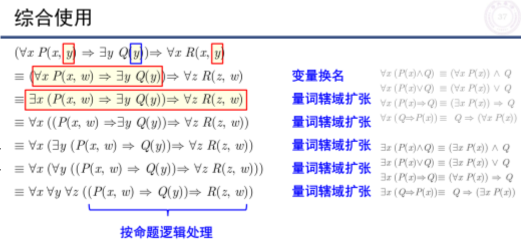
Therefore, according to the **soundness of resolution**, we have $(A \lor B) \land (\neg A \lor C) \land (\neg B \lor D) \land (\neg C \lor G) \land (\neg D \lor G) \vDash G$.

归结（消解）法:为了证明$A \to B$真,先转为$A \land \sim B$,证明该命题公式永假。提取其子句集，归结为空即可。
谓词逻辑:关系是否成立
进行符号和对象、关系、函数之间映射的规则.将常量符号映射到对象,将谓词符号映射到关系,将函数符号映射到函数.
指代对象的表达式称为项.谓词符号作用于项的结果称为原子语句.
¬ Emperor(Gorgon): 多尔衮不是皇帝.
$\forall x$: 全称量词. $\exists x$: 存在量词.
量词嵌套：变量属于引用它的最内层量词，且不再属于其他任何量词.
$\forall x(\text{Crown}(x) \lor (\exists x\ \text{Brother}(\text{Taiji}, x)))$.
嵌套时使用不同的变量: $\forall x(\text{Crown}(x) \lor (\exists y\ \text{Brother}(\text{Taiji}, y)))$
等词:声明两个项指代同一个对象.
换名:辖域中约束出现的变量名换掉/替代:自由出现的个体变量名字可换掉
谓词公式永真称为逻辑有效/永真的，谓词公式永假成为不一致/不可满足。
谓词演算公式
互反性: $\sim(\forall x)P(x) \Leftrightarrow (\exists y)\sim P(y)$.每个人都喜欢吃冰淇淋=没有人不喜欢吃冰淇淋
摩根律$\sim(\exists x)P(x) \Leftrightarrow (\forall y)\sim P(y)$; $\forall x \neg \boldsymbol{P}(x) \equiv \neg\exists x \boldsymbol{P}(x)$;
$\neg\forall x P(x) \equiv \exists x \neg P(x)$.

全称量词辖域收缩与扩张: Q 里不能出现 x
$\forall x(P(x) \land Q) \equiv (\forall x P(x)) \land Q$; $\forall x(P(x) \lor Q) \equiv (\forall x P(x)) \lor Q$
$\forall x(P(x) \Rightarrow Q) \equiv (\exists x P(x)) \Rightarrow Q$; $\forall x(Q \Rightarrow P(x)) \equiv Q \Rightarrow (\forall x P(x))$
证明$\forall x P(x) \equiv P(a) \land P(b) \land P(c) \land \dots$
存在量词辖域收缩与扩张: Q 里不能出现 x
$\exists x(P(x) \land Q) \equiv (\exists x P(x)) \land Q$; $\exists x(P(x) \lor Q) \equiv (\exists x P(x)) \lor Q$
$\exists x(P(x) \Rightarrow Q) \equiv (\forall x P(x)) \Rightarrow Q$; $\exists x(Q \Rightarrow P(x)) \equiv Q \Rightarrow (\exists x P(x))$
全称量词的分配：全称量词可以对合取分配，不能对析取分配
$\forall x(P(x) \land Q(x)) \equiv \forall x P(x) \land \forall x Q(x)$; $\forall x(P(x) \lor Q(x)) \ne \forall x P(x) \lor \forall x Q(x)$
存在量词的分配：存在量词可以对析取分配，不能对合取分配
$\exists x(P(x) \lor Q(x)) \equiv \exists x P(x) \lor \exists x Q(x)$; $\exists x(P(x) \land Q(x)) \ne \exists x P(x) \land \exists x Q(x)$

