14
Parametric Filters

The previous chapter introduced a probabilistic framework that can be used to design algorithms for robot localization and state estimation. The chapter concluded with the introduction of the Bayes filter, which is a fundamental algorithm for maintaining and updating a belief distribution (a probability distribution over possible states). While the Bayes filter is generally intractable to implement in practice, it lays a mathematical foundation for the development of algorithms that can exploit structure or other approximations to generate tractable approaches. One such example is the discrete Bayes filter, which assumes that the number of possible states is finite such that the belief distribution can be represented by simply storing the probability of each state individually. This type of distribution is referred to as *non-parametric* since the belief distribution is not required to have a particular structure.

Alternatively, it is possible to develop tractable algorithms for probabilistic localization and state estimation by leveraging *parametric* belief distributions. Parametric distributions are distributions that are fully specified by a fixed number of parameters, for example Gaussian distributions are defined by the mean and covariance parameters. These *parametric filters* can generally be viewed as practical implementations of Bayes filter that exploit structure for efficiency, and include the Kalman filter family of algorithms.

Parametric Filters

Parametric filters[1] are a family of algorithms for robot localization and state estimation that model the robot's belief with parametric distributions. Therefore, as the robot's state evolves and new measurement information arrives, updating the belief distribution is accomplished by simply updating the parameters that define the distribution. This can lead to practically implementable algorithms since the number of parameters is generally not too large. For example, a Gaussian distribution in one dimension only requires the specification of two parameters: the mean and standard deviation. Yet with these two parameters a probability distribution is defined over an infinite number of values! This is an example of how parametric filters exploit structure for efficiency.

[1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005

14.1   Gaussian Distribution

The Gaussian distribution (also referred to as a normal distribution) is likely the most commonly used parametric distribution in many disciplines, including robotics. The probability density function for a one-dimensional (univariate) Gaussian distribution is given by:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}, \tag{14.1}$$

where the parameters are the mean μ and standard deviation σ (the quantity σ^2 is referred to as the variance). A shorthand notation for saying that a random variable X is distributed according to a Gaussian (normal) distribution is $X \sim \mathcal{N}(\mu, \sigma^2)$. For the multi-dimensional case, the multivariate Gaussian distribution is defined by the probability density function:

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \tag{14.2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and the parameters are the mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$. Again, a shorthand to say a random variable X is distributed according to the multivariate Gaussian distribution is $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. These distributions are represented graphically for the univariate and bivariate case in Figure 14.1. These distributions exhibit several important properties which
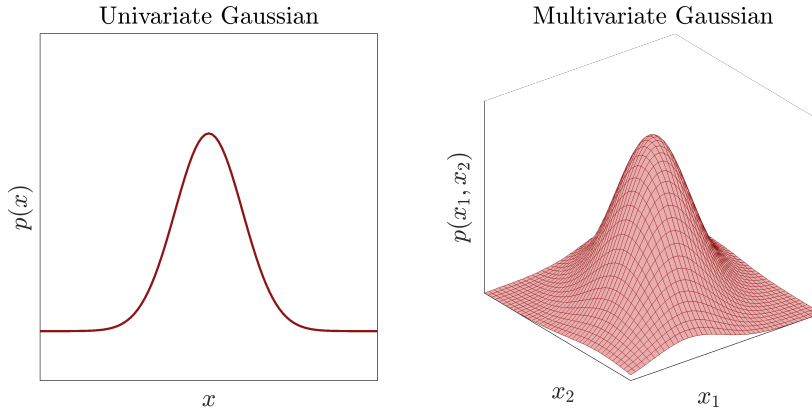

Figure 14.1: Univariate and multivariate Gaussian distributions.

make them particularly attractive for algorithm development:

1. The affine transformation of a Gaussian random variable is also a Gaussian random variable. In particular, suppose the random variable X has a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Then the random variable Y resulting from an affine transformation:

$$Y = AX + b,$$

also has a multivariate Gaussian distribution with expected value $A\boldsymbol{\mu} + b$ and covariance $A\boldsymbol{\Sigma}A^T$. In other words, if $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ then $Y \sim \mathcal{N}(A\boldsymbol{\mu} + b, A\boldsymbol{\Sigma}A^T)$.

