ROB422/EECS465
Introduction to Algorithm Robotics

HW5 - Probabilistic Robot

Junhao TU

November 28, 2023

Question 1

The axiom $P(a \vee \neg a) = 1$ is true because it represents the certainty that one of the events a or $\neg a$ will occur, as they encompass all possible outcomes. In probability, this is the principle of the Law of Total Probability, which states that the sum of the probabilities of all mutually exclusive and exhaustive outcomes is 1. Since a and $\neg a$ are the only two possible outcomes and they cannot both occur at the same time, their probabilities must sum to 1. Hence, $P(a) + P(\neg a) = 1$, and rearranging gives $P(\neg a) = 1 - P(a)$.

Question 2

a)

$$\begin{aligned} P(\text{toothache}) =& P(\text{toothache}, \text{catch}, \text{cavity}) + P(\text{toothache}, \neg\text{catch}, \text{cavity}) \\ &+ P(\text{toothache}, \text{catch}, \neg\text{cavity}) + P(\text{toothache}, \neg\text{catch}, \neg\text{cavity}) \\ =& \ 0.2 \end{aligned}$$

b)

$$\begin{aligned} P(\text{cavity}) =& P(\text{cavity}, \text{toothache}, \text{catch}) + P(\text{cavity}, \text{toothache}, \neg\text{catch}) \\ &+ P(\text{cavity}, \neg\text{toothache}, \text{catch}) + P(\text{cavity}, \neg\text{toothache}, \neg\text{catch}) \\ =& \ 0.2 \end{aligned}$$

c)

$$\begin{aligned} P(\text{toothache}|\text{cavity}) =& \frac{P(\text{toothache}, \text{cavity})}{P(\text{cavity})} \\ =& \ 0.6 \end{aligned}$$

d)

$$\begin{aligned} P(\text{cavity}|\text{toothache} \vee \text{catch}) &= \frac{P(\text{cavity}, (\text{toothache} \vee \text{catch}))}{P(\text{toothache} \vee \text{catch})} \\ &= \frac{P(\text{cavity}, \text{toothache}) + P(\text{cavity}, \text{catch}) - P(\text{cavity}, \text{toothache}, \text{catch})}{P(\text{toothache}) + P(\text{catch}) - P(\text{toothache}, \text{catch})} \\ &= 0.4615 \end{aligned}$$

3. (a) $P(x, Y | e) = \frac{P(x, Y, e)}{P(e)}$

$P(x | Y, e) P(Y | e) = \frac{P(x, Y, e)}{P(Y, e)} \cdot \frac{P(Y, e)}{P(e)} = \frac{P(x, Y, e)}{P(e)}$

Therefore, $P(x, Y | e) = P(x | Y, e) P(Y | e)$

(b) $P(Y | x, e) = \frac{P(Y, x, e)}{P(x, e)}$

$\frac{P(x | Y, e) P(Y | e)}{P(x | e)} = \frac{\frac{P(x, y, e)}{P(y, e)} \frac{P(y, e)}{P(e)}}{\frac{P(x, e)}{P(e)}} = \frac{P(x, y, e)}{P(x, e)}$

Therefore, $P(Y | x, e) = \frac{P(x | Y, e) P(Y | e)}{P(x | e)}$

4. Let X be any unobserved variable in the network, not in MB(Y). The posterier distribution of X, given all other observed variables is P(X / MB(Y), O) where O represents other observed variables not in MB(Y). By the local Markov property, Y is conditionally independent of X given MB(Y). Therefore, P(X, Y, MB(Y), O) = P(X / MB(Y), O)

It can be observed that the absence of Y does not change the posterior distribution of X. Since MB(Y) already contains all information Y could provide about X.

5. (a) Network C is identified as the network that claims independence between the gene variables of the father, mother, and child.

(b) Network A and B are consistent with the hypothesis about the inheritance of handedness as they show dependencies aligning with genetic inheritance patterns.

(c) Network A provides the best description of the hypothesis, reflecting the direct influence of the parent's genes on the child's gene.

(d) CPT for G child node in network (a):

G_{mother}	G_{father}	$P(G_{child}=l \mid \dots)$	$P(G_{child}=r \mid \dots)$
l	l	$1-m$	m
l	r	0.5	0.5
r	l	0.5	0.5
r	r	m	$1-m$.