$\forall x\ (P(x) \land Q) \equiv (\forall x\ P(x)) \land Q$
$\forall x\ (P(x) \lor Q) \equiv (\forall x\ P(x)) \lor Q$
$\forall x\ (P(x) \Rightarrow Q) \equiv (\exists x\ P(x)) \Rightarrow Q$
$\forall x\ (Q \Rightarrow P(x)) \equiv Q \Rightarrow (\forall x\ P(x))$

$\exists x\ (P(x) \land Q) \equiv (\exists x\ P(x)) \land Q$
$\exists x\ (P(x) \lor Q) \equiv (\exists x\ P(x)) \lor Q$
$\exists x\ (P(x) \Rightarrow Q) \equiv (\forall x\ P(x)) \Rightarrow Q$
$\exists x\ (Q \Rightarrow P(x)) \equiv Q \Rightarrow (\exists x\ P(x))$

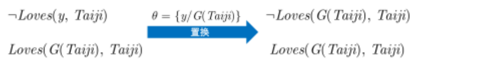


前束范式:语句中所有量词（不含否定词）都在最左边,并且这些量词的辖域都到公式的最末端.前束范式不唯一.
全称量词的实例化：将变量替换成论域中的任意对象. 直接消去全称量词，待需要替换时再考虑置换为哪个对象.
置换：$\theta = \{x/\text{Taiji}\}$. 把 x 换成 Taiji
存在量词的实例化：给这个特定对象起名字$\theta = \{y/C_N\}$. 但是，如果该特定对象被全称量词所管辖,则该对象的起名依赖于管辖它的变量，此时置换应为函数.
合取范式：在前束范式的基础上，消去量词，转化为析取子句的合取形式. 获取合取范式的步骤：1. 等价消去; 2. 隐含消去; 3. 否定深入; 4. 变量换名; 5. 量词前移（量词辖域扩张）6. 量词消去（先消去存在量词，再消去全称量词）7. 分配律

$$\forall x\ [(\forall y\ Person(y) \Rightarrow Loves(x, y)) \Rightarrow (\exists y\ Loves(y, x))]$$

- **等价消去**：$P \Leftrightarrow Q \equiv (P \Rightarrow Q) \land (Q \Rightarrow P)$
- **隐含消去**：$P \Rightarrow Q \equiv \neg P \lor Q$
  $\forall x\ [\neg(\forall y\ \neg Person(y) \lor Loves(x, y)) \lor (\exists y\ Loves(y, x))]$
- **否定深入**：$\neg(\forall y\ P) \equiv \exists y\ \neg P$, $\neg(\exists y\ P) \equiv \forall y\ \neg P$
  $\forall x\ [(\exists y\ Person(y) \land \neg Loves(x, y)) \lor (\exists y\ Loves(y, x))]$
- **变量换名**：
  $\forall x\ [(\exists y\ Person(y) \land \neg Loves(x, y)) \lor (\exists z\ Loves(z, x))]$
- **消去存在量词：斯柯林化**
  $\forall x\ [(Person(F(x)) \land \neg Loves(x, F(x))) \lor Loves(G(x), x)]$
- **消去全称量词：直接消去**
  $(Person(F(x)) \land \neg Loves(x, F(x))) \lor Loves(G(x), x)$
- **分配律**：
  $(Person(F(x)) \lor Loves(G(x), x)) \land (\neg Loves(x, F(x)) \lor Loves(G(x), x))$

谓词逻辑归结原理
合一:使多个语句变得相同的置换称为合一；变量分离：不同语句中的变量实际上是不同的.
归结原理（命题逻辑）：如果$\alpha \lor \beta$和$\neg\alpha \lor \gamma$均为真，则$\boldsymbol{\beta} \lor \gamma$亦为真.
归结原理（谓词逻辑）：如果存在合一置换使得两个子句包含互补的原子语句，则可通过消去该原子语句对这两个子句进行归结.



