Robotics Group Project - 5CCS2RGP

Lecture 5: Bayesian Filter for Localisation

Hongbin Liu, King's College London, 2013

Recursive Bayesian Updating

$$P(x \mid y) = \frac{P(y \mid x)\ P(x)}{P(y)} = \frac{\textbf{likelihood} \cdot \textbf{prior}}{\textbf{evidence}}$$

$$\Downarrow$$

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1}) P(x \mid z_1, \ldots, z_{n-1}, u)}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

Z measurements

Recursive Bayesian Updating

$$P(x \mid z_1, \ldots, z_n) = \frac{P(z_n \mid x, z_1, \ldots, z_{n-1}) P(x \mid z_1, \ldots, z_{n-1}, u)}{P(z_n \mid z_1, \ldots, z_{n-1})}$$

Markov assumption: z_n is independent of $z_1, \ldots, z_{n-1}$ if we know x.

$$P(robot = [a,b], t = n)$$

$$= \frac{P(measure = c \mid robot = [a,b]) P(robot = [a,b], action)}{P(measure = c)}$$

Recursive Bayesian Updating

Initialization (assume equal probability at t=0)

$$Bel(robot = [a,b]) = 1 / states\_number$$

Sense and update: (measure, t=k+1)

$$Bel(robot = [a,b]) = \eta P(measure = z \mid robot = [a,b])$$

Move robot (action) and update *Bel*

$$Bel(robot = [a,b])$$
$$= \eta P(measure = z \mid robot = [a,b]) P(robot = [a,b], action)$$
$$= \eta P(measure = z \mid robot = [a,b]) f(action, Bel(robot, t = k))$$

Recursive Bayesian Filter-1D World

0.01	0.1	0.2	0.4	0.1	0.01	0.01	0.01	0.01	0.01

P(Sensor_work) = 0.8 , P(Sensor_wrong)=0.2

P(x=i, measure)

if measure=red

P(x=4, t)= P(Sensor work)*P(x=4,t-1)

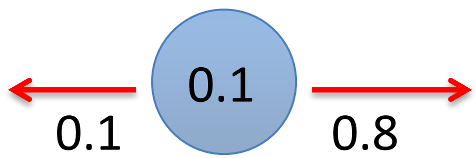
if measure=blue

P(x=4, t)= P(Sensor wrong)*P(x=4,t-1)

Recursive Bayesian Filter-1D World

0.01	0.1	0.2	0.4	0.1	0.01	0.01	0.01	0.01	0.01

P(x=i, U)



