cisc3665
game design
fall 2011
lecture # V.3 non-deterministic behaviors

topics:

- non-deterministic behaviors
- probabilistic behaviors
- Bayesian networks

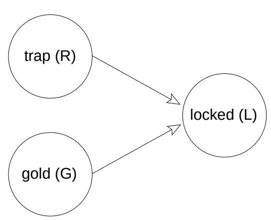
references:

- notes from:
  - *AI for Game Developers*, by David M. Bourg and Glenn Seemann. O'Reilly Media, 2004, chapter 13.

non-deterministic behaviors

- Probability can be used to guide behaviors for non-player characters (NPCs) in non-deterministic environments.
- Probabilistic behaviors are useful when your NPCs don't have perfect knowledge about the world, but they know something about the likelihood that things may be true (or false).
- Here is an example:
  You are playing a game in which there are a number of boxes. The human player can hide gold in any box. The human player can also set a trap in any box. The human player has a limited amount of gold and a limited number of traps. There are a limited number of NPCs in the game environment, who have to find gold, but will die if they fall into traps. The goal of the game is for the human player to outsmart all the NPCs and kill them off (using the traps in the boxes) before they find all the gold.
  How will you code your NPC so that it knows whether to look in a box for gold?

- The NPC has to balance the benefits of finding gold in a box with the risk of finding a trap in the box.
- The NPC has some added information, which is that the box may or may not be locked. Since the human has a limited number of locks, it is reasonable to assume that there is some correspondance between the boxes that are locked and the boxes that have gold.
- The figure below represents the *causal relationship* between a box being locked and a box containing gold or containing a trap.



Meaning: the box could be locked *because* it contains a trap, or the box could be locked *because* it contains gold, or both. In other words, a trap could be the *cause* of the box being locked, or gold could be the *cause* of the box being locked, or both.

- We can represent this information in a table:

if the box contains a trap... (R)	*if the box contains gold...* (G)	*what is the probability that the box is locked/not locked?* (L or $\neg L$)
true	true	$Pr(L\|R \wedge G)$ $Pr(\neg L\|R \wedge G)$
true	false	$Pr(L\|R \wedge \neg G)$ $Pr(\neg L\|R \wedge \neg G)$
false	true	$Pr(L\|\neg R \wedge G)$ $Pr(\neg L\|\neg R \wedge G)$
false	false	$Pr(L\|\neg R \wedge \neg G)$ $Pr(\neg L\|\neg R \wedge \neg G)$

- We want to use this information to reason about whether to open a box or not.
- For example, given a box that is locked:
  - what is the probability (i.e., likelihood) that the box contains gold?
  - what is the probability that it contains a trap?
  - what is the probability that the box contains both?

- A quick word about notation:
  $Pr(...)$ means "the probability of ..."

  a vertical bar $|$ means "given"

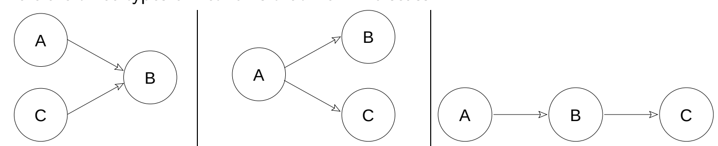
  $Pr(A|B)$ means "the probability of A given B"

  $\neg$ means "not" (the book chapter uses $\sim$)

  $\wedge$ means "and" (the book chapter uses $\cap$)

  $\vee$ means "or" (the book chapter uses $\cup$)

- There are three types of networks that we will discuss:

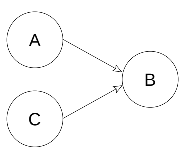


common effect network | common cause network, or Naïve Bayesian network, or Bayesian classifier | causal chain (with three nodes)

- These correspond, respectively, to three types of *inference*:
  - explaining away (common effect network)
  - diagnostic reasoning (common cause network)
  - predictive reasoning (causal chain)

