Instrumentation Tutorial 6 Answers

Instrumentation (Flinders University)

INSTRUMENTATION

ENGR4732

TUTORIAL 6: PARTICLE FILTERS

QUESTIONS

QUESTION 1

Consider robot with two dimensional motion with which we want to estimate its position and orientation. The robot has a bearing sensor that provides measurements with a 1σ uncertainty of 20 degrees. The robots motion is performed using the odometery model in "Motion Models" consisting of rotation command forward motion command and final rotation command.

The initial position and velocity are 0 m and 0^o respectively, with a 1σ uncertainty of 1 m and 10 degrees, for position and orientation respectively. Given the Bootstrap Particle Filter structure;

$\boldsymbol{x}_k^i \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^i, \boldsymbol{u}_k)|_{i=1,2,\ldots,N}$ **Prediction**

$w_k^i \propto w_{k-1}^i p(\boldsymbol{z}_k|\boldsymbol{x}_k^i)|_{i=1,2,\ldots,N}$ **Importance Sampling**

$[\boldsymbol{w}_k, \boldsymbol{x}_k] = systematic\_resampling(\boldsymbol{w}_k, \boldsymbol{x}_k)$ **Resampling**

```
[w[], x[]] = systematic_resampling(w[], x[], N)

c[1] = w[1]
for i = 2..N
      c[i] = c[i-1] + w[i]
end

u ~ U[0,1/N]

for j = 1..N
      while u[j] > c[i]
            i = i + 1
      end
      x[j] = x[i]
      w[j] = 1/N
      u = u + 1/N
end
```

S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics vol. 1: MIT press Cambridge, 2005.

Note: As seen above the pseudo code for the resampling process has been provided but only mathematical representations of the prediction and importance sampling. For convenience define the population size as *N = 10*.

(a) Define the required state vector, $\boldsymbol{x}_k$.

(b) Define a process that implements, $\boldsymbol{x}_k^i \sim p\left(\boldsymbol{x}_k \middle| \boldsymbol{x}_{k-1}^i, \boldsymbol{u}_k\right)$, this includes the motion model, $f_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k)$, and an ability to draw a sample from the process noise $\boldsymbol{Q}$.

(c) Define a process that implements, $w_k^i \propto w_{k-1}^i p\left(\boldsymbol{z}_k \middle| \boldsymbol{x}_k^i\right)$, this includes the measurement model, $h_k(\boldsymbol{x}_k)$, and an ability to calculate the support from the measurement noise $\boldsymbol{R}$.

(d) Define the initial state estimate i.e. initial particle population, $\langle \boldsymbol{x}_0, \boldsymbol{w}_0 \rangle$

S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics vol. 1: MIT press Cambridge, 2005.

QUESTION 2

Given robot in *Question 1*, with an update period of T=0.1 sec, the vehicle performs the given command sequence 20 degrees anticlockwise rotation, forward 0.1 m, and a final rotation of 10 degrees clockwise.

The command uncertainties are;

- $\alpha_1 = 0.05$
- $\alpha_2 = 0.001$
- $\alpha_3 = 0.05$
- $\alpha_4 = 0.01$

The first bearing measurement is 90 degrees to a known landmark located at (x,y) = (0,1), find;

(a) Draw the first sample, $\boldsymbol{x}_k^i \sim p\left(\boldsymbol{x}_k \middle| \boldsymbol{x}_{k-1}^i, \boldsymbol{u}_k\right)\Big|_{i=1}$

(b) Calculate the support of the first sample, $w_k^i \propto w_{k-1}^i p\left(\boldsymbol{z}_k \middle| \boldsymbol{x}_k^i\right)\Big|_{i=1}$

(c) Given the particle population, define a new population, $\langle \boldsymbol{x}_k^*, \boldsymbol{w}_k^* \rangle$, using *systematic resampling*.

