Chapter 17 2nd Part
Making Complex Decisions
--- Decision-theoretic Agent Design

Xin Lu

11/04/2002

POMDP: UNCERTAINTY

- Uncertainty about the action outcome
- Uncertainty about the world state due to imperfect (partial) information

  --- *Huang Hui*

Outline

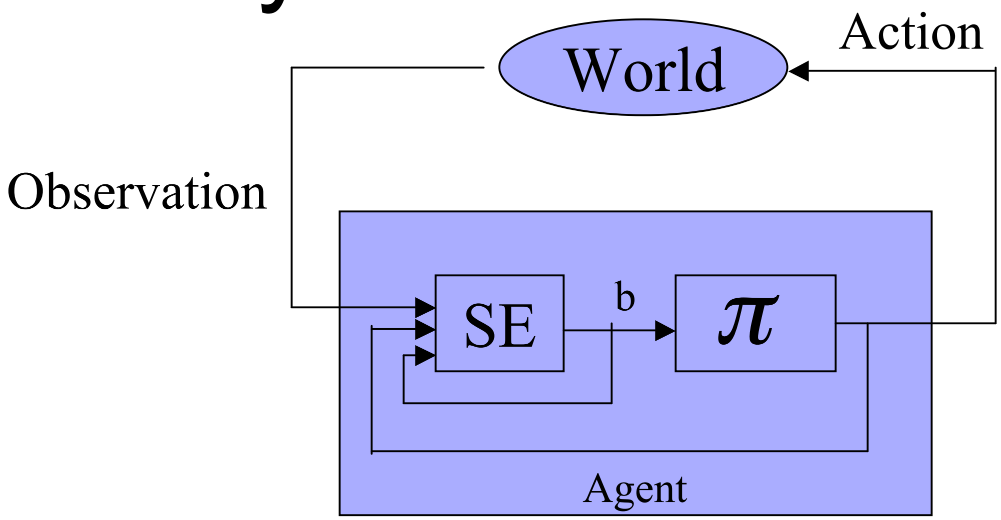
- POMDP agent

  Constructing a new MDP in which the current probability distribution over states plays the role of the state variable causes the state new state space characterized by real-valued probabilities and *infinite*.

- Decision-theoretic Agent Design for POMDP

  a limited lookahead using the technology of decision networks

Decision cycle of a POMDP agent



- Given the current belief state *b*, execute the action $a = \pi^*(b)$
- Receive observation *o*
- Set the current belief state to *SE(b,a,o)* and repeat.

Belief state

- *b(s)* is the probability assigned to the actual state *s* by belief state *b*.

0.111	0.111	0.111	<u>0.000</u>
0.111		0.111	<u>0.000</u>
0.111	0.111	0.111	0.111

$$\left(\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},0,0\right)$$

$$b'(s_j) = \mathrm{P}(s_j \mid o,a,b) = \frac{P(o \mid s_j,a)\sum_{s_i \in S} P(s_j \mid s_i,a)\, b(s_i)}{\sum_{s_j \in S} P(o \mid s_j,a)\sum_{s_i \in S} P(s_j \mid s_i,a)\, b(s_i)} \longrightarrow b' = SE(b,a,o)$$

Belief MDP

- A belief MDP is a tuple <B, A, ρ, P>:

  B = infinite set of belief states

  A = finite set of actions

  $\rho(b, a) = \sum_{s \in S} b(s) R(s, a)$ (reward function)

  $P(b'|b, a) = \sum_{o \in O} P(b'|b, a, o) P(o|a, b)$ (transition function)

Where P(b'|b, a, o) = 1 if SE(b, a, o) = b',P(b'|b, a, o) = 0 otherwise;

Move West once

b

0.111	0.111	0.111	0.000
0.111		0.111	0.000
0.111	0.111	0.111	0.111

b'

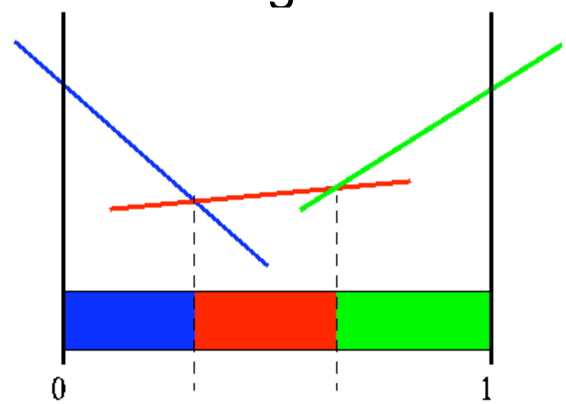
0.222	0.111	0.000	0.000
0.111		0.111	0.000
0.222	0.111	0.111	0.000

