第三次作业（贝叶斯网络）

彭煜峰　PB22051087

2024 年 6 月 9 日

本次作业需独立完成，不允许任何形式的抄袭行为，如被发现会有相应惩罚。在上方修改你的姓名学号，说明你同意本规定。

问题 1：概率推断（25%）

a. 画图（3 分）

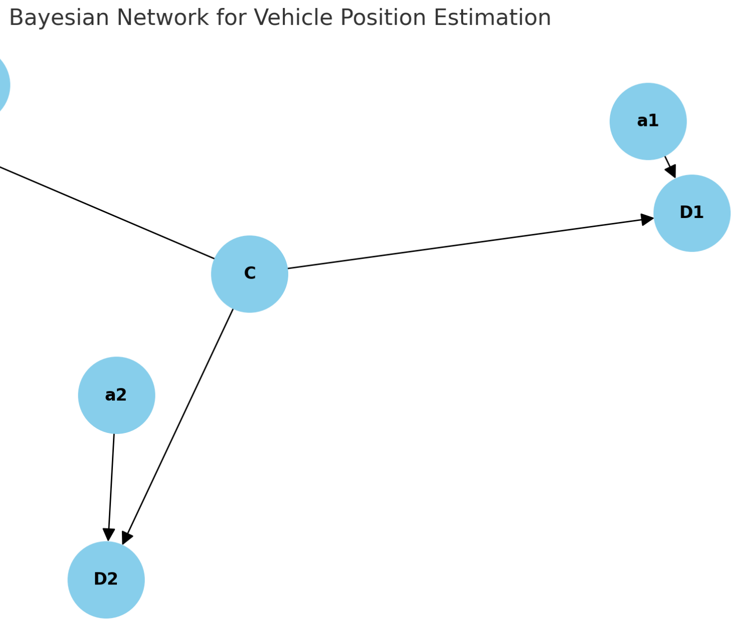

图 1: 贝叶斯网络

b. 计算（5 分）

根据图 1 可以得出联合分布为：

$$
\begin{aligned}
&P(C=c, D_1=d_1, D_2=d_2, D_3=d_3)\\
&=P(C=c)\cdot P(D_1=d_1|C=c,a_1)\cdot P(D_2=d_1|C=c,a_2)\cdot P(D_3=d_3|C=c,a_3)
\end{aligned}
$$

由于 $D_t \sim \mathcal{N}(\|a_t - C_t\|, \sigma^2)$，上式可以展开为：

$$P(C=c, D_1=d_1, D_2=d_2, D_3=d_3) = \frac{P(C=c)}{(2\pi\sigma^2)^{3/2}} \cdot \exp\left\{-\frac{\sum_{i=1}^{3}(d_i - \|c-a_i\|)^2}{2\sigma^2}\right\}$$

c. 证明（5 分）

证明. 根据图 1 表示的贝叶斯网络，车辆的位置 C_t 是一个独立的随机变量，可作为样本空间的一个完备事件群。

而 D_t 只与 C_t 有关，若作出观察 $D_t = d_t$，由贝叶斯定理可计算后验概率：

$$\begin{aligned}&P(C_t=c_t|D_1=d_1,\cdots,D_t=d_t)\\=&\frac{P(C_t=c_t|D_1=d_1,\cdots,D_t=d_{t-1})\cdot p(D_t=d_t|C_t=c_t)}{\sum_{c'\in\{c_t\}}P(C_t=c'|D_1=d_1,\cdots,D_{t-1}=d_{t-1})\cdot p(D_t=d_t|C_t=c')}\end{aligned}$$

去掉上式的归一化因子：

$$P(C_t=c_t|D_1=d_1,\cdots,D_t=d_t) \propto P(C_t=c_t|D_1=d_1,\cdots,D_t=d_{t-1})\cdot p(D_t=d_t|C_t=c_t)$$

又因为假设 $C = c_t = c$ 为常数，因此：

$$P(C=c|D_1=d_1,\cdots,D_t=d_t) \propto P(C=c|D_1=d_1,\cdots,D_t=d_{t-1})\cdot p(D_t=d_t|C=c)$$

□

d. 编程（12 分）

```python
def observe(self, agentX: int, agentY: int, observedDist: float) -> None:
        # BEGIN_YOUR_CODE (our solution is 7 lines of code, but don't worry if you
            deviate from this)
        for row in range(self.belief.numRows):
            for col in range(self.belief.numCols):
                dist = math.sqrt((util.colToX(col) - agentX) ** 2 + (util.rowToY(row) -
                    agentY) ** 2)
                prob_distr = util.pdf(dist, Const.SONAR_STD, observedDist)
                self.belief.setProb(row, col, self.belief.getProb(row, col) * prob_distr
                    )
        self.belief.normalize()
        # END_YOUR_CODE
```

