2023年

判断题：

> **没有智能体可以在一个完全不可观察的环境中理性地工作。**（正确）
>
> - 理由：在一个完全不可观察的环境中，智能体无法获取任何信息，因此无法进行理性的决策。
>
> **如果对搜索树中的每一步的代价都增加一个正数 CCC，则一致代价（Uniform Cost）搜索算法可能会给出不一样的搜索路径。**（错误）
>
> - 理由：一致代价搜索算法选择的是代价最小的路径。如果所有路径的代价都增加了相同的正数 CCC，则相对顺序不会改变，最终的路径也不会改变。
>
> **如果 h1(s)h_1(s)h1(s) 和 h2(s)h_2(s)h2(s) 是两个 A* 算法的可采纳（Admissible）启发式函数，则它们的平均值 h3(s)=12h1(s)+12h2(s)h_3(s) = \frac{1}{2} h_1(s) + \frac{1}{2} h_2(s)h3(s)=21h1(s)+21h2(s) 也必然是可采纳的。**（正确）
>
> - 理由：可采纳的启发式函数是不会高估实际代价的，因此它们的平均值也不会高估实际代价，仍然是可采纳的。
>
> **在贝叶斯网络中，已知随机变量 A 和 B 互相独立（ALB），对任意新的变量 C，仍然成立 A⊥B∣CA \bot B \mid CA⊥B∣C。**（错误）
>
> - 理由：如果 A 和 B 互相独立，但在给定 C 的条件下，A 和 B 可能会变得不独立，这取决于变量之间的具体关系。
>
> **针对包含一些噪声数据点的非线性分类数据，使用具有松弛变量（Slack Variable）的 SVM 创建软间隔（Soft Margin）分类器。对其中控制错误分类惩罚程度的惩罚参数 CCC 选择较小的值，通常会减少训练数据的过拟合（overfitting）。**（正确）
>
> - 理由：较小的 CCC 值会减少对误分类的惩罚，从而使模型变得更加柔和，减少过拟合。

选择题

T1

对于线性核函数（Linear Kernel）的支持向量机（SVM），分类边界应为线性边界，即一条直线。我们需要在图中找到具有线性分界线的分类结果。

- 边界明显是曲线，因此不是线性边界。
- **B**：边界是直线，因此是线性边界。
- **C**：边界是直线，因此是线性边界。
- **D**：边界是曲线，因此不是线性边界。
- **E**：边界明显是曲线，因此不是线性边界。
- **F**：边界明显是曲线，因此不是线性边界。

修正后的答案

所以，使用线性核的SVM对应的可能结果是 **B 和 C**。

T2

给定命题 PPP 和 QQQ，下列情况哪种是公式 ¬P∨Q→¬P∧Q\neg P \lor Q \rightarrow \neg P \land Q¬P∨Q→¬P∧Q 的模型?

我们需要验证公式 ¬P∨Q→¬P∧Q\neg P \lor Q \rightarrow \neg P \land Q¬P∨Q→¬P∧Q 在四种情况下的真假性。公式 ¬P∨Q→¬P∧Q\neg P \lor Q \rightarrow \neg P \land Q¬P∨Q→¬P∧Q 可以用真值表的方法逐一验证每种情况下公式的真值。

真值表方法验证公式

PPP	QQQ	¬P\neg P¬P	¬Q\neg Q¬Q	¬P∨Q\neg P \lor Q¬P∨Q	¬P∧Q\neg P \land Q¬P∧Q	¬P∨Q→¬P∧Q\neg P \lor Q \rightarrow \neg P \land Q¬P∨Q→¬P∧Q
F	F	T	T	T	F	F
F	T	T	F	T	T	T
T	F	F	T	F	F	T
T	T	F	F	T	F	F

由真值表可以看出，公式 ¬P∨Q→¬P∧Q\neg P \lor Q \rightarrow \neg P \land Q¬P∨Q→¬P∧Q 在下列情况下为真：

- P=false,Q=trueP = \text{false}, Q = \text{true}P=false,Q=true
- P=true,Q=falseP = \text{true}, Q = \text{false}P=true,Q=false

所以答案是：

B, C

T3

A. 给定 A, C, D, E 时，B 和 F 条件独立
B. 给定 A, C, E 时，B 和 F 条件独立
C. 给定 A，E 时，B 和 F 条件独立
D. 给定 C，D 时，B 和 F 条件独立
F. 给定 C，E 时，B 和 F 条件独立

> ### 给定 A,C,D,EA, C, D, EA,C,D,E 时，BBB 和 FFF 条件独立。
>
> - 路径：B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F
> - 在给定 DDD 和 EEE 时，路径 B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F 被 DDD 和 EEE 封闭。
> - 因此，给定 A,C,D,EA, C, D, EA,C,D,E 时，BBB 和 FFF 条件独立。

> ### B. 给定 A,C,EA, C, EA,C,E 时，BBB 和 FFF 条件独立。
>
> - 路径：B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F
> - 在给定 EEE 时，路径 B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F 被 EEE 封闭。
> - 因此，给定 A,C,EA, C, EA,C,E 时，BBB 和 FFF 条件独立。
>
> ### C. 给定 A,EA, EA,E 时，BBB 和 FFF 条件独立。
>
> - 路径：B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F
> - 在给定 EEE 时，路径 B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F 被 EEE 封闭。
> - 因此，给定 A,EA, EA,E 时，BBB 和 FFF 条件独立。
>
> ### D. 给定 C,DC, DC,D 时，BBB 和 FFF 条件独立。
>
> - 路径：B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F
> - 在给定 DDD 时，路径 B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F 被 DDD 封闭。
> - 因此，给定 C,DC, DC,D 时，BBB 和 FFF 条件独立。
>
> ### E. 给定 C,EC, EC,E 时，BBB 和 FFF 条件独立。
>
> - 路径：B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F
> - 在给定 EEE 时，路径 B→D→E→FB \rightarrow D \rightarrow E \rightarrow FB→D→E→F 被 EEE 封闭。
> - 因此，给定 C,EC, EC,E 时，BBB 和 FFF 条件独立。
>
> ## 结论
>
> 所有选项都是正确的，给定的条件都使得 BBB 和 FFF 条件独立。要选择“哪种说法一定正确”，所有的选项都是正确的。这可能是一个出题错误，所有选项都满足条件独立。
>
> 如果必须选择一个答案，可以选择 **A, B, C, D, E**，因为它们都满足条件独立性。

T4

解答：

A. **预测一个信用卡交易是欺诈还是合法**（适合分类学习算法）

- 这是一个典型的二分类问题，可以使用分类算法来预测交易是否为欺诈。

B. **预测明天下多少雨**（不适合分类学习算法）

- 这是一个回归问题，因为需要预测一个连续的数值。

C. **预测手写字符图像所表示的字母**（适合分类学习算法）

- 这是一个多分类问题，可以使用分类算法来识别手写字符图像所表示的字母。

D. **根据客户的购买模式将客户数据库分成不同的簇（Cluster），其中簇的性质由计算机自动确定，而不是以任何方式由人类提供**（不适合分类学习算法）

- 这是一个聚类问题，而不是分类问题。适合使用聚类算法如 K-means。

因此，适合分类学习算法的任务是 A 和 C。

T5

1. $C1$ 的质心：

$$质心=\left(\frac{0+1}{2},\frac{1+0}{2}\right)=\left(\frac{1}{2},\frac{1}{2}\right)=(0.5,0.5)$$

