State Modeling

Models

- Models enable computation and reasoning by robots that operate in the real world.

State

- Choosing the right representation for state is key to effective reasoning about actions in the world.
- **Representing the World State**

The robot's information about its environment is generally referred to as the *world state*.

The world state depends on the kind of information that is required for the specific tasks to be performed by the robot.

Representing the World State

- High-level, symbolic representations are appropriate when low-level controllers are available to reliably execute specific tasks.
- High-level representations allow the robot to build high-level plans, sometimes called *task plans*, that will be executed using low-level controllers

- High-level, qualitative state descriptions may be useful for task-level planning, because they fail to capture any of the geometric aspects of the environment, they are less useful when the robot begins to actually move in, and interact with, the world.

Representing the Robot’s State

- Robot’s state is merely a description of the robot’s location (and orientation) in its environment, which we will define as the robot’s *configuration*.
- The set of all possible configurations will be called the *configuration space*.
- This information could be a qualitative, high-level description , coordinates for the robot’s position in a grid or continuous position coordinates in the plane, continuous coordinates for a position and orientation in the plane.

Actions

- By executing actions, robots change the state of the world, as well as their own state.
- For robot’s that have continuous configuration or state spaces.

Sensors

- Actions allow the robot to affect the world. Sensors allow the robot to perceive the world.

$$Z=h(x)$$

$h(\cdot)$ maps from the current state to a sensor value

Reasoning

- Reasoning provides the *intelligence* of so-called intelligent robotic systems.
- Reasoning involves manipulating data and models, ultimately reaching conclusions.

Perception

Perception uses sensor data to drive inference about the world.

Sensors are characterized by observation models that map the state of the world to sensor values.

Planning

- Planning is the process of determining which actions to execute in order to effect desired changes in the world.

Task Planning

Task planning is useful when either (a) the robot has a set of basic skills, sometimes called *primitives* or *motion primitives*

Path Planning

Path planning deals with the problem of moving the robot from one position to another

Trajectory Planning

While path planning considers the “shape” of the path from start to goal, trajectory planning also considers the time parameterization of that path

Planning

- Planning is the process of determining which actions to execute in order to effect desired changes in the world.

Task Planning

Task planning is useful when either (a) the robot has a set of basic skills, sometimes called *primitives* or *motion primitives*

Path Planning

Path planning deals with the problem of moving the robot from one position to another

Trajectory Planning

While path planning considers the "shape" of the path from start to goal, trajectory planning also considers the time parameterization of that path

Learning

- By learning from data we can make robots smarter.
- Probability theory and statistics, linear algebra, and optimization are key mathematical tools for modern robotics.
- **Probability Theory and Statistics**

Statisticians study data. Probability theorists understand why that’s a good idea

Probability theory

- *Probability theory* provides a rigorous mathematical framework for modeling and reasoning about uncertainty.
- Uncertain quantities are characterized by *probability distributions* that characterize the likelihood of various possibilities, such as the probability that a piece of trash will be made of glass vs. metal.
- *Conditional probability distributions* describe uncertainty in the relationships among various quantities, such as the actual weight of an object and the weight returned by a sensor (as for the trash sorting robot), or the distance traveled by a robot and the commanded motion

Probability theory

- *expectation* quantifies what we would expect to see *on average* after many trials or many observations. For example, if we command the logistics robot to move forward by one meter and repeat this many times, what would be the average distance moved by the robot at each step.

Bayes Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.
P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.
P(A) is Prior Probability: Probability of hypothesis before observing the evidence.
P(B) is Marginal Probability: Probability of Evidence.
P(A|B) = (P(B|A) * P(A))/ P(B) == P(A ∩ B) / P(B)
|||ly P(B|A) = (P(A|B) * P(B))/ P(A) == P(B ∩ A) / P(A)
if P(A∩B) == P(B∩A).
then P(B|A) * P(A) = P(A|B) * P(B)

- **You are given a deck of cards. You have to find the probability of a card being king if you know that it is a face card.**
- Let A be the event of a given card being a face card.
- Let B be the event of a card being a King.
- Now, if we need to find the probability of a card being king if you know that it is a face card, we need to find the probability P(B/A).