$$\boldsymbol{x}_k =$$

$$\begin{bmatrix} 0.5377 & 0.8622 & -0.4336 & 2.7694 & 0.7254 & -0.2050 & 1.4090 & -1.2075 & 0.4889 & -0.3034 \\ 1.8339 & 0.3188 & 0.3426 & -1.3499 & -0.0631 & -0.1241 & 1.4172 & 0.7172 & 1.0347 & 0.2939 \\ -0.3942 & -0.2282 & 0.6245 & 0.5297 & 0.1247 & 0.2600 & 0.1172 & 0.2845 & 0.1269 & -0.1374 \end{bmatrix}$$

$$\boldsymbol{w}_k =$$

$$[\ 0.1775 \quad 0.0080 \quad 0.0696 \quad 0.0116 \quad 0.0244 \quad 0.2070 \quad 0.1747 \quad 0.0797 \quad 0.2389 \quad 0.0087]$$

S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics vol. 1: MIT press Cambridge, 2005.

INSTRUMENTATION

ENGR4732

TUTORIAL 6: PARTICLE FILTERS

QUESTIONS

QUESTION 1

Consider robot with two dimensional motion with which we want to estimate its position and orientation. The robot has a bearing sensor that provides measurements with a 1σ uncertainty of 20 degrees. The robots motion is performed using the odometery model in "Motion Models" consisting of rotation command forward motion command and final rotation command.

The initial position and velocity are 0 m and 0^o respectively, with a 1σ uncertainty of 1 m and 10 degrees, for position and orientation respectively. Given the Bootstrap Particle Filter structure;

$\boldsymbol{x}_k^i \sim p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^i, \boldsymbol{u}_k)|_{i=1,2,\ldots,N}$ **Prediction**

$w_k^i \propto w_{k-1}^i p(\boldsymbol{z}_k | \boldsymbol{x}_k^i)|_{i=1,2,\ldots,N}$ **Importance Sampling**

$[\boldsymbol{w}_k, \boldsymbol{x}_k] = systematic\_resampling(\boldsymbol{w}_k, \boldsymbol{x}_k)$ **Resampling**

```
[w[], x[]] = systematic_resampling(w[], x[], N)

c[1] = w[1]
for i = 2..N
      c[i] = c[i-1] + w[i]
end

u ~ U[0,1/N]

for j = 1..N
      while u[j] > c[i]
            i = i + 1
      end
      x[j] = x[i]
      w[j] = 1/N
      u = u + 1/N
end
```

Note: As seen above the pseudo code for the resampling process has been provided but only mathematical representations of the prediction and importance sampling. For convenience define the population size as *N = 10*.

(a) Define the required state vector, $\boldsymbol{x}_k$.

(b) Define a process that implements, $\boldsymbol{x}_k^i \sim p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^i, \boldsymbol{u}_k)$, this includes the motion model, $f_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k)$, and an ability to draw a sample from the process noise $\boldsymbol{Q}$.

(c) Define a process that implements, $w_k^i \propto w_{k-1}^i p(\boldsymbol{z}_k | \boldsymbol{x}_k^i)$, this includes the measurement model, $h_k(\boldsymbol{x}_k)$, and an ability to calculate the support from the measurement noise $\boldsymbol{R}$.

(d) Define the initial state estimate i.e. initial particle population, $\langle \boldsymbol{x}_0, \boldsymbol{w}_0 \rangle$

ANSWER

(a) $\boldsymbol{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}$

(b)

for i = 1 to N

Draw 3 random numbers according to control uncertainty

$\varepsilon_{rot1} \sim \mathcal{N}(0, \alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|)$ % Gaussian number with $\mu = 0$

$\varepsilon_{trans} \sim \mathcal{N}(0, \alpha_3 |\delta_{trans}| + \alpha_4 |\delta_{rot1} + \delta_{rot2}|)$ % and $\sigma = \alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|$

$\varepsilon_{rot2} \sim \mathcal{N}(0, \alpha_1 |\delta_{rot2}| + \alpha_2 |\delta_{trans}|)$

Add control noise to control values

$\hat{\delta}_{rot1} = \delta_{rot1} + \varepsilon_{rot1}$

$\hat{\delta}_{trans} = \delta_{trans} + \varepsilon_{trans}$

$\hat{\delta}_{rot2} = \delta_{rot2} + \varepsilon_{rot2}$

Apply corrupted control values to motion model

$x_k^i = x_{k-1}^i + \hat{\delta}_{trans} \cos(\theta_{k-1}^i + \hat{\delta}_{rot1})$