2. $C2$ 的质心：

$$质心=\left(\frac{2+4}{2},\frac{2+4}{2}\right)=(3,3)$$

3. $C3$ 的质心：

$$质心=\left(\frac{5+6+7}{3},\frac{5+6+7}{3}\right)=(6,6)$$

对比选项：

A. $C1:(3,3),C2:(4,4),C3:(6,6)$

- 不正确，因为 $C1$ 的质心是 $(0.5,0.5)$

B. $C1:(1,1),C2:(3,3),C3:(6,6)$

- 不正确，因为 $C1$ 的质心是 $(0.5,0.5)$

C. $C1:(0.5,0.5),C2:(3,3),C3:(6,6)$

- 正确，这符合计算的质心

D. $C1:(1,1),C2:(2,2),C3:(12,12)$

- 不正确，$C3$ 的质心是 $(6,6)$，而 $C1$ 的质心是 $(0.5,0.5)$

E. 以上都不对。

- 不正确，因为选项 C 是正确的

Question3：

1. 广度优先搜索 (BFS)

广度优先搜索会按照层次逐层访问节点：

- 开始于根节点A，访问顺序：[A]
- 下一层节点：[B, C, D]
- 再下一层节点：[E, F, G, H, I, J]

所以，访问顺序为：

rust 复制代码

```rust
A -> B -> C -> D -> E -> F -> G -> H -> I -> J
```

2. 深度优先搜索 (DFS)

深度优先搜索会尽可能深入搜索树的每一个分支：

- 从根节点A开始，访问顺序：[A]
- 访问节点B：[A -> B]
- 访问节点E：[A -> B -> E]
- 回溯到B，访问节点F：[A -> B -> E -> F]
- 回溯到B，访问节点G：[A -> B -> E -> F -> G]
- 回溯到A，访问节点C：[A -> B -> E -> F -> G -> C]
- 访问节点H：[A -> B -> E -> F -> G -> C -> H]
- 回溯到A，访问节点D：[A -> B -> E -> F -> G -> C -> H -> D]
- 访问节点I：[A -> B -> E -> F -> G -> C -> H -> D -> I]
- 访问节点J：[A -> B -> E -> F -> G -> C -> H -> D -> I -> J]

3. 一致代价搜索 (Uniform Cost)

一致代价搜索按照路径代价从小到大依次扩展节点:

- 初始状态A，访问顺序: [A]
- 最小代价边为B（1），访问顺序: [A -> B]
- 接下来最小代价边为E（6），访问顺序: [A -> B -> E]
- 接下来最小代价边为F（6），访问顺序: [A -> B -> E -> F]
- 接下来最小代价边为C（2），访问顺序: [A -> B -> E -> F -> C]
- 接下来最小代价边为D（3），访问顺序: [A -> B -> E -> F -> C -> D]
- 接下来最小代价边为I（3），访问顺序: [A -> B -> E -> F -> C -> D -> I]
- 最后访问目标节点J（3），访问顺序: [A -> B -> E -> F -> C -> D -> I -> J]

4. 迭代加深深度优先搜索 (Iterative Deepening)

迭代加深深度优先搜索在每一步都会增加深度限制:

- 深度0: [A]
- 深度1: [A -> B, C, D]
- 深度2: [A -> B -> E, F, G; C -> H; D -> I, J]

所以最终的访问顺序是:

rust 复制代码

```rust
A -> B -> E -> F -> G -> C -> H -> D -> I -> J
```

5. 最佳优先搜索 (Best First)

最佳优先搜索会依据启发式函数（估计值）选择最优节点进行扩展。假设启发式函数为图中给出的估计值：

- 起始于A，访问顺序：[A]
- 最小估计值为B（2），访问顺序：[A -> B]
- 最小估计值为E（0），访问顺序：[A -> B -> E]

6. A* 搜索算法

A*算法结合路径代价和启发式估计值：

- 起始于A，访问顺序：[A]
- 路径代价+估计值：B (1+2=3), C (2+4=6), D (3+4=7)
- 选择代价最小的B，访问顺序：[A -> B]
- 路径代价+估计值：E (1+6=7), F (2+3=5), G (3+4=7)
- 选择代价最小的F，访问顺序：[A -> B -> F]
- 最终访问目标节点J

所以访问顺序为：

rust 复制代码

```rust
A -> B -> F -> J
```

总结不同算法的状态访问顺序：

1. **广度优先搜索**：`A -> B -> C -> D -> E -> F -> G -> H -> I -> J`
2. **深度优先搜索**：`A -> B -> E -> F -> G -> C -> H -> D -> I -> J`
3. **一致代价搜索**：`A -> B -> E -> F -> C -> D -> I -> J`
4. **迭代加深深度优先搜索**：`A -> B -> E -> F -> G -> C -> H -> D -> I -> J`
5. **最佳优先搜索**：`A -> B -> E`
6. *A 搜索**：`A -> B -> F -> J`

Question5：

1. 转换命题为一阶逻辑形式

- $D(x)$: x是鸭子
- $A(x)$: x是禽类
- $S(x)$: x会游水
- $E(x)$: x是旱鸭子

已知命题：

- (1) $\forall x(D(x) \rightarrow A(x))$ 鸭子是一种禽类
- (2) $\forall x((D(x) \wedge egS(x)) \rightarrow E(x))$ 不会游水的鸭子都是旱鸭子
- (3) $A(\text{Twitter})$ Twitter是一种禽类
- (4) $egS(\text{Twitter})$ Twitter不会游水

待证明的结论：

- (5) $egD(\text{Twitter}) \vee E(\text{Twitter})$ Twitter或者不是鸭子，或者是旱鸭子

2. 使用归结原理

步骤 1: 转换为子句形式

- (1) $\forall x(D(x) \rightarrow A(x))$ 转换为 $\forall x(egD(x) \vee A(x))$
- (2) $\forall x((D(x) \wedge egS(x)) \rightarrow E(x))$ 转换为 $\forall x(egD(x) \vee S(x) \vee E(x))$
- (3) $A(\text{Twitter})$
- (4) $egS(\text{Twitter})$
- (5) $egD(\text{Twitter}) \vee E(\text{Twitter})$

步骤 2: 实例化和简化

由于我们关注的是Twitter，我们可以实例化变量x为Twitter：

- $egD(\text{Twitter}) \vee A(\text{Twitter})$
- $egD(\text{Twitter}) \vee S(\text{Twitter}) \vee E(\text{Twitter})$
- $A(\text{Twitter})$
- $egS(\text{Twitter})$
- $egD(\text{Twitter}) \vee E(\text{Twitter})$