监督学习:线性回归

线性回归: $y = \omega x + b$ 找函数族/找优化准则/找最优函数
$S_{xx} = \sum (x_i - \bar{x})^2, S_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y}), S_{yy} = \sum (y_i - \bar{y})^2$
最小二乘: $\hat{\omega} = \frac{S_{xy}}{S_{xx}} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2} = \frac{S_{xY}}{S_{xx}}, \hat{b} = \bar{y} - \hat{\omega}\bar{x} = \bar{Y} - \hat{\omega}\bar{x}$
总平方和=回归的平方和+残差的平方和 $\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^n (y_i - \hat{y}_i)^2$
评价指标 $\text{MSE} = \frac{1}{n}\sum_{i=1}^n \left(y_i - (wx_i + b)\right)^2$
最大似然估计: $Y_i \mid x_i \sim N(\omega x_i + b, \sigma^2)$.
概率密度函数:$f(y_i \mid \omega, b, x_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{[y_i - (\omega x_i + b)]^2}{2\sigma^2}\right\}$
确定系数:$r^2 = S_{xy}^2 / (S_{xx} S_{yy})$，越接近 1 拟合效果越好
多项式回归: $\mathrm{y} = \mathrm{Xw}$.
最小化残差平方和$min(\mathbf{y} - \mathbf{Xw})^T(\mathbf{y} - \mathbf{Xw})$
$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial (\mathbf{y} - \mathbf{Xw})^T(\mathbf{y} - \mathbf{Xw})}{\partial \mathbf{w}} = 2\frac{\partial (\mathbf{y} - \mathbf{Xw})^T}{\partial \mathbf{w}}(\mathbf{y} - \mathbf{Xw}) = -2(\mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{Xw})$
解得$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$
多元回归: $y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_m x_m$
正则化: $min(\mathbf{y} - \mathbf{Xw})^T(\mathbf{y} - \mathbf{Xw}) + \lambda f(\mathbf{w})$

监督学习:Logistic 回归

$P(Y = 1) = \pi = \frac{1}{1 + e^{-(wx+b)}}$. $\frac{d}{dx}\pi(x) = w\pi(x)(1 - \pi(x))$

似然函数$P(Y=y)=\pi^y(1-\pi)^{1-y}$.

则$f(\mathbf{y}\mid w,b)=\prod_{i=1}^n f(y_i)=\prod_{i=1}^n \left(\pi(x_i)\right)^{y_i}\left(1-\pi(x_i)\right)^{1-y_i}$

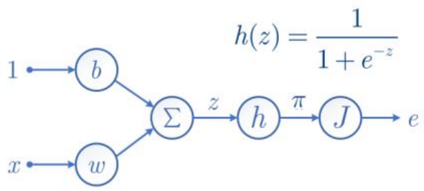
最大似然$max\sum_{i=1}^n \left(y_i\log\pi(x_i)+(1-y_i)\log\left(1-\pi(x_i)\right)\right)$

信息量 $I=-\log_2 p$. 信息熵 $H=-\sum_{i=1}^l p_i\log_2 p_i$

交叉熵 $H(p,q)=-\sum_{i=1}^l p_i\log_2 q_i$

交叉熵损失: $-\frac{1}{n}\sum_{i=1}^n \left(y_i\log\pi(x_i)+(1-y_i)\log\left(1-\pi(x_i)\right)\right)$

其中$\pi(x_i)=\frac{e^{ux_i+b}}{1+e^{wx_i+b}}$. 注意最大化对数似然和最小化交叉熵损失是等价的.



从输入算输出 $z=wx+b,\ \pi=h(z)=\frac{e^z}{1+e^z}=\frac{1}{1+e^{-z}}$,

$$e=J(\pi)=-y\log\pi-(1-y)\log(1-\pi)$$

从输出算梯度:

$\frac{\partial e}{\partial b}=\frac{de}{d\pi}\frac{d\pi}{dz}\frac{\partial z}{\partial b}=\frac{\pi-y}{\pi(1-\pi)}\pi(1-\pi)=\pi-y$;

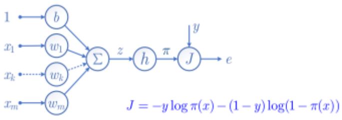
$\frac{\partial e}{\partial w}=\frac{de}{d\pi}\frac{d\pi}{dz}\frac{\partial z}{\partial w}=\frac{\pi-y}{\pi(1-\pi)}\pi(1-\pi)x=(\pi-y)x$