Solutions for POMDP

- Belief MDP has reduced POMDP to MDP, the MDP obtained has a continuous state space.
- Methods based on *value* and *policy iteration*:

  A policy $\pi(b)$ can be represented as a set of *regions* of belief state space, each of which is associated with a particular optimal action. The value function associates a distinct *linear* function of *b* with each region. Each value or policy iteration step refines the boundaries of the regions and may introduce new regions.
- A Method based on lookahead search:

  decision-theoretic agents



Decision Theory

= probability theory + utility theory

The fundamental idea of decision theory is that an agent is rational if and only if it chooses the action that yields the highest expected utility, averaged over all possible outcomes of the action.

A decision-theoretic agent

function DECISION-THEORETIC-AGENT(*percept*) returns *action*

calculate updated <u>probabilities for current state</u> based on available evidence including current percept and previous action

calculate outcome probabilities for actions

given action descriptions and probabilities of current states

select *action* with highest expected utility

given probabilities of outcomes and utility information

return *action*

Basic elements of decision-theoretic agent design

- Dynamic belief network--- the transition and observation models
- Dynamic decision network (DDN)--- decision and utility
- A filtering algorithm (e.g. Kalman filtering)---incorporate each new percept and action and update the belief state representation.
- Decisions are made by projecting forward possible action sequences and choosing the best action sequence.

Definition of Belief

- The belief about the state at time *t* is the probability distribution over the state given all available evidence:

$$Bel(X_t) = P(X_t \mid E_1 \ldots E_t, A_1 \ldots A_{t-1}) \qquad (1)$$

X_t is *state variable*, refers the current state of the world

E_t is evidence variable.

Calculation of Belief (1)

- Assumption 1: the problem is *Markovian,*

$$P(X_t \mid X_1 \ldots X_{t-1}, A_1 \ldots A_{t-1}) = P(X_t \mid X_{t-1}, A_{t-1}) \quad (2)$$

- Assumption 2: each percept depends only on the state at the time

$$P(E_t \mid X_1 \ldots X_t, A_1 \ldots A_{t-1}, E_1 \ldots E_{t-1}) = P(E_t \mid X_t) \qquad (3)$$

- Assumption 3: the action taken depends only on the percepts the agent has received to date

$$P(A_{t-1} \mid A_1 \ldots A_{t-2}, E_1 \ldots E_{t-1}) = P(A_{t-1} \mid E_1 \ldots E_{t-1}) \quad (4)$$

Calculation of Belief (2)

- Prediction phase:

$$\hat{Bel}(X_t) = \sum_{X_{t-1}} P(X_t \mid X_{t-1} = x_{t-1}, A_{t-1}) Bel(X_{t-1} = x_{t-1}) \quad (5)$$

  x_{t-1} ranges over all possible values of the state variables X_{t-1}

- Estimation phase:

$$Bel(X_t) = \alpha P(E_t \mid X_t) \hat{Bel}(X_t) \quad (6)$$

  α is a normalization constant

Design for a decision-theoretic Agent

Function DECISION-THEORETIC-AGENT(E_t) returns an action

inputs: E_t, the percept at time *t*

static: *BN,* a belief network with nodes X

Bel(X), a vector of probabilities, updated over time

$$\hat{Bel}(X_t) \leftarrow \sum_{X_{t-1}} P(X_t \mid X_{t-1} = x_{t-1}, A_{t-1}) Bel(X_{t-1} = x_{t-1})$$

$$Bel(X_t) \leftarrow \alpha P(E_t \mid X_t)\, \hat{Bel}(X_t)$$

$$action \leftarrow \arg\max_{A_t} \sum_{X_t} [Bel(X_t = x_t) \sum_{X_{t+1}} P(X_{t+1} = x_{t+1} \mid X_t = x_t, A_t) U(x_{t+1})]$$

return *action*

Sensing in uncertain worlds

- Sensor model: $P(E_t \mid X_t)$, describes how the environment generates the sensor data.