步骤 3: 应用归结操作

1. 从$egS(\text{Twitter})$ 和 $egD(\text{Twitter}) \vee S(\text{Twitter}) \vee E(\text{Twitter})$ 归结得到 $egD(\text{Twitter}) \vee E(\text{Twitter})$ 。
2. $egD(\text{Twitter}) \vee A(\text{Twitter})$ 和 $A(\text{Twitter})$ 归结得到 $egD(\text{Twitter})$。
3. 最后，$egD(\text{Twitter}) \vee E(\text{Twitter})$ 和 $egD(\text{Twitter})$ 归结得到 $E(\text{Twitter})$。

结论

通过归结原理，我们证明了如果Twitter是禽类且不会游水，则Twitter或者是旱鸭子。因此，结论(5) $egD(\text{Twitter}) \vee E(\text{Twitter})$ 被成功证明。

Question7：

1. 构造决策树

首先，我们计算初始数据集的熵。

初始熵 $H(D)$

我们有以下数据点：

- 正样本 (3 个)：(0,0)，(1,0)，(1,1)
- 负样本 (2 个)：(-1,0)，(2,1)，(2,-2)

初始熵 $H(D)$ 计算：

$$H(D) = -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right)$$

$$H(D) = -(0.6\log_2 0.6 + 0.4\log_2 0.4)$$

我们计算：

$$0.6\log_2 0.6 \approx 0.6 \times (-0.736965) = -0.442179$$

$$0.4\log_2 0.4 \approx 0.4 \times (-1.321928) = -0.528771$$

$$H(D) = -(-0.442179 - 0.528771) = 0.97095$$

2. 计算信息增益

我们要计算不同分裂点的条件熵，并选择信息增益最大的分裂点。

分裂点 f_1

假设我们选择 f_1 的不同值作为分裂点：-1, 0, 1。

分裂点 $f_1 = 0$

数据分成两组：

- 左边：$(-1, 0)$，$(0, 0)$
- 右边：$(1, 0)$，$(2, 1)$，$(2, -2)$

左边熵：

左边组有一个正样本和一个负样本。

$$H(D_{left}) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1$$

右边熵：

右边组有两个正样本和一个负样本。

$$H(D_{right}) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right)$$

$$\frac{2}{3}\log_2\frac{2}{3} \approx \frac{2}{3} \times (-0.584963) = -0.389975$$

$$\frac{1}{3}\log_2\frac{1}{3} \approx \frac{1}{3} \times (-1.584963) = -0.528320$$

$$H(D_{right}) = -(-0.389975 - 0.528320) = 0.918295$$

条件熵:

$$H(D|f_1 = 0) = \frac{2}{5} \cdot 1 + \frac{3}{5} \cdot 0.918295 = 0.950$$

信息增益:

$$IG(f_1 = 0) = H(D) - H(D|f_1 = 0) = 0.97095 - 0.950 = 0.02095$$

分裂点 $f_1 = 1$

数据分成两组:

- 左边: $(0, 0)$, $(1, 0)$, $(-1, 0)$
- 右边: $(2, 1)$, $(2, -2)$

左边熵:

左边组有两个正样本和一个负样本。

$$H(D_{left}) = -\left(\frac{2}{3} \log_2 \frac{2}{3} + \frac{1}{3} \log_2 \frac{1}{3}\right) = 0.918295$$

右边熵:

右边组有一个正样本和一个负样本。

$$H(D_{right}) = -\left(\frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2}\right) = 1$$

条件熵:

$$H(D|f_1 = 1) = \frac{3}{5} \cdot 0.918295 + \frac{2}{5} \cdot 1 = 0.950$$

信息增益：

$$IG(f_1=-1)=H(D)-H(D|f_1=-1)=0.97095-0.97095=0$$

选择最佳分裂点

比较所有分裂点的信息增益，我们发现 $f_1=0$ 和 $f_1=1$ 的信息增益相同，均为 0.02095。我们可以选择 $f_1=0$ 作为第一个分裂点。

构造决策树

第一个节点： $f_1=0$

- 左子树（$f_1 \le 0$）：$(-1,0)$，$(0,0)$
- 右子树（$f_1 > 0$）：$(1,0)$，$(2,1)$，$(2,-2)$

左子树进一步分裂

对于左子树，只有两个数据点：(-1,0) 和 (0,0)，我们可以直接标记为叶子节点。

- $(-1,0)$：负样本
- $(0,0)$：正样本

右子树进一步分裂

对于右子树，有三个数据点：$(1,0)$，$(2,1)$，$(2,-2)$。我们继续选择 f_2 或 f_1 作为分裂点。

通过类似的步骤，我们可以计算信息增益，并选择最佳分裂点，直到叶子节点为止。

3. 对新数据点 (1, -1.01) 进行预测

根据构造的决策树，通过决策树路径进行预测，判断新数据点属于正样本还是负样本。

由于具体计算过程较为复杂，请根据上述方法进行详细计算，最终预测新数据点的类别。

Question8：

若当前模型为软间隔支持向量机（Soft Margin SVM），现额外有一个点不能被 SVM 正确分类且远离决策边界，如果将该点加入到训练集重新训练，原 SVM 的决策边界会受到影响吗？为什么？

解答：

软间隔支持向量机通过引入松弛变量允许一些误分类，以便最大化决策边界的间隔。以下是详细解释：

1. **远离决策边界**：既然新点远离决策边界，它对支持向量的影响很小，因为支持向量机的决策边界主要由支持向量决定，这些支持向量是那些距离决策边界最近的点。
2. **不能被正确分类**：这个新点的引入可能会增加一些误分类的代价，但由于它远离决策边界，不会成为新的支持向量。
3. **软间隔 SVM**：软间隔 SVM 允许一些误分类以获得更好的分类效果。由于新点远离决策边界，它对支持向量的影响较小。

综上，虽然新点不能被正确分类且远离决策边界，但因为它不影响当前的支持向量集，因此对决策边界的影响不大。

问题 2

若当前模型为线性最小二乘分类（Least Squares Classification），现额外有一个点不能被该模型正确分类且远离决策边界，如果将该点加入到训练集重新训练，最小二乘分类的决策边界会受到影响吗？为什么？

解答：

线性最小二乘分类是一种基于最小化误差平方和的分类方法。以下是详细解释：

1. **远离决策边界**：即使新点远离决策边界，最小二乘分类会试图最小化所有点的误差平方和。这意味着每个数据点都会对模型的参数产生影响。
2. **误差平方和**：新点如果误差较大，会显著增加整体的误差平方和。因此，即使新点远离决策边界，它依然会对决策边界的参数产生较大的影响。
3. **模型重训练**：加入新点后，最小二乘分类的目标是最小化所有数据点的误差平方和，因此会调整决策边界以尽可能减少整体误差。