$\frac{de}{d\pi}=\frac{d}{d\pi}J(\pi)=-\frac{y}{\pi}+\frac{1-y}{1-\pi}=\frac{\pi-y}{\pi(1-\pi)}$

$\frac{d\pi}{dz}=\frac{d}{dz}h(z)=\frac{e^{-z}}{(1+e^{-z})^2}=\frac{1}{1+e^{-z}}\frac{e^{-z}}{1+e^{-z}}=\pi(1-\pi)$

$\frac{\partial z}{\partial b}=1,\ \frac{\partial z}{\partial w}=x$

多元 Logistic 回归:



$\frac{\partial e}{\partial b}=\frac{de}{d\pi}\frac{d\pi}{dz}\frac{\partial z}{\partial b}=\pi-y$; $\frac{\partial e}{\partial w_k}=\frac{de}{d\pi}\frac{d\pi}{dz}\frac{\partial z}{\partial w_k}=(\pi-y)x_k$

二分类问题的评价

判断为正样本如果$\pi(x_i)=\frac{e^{\widehat{\omega x_i+b}}}{1+e^{\hat{\omega}x_i+\hat{b}}}>c$

TN: 本来就是错的，预测也是错的. FP: 本来是错的，预测认为是对的

Sensitivity = True positive rate = TP / (TP+FN)

Specificity = True negative rate = TN / (TN+FP)

1-Sensitivity = False negative rate = FN / (TP+FN)

1-Specificity = False positive rate = FP / (TN+FP)

Recall = TP / (TP+FN); Precision = TP / (TP+FP)

Fall-out = FP / (TP+FP); F1-measure = 2×Precision× Recall / (Precision + Recall)

$ACC=\frac{TP+TN}{TP+TN+FP+FN}$

ROC：横轴: 1-Specificity；纵轴: Sensitivity

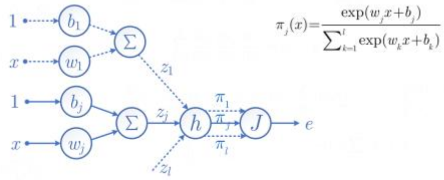
Softmax 回归: $P(Y=j)=\pi_j(x)=\frac{e^{w_jx+b_j}}{\sum_{k=1}^l e^{w_kx+b_k}}$

独热码:如果该观测属于第 j 类，则其独热码向量的第 j 维y_{ij}为 1，其他维为 0

softmax $(j,z_1,\dots,z_l)=\frac{e^{z_j}}{\sum_{k=1}^l e^{z_k}}$. 对数似然函数$\sum_{i=1}^n\sum_{j=1}^l y_{ij}\log\pi_j(x_i)$

最大化对数似然$max\sum_{i=1}^n\sum_{j=1}^l y_{ij}\log\pi_j(x_i)$

最小化交叉熵损失 $min-\frac{1}{n}\sum_{i=1}^n\sum_{j=1}^l y_{ij}\log\pi_j(x_i)$. 两者是一致的.



$e=-\sum_{k=1}^l y_k\log\pi_k(J(\mathbf{w},\mathbf{b})=-\sum_{k=1}^l y_k\log\pi_k)$

$\frac{\partial e}{\partial b_i}=-\sum_{k=1}^l\frac{\partial e_k}{\partial\pi_k}\frac{\partial\pi_k}{\partial z_i}\frac{\partial z_i}{\partial b_i}=-\frac{y_i}{\pi_i}\pi_i+\sum_{k=1}^l\frac{y_k}{\pi_k}\pi_i\pi_k=\pi_i-y_i$

$\frac{de}{dw_i}=-\sum_{k=1}^l\frac{\partial e_k}{\partial\pi_k}\frac{\partial\pi_k}{\partial z_i}\frac{\partial z_i}{\partial w_i}=(\pi_i-y_i)x$

