AI & Robotics: Research

Filtering

Marc Toussaint
Technical University of Berlin
Summer 2020

Outline

- Tree Search Basics
- Stochastic Domains: MDPs
- Partial Observability: POMDPs
- Reparameterization Trick & DESPOT

Recall Bayes' theorem

$$P(X|Y) = \frac{P(Y|X)\ P(X)}{P(Y)}$$

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{(normalization)}}$$

Note on notation

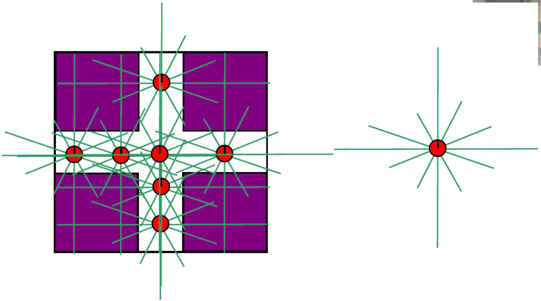
- In this lecture the $P(\cdots)$ generically means *probability*, namely the probability for some random variables in some previously defined model. Others would also write Prob$(\cdots)$ or $\mathbb{P}(\cdots)$.
- That's in constrast to $p_t(x_t)$, or $\hat p_t(x_t)$, which are defined to refer to a specific quantity. E.g., in this lecture, we will define $p_t(x_t) := P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1})$; which means that $p_t(\cdots)$ does not generically mean "probability", but a very specific the probability, namely that probability of a state at interation t conditional to all previous information.

- How can we apply this to the State Estimation Problem?

- How can we apply this to the State Estimation Problem?

Using Bayes Rule:

$P(\text{location} \mid \text{sensor}) = \frac{P(\text{sensor} \mid \text{location})P(\text{location})}{P(\text{sensor})}$



Bayes Filter

x_t = state (location) at time t
y_t = sensor readings at time t
$u_{t\text{-}1}$ = control command (action, steering, velocity) at time t-1

- Given the history $y_{0:t}$ and $u_{0:t\text{-}1}$, we want to compute the **belief** over the state at time t

$$p_t(x_t) := P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1})$$

Bayes Filter

$$p_t(x_t) := P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1})$$

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t\; P(y_t \,|\, x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1})\; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1})
\end{aligned}
$$

using Bayes rule $P(X|Y,Z) = c\; P(Y|X,Z)\; P(X|Z)$ with some normalization constant c_t

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1})
\end{aligned}
$$

uses conditional independence of the observation on past observations and controls

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t) \; \int_{x_{t\text{-}1}} P(x_t, x_{t\text{-}1} \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1}
\end{aligned}
$$

by definition of the marginal

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t) \; \int_{x_{t\text{-}1}} P(x_t, x_{t\text{-}1} \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \,|\, x_t) \; \int_{x_{t\text{-}1}} P(x_t \,|\, x_{t\text{-}1}, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_{t\text{-}1} \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1}
\end{aligned}
$$

by definition of a conditional

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t) \; \int_{x_{t\text{-}1}} P(x_t, x_{t\text{-}1} \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \,|\, x_t) \; \int_{x_{t\text{-}1}} P(x_t \,|\, x_{t\text{-}1}, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_{t\text{-}1} \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \,|\, x_t) \; \int_{x_{t\text{-}1}} P(x_t \,|\, x_{t\text{-}1}, u_{t\text{-}1}) \; P(x_{t\text{-}1} \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1}
\end{aligned}
$$

given $x_{t\text{-}1}$, x_t depends only on the controls $u_{t\text{-}1}$ (Markov Property)

Bayes Filter

$$
\begin{aligned}
p_t(x_t) &:= P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t) \; P(x_t \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \\
&= c_t \; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t, x_{t\text{-}1} \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, x_{t\text{-}1}, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; P(x_{t\text{-}1} \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, x_{t\text{-}1}, u_{t\text{-}1}) \; P(x_{t\text{-}1} \,|\, y_{0:t\text{-}1}, u_{0:t\text{-}1}) \; dx_{t\text{-}1} \\
&= c_t \; P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) \; p_{t\text{-}1}(x_{t\text{-}1}) \; dx_{t\text{-}1}
\end{aligned}
$$

- A Bayes filter updates the posterior belief $p_t(x_t)$ in each time step using the:
  observation model $P(y_t \,|\, x_t)$
  transition model $P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})$

Bayes Filter

$$\underbrace{p_t(x_t)}_{\text{new belief}} \propto \underbrace{P(y_t \,|\, x_t)}_{\text{likelihood}} \underbrace{\int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) \underbrace{p_{t\text{-}1}(x_{t\text{-}1})}_{\text{old belief}} \, dx_{t\text{-}1}}_{\text{predictive distribution}\hat{p}_t(x_t)}$$

