1 Monte Carlo Inference

Motivation

In real robots we often deal with continuous variables and continuous state spaces. Hence, we need to extend the notion of probability to continuous variables.

With nonlinear dynamics and/or measurement models exact inference is computationally intractable. One way out is to perform approximate inference using sampling. One of the best known such algorithms are particle filters, including Monte Carlo Localization

1.1 Continuous Probability Densities

In robotics we typically need to model a belief over continuous, multivariate random variables $x \in \mathbb{R}^n$. We do this using **probability density functions** (PDFs) $p(x)$ over the variables x, satisfying

$$\int p(x)dx = 1. \tag{1}$$

In terms of notation, for continuous variables we use lowercase letters for random variables, and uppercase letters to denote sets of them. We drop the notational conventions of making distinctions between random variables X and their realized values x, which helped us get used to thinking about probabilities, but will get in the way of clarity below.

1.2 Gaussian Densities

A Gaussian probability density is given by

$$\mathcal{N}(\theta; \mu, \Sigma) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left\{-\frac{1}{2}\left\|\theta-\mu\right\|_\Sigma^2\right\}, \tag{2}$$

where $\mu \in \mathbb{R}^n$ is the mean, Σ is an $n \times n$ covariance matrix, and

$$\left\|\theta-\mu\right\|_\Sigma^2 \overset{\Delta}{=} (\theta-\mu)^\top \Sigma^{-1} (\theta-\mu) \tag{3}$$

denotes the squared Mahalanobis distance.

1.3 Bayes Nets and Mixture Models

Continuous Bayes nets are exactly like discrete Bayes nets, except with continuous variables. And we can mix and match discrete and continuous variables. A particularly simple Bayes net is a **Gaussian mixture model**.

Continuous probability densities present a representational challenge. We can no longer specify CPTs: either an equation needs to be available, as with the Gaussian density above, or an arbitrary density has to be somehow approximated. One such approximation is exactly using mixture densities: we can "mix" many simpler densities, like Gaussians, to approximate a more complicated density. This is known as **Parzen window density estimation**.

1.4 Continuous Measurement Models

In many cases it is both justified and convenient to model measurements as corrupted by zero-mean Gaussian noise. For example, a bearing measurement from a given pose x to a given landmark l would be modeled as

$$z = h(x, l) + \eta, \tag{4}$$

where $h(.)$ is a **measurement prediction function**, and the noise η is drawn from a zero-mean Gaussian density with measurement covariance R. This yields the following conditional density $p(z|x, l)$ on the measurement z:

$$p(z|x, l) = \mathcal{N}(z; h(x, l), R) = \frac{1}{\sqrt{|2\pi R|}} \exp\left\{-\frac{1}{2}\|h(x, l) - z\|_R^2\right\}. \tag{5}$$

The **measurement functions** $h(.)$ are often nonlinear in practical robotics applications. Still, while they depend on the actual sensor used, they are typically not difficult to reason about or write down. The measurement function for a 2D bearing measurement is simply

$$h(x, l) = atan2(l_y - x_y, l_x - x_x), \tag{6}$$

where $atan2$ is the well-known two-argument arctangent variant. Hence, the final **probabilistic measurement model** $p(z|x, l)$ is obtained as

$$p(z|x, l) = \frac{1}{\sqrt{|2\pi R|}} \exp\left\{-\frac{1}{2}\|atan2(l_y - x_y, l_x - x_x) - z\|_R^2\right\}. \tag{7}$$

Note that we will not *always* assume Gaussian measurement noise: to cope with the occasional data association mistake, we can use robust measurement densities, with heavier tails than a Gaussian density.