$\frac{\partial\pi_k}{\partial z_i}=\frac{de^{z_k}}{dz_i}\left(\sum_{j=1}^l e^{z_j}\right)^{-1}+e^{z_k}\frac{\partial}{\partial z_i}\left(\sum_{j=1}^l e^{z_j}\right)^{-1}=\begin{cases}\pi_i(1-\pi_i) & k=i\\ -\pi_i\pi_k & k\neq i\end{cases}$

$\frac{\partial e_k}{\partial\pi_k}=\frac{y_k}{\pi_k}$; $\frac{\partial z_i}{\partial b_i}=1$; $\frac{\partial z_i}{\partial w_i}=x$.

前馈神经网络:输入单元—隐层单元—输出单元

Logistic 单元: $z=\mathbf{w}^T\mathbf{x}+b$. $a=\frac{1}{1+e^{-z}}$

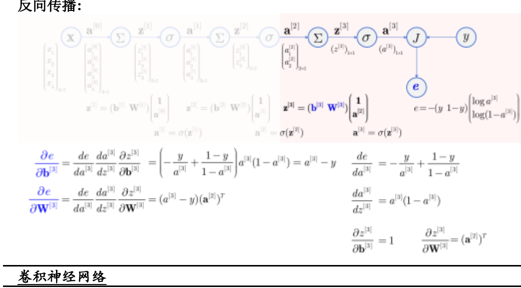
计算梯度: $\frac{da}{dz}=a(1-a)$; $\frac{\partial z}{\partial b}=1$; $\frac{\partial z}{\partial\mathbf{w}}=\mathbf{x}^T$

$\frac{\partial a}{\partial b}=\frac{da}{dz}\frac{\partial z}{\partial b}=a(1-a)$; $\frac{\partial a}{\partial\mathbf{w}}=\frac{da}{dz}\frac{\partial z}{\partial\mathbf{w}}=a(1-a)\mathbf{x}^T$

双曲正切单元: $a=\frac{e^z-e^{-z}}{e^z+e^{-z}}$; $\frac{da}{dz}=1-a^2=(1+a)(1-a)$

ReLU 单元: $a=max(0,z)$; $\frac{da}{dz}=\begin{cases}1 & z\geq 0\\ 0 & \text{otherwise}\end{cases}$

反向传播:



卷积神经网络

卷积: $g(x,y)*h(x,y)=\sum_i\sum_j g(i,j)h(i,j)$

填充后卷积：卷积结果与原图大小一致.

特点一：稀疏连接；特点二：参数共享；特点三：等变表示.

多通道卷积：图像有多个颜色通道,各通道分别进行卷积,再把卷积的结果相加.

一个卷积核提取一个特征,多个卷积核产生多个特征,每个特征对应于一个通道.

原始图$N\times N$，卷积核$m\times m$，卷积核$(N-m+1)\times(N-m+1)$.

池化：最大池化，平均池化，随机池化.

令 O=输出图像的尺寸, I=输入图像的尺寸, K=卷积层的核尺寸, S=移动步长, P =填充数,则输出图像尺寸

$$O=\frac{I-K+2P}{S}+1$$

严格来说这个尺寸是要取整的.

强化学习

马尔可夫过程：$P(S_{t+1}\mid S_t,S_{t-1},\dots,S_1)=P(S_{t+1}\mid S_t)$

考虑两个相邻时刻，定义状态转移概率：$p_{ss'}=P(S_{t+1}=s'\mid S_t=s)$

状态转移矩阵：从行转移到列，每行之和为 1. 马尔可夫过程由二元组(S，P) 描述，其中$\mathbf{S}=\{s_1,\dots,s_n\}$称为状态空间，$\mathbf{P}=(p_{ss'})_{n\times n}$称为状态转移矩阵

马尔可夫回报过程

由四元组$(\mathbf{S},\mathbf{P},\mathbf{r},\gamma)$描述．状态空间$\mathbf{S}=\{s_1,\dots,s_n\}$．状态转移矩阵$\mathbf{P}=(p_{ss'})_{n\times n}$.