2. The sum of two independent Gaussian random variables is also a Gaussian random variable. In particular, suppose X_1 and X_2 have multivariate Gaussian distributions with means μ_1 and μ_2 and covariances Σ_1 and Σ_2. Then the random variable Y given by the sum:

$$Y = X_1 + X_2,$$

also has a multivariate Gaussian distribution with expected value $\mu_1 + \mu_2$ and covariance $\Sigma_1 + \Sigma_2$. In other words, if $X_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $X_2 \sim \mathcal{N}(\mu_2, \Sigma_2)$ then $Y \sim \mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$.

3. The product of two Gaussian random variables is also a Gaussian random variable.

14.2   Kalman Filter

The Kalman filter is an extremely well known algorithm for state estimation that leverages the Gaussian distribution for efficiency. Unlike the discrete Bayes filter from the previous chapter, this filter can be applied to problems with *continuous* states. In particular, a multivariate Gaussian distribution is used to parameterize the belief distribution over possible states, in other words the state $x_t \sim \mathcal{N}(\mu_t, \Sigma_t)$, and in long form this can be expressed as:

$$bel(x_t) = p(x_t) = \frac{1}{\sqrt{\det(2\pi\Sigma_t)}} \exp\left(-\frac{1}{2}(x_t - \mu_t)^T \Sigma_t^{-1}(x_t - \mu_t)\right).$$

14.2.1   Assumptions

To ensure that the belief *remains* Gaussian after the prediction and measurement update steps of the filtering algorithm, several additional assumptions are required. First, it is assumed that the initial belief $bel(x_0)$ is Gaussian with $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ and that the state transition model is linear and is given by:

$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t, \tag{14.3}$$

where x_{t-1} is the previous state, u_t is the most recent control input, and ϵ_t is an independent *process noise* that is normally distributed with $\epsilon_t \sim \mathcal{N}(0, R_t)$. Because of the properties of the Gaussian distribution presented earlier, this affine model preserves the Gaussian structure. In particular, the state transition model can be explicitly written as:

$$p(x_t \mid x_{t-1}, u_t) = \frac{1}{\sqrt{\det(2\pi R_t)}} \exp\left(-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1}(x_t - A_t x_{t-1} - B_t u_t)\right).$$

In other words, this can be expressed in shorthand as $x_t \sim \mathcal{N}(A_t x_{t-1} + B_t u_t, R_t)$.

Second, the measurement model is also assumed to be linear, which again preserves the Gaussian structure:

$$z_t = C_t x_t + \delta_t, \tag{14.4}$$

where δ_t is an independent measurement noise that is normally distributed with $\mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$. Again, this implies the measurement model can be expressed probabilistically as:

$$p(z_t \mid x_t) = \frac{1}{\sqrt{\det(2\pi \mathbf{Q}_t)}} \exp\left(-\frac{1}{2}(z_t - C_t x_t)^T \mathbf{Q}_t^{-1}(z_t - C_t x_t)\right),$$

and in shorthand as $z_t \sim \mathcal{N}(C_t x_t, \mathbf{Q}_t)$.

To summarize, if the belief is modeled as a Gaussian distribution and the state transition and measurement models are both linear with Gaussian noise, then the Bayes filter updates can be applied and the belief will always remain Gaussian (i.e. the prediction and measurement correction steps will not warp or alter the *structure* of the belief distribution)! This results in a very practically efficient algorithm since now only the parameters μ and Σ need to be updated by the algorithm.

14.2.2   *Algorithm*

The Kalman filter algorithm is a recursive Bayes filter whose prediction and measurement correction steps take on a special form due to the structure of the Gaussian belief distributions and the assumptions listed above. In particular, the Kalman filter algorithm is given in Algorithm 1.

Algorithm 1: Kalman Filter Algorithm

   Data: $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$
   Result: μ_t, Σ_t
   $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
   $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$
   $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
   $\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$
   $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$
   return μ_t, Σ_t

In this algorithm, the first two steps define the predicted mean $\bar{\mu}_t$ and covariance $\bar{\Sigma}_t$, and the remaining steps perform the measurement correction. The matrix K_t that is computed for the measurement correction is typically referred to as the *Kalman gain*.

14.2.3   *Practical Considerations*

Due to the exploitation of the Gaussian distribution, the Kalman filter is a computationally efficient algorithm that can handle continuous state values. However, the consideration of Gaussian beliefs also restricts the flexibility of the probabilistic model. In particular, the belief is forced to be unimodal which may limit performance. Additionally, the assumption about the linearity of the state transition and measurement models may not be very accurate for some robots

and certain sensors, which can make the Kalman filter not perform well for some robotics applications.