综上，虽然新点远离决策边界，但由于最小二乘分类最小化整体误差平方和的性质，这个新点会对模型的参数产生较大的影响，进而改变决策边界。

问题 3

从问题 1 和 2 中，哪一个模型受到这些样本的影响更大？请解释原因。（提示：可以从损失函数的角度考虑）

解答：

比较软间隔支持向量机（SVM）和线性最小二乘分类（Least Squares Classification）：

1. **软间隔 SVM**：
   - 目标是最大化间隔，同时允许一定的误分类，通过引入松弛变量来处理误分类。
   - 误分类的点对模型影响有限，特别是那些远离决策边界的点。
2. **线性最小二乘分类**：
   - 目标是最小化所有点的误差平方和。
   - 任何点的误差都会对整体误差平方和产生影响，尤其是那些误差较大的点（即远离决策边界的点）。

因此，从损失函数的角度来看，**线性最小二乘分类（Least Squares Classification）** 受新点影响更大。因为它最小化的是所有数据点的误差平方和，任何一个点的较大误差都会显著影响整体误差，而 SVM 的目标是最大化间隔，对远离决策边界的点不敏感。

现额外有一个点能被 SVM 正确分类且远离决策边界，如果将该点加入到训练集重新训练，SVM 的决策边界会受到影响吗？为什么？

解答：

对于硬间隔支持向量机，决策边界主要由支持向量决定。这些支持向量是那些位于间隔边界上的点。因为新增的点能被正确分类且远离决策边界，所以它不会成为新的支持向量，对决策边界的影响很小甚至没有影响。

3. 现额外有一个点不能被当前的 SVM 正确分类且远离决策边界，如果将该点加入到训练集并将硬间隔支持向量机扩展为软间隔支持向量机（Soft Margin SVM），SVM 的决策边界会受到影响吗？为什么?

解答：

在软间隔支持向量机中，允许一些误分类以获得更好的分类效果。虽然这个新点远离决策边界且不能被正确分类，它会引入松弛变量，增加误分类的代价。

具体影响如下：

- **误分类代价**：软间隔 SVM 会引入一个松弛变量来处理误分类的点，这个变量会增加目标函数的值。
- **决策边界**：由于增加了误分类的代价，SVM 会调整决策边界以平衡间隔最大化和误分类代价最小化。这可能会导致决策边界发生变化。

因此，尽管新点远离决策边界，软间隔 SVM 仍会因为引入松弛变量而调整决策边界，使其对新点的误分类进行某种补偿。

2021

Question6

一阶逻辑公式表达

首先，我们需要将自然语言语句转换为一阶逻辑公式。定义以下谓词：

- $R(x)$: x是机器人
- $S(x)$: x是聪明的
- $F(x,y)$: x对y友好
- $M(x,y)$: x是y的主人
- $H(x)$: x是人

1. **所有的机器人都是聪明的。**

$$\forall x(R(x) \rightarrow S(x))$$

2. **有些机器人是聪明的。**

$$\exists x(R(x) \wedge S(x))$$

3. **Robbie是一个机器人。**

$$R(\text{Robbie})$$

4. **所有的机器人都对它的主人友好。**

$$\forall x(R(x) \rightarrow \forall y(M(y,x) \rightarrow F(x,y)))$$

5. **Bob是Robbie的主人。**

$$M(\text{Bob}, \text{Robbie})$$

6. **有些拥有机器人的人是聪明的。**

$$\exists x(H(x) \wedge \exists y(R(y) \wedge M(x,y)) \wedge S(x))$$

归结方法证明“Robbie对Bob友好”

已知：

- $R(\text{Robbie})$
- $M(\text{Bob}, \text{Robbie})$
- $\forall x(R(x) \rightarrow \forall y(M(y,x) \rightarrow F(x,y)))$

证明：

1. 从$\forall x(R(x) \rightarrow \forall y(M(y,x) \rightarrow F(x,y)))$，由于$R(\text{Robbie})$，可以推出$\forall y(M(y, \text{Robbie}) \rightarrow F(\text{Robbie}, y))$。
2. 由于$M(\text{Bob}, \text{Robbie})$，根据上一步，可以得到$F(\text{Robbie}, \text{Bob})$。

因此，"Robbie对Bob友好"得证。

基于知识库证明“Bob是聪明的”

目前的知识库和已知事实中，没有直接关于Bob是否聪明的信息。虽然我们知道有些拥有机器人的人是聪明的，但由于没有直接的信息表明Bob拥有机器人，我们无法直接证明“Bob是聪明的”。

因此，基于当前的知识库，**不能证明“Bob是聪明的”**。

Question7:

1. 计算 $P(\neg A, B, \neg C, D)$

根据贝叶斯网络的结构和条件概率表（CPT），我们可以使用链式法则来计算联合概率。

$$P(\neg A, B, \neg C, D) = P(\neg A) \cdot P(B) \cdot P(\neg C|\neg A) \cdot P(D|\neg A, B)$$

- $P(\neg A) = 1 - P(A) = 1 - 0.1 = 0.9$
- $P(B) = 0.5$
- $P(\neg C|\neg A) = 0.8$（因为 $P(C|\neg A) = 0.2$）
- $P(D|\neg A, B) = 0.7$

代入公式：

$$P(\neg A, B, \neg C, D) = 0.9 \times 0.5 \times 0.8 \times 0.7 = 0.252$$

2. 判断 A 与 B 在给定 D 时，是否条件独立?

条件独立性可以通过 d-separation 判断。在给定 D 的情况下，路径 $A \rightarrow D \leftarrow B$ 被 D 封闭。因此，A 和 B 在给定 D 的情况下不条件独立。

3. 计算 $P(A|B, C, D)$

根据贝叶斯定理和链式法则，我们需要先计算联合概率 $P(A, B, C, D)$ 和 $P(B, C, D)$:

$$P(A|B, C, D) = \frac{P(A, B, C, D)}{P(B, C, D)}$$

首先计算 $P(A, B, C, D)$:

$$P(A, B, C, D) = P(A) \cdot P(B) \cdot P(C|A) \cdot P(D|A, B)$$

- $P(A) = 0.1$
- $P(B) = 0.5$
- $P(C|A) = 0.7$
- $P(D|A, B) = 0.9$

代入公式:

$$P(A,B,C,D)=0.1\times0.5\times0.7\times0.9=0.0315$$

然后计算 $P(B,C,D)$（使用所有可能的 A 的取值）：

$$P(B,C,D)=P(A,B,C,D)+P(\neg A,B,C,D)$$

再计算 $P(\neg A,B,C,D)$:

$$P(\neg A,B,C,D)=P(\neg A)\cdot P(B)\cdot P(C|\neg A)\cdot P(D|\neg A,B)$$

- $P(\neg A)=0.9$
- $P(C|\neg A)=0.2$
- $P(D|\neg A,B)=0.7$

代入公式:

$$P(\neg A,B,C,D)=0.9\times0.5\times0.2\times0.7=0.063$$

所以,

$$P(B,C,D)=0.0315+0.063=0.0945$$

最后计算:

$$P(A|B,C,D)=\frac{0.0315}{0.0945}\approx0.3333$$

4. 计算 $P(A|\neg D)$

根据贝叶斯定理和边缘化计算:

$$P(A|\neg D)=\frac{P(A,\neg D)}{P(\neg D)}$$

首先计算 $P(A, \neg D)$:

$$P(A, \neg D) = P(A) \cdot P(\neg D|A)$$

- $P(\neg D|A)$ 需要边缘化 B:

$$P(\neg D|A) = P(\neg D|A, B) \cdot P(B) + P(\neg D|A, \neg B) \cdot P(\neg B)$$