- Let's discuss each type of network in turn.

common effect network



- This *common effect* network says that:
  - $Pr(A)$ effects $Pr(B)$
  - $Pr(C)$ effects $Pr(B)$
- The two probabilities $Pr(A)$ and $Pr(C)$ are *conditionally independent* of each other, as long as B is not known to be true.
- If B is known to be true, then $Pr(A)$ and $Pr(C)$ are *conditionally dependent* on each other.
- The idea that "B is known to be true" means: $Pr(B) = 1$ or 0
- Most of the time, B is not known to be true; so $Pr(A)$ and $Pr(C)$ are independent of

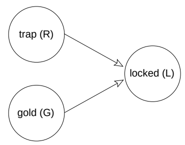
each other, and we can compute the probability of B, i.e., $Pr(B)$ using information about $Pr(A)$ and about $Pr(C)$.

- Given the common effect network above, $Pr(B)$ is computed as follows:

$$\begin{aligned} Pr(B) = \quad & Pr(B|A \wedge C)Pr(A)Pr(C) \quad + \\ & Pr(B|\neg A \wedge C)Pr(\neg A)Pr(C) \quad + \\ & Pr(B|A \wedge \neg C)Pr(A)Pr(\neg C) \quad + \\ & Pr(B|\neg A \wedge \neg C)Pr(\neg A)Pr(\neg C) \end{aligned}$$

- It means that the probability of B depends on both the probability of A and the probability of C. Thus the probability of B is equal to the sum of the probabilities of all combinations A and C and $\neg A$ and $\neg C$.
- Conditions A and C produce a common effect, B.
- This is called "explaining away".

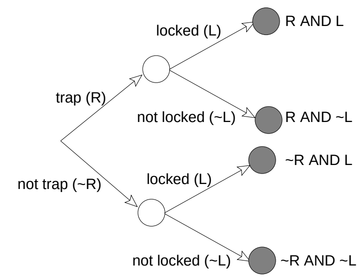
common effect example



- The network above represents the common effect relationship between L (a box being locked) and the box containing a trap (R) and/or the box containing gold (G). A *common effect* of either R or G is that the box is more likely to be locked (L).
- We do not know if L is true, so we can consider the causal relationships $R \rightarrow L$ and $G \rightarrow L$ independently.
- Let's derive the probabilities for one of them, $R \rightarrow L$,
  i.e., the relationship between "trap" (R) and "locked" (L)

tree diagram

- The *tree diagram* shows all the possible "truth" relationships between R and L:



- The tree illustrates some probability relationships, for example:

$$Pr(L) = Pr(L|R)Pr(R) + Pr(L|\neg R)Pr(\neg R)$$

and

$$Pr(\neg L) = Pr(\neg L|R)Pr(R) + Pr(\neg L|\neg R)Pr(\neg R)$$

- We can determine the probability of each condition being true or false by collecting data (statistics) during the game. Then the NPC can use those statistics to perform inference when playing the game.
- Suppose that the NPC wants to know the probability that the box contains a trap (R), given that the box is locked (L), i.e., $Pr(R|L)$. Assume that the NPC can sense if the box is locked without doing itself any damage. The risk is in opening the box, when it would be revealed if there is a trap in the box.
- The value $Pr(R|L)$ can be determined by using **Bayes Rule**, as follows:

$$Pr(R|L) = \frac{Pr(L|R)Pr(R)}{Pr(L)}$$

- In order to solve this equation, we need to know:
  - $Pr(R) =$ the probability that the box contains a trap
  - $Pr(L) =$ the probability that the box is locked
  - $Pr(L|R) =$ the probability that the box is locked given that it contains a trap

- Suppose that we play the game and the NPC encountered 100 boxes.
  37 of the boxes contained traps (R), and 63 did not ($\neg R$).
  29 of the boxes containing traps were locked ($L|R$); 8 were not locked ($\neg L|R$).
  18 of the boxes that did not contain traps were locked ($L|\neg R$); 45 were not ($\neg L|\neg R$).