14.2.4 Derivation

While it is possible to derive the Kalman filter algorithm by evaluating the Bayes filter updates from the previous chapter (i.e. computing the integral of $p(x_t \mid x_{t-1}, u_t)p(x_{t-1}$, etc.), it is more intuitive to directly leverage the properties of Gaussians presented in the preceding section. First, from the prior belief distribution $bel(x_{t-1}) \sim \mathcal{N}(\mu_{t-1}, \Sigma_{t-1})$ the predicted belief $\overline{bel}(x_{t-1})$ can be computed by using the affine transformation property of Gaussian random variables and the sum of two independent Gaussian random variables property. Specifically, these properties can be applied to the assumed linear state transition model (14.3) to give:

$$\bar{\mu}_t = A_t \mu_{t-1} + B_t \bar{u}_t + \mathbf{0},$$

where the $\mathbf{0}$ is the mean of the independent noise $\epsilon_t \sim \mathcal{N}(\mathbf{0}, R_t)$. The covariance properties can similarly be used to give:

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t.$$

For the measurement update it is possible to again use the properties of Gaussians to simplify the derivation of the Kalman filter measurement correction step. In particular, that the product of two Gaussians is also Gaussian. In fact, the product of the two Gaussians:

$$bel(x_t) = p(z_t \mid x_t)\overline{bel}(x_t) = \mathcal{N}(C_t x_t, Q_t)\mathcal{N}(\bar{\mu}_t, \bar{\Sigma}_t),$$

can be expressed as:

$$bel(x_t) = \eta \exp\left(-\frac{1}{2} J_t\right),$$

where η is a normalization constant and:

$$J_t = (z_t - C_t x_t)^T Q_t^{-1}(z_t - C_t x_t) + (x_t - \bar{\mu}_t)^T \bar{\Sigma}_t^{-1}(x_t - \bar{\mu}_t).$$

To determine the mean μ_t and covariance Σ_t of this new Gaussian, one simple approach is just take the first and second derivative of J_t with respect to x_t. The mean is found where the first derivative is zero, and the covariance is the (inverse) of the constant second derivative:

$$0 = -C_t^T Q_t^{-1}(z_t - C_t \mu_t) + \bar{\Sigma}_t^{-1}(\mu_t - \bar{\mu}_t),$$
$$\Sigma_t^{-1} = C_t^T Q_t^{-1} C_t + \bar{\Sigma}_t^{-1}.$$

Through some algebraic manipulation the mean can be written in terms of the covariance Σ_t:

$$\mu_t = \bar{\mu}_t + \Sigma_t C_t^T Q_t^{-1}(z_t - C_t \bar{\mu}_t),$$

and of course the covariance can be written as:

$$\Sigma_t = (C_t^T Q_t^{-1} C_t + \bar{\Sigma}_t^{-1})^{-1}.$$

While it is technically possible to stop here, this is not quite the form of the Kalman filter equations. In particular a few more algebraic steps are needed, based on the matrix inversion lemma result:

$$(C_t^T Q_t^{-1} C_t + \bar{\Sigma}_t^{-1})^{-1} = \bar{\Sigma}_t - \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} C_t \bar{\Sigma}_t.$$

By choosing to define the Kalman gain as $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$ it can be seen that the covariance can be expressed as:

$$\Sigma_t = \bar{\Sigma}_t - K_t C_t \bar{\Sigma}_t,$$

Through some additional algebra, the expression for the mean can also be expressed in terms of the Kalman gain and simplified to its final form:

$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t).$$

For more details on the algebraic steps see Thrun et al.[2].

[2] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* MIT Press, 2005

14.3   Extended Kalman Filter (EKF)

The extended Kalman filter (EKF) is a modified version of the Kalman filter that revisits the assumption of linearity for the state transition and measurement models. This extension still exploits the Gaussian distribution to represent the belief in a computationally efficient parametric way, but by generalizing to more complex models the EKF can be applied to a wider variety of robotics state estimation and localization problems. In particular, the EKF considers general nonlinear state transition and measurement models defined as:

$$\begin{aligned} x_t &= g(u_t, x_{t-1}) + \epsilon_t, \\ z_t &= h(x_t) + \delta_t, \end{aligned} \tag{14.5}$$

where again $\epsilon_t \sim \mathcal{N}(0, R_t)$ and $\delta_t \sim \mathcal{N}(0, Q_t)$ are normally distributed noise terms.