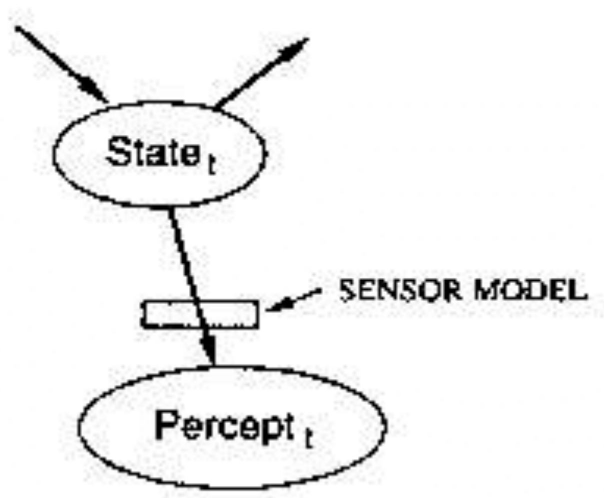
  vs Observation model *O(s,o)*
- Action model: $P(X_t \mid X_{t-1}, A_{t-1})$, describes the effects of actions

  vs Transition model $T(s,a,s')$
- Stationary sensor model: $\forall t \quad P(E_t \mid X_t) = P(E \mid X)$

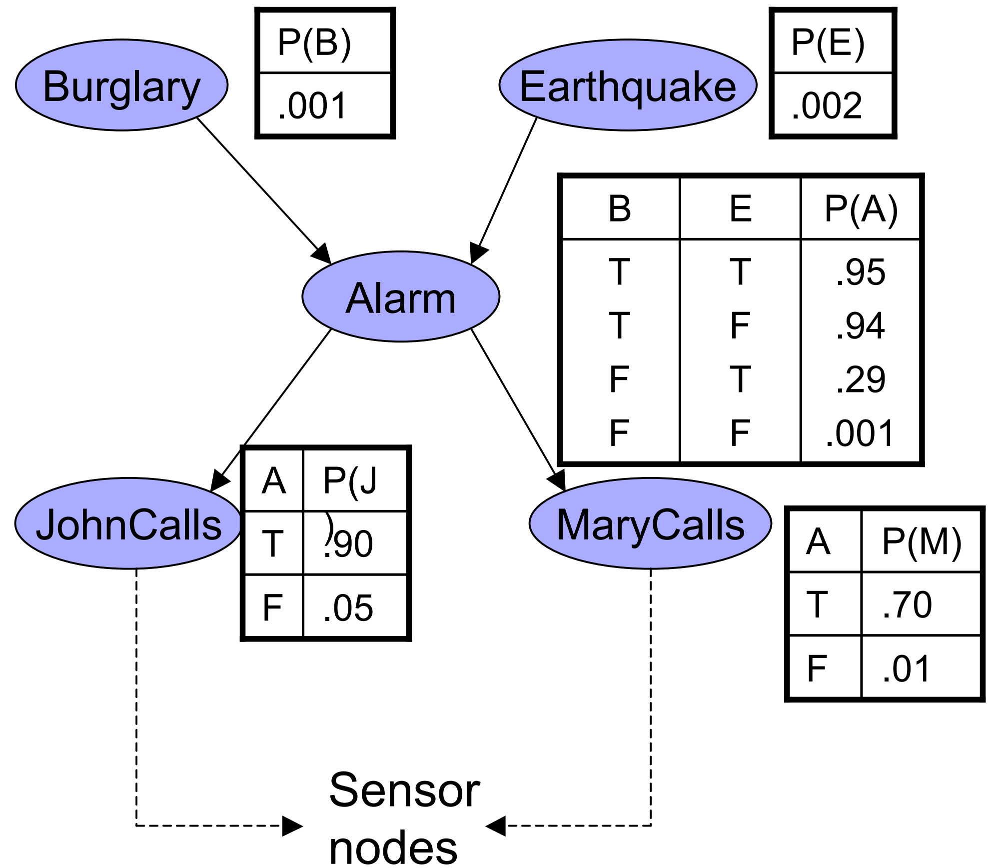
  where E and X are random variables ranging over percepts and states