- $P(B/A)=P(A/B)*P(B)/P(A)$
- To find $P(B/A)$, we need to find the following probabilities.
- $P(A)$ i.e. the probability of a card being a face card. As there are 12 face cards out of 52, $P(A)=12/52$.
- $P(B)$ i.e. the probability of a card being a King. As there are 4 Kings, $P(B)=4/52$.
- $P(A/B)$ i.e. the probability of a King being a face card. As all the kings are face cards, $P(A/B)=1$.

- using Bayes theorem, we can easily find the probability of a card being a King if it is a face card.
- $P(B/A)=P(A/B)*P(B)/P(A) = 1*(4/52)/(12/52) = 4/12 = 1/3$

Consider the given Dataset ,Apply Naive Baye's Algorithm and Predict that if a fruit has the following properties then which type of the fruit it is
Fruit = {Yellow , Sweet ,long}

Frequency Table:

Fruit	Yellow	Sweet	Long	Total
Mango	350	450	0	650
Banana	400	300	350	400
Others	50	100	50	150
Total	800	850	400	1200

- P(A|B) = (P(B|A) * P(A))/ P(B)

Mango:

- **P(X | Mango) = P(Ye | Yellow) * P(Sweet | Mango) * P(Long | Mango)**

a)P(Yellow | Mango) = (P(Mango | Yellow) * P(Yellow))/ P (Mango)

= ((350/800) * (800/1200)) / (650/1200)

P(Yellow | Mango)= 0.53 →1

b)P(Sweet | Mango) = (P(Sweet | Mango) * P(Sweet))/ P (Mango)

= ((450/850) * (850/1200)) / (650/1200)

- *P(Sweet | Mango)= 0.69 → 2*

c)P(Long | Mango) = (P(Long | Mango) * P(Long))/ P (Mango)

= ((0/650) * (400/1200)) / (800/1200)

P(Long | Mango)= 0 → 3

- On multiplying eq 1,2,3 ==> P(X | Mango) = 0.53 * 0.69 * 0
- **P(X | Mango) = 0**

- 2. Banana:
- P(X | Banana) = P(Yellow | Banana) * P(Sweet | Banana) * P(Long | Banana)

a) P(Yellow | Banana) = (P(Banana | Yellow) * P(Yellow))/ P (Banana)

- = ((400/800) * (800/1200)) / (400/1200)
- *P(Yellow | Banana) = 1 → 4*

b) P(Sweet | Banana) = (P(Banana | Sweet) * P(Sweet))/ P (Banana)

- = ((300/850) * (850/1200)) / (400/1200)
- *P(Sweet | Banana) = .75→ 5*

c)P(Long | Banana) = (P(Banana | Yellow) * P(Long))/ P (Banana)

- = ((350/400) * (400/1200)) / (400/1200)
- *P(Yellow | Banana) = 0.875 → 6*
- On multiplying eq 4,5,6 ==> P(X | Banana) = 1 * .75 * 0.875
- **P(X | Banana) = 0.6562**

P(X | Others) = P(Yellow | Others) * P(Sweet | Others) * P(Long | Others)

a) P(Yellow | Others) = (P(Others| Yellow) * P(Yellow))/ P (Others)

- = ((50/800) * (800/1200)) / (150/1200)
- *P(Yellow | Others) = 0.34→ 7*

b) P(Sweet | Others) = (P(Others| Sweet) * P(Sweet))/ P (Others)

- = ((100/850) * (850/1200)) / (150/1200)
- *P(Sweet | Others) = 0.67 → 8*

c) P(Long | Others) = (P(Others| Long) * P(Long))/ P (Others)

- = ((50/400) * (400/1200)) / (150/1200)
- *P(Long | Others) = 0.34 → 9*
- On multiplying eq 7,8,9 ==> P(X | Others) = 0.34 * 0.67* 0.34
- **P(X | Others) = 0.07742**

Linear Algebra

- Linear algebra is a powerful mathematical tool, even in a nonlinear world.
- Optimization
- Why settle, if you can have the best?