The EKF handles these nonlinear functions by taking advantage of their first order Taylor series expansions (which are linear functions, similar to those used in the Kalman filter). In particular the Taylor series expansion of the state transition model g is performed about the *most likely state* from the current belief distribution, which is the expected value μ_{t-1}:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}),$$

where $G_t = \nabla_x g(u_t, \mu_{t-1})$ is the Jacobian of g evaluated at μ_{t-1}. From this linear approximation the state transition model can be expressed as:

$$p(x_t \mid x_{t-1}, u_t) = \frac{1}{\sqrt{\det(2\pi R_t)}} \exp\left(-\frac{1}{2} \Delta x_t^T R_t^{-1} \Delta x_t \right),$$

where

$$\Delta x_t = x_t - g(u_t, \mu_{t-1}) - G_t(x_{t-1} - \mu_{t-1}).$$

From this result, the linear predictions that are used in the Kalman filter algorithm can be replaced by the nonlinear generalizations:

$$\bar{\mu}_t = g(u_t, \mu_{t-1}),$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t.$$

As can be seen the prediction of the new mean is simply an evaluation of the nonlinear function g, and the updated covariance is very similar to the Kalman filter but leverages the Jacobian G_t.

   A very similar procedure is used for the measurement corrections. The measurement model is also Taylor series expanded to yield (this time about the *predicted* point $\bar{\mu}_t$):

$$h(x_t) \approx h(\bar{\mu}_t) + H_t(x_t - \bar{\mu}_t),$$

where $H_t = \nabla_x h(\bar{\mu}_t)$ is the Jacobian of h evaluated at $\bar{\mu}_t$. The measurement model can then be expressed using this approximation as:

$$p(z_t \mid x_t) = \frac{1}{\sqrt{\det(2\pi Q_t)}} \exp\left(-\frac{1}{2}\Delta z_t^T Q_t^{-1} \Delta z_t\right),$$

where $\Delta z_t = z_t - h(\bar{\mu}_t) - H_t(x_t - \bar{\mu}_t)$. From this result the measurement correction in the EKF can be shown to be similar to the Kalman filter, but where the Jacobians H_t are used:

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)),$$
$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t,$$

where the Kalman gain is $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$.

14.3.1   Algorithm

The extended Kalman filter algorithm is quite similar to the Kalman filter algorithm outlined in Algorithm 1. In particular the main differences are that the updates use the nonlinear functions and their Jacobians rather than assuming strictly linear models. The EKF algorithm is outlined in Algorithm 2.

14.3.2   Practical Considerations

The extended Kalman filter can provide more accurate results than the Kalman filter in many applications due to its ability to consider more general nonlinear models. However, the approximation of the nonlinear models by a Taylor series expansion can lead to the filter to diverge if the approximation is not accurate enough. Additionally, the EKF still suffers from the same unimodal modeling assumption as the Kalman filter since the beliefs are still represented by a single Gaussian distribution.

Algorithm 2: Extended Kalman Filter Algorithm

Data: $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$

Result: μ_t, Σ_t

$\bar{\mu}_t = g(u_t, \mu_{t-1})$

$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$

$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$

$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$

$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$

return μ_t, Σ_t

14.4   Unscented Kalman Filter

The unscented Kalman filter (UKF) is another variation of the Kalman filter (still uses Gaussian distribution to parameterize the belief). This filter is also similar to the EKF in that it can handle nonlinear state transition and measurement models. However, this filter improves upon the EKF by not relying on Taylor series expansions, which can cause filter divergence due to approximation errors. This is accomplished by representing the Gaussian through a set of *sigma-points* that are transformed through the nonlinear functions. Once each sigma-point has been updated, a new Gaussian distribution is computed to represent the updated belief.

14.5   Exercises

14.5.1   EKF Localization

Complete *Problem 1: EKF Localization* located in the online repository:

    https://github.com/PrinciplesofRobotAutonomy/AA274A_HW4,

where you will implement an EKF for robot localization based on linear feature extraction and a map of known features.

Bibliography

[1]   S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.