Advantage: $P(E \mid X)$ *can be used at each time step.*

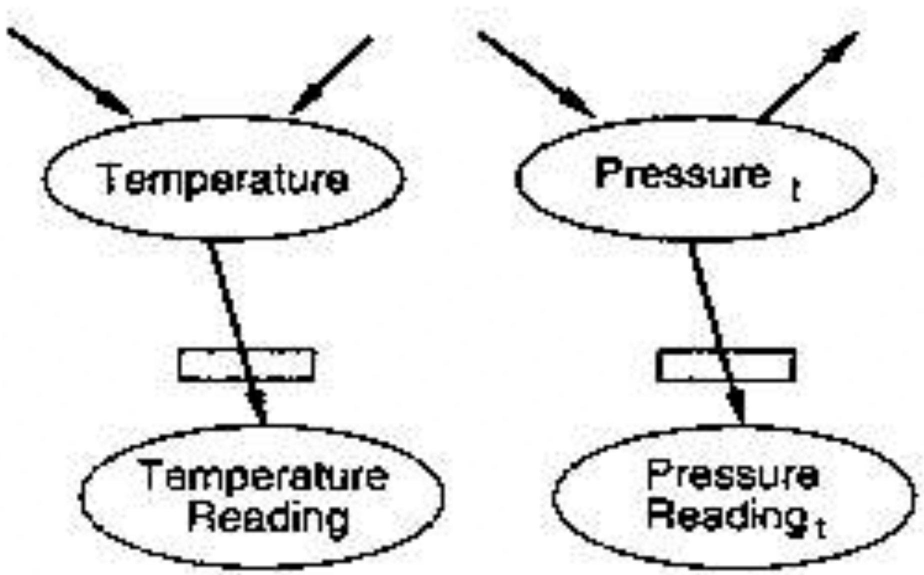
A sensor model in a belief network



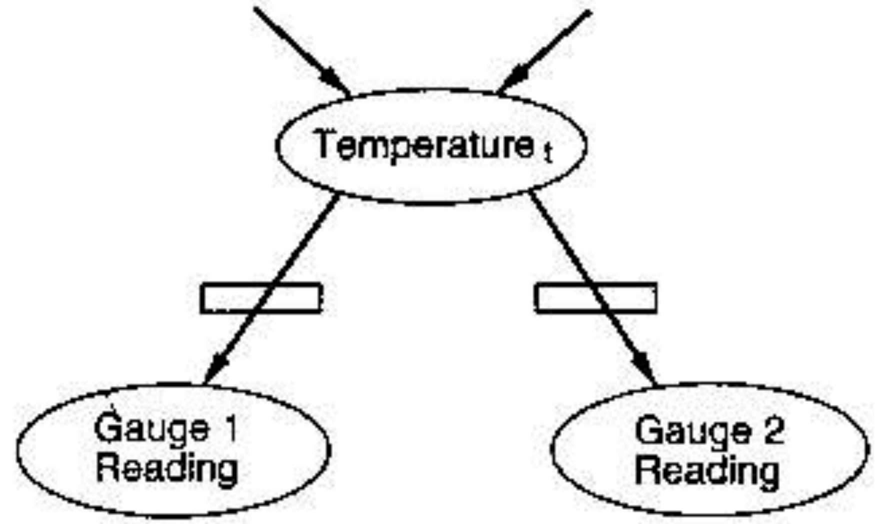
(a) Belief network fragment showing the general relationship between state variables and sensor variables.



Next step: break apart the generalized state and sensor variables into their components.



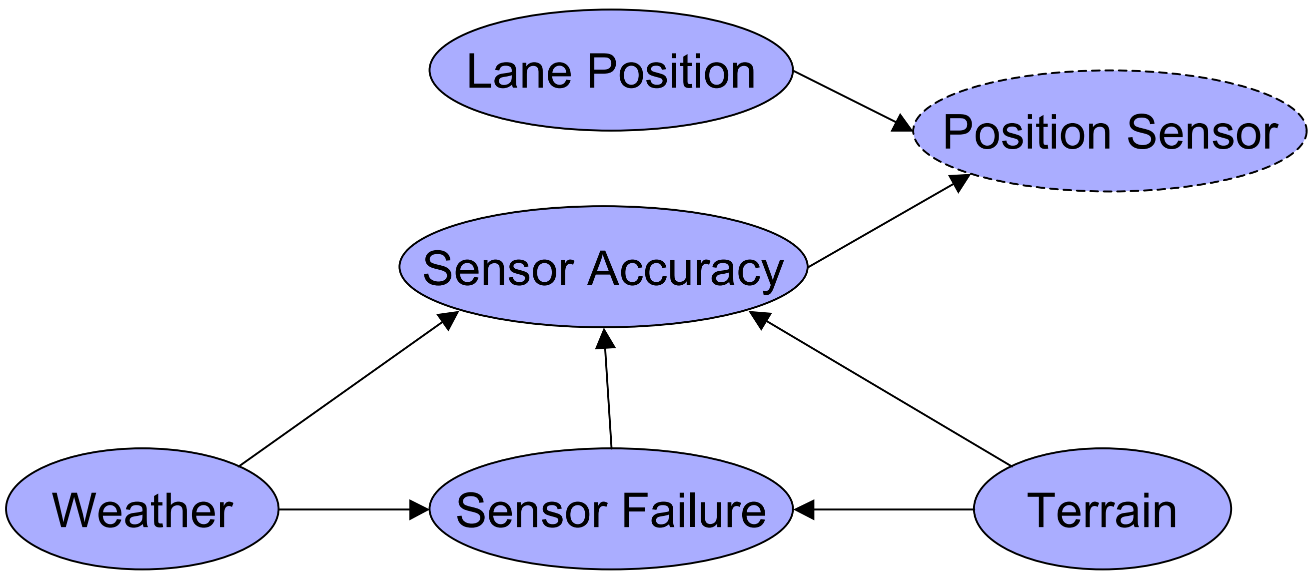
(b) An example with pressure and temperature gauges