1. We have a belief $p_{t\text{-}1}(x_{t\text{-}1})$ of our previous position

2. We use the motion model to predict the current position
$$\hat{p}_t(x_t) \propto \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) \, p_{t\text{-}1}(x_{t\text{-}1}) \, dx_{t\text{-}1}$$

3. We integetrate this with the current observation to get a better belief
$$p_t(x_t) \propto P(y_t \,|\, x_t) \, \hat{p}_t(x_t)$$

Kalman filter := Bayesian Filtering with Gaussians

- Assume that
  - belief is Gaussian,

$$p_t(x_t) = \mathcal{N}(x_t \,|\, s_t, S_t)$$

  - dynamics are linear-Gaussian,

$$P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, A(u_{t\text{-}1})\; x_{t\text{-}1} + a(u_{t\text{-}1}), Q)$$

  - observations are linear-Gaussian,

$$P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, C x_t + c, W)$$

- Then we can compute the Bayes filter analytically

Kalman filter derivation

$$
\begin{aligned}
p_t(x_t) &\propto P(y_t \,|\, x_t) \int_{x_{t\text{-}1}} P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})\; p_{t\text{-}1}(x_{t\text{-}1})\; dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, Cx_t + c, W) \int_{x_{t\text{-}1}} \mathcal{N}(x_t \,|\, Ax_{t\text{-}1} + a, Q)\; \mathcal{N}(x_{t\text{-}1} \,|\, s_{t\text{-}1}, S_{t\text{-}1})\; dx_{t\text{-}1} \\
&= \mathcal{N}(y_t \,|\, Cx_t + c, W)\; \mathcal{N}(x_t \,|\, \underbrace{As_{t\text{-}1} + a}_{=:\hat s_t},\; \underbrace{Q + AS_{t\text{-}1}A^\top}_{=:\hat S_t}) \\
&= \mathcal{N}(Cx_t + c \,|\, y_t, W)\; \mathcal{N}(x_t \,|\, \hat s_t, \hat S_t) \\
&= \mathcal{N}[x_t \,|\, C^\top W^{\text{-}1}(y_t - c),\; C^\top W^{\text{-}1} C]\; \mathcal{N}(x_t \,|\, \hat s_t, \hat S_t) \\
&= \mathcal{N}(x_t \,|\, s_t, S_t) \cdot \langle \text{terms indep. of } x_t \rangle
\end{aligned}
$$

$$
S_t = (C^\top W^{\text{-}1} C + \hat S_t^{\text{-}1})^{\text{-}1} = \hat S_t - \underbrace{\hat S_t C^\top (W + C \hat S_t C^\top)^{\text{-}1}}_{\text{“Kalman gain” } K} C \hat S_t
$$

$$
s_t = S_t[C^\top W^{\text{-}1}(y_t - c) + \hat S_t^{\text{-}1} \hat s_t] = \hat s_t + K(y_t - C\hat s_t - c)
$$

The second to last line uses the general Woodbury identity.
The last line uses $S_t C^\top W^{\text{-}1} = K$ and $S_t \hat S_t^{\text{-}1} = \mathbf{I} - KC$
(more identities: see “Gaussian identities”
`http://ipvs.informatik.uni-stuttgart.de/mlr/marc/notes/gaussians.pdf`)

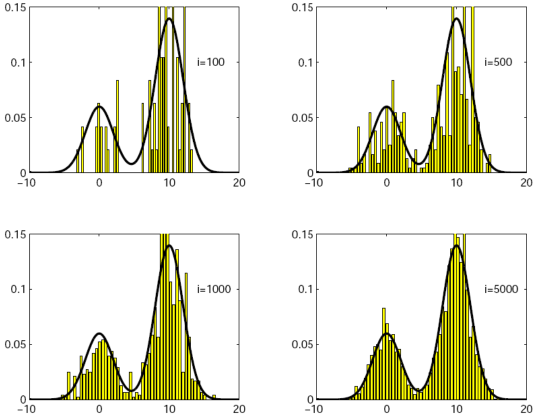
Extended Kalman filter (EKF) and Unscented Transform

- If $P(y_t \,|\, x_t)$ or $P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1})$ are not linear:
  $P(y_t \,|\, x_t) = \mathcal{N}(y_t \,|\, g(x_t), W)$
  $P(x_t \,|\, u_{t\text{-}1}, x_{t\text{-}1}) = \mathcal{N}(x_t \,|\, f(x_{t\text{-}1}, u_{t\text{-}1}), Q)$
  – approximate f and g as locally linear (*Extended Kalman Filter*)
  – or sample locally from them and reapproximate as Gaussian (*Unscented Transform*)