- $P(\neg D|A, B) = 1 - P(D|A, B) = 1 - 0.9 = 0.1$
- $P(\neg D|A, \neg B) = 1 - P(D|A, \neg B) = 1 - 0.6 = 0.4$

所以,

$$P(\neg D|A) = 0.1 \times 0.5 + 0.4 \times 0.5 = 0.25$$

代入公式:

$$P(A, \neg D) = 0.1 \times 0.25 = 0.025$$

然后计算 $P(\neg D)$ (使用所有可能的 A 的取值) :

$$P(\neg D) = P(A, \neg D) + P(\neg A, \neg D)$$

再计算 $P(\neg A, \neg D)$:

$$P(\neg A, \neg D) = P(\neg A) \cdot P(\neg D|\neg A)$$

- $P(\neg D|\neg A)$ 需要边缘化 B:

$$P(\neg D|\neg A) = P(\neg D|\neg A, B) \cdot P(B) + P(\neg D|\neg A, \neg B) \cdot P(\neg B)$$

- $P(\neg D|\neg A, B) = 1 - P(D|\neg A, B) = 1 - 0.7 = 0.3$
- $P(\neg D|\neg A, \neg B) = 1 - P(D|\neg A, \neg B) = 1 - 0.3 = 0.7$

所以,

$$P(\neg D|\neg A) = 0.3 \times 0.5 + 0.7 \times 0.5 = 0.5$$

代入公式:

$$P(\neg A, \neg D) = 0.9 \times 0.5 = 0.45$$

所以,

$$P(\neg D) = 0.025 + 0.45 = 0.475$$

最后计算:

$$P(A|\neg D) = \frac{0.025}{0.475} \approx 0.0526$$

结论

1. $P(\neg A, B, \neg C, D) = 0.252$
2. A 和 B 在给定 D 时不条件独立。
3. $P(A|B, C, D) \approx 0.3333$
4. $P(A|\neg D) \approx 0.0526$

> d-separation 是贝叶斯网络中用于判断变量之间是否条件独立的一种方法。通过d-separation，可以确定在给定一些条件变量的情况下，其他变量是否条件独立。
>
> 贝叶斯网络中的变量通过有向无环图（DAG）来表示，d-separation 的基本思想是通过图结构来判断信息是否能够通过某些路径传播。具体地，d-separation 定义了三种路径的情况，来判断变量之间的条件独立性。
>
> ### 三种路径类型
>
> 1. **串联结构（Serial/Chain Structure）**：
>   - 形式： A→B→CA \rightarrow B \rightarrow CA→B→C 或 A←B←CA \leftarrow B \leftarrow CA←B←C
>   - 条件独立性：给定中间节点 BBB 后，AAA 和 CCC 条件独立。
>   - 示例：在路径 A→B→CA \rightarrow B \rightarrow CA→B→C 中，如果知道 BBB，则 AAA 和 CCC 条件独立。
> 2. **分叉结构（Diverging Structure）**：
>   - 形式： A←B→CA \leftarrow B \rightarrow CA←B→C
>   - 条件独立性：给定父节点 BBB 后，AAA 和 CCC 条件独立。
>   - 示例：在路径 A←B→CA \leftarrow B \rightarrow CA←B→C 中，如果知道 BBB，则 AAA 和 CCC 条件独立。
> 3. **聚合结构（Converging Structure，亦称v-结构或Collider）**：
>   - 形式： A→B←CA \rightarrow B \leftarrow CA→B←C
>   - 条件独立性：给定中间节点 BBB 的任何后代后，AAA 和 CCC 不条件独立。如果不考虑 BBB 或其后代，AAA 和 CCC 条件独立。
>   - 示例：在路径 A→B←CA \rightarrow B \leftarrow CA→B←C 中，如果不知道 BBB 及其后代，AAA 和 CCC 条件独立。
>
> ## d-separation 的判断步骤
>
> 1. **确定路径**：在贝叶斯网络中找到从一个变量到另一个变量的所有可能路径。
> 2. **检查每条路径**：根据三种路径类型的规则，检查路径是否被给定条件变量阻断。
> 3. **判断条件独立性**：如果所有路径都被阻断，则变量条件独立；如果至少有一条路径未被阻断，则变量不条件独立。

2020:

判断题

> ## 广度优先搜索是代价一致搜索的特例。
>
> - **正确**。广度优先搜索可以看作是一致代价搜索的特例，其中每一步的代价都是相同的，因此代价一致搜索会等价于广度优先搜索。

2. 深度优先搜索扩展的结点个数不会少于使用可采纳启发式的 A* 搜索算法。

- **错误**。深度优先搜索在最坏情况下可能扩展更多的节点，因为它可能探索到搜索空间的很深的部分，而A*算法利用可采纳的启发式函数能够更有效地引导搜索，从而可能扩展更少的节点。

深度优先搜索（DFS）和 A* 搜索算法在扩展节点个数上的比较需要考虑算法的特性。

- **深度优先搜索（DFS）**：
  - DFS 不考虑路径的代价，只是沿着路径不断深入，直到找到一个解决方案或者达到最大深度。如果路径很深且解在较浅的深度上，DFS 可能会扩展大量不必要的节点。
  - DFS 的优点是占用内存较少，但可能会陷入深层的搜索，扩展大量节点。
- **A* 搜索算法**:
  - A* 搜索利用启发式函数 h(n)h(n)h(n) 来估计从节点 nnn 到目标的代价，并使用总代价 f(n)=g(n)+h(n)f(n) = g(n) + h(n)f(n)=g(n)+h(n) 来引导搜索，其中 g(n)g(n)g(n) 是从起点到 nnn 的实际代价。
  - 如果启发式函数是可采纳的（即永远不会超过实际代价），A* 搜索将保证找到最优解，同时通常会扩展较少的节点，因为启发式函数可以有效地引导搜索。

综上所述，深度优先搜索（DFS）扩展的节点个数可能会远多于使用可采纳启发式的 A* 搜索算法，特别是在启发式函数良好的情况下。因此，原命题"深度优先搜索扩展的结点个数不会少于使用可采纳启发式的 A* 搜索算法"是 **错误** 的。