(c) Measuring temperature using two separate gauges

Sensor Failure

- In order for the system to handle sensor failure, the sensor model must include the possibility of failure.



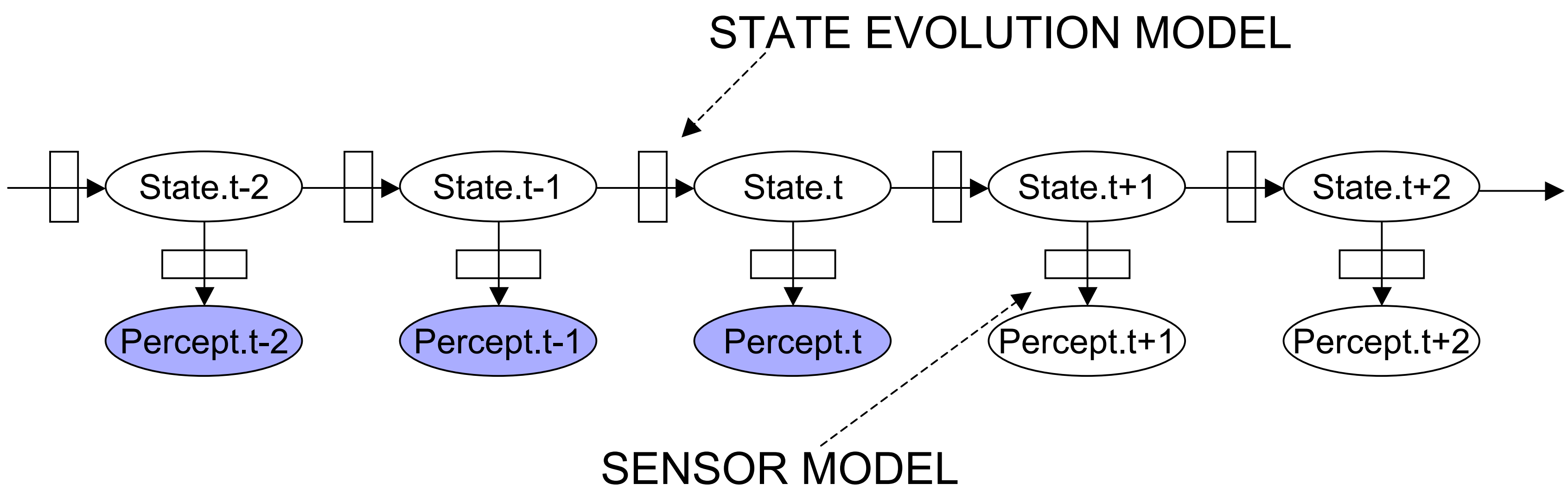
Dynamic Belief Network

- Markov chain (state evolution model):

  a sequence of X_t values where each one is determined solely by the previous one: $P(X_t \mid X_{t-1})$

- Dynamic belief network (DBN): a belief network with one node for each state and sensor variable for each time step.

