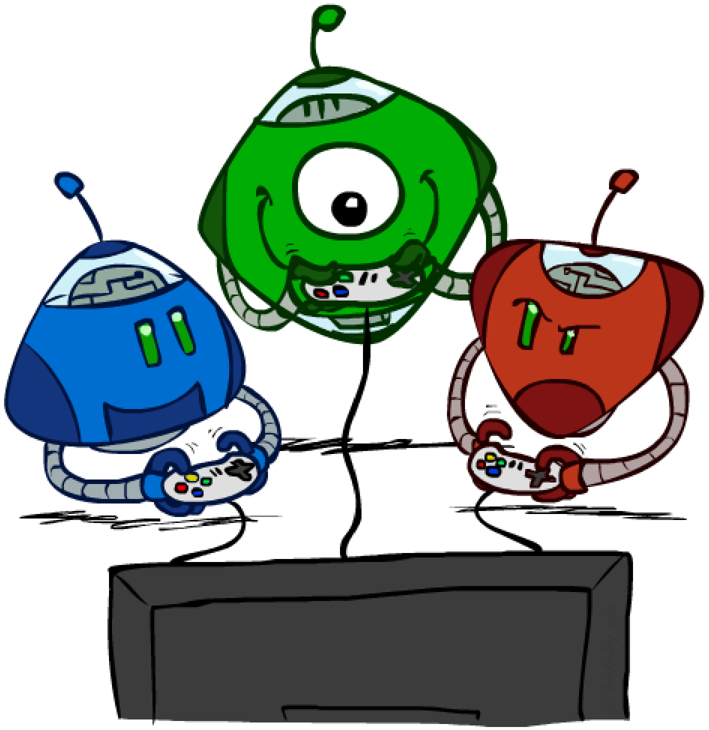
Reasoning Under Uncertainty

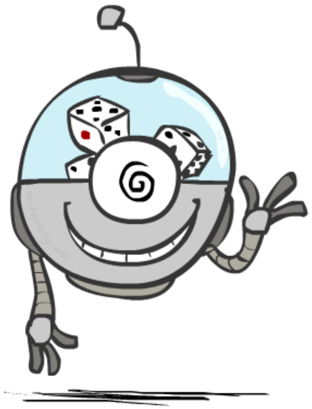
With slides from Dan Klein and Pieter Abbeel

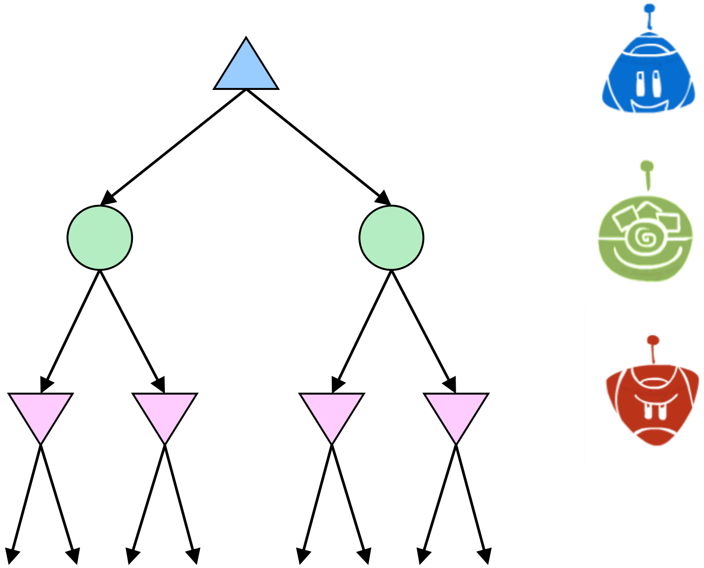
Other Game Types



Mixed Layer Types

- E.g. Backgammon
- Expectiminimax
  - Environment is an extra “random agent” player that moves after each min/max agent
  - Each node computes the appropriate combination of its children





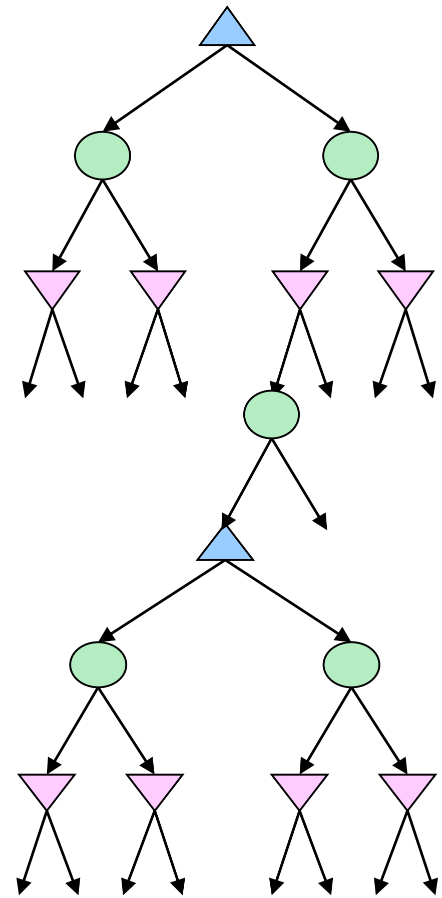
Example: Backgammon

- Dice rolls increase *b*: 21 possible rolls with 2 dice
  - Backgammon ≈ 20 legal moves
  - Depth 2 = $20 \times (21 \times 20)^3 = 1.2 \times 10^9$
- As depth increases, probability of reaching a given search node shrinks
  - So usefulness of search is diminished
  - So limiting depth is less damaging
  - But pruning is trickier…

Image: Wikipedia

Backgammon: Impact on Lookahead

- Dice rolls increase branching factor
  - 21 possible rolls with 2 dice
- Backgammon has ~20 legal moves for a given roll
  - ~6K with 1-1 roll
- At depth 4 there are 20 * (21 * 20)**3 ≈ 1.2B boards
- As depth increases, probability of reaching a given node shrinks
  - value of lookahead is diminished
  - alpha-beta pruning is much less effective
- [TDGammon]() used depth-2 search + very good static evaluator to achieve world-champion level