$y_k^i = y_{k-1}^i + \hat{\delta}_{trans} \sin(\theta_{k-1}^i + \hat{\delta}_{rot1})$

$\theta_k^i = \theta_{k-1}^i + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$

end

(c) for i = 1 to N

Determine measurement estimate from current robot estimate

$z_{detected} = \langle i, \alpha \rangle$ % Detected Landmark

$\hat{\alpha} = atan2\left(m_y(i) - y_k, m_x(i) - x_k\right) - \theta_k$ % Where $m_x(i)$ and $m_y(i)$ are the xy % coordinate of the i^th landmark

Update particle weight according to Gaussian likelihood

$$w_k^i = \frac{1}{\sqrt{2\pi {\varepsilon_\alpha}^2}} e^{-\frac{(\hat{\alpha}-\alpha)^2}{2{\varepsilon_\alpha}^2}} w_{k-1}^i$$

Update normalising factor

$\eta = \eta + w_k^i$

end

Normalise particle weights

for i = 1 to N

$$w_k^i = \frac{w_k^i}{\eta}$$

end

(d) Using randn() in Matlab

$$\boldsymbol{x}_0^N = \begin{bmatrix} 0.537 & 0.862 & -0.433 & 2.769 & 0.725 & -0.205 & 1.409 & -1.207 & 0.488 & -0.303 \\ 1.833 & 0.318 & 0.342 & -1.349 & -0.063 & -0.124 & 1.417 & 0.717 & 1.034 & 0.294 \\ -0.394 & -0.228 & 0.624 & 0.529 & 0.124 & 0.260 & 0.117 & 0.284 & 0.126 & -0.137 \end{bmatrix}$$

$$\boldsymbol{w}_0^N = [0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1]$$

Question 2

Given robot in *Question 1*, with an update period of T=0.1 sec, the vehicle performs the given command sequence 20 degrees anticlockwise rotation, forward 0.1 m, and a final rotation of 10 degrees clockwise.

The command uncertainties are;

- $\alpha_1 = 0.05$
- $\alpha_2 = 0.001$
- $\alpha_3 = 0.05$
- $\alpha_4 = 0.01$

The first bearing measurement is 90 degrees to a known landmark located at (x,y) = (0,10), find;

(a) Draw the first sample, $\boldsymbol{x}_k^i \sim p\left(\boldsymbol{x}_k \middle| \boldsymbol{x}_{k-1}^i, \boldsymbol{u}_k\right)\Big|_{i=1}$

(b) Calculate the support of the first sample, $w_k^i \propto w_{k-1}^i p\left(\boldsymbol{z}_k \middle| \boldsymbol{x}_k^i\right)\Big|_{i=1}$

(c) Given the particle population, define a new population, $\langle \boldsymbol{x}_k^*, \boldsymbol{w}_k^* \rangle$, using *systematic resampling*.

$\boldsymbol{x}_k^N =$

$$\begin{bmatrix} 0.5377 & 0.8622 & -0.4336 & 2.7694 & 0.7254 & -0.2050 & 1.4090 & -1.2075 & 0.4889 & -0.3034 \\ 1.8339 & 0.3188 & 0.3426 & -1.3499 & -0.0631 & -0.1241 & 1.4172 & 0.7172 & 1.0347 & 0.2939 \\ -0.3942 & -0.2282 & 0.6245 & 0.5297 & 0.1247 & 0.2600 & 0.1172 & 0.2845 & 0.1269 & -0.1374 \end{bmatrix}$$

$\boldsymbol{w}_k^N =$

$$[\ 0.1775 \quad 0.0080 \quad 0.0696 \quad 0.0116 \quad 0.0244 \quad 0.2070 \quad 0.1747 \quad 0.0797 \quad 0.2389 \quad 0.0087]$$