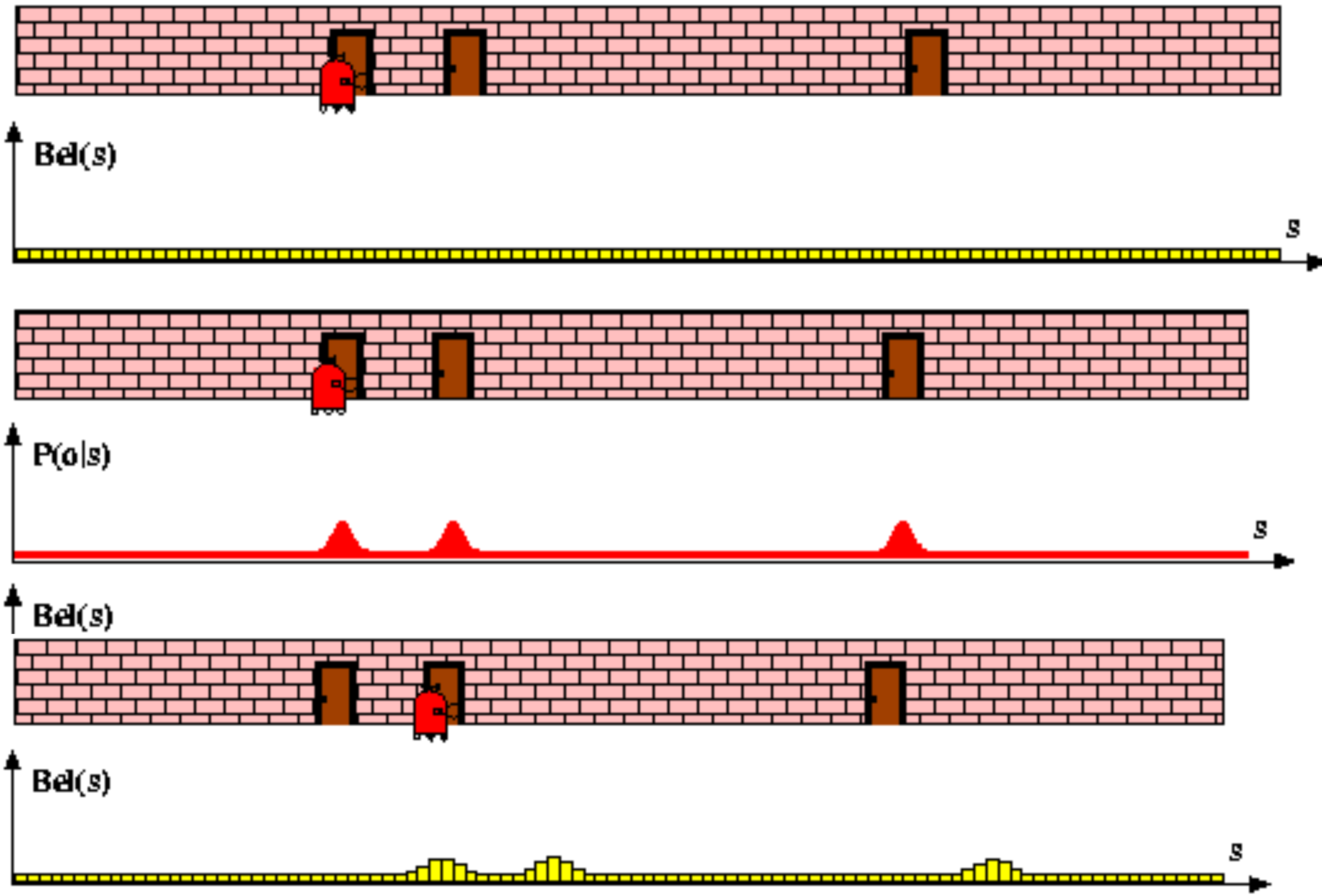
U= Move right 1 grid

$$P([a,b] \mid action) = P(x \mid u) = \sum P(x \mid u, x')P(x')$$