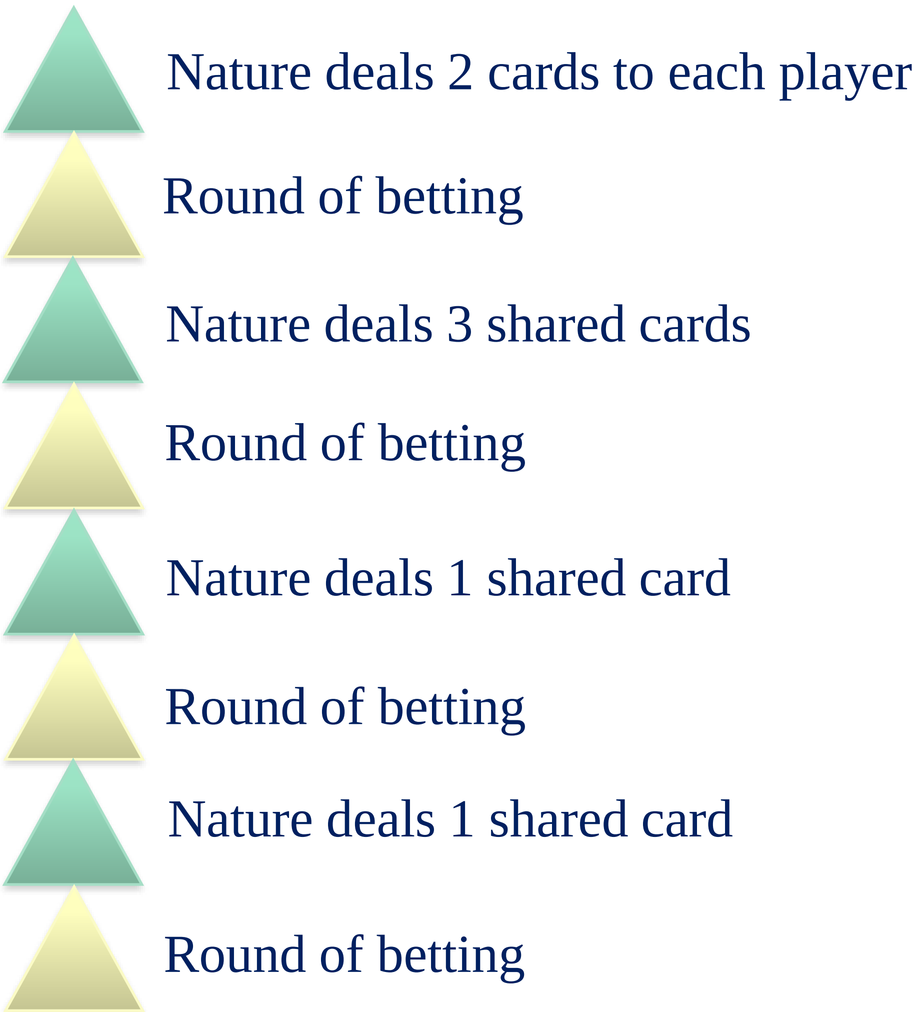
Games with **imperfect** information

- Example: card games, where opponent's initial cards are unknown
  - We can calculate a probability for each possible deal
  - Like having one big dice roll at the beginning of the game
- Possible approach: compute minimax value of each action in each deal, then choose the action with highest expected value over all deals
- Special case: if action is optimal for all deals, it's optimal
- GIB, a top bridge program, approximates this idea by
  1) Generate 100 deals consistent with bidding information
  2) pick the action that wins most tricks on average

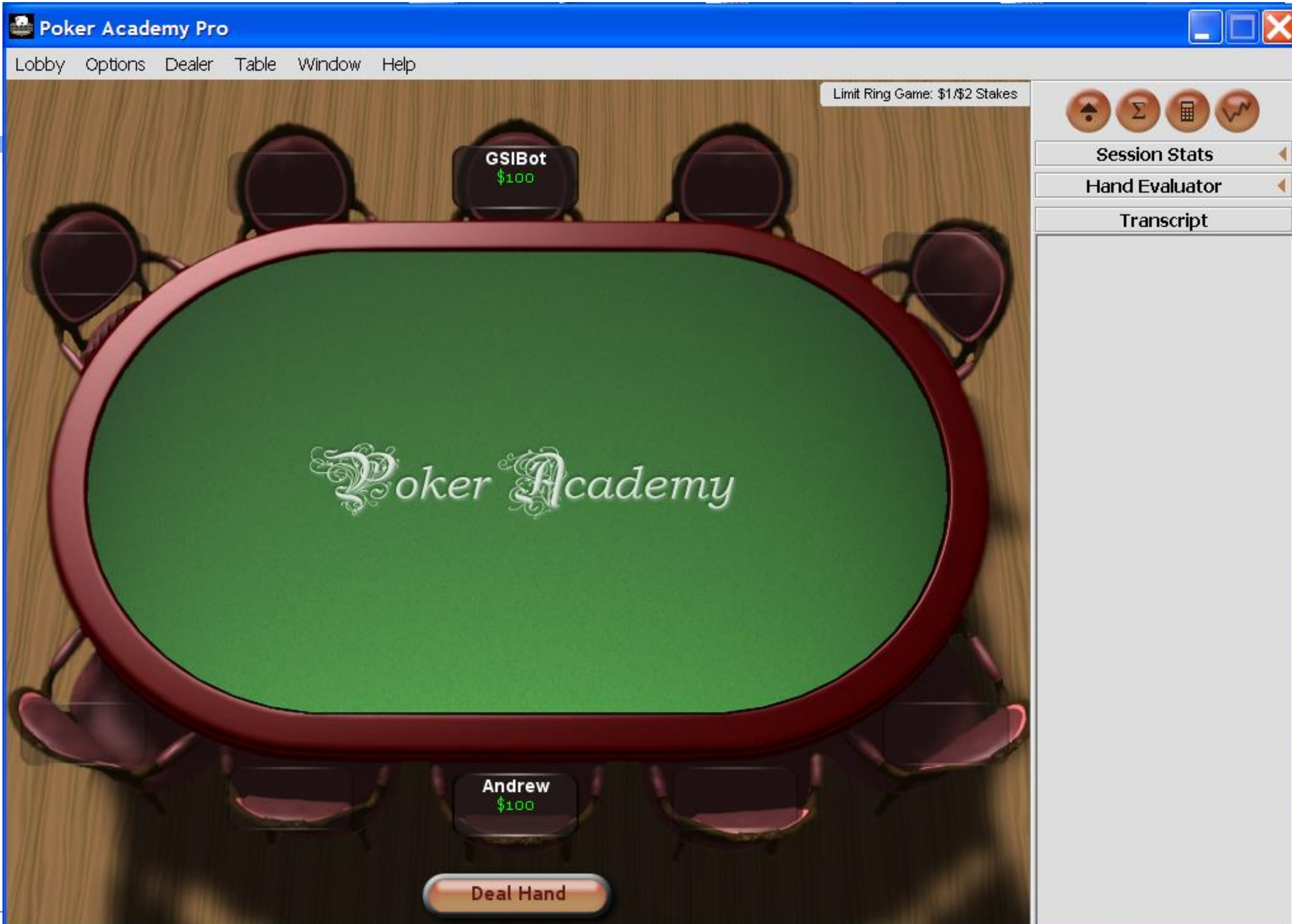
Poker

- Recognized challenge problem in AI
  - Hidden information: other players' cards
  - Uncertainty about future events
  - Deceptive strategies needed in a good player
- Very large game trees
- Texas Hold'em: most popular variation

Texas Hold'em poker



- 2-player Limit Texas Hold'em has ~10^{18} leaves in game tree
- Losslessly abstracted game too big to solve => abstract more => lossy


Poker Academy Pro
Lobby
Options
Dealer
Table
Window
Help
Limit Ring Game: $1/$2 Stakes
Session Stats
Hand Evaluator
Transcript
GSIBot
$100
Poker Academy
Andrew
$100
Deal Hand


Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
GSIBot
$99
Call $0.50
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Pot: $2
Poker Academy
Andrew
$99
Fold
Check
Bet $1


Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
GSIBot
$99
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Pot: $2
Andrew
$99
Fold
Check
Bet $1


Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
Session Stats
Hand Evaluator
Transcript
GSIBot
$98
Bet $1
Pot: $3
Andrew
$99
Fold
Call $1
Raise $1
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1


Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
GSIBot
$98
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1
Andrew calls $1
TURN: Qd 7s 4c 3s
Pot: $4
Andrew
$98
Fold
Check
Bet $2

Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1
Andrew calls $1
TURN: Qd 7s 4c 3s
Andrew bets $2
GSIBot calls $2
RIVER: Qd 7s 4c 3s Qs
GSIBot
$96
Pot: $8
Andrew
$96
Fold
Check
Bet $2


Poker Academy Pro
Lobby
Options
Dealer
Table
Window
Help
Limit Ring Game: $1/$2 Stakes
Session Stats
Hand Evaluator
Transcript
GSIBot
$94
Bet $2
Pot: $10
Andrew
$96
Fold
Call $2
Raise $2
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1
Andrew calls $1
TURN: Qd 7s 4c 3s
Andrew bets $2
GSIBot calls $2
RIVER: Qd 7s 4c 3s Qs
Andrew checks
GSIBot bets $2