Recall: Particle Representation of a Distribution

- Weighed set of N particles $\{(x^i, w^i)\}_{i=1}^N$

$$p(x) := \sum_{i=1}^N w^i \delta(x, x^i)$$



Particle Filter := Bayesian Filtering with Particles

- Assume the old belief is a particle distribution,

$$p_{t\text{-}1}(x) := \sum_{i=1}^{N} w_{t\text{-}1}^{i}\ \delta(x, x_{t\text{-}1}^{i})$$

Particle Filter := Bayesian Filtering with Particles

- Assume the old belief is a particle distribution,

$$p_{t\text{-}1}(x) := \sum_{i=1}^N w_{t\text{-}1}^i \; \delta(x, x_{t\text{-}1}^i)$$

- The predictive distribution is

$$\hat p_t(x_t) = \int_x P(x_t \mid u_{t\text{-}1}, x) \; p_{t\text{-}1}(x) \; dx$$

Particle Filter := Bayesian Filtering with Particles

- Assume the old belief is a particle distribution,

$$p_{t\text{-}1}(x) := \sum_{i=1}^N w_{t\text{-}1}^i \; \delta(x, x_{t\text{-}1}^i)$$

- The predictive distribution is

$$\begin{aligned}\hat p_t(x_t) &= \int_x P(x_t \mid u_{t\text{-}1}, x) \; p_{t\text{-}1}(x) \; dx \\ &= \int_x P(x_t \mid u_{t\text{-}1}, x) \; \sum_{i=1}^N w_{t\text{-}1}^i \; \delta(x, x_{t\text{-}1}^i) \; dx\end{aligned}$$

Particle Filter := Bayesian Filtering with Particles

- Assume the old belief is a particle distribution,

$$p_{t\text{-}1}(x) := \sum_{i=1}^N w_{t\text{-}1}^i \; \delta(x, x_{t\text{-}1}^i)$$

- The predictive distribution is

$$\begin{aligned}\hat p_t(x_t) &= \int_x P(x_t \mid u_{t\text{-}1}, x) \; p_{t\text{-}1}(x) \; dx \\ &= \int_x P(x_t \mid u_{t\text{-}1}, x) \; \sum_{i=1}^N w_{t\text{-}1}^i \; \delta(x, x_{t\text{-}1}^i) \; dx \\ &= \sum_{i=1}^N w_{t\text{-}1}^i \; P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}^i)\end{aligned}$$

where $P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}^i)$ is the prediction for the single particle $x_{t\text{-}1}^i$.

Particle Filter := Bayesian Filtering with Particles

- Assume the old belief is a particle distribution,

$$p_{t\text{-}1}(x) := \sum_{i=1}^N w_{t\text{-}1}^i \ \delta(x, x_{t\text{-}1}^i)$$

- The predictive distribution is

$$\begin{aligned}\hat p_t(x_t) &= \int_x P(x_t \mid u_{t\text{-}1}, x) \ p_{t\text{-}1}(x) \ dx \\ &= \int_x P(x_t \mid u_{t\text{-}1}, x) \ \sum_{i=1}^N w_{t\text{-}1}^i \ \delta(x, x_{t\text{-}1}^i) \ dx \\ &= \sum_{i=1}^N w_{t\text{-}1}^i \ P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}^i)\end{aligned}$$

where $P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}^i)$ is the prediction for the single particle $x_{t\text{-}1}^i$.We approximate each $P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}^i)$ by a single-particle-distribution $\delta(x, x_t^i)$ by sampling one new particle:

$$\hat p_t(x_t) \approx \sum_{i=1}^N w_{t\text{-}1}^i \ \delta(x, x_t^i) \ , \quad \text{where } x_t^i \sim P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1}^i)$$

Particle Filter := Bayesian Filtering with Particles

- Assume the old belief is a particle distribution,

$$p_{t\text{-}1}(x) := \sum_{i=1}^N w^i_{t\text{-}1}\ \delta(x, x^i_{t\text{-}1})$$

- The predictive distribution is

$$\begin{aligned}\hat p_t(x_t) &= \int_x P(x_t \mid u_{t\text{-}1}, x)\ p_{t\text{-}1}(x)\ dx \\ &= \int_x P(x_t \mid u_{t\text{-}1}, x)\ \sum_{i=1}^N w^i_{t\text{-}1}\ \delta(x, x^i_{t\text{-}1})\ dx \\ &= \sum_{i=1}^N w^i_{t\text{-}1}\ P(x_t \mid u_{t\text{-}1}, x^i_{t\text{-}1})\end{aligned}$$

where $P(x_t \mid u_{t\text{-}1}, x^i_{t\text{-}1})$ is the prediction for the single particle $x^i_{t\text{-}1}$.We approximate each $P(x_t \mid u_{t\text{-}1}, x^i_{t\text{-}1})$ by a single-particle-distribution $\delta(x, x^i_t)$ by sampling one new particle:

