Attribute Layer

At this point we know something about the lowest levels of abstraction: sensor signals and information, kinematics and motor signals. While this in itself is enough to get some simple things done, experience has shown that sophisticated systems require disciplined techniques to solve difficult problems. For example, programming a mobile robot to follow a wall using its proximity sensors can be easily programed. Programming a mobile robot to roam the halls of the building, looking for trash which it picks and brings to the garbage. In doing this, it must navigate both the hallways as well as unexpected obstacles such as people. Finally, it must identify all of the people it sees and call for help if it sees a person it doesn't recognize. This sort of complicated robotics task requires processing sensor information and generating output at higher and higher levels of abstraction. Today we'll start looking at these higher levels.

	Signal	Information	Attribute	Model	Lifetime
Input Channel	Sensor	Binary	Detection	Maps and Logic	Agent Modeling
Output Channel	Motor	Kinematics	Action Selection	Planning	Goal Selection

The attribute layer is made up of Detection in the input channel and Action Selection in the output layer.

Detection

1. In the input channel of the attribute layer, "detection" takes place. What we mean by this is that the streams of input from the various sensors are combined to detect attributes of the environment.

2. For a simple example,

   (a) suppose a robot is navigating through an area filled with boxes to find a box with a light on it. The root needs to avoid colliding with the obstacle boxes, but must run in to the target box with the light.

   (b) If the robot uses just its proximity sensors (IR, Sonar) it won't be able to distinguish an obstacle from the target.

(c) If the robot just uses its light sensors, it would be unable to distinguish the target box from a light reflection on the floor. At this stage, you should already be familiar with how a simple thing like a light reflection can interfere with the behavior of you robot.

3. Lets look at a more "real world" example.

   (a) It is currenlty popular to design robot systems that interact with people milling around as if at a cocktail party. These robotic butlers need to bring the hors d'oeuvres to guests at the party. The robot is equiped with short range sonars for obstacle avoidance, and a video camera for face recognition, angled up to see the guests faces. What to we do to keep the butler from serving the stuffed mushrooms to the portrait of great-uncle Reginald in the hall?

   (b) Configure sonar to look for "legs"– obstacle in front, but no obstacles front left and front right.

   (c) Note that sonar distance + camera angle can give us hight. If, as with most portraits, Reggie is hung so his head would be ten feet tall, then maybe we won't serve him.

   (d) Maybe we could also add a body heat sensor, so that combined with the other two, we can detect our guests.

   (e) *Note:* All these techniques will sometimes fail. We'll come back to that.

4. The basis of detection is sensor fusion, i.e., we fuse the information from multiple sensors into one identification. We will see more sensor fusion at higher levels as well.

5. Up to now, we have concentrated on detecting the presence of objects in the environment, such as our party guests. Sometimes we want to detect *more* than just their presence, but also properties of the objects: is the person sitting or standanding? Is the person in the hall or in the doorway? We call this detecting an attribute of the world state. The principles for detecting objects or states are the same, and we will often use the phrase "detect state" to mean detecting objects or world state.

6. The simplest way to perform sensor fusion to detect state is the ad-hoc collection of rules just like we described in our butler example above. This is still an extremely common method of performing sensor fusion, though it has serious problems.[1]

7. The most popular modern technique it to infer a program directly from data, using a mathematical method such as conditional probability.

Basic Probability

- P(H=a) is the probability of some variable H having value a.
- e.g. $P(CoinFlip = heads)$, $P(CellOccupied = true)$ This is called a hypothesis.

[1] Is "ad hoc" a dirty word? Merriam Webster defines ad hoc as:
"adjective 1 a : concerned with a particular end or purpose ⟨an ad hoc investigating committee⟩ b : formed or used for specific or immediate problems or needs ⟨ad hoc solutions⟩" In Artificial Intelligence and Robotics, it has come to mean a program solution that was developed by hand to solve that particular instance of the problem. For instance, while our people identifying solution might work for cocktail parties, it might turn out to be terrible for forests. In the past it has often been the case that ad hoc solutions worked poorly in all situations other than precisely what they were designed for. Because of this, in some circles to describe a solution as "ad hoc" is an insult. As with all similar issues, the truth is much more complicated...