(e) $P(G_{child}=l) = \sum P(G_c = l \mid G_f, G_m) P(G_f, G_m)$

$$= (1-m)q^2 + 0.5q(1-q) + 0.5(1-q)q + m(1-q)^2$$

$$= m + q - 2mq$$

(f). Under conditions of genetic equilibrium, the allele frequencies remain constant. Thus. $P(G_{child}=l) = P(G_{father}=l) = P(G_{mother}=l) = q$.

$\therefore\ -2mq + m + q = q \Rightarrow m(1-2q)=0.$

Assuming $m \neq 0$ $\therefore q = \frac{1}{2}$

In fact, most of people is right handedness which conflict with 50% from above calculation. This means that this model is an oversimplification, as environmental factors and multiple genes may influence handedness, and thus the hypothesis of a single gene determining handeness is likely to be incorrect.

6. (a) $P(R_t | U_{1:t}) = \alpha P(U_t | R_t) \sum P(R_t | R_{t-1}) P(R_{t-1} | U_{1:t-1})$

If we end up to a equilibrium state:

$P(R_t | U_{1:t}) = P(R_{t-1} | U_{1:t-1}) = \alpha p \langle 0.9, 0.2\rangle\langle 0.7, 0.3\rangle + (1-p)\langle 0.3, 0.7\rangle$

$= 0.893$

(b) $P(r_{2+k} | U_1, U_2) = 0.3 P(R_{2+k-1} | U_1, U_2) + 0.3 (1 - P(R_{2+k-1} | U_1, U_2))$

$\because$ ~~R(2~~ $P(R_{2+k} | U_1, U_2) = P(R_{2+k-1} | U_1, U_2)$

$\therefore X = 0.7X + 0.3(1-X) \Rightarrow X = 0.5 \quad \therefore P(R_{2+k} | U_1, U_2) = 0.5$

Question 7

Dynamic Bayesian Network

This section presents the formulation of a dynamic Bayesian network (DBN) which the professor can use to filter or predict whether students are getting enough sleep based on a sequence of observations: students sleeping in class and students having red eyes.

Initial State Probability (Prior Probability):

Enough Sleep	Probability
True	0.7
False	0.3

Transition Model:

Enough Sleep (t-1)	Enough Sleep (t)	Probability
True	True	0.8
True	False	0.2
False	True	0.3
False	False	0.7

Sensor Model:

Enough Sleep	Red Eyes	Sleep in Class
True	True	0.2
True	False	0.1
False	True	0.7
False	False	0.3

Hidden Markov Model

Since HMMs have a single observation variable, we need to create a combined observation probability table. The rest of the HMM tables (Initial State Probability and Transition Model) remain the same as for the DBN.

Observation Model:

Enough Sleep	Observation (Red Eyes, Sleep in Class)	Probability
True	(False, False)	0.72
True	(False, True)	0.08
True	(True, False)	0.18
True	(True, True)	0.02
False	(False, False)	0.21
False	(False, True)	0.09
False	(True, False)	0.49
False	(True, True)	0.21

Question 8

Generally, PF is best for complex, non-Gaussian problems but is computationally expensive. EKF is computationally efficient for mildly non-linear problems with Gaussian noise. UKF offers a compromise with better handling of non-linearities at a higher computational cost than EKF. The below are details:

Particle Filter (PF):

- *Advantages:*
  - Can handle non-linear and non-Gaussian models.
  - Suitable for multi-modal state distributions.
- *Disadvantages:*
  - High computational cost, scales poorly with dimensionality.
  - Can suffer from particle degeneracy and impoverishment.

Extended Kalman Filter (EKF):

- *Advantages:*
  - Less computationally intensive than PF.
  - Suitable for real-time applications with mild non-linearities.
- *Disadvantages:*
  - Prone to linearization errors in highly non-linear systems.
  - Assumes Gaussian noise, which may not always be accurate.

Unscented Kalman Filter (UKF):

- *Advantages:*
  - Better at handling non-linearities than EKF due to the unscented transform.
  - More accurate than EKF for non-linear systems while maintaining Gaussian assumptions.
- *Disadvantages:*
  - More computationally intensive than EKF due to sigma point calculations.
  - Although better than EKF, still not suitable for significant non-Gaussian noise or multi-modal distributions.