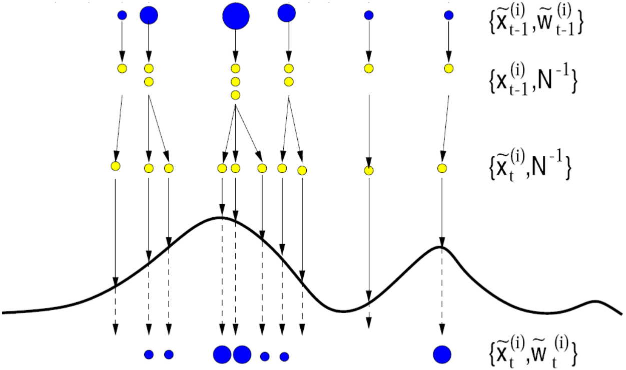
$$\hat p_t(x_t) \approx \sum_{i=1}^N w^i_{t\text{-}1}\ \delta(x, x^i_t)\ , \quad \text{where } x^i_t \sim P(x_t \mid u_{t\text{-}1}, x^i_{t\text{-}1})$$

- The new belief is

$$\begin{aligned}p_t(x_t) &\propto P(y_t \mid x_t)\ \hat p_t(x_t) \\ &\approx P(y_t \mid x_t)\ \sum_{i=1}^N w^i_{t\text{-}1}\ \delta(x, x^i_t) \\ &= \sum_{i=1}^N P(y_t \mid x^i_t)\ w^i_{t\text{-}1}\ \delta(x, x^i_t)\end{aligned}$$

which is a new particle distribution! With weights $w^i_t = P(y_t \mid x^i_t)\ w^i_{t\text{-}1}$

Particle Filter



1. Start with N particles $\{(x^i_{t\text{-}1}, w^i_{t\text{-}1})\}_{i=1}^N$
2. **Resample** particles to get N uniform-weight-particles: $\{\hat{x}^i_{t\text{-}1}, 1/N\}_{i=1}^N$
3. Use motion model to get new “predictive” particles $\{x^i_t\}_{i=1}^N$
   each $x^i_t \sim P(x_t \,|\, u_{t\text{-}1}, \hat{x}^i_{t\text{-}1})$
4. Use observation model to assign new weights $w^i_t \propto P(y_t \,|\, x^i_t)/N$

- “Particle Filter”

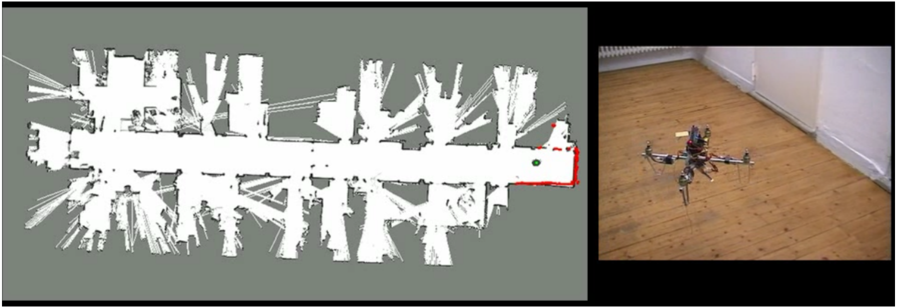
  aka *Monte Carlo Localization* in the mobile robotics community

  Condensation Algorithm in the vision community

- Efficient resampling is important:
  Typically “Residual Resampling”:

  Instead of sampling directly $\hat{n}^i \sim \text{Multi}(\{Nw_i\})$ set $\hat{n}^i = \lfloor Nw_i \rfloor + \bar{n}_i$ with $\bar{n}_i \sim \text{Multi}(\{Nw_i - \lfloor Nw_i \rfloor\})$
  Liu & Chen (1998): Sequential Monte Carlo Methods for Dynamic Systems.
  Douc, Cappé & Moulines: Comparison of Resampling Schemes for Particle Filtering.

