1. About AI

像人一样行动/思考，理性特征

三大学派：符号主义（逻辑推理和符号处理）行为主义（黑箱与机器学习）连接主义（神经网络）

2. 搜索问题

搜索问题：状态空间=初始状态+行为+转移模型

状态空间图 = 结点（一个只出现一次）+ 边

搜索树：无环图，可能存在大量重复和无限延拓

2.1 盲目搜索 Uninformed Search

b 叉树，m 层	Complete	Time	Space	Optimal
DFS 深度优先	when m=inf×	$O(b^m)$	O(bm)	No(Only left)
BFS 广度优先	√	$O(N)=O(b^m)$	$O(N)=O(b^m)$	When cost=1 √
DFS + Iterative Deepening	√	$O(N)=O(b^m)$	O(bm)	√
Uniform-Cost Search UCS	√ 问题成本 C*	$O(b^{C*/\epsilon})$	$O(b^{C*/\epsilon})$	√

UCS：每次推出当前最小代价节点并标记，加入该节点子节点，目标出栈时结束

2.2 有信息搜索 Informed Search

估计 h(n)，实际 h*(n)	Complete	Optimal
启发式函数：估计与目标接近程度	**可接受 0≤h(n)≤h*(n)**	**一致性 h(A)-h(C)≤cost(A to C)**
贪婪算法	×	×
A*算法	√	√
图搜索 Graph Search	√	√

* 满足一致性（效率与准确）一定满足可接受（确保最优解），同时即为最优

* A*算法的思想：f(n) = 代价函数 UCS(g(n)) + 预期函数 Greedy(h(n))

3. 博弈论（对抗搜索）

* 纳什均衡：不能单方面改变自己策略以使自己的收益变得更好的均衡状态

* 极小极大化问题：$V(S) = \mathrm{Max}_{s' \in \mathrm{successors}} V(S')$，每个玩家轮流极小 - 极大取值

* 效率：与 DFS 类似，$\mathrm{Time} = O(b^m)$, $\mathrm{Space} = O(bm)$

* 评价函数：不同特征加权求和评估当前状态 **|** 节约资源：Depth-Limited

* **回溯剪枝**：比较当前节点的价值与已知最值，若此时已不利，提前终止搜索。

* **期望最大值搜索**：解决不确定性问题，加权求和计算出每个节点的期望值，目的是达到平均效用最大（效用：任何理性的偏好值 U=pU(A)+(1-p)U(B)）

4. 约束满足问题 CSP：找到满足所有约束条件的变量取值的问题

三要素：变量 Variable，域 Domain（取值范围），约束 Constraint

* Constraint Graph：图的节点对应变量，节点相连（弧）代表变量之间存在约束

* 分类：离散变量：有限域/无限域。可转化为二元约束（只有两个 V 相互约束）

连续变量：线性约束可解（离散无限域同）

* 解法：初始状态（空集）+后继方程（不断赋值）+目标检测（约束）

4.1 回溯搜索问题：Space = $O(b \cdot n)$，b=域大小，n=变量数

思路：**是每次只给一个变量赋值（按顺序），随时检查约束**的深度 DFS

过滤（Filtering）	排序（Ordering）	**最小剩余值（MRV）**
划去不满足约束的选择	→	优先选择候选值集合最小的变量

* 弧一致性检查（Arc Consistency Check）：**搜索前的预处理**，遍历每对相关变量的所有可能取值组合（尤其是 X 的 neighbor），划去不满足约束条件的情况

4.2 结构

* 复杂度：假设 n 变量，分成 c 变量的小问题，d 是变量域大小

原先 $O(d^n) \rightarrow$ 分而论之 $O(\frac{n}{c} \cdot d^c) \rightarrow$ 无环树状图 $O(n \cdot d^2)$

* 树状结构求解：从 root 开始扩展，在每个节点上，根据约束条件选择变量的一个值。若某节点上无法继续，就会回溯到上一个节点，尝试选择不同的值。

* 优势：弧一致性可以在父到子节点的传播过程中进行维护，使变量域中至少有一个合法取值。减少搜索空间，提高了算法效率，局部变动不影响其他部分。

* **最小割集**：把这个（几个）点删掉，剩下由图变成无环图：$O(d^2 \cdot (n-c) \cdot d^c)$

4.3 局部搜索

爬山 Hill Climbing	模拟退火（全局优化） Simulated Annealing	遗传（全局搜索） Genetic Algorithm
在当前解的邻域中搜索更好的解，局部最优	允许劣质解，迭代中逐渐减小接受劣质解的概率	对解变异交叉+评估选择，保留优质解并迭代