- The hypothesis has some number p, $0 \leq p \leq 1$: $P(CoinFlip = heads) = 0.5$, $P(CellOccupied = true) = 0.33$.
- The sum of the p's across all values of the variable must be 1.: $P(CoinFlip = heads) = 0.5$ and $P(CoinFlip = tails) = 0.5$ or $P(CellOccupied = true) = 0.33 + P(CellOccupied = false) = 0.67$.
- If the values are true or false, we often use a shorthand: $P(CellOccupied = true)$ becomes $P(CellOccupied)$ and $P(CellOccupied = false)$ becomes $P(\neg CellOccupied)$
- the set of p's for all values that H can be is called the probability distribution.
- Sometimes, when we say $P(A)$ we mean not $P(A = true)$ but we mean the whole probability distribution: $P(A = true) \wedge P(A = false)$ or $P(A = 1) \wedge P(A = 2) \wedge P(A = 3)$.
- be careful of that.
- $P(A \wedge B) = P(A)P(B)$
- $P(A \vee B) = P(A) + P(B)$
- $P(A) + P(\neg A) = 1$. Some of what I said above applies only to cases where two variables are are independent.
  - Variables are independent when the value of one does not affect the value of another.
  - When flipping 2 different coins, the value of one does not affect the value of the other.
  - But lets take a different example:
    * Imagine we live in a town where it rains 50% of the time.
    * $P(Raining) = 0.5$
    * Most of the time it rains, the sidewalk gets wet. Sometimes it doesn't because the tent for the local art fair keeps it dry.
    * $P(Wet) = 0.49$
    * If we use the rule $P(A \wedge B) = P(A)P(B)$, then the $P(Raining \wedge Wet) = 0.245$.
    * That says that $\frac{1}{4}$ of the time, it will be raining and the sidewalks will be wet.
    * But does that make sense? You expect that $\frac{1}{2}$ of the time it will be raining, and nearly 100% of the time it is raining, the sidewalks will be wet.
    * It doesn't make sense, so instead we use a Rule Based on Conditional Probability.
  - Conditional Probability is a way of expressing the relationship between two variables, expressed as: $P(A = x|B = y)$ The probability of A being x, given that B is y, or more succinctly: $P(A|B)$, the probability of A given B.
  - Properties of conditional probability:
    * $P(A|B) = P(A)$ if A and B are independent.
    * $P(A \wedge B) = P(A)P(B|A)$ Note that the independent case derives from these two.
    * $P(Raining \wedge Wet) = P(Raining)P(Wet|Raining)$
    * $P(Raining \wedge Wet) = 0.5 \times 0.99 = 0.495$

$$\begin{aligned} P(A \wedge B \wedge C) = & P(ABC) \text{ Just notationally more compact.} \\ & P(AB)P(C|AB) \\ & P(C|AB)P(B|A)P(A) \end{aligned}$$

- If $P(A \wedge B) = P(A)P(B|A)$, then $P(B \wedge A) = P(B)P(A|B)$
- $P(A \wedge B) = P(B \wedge A)$
- $\therefore P(A)P(B|A) = P(B)P(A|B)$
- If we solve for $P(A|B)$ we get: $P(A|B) = \frac{P(A)P(B|A)}{P(B)}$
- If that is true, then $P(\neg A|B) = \frac{P(\neg A)P(B|\neg A)}{P(B)}$ too. These are known as Bayes' Rule.

Probability and Detection

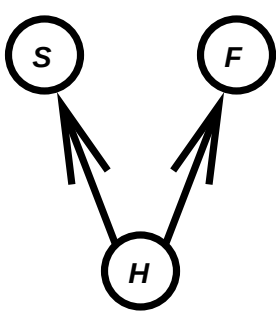
- So how do we apply this? Well, we want to calculate the probability that there is a person in front of us, given that we know what the values of our sensors are. That is, if S_1CLOSE is the proposition that sonar 1 says there is an object nearby, and $(F > .8)$ is the proposition that the face recognition system returned a score of 0.8, then we want to calculate something like: $P(HUMAN = true|S_1CLOSE = false, S_2CLOSE = true, S_3CLOSE = false, (F > .8) = true)$
- How many values can the conditional part take? 16.
- We can build a a table describing each possible value set:

S_1CLOSE	S_2CLOSE	S_3CLOSE	$(F > .8)$	$P(HUMAN = true)$
true	true	true	true	0.6
true	true	true	false	0.1
true	true	false	true	0.6
true	true	false	false	0.1
true	false	true	true	0.4
true	false	true	false	0.1
true	false	false	true	0.5
true	false	false	false	0.1
false	true	true	true	0.6
false	true	true	false	0.1
false	true	false	true	0.8
false	true	false	false	0.3
false	false	true	true	0.5
false	false	true	false	0.1
false	false	false	true	0.5
false	false	false	false	0.0

- Where do these numbers come from?
  - The straightforward way is to measure them.
  - There are 2^4 or 16 cases, and each case is a row in the table. and maybe we want to get about 100 samples for each case. So we'll gather 2000 samples.
  - A sample is just a test of the robot. We look at the values for the sensors and whether or not there is a person there. If 98 times out of 100, when $S_1CLOSE = false, S_2CLOSE = true, S_3CLOSE = false, (F > .8) = true$, there is a person standing there, then $P(HUMAN = true|S_1CLOSE = false, S_2CLOSE = true, S_3CLOSE = false, (F > .8) = true) = 0.98$.

