

RBE 500 Homework #8

Arjan Gupta

Bayes Filter Report

I started my Bayes Filter implementation in a python file called `hw8_bayes_filter.py`. At the top of the file, I added a `VERBOSE` flag that helps me toggle on and off output that is helpful for debugging, but probably superfluous for the grader. I defined a list of states of the door, which are *open* and *closed*. I also defined the list of ‘base’ beliefs, which are given in the slides/textbook as the following.

$$\begin{aligned} \text{bel}(X_0 = \text{open}) &= 0.5 \\ \text{bel}(X_0 = \text{closed}) &= 0.5 \end{aligned}$$

These are the initial beliefs (as denoted by X_0), which means that these values will adjust as the script runs through the iteration cases. In my code, these are simply a python list, followed by two helper functions. One helper function helps us print the current belief values. The other helper function makes sure our states, which are strings, are ‘mapped’ to the belief values, and returns them as such.

```

1 # List of base beliefs for [open, closed]
2 # These are the initial beliefs. As we run our
3 # iterations, these base beliefs change.
4 base_belief_list = [0.5, 0.5]
5
6 # Helper function for printing base beliefs
7 def print_base_beliefs():
8     print(f'bel(open) = {base_belief_list[0]}')
9     print(f'bel(closed) = {base_belief_list[1]}')
10
11 # Helper function to get "base" beliefs
12 def get_base_belief(state) -> float:
13     if state == 'open':
14         return base_belief_list[0]
15     elif state == 'closed':
16         return base_belief_list[1]
17     else:
18         return 0.0

```

Next, I started defining the helper function to get the belief values associated with the sensor, i.e. the measurement beliefs. In the slides, these are given as the following.

$$\begin{aligned} p(Z_t = \text{sense_open} \mid X_t = \text{is_open}) &= 0.6 \\ p(Z_t = \text{sense_closed} \mid X_t = \text{is_open}) &= 0.4 \\ p(Z_t = \text{sense_open} \mid X_t = \text{is_closed}) &= 0.2 \\ p(Z_t = \text{sense_closed} \mid X_t = \text{is_closed}) &= 0.8 \end{aligned}$$

Therefore, my helper function for the sensor is the following.

```

1 # Helper function for retrieving measurement beliefs
2 def get_measurement_belief(sense, true_state) -> float:
3     if true_state == 'open':
4         return 0.6 if sense == 'open' else 0.4
5     elif true_state == 'closed':
6         return 0.2 if sense == 'open' else 0.8
7     else:
8         return 0.0

```

I used the condition on the right side of the probability statements to write the if-else logic branching in the helper function. For the sake of making sure that I only return a valid value from known strings, which are 'open' and 'closed', I returned 0.0 when any other string is retrieved.

My next helper function is used for retrieving values associate beliefs with actions, which are given in the textbook/slides as the following.

$$\begin{aligned}
 p(X_t = \text{is_open} \mid U_t = \text{push}, X_{t-1} = \text{is_open}) &= 1 \\
 p(X_t = \text{is_closed} \mid U_t = \text{push}, X_{t-1} = \text{is_open}) &= 0 \\
 p(X_t = \text{is_open} \mid U_t = \text{push}, X_{t-1} = \text{is_closed}) &= 0.8 \\
 p(X_t = \text{is_closed} \mid U_t = \text{push}, X_{t-1} = \text{is_closed}) &= 0.2 \\
 p(X_t = \text{is_open} \mid U_t = \text{do_nothing}, X_{t-1} = \text{is_open}) &= 1 \\
 p(X_t = \text{is_closed} \mid U_t = \text{do_nothing}, X_{t-1} = \text{is_open}) &= 0 \\
 p(X_t = \text{is_open} \mid U_t = \text{do_nothing}, X_{t-1} = \text{is_closed}) &= 0 \\
 p(X_t = \text{is_closed} \mid U_t = \text{do_nothing}, X_{t-1} = \text{is_closed}) &= 1
 \end{aligned}$$

These are represented in my code as the following.

```

1 # Helper function for retrieving action beliefs
2 def get_action_belief(prediction, action, last_known) -> float:
3     if action == 'open':
4         if last_known == 'open':
5             return 1.0 if prediction == 'open' else 0.0
6         elif last_known == 'closed':
7             return 0.8 if prediction == 'open' else 0.2
8     else:
9         return 0.0
10    elif action == 'do_nothing':
11        if last_known == 'open':
12            return 1.0 if prediction == 'open' else 0.0

```

```
13         elif last_known == 'closed':
14             return 0.0 if prediction == 'open' else 1.0
15         else:
16             return 0.0
17     else:
18         return 0.0
```

```
1  ---- Starting HW8 Bayes Filter ----
2
3  The initial beliefs are:
4  bel(open) = 0.5
5  bel(closed) = 0.5
6  -----
7  Iteration 1:
8  The given action is do.nothing and given measurement is closed
9  The updated beliefs are:
10 bel(open) = 0.3333333333333333
11 bel(closed) = 0.6666666666666666
12 -----
13 Iteration 2:
14 The given action is open and given measurement is closed
15 The updated beliefs are:
16 bel(open) = 0.7647058823529411
17 bel(closed) = 0.23529411764705882
18 -----
19 Iteration 3:
20 The given action is do.nothing and given measurement is closed
21 The updated beliefs are:
22 bel(open) = 0.6190476190476191
23 bel(closed) = 0.38095238095238093
24 -----
25 Iteration 4:
26 The given action is open and given measurement is open
27 The updated beliefs are:
28 bel(open) = 0.9732441471571905
29 bel(closed) = 0.026755852842809368
30 -----
31 Iteration 5:
32 The given action is do.nothing and given measurement is open
33 The updated beliefs are:
34 bel(open) = 0.9909194097616345
35 bel(closed) = 0.009080590238365497
```