3. 在命题逻辑中，以下关系成立：A⇔B⊨¬A∨BA \Leftrightarrow B \models \neg A \lor BA⇔B⊨¬A∨B。

- **正确**。如果 A⇔BA \Leftrightarrow BA⇔B 成立，那么 AAA 和 BBB 在逻辑上是等价的，这意味着 ¬A∨B\neg A \lor B¬A∨B 是一个重言式，因此 A⇔B⊨¬A∨BA \Leftrightarrow B \models \neg A \lor BA⇔B⊨¬A∨B 成立。

4. 越复杂的模型越能有效地发现数据中的规律。

- **错误**。过于复杂的模型可能会过拟合数据，捕捉到噪声而不是实际的规律。因此，模型的复杂度需要和数据的复杂度匹配，过于复杂的模型不一定能有效地发现数据中的规律。

5. 主成分分析 (PCA) 可以起到去噪的作用。

- **正确**。PCA 通过提取数据中的主要成分并忽略次要成分，可以有效地去除噪声，从而起到去噪的作用。

选择题：

- **深度优先搜索**：不保证找到最优解，因为它可能会陷入很深的子树中而不检查更浅的解。
- **广度优先搜索**：保证找到最优解，因为它逐层检查所有可能的解。
- **迭代深化搜索**：结合了深度优先搜索和广度优先搜索的优点，保证找到最优解。
- **遗传算法**：不保证找到最优解，因为它是一种启发式方法，依赖于随机过程。

因此，必然可以得到最优解的算法是广度优先搜索和迭代深化搜索。

答案：C

2. 以下哪一阶逻辑语句和语义上正确地表达了“每一个孩子都爱自己的母亲或父亲”：

A. ∀x (Child(x)→Loves(x,Mother(x))∨Loves(x,Father(x)))\forall x \ (Child(x) \rightarrow Loves(x, Mother(x)) \lor Loves(x, Father(x)))∀x (Child(x)→Loves(x,Mother(x))∨Loves(x,Father(x)))

B. ∀x (Child(x)∨Loves(x,Mother(x))∨Loves(x,Father(x)))\forall x \ (Child(x) \lor Loves(x, Mother(x)) \lor Loves(x, Father(x)))∀x (Child(x)∨Loves(x,Mother(x))∨Loves(x,Father(x)))

C. ∀x (Child(x)∧(Loves(x,Mother(x))∨Loves(x,Father(x))))\forall x \ (Child(x) \land (Loves(x, Mother(x)) \lor Loves(x, Father(x))))∀x (Child(x)∧(Loves(x,Mother(x))∨Loves(x,Father(x))))

D. ∀x (Child(x)→(Loves(x,Mother(x))∧Loves(x,Father(x))))\forall x \ (Child(x) \rightarrow (Loves(x, Mother(x)) \land Loves(x, Father(x))))∀x (Child(x)→(Loves(x,Mother(x))∧Loves(x,Father(x))))

解答：

- A 选项正确表达了“每一个孩子都爱自己的母亲或父亲”。
- B 选项语义不正确。
- C 选项是合取而不是条件语句，不正确。
- D 选项表达的是“每个孩子都同时爱母亲和父亲”，不正确。

答案：A

3. 搜索下列公理集，其中不含合一的是（ ）：

- A. $p(f(x), z)$ 和 $p(y, f(a))$ 中，无法通过合一化匹配 $f(x)$ 和 $f(a)$。
- B. $p(f(a), f(x))$ 和 $p(y, y)$ 可以通过 $f(x) = f(a)$ 和 $y = f(a)$ 合一化。
- C. $p(x, y)$ 和 $p(f(y), z)$ 可以通过 $x = f(y)$ 和 $y = z$ 合一化。
- D. $p(a, h(g(z)))$ 和 $p(z, h(y))$ 无法合一，因为 $h(g(z))$ 和 $h(y)$ 无法匹配。

答案：A 和 D

4. 搜索引擎会观察用户在返回的搜索结果中点击了哪个网页，通过这种方式学习更好地搜索排序规则。如果搜索引擎学习是针对某个任务 T，随着经验的累积提高性能标准 P，那么在搜索引擎的何种 T 中，任务是什么？

A. 对用户的查询返回一个有序的网页列表

B. 观察用户在返回的搜索结果中点击了哪个网页

C. 搜索引擎返回的结果的准确度

D. 以上皆不是

解答：

观察用户在返回的搜索结果中点击了哪个网页，搜索引擎的任务是对用户的查询返回一个有序的网页列表，从而提高返回结果的准确度。

答案：A

5. 考虑一个三维空间当中心以 wTx+b=0w^T x + b = 0wTx+b=0 为分类超平面的支持向量机，其中 w=(1,2,3)Tw = (1, 2, 3)^Tw=(1,2,3)T，b = 4，以下哪些正例点（即 y = 1）将被正确分类？

A. x=(−2,−3,4)Tx = (-2, -3, 4)^Tx=(−2,−3,4)T

B. x=(−1,−3,0)Tx = (-1, -3, 0)^Tx=(−1,−3,0)T

C. x=(−4,−5,3)Tx = (-4, -5, 3)^Tx=(−4,−5,3)T

D. x=(−4,−3,1)Tx = (-4, -3, 1)^Tx=(−4,−3,1)T

解答：

验证每个点是否满足 wTx+b>0w^T x + b > 0wTx+b>0：

1. x=(−2,−3,4)Tx = (-2, -3, 4)^Tx=(−2,−3,4)T

   wTx+b=1(−2)+2(−3)+3(4)+4=−2−6+12+4=8>0w^T x + b = 1(-2) + 2(-3) + 3(4) + 4 = -2 - 6 + 12 + 4 = 8 > 0wTx+b=1(−2)+2(−3)+3(4)+4=−2−6+12+4=8>0

   正确分类。

2. x=(−1,−3,0)Tx = (-1, -3, 0)^Tx=(−1,−3,0)T

   wTx+b=1(−1)+2(−3)+3(0)+4=−1−6+0+4=−3<0w^T x + b = 1(-1) + 2(-3) + 3(0) + 4 = -1 - 6 + 0 + 4 = -3 < 0wTx+b=1(−1)+2(−3)+3(0)+4=−1−6+0+4=−3<0

   未正确分类。

3. x=(−4,−5,3)Tx = (-4, -5, 3)^Tx=(−4,−5,3)T

   wTx+b=1(−4)+2(−5)+3(3)+4=−4−10+9+4=−1<0w^T x + b = 1(-4) + 2(-5) + 3(3) + 4 = -4 - 10 + 9 + 4 = -1 < 0wTx+b=1(−4)+2(−5)+3(3)+4=−4−10+9+4=−1<0

   未正确分类。

4. x=(−4,−3,1)Tx = (-4, -3, 1)^Tx=(−4,−3,1)T

   wTx+b=1(−4)+2(−3)+3(1)+4=−4−6+3+4=−3<0w^T x + b = 1(-4) + 2(-3) + 3(1) + 4 = -4 - 6 + 3 + 4 = -3 < 0wTx+b=1(−4)+2(−3)+3(1)+4=−4−6+3+4=−3<0

