

Homework #1

Bayesian Statistics and Tracking

Due: Start of Thursday lecture in week 6 (25 August)

1 Overview

The goal of this assignment is to test your understanding of Bayesian Statistics and Perception.

You should feel free to ask the Lecturer questions and discuss the assignment with others, but you should not copy someone else's work. (e.g. You can get someone to show you how to answer a question, but you should write your answer up yourself from what you remember at least 30 minutes after they've left.)

Total marks for this assignment: 27. It will be scaled to 5% of your grade.

2 Bayesian Statistics

2.1 Question 2.1 (1 mark):

Give one representation for all the information known to an agent.

2.2 Question 2.2 (1 mark):

You and a friend have been designing a robot for the RoboCup robotic soccer competition. You've decided that you're going to use Bayesian probabilities to estimate the robot's location. The space of hypotheses that the robot is going to keep about its location is the 2D space of locations on the field (you're ignoring angle, and other elements of the problem to get started).

You and your friend are arguing about the robot's beliefs when it is first switched on. Which of the following is correct:

- A Bayesian statistics says that the robot's initial belief should have *all* its probability mass is on the point just next to the centre circle lined up for kickoff.
- B Bayesian statistics says that the robot's initial belief should have equal probability mass at each position on the field - the uniform distribution.
- C Bayesian statistics says that the robot's initial belief should be a mixture of the previous two options; a peak at the kickoff position, but non-zero at every location on the field.
- D Bayesian statistics doesn't say anything about the robot's initial belief, that is up to the designer.

2.3 Question 2.3 (1 mark):

You and a friend are trying to decide what to do on Friday night. She wants to go dancing, and you want to go to the lab and play with robots (Hey - they're cool). You decide to flip a coin to see what to do. She flips and wins, and you go dancing.

Later, you decide to use your new found skills in Bayesian statistics and check to see if the coin was biased. If the coin is biased you'd also like to find out how biased it is (i.e. what is the probability it would come up 'heads' when flipped?).

What space of possible hypotheses/models could you use for this investigation (and how is this space parameterised)?

2.4 Question 2.4 (1 mark):

You decide to use a uniform prior over the model space from the previous question. You then flip the coin in the first trial. It shows heads.

What is $p(\text{Heads}|m)$ for each of your models? (If your model space is continuous then this will be a function.)

2.5 Question 2.5 (1 mark):

What is the posterior distribution over your model space given the single coin flip you've seen?

2.6 Question 2.6 (1 mark):

You flip the coin again. Again it shows heads.

What is the posterior distribution over your model space when you incorporate this new information?

2.7 Question 2.7 (1 mark):

You flip the coin a third time. This time it shows tails.

What is $p(Tails|m)$ for each of your models?

2.8 Question 2.8 (1 mark):

You decide to use a one-dimensional Kalman filter to track the height of some water in a rainwater tank. You remember from class that the pointwise product of two Gaussians can be calculated as follows: $G_1(\mu_1, \sigma_1^2) \cdot G_2(\mu_2, \sigma_2^2) = C \cdot G_3(\mu_3, \sigma_3^2)$ where $\mu_3 = \frac{\sigma_2^2 \cdot \mu_1 + \sigma_1^2 \cdot \mu_2}{\sigma_1^2 + \sigma_2^2}$ and $\sigma_3^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$.

From previous measurements your belief about the height of the water in the tank has $\mu = 2.9\text{m}$ with a variance $\sigma^2 = 0.01$. You receive an observation of water height of 3.1m , the variance of observations is known to be about $\sigma^2 = 0.01$.

What is the mean and variance of the belief state distribution after incorporating this observation?

2.9 Question 2.9 (3 marks):

A friend is working on tracking cars on the road. He has defined a probability distribution over the number and location of cars using the density function defined by the java code in Figure 1.

What is the probability that there are exactly 2 or exactly 3 cars, and every car is between location 10 and 20 and has a velocity between 10 and 20 ms^{-1} ?