5. 概率推理问题 Probabilistic Reasoning

5.1 概率模型：是一组随机变量的联合分布

* 先验概率分布：$P(a|b) = {}^{P(a,b)}/_{P(b)} = {}^{P(b|a) \cdot P(a)}/_{P(b)}$

* 全概率公式：$P(a) = \sum_i P(a|b_i)$

* 链式法则：$P(x1, x2, ..., xn) = P(x1) \cdot P(x2|x1) \cdot P(x3|x1, x2) \ldots$

* 独立：$P(X, Y) = P(X) \cdot P(Y)$

* 条件独立：$P(X, Y|Z) = P(X|Z) \cdot P(Y|Z)$ 或者：$P(X|Y, Z) = P(X|Z)$

5.2 **贝叶斯网络**：用简单的局部分布（条件概率）描述复杂联合分布（模型）

贝叶斯网络 Bayesian's Network = 拓扑图（有向无环图）+概率表

* 联合概率分布 in Bayesian：$P(X1 \ldots Xn) = \prod P(Xi|Parents(Xi))$

* 箭头：**A->B 表示 B 的取值依赖于 A 的取值，并不一定表示 A 是 B 的因果原因**

中间结点状态	H2H：2 个指向 1 个	T2T：1 个指向 2 个	H2T：链式
已知	不独立	独立	独立
未知	独立	不独立	不独立

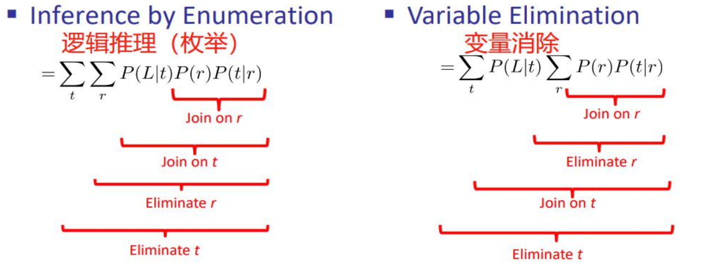
* 多节点图：找 X-Y 的所有路径，只要有一个不独立那就不独立

5.3 变量消除法

1）**因子合并 Join Factors**：将所有的因子合并成一个大的因子，一般通过**乘积**实现，以得到一个包含了所有变量的联合概率分布

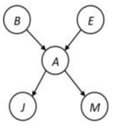
2）**边际化 Marginalization**：对于要消除的变量**求和**，即对包含该变量的因子进行求和，形成一个包含了其他变量的较小的因子，不再涉及该被消除的变量

3）结果（归一化）：反复执行到只剩下目标变量因子，以计算目标的后验概率



$$P(B|j,m) \propto P(B,j,m)$$

$P(B)$	$P(E)$	$P(A\|B,E)$	$P(j\|A)$	$P(m\|A)$



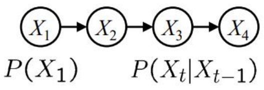
$$\begin{aligned} P(B|j,m) &\propto P(B,j,m) \\ &= \sum_{e,a} P(B,j,m,e,a) \\ &= \sum_{e,a} P(B)P(e)P(a|B,e)P(j|a)P(m|a) \\ &= \sum_{e} P(B)P(e)\sum_{a} P(a|B,e)P(j|a)P(m|a) \\ &= \sum_{e} P(B)P(e)f_1(B,e,j,m) \\ &= P(B)\sum_{e} P(e)f_1(B,e,j,m) \\ &= P(B)f_2(B,j,m) \end{aligned}$$

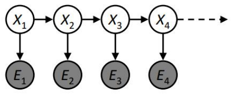
* **规则**：只有所有变量都出现在“|”之后时，最终才会在“|”之后

* **复杂度**：受消元顺序影响，用 Max(2^i)估算，i 是一次消元影响到的变量数

先验抽样	拒绝抽样	极大似然抽象	吉布斯抽样
按拓扑顺序逐层采样，顺序采样	从先验分布中采样，根据观测数据的条件概率进行接受或拒绝→联合分布	基于概率分布参数的最大似然估计，根据观测数据计算似然函数，直到收敛	每一步采样中，固定其他变量，根据条件概率分布采样当前变量，逐步更新其他

6. 马尔可夫链





基本假设：未来的状态只依赖于当前状态，而与过去的状态无关，因此独立

* 由上，特别的，有 $P(X1,X2...Xn) = P(X1)\cdot P(X2|X1)\cdot P(X3|X2)\cdots P(Xn|Xn-1)$

* 假设他是收敛的：$P_{inf}(X) = P_{inf+1}(X) = \sum_x P(X|x)\cdot P_{inf}(x)$

6.1 隐藏马尔可夫链 Hidden Markov

* eg. $P(X1,E1,X2,E2) = P(X1)\cdot P(E1|X1)\cdot P(X2|X1)\cdot P(E2|X2)$

通过观测输出 e 来推断系统状态 X，令 $B(Xt) = P(Xt|e_{1:t})$，$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$