状态期望回报：$\mathbf{r}=(r_1,\dots,r_n)$. $r_s=\mathrm{E}[R_{t+1}\mid S_t=s]$ 折现因子：$\gamma\in[0,1]$

Return：从当前时刻开始的折现回报之和

$G_t=R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots=\sum_{k=0}^\infty\gamma^kR_{t+k+1}$

状态价值就是状态累积回报的期望值

$$v(s)=\mathrm{E}[G_t\mid S_t=s]=\mathrm{E}[R_{t+1}+\gamma v(S_{t+1})\mid S_t=s]$$

贝尔曼期望方程

所有状态（矩阵形式）：$\mathbf{v}=\mathbf{r}+\gamma\mathbf{P}\mathbf{v}$

求解线性方程组，可得$\mathbf{v}=(\mathbf{I}-\gamma\mathbf{P})^{-1}\mathbf{r}$

马尔可夫决策过程:五元组(S,A,P,R,γ)

多了一个行动空间：$\mathbf{A}=\{a_1,\dots,a_m\}$.

策略：$\pi(a\mid s)=P(A_t=a\mid S_t=s)$

状态转移概率$P^\pi(S_{t+1}=s'\mid S_t=s)=\sum_{a\in A}\pi(a\mid s)p_{ss'}^a$

状态价值函数$v_\pi(s)=\mathrm{E}[G_t\mid S_t=s]$.

行动价值函数 $q_\pi(s,a)=\mathrm{E}[G_t\mid S_t=s,A_t=a]$

$$q_\pi(s,a)=r_s^a+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^a v_\pi(s')$$

状态价值的贝尔曼期望方程

$$v_\pi(s)=\sum_{a\in\mathbf{A}}\pi(a\mid s)\left(r_s^a+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^a v_\pi(s')\right)$$

行动价值的贝尔曼期望方程

$$q_\pi(s,a)=r_s^a+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^a\sum_{a'\in\mathbf{A}}\pi(a'\mid s')q_\pi(s',a')$$

动态规划—策略评价(状态价值计算)、策略改进

策略评价：$v_{k+1}(s)=r_s^\pi+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^\pi v_k(s')$

策略的好坏用状态价值来评价.

最优策略：$\pi_*\geq\pi'$: $v_{\pi_*}(s)\geq v_{\pi'}(s),\forall s,\forall\pi'$

最优状态价值即最优策略下的状态价值 $v_*(s)=\max_\pi v_\pi(s)$

最优行动价值即最优策略下的行动价值.

最优状态价值即同时刻最优行动价值 $v_*(s)=\max_{a\in\mathbf{A}}q_*(s,a)$

策略改进：基于原策略，产生新策略

如果$v_\pi(s)\leq q_\pi(s,\pi'(s))$,那么$v_\pi(s)\leq v_{\pi'}(s)$.

策略迭代：交替进行策略评价和策略改进.

价值迭代

策略改进计算量很大，故每次策略评价后立即进行策略改进.

状态价值贝尔曼最优方程

$$v_*(s)=\max_{a\in\mathbf{A}}\left(r_s^a+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^a v_*(s')\right)$$

行动价值贝尔曼最优方程

$$q_*(s,a)=r_s^a+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^a\max_{a'\in\mathbf{A}}q_*(s',a')$$

价值迭代：$v_{k+1}(s)=\max_{a\in\mathbf{A}}\left(r_s^a+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^a v_k(s')\right)$

则策略：$\pi_*(s)=\arg\max_{a\in\mathbf{A}}\left(r_s^a+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^a v_*(s')\right)$

同步迭代：算完所有状态后一次更新，保存两份状态价值，计算量大收敛较慢.

$$v_{k+1}(s)=\max_{a\in\mathbf{A}}\left(r_s^a+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^a v_k(s')\right)$$

异步迭代：算一个更新一个,只有一份状态价值,计算量小收敛较快.

$$v(s)=\max_{a\in\mathbf{A}}\left(r_s^a+\gamma\sum_{s'\in\mathbf{S}}p_{ss'}^a v(s')\right)$$

蒙特卡洛预测

首次访问蒙特卡洛：根据待评价策略产生观测片段，若是第一次出现，计算$G_i=R_{t+1}+\gamma R_{t+2}+\cdots+\gamma^{T-t}R_{T+1}$，重复 n 次，得到$G_1,G_2,\dots,G_n$,则估计状态价值$V(S_t)=\frac{1}{n}\sum_{i=1}^n G_i$

每次访问蒙特卡洛.