If your answer requires integrating a continuous distribution, then you may give the answer informally in terms of the integral. e.g. $P = \int_{-3}^6 \mathcal{N}(5.2, 6.7)$. (Make sure you're clear if the second argument to the normal is a standard deviation or a variance.) There are online calculators that can give the approximate numerical value of this integral for you if you wish.

3 Mad Lecturer Rules the Robot!!!!

There is a small legged robot in your lab that moves about a 2D flat world. At night a mad computer science lecturer is creeping into your lab to move the robot around and get it lost. You have an some Kalman filter code and decide to re-purpose it to thwart the lecturer's evil plans.

The robot has defined 'front'. It can walk forward and sideways, and turn on the spot. It turns about a central point, and has a camera mounted on a panning base immediately above this central point. All distances are measured in meters, all angles are measured in radians. All coordinate systems are right-handed with angles increasing counter-clockwise. The robot does not know its absolute position or which way it is facing in the world, and initially it is completely uncertain about its location.

```
class Car {
    double position;
    double speed;
}

class Density {

double factorial(int k) {
    if (k <= 0)
        return 1.0;
    else
        return k * factorial(k-1);
}

double poisson(double mean, double k) {
    return Math.pow(mean, k)*Math.exp(-mean)/ factorial(k);
}

double normal(double mean, double stdev, double x) {
    double d = (x-mean)/stdev;
    double variance = stdev*stdev;
    return 1/Math.sqrt(2*Math.PI*variance)*Math.exp(-d*d/2.0);
}

double density(List<Car> carList) {
    int n = carList.size();
    double pn = poisson(5, n)
    double pCars = 1;
    for (int i=0; i<n; i++) {
        Car c = carList.get(i);
        double pCar = normal(10*i, 3, c.position)
            * normal(15, 2, c.speed);
        pCars *= pCar;
    }
    return pn * pCars;
}
}
```

Figure 1: Some Java code defining a probability density function.

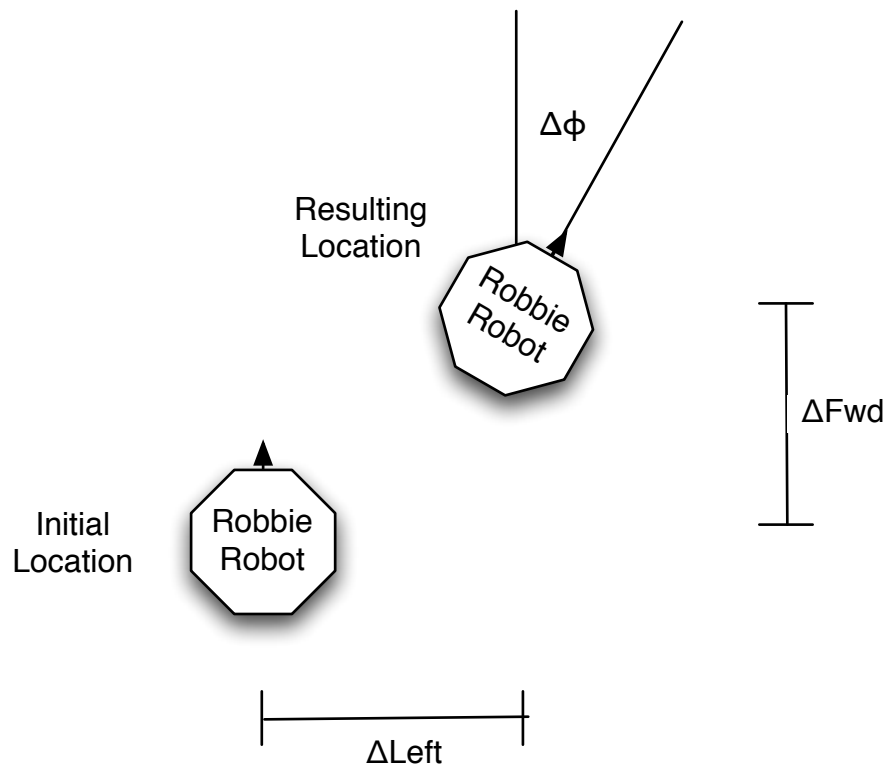


Figure 2: The numbers representing a change in position of our robot. Note that the ΔLeft and $\Delta\phi$ shown in the diagram are both negative for the displayed motion.