1.5 Continuous Motion Models

For a robot operating in the plane, probabilisitic **motion models** are densities of the form $p(x_{t+1}|x_t)$, specifying a **probabilistic motion model** which the robot is assumed to obey. This *could* be derived from odometry measurements, in which case we would proceed exactly as described above. Alternatively, such a motion model could arise from known control inputs u_t. In practice, we often use a conditional Gaussian assumption,

$$p(x_{t+1}|x_t, u_t) = \frac{1}{\sqrt{|2\pi Q|}} \exp\left\{-\frac{1}{2}\|g(x_t, u_t) - x_{t+1}\|_Q^2\right\}, \tag{8}$$

where $g(.)$ is a motion model, and Q a covariance matrix of the appropriate dimensionality, e.g., 3×3 in the case of robots operating in the plane.

1.6 Simulating from a Continuous Bayes Net

1.7 Sampling as an Approximate Representation

1.8 Importance Sampling

Sampling from a factor graph can be done using Gibbs sampling, and will lead to better convergence, but is computationally heavy.

1.9 Particle Filters and Monte Carlo Localization

There are two ways to intuitively derive particle filtering:

1. Let us assume that we have a set of samples on X_{t-1}, i.e., $\{X_{t-1}^{(s)}\} \sim P(X_{t-1}|Z^{t-1})$. We create a tiny Bayes net, $X_{t-1} \rightarrow X_t \rightarrow Z_t$, and just extend the sample set as in ancestral sampling, obtaining $\left\{(X_{t-1}, X_t)^{(s)}\right\} \sim P(X_{t-1}, X_t|Z^{t-1})$. We then marginalize to $\{X_t^{(s)}\} \sim P(X_t|Z^{t-1})$, weight by the likelihood to obtain $\{X_t^{(s)}, \pi_t^{(s)}\} \sim P(X_t|Z^t)$. Finally, **resample** to get back to an unweighted set of particles for the next step.

2. Let us assume that we have a *weighted* set of samples by marginalizing out X_{t-1}, i.e., $\{X_{t-1}^{(r)}, \pi_{t-1}^{(r)}\} \sim P(X_{t-1}|Z^{t-1})$. We can form a mixture density that approximates the **predictive density** $P(X_t|Z^{t-1}) \approx \sum_r \pi_{t-1}^{(r)} P(X_t|X_{t-1}^{(r)})$, sample from it to get $\{X_t^{(s)}\} \sim P(X_t|Z^{t-1})$, and then weight the samples using the likelihood: $\{X_t^{(s)}, \pi_t^{(s)}\} \sim P(X_t|Z^t)$, where $\pi_t^{(s)} = L(X_t^{(s)}; Z_t) = kP(Z_t|X_t^{(s)})$.

The former derivation is a straight extension of importance sampling as in the previous section, but the resampling step is somewhat magical. The latter derivation shows that the resampling can be interpreted as approximating the predictive density for X_t. Of course, both derivations lead to the exact same algorithm.

1.10 Connection with the Elimination Algorithm*

Factor graphs were not harmed in the making of this filter. But, you can interpret the weighted samples as a factor on X_{t-1}, then add a motion model and a likelihood factor, resp. to and on X_t. Eliminating X_{t-1} forms the product $f_1(X_{t-1})P(X_t|X_{t-1})$ and tries to re-write it as $P(X_{t-1}|X_t)\tau(X_t)$. This is normally done by marginalization, so indeed we get $\tau(X_t) = \sum f_1(X_{t-1})P(X_t|X_{t-1})$ and hence we see that $\tau(X_t)$ is indeed the predictive density. We don't need $P(X_{t-1}|X_t)$ so we ignore it. Then, eliminating X_t would form the product $\tau(X_t)L(X_t; Z_t)$. This is hard if we modify the mixture components, but easy if we first sample from them: we just re-weight then with the likelihood, and we are done. Hence, the FG interpretation closely aligns with derivation 2 above.

Summary

We briefly summarize what we learned in this section:

1. Continuous Densities
2. Gaussian Densities
3. Bayes Nets & Mixture Models
4. Continuous Measurement Models
5. Continuous Motion Models
6. Simulating Continuous Bayes Nets
7. Sampling as Approximation
8. Importance Sampling

9. Particle Filters and Monte Carlo Localization
10. Monte Carlo Localization & the Elimination Algorithm