Sheet 6 solutions

June 2, 2017

Exercise 1: Distance-Only Sensor

In this exercise, you try to locate your friend using her cell phone signals. Suppose that in the map of Freiburg, the campus of the University of Freiburg is located at $m_0 = (10, 8)^T$*, and your friend's home is situated at* $m_1 = (6, 3)^T$*. You have access to the data received by two cell towers, which are located at the positions* $x_0 = (12, 4)^T$ *and* $x_1 = (5, 7)^T$*, respectively. The distance between your friend's cell phone and the towers can be computed from the intensities of your friend's cell phone signals. These distance measurements are disturbed by independent zero-mean Gaussian noise with variances* $\sigma_0^2 = 1$ *for tower* 0 *and* $\sigma_1^2 = 1.5$ *for tower* 1*. You receive the distance measurements* $d_0 = 3.9$ *and* $d_1 = 4.5$ *from the two towers.*

(a) *Is your friend more likely to be at home or at the university? Explain your calculations.*

We want to calculate the probability $p(m \mid z)$ of being at a location m, given the sensor measurements z. We can use Bayes rule:

$$p(m \mid z) = \frac{p(z \mid m)p(m)}{p(z)} \tag{1}$$

We do not have any prior information about the location, therefore we assume a uniform prior $p(m)$. $p(z)$ does not depend on m, therefore it can be regarded as a normalization factor. We can see that without prior information, $p(m \mid z)$ is proportional to $p(z \mid m)$:

$$p(m \mid z) \propto p(z \mid m) \tag{2}$$

To answer the question, it is enough to check the likelihood of a measurement z, given the location m. We can assume that the measurements of both towers are independent of each other:

$$\begin{aligned} p(z \mid m) &= p(d_0, d_1 \mid m) && (3)\\ &= p(d_0 \mid m)p(d_1 \mid m) && (4) \end{aligned}$$

The distance measurements of the towers are disturbed by zero-mean Gaussian noise. To obtain the likelihood of our measurement, we calculate the true distances

$\hat{d}$ between the towers and the query locations and compare them to the measured distances d. To this end, we evaluate our sensor model, the probability density of the normal distribution:

$$p(d \mid m) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(d-\hat{d})^2}{2\sigma^2}) \tag{5}$$

In *Python*, you can use the build-in function *scipy.stats.norm.pdf(x,μ,σ)* to evaluate the probability density of a normal distribution.

- At the university:

$$\begin{aligned}
\text{Tower 0:} \qquad \hat{d}_0 &= \sqrt{(12-10)^2+(4-8)^2} = \sqrt{20} && (6)\\
p(d_0 \mid m_0) &= \frac{1}{\sqrt{2\pi 1}} \exp(-\frac{(3.9-\sqrt{20})^2}{2\cdot 1}) && (7)\\
\text{Tower 1:} \qquad \hat{d}_1 &= \sqrt{(5-10)^2+(7-8)^2} = \sqrt{26} && (8)\\
p(d_1 \mid m_0) &= \frac{1}{\sqrt{2\pi 1.5}} \exp(-\frac{(4.5-\sqrt{26})^2}{2\cdot 1.5}) && (9)\\
\rightarrow \qquad p(d_0, d_1 \mid m_0) &= 0.0979 && (10)
\end{aligned}$$

- At home:

$$\begin{aligned}
\text{Tower 0:} \qquad \hat{d}_0 &= \sqrt{(12-6)^2+(4-3)^2} = \sqrt{37} && (11)\\
p(d_0 \mid m_1) &= \frac{1}{\sqrt{2\pi 1}} \exp(-\frac{(3.9-\sqrt{37})^2}{2\cdot 1}) && (12)\\
\text{Tower 1:} \qquad \hat{d}_1 &= \sqrt{(5-6)^2+(7-3)^2} = \sqrt{17} && (13)\\
p(d_1 \mid m_1) &= \frac{1}{\sqrt{2\pi 1.5}} \exp(-\frac{(4.5-\sqrt{17})^2}{2\cdot 1.5}) && (14)\\
\rightarrow \qquad p(d_0, d_1 \mid m_1) &= 0.0114 && (15)
\end{aligned}$$

It is more likely to obtain the given measurements if our friend is at the university.

(b) *Implement a function in Python which generates a 3D-plot of the likelihood $p(z|m)$ over all locations m in the vicinity of the towers. Furthermore, mark m_0, m_1, x_0 and x_1 in the plot. Is the likelihood function which you plotted a probability density function? Give a reason for your answer.*

```python
#!/usr/bin/env python

import math
import numpy as np
import scipy.stats
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

```python
from matplotlib import cm


def likelihood(m):
  """Calculate the likelihood that your friend is at place m.

