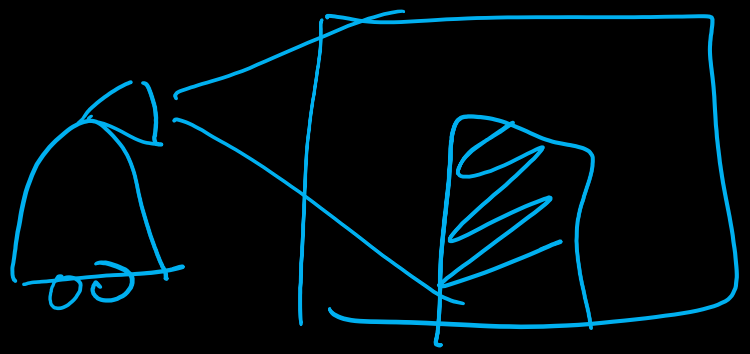
Decision Making I

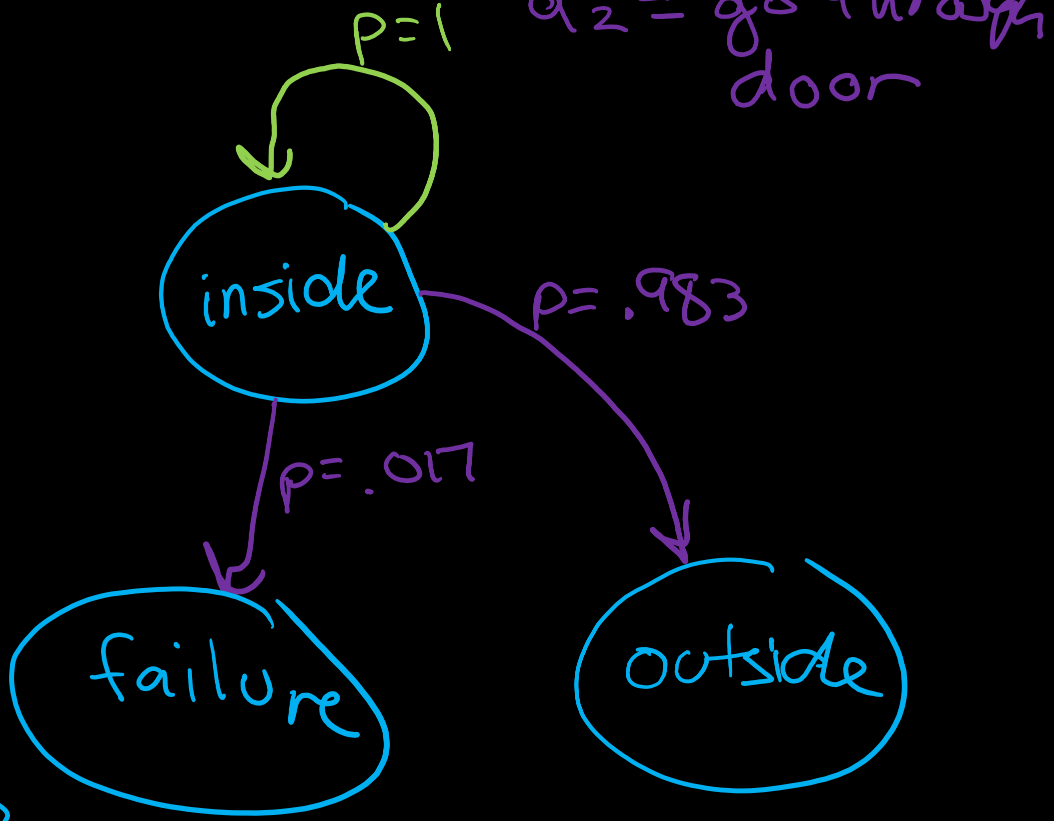
Katie DC

Notes from Decision Making Under Uncertainty, by M.J. Kochenderfer

Recall Estimation Example

a_1 = sense again
a_2 = go through door



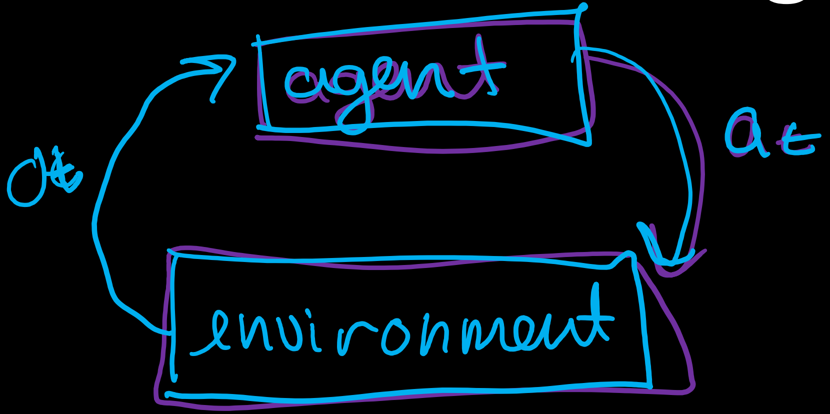


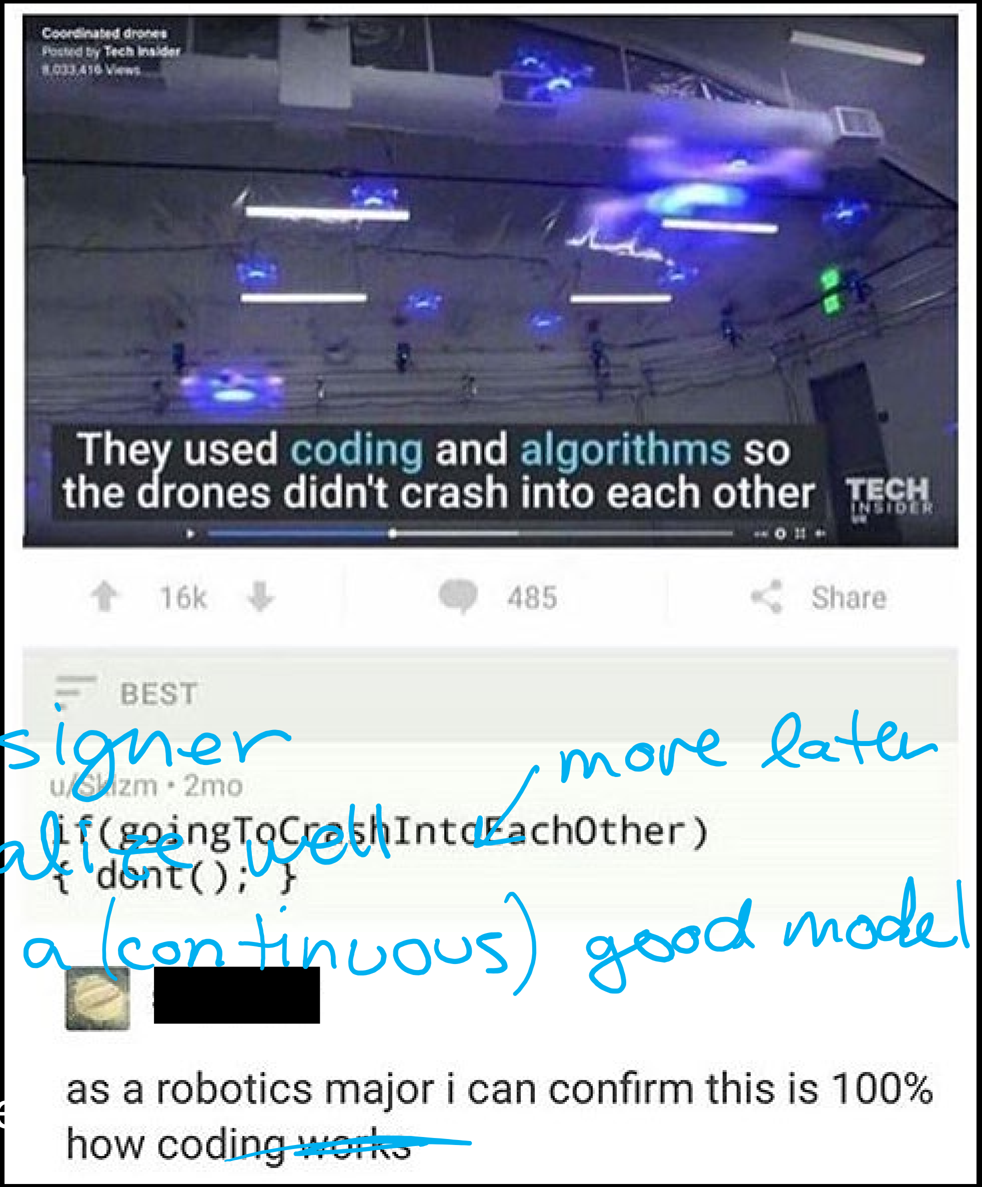
using Bayes Filtering, we can recursively update our belief about the state of the door

$bel(x_{t=2} = \text{is open}) = .983$

→ should we take an action?
→ should we probe or explore?

Decision Making Systems

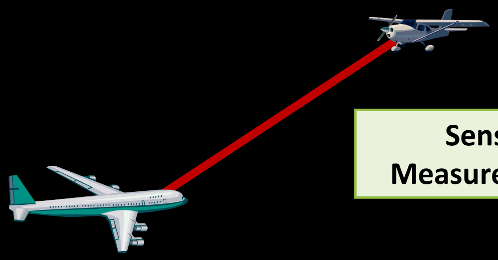




Methods

1. Explicit programming → burden on designer
2. Supervised learning → doesn't generalize well
3. Optimization / optimal control → require a (continuous) good model — more later
4. Planning
   - Given a stochastic model, how to algorithmically de
5. Reinforcement Learning
   - If model is unknown (or very complex), learn policy through experience

Traffic Alert and Collision Avoidance System (TCAS)



Sensor Measurements

```
IF (ITF.A LT G.ZTHR)
 THEN IF(ABS(ITF.VMD) LT G.ZTHR)
  THEN SET ZHIT;
  ELSE CLEAR ZHIT;
 ELSE IF (ITF.ADOT GE P.ZDTHR)
  THEN CLEAR ZHIT
  ELSE
   ITF.TAUV = -ITF.A/ITF.ADOT;
   IF (ITF.TAUV LT TVTHR AND
    ((ABS(ITF.VMD) LT G.ZTHR) OR
    (ITF.TAUV LT ITF.TRTRU))
    THEN SET ZHIT
    ELSE CLEAR ZHIT
IF (ZHIT EQ $TRUE AND
 ABS(ITF.ZDINT) GT P.MAXZDINT
 THEN CLEAR ZHIT
```