增量式蒙特卡洛预测:$V(S_t)\leftarrow V(S_t)+\frac{1}{k}\left(G_t-V(S_t)\right)$，$k\leftarrow k+1$

定步长蒙特卡洛预测：$V(S_t)\leftarrow V(S_t)+\alpha\left(G_t-V(S_t)\right)$

行动价值预测:$Q(S_t,A_t)=\frac{1}{n}\sum_{i=1}^n G_i$

增量式预测：$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\frac{1}{k}\left(G_t-Q(S_t,A_t)\right)$,$k\leftarrow k+1$

定步长预测：$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\alpha\left(G_t-Q(S_t,A_t)\right)$

探索与利用:

$$\pi(a\mid s)=\begin{cases}1-\varepsilon+\frac{\varepsilon}{m} & \text{if } a=\arg\max_{a\in\mathbf{A}}Q(s,a)\\ \frac{\varepsilon}{m} & \text{otherwise}\end{cases}$$

时序差分预测

$$V(S_t)\leftarrow V(S_t)+\alpha\left(R_{t+1}+\gamma V(S_{t+1})-V(S_t)\right)$$

其中$R_{t+1}+\gamma V(S_{t+1})-V(S_t)$称为 TD Error.

$G_t\approx R_{t+1}+\gamma V(S_{t+1})$ 称之为 TD Target.

使用时序差分代替蒙特卡洛进行策略评价，有以下两类控制方法：

On-policy: SARSA

Off-policy: Q-learning, Expected SARSA

SARSA:

$$\boldsymbol{Q}(\boldsymbol{S}_t,\boldsymbol{A}_t)\leftarrow\boldsymbol{Q}(\boldsymbol{S}_t,\boldsymbol{A}_t)+\boldsymbol{\alpha}\left(\boldsymbol{R}_{t+1}+\gamma\boldsymbol{Q}(\boldsymbol{S}_{t+1},\boldsymbol{A}_{t+1})-\boldsymbol{Q}(\boldsymbol{S}_t,\boldsymbol{A}_t)\right)$$

行为策略和目标策略均为ε-贪心.　根据策略取采样值

Q-Learning:

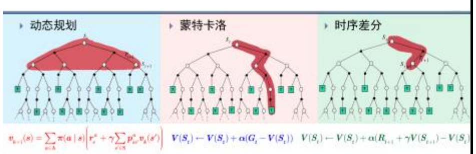
$$\boldsymbol{Q}(\boldsymbol{S}_t,\boldsymbol{A}_t)\leftarrow\boldsymbol{Q}(\boldsymbol{S}_t,\boldsymbol{A}_t)+\boldsymbol{\alpha}\left(\boldsymbol{R}_{t+1}+\gamma\max_{a\in\mathbb{A}}\boldsymbol{Q}(\boldsymbol{S}_{t+1},\boldsymbol{a})-\boldsymbol{Q}(\boldsymbol{S}_t,\boldsymbol{A}_t)\right)$$

行为策略为ε-贪心，目标策略为贪心.　根据策略取最大值

Expected SARSA:

$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\alpha\left(R_{t+1}+\gamma\sum_{a\in\mathrm{A}}\pi(a\mid S_{t+1})Q(S_{t+1},a)-Q(S_t,A_t)\right)$