  Arguments:
  m -- place [x,y]
  """

  x_0 = np.array([12,4]) # tower 0
  x_1 = np.array([5,7]) # tower 1
  d_0 = 3.9 # distance measurement 0
  d_1 = 4.5 # distance measurement 1
  var_0 = 1 # variance 0
  var_1 = 1.5 # variance 1

  #calculate the expected distance measurements
  d_0_hat = math.sqrt(np.sum(np.square(m-x_0)))
  d_1_hat = math.sqrt(np.sum(np.square(m-x_1)))

  #evaluate sensor model
  pdf_0 = scipy.stats.norm.pdf(d_0, d_0_hat,math.sqrt(var_0))
  pdf_1 = scipy.stats.norm.pdf(d_1, d_1_hat,math.sqrt(var_1))

  return pdf_0 * pdf_1


#locations of interest
m_0 = np.array([10,8]) # uni
m_1 = np.array([6,3]) # home
x_0 = np.array([12,4]) # tower 0
x_1 = np.array([5,7]) # tower 1

#mesh grid for plotting
x = np.arange(3.0,15.0,0.5)
y = np.arange(-5.0,15.0,0.5)
X,Y = np.meshgrid(x,y)

#calculate likelihood for each position
z = np.array([likelihood(np.array([x,y])) for x,y in zip(X.flatten(), Y.flatten())])
Z = z.reshape(X.shape)

#plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X,Y,Z,rstride=1,cstride=1,cmap=cm.coolwarm,alpha=0.5)
```

```
ax.scatter(m_0[0],m_0[1],likelihood(m_0),c='g',marker='o',s=100)
ax.scatter(m_1[0],m_1[1],likelihood(m_1),c='r',marker='o',s=100)
ax.scatter(x_0[0],x_0[1],likelihood(x_0),c='g',marker='^',s=100)
ax.scatter(x_1[0],x_1[1],likelihood(x_1),c='r',marker='^',s=100)

ax.set_xlabel('m_x')
ax.set_ylabel('m_y')
ax.set_zlabel('likelihood')