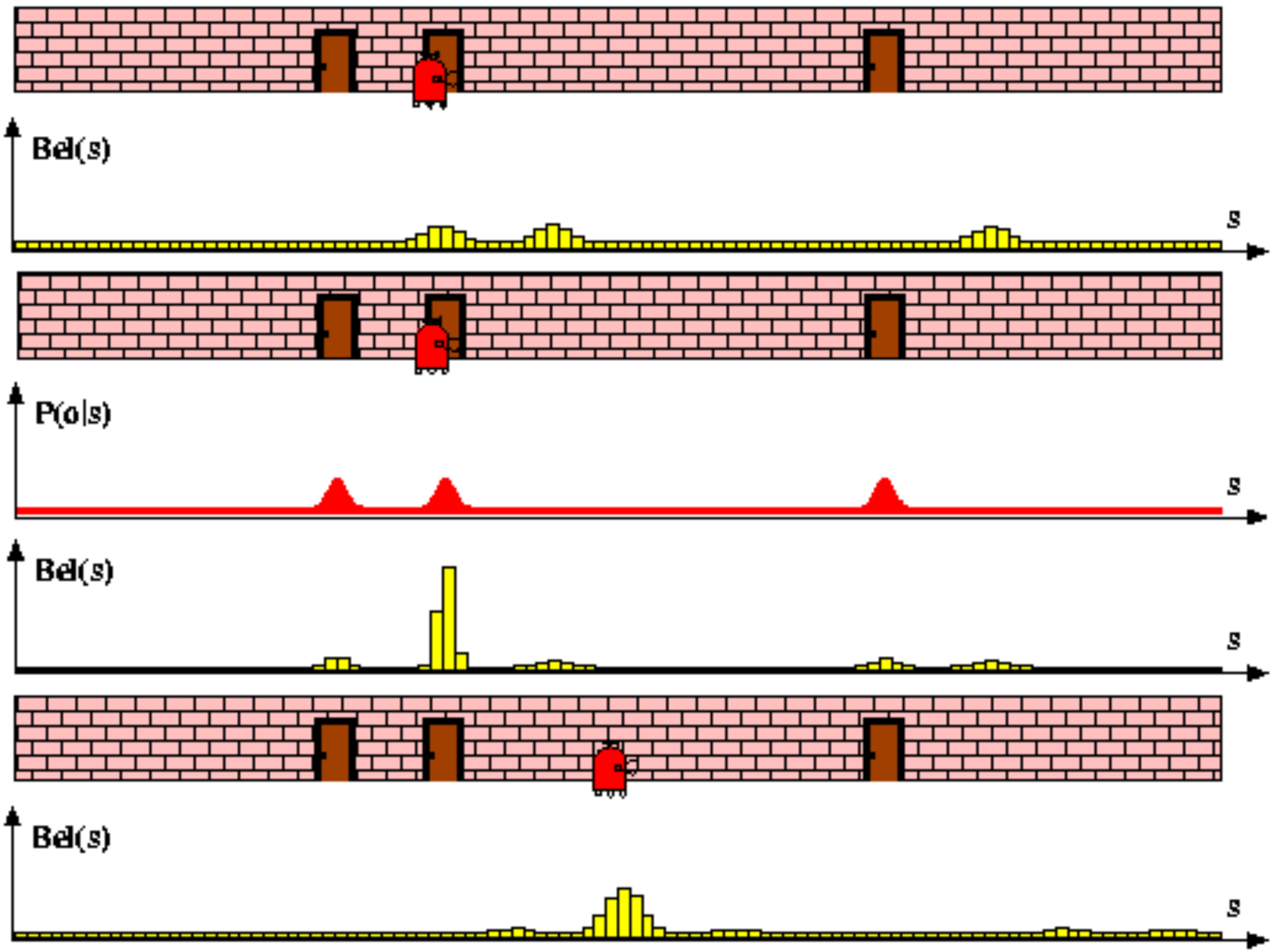
P(x=4, U)= P(x=3,t-1)*0.8+P(x=5,t-1)*0.1+P(x=4,t-1)*0.1