Poker Academy Pro
Lobby Options Dealer Table Window Help
Limit Ring Game: $1/$2 Stakes
GSIBot
$106
GSIBot wins $12 with Two Pair, Queens and Sevens
Andrew
$94
Deal Hand
Session Stats
Hand Evaluator
Transcript
HAND #428,331
GSIBot blinds $0.50
Andrew blinds $1
Your hole cards are: 2s Kh
GSIBot calls $0.50
Andrew checks
FLOP: Qd 7s 4c
Andrew checks
GSIBot bets $1
Andrew calls $1
TURN: Qd 7s 4c 3s
Andrew bets $2
GSIBot calls $2
RIVER: Qd 7s 4c 3s Qs
Andrew checks
GSIBot bets $2
Andrew calls $2
GSIBot shows 2c 7c
Andrew shows 2s Kh
GSIBot wins $12 with Two Pair, Queens and Sevens

Sequential imperfect information games

- Players face uncertainty about the state of the world
- Most real-world games are like this
  - A robot facing adversaries in an uncertain, stochastic environment
  - Almost any card game in which the other players' cards are hidden
  - Almost any economic situation in which the other participants possess private information (*e.g.* valuations, quality information)
    - Negotiation
    - Multi-stage auctions (e.g., English)
    - Sequential auctions of multiple items
  - …
- This class of games presents several challenges for AI
  - Imperfect information
  - Risk assessment and management
  - Speculation and counter-speculation
- Techniques for solving <u>sequential complete-information</u> games (like chess) don't directly apply

Science AAAS

Home | News | Journals | Topics | Careers

Latest News | ScienceInsider | ScienceShots | Sifter | From the Magazine | About News | Quizzes

SHARE

13

Researchers have developed a poker-playing computer program that can defeat even the best human players.

(Illustration) Peter and Maria Hoey/www.peterhoey.com

Texas Hold 'em poker solved by computer

By Emily Conover | Jan. 8, 2015 , 3:30 PM

http://www.sciencemag.org/news/2015/01/texas-hold-em-poker-solved-computer

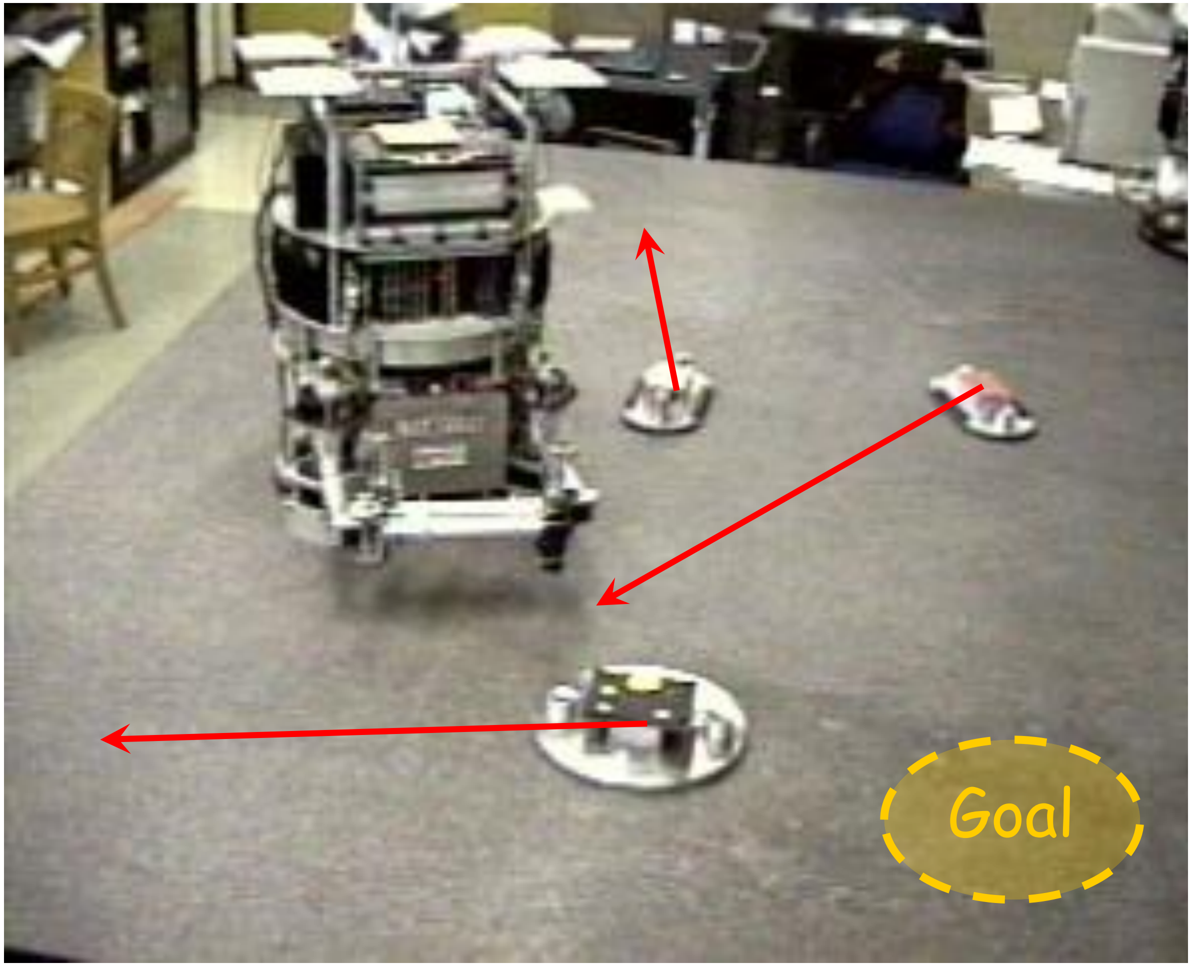
Rough Plan (Next 2 weeks)

- Acting under uncertainty
- **Today:** Probabilistic models (single decision)
  - Inference
  - Begin Bayesian reasoning
- **Next week:** uncertainty+time = sequential decisions
  - Approximate inference
  - Hidden Markov Models (HMMs)
- **Project 3:** Ghost Busters (due after Spring break)

Uncertainty

- Uncertain input (sensors): https://www.youtube.com/watch?v=9OgPAyRUI3I
- Uncertainty in action (outcome): https://www.youtube.com/watch?v=g0TaYhjpOfo
- Dynamic environment (sensors + actions) https://www.youtube.com/watch?v=HacG_FWWPOw

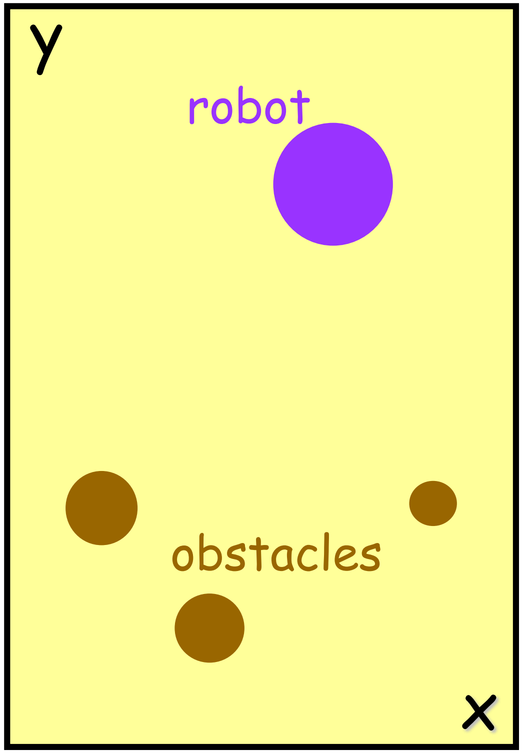
More Detailed Example: Robot Motion



A robot with imperfect sensing must reach a goal location among moving obstacles (dynamic world)