Implementation

a)

State Vector: The state of the robot is defined by its position, which includes both its x and y coordinates. Thus, the state vector is:

$$x = \begin{bmatrix} x_t \\ y_t \end{bmatrix}$$

Control Vector: The control inputs affecting the state are u_1 and u_2. So, the control vector is:

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Matrix A : This matrix represents the state transition without control inputs or noise. Since the state is the position, and the position at the next time step is the same as the current position without control inputs, A is an identity matrix:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Matrix B: This matrix represents how control inputs affect the state. From the motion model, we have:

$$x_{t+1} = x_t + 1.5u_1 + 0.1u_2$$

$$y_{t+1} = y_t + 0.2u_1 - 0.5u_2$$

Hence, the matrix B is:

$$B = \begin{bmatrix} 1.5 & 0.1 \\ 0.2 & -0.5 \end{bmatrix}$$

Matrix C: This matrix links the state to the measurements. From the sensor model, we have:

$$z_1 = 1.05x + 0.01y$$

$$z_2 = 0.01x + 0.90y$$

Therefore, the matrix C is:

$$C = \begin{bmatrix} 1.05 & 0.01 \\ 0.01 & 0.90 \end{bmatrix}$$

b)

Motion Covariance Matrix Q:

$$Q = \begin{bmatrix} 2.50696845 \times 10^{-3} & 1.79957758 \times 10^{-5} \\ 1.79957758 \times 10^{-5} & 2.51063277 \times 10^{-3} \end{bmatrix}$$

Measurement Covariance Matrix R:

$$R = \begin{bmatrix} 0.04869528 & -0.0058636 \\ -0.0058636 & 1.01216104 \end{bmatrix}$$

c)

With these matrices, along with the previously defined state vector x, control vector u, and matrices A, B, and C, a Kalman filter can be implemented. After calculation, the total error is **21.547**.

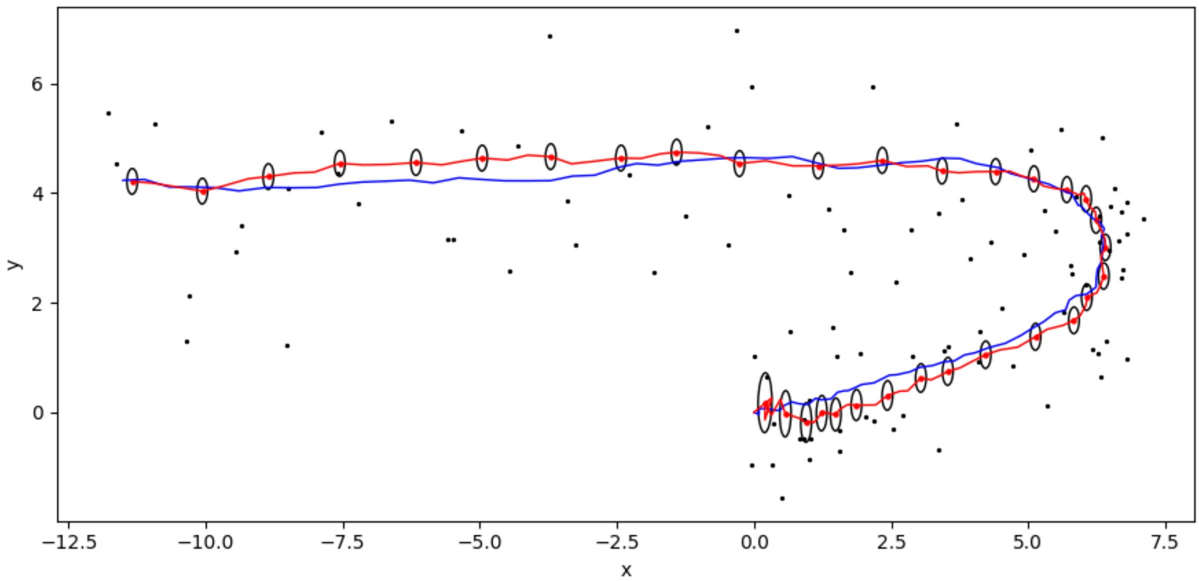

Figure 1: Kalman filtering of the trajectory