- P(x=i,t)= P(x=i, measure)
- η=1/Sum(p(x=i))
- P(x=i,t)=ηP(x=i, measure)*P(x=i, U)

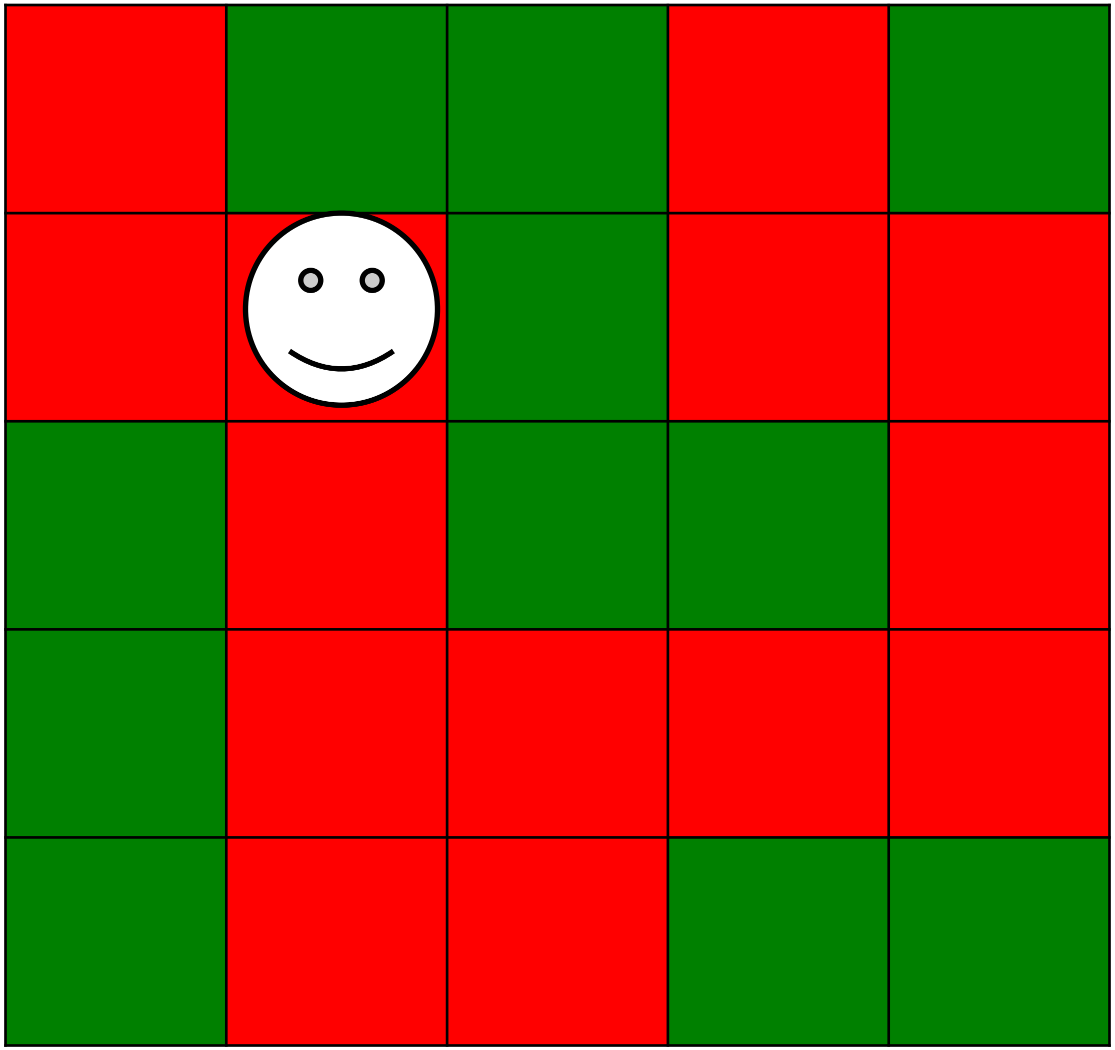
Recursive Bayesian Filter-1D Case



Recursive Bayesian Filter-1D Case



2D Case-Simple Grid World



P(Sensor_work) = 0.7
P(Move_work)=0.8

Action:	Measures:
stay	red
left	green
down	green
left	green
down	red
right	red
......	......

Recursive Bayesian Updating

	b1	b2	b3	b4
a1	0.01	0.1	0.01	0.01
a2	0.2	0.3	0.01	0.01
a3	0.01	0.05	0.01	0.01
a4	0.01	0.01	0.01	0.01

P (X=[a,b], measure)

If measure == map

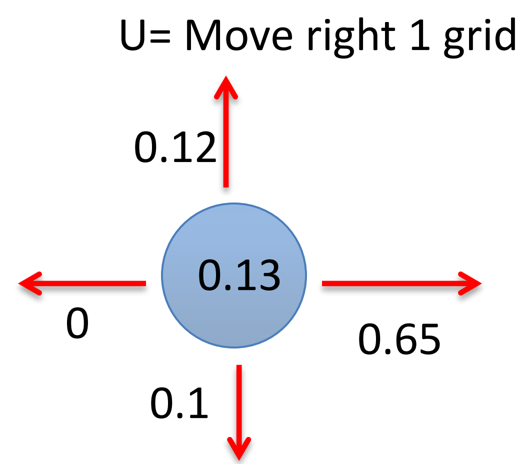
P(x=a2,b2, measure)= P(sensor work)*P(a2, b2)

If measure \= map

P(x=a2,b2, measure)= (1-P(sensor work))*P(a2, b2)

Recursive Bayesian Updating

	b1	b2	b3	b4
a1	0.01	0.1	0.01	0.01
a2	0.2	0.3	0.01	0.01
a3	0.01	0.05	0.01	0.01
a4	0.01	0.01	0.01	0.01

