Control using Bayesian Filtering

Carl Edward Rasmussen

September 25th, 2014

The *state* of the control system contains two parts:

1. the *physical state* x,
2. the *information state* distribution $z \sim \mathcal{N}(b, V)$.

Thus, the *state distribution* is in principle a distribution over the random variables, x, b and V. However, as an approximation we are going to assume that the distribution on the variance V is just a delta function (ie, that the variance is some fixed value). Assuming further that the state distribution is Gaussian

$$\begin{bmatrix} x \\ b \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_x \\ m_b \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xb} \\ \Sigma_{bx} & \Sigma_b \end{bmatrix}\right). \tag{1}$$

Controller instantiation

When an actual controller is applied, it gets three pieces of information: b, V and a noisy observation of the state, $y = x + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$. It updates its (prior) belief $\mathcal{N}(b, V)$ with the observation likelihood to obtain the posterior belief

$$z|b, y \sim \mathcal{N}\big(m_{z|b,y} = Ay + Bb, (V^{-1} + \Sigma_\epsilon^{-1})^{-1}\big), \tag{2}$$

where $A = V(V + \Sigma_\epsilon)^{-1}$ and $B = \Sigma_\epsilon(V + \Sigma_\epsilon)^{-1}$.
The controller then applies the policy f to the mean of the posterior belief to compute the action u

$$u = f(r), \text{ where } r = Ay + Bb. \tag{3}$$

Thus, the joint distribution over posterior state and action, given b and y

$$z, u|b, y \sim \mathcal{N}\left(\mu(b, y) = \begin{bmatrix} r \\ u \end{bmatrix}, \Sigma(b, y) = \begin{bmatrix} (V^{-1} + \Sigma_\epsilon^{-1})^{-1} & 0 \\ 0 & 0 \end{bmatrix}\right), \tag{4}$$

where we note that the variance $\Sigma(b, y)$ doesn't in fact depend on either b or y. Finally, the controller can predict the next state by applying the dynamics model g to the posterior $z, u|b, y$ to obtain a predicted distribution with mean and variance, given by eq. (5) and (7) and the `gph.pdf` document

$$\begin{aligned} \tilde{b}^{t+1} &= s_a^2 \beta_a^\top q\big(x_i, \mu(b, y), \Lambda_a, \Sigma(b, y)\big) + \theta_a^\top \mu(b, y), \\ \tilde{V}^{t+1} &= h\big(\mu(b, y), \Sigma(b, y)\big), \end{aligned} \tag{5}$$

where h is some longish function, which I didn't bother to write down right now. This completes the specification of the instantiated controller.

Simulation

In *simulation* we need to compute the behaviour of the controller and system over the *state distribution*. We have

$$\begin{bmatrix} x \\ r \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_x \\ Am_x + Bm_b \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_x A^\top + \Sigma_{xb} B^\top \\ A\Sigma_x + B\Sigma_{bx} & \Sigma_r \end{bmatrix}\right), \tag{6}$$

where $\Sigma_r = A(\Sigma_x + \Sigma_\epsilon)A^\top + A\Sigma_{xb}B^\top + B\Sigma_{bx}A^\top + B\Sigma_b B^\top$.
The controller function computes

$$(M, S, C) = f(Am_x + Bm_b, \Sigma_r), \tag{7}$$

where M is the predictive mean, S the predictive variance and C is the inverse of the covariance of the input time the input output covariance. Thus we have

$$\begin{bmatrix} x \\ r \\ u \end{bmatrix} \sim \mathcal{N}\left(\mu = \begin{bmatrix} m_x \\ Am_x + Bm_b \\ M \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_x & \Sigma_x A^\top + \Sigma_{xb} B^\top & (\Sigma_x A^\top + \Sigma_{xb} B^\top)C \\ A\Sigma_x + B\Sigma_{bx} & \Sigma_r & \Sigma_r C \\ C^\top(A\Sigma_x + B\Sigma_{bx}) & C^\top \Sigma_r & S \end{bmatrix}\right). \tag{8}$$

The simlulator needs to compute three sets of quanteties for the next time step: 1) the physical state distribution, 2) the information state distribution and 3) the covariance between the two.
For the physical state distribution, we apply the dynamics model g to the joint state action distribution

$$g(\begin{bmatrix} x \\ u \end{bmatrix}), \text{ where } \begin{bmatrix} x \\ u \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_x \\ M \end{bmatrix}, \begin{bmatrix} \Sigma_x & (\Sigma_x A^\top + \Sigma_{xb} B^\top)C \\ C^\top(A\Sigma_x + B\Sigma_{bx}) & S \end{bmatrix}\right), \tag{9}$$

which using the standard result for predictive mean and variance for the GP with Gaussian inputs yields m_x^{t+1} and Σ_x^{t+1}.
For the information state we have compute

$$m_b^{t+1} = \mathbb{E}[\tilde{b}^{t+1}], \quad \Sigma_b^{t+1} = \mathbb{V}[\tilde{b}^{t+1}], \quad V^{t+1} = \mathbb{E}[\tilde{V}^{t+1}], \tag{10}$$

where all expectations and variances are taken over $p(r, u)$. These expressions are given by eq. (11), (13) and (14) from the `gph.pdf` document. In detail

$$m_b^{t+1} = s_a^2 \beta_a^\top q\left(x_i, \begin{bmatrix} Am_x + Bm_b \\ M \end{bmatrix}, \Lambda_a, \begin{bmatrix} \Sigma_r + (V^{-1} + \Sigma_\epsilon^{-1})^{-1} & \Sigma_r C \\ C^\top \Sigma_r & S \end{bmatrix}\right) + \theta_a^\top \begin{bmatrix} Am_x + Bm_b \\ M \end{bmatrix}. \tag{11}$$

Finally, we need the covariance between the physical state and the information state mean which is

$$\Sigma_{xb}^{t+1} = \mathbb{E}[\tilde{b}^{t+1} g(x)] - \mathbb{E}[\tilde{b}^{t+1}]\mathbb{E}[g(x)], \tag{12}$$

where the expectation is over the joint $p(x, r, u)$ and can be calculated with the standard derivation and an extended input representation. The calculation will involve an average like this (SKETCH)

$$\begin{aligned} \mathbb{E}[\tilde{b}^{t+1} g(x)] &= s_a^2 s_b^2 \beta_a^\top \langle q(\cdot, (x, r, u)^\top, \begin{pmatrix} \infty & \\ & \Lambda_a \end{pmatrix}, W) \times q((x, r, u)^\top, \cdot, \begin{pmatrix} \Lambda_b & & \\ & \infty & \\ & & \Lambda_b \end{pmatrix}, W) \rangle \beta_b + \\ &= s_a^2 s_b^2 \beta_a^\top Q(\cdot, \cdot, \Lambda_a, \Lambda_b, W, \mu, \Sigma)\beta_b + \theta_a \Sigma \theta_b^\top, \end{aligned} \tag{13}$$

where the $\langle \ldots \rangle$ average is wrt the distibution eq. (8), the missing argument in the $\mathsf{q}()$ function are the training inputs and

$$W = \begin{pmatrix} 0 & 0 & 0 \\ 0 & (V^{-1} + \Sigma_{\epsilon}^{-1})^{-1} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{14}$$

the result of this integral is given in eq. (2), middle line of the `gph.pdf` document.