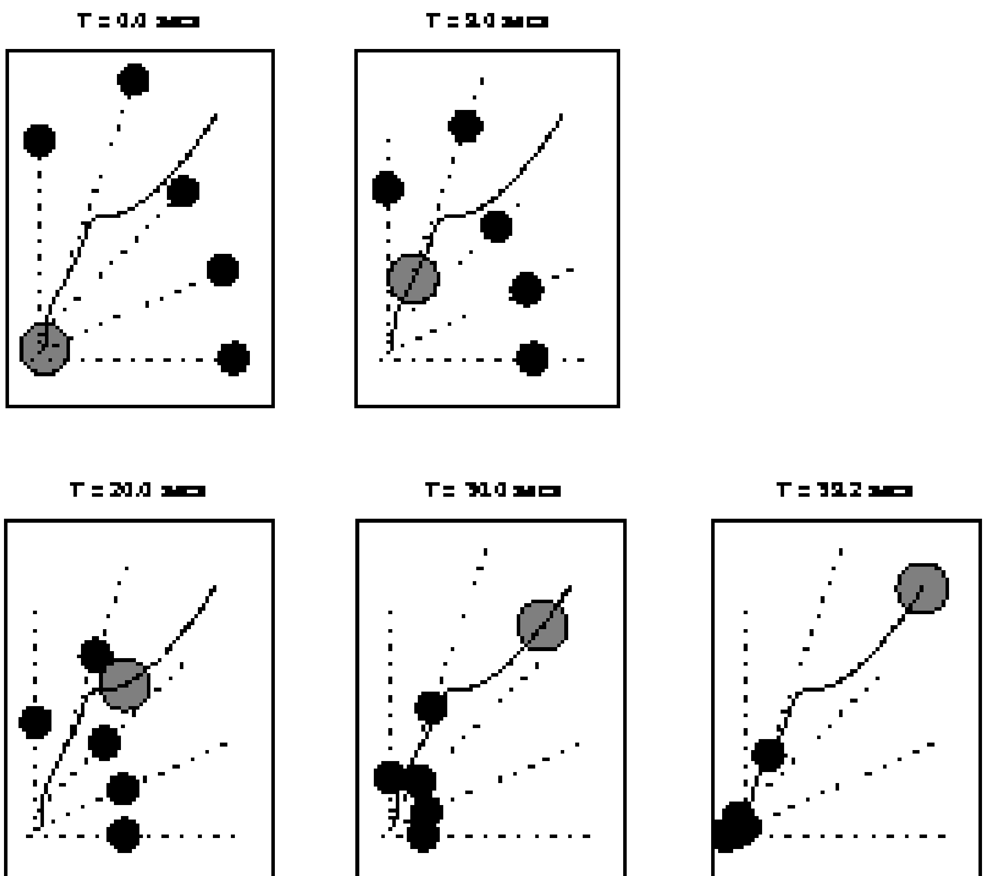
Model, Sensing, and Control

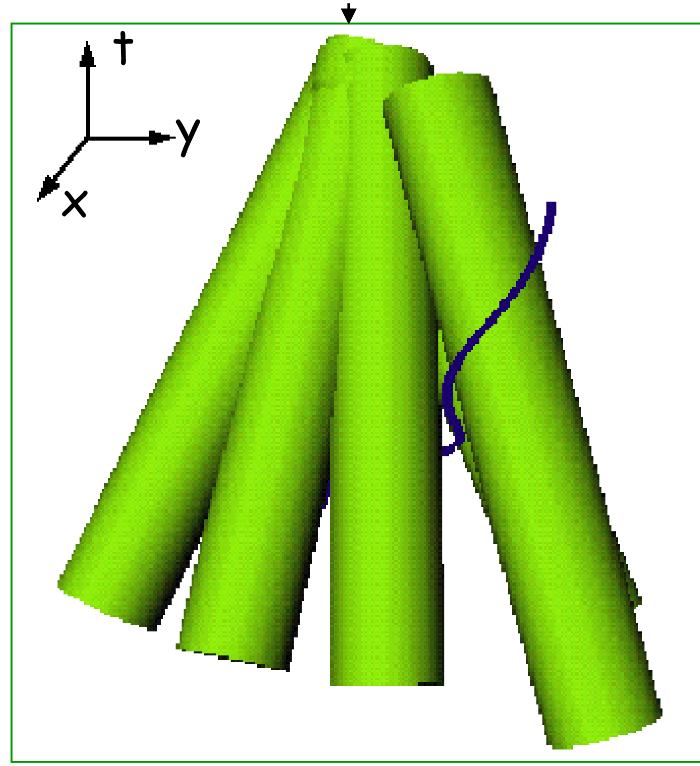
- The robot and the obstacles are represented as disks moving in the plane
- The position and velocity of each disc are measured by an overhead camera every 1/30 sec



Motion Planning

The robot plans its trajectories in configuration×time space using a probabilistic roadmap (PRM) method





Obstacle map to cylinders in configuration×time space

But executing this trajectory is likely to fail ...

1) The measured velocities of the obstacles are inaccurate
2) Tiny particles of dust on the table affect trajectories and contribute further to deviation

   → Obstacles are likely to deviate from their expected trajectories

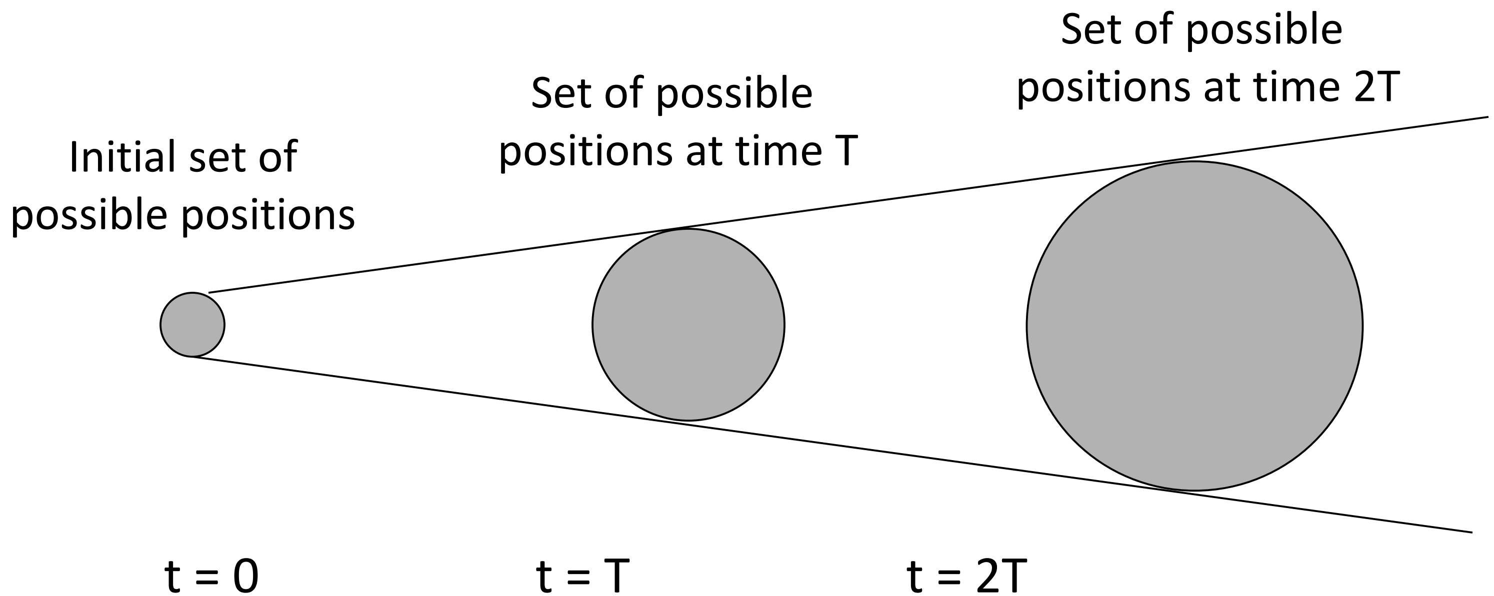
3) Planning takes time, and during this time, obstacles keep moving