When the lecturer creeps into the lab, he feeds the robot's locomotion module walking commands. Unbeknownst to the lecturer, the locomotion module will be feeding those commands to the Kalman Filter as a column vector of mean movements,

$\begin{bmatrix} \Delta\text{Fwd} \\ \Delta\text{Left} \\ \Delta\phi \end{bmatrix}$, for the motion (See Figure 2).

The robot's camera is also being left switched on. As the robot is being moved about it receives observations from that camera and vision processing software. In particular there are 4 coloured beacons placed about the robot's world that it can recognise. Each time the camera detects a beacon it passes an observation to the Kalman filter. This observation is a column vector with three numbers: $\begin{bmatrix} \text{ID} \\ \text{distance} \\ \Delta\text{angle} \end{bmatrix}$, the beacon ID, and the *relative* distance and angle from the robot to the beacon. The locations of the beacons on the field are shown in figure 3.

The robot's motion and observations are not perfect. In particular, for any movement there is gaussian noise with standard deviation $0.01m + 10\%$ forward/backward,

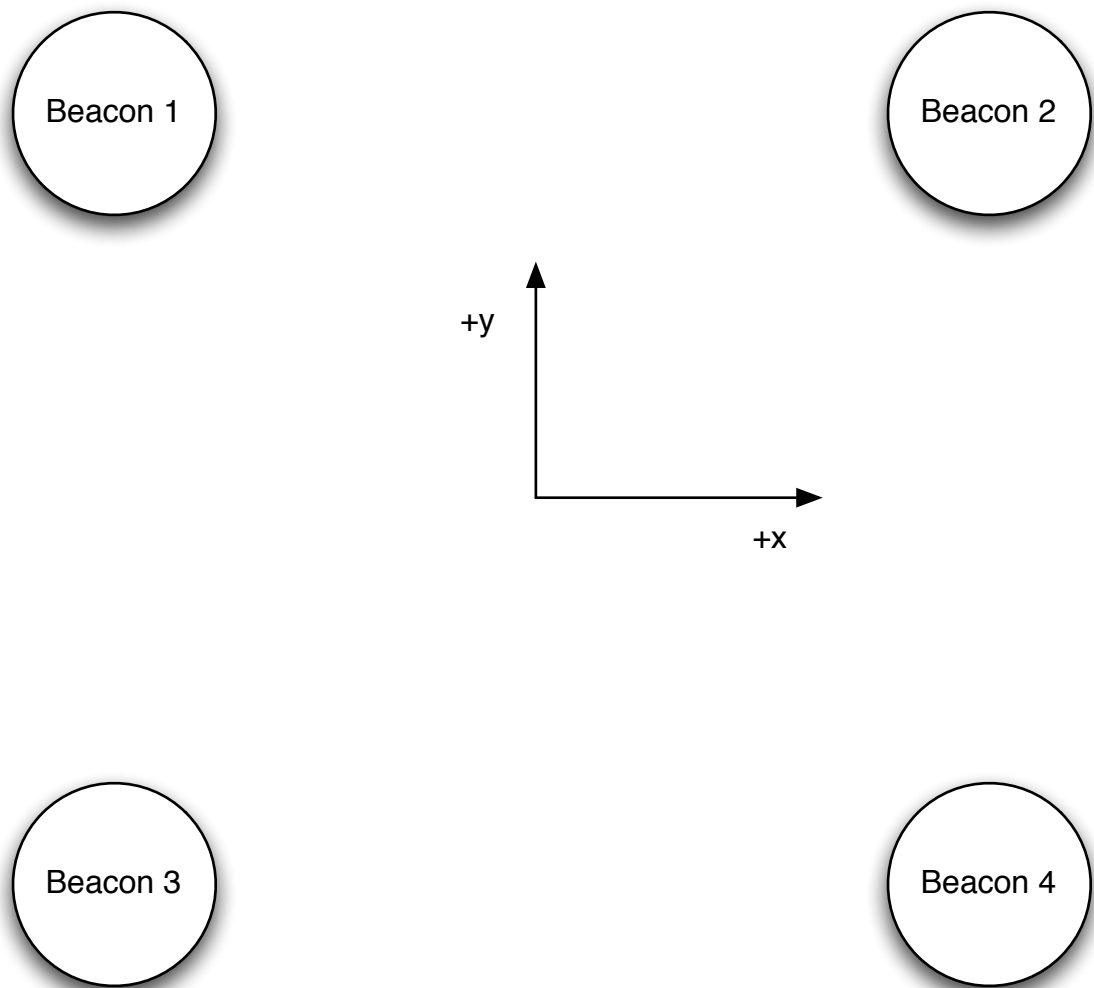


Figure 3: The coordinate system of the field. Beacons are placed at $\pm 2\text{m}$ around the origin with the shown IDs. In order the beacon locations are $(-2, 2)$, $(2, 2)$, $(-2, -2)$ and $(2, -2)$. The x -axis has a global angle of 0 radians.

$0.01m + 15\%$ left/right, and $5/180 * \pi + 10\%$ turning. (*i.e.* if the robot tries to walk forward 1m, then we expect it to have a standard deviation on that motion of $0.01 + 0.1 \times 1 = 0.11m$ forward/backward, $0.01 + 0.1 \times 0 = 0.01m$ left/right and $5/180 * \pi + 0.1 \times 0 = 5^\circ$ in angle.) For any observation there is gaussian noise with standard deviation $0.025m + 1\%$ in distance and $5/180 * \pi + 10\%$ radians in angle. Luckily, beacon identification is perfect (although you might want to think about how you would handle the case if it wasn't). Note that all percentages in this paragraph are absolute values – you don't get negative variance by moving right.

