Rao-Blackwellized Particle Filter

Recall that the main difficulty with particle filtering is that with a high dimensional state variable x_t, an impossibly large number of particles is needed to accurately represent $P(x_t|z_{0:t})$. In some filtering problems, it is possible to exploit conditional independence of components of the state variables $x_{1:t}$ in order to reduce the number of particles needed. In this lecture we will see examples of this technique, known as Rao-Blackwellization, applied to SLAM and visual tracking.

Simultaneous Localization and Mapping (SLAM)

We will start by considering a simple example from Murphy [2]. Let our state vector $x_t = (l_t, m_t)$ be composed of two components, a robot location $l_t \in \{1, \ldots, N_L\}$, and a map m_t, which is a vector of random variables, $m_t(i) \in \{1, \ldots, N_O\}, i \in \{1, \ldots, N_L\}$. The robot can be in one of N_L locations, and the map consists of N_L locations, each with one of N_O possible labels (think free space, wall, doorway, observation of landmarks, etc.)

Let the observation at time t be $z_t \in \{1, \ldots, N_O\}$, i.e. the robot (possibly incorrectly) observes the label of its current location. We can define $P(z_t|x_t)$ by

$$P(z_t = k|x_t) = P(z_t = k|l_t = i, m_t = (m_1, \ldots, m_{N_L})) = \begin{cases} p_o & \text{if } m_i = k \\ 1 - p_o & \text{otherwise} \end{cases} \quad (1)$$

The robot makes an observation error with probability $1 - p_o$.

Remember that we also need to define a transition model $P(x_t|x_{t-1})$ for the state. A reasonable assumption is to define a transition model for the map m_t either by making it static ($m_t = m_{t-1}$), or each location i could change independently over time. The location l_t could evolve based upon both the previous location l_{t-1} and map m_{t-1} (e.g. bumping into walls), as well as another observed variable a_t, the action at time t. The full observation at time t is then $y_t = (a_t, z_t)$.

Inference

Since this world is discrete, we could in fact do exact inference to determine $P(x_t|y_{0:t})$. However, the size of the state space in which x_t lies is $N_L N_O^{N_L}$. Exact inference will be slow, and intractable in all but the smallest problems. Naive particle filtering will also run into trouble, needing many particles to adequately sample this large, high dimensional state space.

The reason for this explosion in complexity is that the seemingly independent map variables $m_t(i)$ become dependent once we have an observation z_t. Without knowing the current location l_t, we do not know which map variable is responsible for the observation z_t.

However, this is also the insight that we will use to make approximate inference efficient. Conditioned on a particular value for the current location l_t, the map variables do become independent. We can then marginalize them out analytically (store only a list of $P(m_t(i))$ rather than one high-dimensional $P(m_t)$) and not have to resort to sampling methods.

The resulting algorithm is known as Rao-Blackwellized particle filtering. Let us denote by $b_t^{(j)}$ the j^{th} particle at time t. Each particle $b_t^{(j)} = (l_t^{(j)}, m_t^{(j)}(i))$ contains both a location $l_t^{(j)}$ and a factored representation of the map distribution $m_t^{(j)}(i)$ (contrast this with full $N_O^{N_L}$ dimensional map). The steps of the algorithm for our particular problem are as follows:

1. Sample $l_{t+1}^{(j)}$ from a proposal distribution. A simple proposal distribution would be to sample the next location $l_{t+1}^{(j)}$ conditioned on the previous location and map ($b_t^{(j)}$ from previous particle set) and action a_{t+1} at this time step.

2. Update the distribution on each component of the map $m_{t+1}^{(j)}(i)$ separately, using the sampled location $l_{t+1}^{(j)}$ and the observation z_{t+1}.

$$m_{t+1}^{(j)}(i) = \begin{cases} P(z_{t+1}|m_{t+1}^{(j)}(i))P(m_{t+1}^{(j)}(i)|m_t^{(j)}(i)) & \text{if } i = l_{t+1}^{(j)} \\ P(m_{t+1}^{(j)}(i)|m_t^{(j)}(i)) & \text{otherwise} \end{cases} \tag{2}$$

3. Update the weights: $w_{t+1}^{(j)} = u_{t+1}^{(j)} w_t^{(j)}$. The factor $u_{t+1}^{(j)}$ is determined by the proposal distribution used, and is proportional to $P(z_{t+1}|l_{t+1}^{(j)}, b_t^{(j)})$ when using the (non-optimal) proposal distribution described above.

4. Resample the new set of particles by uniformly sampling according to these weights.

Visual Tracking - Continuous Variables

We will now consider a problem involving continuous random variables, and see how the Kalman filter we first discussed becomes applicable. We will consider the problem of visual tracking, the method described in [1].