   → The computed robot trajectory is not properly synchronized with those of the obstacles

→ Planning must take both uncertainty in world state and time constraints into account

Dealing with Uncertainty

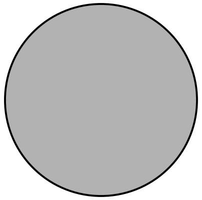
- The robot can handle uncertainty in an obstacle position by representing the set of all positions of the obstacle that the robot think possible at each time (belief state)
- For example, this set can be a disc whose radius grows linearly with time



Dealing with Uncertainty

- The robot can handle uncertainty in an obstacle position by representing the set of all positions of the obstacle that the robot think possible at each time (belief state)
- For example, this set can be a disc whose radius grows linearly with time

The robot must plan to be outside this disc at time t = T

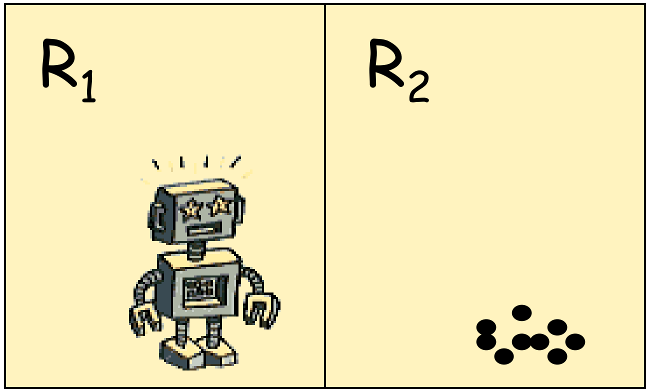


t = T

Imperfect Observation of the World

Observation of the world can be:

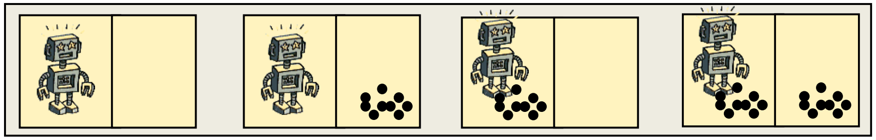
- Partial, e.g., a vision sensor can't see through obstacles (lack of percepts)



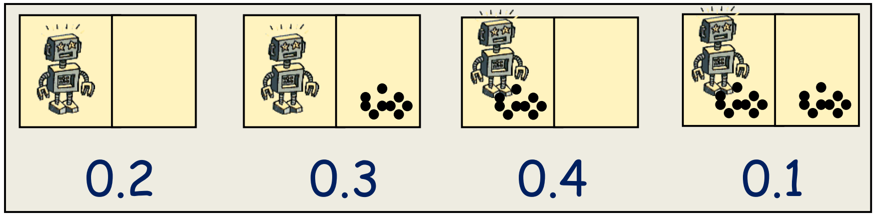
The robot may not know whether there is dust in room R2

Definition: Belief State

- In the presence of non-deterministic **sensory uncertainty**, an agent belief state represents all the states of the world that it thinks are possible at a given time or at a given stage of reasoning

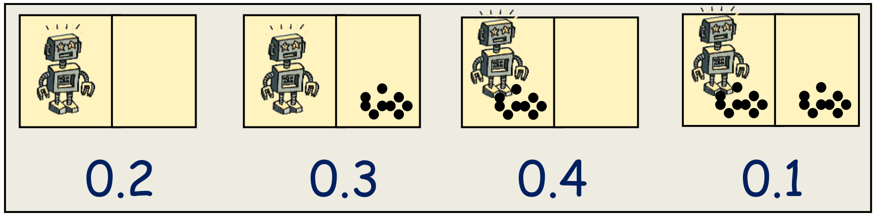


- In the probabilistic model of uncertainty, a probability is associated with each state to measure its likelihood to be the actual state



What do probabilities mean?

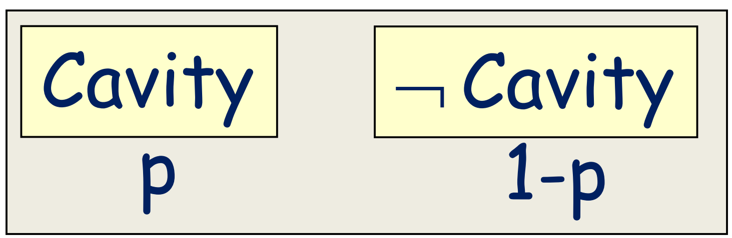
- Probabilities have a natural frequency interpretation
- The agent believes that if it was able to return many times to a situation where it has the same belief state, then the actual states in this situation would occur at a relative frequency defined by the probabilistic distribution



This state would occur
20% of the times

Belief State: Example

- Consider a world where a dentist agent D meets a new patient P
- D is interested in only one thing: whether P has a cavity, which D models using the proposition Cavity
- Before making any observation, D's belief state is:



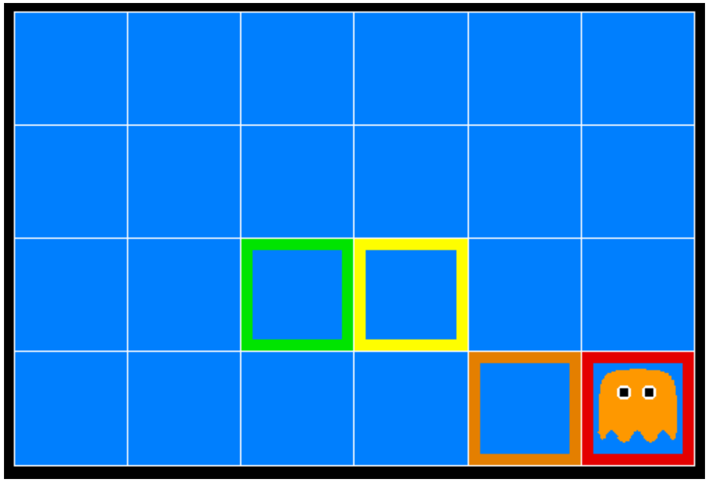
- This means that D believes that a fraction p of patients have cavities

Where do probabilities come from?