Generic structure of a dynamic belief network

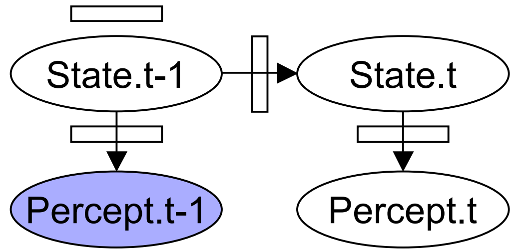


Two tasks of the network:

- Calculate the probability distribution for state at time *t*
- Probabilistic projection: concern with how the state will evolve into the future

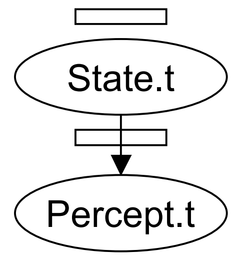
- **Prediction:**

$$\hat{Bel}(X_t) = \sum_{X_{t-1}} P(X_t \mid X_{t-1} = x_{t-1}, A_{t-1}) Bel(X_{t-1} = x_{t-1})$$



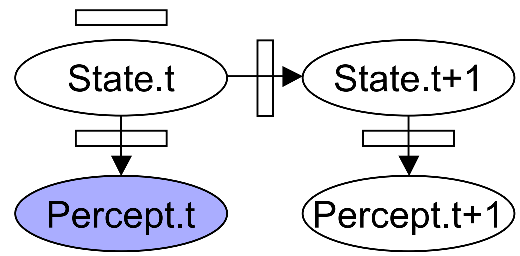
- **Rollup:**

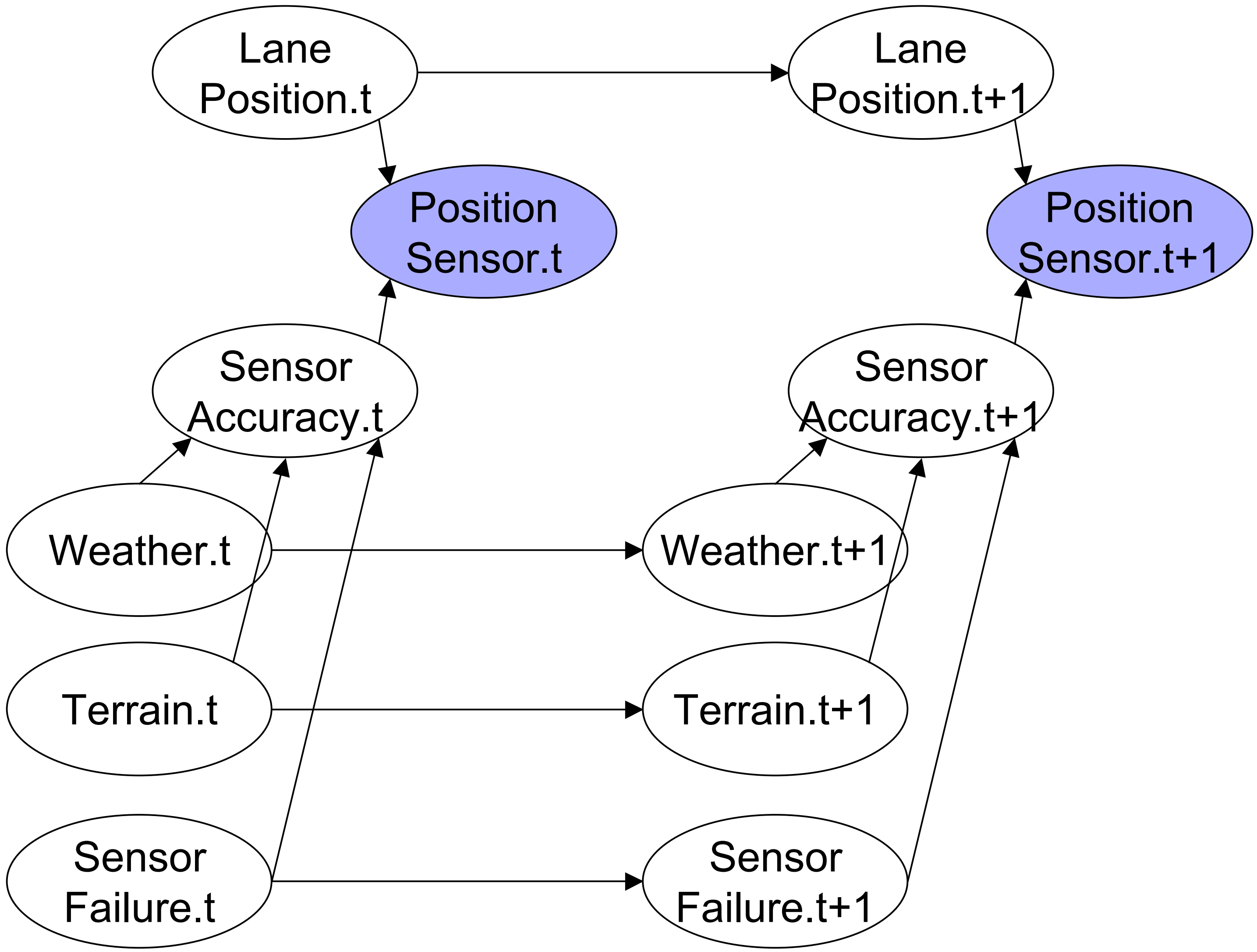
remove slice *t-1*



- **Estimation:**

$$Bel(X_t) = \alpha P(E_t \mid X_t) \hat{Bel}(X_t)$$

