TU München
Fakultät für Informatik
PD Dr. Rudolph Triebel
John Chiotellis

Machine Learning for Robotics and Computer Vision
Winter Term 2015

Solution Sheet 1

Topic 1: Introduction to Probabilistic Reasoning and Learning and Regression
November 6, 2015

Exercise 1:

a)

$$\begin{aligned}
p(X = green|Z = green) &= \frac{p(Z = green|X = green)p(X = green)}{p(Z = green)} \\
&= \frac{p(Z = green|X = green)p(X = green)}{\sum_{x\in\{red,green,blue\}} p(Z = green|X = x)p(X = x)} \\
&= \frac{0.6\frac{2}{5}}{0.1\frac{2}{5} + 0.6\frac{2}{5} + 0.2\frac{1}{5}} \\
&= \frac{0.24}{0.04 + 0.24 + 0.04} \\
&= \frac{0.24}{0.32} = \frac{3}{4} = 0.75
\end{aligned}$$

b)

$$p(z \mid x) = \mathcal{N}(z|x, \sigma_1^2) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{1}{2}\frac{(z-x)^2}{\sigma_1^2}} = \frac{1}{0.3\sqrt{2\pi}} e^{-5.55(z-x)^2} \quad (1)$$

where $\sigma_1 = 0.3$ is the sensor noise.

c) The motion can also be modeled with a Gaussian. We just need to think about what is our mean and what is our variance. The variance is given as the actuator noise $\sigma_2 = 0.1$. Our mean is the position we expect our robot to be at, after the motion u_t. Since our robot moves with constant speed v, the expected position is simply $\mu = x_{t-1} + v\Delta t$. Therefore we have

$$\begin{aligned}
p(x_t|x_{t-1}, u_t) &= \mathcal{N}(x_t|x_{t-1} + v\Delta t, \sigma_2^2) \\
&= \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{1}{2}\frac{(x_t-(x_{t-1}+v\Delta t))^2}{\sigma_2^2}} \\
&= \frac{10}{\sqrt{2\pi}} e^{-50(x_t-(x_{t-1}+v\Delta t))^2}
\end{aligned}$$

d) We model the state variable x as a discrete random variable with values between 0 and 5, where 0 means that the robot is at the door. We want to compute the robot's belief. Initially, the robot knows it is located at the door (x=0), therefore we have $Bel(x_0 = 0) = 1$. We then use the Bayes filter algorithm to compute the belief after 3 seconds, namely $Bel(x_3)$. Since it is a recursive algorithm we have to compute the belief at every time step. The general equation of the Bayes filter is:

$$Bel(x_t) = \eta \quad p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1} \tag{2}$$

Our u_i, namely our action is always the same: move with constant speed $v = 1m/s$. Here is the first step:

$$\begin{aligned} Bel(x_1) &= \eta_1 p(z_1|x_1) \int p(x_1|u_1, x_0) Bel(x_0) dx_0 &(3)\\ &= \eta_1 p(z_1|x_1) \int p(x_1|u_1, x_0) dx_0 &(4)\\ &= \eta_1 \mathcal{N}(z_1|x_1, \sigma_1^2) \sum_{x_0=0}^{5} \mathcal{N}(x_1|x_0 + 1, \sigma_2^2) &(5)\\ &= \eta_1 \mathcal{N}(z_1|x_1, \sigma_1^2) \mathcal{N}(x_1|x_0 + 1, \sigma_2^2) &(6) \end{aligned}$$

Let us separate the computation to the *motion* and the *sensing* part. Since we have $Bel(x_0 = 0) = 1$ we begin from the motion u_1.

$$p(x_1|u_1, x_0) = \mathcal{N}(x_1|x_0 + 1, \sigma_2^2) \tag{7}$$

The belief of the robot after the first motion can be estimated as

$$Bel'(x_1) = \sum_{x_0=0}^{5} \mathcal{N}(x_1|x_0 + 1, \sigma_2^2) \tag{8}$$

Now we take into account the sensor measurement z_1:

$$p(z_1|x_1) = \mathcal{N}(z_1|x_1, \sigma_1^2) \tag{9}$$

Therefore

$$Bel(x_1) = \eta_1 p(z_1|x_1) Bel'(x_1) \tag{10}$$

Since our positions are restricted to a space $x_t \in \{0, 1, 2, 3, 4, 5\}$, we can compute our normalizers η_i using (the inverse of) the sum of the probabilities for all possible states.

$$\eta_1^{-1} = \sum_{x_1'=0}^{5} Bel(x_1 = x_1') \tag{11}$$

If we recursively substitute the beliefs we get:

$$\begin{aligned} Bel(x_3) &= \eta_3 p(z_3|x_3) \int p(x_3|u_3, x_2) Bel(x_2) dx_2 \\ &= \eta_3 p(z_3|x_3) \int p(x_3|u_3, x_2) \eta_2 p(z_2|x_2) \int p(x_2|u_2, x_1) Bel(x_1) dx_1 dx_2 \end{aligned}$$

Plugging the numbers in we get the following table:

x	0	1	2	3	4	5
$Bel(x_0)$	1	0	0	0	0	0
$Bel(x_1)$	0.0001	0.9998	6.8798e-24	2.6318e-95	5.5977e-215	0
$Bel(x_2)$	3.9381e-11	0.0762	0.9237	8.3773e-27	3.7763e-59	2.7476e-138
$Bel(x_3)$	2.6757e-26	2.3196e-07	0.2499	0.7500	2.1622e-27	4.3797e-63

We can see that the robot believes that it is $3m$ away from the door and is about 75% certain.