问题 2: 转移概率（25%）

a. 画图（3 分）

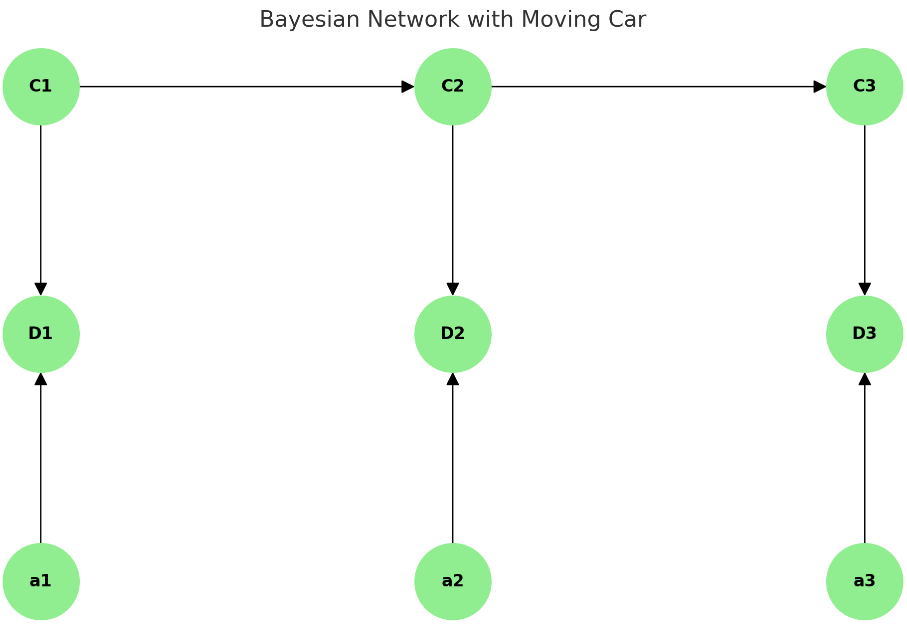

图 2: 贝叶斯网络

b. 计算（5 分）

根据图 2，可以得出：

$$
\begin{aligned}
&P(C_1=c_1, C_2=c_2, C_3=c_3, D_1=d_1, D_2=d_2, D_3=d_3)\\
=&P(C_1=c_1)\times P(C_2=c_2|C_1=c_1)\times P(C_3=c_3|C_2=c_2)\\
&\times P(D_1=d_1|C_1=c_1,a_1)\times P(D_2=d_2|C_2=c_2,a_2)\times P(D_3=d_3|C_3=c_3,a_3)
\end{aligned}
$$

同理，可根据 $D_t \sim \mathcal{N}(\|a_t - C_t\|, \sigma^2)$ 将上式展开：

$$
\begin{aligned}
&P(C_1=c_1, C_2=c_2, C_3=c_3, D_1=d_1, D_2=d_2, D_3=d_3)\\
=&\frac{P(C_1=c_1)\cdot P(C_2=c_2|C_1=c_1)\cdot P(C_3=c_3|C_2=c_2)}{(2\pi\sigma^2)^{3/2}}\cdot \exp\left\{-\frac{\sum_{i=1}^{3}(d_i-\|c_i-a_i\|)^2}{2\sigma^2}\right\}
\end{aligned}
$$

c. 证明（5 分）

证明. 根据条件概率定义：

$$
P(C_{t+1}=c_{t+1}|D_1=d_1,\cdots,D_t=d_t)=\frac{P(C_{t+1}=c_{t+1},D_1=d_1,\cdots,D_t=d_t)}{P(D_1=d_1,\cdots,D_t=d_t)}
$$

再由全概率公式：

$$
P(C_{t+1}=c_{t+1},D_1=d_1,\cdots,D_t=d_t)=\sum_{c_t}p(c_{t+1}|c_t)\cdot P(C_t=c_t|D_1=d_1,\cdots,D_t=d_t)
$$

而 $P(D_1=d_1,\cdots,D_t=d_t)$ 为归一化因子，从而得出：

$$P(C_{t+1}=c_{t+1}|D_1=d_1,\cdots,D_t=d_t)\propto\sum_{c_t}p(c_{t+1}|c_t)\cdot P(C_t=c_t|D_1=d_1,\cdots,D_t=d_t)$$