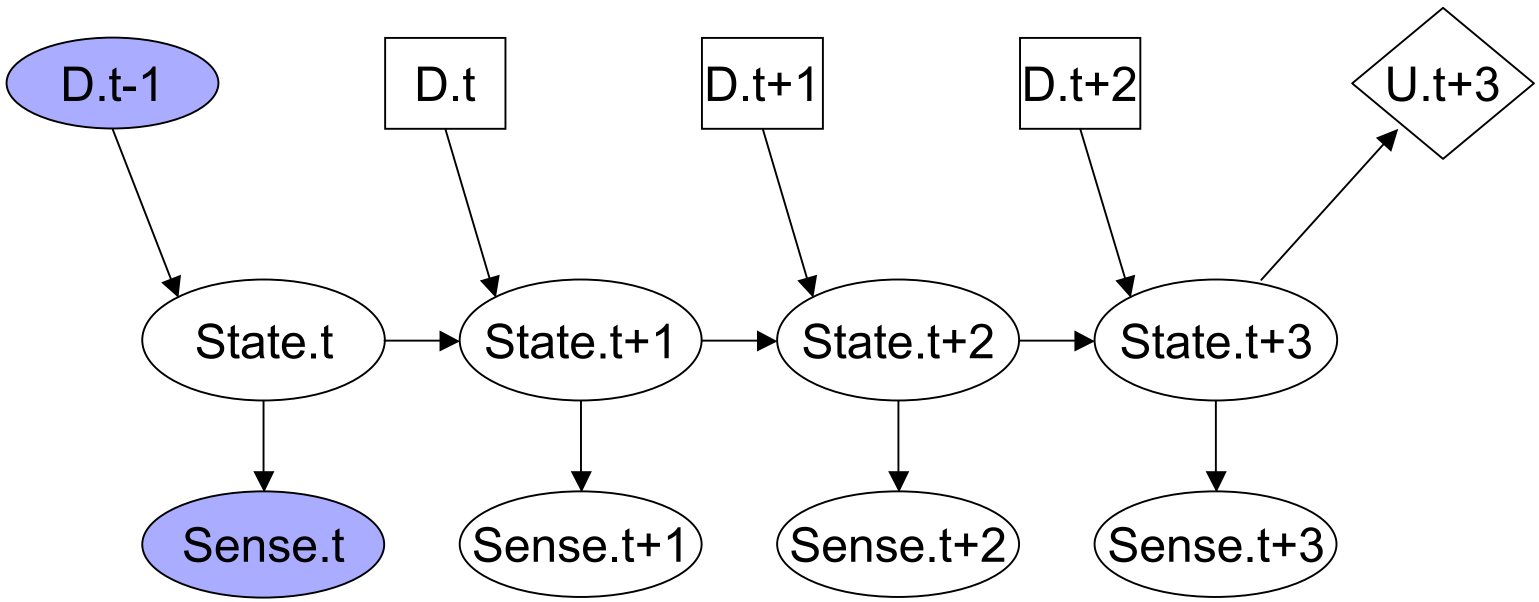



Lane Position.t
Lane Position.t+1
Position Sensor.t
Position Sensor.t+1
Sensor Accuracy.t
Sensor Accuracy.t+1
Weather.t
Weather.t+1
Terrain.t
Terrain.t+1
Sensor Failure.t
Sensor Failure.t+1

Dynamic Decision Networks

- Dynamic Decision Networks: add utility nodes and decision nodes for actions into dynamic belief networks.

The generic structure of a dynamic decision network



- The decision problem involves calculating the value of D_t that maximizes the agent's expected utility over the remaining state sequence.