A Trash Sorting Robot (ref:https://www.roboticsbook.org/intro.html)

- A simple robot that sorts trash into appropriate bins
- The robot is presented with items of trash, one at a time, on a conveyor belt.
- Depending on the *category of the item* (state), e.g., paper, metal, etc., the robot's task is to *move the item* (actions) into the appropriate bin.
- The robot can *refine its model of the world based on data* acquired during operation (learning), and this new model can be used to improve operation over time.

Modeling the World State

- The physical properties of a piece of trash comprise all of the information needed by the robot.
- The world state explicitly in terms of the category of the current item of trash.
- cardboard
- paper
- cans
- scrap metal
- bottle

Using Probability to Model Uncertainty

- Probability theory provides a rigorous methodology for reasoning about uncertainty.
- The starting point for reasoning with uncertainty is to define the set of outcomes that might occur
- This set of all possible outcomes is called the sample space, often denoted by Ω

Probability Distributions

- Ω = {cardboard, paper, cans,scrap metal, bottle}
- Subsets of Ω are called events .
- Prior Probability Distributions
- A *prior* that describes our beliefs before any sensor data is obtained.
- cardboard: 200
- paper: 300
- cans: 250
- scrap metal: 200
- bottle: 50

Probability Distributions

Category (C)	*P(C)*
cardboard	0.20
paper	0.30
can	0.25
scrap metal	0.20
bottle	0.05

Cumulative Distribution Function

k	x_k	$F_X(x_k)$
0	cardboard	0.20
1	paper	0.50
2	can	0.75
3	scrap metal	0.95
4	bottle	1.00

Modeling Actions and Their Effects

We assign labels to these actions as follows:

- a_1: put in glass bin
- a_2: put in metal bin
- a_3: put in mixed paper bin
- a_4: nop (let the object continue, unsorted)

Modeling Actions and Their Effects

	cardboard	paper	can	scrap metal	bottle
glass bin	2	2	4	6	0
metal bin	1	1	0	0	2
paper bin	0	0	5	10	3
nop	1	1	1	1	1

Expectation

Suppose that the random variable X takes its values from a finite set, $X \in \{x_1, \ldots, x_n\}$. The **expected value** of X, which we denote by $E[X]$ is defined by

$$E[X] = \sum_{i=1}^{n} x_i p_X(x_i)$$

For the example above, the expected value of cost, $E[X]$, is given by

$$E[X] = (0 \times 0.2) + (0 \times 0.3) + (5 \times 0.25) + (10 \times 0.2) + (3 \times 0.05) = 3.4$$

The concept of **expectation** is average behavior over many trials.

Sensors for Sorting Trash

- a conductivity sensor, outputting the value True or False
- a camera with a three detection algorithms: bottle, cardboard, paper
- a scale, which gives a continuous value in kg.

Conductivity sensor

$$P(conductive|trash\ category)$$

$$P(conductive = True|trash\ category = can) = 0.9$$

$$P(conductive = False|trash\ category = can) = 0.1$$

Category	false	true
cardboard	0.99	0.01
paper	0.99	0.01
can	0.1	0.9
scrap metal	0.15	0.85
bottle	0.95	0.05

Multi-Valued Sensors

- Binary sensors are easily generalized to multi-valued sensors.
- For our running example, assume that there is a camera mounted in the work cell, looking down on the trash conveyor belt.
- The camera is connected to a computer which runs a vision algorithm that can output three possible detected classes:
- bottle , cardboard , and paper. We can model this sensor using the conditional probability distribution

Multi-Valued Sensors

$$P(detection|trash\ category)$$

Category	bottle	cardboard	paper
cardboard	0.02	0.88	0.1
paper	0.02	0.2	0.78
can	0.33	0.33	0.34
scrap metal	0.33	0.33	0.34
bottle	0.95	0.02	0.03

Continuous Valued Sensors

Trash sorting system, it stands to reason that the weight of an object is a great indicator of what category it might belong to.
how should we treat *continuous* measurements?
We could use a very finely quantized histogram on some discretized weight scale, allowing us to use the Discrete Conditional machinery from above, but we can do much better by explicitly representing weight as a continuous quantity. (PDF- Probability Density function)

$$p(Weight|Category)$$

Perception

- Perception is the process of turning sensor measurements in actionable information.
- Maximum likelihood estimation outputs the state that “agrees” most with the measurement.
- An unknown object enters the sorting area, and the weight sensor outputs *50 grams*. What state is the most *likely*, given this measurement?