Answer

(a)

$$\varepsilon_{rot1} \sim \mathcal{N}\left(0, 0.05\left|-{}^{\pi}/_{9}\right| + 0.001|0.1|\right) = \mathcal{N}(0, 0.0176) \sim 0.0156$$
$$\varepsilon_{trans} \sim \mathcal{N}\left(0, 0.05|0.1| + 0.01\left|-{}^{\pi}/_{18}\right|\right) = \mathcal{N}(0, 0.0067) \sim -0.0072$$
$$\varepsilon_{rot2} \sim \mathcal{N}\left(0, 0.05\left|{}^{\pi}/_{18}\right| + 0.001|0.1|\right) = \mathcal{N}(0, 0.0088) \sim -0.0055$$

Add control noise to control values

$$\hat{\delta}_{rot1} = -{}^{\pi}/_{9} + 0.0156 = -0.3335$$
$$\hat{\delta}_{trans} = 0.1 - 0.0072 = 0.9928$$
$$\hat{\delta}_{rot2} = {}^{\pi}/_{18} - 0.0055 = 0.1690$$

Apply corrupted control values to motion model

$$x_1^1 = 0.537 + 0.1\cos(-0.394 - 0.3335) = 0.6117$$
$$y_1^1 = 1.833 + 0.1\sin(-0.394 - 0.3335) = 1.7665$$

$$\theta_1^1 = -0.394 - 0.3335 + 0.1690 = -u90.5585$$

$$\boldsymbol{x}_1^1 = \begin{bmatrix} 0.6117 \\ 1.7665 \\ -0.5585 \end{bmatrix}$$

(b)

$$\hat{\alpha} = atan2(10 - 1.7665, 0 - 0.6117\ ) + 0.5585 = 2.2035$$

$$w_k^i = \frac{1}{\sqrt{2\pi \times 0.3491^2}} e^{-\frac{(2.2035-1.5708)^2}{2\times 0.3491^2}} 0.1 = 0.0221$$

c)

```
[w[], x[]] = systematic_resampling(w[], x[], N)

c[1] = w[1]
for i = 2..N
      c[i] = c[i-1] + w[i]
end
```

$\boldsymbol{c}_k^N =$

[0.1775 0.1855 0.2551 0.2667 0.2911 0.4981 0.6728 0.7525 0.9914 1]

```
u ~ U[0,1/N]
```

$u = 0.0097$

```
for j = 1..N
      while u[j] > c[i]
            i = i + 1
      end
      x[j] = x[i]
      w[j] = 1/N
      u = u + 1/N
end

j = 1, i = 1 while false
      x[1] = x[1]
      w[1] = 0.1
      u = 0.0097 + 0.1 = 0.1097

j = 2, i = 1 while false
      x[2] = x[1]
      w[2] = 0.1
      u = 0.1097 + 0.1 = 0.2097

j = 3, i = 1 while true
      i = 2

j = 3, i = 2 while true
      i = 3
```

```
j = 3, i = 3 while false
      x[3] = x[3]
      w[3] = 0.1
      u = 0.2097 + 0.1 = 0.3097

j = 4, i = 3 while true
      i = 4

j = 4, i = 4 while true
      i = 5

j = 4, i = 5 while true
      i = 6

j = 4, i = 6 while false
      x[4] = x[6]
      w[4] = 0.1
      u = 0.3097 + 0.1 = 0.4097

j = 5, i = 6 while false
      x[5] = x[6]
      w[5] = 0.1
      u = 0.4097 + 0.1 = 0.5097

j = 6, i = 6 while true
      i = 7

j = 6, i = 7 while false
      x[6] = x[7]
      w[6] = 0.1
      u = 0.5097 + 0.1 = 0.6097

j = 7, i = 7 while false
      x[7] = x[7]
      w[7] = 0.1
      u = 0.6097 + 0.1 = 0.7097

j = 8, i = 7 while true
      i = 8

j = 8, i = 8 while false
      x[8] = x[8]
      w[8] = 0.1
      u = 0.7097 + 0.1 = 0.8097

j = 9, i = 8 while true
      i = 9

j = 9, i = 8 while false
      x[9] = x[8]
      w[9] = 0.1
      u = 0.8097 + 0.1 = 0.9097

j = 10, i = 8 while false
      x[10] = x[8]
      w[10] = 0.1
      u = 0.9097 + 0.1 = 1.0097
```