Exercise 2:

Here are several examples of learning algorithms:

- Mean-shift clustering: Unsupervised learning
- Perceptron algorithm: Discriminant function
- Bayes classifier: Generative model
- Conditional Random Field: Discriminative model
- AdaBoost: Discriminant function

For a detailed explanation, please see the textbook *Pattern Recognition and Machine Learning* by *C.M. Bishop* or the slides.

Exercise 3:

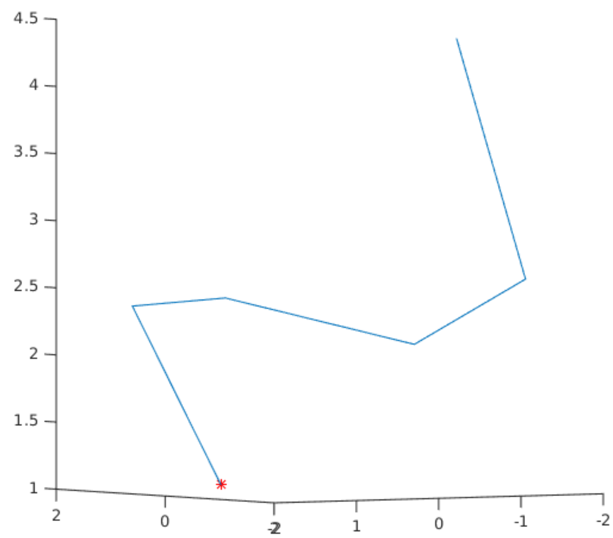

Abbildung 1: Tracker data from quadrocopter. The lines are just an interpolation between the tracked positions (data points).

a) See figure 1.

b) The task is to estimate the speed of the quadrocopter. We do this using polynomial regression. The functions that we learn are dependent on time. We have to find three functions, one for each coordinate (x, y, z). The regression is done with the matrix Φ and vectors $\mathbf{t}_i$:

$$\Phi = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix} \quad \mathbf{t}_x = \begin{pmatrix} 2 \\ 1.08 \\ -0.83 \\ -1.97 \\ -1.31 \\ 0.57 \end{pmatrix} \mathbf{t}_y = \begin{pmatrix} 0 \\ 1.68 \\ 1.82 \\ 0.28 \\ -1.51 \\ -1.91 \end{pmatrix} \mathbf{t}_z = \begin{pmatrix} 1 \\ 2.38 \\ 2.49 \\ 2.15 \\ 2.59 \\ 4.32 \end{pmatrix}$$

The second column of Φ are the timestamps at which the measurements have been taken. In this first case, we assume constant velocity, i.e. we don't have acceleration and the motion equation has only two unkowns w_0 and w_1, i.e. for the case of the x-coordinates we have

$$x(\tau) = w_0 + w_1\tau, \qquad \mathbf{w}_x = (w_0, w_1)^T$$

where $\tau = 0, 1, \dots$ is the time stamp. Thus, Φ has two cloumns.

The pseudoinverse of Φ is

$$\Phi^\dagger = \begin{pmatrix} 0.524 & 0.381 & 0.238 & 0.095 & -0.048 & -0.190 \\ -0.143 & -0.086 & -0.029 & 0.029 & 0.086 & 0.143 \end{pmatrix}$$

With this we compute $\mathbf{w}_i = \Phi^\dagger \mathbf{t}_i$:

$$\mathbf{w}_{x1} = \Phi^\dagger \mathbf{t}_x = \begin{pmatrix} 1.0267 \\ -0.4421 \end{pmatrix} \quad \mathbf{w}_y = \begin{pmatrix} 1.5383 \\ -0.5918 \end{pmatrix} \quad \mathbf{w}_z = \begin{pmatrix} 1.2825 \\ 0.4830 \end{pmatrix}$$

To compute the speed we need $\mathbf{v} = (-0.4421, -0.5918, 0.4830)^T$. The speed is $\|\mathbf{v}\| = 0.8827$.

The residual errors are defined as

$$\begin{aligned} r_x &= \|\Phi \mathbf{w}_x - \mathbf{t}_x\| = 2.8902 && (12) \\ r_y &= \|\Phi \mathbf{w}_y - \mathbf{t}_y\| = 2.4571 && (13) \\ r_z &= \|\Phi \mathbf{w}_z - \mathbf{t}_z\| = 1.2807 && (14) \\ & && (15) \end{aligned}$$

c) Now we have a quadratic motion equation:

$$x(\tau) = w_0 + w_1\tau + w_2\tau^2, \qquad \mathbf{w}_x = (w_0, w_1, w_2)^T,$$

where w_1 is velocity and w_2 is (half the) acceleration. This means we have to estimate 3 function parameters. Thus, the matrix Φ has one more column, i.e.

$$\Phi = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 5 & 25 \end{pmatrix}$$

Again, we compute the pseudoinverse and multiply it with the vectors $\mathbf{t}_i$. We obtain:

$$\mathbf{w}_x = \begin{pmatrix} 2.4739 \\ -2.6128 \\ 0.4341 \end{pmatrix} \quad \mathbf{w}_y = \begin{pmatrix} 0.4573 \\ 1.0297 \\ -0.3243 \end{pmatrix} \quad \mathbf{w}_z = \begin{pmatrix} 1.4656 \\ 0.2084 \\ 0.0549 \end{pmatrix}$$

The residual errors are now

$$\begin{aligned} r_x &= 1.1474 && (16) \\ r_y &= 1.4527 && (17) \\ r_z &= 1.2359 && (18) \\ & && (19) \end{aligned}$$

d) If we want to estimate the position in the next second we can imagine a new row in our Φ matrix $\phi_6 = (1 \quad 6 \quad 36)$. Multiplying this row with our model parameters w for the last model gives us the estimate:

$$t_6' = \phi_6 w = (2.4259 \quad -5.0397 \quad 4.6930)^T \qquad (20)$$

Exercise 4: Programming

See Code.