plt.show()
```

The plotted likelihood is not a probability density function, because we are plotting $p(z \mid m)$ over map locations m, not measurements z. To get the probability distribution $p(m \mid z)$ over map locations m, we need to know $p(m)$ and $p(z)$ to normalize the distribution.

(c) *Now, suppose you have prior knowledge about your friend's habits which suggests that your friend currently is at home with probability $P(\text{at home}) = 0.7$, at the university with $P(\text{at university}) = 0.3$, and at any other place with $P(\text{other}) = 0$. Use this prior knowledge to recalculate the likelihoods of a).*

We use Bayes Rule from Eq. 1. We can either (a) calculate $p(z)$ by summing over all possible values (law of total probability)

$$p(z) = \sum_i p(z|m_i)p(m_i), \tag{16}$$

or (b) solve Eq. 1 by normalizing.

(a) Explicitly calculate $p(z)$:

$$\begin{aligned} p(z) = p(d_0, d_1) &= p(d_0, d_1 \mid m_0)p(m_0) + p(d_0, d_1 \mid m_1)p(m_1) && (17) \\ &= 0.0979 \cdot 0.3 + 0.0114 \cdot 0.7 = 0.0374 && (18) \end{aligned}$$

In 1:

$$\begin{aligned} \text{Uni:} \quad p(m_0 \mid d_0, d_1) &= \frac{p(d_0, d_1 \mid m_0)p(m_0)}{p(d_0, d_1)} && (19) \\ &= \frac{0.0979 \cdot 0.3}{0.0374} = 0.786 && (20) \\ \text{Home:} \quad p(m_1 \mid d_0, d_1) &= \frac{p(d_0, d_1 \mid m_1)p(m_1)}{p(d_0, d_1)} && (21) \\ &= \frac{0.01147 \cdot 0.7}{0.0374} = 0.214 && (22) \end{aligned}$$

(b) Solve by normalizing:

$$\begin{aligned} \text{Uni:} \quad p(m_0 \mid d_0, d_1) &= \mu \cdot p(d_0, d_1 \mid m_0)p(m_0) && (23) \\ \text{Home:} \quad p(m_1 \mid d_0, d_1) &= \mu \cdot p(d_0, d_1 \mid m_1)p(m_1) && (24) \end{aligned}$$

We use the fact that both probabilities need to sum up to 1 to calculate the normalization factor μ:

$$1 = \mu \cdot p(d_0, d_1 \mid m_0)p(m_0) + \mu \cdot p(d_0, d_1 \mid m_0)p(m_1) \tag{25}$$
$$\mu = \frac{1}{p(d_0, d_1 \mid m_0)p(m_0) + p(d_0, d_1 \mid m_1)p(m_1)} \tag{26}$$

We can see that the normalization factor μ is just the reciprocal of $p(z)$. Normalization therefore results in exactly the same calculations like in (a).

Exercise 2: Sensor Model

Assume you have a robot equipped with a sensor capable of measuring the distance and bearing to landmarks. The sensor furthermore provides you with the identity of the observed landmarks.

A sensor measurement $z = (z_r, z_\theta)^T$ is composed of the measured distance z_r and the measured bearing z_θ to the landmark l. Both the range and the bearing measurements are subject to zero-mean Gaussian noise with variances σ_r^2, and σ_θ^2, respectively. The range and the bearing measurements are independent of each other.

A sensor model

$$p(z \mid x, l)$$

models the probability of a measurement z of landmark l observed by the robot from pose x.

Design a sensor model $p(z \mid x, l)$ for this type of sensor. Furthermore, explain your sensor model.

We want to design a sensor model $p(z \mid x, l)$ to calculate the probability to obtain a measurement z, given a pose x and a landmark pose l. A sensor measurement $z = (z_r, z_\theta)$ consists of a distance z_r and angle z_θ measurement for the landmark. Both measurements are subject to Gaussian noise with variances σ_r^2 and σ_θ^2, respectively. The robot pose is given by $x = (x_x, x_y, x_\theta)$, the landmark position by $l = (l_x, l_y)$.

Range and bearing measurements are independent:

$$p(z \mid x, l) = p(z_r, z_\theta \mid x, l) = p(z_r \mid x, l)p(z_\theta \mid x, l) \tag{27}$$

To evaluate our sensor model, we need to evaluate the probability density of a normal distribution at the measurement z_i, with the expected measurement $z_{\text{exp},i}$ as the mean and the standard deviation σ_i:

$$p(z_i \mid x, l) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{(z_i - z_{\text{exp},i})^2}{2\sigma_i^2}) \tag{28}$$
$$= \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp(-\frac{(\Delta z_i)^2}{2\sigma_i^2}) \tag{29}$$

1. range measurement:

   For the range, the expected distance measurement to the landmark is

$$z_{\text{exp},r} = \sqrt{(l_x - x_x)^2 + (l_y - x_y)^2}. \tag{30}$$

   Δz is the difference between the measured and the true distance to landmark l:

$$\begin{aligned} \Delta z_r &= z_r - z_{\text{exp},r} && (31) \\ \rightarrow \quad p(z_r \mid x, l) &= \frac{1}{\sqrt{2\pi\sigma_r^2}} \exp(-\frac{(\Delta z_r)^2}{2\sigma_r^2}) && (32) \end{aligned}$$

2. bearing measurement:

   For the bearing, the expected angle measurement to the landmark is

$$z_{\text{exp},\theta} = \operatorname{atan2}((l_y - x_y), (l_x - x_x)) - x_\theta. \tag{33}$$

   We cannot simply subtract the expected and measured bearing because of the discontinuity of the angle representation between $(-\pi, \pi]$. We use the dot product rule to find the smallest angle between the unit vectors $\vec{v}_z = (\cos(z_\theta), \sin(z_\theta))^T$ and $\vec{v}_{z,\text{exp}} = (\cos(z_{\text{exp},\theta}), \sin(z_{\text{exp},\theta}))^T$, defined by z_θ and $z_{\text{exp},\theta}$:

$$\begin{aligned} \vec{v}_z \bullet \vec{v}_{z,\text{exp}} &= |\vec{v}_z| \cdot |\vec{v}_{z,\text{exp}}| \cdot \cos(\Delta z_\theta) && (34) \\ \Delta z_\theta &= \arccos(\cos(z_\theta)\cos(z_{\text{exp},\theta}) + \sin(z_\theta)\sin(z_{\text{exp},\theta})) && (35) \\ \rightarrow \quad p(z_\theta \mid x, l) &= \frac{1}{\sqrt{2\pi\sigma_\theta^2}} \exp(-\frac{(\Delta z_\theta)^2}{2\sigma_\theta^2}) && (36) \end{aligned}$$