Probablistic Robotics

- Going "full Bayesian" : modelling change in behaviour of sensors, under environmental effects. "online auto-calibration".

1 Bayesian Network

1.1 Background

- In most practical scenarios , you are working with $\approx 100{,}000$ variables. eg: in a mapping task, each location of each keypoint is a variable. the probability distribution over these different variable will be in some 100,000 dim state space.
- even though the state vector is high dim, each component depends on very few factors (measurements are highly structured). eg : out of 100,000 camera pixel values (each being a variable), each pixel, depends only on where the world point location, and camera location.
- in general, the relation between the different states are **VERY** sparse
- since the relations / dependencies between variables are so sparse, there is a lot of **conditional independence that can be used**

1.2 Definition

- A Bayesian network is a **Directed Acyclic graph** : $D = (V, E)$ that models the factorization of a (Joint) probability distribution $P(X_1, X_2, X_3, ...)$ as a product of conditionals of the form :

$$p(X_1, X_2, X_3, ..., X_n) = \prod_{i=1}^{n} p(X_i|pa(X_i)) \qquad \text{pa(x) denotes all parents of x}$$

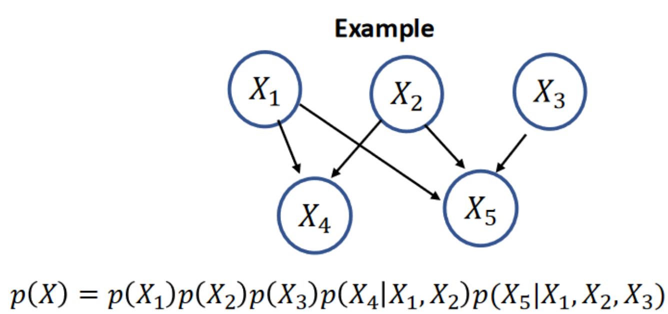

Figure 1: Bayes Net example

- Local Markov property : conditional independence between variables.

$$X_i \perp\!\!\!\perp X_{v-de(X_i)}|pa(X_i)$$

- : X_i is independent to all variables except its decendents ($X_{V-de(X_i)}$), when conditioned on they all are conditioned on X_i's parents
- **Generic joint distribution: N binary variables (from 43:00 (prob robotics lec (oct 1)))**:
  - objective : express a generic joint distribution over N binary variables $X_{1\to n} : X_i \in \{0, 1\}$
  - 2^n total states, a distribution and a distribution over each of these states.
  - Discrete distribution (discrete states), meaning a PMF, meaning , a **table**, with scalar probability values ($\in [0, 1]$) per state. The sum of all these values sum to 1.
  - lets say 100 variables, 2^{100} states, you have to keep track of 2^{100} probabilities.

    - **To sum up, if you look at generic joint distributions, you have 2^{100} probabilities to keep track of.**
    - **CONTRARILY** let's say , each variable, depends on only 2 other variables (like how a pixel value depends only on world coordinate, and camera location), each factor can be stored with 2^3 values. then you have to store only n * 8 probabilities, (800).
    - **Instead of maintaining 1 big table (containing probabilities for each possible state), and looking up a joint probability, you can query values from smaller tables (1 corresponding to each factor), and take their product, which will be equivalent to the value you would have gotten if you maintained a big table, due to inherent conditional independence between variables** CONFIRM.
    - if each distribution is a gaussian, how would you take a product of gaussians?
  - **Bayesian Networks as generative models:** Topologically order DAG, and then sample values in that order from the distribution corresponding to it's corresponding factor. (**Ancestoral Sampling**)

2 Recurscive Bayesian Estimation

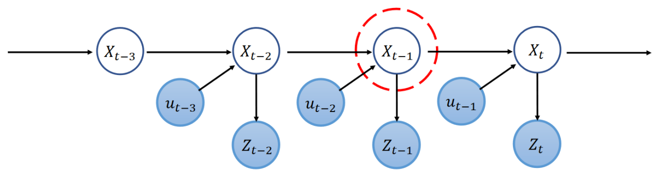

Figure 2: Recursive Bayesian Estimation

- When you have a system, you also get a corresponding graph, The recursive nature of the problem can be naturally modelled by a Hidden markov model, which results in a property : "the future if conditionally independent of the past given the present." How to write this conditional independence in the form of $p(x, y|z) = p(x|z)\ p(y|z)$
- slides are descriptive enough
- the question to answer : what is $p(X_{t+1}|u_{0:t}, Z_{1:t})$
- Consider the joint distribution over X_{t+1}, and X_t:

$$p(X_{t+1}, X_t|u_{0:t}, Z_{1:t}) = p(X_{t+1}|X_t, u_{0:t}, Z_{1:t})P(X_t|u_{0:t}, Z_{1:t}) \qquad \text{chain rule of probability}$$

$$\Rightarrow p(X_{t+1}|X_t, u_t)p(X_t|u_{0:t}, Z_{1:t})$$

- Now that we have an equation for the joint distribution between X_t, and X_{t+1}, conditioned on previous measurements and actions., we can **Marginalize X_t out!**

$$p(X_{t+1}|u_{0:t}, Z_{1:t}) = \int p(X_{t+1}|X_t + u_t)p(X_t|u_{1:t}, Z_{1:t})dX_t$$

- shouldnt $\int p(X_t|u_{0:t}, Z_{1:t})dX_t = 1$?

  NOTE: Z_t is the measurement at time T, (from previous state)
- Now, How to incorporate info of measurement from X_{t+1} to $p(X_{t+1}|u_{1:t}, Z_{1:t})$? **BAYES RULE!**

$$P(x_{t+1}|u_{0:t}, Z_{1:t+1}) = \frac{P(Z_{t+1}|X_{t+1}, u_{0:t}, Z_{1:t})P(X_t|u_{0:t}, Z_{1+t})}{P(Z_{t+1}|u_{0:t}, Z_{1:t}))} \qquad \text{Forward model of sensor can be simplified}$$

$$p(Z_{t+1}|u_{0:t}, Z_{0:t}) = \int p(Z_{t+1}|X_t)p(X_t|u_{0:t}, Z_{1:t}) \qquad \text{calculation of evidence}$$

- from the above calculation, you see how the evidence term is simply used to keep $\int p(X_{t+1}|u_{0:t}, Z_{1:t}) = 1$

3 Bayes Filter

Both steps together form the BAYES FILTER: for each time step :

1. initial estimate of distribution of current state on application of action u_t (before measurement)

$$p(X_{t+1}|u_{1:t}, Z_{1:t}) = \int p(X_{t+1}|X_t, u_t)p(x_t|u_{1:t-1}, Z_{1:t})$$

   in this context measurement is taken, then action is done. (measurement completes current estimate, action begins next estimate)

2. estimate of time step computed after applying action is enriched by information from sensor using bayes rule

$$p(X_{t+1}|u_{1:t}, Z_{1:t+1}) = \frac{p(Z|X_{t+1})p(X_t|u_{1:t}Z_{1:t})}{p(Z_{t+1}|u_{1:t}, Z_{1:t})}$$

3.1 Types of Bayes Filters:

1. Linear Gaussian case : (Kalman Filter)
2. Nonlinear Gaussian case : (Extended Kalman Filter)
3. Nonlinear non-Gaussian case : (Particle Filter)