  - It's important to note that just because we get 2000 samples does not mean we'll get 100 for each row unless we're incredibly careful about how we set up the data gathering, because it's hard to force the sensors to have particular readings.

- If it takes us 1 minute to gather each sample (something of an overstatement), and we need 2000 samples, then it would take 2 days 9 hours to gather the data, given the standard graduate student workload of 12 hours per day.
- But we have a problem: what is "close" defined as? Where did the value 0.8 come from? these things smack of ad hoc-ness, something we were explicitly trying to avoid. yuk.
- What we can do is move to a more general formulation, where we use the readings from the sensors directly: $P(HUMAN = true|S_1 = d_1, S_2 = d_2, S_3 = d_3, F = s)$, where the ds and s are values from the sensor.
- OK, lets assume the sonar returns a value between 0 and 100, and the face sensor returns a number between 0 and 1. If we just look at 2 digit values of the face sensor, each sensor has 100 possible values. This gives us 100^4 or 10^8 possible rows for the table. If we still want around 100 values for each, we need to gather at least 10^{10} samples. Even if we work our poor graduate students 24 hours a day, it will still take almost 20,000 years! Ouch
- So what we'll do is try to reduce this number through our knowledge of probability.
- First, we note that:$P(A|B, C) = \frac{P(A)P(B,C|A)}{P(B,C)}$ This is just Bayes Rule again, but we treat B, C as a single unit.
- Thus, if we can say: $P(H|S_1, S_2, S_3, F) = \frac{P(S_1,S_2,S_3,F|H)P(H)}{P(S_1,S_2,S_3,F)}$
- Doesn't seem to have helped, but be patient.
- $P(S_1, S_2, S_3, F|H) = \frac{P(S_1,S_2,S_3,F,H)}{P(H)}$ by definition of conditional probability.
- $P(S_1, S_2, S_3, F, H) = P(S_1)P(S_2)P(S_3)P(F)P(H)$ if they were all independent.
- That would be good, because instead of one really **big** table, we'd need 5 small tables (100,100,100,100,2) of a total of 502 rows, requiring a measly 50,200 samples.
- But these things are not independent. For instance, the sensor readings are directly dependent on whether or not there is a human there. duh.
- So now we'll introduce a new idea: conditional independence. Two bits of knowledge are conditionally independent given a third: $I(A, B|C)$
- Lets look a a graph that describes this situation: Vertices are variables, and edges are contitional tables:



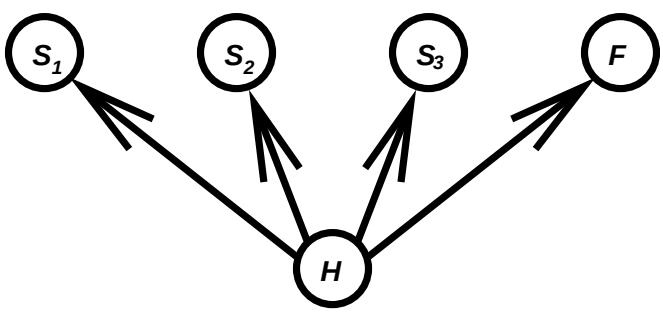
In the picture, the sonar sensor reading and the Face recognizer are both dependent on whether or not there is a human there.

- If we know that $F = .9$, we think its likely there's a face there, and thus also likely that S_2 will be small. The values of F and S_2 depend on each other. But if we know that there's a person there, then S_2 will only return a large number if something extraordinary happens, like interference or reflection. If we further find out that $F = .9$ does that change what we think S_2 will be? No because we *already* know there's a person there, recognizing the face of the person doesn't tell us anything new. Know the presence of the person separates S_2 from F.
- Conversely, if we know a human is *not* there, then finding out there is no reading of a face doesn't really tell us any more about what the sensor reading will say.
- Because of that interpretation, some people like to define conditional independence as $I(A, B|C) \leftrightarrow P(A|B, C) = P(A|C)$ i.e. once we know C, adding or taking away knowledge of B doesn't change anything. from this definition can be derived an interesting item:

$$\begin{aligned}
P(A|B,C) =& P(A|C) \text{ our definition} \\
P(A|B,C) =& P(A|B) \text{ our definition} \\
P(A,B,C) =& P(A|B,C)P(B,C) \text{ Bayes Rule} \\
P(A,B,C) =& P(A,B|C)P(C) \\
\therefore P(A,B|C)P(C) =& P(A|B,C)P(B,C) \\
P(A,B|C) =& \frac{P(A|B,C)P(B,C)}{P(C)} \\
& \frac{P(A|C)P(B,C)}{P(C)} \text{ substitution} \\
& \frac{P(A|C)P(B|C)P(C)}{P(C)} \text{ conjunction} \\
P(A,B|C) =& P(A|C)P(B|C)
\end{aligned}$$

By analogy, this is similar to regular independence. With regular independence, $P(A, B) = P(A)P(B)$. Conditional independence: $P(A, B|C) = P(A|C)P(B|C)$. That is, while A and B are not normally independent, when we know C they are.