   未正确分类。

Question7:

我们考虑的问题是加权后的最小二乘分类问题:

$$\min_{\mathbf{w}} \sum_i r_i(y_i - \mathbf{w}^T\mathbf{x}_i)^2$$

步骤1:展开目标函数

我们先将目标函数展开:

$$\sum_i r_i(y_i - \mathbf{w}^T\mathbf{x}_i)^2 = \sum_i r_i(y_i^2 - 2y_i\mathbf{w}^T\mathbf{x}_i + (\mathbf{w}^T\mathbf{x}_i)^2)$$

可以写成矩阵形式:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{R} = \mathrm{diag}(r_1, r_2, \ldots, r_n)$$

则目标函数可以写成:

$$\min_{\mathbf{w}} \left(\mathbf{y}^T\mathbf{R}\mathbf{y} - 2\mathbf{y}^T\mathbf{R}\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{R}\mathbf{X}\mathbf{w}\right)$$

步骤2:计算梯度并设为零

为了找到最优的 $\mathbf{w}$,我们需要计算目标函数对 $\mathbf{w}$ 的梯度,并设为零:

$$\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{y}^T\mathbf{R}\mathbf{y} - 2\mathbf{y}^T\mathbf{R}\mathbf{X}\mathbf{w} + \mathbf{w}^T\mathbf{X}^T\mathbf{R}\mathbf{X}\mathbf{w}\right) = 0$$

根据给出的求导规则:

根据给出的求导规则：

$$\frac{\partial}{\partial \mathbf{w}}\left(-2\mathbf{y}^T\mathbf{R}\mathbf{X}\mathbf{w}\right) = -2\mathbf{X}^T\mathbf{R}\mathbf{y}$$

$$\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{w}^T\mathbf{X}^T\mathbf{R}\mathbf{X}\mathbf{w}\right) = 2\mathbf{X}^T\mathbf{R}\mathbf{X}\mathbf{w}$$

将这些导数项加在一起：

$$-2\mathbf{X}^T\mathbf{R}\mathbf{y} + 2\mathbf{X}^T\mathbf{R}\mathbf{X}\mathbf{w} = 0$$

步骤3：求解 $\mathbf{w}$

我们可以解这个方程来得到 $\mathbf{w}$：

$$2\mathbf{X}^T\mathbf{R}\mathbf{X}\mathbf{w} = 2\mathbf{X}^T\mathbf{R}\mathbf{y}$$

消去常数项：

$$\mathbf{X}^T\mathbf{R}\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{R}\mathbf{y}$$

因此，解得：

$$\mathbf{w} = (\mathbf{X}^T\mathbf{R}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}\mathbf{y}$$

结论

加权最小二乘分类问题的解为：

$$\mathbf{w} = (\mathbf{X}^T\mathbf{R}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}\mathbf{y}$$

这个解表明我们通过加权的设计可以最小化误差平方和，从而找到最优的分类向量 $\mathbf{w}$。

Question8：

K-means 聚类算法是一种迭代的分区聚类方法，其目标是通过最小化簇内的平方误差来确定每个数据点的簇分配。算法的主要步骤包括：

1. 初始化 K 个聚类中心。
2. 分配每个数据点到最近的聚类中心。
3. 重新计算每个簇的聚类中心。
4. 重复步骤 2 和 3 直到收敛。

K-means 聚类算法的收敛性证明可以通过以下两个方面来进行：

1. **目标函数单调递减**：
   - K-means 聚类算法在每次迭代中都会使目标函数（即平方误差和）单调递减。
   - 目标函数定义为：

$$J = \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

     其中，μ_i 是第 i 个聚类的中心，C_i 是属于第 i 个聚类的点集。
   - 在每次迭代中，算法分两步进行：
     - **分配步骤**：将每个数据点分配给最近的聚类中心。这一步会使得每个点到其所属簇中心的距离最小化，因此不会增加目标函数的值。
     - **更新步骤**：重新计算每个簇的聚类中心。这一步会使得新的中心是该簇中所有点的几何中心，从而减少簇内的平方误差和。

2. **有限的可能簇划分**：
   - 由于数据集是有限的，并且每个数据点只能属于其中的一个簇，因此可能的簇划分数量是有限的。
   - 每次迭代算法都会选择一种新的簇划分方案，并且由于目标函数的单调递减性，不会出现循

环。

综合以上两点，K-means 聚类算法必然会在有限步内收敛到一个局部最优解，即不再进行任何更新的状态。以下是详细的证明步骤：

证明步骤：

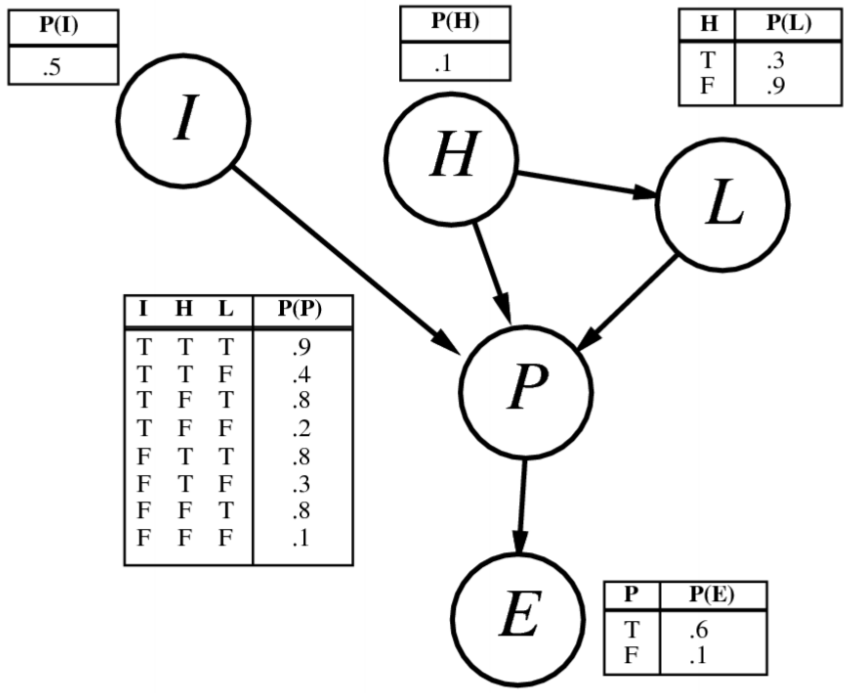
1. **初始化**：
   - 随机选择 K 个聚类中心。
2. **分配步骤**：
   - 对于每个数据点 x_j，找到最近的聚类中心 μ_i，并将 x_j 分配给聚类 C_i。
   - 这一分配步骤不会增加目标函数 J，因为每个点都被分配给了使得平方误差最小的聚类中心。
3. **更新步骤**：
   - 对于每个聚类 C_i，重新计算聚类中心 μ_i，使得 μ_i 成为簇 C_i 内所有点的几何中心。
   - 这一更新步骤也不会增加目标函数 J，因为新的聚类中心是簇内所有点的均值。
4. **单调递减**：
   - 由于分配步骤和更新步骤都不会增加目标函数 J，因此 J 在每次迭代中都单调递减。
5. **有限可能簇划分**：
   - 给定 n 个数据点和 K 个簇，可能的簇划分数量是有限的。因此，不可能无限次迭代而不收敛。
6. **收敛性**：
   - 由于目标函数 J 是单调递减且有下界（目标函数的值不能为负），再加上可能的簇划分数量是有限的，算法必然在有限步内收敛到一个局部最优解。