Perception

- Perception is the process of turning sensor measurements in actionable information.
- One idea is to simply check the conditional probability value for each of the possible categories

```
cardboard   :0.00044
paper       :0.00000
can         :0.00000
scrap metal :0.00242
bottle      :0.00091
```

Perception

- Maximum Likelihood

Maximum likelihood estimation outputs the state that “agrees” most with the measurement.

$$x^*_{ML} \doteq \arg\max_x p(Z = z | X = x).$$

In other words, we vary the *state* X in the conditional probability while keeping the measurement Z fixed, with known value z, and then select the X that maximizes the resulting number.

The Likelihood Function

- Likelihoods are not probabilities.

It is convenient to define the **likelihood** of a state X given the measurement z as any function $L(X;z)$ that is proportional to the conditional density $p(z|X)$:

$$L(X;z) \propto p(z|X).$$

The notation emphasizes that the measurement z is given, and that the likelihood is a function defined on the state X. This is something many people get confused about. It is the *state* that is likely or unlikely, not the measurement.

Because of the definition above, the maximum likelihood estimate x^*_{ML} is equivalently obtained as

$$x^*_{ML} \doteq \arg\max_{x} L(X=x;z).$$

Note that the value x^* that maximizes the value of a function $f(x)$ also maximizes the value of the function $\alpha f(x)$ for any constant $\alpha > 0$, and therefore maximizing $p(z|X)$ is equivalent to maximizing $L(X=x;z)$. This is a theme we will encounter quite often: when maximizing (or minimizing) a quantity, we will often not be so concerned with the proportionality constants, which can be expensive to compute.

Decision Theory

- Decision theory is about turning information into action.
- Naive Decision Making Using Priors
  - A robot without sensors can make naive decisions using prior information.
- Optimizing for the Worst Case

  A conservative approach to accounting for costs is to minimize the damage that can occur in the worst case scenario.

Decision Theory

One way to account for the costs of actions would be to apply an action that minimizes the worst-case cost. This provides a quantitative upper bound on how badly things could go. Denote by $\mathbf{cost}(a_i, c)$ the cost of applying action a_i when the piece of trash in the work cell is from category c. We then write our decision rule as a minimization problem

$$a^* = \arg\min_{a_i} \max_{c \in \Omega} \mathrm{cost}(a_i, c).$$

From the table of costs given in previous sections, we see that this approach leads to always executing the *nop* action, since the worst-case cost for this action is 1, while the worst-case costs for the other three actions are 6, 2, and 10. This approach, however, merely reduces our trash sorting system to a conveyor belt that allows all items of trash to pass through, unsorted. In this case, the conservative action is to take no action, and the robot becomes a motionless spectator.

Decision Theory

- Minimizing Expected Cost
  - if the system will operate over a long period of time, we can optimize the long-term average cost.

Decision Theory

The idea of expected cost is this: what do we expect to be the average cost of performing an action many times. The expected value for the cost of applying action a is merely the weighted average of the costs $cost(a, c)$, where the weights are exactly the prior probabilities assigned to the categories, c:

$$E[cost(a, C)] = \sum_{c \in \Omega} cost(a, c)P(c)$$

In the equation above for expectation, the notation $E[cost(a, C)]$ denotes the expected cost for executing the action a with the expectation being taken with respect to the randomly occurring trash category C. We use upper case C to indicate that the category is a random quantity, and that the expectation should be computed with respect to the probability distribution on categories (i.e., the priors given in the previous section).

We can now formulate our decision process as the following minimization problem

$$a^* = \arg\min_{a_i} E[cost(a_i, C)]$$

Learning

- We can learn prior and sensor models from data we collect.
- Estimating a Discrete PMF

  Count the occurrences for each category.
- Smoothing

  We make up fake data to deal with sparse data sets.
- Modeling a Sensor from Data

  When learning a conditional distribution, we need to separate out the counts based on the conditioning variable.
- Fitting Gaussian

  Gaussian, we need to only estimate its mean μ and its a variance σ2, and we used techniques from statistics to estimate these

Reference

- https://www.roboticsbook.org/intro.html