Resolution Advisory

Surveillance

Advisory Logic

Display

Slide Credit: Mykel Kochenderfer

M.J. Kochenderfer

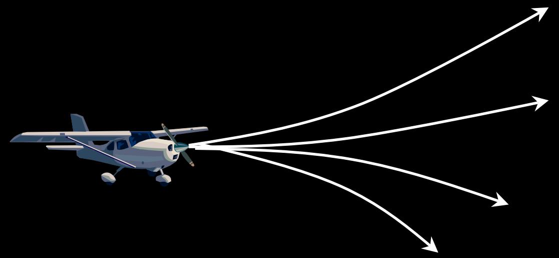

```
PROCESS Reversal_modeling;

    Default modeled separation for current RA is 0 if current RA is negative;
    Set own altitude and own rate to own tracked altitude and own tracked rate;

    IF (own does not follow his RAs)
        THEN  Model separation achieved assuming RA not followed;
                IF (current RA is a climb RA)
                    THEN  CLEAR flag indicating the sense of the RA after a reversal;
                    ELSE  SET flag indicating the sense of the RA after a reversal;
                IF (modeled separation achieved by continuing current RA greater than 1.2 *
                    P.CROSSTHR)
                    THEN  CLEAR reversal flag in ITF
        ELSE
        <Begin own is assumed to follow its RA>
                IF (current RA is positive)
                    THEN  model response to current RA;
                          <model maximum displayable rate for climb if current rate exceeds
                          maximum displayable rate or minimum displayable rate for descent if
                          current rate is less than minimum displayable rate>

                          IF (tracked response lags modeled response in RA direction AND
                              time since RA less than a parameter time AND
                              own's rate has not changed by more than P.MODEL_ZD since the
                               RA was first issued)
                               THEN  set own altitude and own rate to modeled altitude and rate
                                     for  use in reversal modeling;
                          Model separation achieved by continuing current RA;
                Set delay time to greater of pilot delay time remaining for last advisory against a
                    new threat, and the pilot quick reaction time;

                IF (considering a reversal from a descend RA to a climb RA)
                    THEN  set own goal rate to greater of own tracked rate (or maximum
                          displayable rate, whichever is less) and nominal climb rate;
                    ELSE  IF (own too close to ground to descend)
                               THEN  set own goal rate to zero;
                               ELSE  set own goal rate to lesser of own tracked rate (or minimum
                                     displayable rate, whichever is greater) and nominal descent
                                     rate;
                IF (vertical chase, low VMD geometry was not the reason for considering
                    reversal)
                    THEN  IF (intruder causing crossing OR (intruder level AND own crossing
                               from above) OR intruder rate and own modeled rate are opposite in sign)
                               THEN  use outer rate bound to model intruder;
                               ELSE  use inner rate bound to model intruder;
                    ELSE  use intruder's tracked vertical rate to model intruder;
                CALL MODEL_SEP
                    IN (delay, goal rate, own altitude, own rate, acceleration response, sense after
                        reversal, intruder altitude, modeled intruder rate, ITF entry)
                    OUT (predicted separation for sense reversal);

                IF (Predicted separation for sense reversal is not positive OR
                   modeled separation achieved by continuing current RA GE G.ALIM)
                    THEN  CLEAR reversal flag in ITF;
        <End own is assumed to follow its RA>

END Reversal_modeling;
```



```
PROCESS Reversal_modeling;

    NOMINAL_SEP = 0;
    Z = G.ZOWN;
    ZD = G.ZDOWN;
    DELAY = 0;

    IF (G.OWN_FOLLOW EQ FALSE)
        THEN  CALL MODEL_SEP
                  IN (DELAY, ZD, Z, ZD, P.VACCEL, OWNTENT(7), ITF.ZINT, ITF.ZDINT, ITF entry)
                  OUT (NOMINAL_SEP);
              IF (OWNTENT(7) EQ $TRUE)
                  THEN  NEW_SENSE = $FALSE;
                  ELSE  NEW_SENSE = $TRUE;
              IF (NOMINAL_SEP GT 1.2 * P.CROSSTHR)
                  THEN  CLEAR ITF.REVERSE;
        ELSE
        <Begin own is assumed to follow its RA>
            IF (OWNTENT(5,6) EQ '00')
                THEN  DELAY = MAX(P.TV1 - (G.TCUR - G.TPOSRA), 0);
                      IF (OWNTENT(7) EQ $FALSE)
                          THEN  ZDGOAL = MAX(MIN(G.ZDOWN, P.MAXDRATE), P.CLMRT);
                          ELSE  ZDGOAL = MIN(MAX(G.ZDOWN, P.MINDRATE), P.DESRT);
                      CALL PROJECT_VERTICAL_GIVEN_ZDGOAL
                          IN ((G.TCUR - G.TPOSRA), G.ZTV, G.ZDTV, ZDGOAL, P.TV1, P.VACCEL)
                          OUT (ZPROJ, ZDPROJ);
                      IF  (((OWNTENT(7) EQ $FALSE AND ZPROJ GT G.ZOWN AND
                          (G.ZDOWN GE G.ZDTV - P.MODEL_ZD)) OR
                          (OWNTENT(7) EQ $TRUE AND ZPROJ LT G.ZOWN AND
                          (G.ZDOWN LE G.ZDTV + P.MODEL_ZD))) AND
                          G.TCUR – G.TPOSRA LT P.MODEL_T)
                          THEN  Z = ZPROJ;
                                ZD = ZDPROJ;
                      CALL MODEL_SEP
                          IN (DELAY, ZDGOAL, Z, ZD, P.VACCEL, OWNTENT(7),
                              ITF.ZINT, ITF.ZDINT, ITF entry)
                          OUT (NOMINAL_SEP);
            IF (OWNTENT(7) EQ $TRUE)
                THEN  NEW_SENSE = $FALSE;
                ELSE  NEW_SENSE = $TRUE;
            DELAY = MAX(P.TV1 - (G.TCUR - G.TLASTNEWRA), P.QUIKREAC);

            IF (NEW_SENSE EQ $FALSE)
                THEN  ZDGOAL = MAX(P.CLMRT, MIN(G.ZDOWN, P.MAXDRATE));
                ELSE  IF (G.NODESCENT EQ $TRUE)
                          THEN  ZDGOAL = 0;
                          ELSE  ZDGOAL = MIN(P.DESRT, MAX(G.ZDOWN,            P.MINDRATE));
            IF (G.REV_CONSDRD EQ FALSE)
                THEN  IF ((ITF.INT_CROSS EQ $TRUE) OR (ITF.ZDINT EQ 0 AND
                          ITF.RZ GT 0) OR (ITF.ZDINT * G.ZDMODEL LT 0))
                          THEN  MZDINT = ITF.ZDOUTR;
                          ELSE  MZDINT = ITF.ZDINR;
                ELSE  MZDINT = ITF.ZDINT;

            CALL MODEL_SEP
                IN (DELAY, ZDGOAL, Z, ZD, P.RACCEL, NEW_SENSE, ITF.ZINT, MZDINT, ITF entry)
                OUT (ZMP);

            IF (ZMP LE 0 OR NOMINAL_SEP GE G.ALIM)
                THEN  CLEAR ITF.REVERSE;
        <End own is assumed to follow its RA>
END Reversal_modeling;
```