In the case of visual tracking, the state vector $x_t = (l_t, a_t)$, the location l_t of the target being tracked, and its appearance a_t. As usual, we will be interested in a variation of the following filtering equation:

$$P(l_t, a_t|z_{0:t}) = \alpha P(z_t|l_t, a_t) \int_{l_{t-1}} \int_{a_{t-1}} P(l_t, a_t|l_{t-1}, a_{t-1}) P(l_{t-1}, a_{t-1}|z_{0:t-1}) \tag{3}$$

In particular, we desire the marginal posterior over location:

$$P(l_t|z_{0:t}) = \alpha \int_{a_t} P(z_t|l_t, a_t) \int_{l_{t-1}} \int_{a_{t-1}} P(l_t, a_t|l_{t-1}, a_{t-1}) P(l_{t-1}, a_{t-1}|z_{0:t-1}) \tag{4}$$

We will approximate the posterior $P(l_{t-1}, a_{t-1}|z_{0:t-1})$ by a set of particles, $\{l_{t-1}^{(i)}, w_{t-1}^{(i)}, \alpha_{t-1}^{(i)}(a_{t-1})\}_{i=1}^N$. The $\alpha_{t-1}^{(i)}(a_{t-1})$ are conditional *distributions* over appearance (rather than samples of appearance states).

$$\begin{aligned} P(l_{t-1}, a_{t-1}|z_{0:t-1}) &= P(l_{t-1}|z_{0:t-1}) P(a_{t-1}|l_{t-1}, z_{0:t-1}) && (5) \\ &\approx \sum_i w_{t-1}^{(i)} \delta(l_{t-1}^{(i)}) \alpha_{t-1}^{(i)}(a_{t-1}) && (6) \end{aligned}$$

$$\alpha_{t-1}^{(i)}(a_{t-1}) \equiv P(a_{t-1}|l_{t-1}^{(i)}, z_{0:t-1}) \tag{7}$$

Substituting this approximation into Equation 4, we obtain:

$$P(l_t|z_{0:t}) \approx \alpha \sum_i w_{t-1}^{(i)} \int_{a_t} P(z_t|l_t, a_t) \int_{a_{t-1}} P(a_t|l_t, l_{t-1}^{(i)}, a_{t-1}) P(l_t|l_{t-1}^{(i)}, a_{t-1}) \alpha_{t-1}^{(i)}(a_{t-1}) \quad (8)$$

In order to proceed, we make one further simplification, that location l_t is conditionally independent of a_{t-1} given l_{t-1}:

$$P(l_t|l_{t-1}^{(i)}, a_{t-1}) = P(l_t|l_{t-1}^{(i)}) \quad (9)$$

We now obtain our final filtering equation by plugging this into Equation 8:

$$P(l_t|z_{0:t}) \approx \alpha \sum_i w_{t-1}^{(i)} P(l_t|l_{t-1}^{(i)}) \int_{a_t} P(z_t|l_t, a_t) \int_{a_{t-1}} P(a_t|l_t, l_{t-1}^{(i)}, a_{t-1}) \alpha_{t-1}^{(i)}(a_{t-1}) \quad (10)$$

We can do importance sampling from this distribution in the usual fashion, using $\sum_i w_{t-1}^{(i)} P(l_t|l_{t-1}^{(i)})$ as the proposal density.

Similar to the previous section, the final algorithm is as below. For each time step, repeat for $j = 1, 2, \ldots, N$:

1. Randomly select a particle $l_{t-1}^{(i)}$ from the previous time set, by sampling uniformly according to weights $w_{t-1}^{(i)}$.
2. Sample a new particle $l_t^{(j)} \sim P(l_t|l_{t-1}^{(i)})$
3. Calculate the posterior density $\alpha_t^{(j)}(a_t)$ on the appearance a_t:

$$\alpha_t^{(j)}(a_t) = k_t^{(j)} P(z_t|l_t^{(j)}, a_t) \int_{a_{t-1}} P(a_t|l_t^{(j)}, l_{t-1}^{(i)}, a_{t-1}) \alpha_{t-1}^{(i)}(a_{t-1}) \quad (11)$$

4. Set $w_t^{(j)} = 1/k_t^{(j)}$.

If we restrict the distributions $P(z_t|l_t^{(j)}, a_t)$ and $P(a_t|l_t^{(j)}, l_{t-1}^{(i)}, a_{t-1})$ to be linear Gaussians, then each $\alpha_t^{(j)}(a_t)$ will be a multivariate Gaussian, and its update can be performed using a Kalman filter. This also makes the computation of $k_t^{(j)}$, and hence $w_t^{(j)}$ straight-forward.

Acknowledgements

These notes are derived from [2] and [1].

References

[1] Z. Khan, T. Balch, and F. Dellaert. A rao-blackwellized particle filter for eigentracking. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 2004.

[2] K. Murphy. Bayesian map learning in dynamic environments. In *NIPS '99 (Neural Info. Proc. Systems)*, 1999.