- Now that we have this definition of conditional independence, can we use it? Let's look at the DAG below.

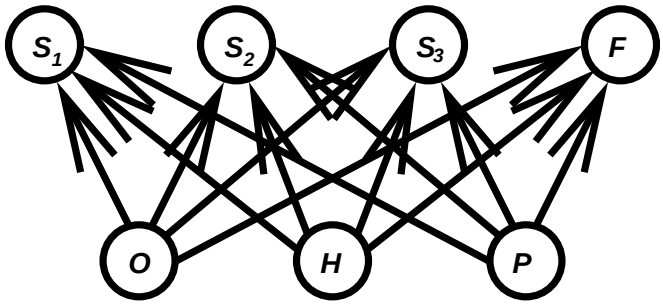


In this model, the sensor readings are all caused by (and only by) the presence or absence of a human. Thus if we know that a human is there, all the sensors become independent. That

would mean that:

$$P(H|S_1, S_2, S_3) = P(H|S_3)P(H|S_3)P(H|S_3)$$

- Is this assumption valid? No. This DAG more accurately describes the situation.



- But, let's naively assume that the simple model is true. How would that help us? Using the general form of conditional independence, $I(V_1, V_2, ...V_n|W) \rightarrow P(V_1, V_2, ...V_n|W) = P(V_1|W)P(V_2|W)...P(V_n|W)$ we know:
  $P(H|S_1, S_2, S_3, F) = \frac{P(S_1|H)P(S_2|H)P(S_3|H)P(F|H)P(H)}{P(S_1,S_2,S_3,F)}$

- Is that a help? Sure! The numberator now becomes 5 tables with 200,200,200,200, and 2 rows, respectively, for a total of 802 rows.
- Now the only thing to deal with is: $P(S_1, S_2, S_3, F)$.
- We know that $P(A|B) + P(\neg A|B) = 1$
- and substituting in $P(A|B) + P(\neg A|B) = 1$, we get:
  $1 = \frac{P(A)P(B|A)}{P(B)} + \frac{P(\neg A)P(B|\neg A)}{P(B)}$.
- Solve for P(B) to get:
  $P(B) = P(A)P(B|A) + P(\neg A)P(B|\neg A)$
- For our purposes:

$$\begin{aligned}
P(H|S_1, S_2, S_3, F) =& \frac{P(S_1, S_2, S_3, F|H)P(H)}{P(S_1, S_2, S_3, F)} \\
P(\neg H|S_1, S_2, S_3, F) =& \frac{P(S_1, S_2, S_3, F|\neg H)P(\neg H)}{P(S_1, S_2, S_3, F)} \\
1 =& P(H|S_1, S_2, S_3, F) + P(\neg H|S_1, S_2, S_3, F) \\
=& \frac{P(S_1, S_2, S_3, F|H)P(H)}{P(S_1, S_2, S_3, F)} + \frac{P(S_1, S_2, S_3, F|\neg H)P(\neg H)}{P(S_1, S_2, S_3, F)} \\
\therefore & P(S_1, S_2, S_3, F) = \\
P(H|S_1, S_2, S_3, F) =& \frac{P(S_1|H)P(S_2|H)P(S_3|H)P(F|H)P(H)}{P(S_1, S_2, S_3, F|H)P(H) + P(S_1, S_2, S_3, F|\neg H)P(\neg H)} \\
=& \frac{P(S_1|H)P(S_2|H)P(S_3|H)P(F|H)P(H)}{P(S_1|H)P(S_2|H)P(S_3|H)P(F|H) + P(S_1|\neg H)P(S_2|\neg H)P(S_3|\neg H)P(F|\neg H)}
\end{aligned}$$

- Now we've fixed the denominator so that it requires only small tables. Notice that in fact, it is the SAME tables as the numerator. Well, the tables with the $\neg H$ are actually different tables, but the can be built from the same data we used to build the other tables.
- So we have 80,000 pieces of data to gather, or roughly 100 days. That's a huge improvement, though still inconvenient. But another interesting thing to note is that since the tables are independent, we can gather the data for all *at the same time*. Thus we really only need to gather 20,000 pieces of data, for 28 days. Now we're talking!

8. Thus, using probability we can come up with a reasonable data-driven solution to the detection problem.