Why is it hard?

State Uncertainty	Dynamic Uncertainty	Multiple Objectives



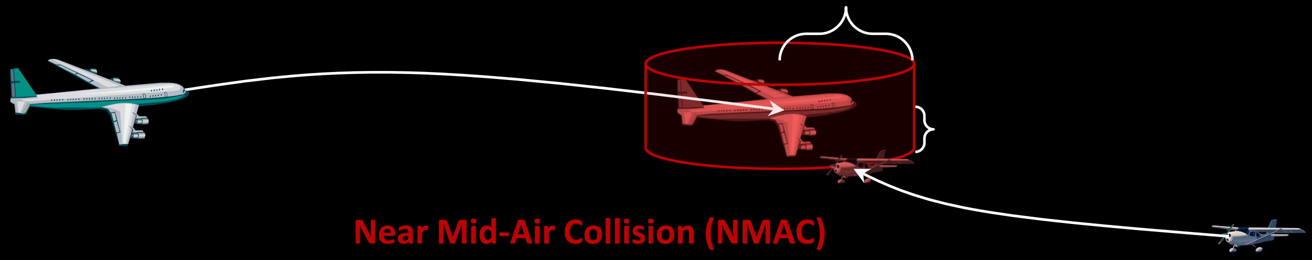
Slide Credit: Mykel Kochenderfer

Simplified MDP

State space	Action space
• Relative altitude • Own vertical rate • Intruder vertical rate • Time to lateral NMAC • State of advisory	• Clear of conflict • Climb > 1500 ft/min • Climb > 2500 ft/min • Descend > 1500 ft/min • Descend > 2500 ft/min
Dynamic model	**Reward model**
• Head-on, constant closure • Random vertical acceleration • Pilot response delay (5 s) • Pilot response strength (1/4 g) • State of advisory	• NMAC (-1) • Alert (-0.01) • Reversal (-0.01) • Strengthen (-0.009) • Clear of conflict (0.0001)