Example: Quadcopter Localization



http://www.slawomir.de/publications/grzonka09icra/grzonka09icra.pdf

Quadcopter Indoor Localization

Typical Pitfall in Particle Filtering

- Predicted particles $\{x_t^i\}_{i=1}^N$ have very low observation likelihood $P(y_t \mid x_t^i) \approx 0$
  ("particles die over time")

- Classical solution: generate particles also with other than purely forward proposal $P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1})$:
  - Choose a proposal that depends on the new observation y_t (with importance weights), ideally approximating $P(x_t \mid y_t, u_{t\text{-}1}, x_{t\text{-}1})$
  - Or mix particles sampled directly from $P(y_t \mid x_t)$ and from $P(x_t \mid u_{t\text{-}1}, x_{t\text{-}1})$.
    (*Robust Monte Carlo localization for mobile robots.* Sebastian Thrun, Dieter Fox, Wolfram Burgard, Frank Dellaert)

Factorizing the Belief Representation

- Let's assume that the state $x_t = (r_t, q_t)$ has two variables
- We have a factored belief representation $p_t(r_t, q_t) = p_t(r_t \mid q_t) p_t(q_t)$
  Let's assume that the state $x_t = (r_t, q_t)$ has two variables
- Let's also assume dynamics factor
  $P(r_t, q_t \mid u_{t\text{-}1}, r_{t\text{-}1}, q_{t\text{-}1}) = P(q_t \mid q_{t\text{-}1})\ P(r_t \mid r_{t\text{-}1}, q_{t\text{-}1})$
- The predictive distribution is:

$$\hat{p}_t(q_t) = \int_{q_{t\text{-}1}} P(q_t \mid q_{t\text{-}1})\ p_{t\text{-}1}(q_{t\text{-}1})\ dq_{t\text{-}1}$$
$$\hat{p}_t(r_t \mid q_t) = \int_{r_{t\text{-}1}} P(r_t \mid r_{t\text{-}1}, q_{t\text{-}1})\ p_t(r_{t\text{-}1} \mid q_{t\text{-}1})\ dr_{t\text{-}1}$$

- The Likelihood updates are

$$\begin{aligned} p_t(r_t, q_t) &= P(y_t \mid r_t, q_t)\ \hat{p}_t(r_t, q_r) \\ &= \frac{P(r_t \mid y_t, q_t)\ P(y_t \mid q_t)}{P(r_t \mid q_t)}\ \hat{p}_t(r_t \mid q_t)\ \hat{p}_t(q_t) \end{aligned}$$

Factorizing the Belief Representation

- This shows how we can in principle factorize the belief
- We have made independence assumptions (for the dynamics), but no representation assumptions (Gauss/particle) yet!

Factorizing the Belief Representation

- This shows how we can in principle factorize the belief
- We have made independence assumptions (for the dynamics), but no representation assumptions (Gauss/particle) yet!

- Assume that we choose to represent
  - $p_t(q)$ as a particle distribution
  - $p_t(r \mid q)$ as a multinomial distribution (over K discrete states r)

$$p_t(r,q) = p_t(r \mid q) \Big[\sum_{i=1}^N w_t^i \, \delta(q, q_t^i) \Big]$$

$$= \sum_{i=1}^N p_t^i(r) \; w_t^i \; \delta(q, q_t^i) \; , \quad p_t^i(r) := p_t(r \mid q_t^i)$$

So we need to store N particles $\{q_t^i\}$, N weights w_t^i, and N multinomial tables $p_t^i(r)$.

Factorizing the Belief Representation

- This shows how we can in principle factorize the belief
- We have made independence assumptions (for the dynamics), but no representation assumptions (Gauss/particle) yet!

- Assume that we choose to represent
  - $p_t(q)$ as a particle distribution
  - $p_t(r \mid q)$ as a multinomial distribution (over K discrete states r)

$$
\begin{aligned}
p_t(r, q) &= p_t(r \mid q) \left[\sum_{i=1}^{N} w_t^i \, \delta(q, q_t^i) \right] \\
&= \sum_{i=1}^{N} p_t^i(r) \, w_t^i \, \delta(q, q_t^i) \,, \quad p_t^i(r) := p_t(r \mid q_t^i)
\end{aligned}
$$

So we need to store N particles $\{q_t^i\}$, N weights w_t^i, and N multinomial tables $p_t^i(r)$.

- Belief updates on $p_t(q)$ are done as in particle filtering; the weight and $p_t^i(r)$ updates are given by above equations.

PoseRBPF paper

- Relating back to the paper, why can they drop the term $\frac{1}{P(r_t \mid q_t)}$ from the likelihood update?
- The dynamics model Eq (5) is actually 2nd-order Markov – Is this "allowed"?
- Eq (5) uses Euler angles! And has a Gaussian distribution over Euler angles!!
- Why is there no "injection" in the default framework (above (7))?