- Frequencies observed in the past, e.g., by the agent, its designer, or others
- Symmetries, e.g.:
  - If I roll a dice, each of the 6 outcomes has probability 1/6
- Subjectivism, e.g.:
  - If I drive on Highway 280 at 120mph, I will get a speeding ticket with probability 0.6
  - Principle of indifference: If there is no knowledge to consider one possibility more probable than another, give them the same probability

Pacman: Ghost position is uncertain

- A ghost is in the grid somewhere
- Sensor readings tell how close a square is to the ghost
  - On the ghost: red
  - 1 or 2 away: orange
  - 3 or 4 away: yellow
  - 5+ away: green



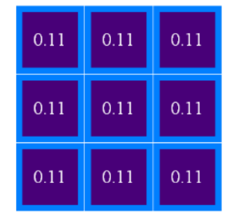
- Sensors are noisy, but we know P(Color | Distance)

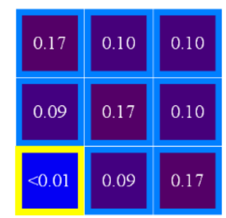
P(red | 3)	P(orange | 3)	P(yellow | 3)	P(green | 3)
0.05	0.15	0.5	0.3

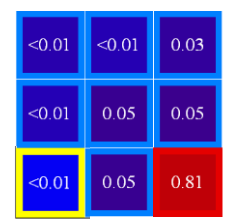
Pacman Uncertainty: 2

- General situation:
  - **Observed variables (evidence)**: Agent knows certain things about the state of the world (e.g., sensor readings or symptoms)
  - **Unobserved variables**: Agent needs to reason about other aspects (e.g. where an object is or what disease is present)
  - **Model**: Agent knows something about how the known variables relate to the unknown variables

- Probabilistic reasoning gives us a framework for managing our beliefs and knowledge

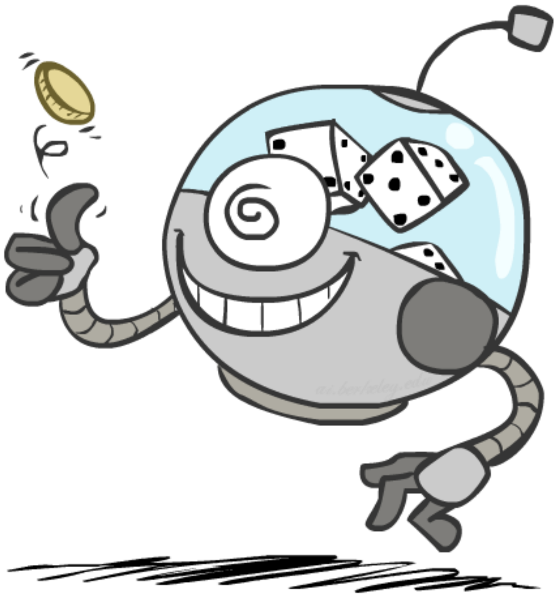






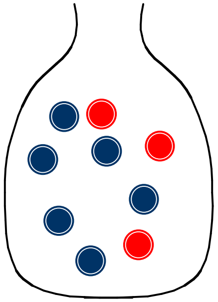
Random Variables

- A random variable is some aspect of the world about which we (may) have uncertainty
  - R = Is it raining?
  - T = Is it hot or cold?
  - D = How long will it take to drive to work?
  - L = Where is the ghost?
- We denote random variables with capital letters
- Like variables in a CSP, random variables have domains
  - R in {true, false}   (often write as {+r, -r})
  - T in {hot, cold}
  - D in $[0, \infty)$
  - L in possible locations, maybe {(0,0), (0,1), …}



Probability Review

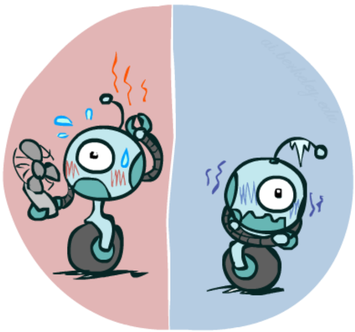
- Bag with 10 marbles: 3 red, 7 blue



  - Reach in, take one, put it back
  - Repeat lots of times.
  - What fraction red? About .3
  - P(red) = .3

Probability Distributions

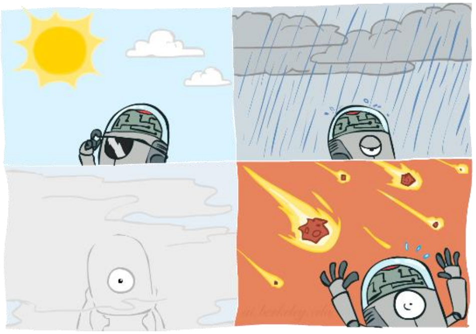
- Associate a probability with each value
  - Temperature:



$P(T)$

T	P
hot	0.5
cold	0.5

- Weather:



$P(W)$

W	P
sun	0.6
rain	0.1
fog	0.3
meteor	0.0

Probability Distribution

- The probability for each value of a random variable

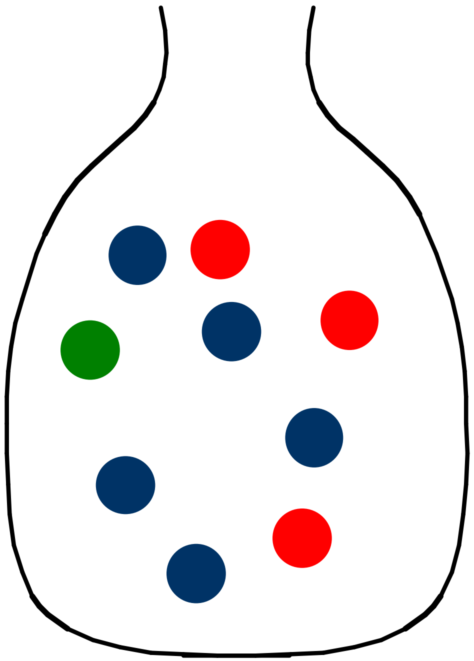
    if color = (red, blue)

    P(color) = (.3, .7)

Basic Properties

- $0 \leq P(A) \leq 1$
- $P(true) = 1$

  $P(red \vee blue \vee green) = 1$



- $P(false) = 0$

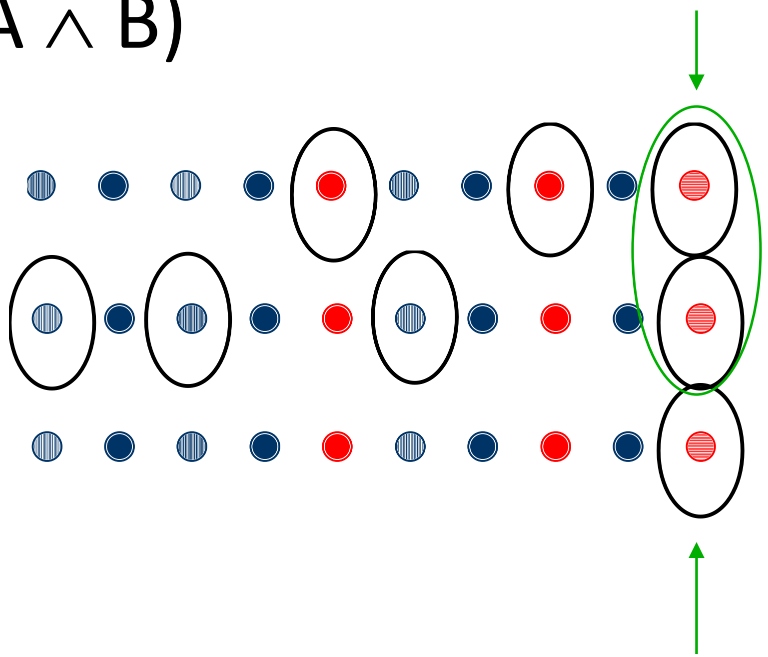
  $P(black) = 0$

Basic Properties

- $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

.3	P(red)
+ .4	P(striped)
- .1	P(red $\wedge$ striped)

Counted twice



So subtract once

P(red $\vee$ striped) = .6

Probability Distributions

- Unobserved random variables have distributions

$P(T)$

T	P
hot	0.5
cold	0.5

$P(W)$

W	P
sun	0.6
rain	0.1
fog	0.3
meteor	0.0

Shorthand notation:

$$P(hot) = P(T = hot),$$
$$P(cold) = P(T = cold),$$
$$P(rain) = P(W = rain),$$
$$\ldots$$

OK *if* all domain entries are unique

- A distribution is a TABLE of probabilities of values

- A probability (lower case value) is a single number $P(W = rain) = 0.1$

- Must have: $\forall x \ \ P(X = x) \geq 0$ and $\sum_x P(X = x) = 1$

Joint Distributions

- A *joint distribution* over a set of random variables: $X_1, X_2, \ldots X_n$ specifies a real number for each assignment (or *outcome*):

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

$$P(x_1, x_2, \ldots x_n)$$

  - Must obey: $P(x_1, x_2, \ldots x_n) \geq 0$

$$\sum_{(x_1, x_2, \ldots x_n)} P(x_1, x_2, \ldots x_n) = 1$$

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

- Size of distribution of **n** variables with domain sizes **d**?
  - O(size) = ?
  - For all but the smallest distributions, impractical to write out!