Slide Credit: Mykel Kochenderfer

Simplified MDP



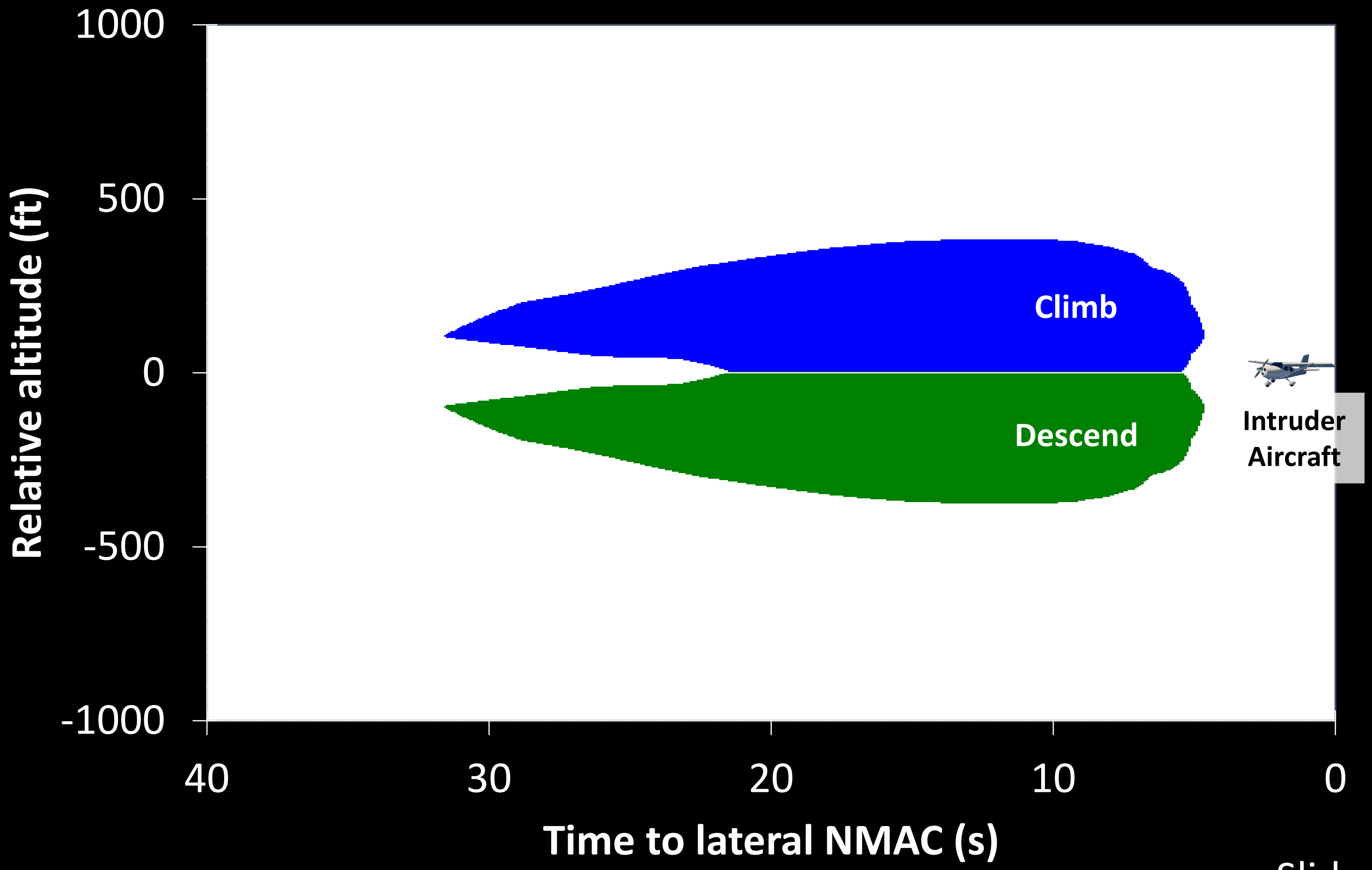
State space	Action space
• Relative altitude • Own vertical rate • Intruder vertical rate • Time to lateral NMAC • State of advisory	• Clear of conflict • Climb > 1500 ft/min • Climb > 2500 ft/min • Descend > 1500 ft/min • Descend > 2500 ft/min
Dynamic model	**Reward model**
• Head-on, constant closure • Random vertical acceleration • Pilot response delay (5 s) • Pilot response strength (1/4 g) • State of advisory	• NMAC (-1) • Alert (-0.01) • Reversal (-0.01) • Strengthen (-0.009) • Clear of conflict (0.0001)

Slide Credit: Mykel Kochenderfer

Optimized Logic

Both Own and Intruder Level

Metric	TCAS	ACAS X
NMACs	169	3
Alerts	994,317	690,406
Strengthens	40,470	92,946
Reversals	197,315	9,569