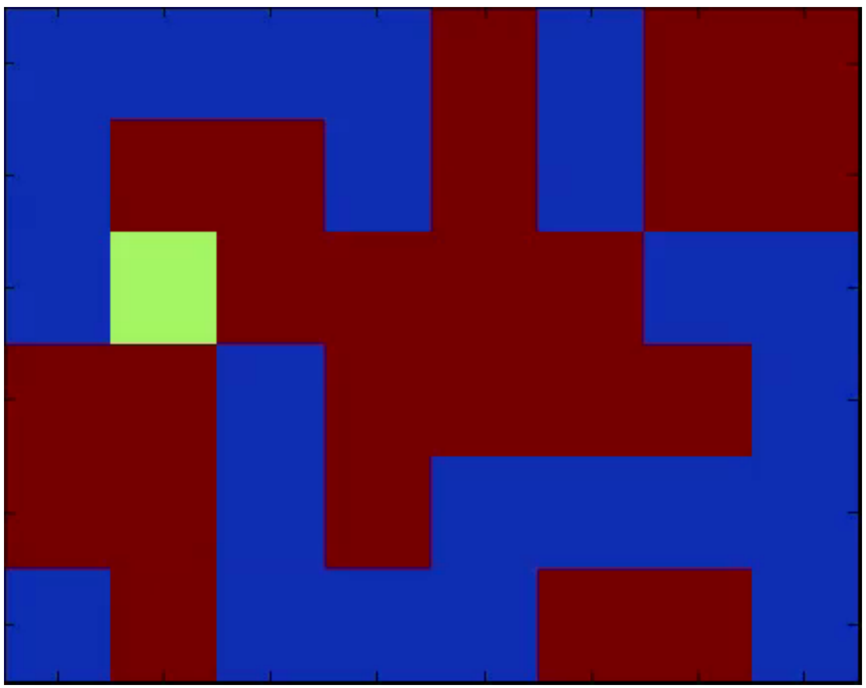


$$P([a,b]\,|\,action) = P(x\,|\,u) = \sum P(x\,|\,u,x')P(x')$$

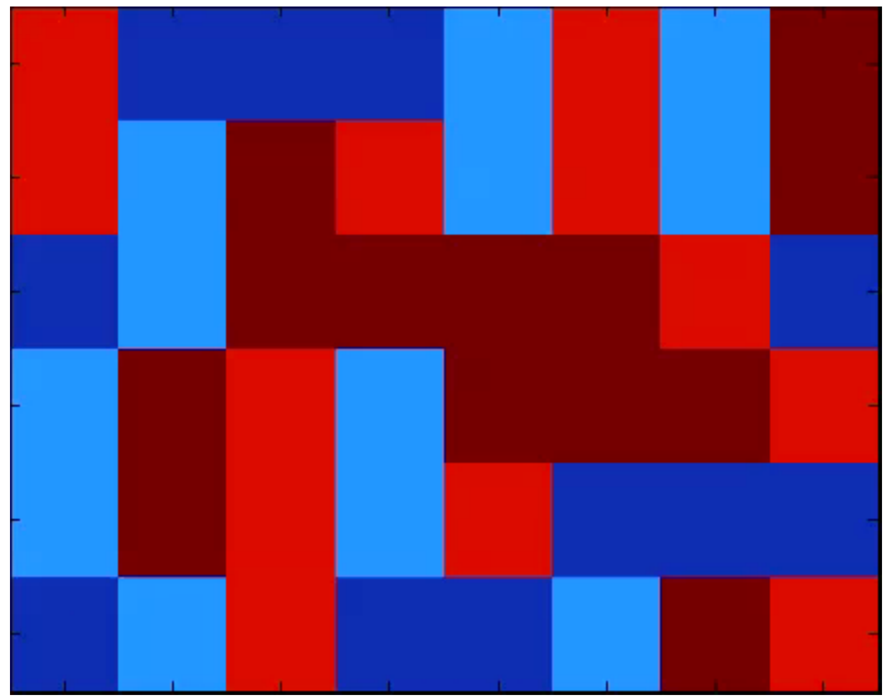
P(x=a2,b2, U)= P(a2,b1)*0.65+P(a1,b2)*0.1+P(a3,b2)*0.12+P(a2,b2)*0.13

Recursive Bayesian Filter

True Robot Path



Estimated Path using Bayesian Filter



2D Case-Simple Grid World

P(Sensor_work) = 0.7 , P(Sensor_wrong)=0.3

Initial condition: Assume equal probability for all grids

Sense and update P: (measure, t=k)

Recursive Bayesian Updating

```
for all grids
    if measure= grid [i,j]
        P([i,j], t=k) =P([i,j], t=k-1)*P(Sensor_work)
    if measure ≠ grid [i,j]
        P([i,j], t=k) =P([i,j], t=k-1)*P(Sensor_wrong)
end

P(i,j)= P(i,j)/ [Sum P(i,j)]
```

MOVE robot (u,v) and update P

P[i, j]=P(Move_work)*P[i-u, j-v]+P(Move_fail)*P[i, j]

Bayes Filters

z = observation
u = action
x = state

$$\boxed{Bel(x_t|z_t)} = P(x_t \mid u_1, z_1 \ldots, u_t, z_t)$$

Bayes $$= \eta P(z_t \mid x_t, u_1, z_1, \ldots, u_t) P(x_t \mid u_1, z_1, \ldots, z_{t-1})$$

Markov $$= \eta P(z_t \mid x_t) P(x_t \mid u_1, z_1, \ldots, z_{t-1})$$

Total prob. $$= \eta P(z_t \mid x_t) \int P(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) P(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1}) dx_{t-1}$$

Markov $$= \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) P(x_{t-1} \mid z_{t-1}) dx_{t-1}$$

$$\boxed{= \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}}$$

$$Bel(x_t) = \eta \; P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) \; Bel(x_{t-1}) \; dx_{t-1}$$

Algorithm **Bayes_filter**(*Bel(x),d*):

η=0

If *d* is a perceptual data item *z* then

    For all *x* do

$$Bel'(x) = P(z \mid x) Bel(x)$$

$$\eta = \eta + Bel'(x)$$

    For all *x* do

$$Bel'(x) = \eta^{-1} Bel'(x)$$

Else if *d* is an action data item *u* then

    For all *x* do $Bel'(x) = \quad P(x \mid u, x') \; Bel(x') \; dx'$

$discrete: \quad {}_{x}\; P(x \mid u, x') \; Bel(x')$

Return *Bel'(x)*