$$\begin{array}{ll} Pr(R) = 37/100 = 0.37 & Pr(\neg R) = 63/100 = 0.63 \\ Pr(L|R) = 29/37 = 0.78 & Pr(L|\neg R) = 18/63 = 0.29 \\ Pr(\neg L|R) = 8/37 = 0.22 & Pr(\neg L|\neg R) = 45/63 = 0.71 \end{array}$$

- In order to use Bayes rule to determine $Pr(R|L)$, we need to know $Pr(L)$, which we can do using the above data and the rule given on the tree diagram slide:

$$Pr(L) = Pr(L|R)Pr(R) + Pr(L|\neg R)Pr(\neg R)$$
$$= (0.78)(0.37) + (0.29)(0.63) = 0.4713$$

and then substituting in Bayes rule:

$$Pr(R|L) = \frac{Pr(L|R)Pr(R)}{Pr(L)}$$
$$= \frac{(0.78)(0.37)}{0.4713} = 0.61$$

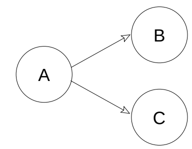
- So, there is a 63% probability that the box contains a trap, given that the NPC knows that the box is locked.

- In the course of playing the game, the NPC will want to know the likelihood that a box contains a trap or not, regardless of whether the box is locked or not.
- So the NPC will need to know the probability that the box contains a trap (R), given that the box is not locked ($\neg L$), i.e., $Pr(R|\neg L)$.
- We can compute $Pr(R|\neg L)$ in the same way as before:

$$
\begin{aligned}
Pr(R|\neg L) &= \frac{Pr(\neg L|R)Pr(R)}{Pr(\neg L)} \\
&= \frac{Pr(\neg L|R)Pr(R)}{Pr(\neg L|R)Pr(R) + Pr(\neg L|\neg R)Pr(\neg R)} \\
&= \frac{(0.22)(0.37)}{(0.22)(0.37) + (0.71)(0.63)} = 0.15
\end{aligned}
$$

- So, there is a 15% probability that the box contains a trap, given that the NPC knows that the box is not locked.

common cause network



- This *common cause* network says that:
  - $Pr(A)$ effects $Pr(B)$
  - $Pr(A)$ effects $Pr(C)$
- The two probabilities $Pr(B)$ and $Pr(C)$ are *conditionally dependent* on each other; as long as A is not known to be true.
  - Learning something about $Pr(A)$ tells you something about $Pr(B)$ and $Pr(C)$.
  - Learning something about $Pr(B)$ tells you something about $Pr(A)$, which in turn tells you something about $Pr(C)$.
  - Similarly, learning something about $Pr(C)$ tells you something about $Pr(A)$, which in turn tells you something about $Pr(B)$.

- If A is known to be true, then $Pr(B)$ and $Pr(C)$ are *independent* of each other.
- Otherwise, you can compute the probability of B using information about $Pr(A)$:

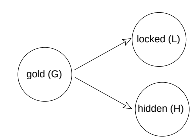
$$Pr(B) = Pr(B|A)Pr(A) + Pr(B|\neg A)Pr(\neg A)$$

  And, you can compute the probability of C using information about $Pr(A)$:

$$Pr(C) = Pr(C|A)Pr(A) + Pr(C|\neg A)Pr(\neg A)$$

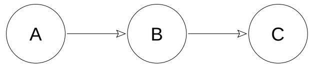
- We can say that the effects B and C can both be *caused* by the same condition, A; i.e., they share a common cause.
- This is called "diagnostic reasoning".

common cause example



- The network above shows two common causes resulting from gold being in a box (G): either L, the box is locked because it contains gold; or H, the box is hidden because it contains gold.
- If we know that the box contains gold, then we can compute, independently, the probability that the box is locked and the probability that the box is hidden.
- If we don't know whether the box contains gold, then the other probabilities (of locked and of hidden) are dependent on each other.