□

d. 编程（5 分）

```
def elapseTime(self) -> None:
        if self.skipElapse: ### ONLY FOR THE GRADER TO USE IN Problem 1
            return
        # BEGIN_YOUR_CODE (our solution is 7 lines of code, but don't worry if you
            deviate from this)
        newBelief=util.Belief(self.belief.numRows,self.belief.numCols,value=0.0)
        for oldTile,newTile in self.transProb:
            value=self.transProb[(oldTile,newTile)]
            if value>0:
                newBelief.addProb(newTile[0],newTile[1],self.belief.getProb(oldTile[0],
                    oldTile[1])*value)
        newBelief.normalize()
        self.belief=newBelief
        # END_YOUR_CODE
```

问题 3：是哪辆车？(30%)

a. 计算（8 分）

由题意可知，所求条件概率为一个后验概率，因此根据贝叶斯定理：

$$p(C_{11}=c_{11},C_{12}=c_{12}|E_1=e_1)=\frac{p(E_1=e_1|C_{11}=c_{11},C_{12}=c_{12})\cdot p(C_{11}=c_{11},C_{12}=c_{12})}{p(E_1=e_1)}$$

其中

$$\begin{aligned}&p(E_1=e_1|C_{11}=c_{11},C_{12}=c_{12})\\=&p(D_{11}=e_{11},D_{12}=e_{12}|C_{11}=c_{11},C_{12}=c_{12})+p(D_{12}=e_{11},D_{11}=e_{12}|C_{11}=c_{11},C_{12}=c_{12})\\=&p_N(e_{11},\|a_1-c_{11}\|_2,\sigma^2)+p_N(e_{11},\|a_2-c_{12}\|_2,\sigma^2)\end{aligned}$$

因此

$$p(C_{11}=c_{11},C_{12}=c_{12}|E_1=e_1)\propto\left[p_N(e_{11},\|a_1-c_{11}\|_2,\sigma^2)+p_N(e_{11},\|a_2-c_{12}\|_2,\sigma^2)\right]p(c_{11})p(c_{12})$$

b. 计算（8 分）

辅助变量： 变量 z_t 表示在时间步 t 时车辆位置列表 c_t 和感测到的位置列表 e_t 之间的索引偏移量：

$$z_t\in\{0,1,2,\cdots,k-1\},\quad k\text{ 为车辆的数量}$$

概率计算： 根据 z_t 的定义：

$$c_{t-1} = g^{-1}(e_{t-1}, z_{t-1})$$

其中函数 g 为 c_t 按 z_t 偏移的映射。

因此条件概率：

$$p(c_t|c_{t-1}) = p(c_t|c_{t-1} = g^{-1}(e_{t-1}, z_{t-1}))$$

其中 z_t 均匀分布且独立，概率为 $p(z_t) = \frac{1}{k}$。因此：

$$\begin{aligned} p(c_t|c_{t-1}) =& \sum_{p=0}^{k-1} p(c_t|g^{-1}(e_{t-1}, p)) \cdot p(z_{t-1} = p) \\ =& \frac{1}{k} \sum_{p=0}^{k-1} p(c_t|g^{-1}(e_{t-1}, p)) \\ =& \frac{1}{k} \sum_{p=0}^{k-1} \prod_{i=1}^{k} p(c_{ti}|g^{-1}(e_{t-1}, p)_i) \end{aligned}$$

且 $p(c_{ti}|g^{-1}(e_{t-1}, z_{t-1})_i)$ 服从车辆独立的转移概率。

c. 建模（10 分）

贝叶斯网络描述： 我们定义以下变量：

1. $C_{ti} \in \mathbb{R}^2$ 表示时间步 t 时车辆 i 的真实位置。
2. E_t 表示时间步 t 的感测到的位置列表，是车辆真实位置的一个随机排列。
3. $Z_t \in \{0, 1, \ldots, K-1\}$ 表示时间步 t 的索引偏移量。

我们有以下条件概率：

1. 车辆位置的转移概率 $p(C_{ti}|C_{(t-1)i})$ 表示车辆 i 在时间步 t 的位置仅依赖于其在时间步 $t-1$ 的位置。
2. 感测到的位置列表 E_t 是真实位置列表 C_t 的随机排列，且均匀分布。

因子图描述

构建因子图：

1. 对于每个时间步 t，创建变量节点 C_t 表示所有车辆在时间步 t 的位置。
2. 创建因子节点表示车辆位置的转移概率 $p(C_t|C_{t-1})$。
3. 创建变量节点 E_t 表示感测到的位置列表。
4. 创建因子节点表示感测到的位置列表是车辆真实位置的随机排列。

