Autonomous Mobile Robotics

Bayesian Filter

Geesara Kulathunga

September 29, 2022

Bayesian Filter

Contents

- Basic of Probability
- Probabilistic Generative Laws
- Estimation from Measurements
- Estimation from Measurements and Controls

Basic of Probability

Let X be a random variable and x be a specific value that is a outcome of sample space of X. For example, coin flip, where sample space consists of two values: head and tail

- p(X =x) probability that the random variable X has value x
- $\Sigma_x p(X = x) = 1$, where $p(X = x) \geq 0$ for discrete probability. We use p(x) instead of writing p(X=x)
- $\int p(x)dx = 1$ for continuous probability
- $p(x, y) = (X = x, Y = y)$, if random variable X and Y are independent, $p(x, y) = p(x)p(y)$
- $p(x|y) = \frac{p(x,y)}{p(y)}$ is called conditional probability
- $p(x) = \Sigma_y p(x, y) = \Sigma_y p(x|y)p(y)$ discrete case
- $p(x) = \int p(x|y)p(y)dy$ continuous case

Basic of Probability

- $p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\Sigma_{x^{\text{‘}}} p(y|x^{\text{‘}})p(x^{\text{‘}})}$ discrete
- $p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x^{\text{‘}})p(x^{\text{‘}})} dx^{\text{‘}}$ continuous
- $p(x)$ prior
- y can be called the data, e.g., sensor measurements
- $p(x|y)$ posterior (MAP)
- $$posterior = \frac{likelihood \times prior}{marginal\ likelihood}$$
- $p(y)$ does not depend on x. Thus, p(y) is same for any value of x, which yields $p(y)^{-1}$ as a normalizer, i.e., $p(x|y) = \eta p(y|x)p(x)$
- $p(x|y, z) = \frac{p(y|x,z)p(x|z)}{p(y|z)}$ as long as $p(y|z) > 0$

- $$p(x|y,z) = \frac{p(y|x,z)p(x|z)}{p(y|z)}, \quad p(y|z) = \Sigma_x p(y|z,x)p(x|z)dx$$

  as long as p(y|z) > 0
- p(x,y|z) = p(x|z,y)p(y|z,x) = p(x|z)p(y|z) Conditional independent does not imply absolute independence
- $E[X] = \Sigma_x xp(x)$ discrete
- $E[X] = \int xp(x)dx$ continuous
- $E[aX + b] = aE[X] + b$
- $Cov[X] = E[X - E[X]]^2 = E[X^2] - E[X]^2$

Example 01

Let's say, robot has to find a path between current pose and target pose. There are three path planning algorithms: A^*, RRT^*, and D^*, each of them are having 0.4, 0.4, and 0.2 possibility to find the optimal path between current and goal location. However, since the kinematics are incorporated when planning path, possibility of reaching to the target pose has following percentages: $40\%, 80\%,$ and 30%, respectively.

- Determine what would be probability that robot could not reach the target pose?
- What is the probability that robot could not reach to the target when robot decided to move on the path that A* is proposed?

Example 01

- Let random variable U be a selecting a path, where $A \in u_1, u_2, u_3$. Events: $u_1, u_2, \text{and}, u_3$, denoted A^* path, RRT^* path, and D^* path, respectively. Further, random variable V be a percentage of not reaching to the target pose. Thus, we can estimate p(V) as follows:

$$\begin{aligned} p(V) &= \Sigma_{u_i \in U}\, p(V|U = u_i)p(u_i) \\ &= 0.4 * 0.6 + 0.4 * 0.2 + 0.2 * 0.7 = 0.46 \end{aligned} \quad (1)$$

- $$\begin{aligned} p(u_1|V) &= \frac{p(B|u_1)p(u_1)}{\Sigma_{u_i \in U}\, p(V|U = u_i)p(u_i)} \\ &= (0.4 \cdot 0.6)/0.46 \end{aligned} \quad (2)$$

Example 02

https://arxiv.org/pdf/1711.02144.pdf

Example 02

You have trained a neural network for detecting space is free or occupied. Probability of correctly detecting space is free 0.3, and occupied 0.7, respectively. To conform detected information, second image processing-based technique is used. In that method, it can classify correctly 60% if the network detected object space as free. Similarly, it can classify correctly 80% if the network detected space is occupied.

- What is the probability that neural network detected space is free and if image processing technique also classified as free?

References

Gregor Klancar, Andrej Zdesar, Saso Blazic, and Igor Skrjanc.
***Wheeled mobile robotics: from fundamentals towards autonomous systems*.**
Butterworth-Heinemann, 2017.

Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza.

***Introduction to autonomous mobile robots*.**
MIT press, 2011.

Sebastian Thrun.
Probabilistic robotics.
Communications of the ACM, 45(3):52–57, 2002.