causal chain



- This *causal chain* says that:
  - $Pr(A)$ effects $Pr(B)$
  - $Pr(B)$ effects $Pr(C)$
- The two probabilities $Pr(A)$ and $Pr(C)$ are *conditionally dependent* on each other through B, as long as B is not known to be true.
  - Learning something about $Pr(A)$ tells you something about $Pr(B)$.
  - Learning something about $Pr(B)$ tells you something about $Pr(C)$.
  - and vice versa:
    Learning something about $Pr(B)$ tells you something about $Pr(C)$.
    learning something about $Pr(C)$ tells you something about $Pr(B)$.
- If B is known to true, then $Pr(A)$ and $Pr(C)$ are *conditionally independent*; i.e., their probabilities are independent of each other.

- We can compute the probability of B given information about $Pr(A)$:

$$Pr(B) = Pr(B|A)Pr(A) + Pr(B|\neg A)Pr(\neg A)$$

  And we can compute the probability of C given information about $Pr(B)$:

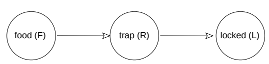
$$Pr(C) = Pr(C|B)Pr(B) + Pr(C|\neg B)Pr(\neg B)$$

- We use the **Chain Rule** to compute the conditional probability of C given information about A:

$$Pr(C|A) = Pr(C|B)Pr(B|A) + Pr(C|\neg B)Pr(\neg B|A)$$

- Performing this type of inference is called "predictive reasoning".

causal chain example



- The causal chain above illustrates that food being in the box F is a cause of the box being a trap R which is a cause of the box being locked L.
- In this case, the probability of the box containing a trap depends on the probability that the box contains food.
- We want to know the probability that the box contains food given that it is locked: $Pr(F|L)$.
- We can use Bayes rule, as follows:

$$Pr(F|L) = \frac{Pr(L|F)Pr(F)}{Pr(L)}$$

- Assume that we know the probability that there is food in the box ($Pr(F)$), so we just need to find $Pr(L)$ and $Pr(L|F)$.

- To find $Pr(L)$, the probability that the box is locked, we can use the formula we derived earlier, based on the causal relationship between L and R, i.e., $R \rightarrow L$:

$$Pr(L) = Pr(L|R)Pr(R) + Pr(L|\neg R)Pr(\neg R)$$

- We can find the probability of the box being locked given that it contains food, $Pr(L|F)$ using the **Chain Rule**, as follows:

$$Pr(L|F) = Pr(L|R)Pr(R|F) + Pr(L|\neg R)Pr(\neg R|F)$$

- Thus:

$$Pr(F|L) = \frac{(Pr(L|R)Pr(R|F) + Pr(L|\neg R)Pr(\neg R|F))Pr(F)}{Pr(L|R)Pr(R) + Pr(L|\neg R)Pr(\neg R)}$$

fuzzy logic

- Often, it is hard to write code where decisions are based on precise numbers. Instead, it is easier to use a *range* of number values.
- For example, we might say that a *high* risk is opening a box when there is a greater than 80% chance that the box contains a trap; whereas a *low* risk is opening a box with a 10% or less chance that it contains a trap.
- The area of *fuzzy logic* associates terms, such as *high*, and *low*, with ranges of values.
- Then, in the configuration of the game, perhaps in a script, you can define the risk for individual personalities of NPCs. For example, a *cautious* NPC might define the levels risk of opening a box given the following probabilities of there being a trap in the box:

high	medium	low
$> 50\%$	$\leq 50\%\ and\ \geq 20\%$	$< 20\%$

  whereas a *daring* NPC might define the levels risk of opening a box given the following probabilities of there being a trap in the box:

high	medium	low
$> 90\%$	$\leq 90\%\ and\ \geq 50\%$	$< 50\%$

to do:

- read chapter 13 from *AI for Game Developers*