定义因子：

1. 位置转移因子：$\phi_t(C_t, C_{t-1}) = \prod_{i=1}^{K} p(C_{ti}|C_{(t-1)i})$。
2. 感测因子：$\psi_t(E_t, C_t, Z_t) = \delta(E_t = g(C_t, Z_t))$，这里 δ 是指示函数，表示 E_t 是 C_t 的随机排列 g 的结果。

算法步骤：

1. **初始化：**设定初始的先验分布 $p(C_1)$。
2. **向前传递：**从时间步 $t=1$ 到 $t=T$，递归计算每个时间步的边缘分布 $p(C_t|e_1,\ldots,e_t)$：

$$p(C_t|e_1,\ldots,e_t) = \sum_{C_{t-1}} p(C_t|C_{t-1})p(C_{t-1}|e_1,\ldots,e_{t-1})$$

3. **结合感测数据：**更新每个时间步的边缘分布以包含感测数据：

$$p(C_t|e_1,\ldots,e_t) \propto p(C_t|e_1,\ldots,e_{t-1})\psi_t(E_t, C_t, Z_t)$$

4. **向后传递：**从时间步 $t=T$ 到 $t=1$，递归计算每个时间步的平滑分布 $p(C_t|e_1,\ldots,e_T)$：

$$p(C_t|e_1,\ldots,e_T) \propto p(C_t|e_1,\ldots,e_t)\sum_{C_{t+1}} p(C_{t+1}|C_t)p(C_{t+1}|e_1,\ldots,e_T)$$

d. 回答问题（4 分）

粒子滤波器的原理是根据每个粒子的概率不同设置新的权重，再根据权重选择点进行计算，而精确计算则要考虑所有的网格。因此前者的时间复杂度低于后者。

问题 4：模型学习（10%）

a. 计算（10 分）

E 步骤： 根据所给出算法，可以写出后验概率计算公式：

$$P(Z1, Z2|X1, X2) \propto P(X1, X2|Z1, Z2)P(Z1, Z2)$$

其中

$$P(X1, X2|Z1, Z2) = P(X1|Z1)P(X2|Z2), \quad P(Z1, Z2) = P(Z1)P(Z2|Z1)$$

M 步骤： 似然函数：

$$L(\theta) = \prod_{i=1}^{N}\sum_{Z1,Z2} P(Z1)P(Z2|Z1)P(X1^{(i)}|Z1)P(X2^{(i)}|Z2)$$

取对数似然函数：

$$\log L(\theta) = \sum_{i=1}^{N}\log\left(\sum_{Z1,Z2} P(Z1)P(Z2|Z1)P(X1^{(i)}|Z1)P(X2^{(i)}|Z2)\right)$$

计算结果： 利用 python 计算得到：

概率	值
$P(Z1 = \text{true})$	0.721
$P(Z1 = \text{false})$	0.279
$P(Z2 = \text{true}\|Z1 = \text{true})$	0.682
$P(Z2 = \text{false}\|Z1 = \text{true})$	0.318
$P(Z2 = \text{true}\|Z1 = \text{false})$	0.187
$P(Z2 = \text{false}\|Z1 = \text{false})$	0.813
$P(X1 = \text{true}\|Z1 = \text{true})$	1.0
$P(X1 = \text{false}\|Z1 = \text{true})$	0.0
$P(X1 = \text{true}\|Z1 = \text{false})$	1.0
$P(X1 = \text{false}\|Z1 = \text{false})$	0.0
$P(X2 = \text{true}\|Z2 = \text{true})$	0.362
$P(X2 = \text{false}\|Z2 = \text{true})$	0.638
$P(X2 = \text{true}\|Z2 = \text{false})$	0.665
$P(X2 = \text{false}\|Z2 = \text{false})$	0.335

表 1: 更新后的概率表

b. 阅读网页（0 分）

EM 算法可以保证收敛到一个稳定点，但是却不能保证收敛到全局的极大值点，因此它是局部最优的算法，当然，如果我们的优化目标是凸的，则 EM 算法可以保证收敛到全局最大值，这点和梯度下降法这样的迭代算法相同。

反馈（10 分）

在每次实验报告的最后欢迎反馈你上这门课的感受，你可以写下任何反馈，包括但不限于以下几个方面：课堂、作业难度和工作量、助教工作等等。

- 这次作代码难度不高，主要是手写证明部分难度较大。
- 题目的符号用的有点乱，感觉不是很清晰。
- 花费时间在 15 个小时左右（累计）。