行为策略为ε-贪心，目标策略为行动期望.　根据策略取期望值



增量式价值函数近似

状态价值近似:特征提取$\mathrm{x}(s)=(x_1(s),x_2(s),\cdots,x_m(s))^T$,函数族

$\hat{v}(\mathrm{s}\mid\mathrm{x},\mathrm{w})=\mathrm{w}^T\mathrm{x}$ (线性函数),

优化准则 $\min\sum_{i=1}^n(v_\pi(s_i)-\hat{v}(s_\mathrm{i}\mid\mathrm{x},\mathrm{w}))^2$,

优化方法 SGD: $\mathrm{w}^{(\mathrm{new})}\leftarrow\mathrm{w}+\alpha(v_\pi(s)-\hat{v}(s\mid\mathrm{x},\mathrm{w}))\mathrm{x}$

$\mathbf{w}^{(\mathrm{new})}\leftarrow\mathbf{w}+\alpha(r+\gamma\hat{v}(s'\mid\mathbf{x},\mathbf{w})-\hat{v}(s\mid\mathbf{x},\mathbf{w}))\mathbf{x}$

蒙特卡洛：采样长期回报$\mathbf{w}^{(\mathrm{new})}\leftarrow\mathbf{w}+\boldsymbol{\alpha}(G_t-\hat{\boldsymbol{v}}(s\mid\mathbf{x},\mathbf{w}))\mathbf{x}$

行动价值近似:特征提取$\mathrm{x}(s,a)=(x_1(s,a),\cdots,x_m(s,a))^T$,函数族

$\hat{q}(\mathrm{s},\mathrm{a}\mid\mathrm{x},\mathrm{w})=\mathrm{w}^T\mathrm{x}$ (线性函数),

优化准则 $\min\sum_{i=1}^n(q_\pi(s_i,a_i)-\hat{q}(s_\mathrm{i},a_\mathrm{i}\mid\mathrm{x}_\mathrm{i},\mathrm{w}))^2$,

优化方法 SGD: $\mathrm{w}^{(\mathrm{new})}\leftarrow\mathrm{w}+\alpha(q_\pi(s,a)-\hat{q}(s,a\mid\mathrm{x},\mathrm{w}))\mathrm{x}$

$\mathbf{w}^{(\mathrm{new})}\leftarrow\mathbf{w}+\alpha(r+\gamma\hat{q}(s',a'\mid\mathbf{x},\mathbf{w})-\hat{q}(s,a\mid\mathbf{x},\mathbf{w}))\mathbf{x}$

批量式价值近似(经验回放)

经验回放的 SGD 方法: $\mathrm{w}^{(\mathrm{new})}\leftarrow\mathrm{w}+\alpha(v_\pi(s)-\hat{v}(s\mid\mathrm{x},\mathrm{w}))\mathrm{x}$

交替进行采样和梯度下降至收敛 $\min\sum(v_\pi(s)-\hat{v}(s\mid\mathrm{x},\mathrm{w}))^2$

状态价值的蒙特卡洛经验回放:$v_\pi(s)\approx G_t$

积累经验: $\mathrm{D}=\{(s_1,G_t(s_1)),\cdots,(s_n,G_t(s_n))\}$

从经验采样: $(s,G_t(s))\sim\mathrm{D}$，计算系数: $\mathrm{w}=(\mathrm{X}^\mathrm{T}\mathrm{X})^{-1}\mathrm{X}^\mathrm{T}g$

状态价值的时序差分经验回放:$v_\pi(s)\approx R_{t+1}+\hat{v}(S_{t+1}\mid\mathrm{x},\mathrm{w})$

$\mathrm{D}=\{(s_1,r_1',s_1'),\cdots,(s_n,r_n',s_n')\}$, $(s_t,r_t',s_t')\sim\mathrm{D}$, $\mathrm{w}=(\mathrm{X}^\mathrm{T}(\mathrm{X}-\gamma\mathrm{X}))^{-1}\mathrm{X}^\mathrm{T}\mathrm{r}$

行动价值的时序差分经验回放:

$\mathrm{D}=\{(s_1,a_1,r_1',s_1',a_1'),\cdots,(s_n,a_n,r_n',s_n',a_n')\}$, $(s_t,a_t,r_t',s_t',a_t')\sim\mathrm{D}$