Probabilistic Models

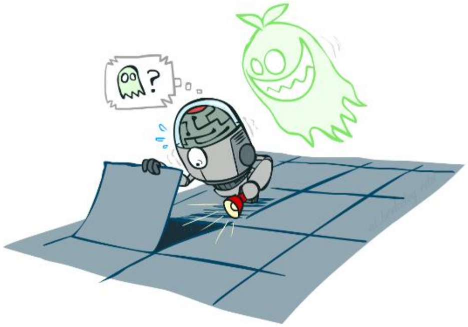
- A probabilistic model is a joint distribution over a set of random variables

- Probabilistic models:
  - (Random) variables with domains
  - Assignments are called *outcomes*
  - Joint distributions: say whether assignments (outcomes) are likely
  - *Normalized:* sum to 1.0
  - Ideally: only certain variables directly interact

- Constraint satisfaction problems:
  - Variables with domains
  - Constraints: state whether assignments are possible
  - Ideally: only certain variables directly interact

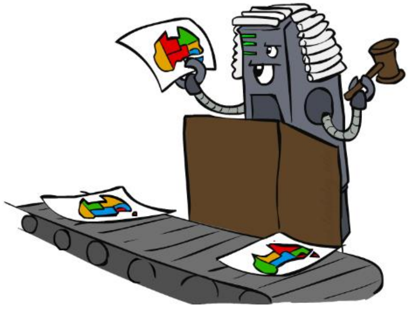
Distribution over T,W

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3



Constraint over T,W

T	W	P
hot	sun	T
hot	rain	F
cold	sun	F
cold	rain	T



Events

- An *event* is a set E of outcomes

$$P(E) = \sum_{(x_1 \ldots x_n) \in E} P(x_1 \ldots x_n)$$

- From a joint distribution, we can calculate the probability of any event
  - Probability that it's hot AND sunny
    P(+hot, + sun) =
  - Probability that it's hot?
    P(+hot) =
  - Probability that it's hot OR sunny?
  - **P(+hot OR +sun)=**

- Typically, the events we care about are *partial assignments*, like P(T=hot)

$$P(T,W)$$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Events

- An *event* is a set E of outcomes

$$P(E) = \sum_{(x_1 \dots x_n) \in E} P(x_1 \dots x_n)$$

- From a joint distribution, we can calculate the probability of any event
  - Probability that it's hot AND sunny?
  - Probability that it's hot?
  - Probability that it's hot OR sunny?
- Typically, the events we care about are *partial assignments*, like P(T=hot)

$$P(T, W)$$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

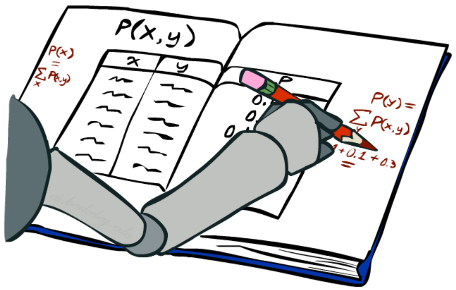
Quiz 1: Events (work in pairs)

- P(+x, +y) ?
- P(+x) ?
- P(-y OR +x) ?

$$P(X, Y)$$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

Marginal Distributions



- Marginal distributions are sub-tables which eliminate variables
- Marginalization (summing out): Combine collapsed rows by adding

$P(T,W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(t) = \sum_s P(t,s)$$

$P(T)$

T	P
hot	?
cold	?

$$P(s) = \sum_t P(t,s)$$

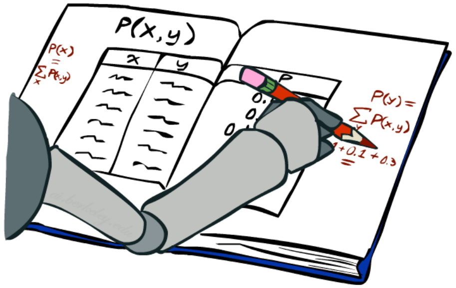
$P(W)$

W	P
sun	?
rain	?

$$P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2)$$

Marginal Distributions

- Marginal distributions are sub-tables which eliminate variables
- Marginalization (summing out): Combine collapsed rows by adding



$P(T,W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(t) = \sum_s P(t,s)$$

$P(T)$

T	P
hot	0.5
cold	0.5

$$P(s) = \sum_t P(t,s)$$

$P(W)$

W	P
sun	0.6
rain	0.4

$$P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2)$$

Quiz P2: Marginal Distributions

$P(X,Y)$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

$$P(x) = \sum_y P(x,y)$$

$$P(y) = \sum_x P(x,y)$$

$P(X)$

X	P
+x	
-x	

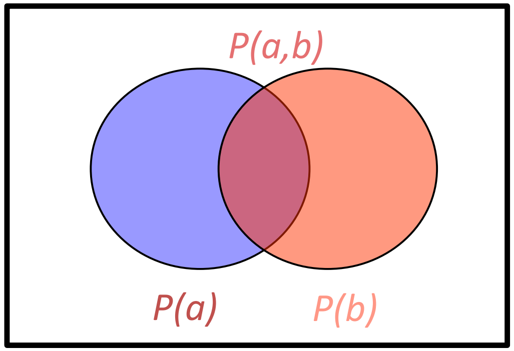
$P(Y)$

Y	P
+y	
-y	

Conditional Probabilities

- Relates joint and conditional probabilities
  - In fact, this is taken as the *definition* of a conditional probability

$$\boxed{P(a|b) = \frac{P(a,b)}{P(b)}}$$



$$P(T,W)$$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(W=s|T=c) = \frac{P(W=s,T=c)}{P(T=c)} = \frac{0.2}{0.5} = 0.4$$

$$= P(W=s,T=c) + P(W=r,T=c)$$
$$= 0.2 + 0.3 \quad = 0.5$$

P(hot|sun) = ?

P(cold|rain) = ?

Quiz P3: Conditional Probabilities

$P(X, Y)$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

- P(+x | +y) ?
- P(-x | +y) ?
- P(-y | +x) ?