结论

K-means 聚类算法在每次迭代中都会使目标函数单调递减，并且由于可能的簇划分数量是有限的，因此 K-means 聚类算法可以保证在有限步内收敛到一个局部最优解。

Question6：

六、 （16 分）考虑下图所示的贝叶斯网络，其中包括五个布尔随机变量 *I=Intelligent*（聪明）， *H=Honest*（诚实），*P=Popular*（受欢迎）， *L=LotsOfCampaignFunds*（有大量竞选资金）， *E=Elected*（当选）。



(1) 单选：以下关于该贝叶斯网络结构的那个等式哪一个是<u>不正确</u>的（先不考虑 CPT）?（　　）

A. $\mathbf{P}(I, L) = \mathbf{P}(I)\mathbf{P}(L)$

B. $\mathbf{P}(E \mid P, L) = \mathbf{P}(E \mid P, L, H)$

C. $\mathbf{P}(P \mid I, H) = \mathbf{P}(P \mid I, H, L)$

(2) 计算 $P(i, h, \neg l, p, \neg e)$

(3) 计算一个人在他是诚实的，拥有的竞选资金很少，当选成功等前提下，他是聪明的概率有多大。

问题 1

单选：以下关于该贝叶斯网络结构的那个等式哪个是不正确的?

A. $P(I, L) = P(I)P(L)$

B. $P(E \mid P, L) = P(E \mid P, L, H)$

C. $P(P \mid I, L, H) = P(P \mid I, H, L)$

解答：

我们分析每个选项是否符合贝叶斯网络的条件独立性原则：

A. $P(I, L) = P(I)P(L)$

- 根据贝叶斯网络结构，I 和 L 之间没有直接的边，但是它们之间并没有条件独立的说明。贝叶斯网络表示的是条件独立性，而不是完全独立性。因此这个等式是不正确的。

B. $P(E \mid P, L) = P(E \mid P, L, H)$

- 根据贝叶斯网络结构，E 直接依赖于 P。即使考虑 L 和 H，它们对 E 的影响是通过 P 间接实现的。因此这个等式是正确的。

C. $P(P \mid I, L, H) = P(P \mid I, H, L)$

- 这个等式是正确的，因为条件概率的顺序不影响条件概率的值。

所以，答案是：A. $P(I, L) = P(I)P(L)$

解答：

根据贝叶斯网络的结构和给定的条件概率表（CPT），我们可以计算：

$$P(i, h, \neg l, p, \neg e) = P(i) \cdot P(h) \cdot P(\neg l \mid h) \cdot P(p \mid i, h, \neg l) \cdot P(\neg e \mid p)$$

1. $P(i) = 0.5$
2. $P(h) = 0.1$
3. $P(\neg l \mid h) = 1 - P(l \mid h) = 1 - 0.9 = 0.1$
4. $P(p \mid i, h, \neg l)$：
   - 查表：当 $I = T$，$H = T$，$L = F$ 时，$P = T$ 的概率为 0.4。
5. $P(\neg e \mid p) = 1 - P(e \mid p) = 1 - 0.6 = 0.4$

把这些值代入：

$$P(i, h, \neg l, p, \neg e) = 0.5 \cdot 0.1 \cdot 0.1 \cdot 0.4 \cdot 0.4 = 0.0008$$

问题 3

计算一个人他是诚实的，拥有的竞选资金很少，当选成功的条件下，他是聪明的概率有多大。

即计算 $P(I \mid H, \neg L, E)$

解答：

根据贝叶斯定理，我们有：

$$P(I \mid H, \neg L, E) = \frac{P(I, H, \neg L, E)}{P(H, \neg L, E)}$$

我们需要计算 $P(I, H, \neg L, E)$ 和 $P(H, \neg L, E)$。

首先计算 $P(I, H, \neg L, E)$：

首先计算 $P(I, H, \neg L, E)$:

$$P(I, H, \neg L, E) = \sum_p P(I, H, \neg L, E, p)$$

$$= \sum_p P(I) \cdot P(H) \cdot P(\neg L \mid H) \cdot P(p \mid I, H, \neg L) \cdot P(E \mid p)$$

分 p 为真和假的两种情况:

1. 当 $p = T$ 时:

$$P(I, H, \neg L, E, p = T) = 0.5 \cdot 0.1 \cdot 0.1 \cdot 0.4 \cdot 0.6 = 0.0012$$

2. 当 $p = F$ 时:

$$P(I, H, \neg L, E, p = F) = 0.5 \cdot 0.1 \cdot 0.1 \cdot 0.6 \cdot 0.1 = 0.0003$$

所以,

$$P(I, H, \neg L, E) = 0.0012 + 0.0003 = 0.0015$$

然后计算 $P(H, \neg L, E)$:

$$P(H, \neg L, E) = \sum_i P(I = i, H, \neg L, E)$$

同样分 $I = T$ 和 $I = F$ 两种情况计算:

1. 当 $I = T$ 时，已经计算过 $P(I, H, \neg L, E) = 0.0015$

2. 当 $I = F$ 时:

$$P(I = F, H, \neg L, E) = \sum_p P(I = F, H, \neg L, E, p)$$

分 p 为真和假的两种情况：

- 当 $p = T$ 时：

$$P(I = F, H, \neg L, E, p = T) = 0.5 \cdot 0.1 \cdot 0.1 \cdot 0.8 \cdot 0.6 = 0.0024$$

- 当 $p = F$ 时：

$$P(I = F, H, \neg L, E, p = F) = 0.5 \cdot 0.1 \cdot 0.1 \cdot 0.2 \cdot 0.1 = 0.0001$$

所以，

$$P(I = F, H, \neg L, E) = 0.0024 + 0.0001 = 0.0025$$

最后，

$$P(H, \neg L, E) = P(I = T, H, \neg L, E) + P(I = F, H, \neg L, E) = 0.0015 + 0.0025 = 0.004$$

所以，

$$P(I \mid H, \neg L, E) = \frac{P(I, H, \neg L, E)}{P(H, \neg L, E)} = \frac{0.0015}{0.004} = 0.375$$

结论

1. **不正确的等式是**：A. $P(I, L) = P(I)P(L)$
2. **计算** $P(i, h, \neg l, p, \neg e)$：0.0008
3. **计算** $P(I \mid H, \neg L, E)$：0.375