Slide Credit: Mykel Kochenderfer

Probabilities

1) conditional probability

$$P(A|B) = P(A,B)/P(B)$$

2) total probability

$$P(A) = \sum_{B \in \mathcal{B}} P(A|B)P(B)$$

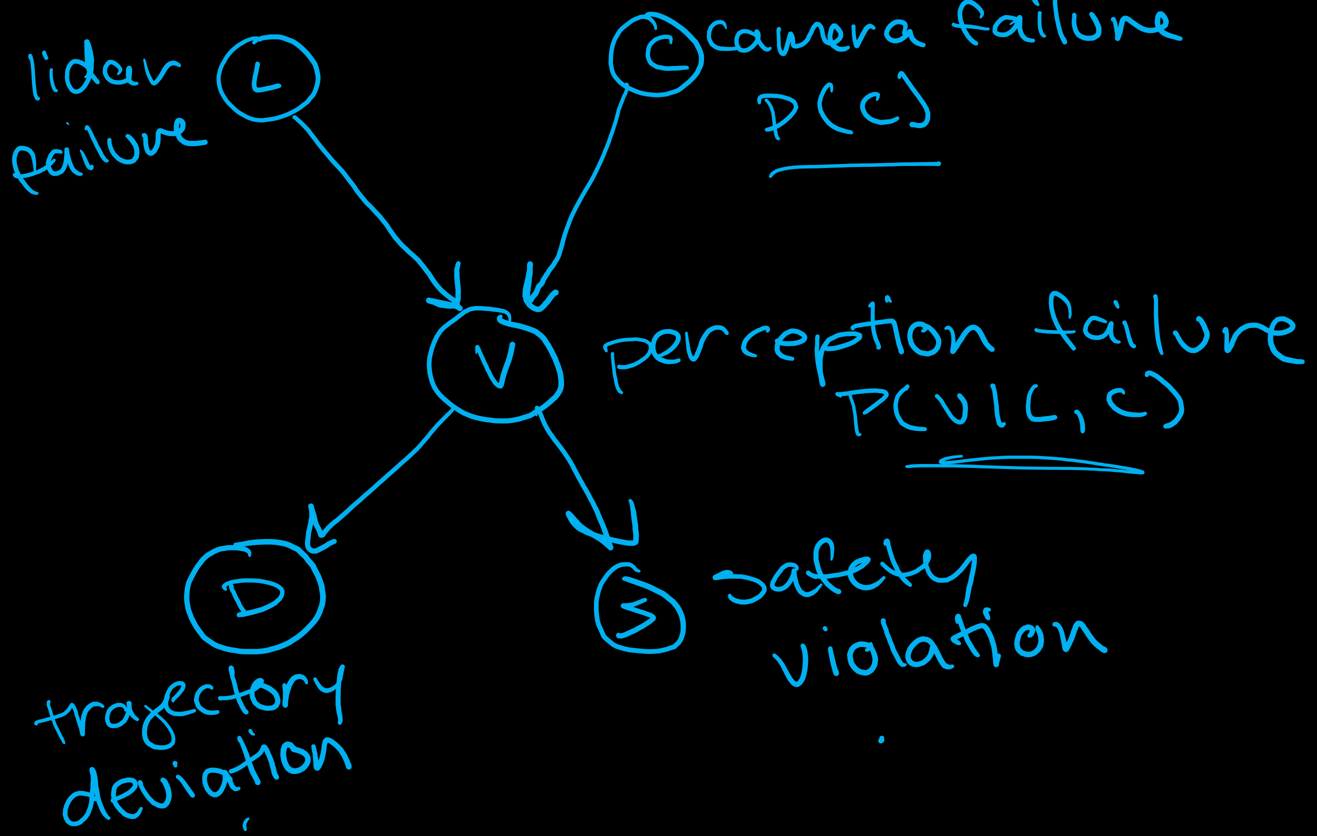
3) Bayes Rule

$$P(A|B) = P(B|A)P(A)/P(B)$$

A	B	C	P(A,B,C)
0	0	0	0.08
0	0	1	0.15
0	1	0	0.05
0	1	1	0.10
1	0	0	0.14
1	0	1	0.18
1	1	0	0.19
1	1	1	0.11

Bayesian Networks (1)

compact representations of a joint probability distribution

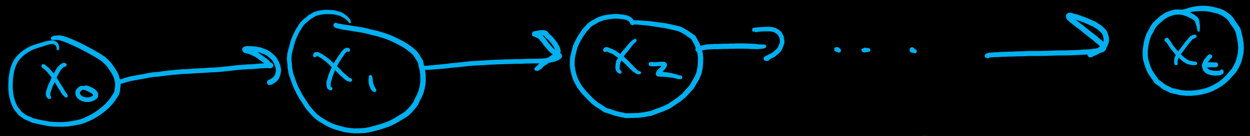


every node has a dist:

$P(X_i \mid pa_{X_i})$

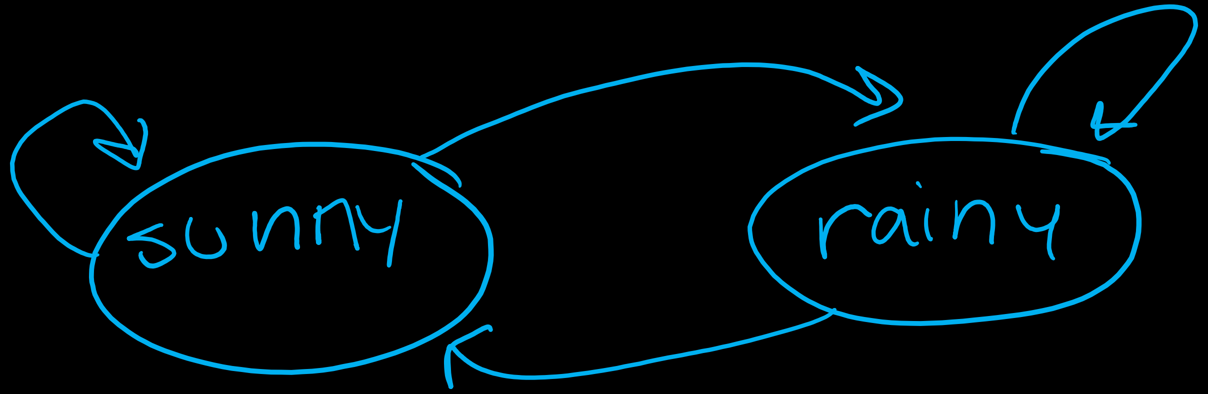
Continuous / Mixed Bayes Nets

Temporal Models: Markov Chains



X_t is the state of our system at time t

↳ pos & vel



x \ x'	S	R
S	.9	.1
R	.2	.8

Markov Assumptions and Common Violations

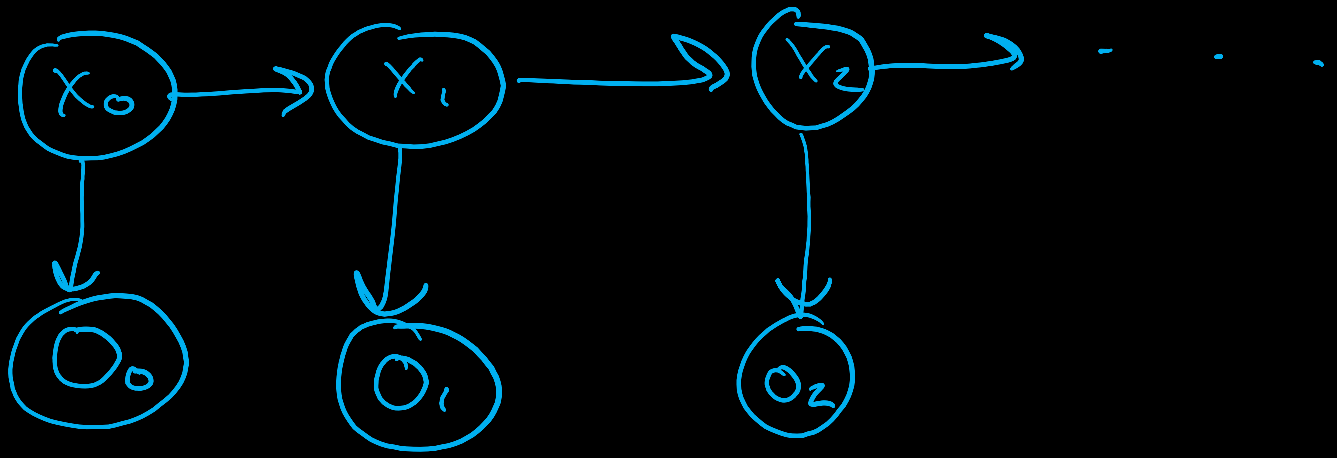
Markov Assumption postulates that past and future data are independent if you know the current state.

What are some common violations?

- Unmodeled dynamics in the environment not included in state
  - E.g., moving people and their effects on sensor measurements in localization
- Inaccuracies in the probabilistic model
  - E.g., error in the map of a localizing agent or incorrect model dynamics
- Approximation errors when using approximate representations
  - E.g., discretization errors from grids, Gaussian assumptions
- Variables in control scheme that influence multiple controls
  - E.g., the goal or target location will influence an entire sequence of control commands

Probabilistic Robotics, Section 2.4.4

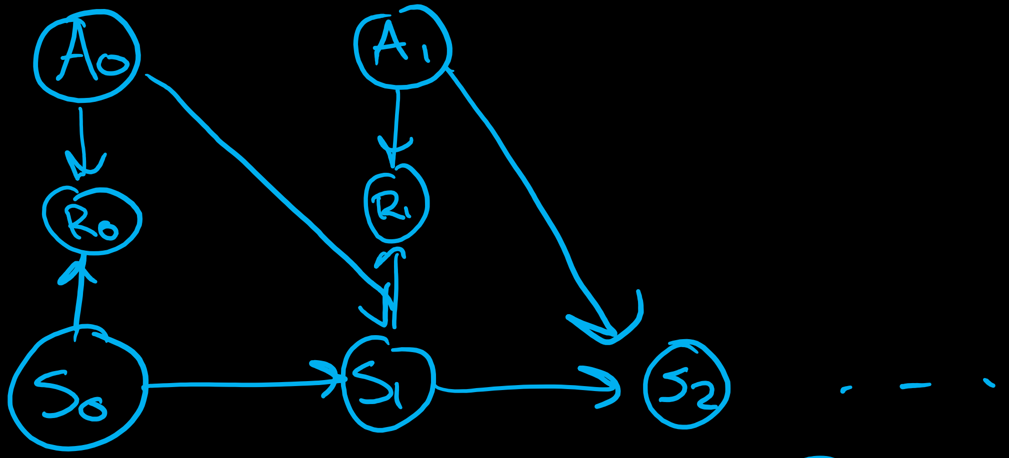
Hidden Markov Models



inference tools:

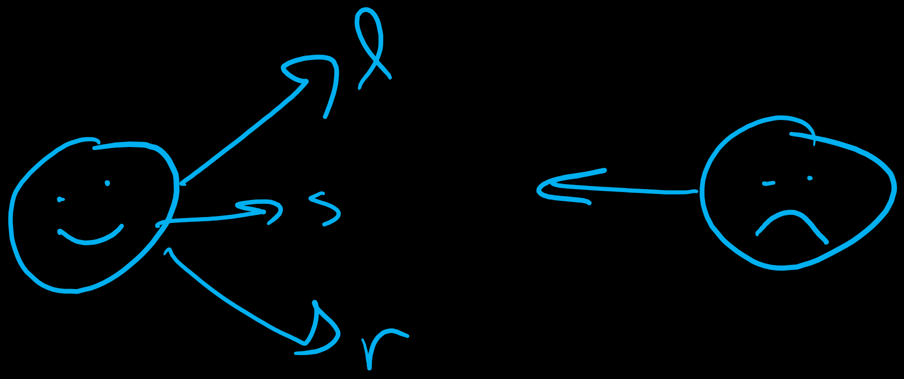
- filtering $P(X_k \mid O_{0:k})$
- prediction $P(X_k \mid O_{0:t})$, $k > t$
- smoothing $P(X_k \mid O_{0:t})$, $k < t$