Conditional Distributions

- Conditional distributions are probability distributions over some variables given fixed values of others

Conditional Distributions

$P(W|T)$

$P(W|T = hot)$

W	P
sun	0.8
rain	0.2

$P(W|T = cold)$

W	P
sun	0.4
rain	0.6

Joint Distribution

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Normalization Trick

$$P(T, W)$$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$\begin{aligned} P(W = s | T = c) &= \frac{P(W = s, T = c)}{P(T = c)} \\ &= \frac{P(W = s, T = c)}{P(W = s, T = c) + P(W = r, T = c)} \\ &= \frac{0.2}{0.2 + 0.3} = 0.4 \end{aligned}$$

$$\begin{aligned} P(W = r | T = c) &= \frac{P(W = r, T = c)}{P(T = c)} \\ &= \frac{P(W = r, T = c)}{P(W = s, T = c) + P(W = r, T = c)} \\ &= \frac{0.3}{0.2 + 0.3} = 0.6 \end{aligned}$$

$$P(W | T = c)$$

W	P
sun	0.4
rain	0.6

Normalization Trick

$$P(W=s|T=c) = \frac{P(W=s,T=c)}{P(T=c)} = \frac{P(W=s,T=c)}{P(W=s,T=c)+P(W=r,T=c)} = \frac{0.2}{0.2+0.3} = 0.4$$

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

SELECT the joint probabilities matching the evidence

$P(c, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3

NORMALIZE the selection (make it sum to one)

$P(W|T=c)$

W	P
sun	0.4
rain	0.6

$$P(W=r|T=c) = \frac{P(W=r,T=c)}{P(T=c)} = \frac{P(W=r,T=c)}{P(W=s,T=c)+P(W=r,T=c)} = \frac{0.3}{0.2+0.3} = 0.6$$

Normalization Trick

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

SELECT the joint probabilities matching the evidence

$P(c, W)$

T	W	P
cold	sun	0.2
cold	rain	0.3

NORMALIZE the selection (make it sum to one)

$P(W|T = c)$

W	P
sun	0.4
rain	0.6

- Why does this work? Sum of selection is P(evidence)! (P(T=c), here)

$$P(x_1|x_2) = \frac{P(x_1, x_2)}{P(x_2)} = \frac{P(x_1, x_2)}{\sum_{x_1} P(x_1, x_2)}$$

Quiz: Normalization Trick

- P(X | Y=-y) ?

$$P(X, Y)$$

X	Y	P
+x	+y	0.2
+x	-y	0.3
-x	+y	0.4
-x	-y	0.1

SELECT the joint probabilities matching the evidence

NORMALIZE the selection (make it sum to one)

To Normalize

- (Dictionary) To bring or restore to a $\boxed{\text{normal condition}}$

All entries sum to ONE

- Procedure:
  - Step 1: Compute Z = sum over all entries
  - Step 2: Divide every entry by Z

- Example 1

W	P
sun	0.2
rain	0.3

Normalize

Z = 0.5

W	P
sun	0.4
rain	0.6

- Example 2

T	W	P
hot	sun	20
hot	rain	5
cold	sun	10
cold	rain	15

Normalize

Z = 50

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Probabilistic Inference

- Probabilistic inference: compute a desired probability from other known probabilities (e.g. conditional from joint)

- We generally compute conditional probabilities
  - P(on time | no reported accidents) = 0.90
  - These represent the agent's *beliefs* given the eviden

- Probabilities change with new evidence:
  - P(on time | no accidents, 5 a.m.) = 0.95
  - P(on time | no accidents, 5 a.m., raining) = 0.80
  - Observing new evidence causes *beliefs to be update*

Inference by Enumeration

- General case:
  - Evidence variables: $E_1 \dots E_k = e_1 \dots e_k$
  - Query* variable: Q
  - Hidden variables: $H_1 \dots H_r$

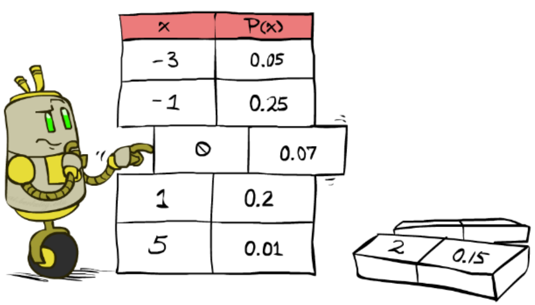
$\left.\begin{array}{l} E_1 \dots E_k = e_1 \dots e_k \\ Q \\ H_1 \dots H_r \end{array}\right\}$ $X_1, X_2, \dots X_n$ *All variables*

- We want:

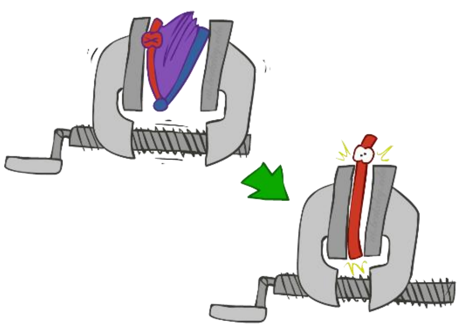
** Works fine with multiple query variables, too*

$$P(Q|e_1 \dots e_k)$$

- Step 1: Select the entries consistent with the evidence



- Step 2: Sum out H to get joint prob of Query and evidence



$$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} \underbrace{P(Q, h_1 \dots h_r, e_1 \dots e_k)}_{X_1, X_2, \dots X_n}$$

- Step 3: Normalize

$$\times \frac{1}{Z}$$

$$Z = \sum_q P(Q, e_1 \cdots e_k)$$

$$P(Q|e_1 \cdots e_k) = \frac{1}{Z} P(Q, e_1 \cdots e_k)$$

Inference by Enumeration

- P(W)?

- P(W | winter)?

- P(W | winter, hot)?

S	T	W	P
summer	hot	sun	0.30
summer	hot	rain	0.05
summer	cold	sun	0.10
summer	cold	rain	0.05
winter	hot	sun	0.10
winter	hot	rain	0.05
winter	cold	sun	0.15
winter	cold	rain	0.20

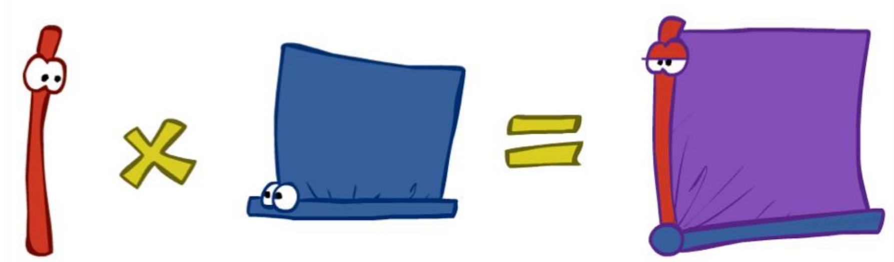
Inference by Enumeration: Issues

- Obvious problems:
  - Worst-case time complexity $O(d^n)$
  - Space complexity $O(d^n)$ to store the joint distribution

More Efficient Inference: The Product Rule

- Sometimes **given** conditional distributions but **want** the joint

$$P(y)P(x|y) = P(x,y) \quad \Longleftrightarrow \quad P(x|y) = \frac{P(x,y)}{P(y)}$$



The Product Rule

$$P(y)P(x|y) = P(x, y)$$

- Example:

$P(W)$

R	P
sun	0.8
rain	0.2

$P(D|W)$

D	W	P
wet	sun	0.1
dry	sun	0.9
wet	rain	0.7
dry	rain	0.3

$\Longleftrightarrow$

$P(D, W)$

D	W	P
wet	sun	
dry	sun	
wet	rain	
dry	rain	

The Chain Rule

- More generally, can always write any joint distribution as an incremental product of conditional distributions

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$$

$$P(x_1, x_2, \ldots x_n) = \prod_i P(x_i|x_1 \ldots x_{i-1})$$

- Why is this true?
  - Recursive decomposition using product rule

Bayes' Rule

- Two ways to factor a joint distribution over two variables:

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x)$$

- Dividing, we get:

$$P(x|y) = \frac{P(y|x)}{P(y)}P(x)$$

- Why is this at all helpful?
  - Lets us build one conditional from its reverse
  - Often one conditional is tricky but the other one is simple
  - Foundation of many systems we'll see later (e.g. ASR, MT)

- **In the running for most important AI, ML, DM equation!**

That's my rule!