Quiz-3, CSE 643: AI-SecB   Full marks: 20   Duration 40 mins   20-11-2025

---

1. A random variable X, where $X \sim P$, has the following probability distribution:

$$P(X) = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{256}, \ldots, \frac{1}{256}\right)$$

   (a) Calculate entropy of X. 2

   (b) Now, consider another random variable Y with the following probability distribution

$$\left(\frac{3}{4}, \frac{1}{256}, \ldots, \frac{1}{256}\right)$$

   Without explicitly calculating entropy of Y, answer which one of X and Y has higher entropy. Give reason. 1.5

   (c) Entropy of a random variable can never be negative. Why ? 1

   (d) Suppose a robot has belief b. It applies action u and has a specific observation z. Write a mathematical expression for information gain after applying u. 1.5

(1)

Solution-1:

Please look into the next page.

Quiz-3 [Rubric]

1. (a) $H(X) = -\sum_x P(x) \cdot \log P(x)$

$= -\frac{1}{2} \cdot \log \frac{1}{2} - \frac{1}{4} \log \frac{1}{4} + \frac{1}{4} \cdot [\log 256]$

$= 1 + \frac{1}{4} \cdot 8 = 3$ (Answer).

(b) Second prob. dist has lower uncertainty / randomness compared to the first one.
So, X has higher entropy.

(c) $H(X) = -\sum_x P(x) \cdot \log P(x)$

$P(x) \geq 0$; so the above expression is always non-negative.

(d) Suppose, a robot has belief b; action is u and observation is z.
Information gain is the reduction in entropy for applying u and observing z:

$I_b(u) = H(x) - H(x' \mid z, u)$

- Deduct 1 mark if $E_z[H(x' \mid z, u)]$ is written.
This is because z is given here.

2. Answer the following questions related to robotic exploration:

   (a) Consider a robot with belief b. It would like to select an action that maximizes the difference of expected information gain and expected cost of applying u at state x. Equip the robot exploration function with a greedy strategy for achieving the above. 1.5

   (b) It is difficult to compute a greedy strategy for selecting an optimal action when the state is continuous. Write an algorithm that eases off this problem. 2.5

Solution-2a: Any of the following answers is correct.

- $\pi(b) = \underset{u}{\arg\max}\ \alpha\ (H_p(x) - \underset{z}{\mathbb{E}}\ [H_b(x' \mid z, u)]\ )\ +\ \int r(x,u)b(x)dx$
- $\pi(b) = \underset{u}{\arg\max}\ \int [r(x,u) - \alpha \int \int H_b(x' \mid z, u)\ p(z \mid x')\ p(x' \mid u, x)\ dz\ dx']\ b(x)\ dx$

– Any of the above answer is correct.
– 0.5 for expected information gain, 0.5 for expected cost, 0.5 for selecting the action with arg max

Solution-2b:

```
Algorithm Monte_Carlo_exploration(b):
    set ρ_u = 0 for all actions u
    for i = 1 to N do
        sample x ~ b(x)
        for all control action u do
            sample x' ~ p(x' | u, x)
            sample z ~ p(z | x')
            b' = Bayes_filter(b, z, u)
            ρ_u = ρ_u + r(x, u) − α H_b'(x')
        endfor
    endfor
    return argmax_u ρ_u
```

Table 17.1 A Monte Carlo implementation of the greedy exploration algorithm, which chooses actions by maximizing the trade-off between information gain and cost.

– Give 1 mark if a student writes QMDP.

3. Consider a set P of 1-dimensional points. For a given uery range $[x : x']$ we have to find the points in P. If we form a BST for P, dark grey nodes contain values that must be selected and light grey nodes are the ones that may be selected.

   Suppose function ColorDark(v) colors the enture subtree rooted at v and ColorLight(v) colors node v only. Also the splitting node v_{split} is given to you. Write an algorithm *to color the vertices dark and light grey* for selecting the points in the range $[x : x']$. 4

   Solution:

Algorithm 1DRangeQuery($\mathcal{T}, [x : x']$)
Input. A binary search tree $\mathcal{T}$ and a range $[x : x']$.
Output. All points stored in $\mathcal{T}$ that lie in the range.

1. $v_{\text{split}} \leftarrow$ FindSplitNode($\mathcal{T}, x, x'$)
2. **if** v_{split} is a leaf
3.   **then** ~~Check if the point stored at v_{split} must be reported.~~ 3, ColorLight (v_{split})
4.   **else** (∗ Follow the path to x and report the points in subtrees right of the path. ∗)
5. $v \leftarrow lc(v_{\text{split}})$
6.   **while** v is not a leaf — 6a. Color Light (v)
7.   **do if** $x \leqslant x_v$
8.   **then** ~~ReportSubtree($rc(v)$)~~ 8. Color Dark ($rc(v)$)
9. $v \leftarrow lc(v)$
10.   **else** $v \leftarrow rc(v)$
11.   ~~Check if the point stored at the leaf v must be reported.~~ 11. ColoLight(v)
12.   Similarly, follow the path to x', report the points in subtrees left of the path, and check if the point stored at the leaf where the path ends must be reported.

– Give 2 if the algorithm is written for reporting and not for coloring the nodes.
– Give 0.5 for each of the four calls to color functions.

4. We want to find a function $\hat{f}$ so that $\hat{f}$ estimates $f : \mathbb{R}^p \mapsto \mathbb{R}$.

   (a) Write a general form of $\hat{f}$ for linear regression. 1

   (b) When $p = 1$, suggest a way to estimate the unknown parameters of $\hat{f}$. 2

   (c) Write a general form of $\hat{f}$ for k-NN regression. 1

   (d) For k-NN regressor, why is it better to have higher value of k and n (number of data points)? 2

Solution-4

(a) The general form of $\hat{f}$ for linear regression is

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

(b) When $p = 1$, the linear model becomes

$$\hat{f}(x) = \beta_0 + \beta_1 x.$$

We estimate the unknown parameters (β_0, β_1) using ordinary least squares:

$$\min_{\beta_0, \beta_1} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2. \tag{2}$$

The closed-form solutions are

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

where

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i, \qquad \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i.$$

– For partially correct answer, give 1 mark if 2 is written.

(c) A general form of $\hat{f}$ for k-NN regression is

$$\hat{f}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i,$$

where $N_k(x)$ denotes the indices of the k nearest neighbours of x.

(d) For the k-NN regressor, a larger value of k and a larger number of data points n improve performance because:

- A larger k reduces variance by averaging over more neighbours, making the estimator less sensitive to noise and outliers.

- A larger n ensures that the nearest neighbours to x lie closer to the true point, reducing variance and improving the approximation to the true regression function.
- For k-NN to be *consistent*, both k and n should grow such that

$$k \to \infty, \qquad n \to \infty, \qquad \frac{k}{n} \to 0.$$

  This ensures that we average over enough neighbours to reduce variance (since $k \to \infty$), while keeping the neighbourhood size small relative to the dataset (since $k/n \to 0$).

$$\hat{f}(x) \to \mathbb{E}\left[Y \mid X = x\right]$$

– 1.5 marks for the mathematical expression and 0.5 marks for brief description of the expression.

– Give full marks if proper explanation is given (even without mathematical expresison).