Sequential Decision Making (1)



Markov Decision Process (S, A, T, R, γ)

$T(s'|s,a) + R(s,a)$



Utility and Reward

MDP rewards are generally additive components in a utility function:

for some finite horizon w/ n decisions the utility associated w/ the sequence of rewards is

$$\sum_{t=0}^{n-1} r_t$$

Policies and Utility

- policy π specifies what action to executed from every state

$$a_t \leftarrow \pi(s_t) \text{ or } \pi(o_t)$$

- Expected utility ~~of~~ from executing π when starting at state s is denoted $U^\pi(s)$
- optimal policy π^* maximized expected utility

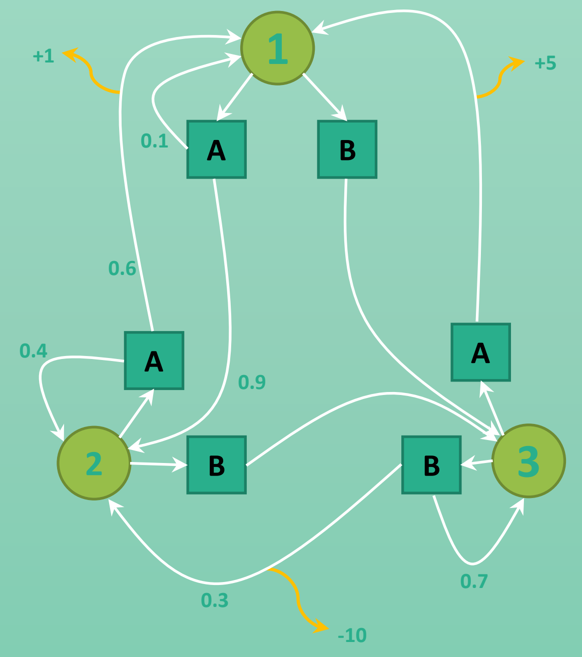
$$\pi^*(s) = \arg\max_\pi U^\pi(s)$$

Models of Optimal Behavior

- In the finite-horizon model, the agent should optimize expected reward for the next H steps: $E[\sum_{t=0}^{H} r_t]$
  - Continuously executing H-step optimal actions is known as receding horizon control
- In the infinite-horizon discounted model, agent should optimize: $E[\sum_{t=0}^{H} \gamma^t r_t]$
  - Discount factor is between 0 and 1, can be thought of as interest rate (reward now is worth more than reward later)
  - Keeps the utility of an infinite sequence finite

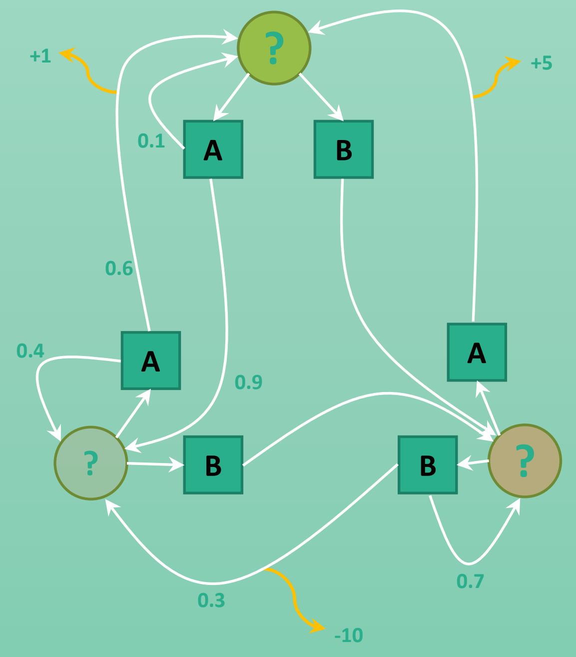
Markov Models

Markov Decision Processes (MDP)



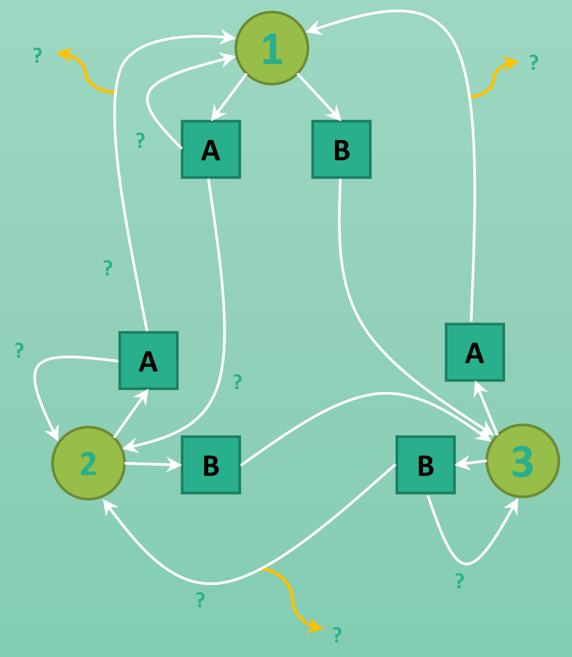
Uncertainty in effects of actions

Partially Observable MDP



Uncertainty in current state

Reinforcement Learning



Uncertainty in model