Notes:

- In the following questions, feel free to use the $\text{atan2}(dy, dx)$ function for getting angles from relative coordinates.
- Approximate the partial derivatives of the $\text{atan2}(dy, dx)$ function as follows:
 $\frac{d}{dx} \text{atan2}(dy, dx) = \frac{dy}{dx^2 + dy^2}$, and $\frac{d}{dy} \text{atan2}(dy, dx) = \frac{-dx}{dx^2 + dy^2}$.

3.1 Question 3.1 (4 marks):

Give (nonlinear) equations to describe the motion of this robot. *i.e.* give an equation for

$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix}$ in terms of $\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ and $\begin{bmatrix} \Delta Fwd \\ \Delta Left \\ \Delta \phi \end{bmatrix}$. You may use a term $N(\mu, \sigma^2)$ to describe

adding 1D gaussian noise, or $N(\vec{\mu}, \Sigma)$ to describe N -D gaussian noise, but you must specify the appropriate parameters.

3.2 Question 3.2 (4 marks):

Give three (parameterised) matrices that could be used to model the motion of this robot in an extended Kalman filter – the F , B and Q matrices describing how the robot moves relative to its previous state, the action, and giving the noise covariance. (*i.e.* the linearised version of the equations you just gave.)

3.3 Question 3.3 (4 marks):

Give (nonlinear) equations to describe the observation model of this robot. *i.e.* give an equation for the probability of an observation given a state.

3.4 Question 3.4 (4 marks):

Give two (parameterised) matrices that could be used to model the motion of this robot in an extended Kalman filter – the H and R matrices describing how the robot's observations are related to its state and the noise covariance. (*i.e.* the linearised version of the equations you just gave.) Note that the row of H for the beacon ID should be all 0.