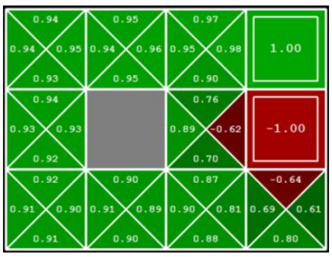
$$P(X_{t+1}|e_{1:t}) = \sum_{xt} P(X_{t+1}|xt)\cdot P(xt|\,e_{1:t})$$
$$P(X_{t+1}|e_{1:t+1}) = P(e_{t+1}|X_{t+1})\cdot P(X_{t+1}|e_{1:t})$$

6.2 马尔可夫决策过程 Markov Decision Process

考虑一个决策者（agent）在环境中采取一系列动作并观察到相关的奖励

模型：状态 State + 动作 Action + 转移模型 Transition Model + 奖励函数 Reward Function + 价值函数 Value Function + 折扣 Discount（Maybe）

* 决策过程解法：

值迭代	策略迭代
1）所有状态的价值函数初始化 2）价值函数：对所有动作 a， $V^*(S) = \mathrm{Max}\sum T\cdot[R+\gamma V^*(S')]$ 3）迭代更新，选择最大化的策略 4）重复直到收敛 $O(S^2A)$	1）初始化：随机选择一个初始策略 2）策略评估：根据当前策略进行策略评估，计算每个状态的价值函数 3）策略改进：对每个状态选择最优策略 4）重复直到收敛 $O(S^2)$
	

6.3 完美期望价值 Value of Perfect Information（信息带来的收益、价值）

E 表示事件，e 表示信息，有一撇表示有信息条件下

MEU：最大期望效用：$MEU(e) = Max_a \sum_s P(s|e)U(s,a)$

$$MEU(e,e') = Max_a \sum P(s|e,e')U(s,a)$$

$$MEU(e,E') = \sum_{e'} P(e'|e)\cdot MEU(e,e')$$

$$\mathbf{VPI(E'|e) = VPI = MEU(完全信息) - MEU(有限信息)}$$
$$\mathbf{= MEU(e,E') - MEU(e)}$$

7. 朴素贝叶斯与感知机 Naive Bayes & Perceptron

7.1 朴素贝叶斯

思想：通过已知的特征值来预测未知实例的类别，将每个特征视为独立且对分类结果的影响相互独立

* **贝叶斯推断：计算目标变量 Y 的后验分布**

1）计算每个目标变量标签和证据的联合概率 $P(Y)\cdot P(F1...Fn|Y)$

2）对所有可能的目标变量标签进行求和，得到证据概率 $P(F1...Fn)$

3）将第一步计算的联合概率除以第二步计算的证据概率，对联合概率归一化

* **参数估计**：$P_{ML}(x) = {}^{count(x)}/_{total\ samples}$

* 拉普拉斯平滑：为所有事件添加一个先验的“假想计数”，以避免零概率的问题，有：$P_{ML}(x) = {}^{count(x)}/_{N+|x|\cdot k}$（N 采样总数，|x|类别）

* 过拟合：过度拟合训练数据，与（**训练集**）数据中的噪声建立了过多的关联，在新数据（**测试集**和**验证集**）上的表现不佳

7.2 感知机

二分类线性算法，通过学习一组权重参数，将输入实例映射到预定义的类别

W：每一类的权重向量 | $w\cdot f(x)$：每一类的学习得分

$$y = \arg\max\ w_y\cdot f(x)$$

一开始：w=0 → if wrong：$w_y = w_y - f(x)$；$w_{y^*} = w_{y^*} + f(x)$

* 缺点：噪声；过拟合风险；学习速度较难选择；只能线性

* MIRA：引入规范化步长，对步长进行调整，从而更好地适应噪声和不均衡数据（步长 C）

$$\tau^* = \min\left(\frac{(w'_y - w'_{y^*})\cdot f + 1}{2f\cdot f}, C\right)$$

MIRA

$$\min_w \frac{1}{2}||w-w'||^2$$
$$w_{y^*}\cdot f(x_i) \geq w_y\cdot f(x_i) + 1$$

SVM 支持向量机

$$\min_w \frac{1}{2}||w||^2$$
$$\forall i,y\ w_{y^*}\cdot f(x_i) \geq w_y\cdot f(x_i) + 1$$

8. 神经网络 Neural Network

* 基本模型：由多个称为神经元（neuron）或节点（node）的单元组成，相互连接形成层次结构，并通过学习权重参数来实现输入数据的非线性转换和预测。

* **基本架构**：

1）输入层：原始数据和特征向量

2）隐藏层：通过连接方式进行信息传递，加权求和后通过激活函数进行非线性转换，隐藏层的存在使得神经网络能够学习和提取更高级别的特征表示

3）输出层：通过输出层生成预测结果，每个神经元对应一个类别或输出值

9. 物联网 IoT

爬