Search tree of the lookahead DDN

D_t in $P(X_t \mid E_{1:t})$

E_{t+1}

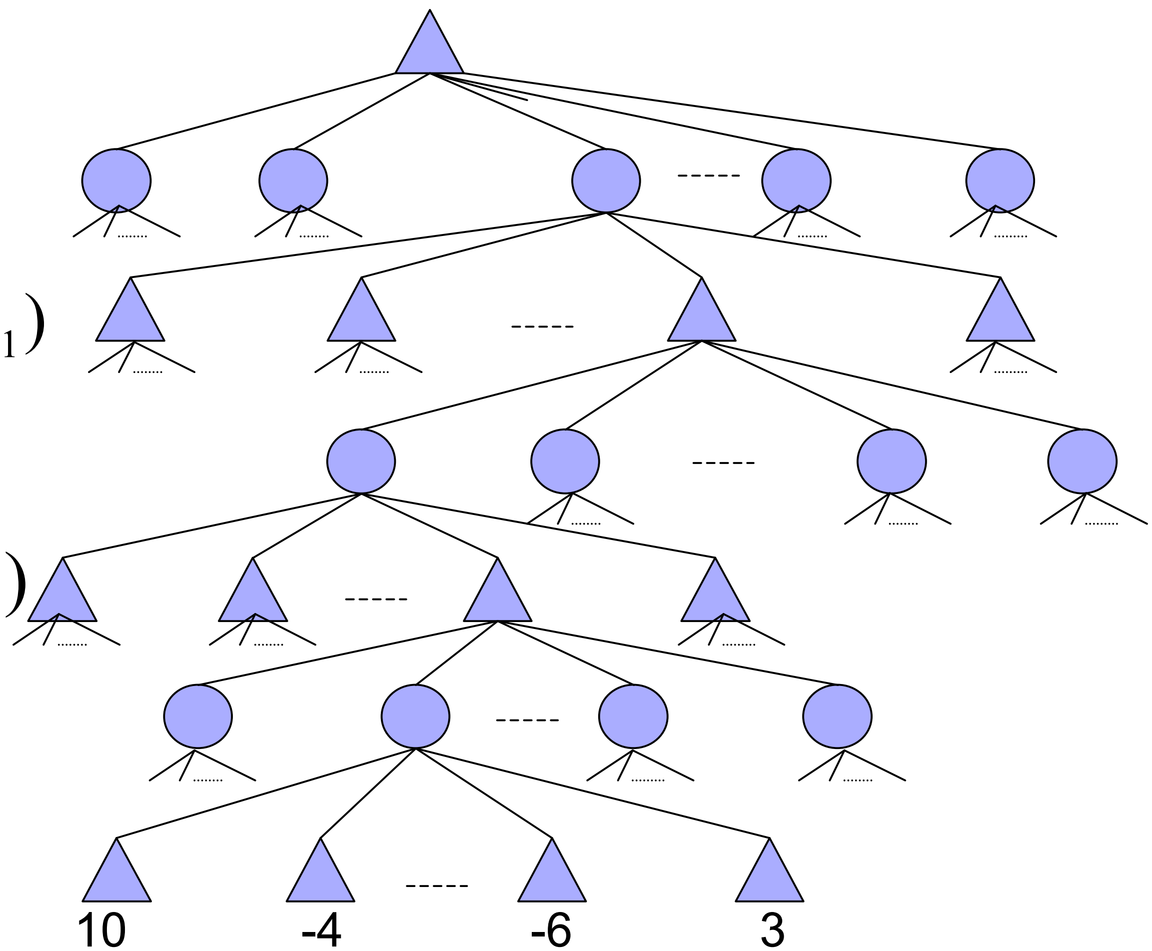
D_{t+1} in $P(X_{t+1} \mid E_{1:t+1})$

E_{t+2}

D_{t+2} in $P(X_{t+2} \mid E_{1:t+2})$

E_{t+3}

$U(X_{t+3})$



Some characters of DDN search tree

- The search tree of DDN is very similar to the EXPECTIMINIMAX algorithm for game trees with chance nodes, expect that:
  - There can also be rewards at non-leaf states
  - The decision nodes correspond to belief states rather than actual states.
- The time complexity: $O(|D|^d \bullet |E|^d)$

  d is the depth, |*D*| is the number of available actions, |*E*| is the number of possible observations

Discussion of DDN

- The DDN promises potential solutions to many of the problems that arise as AI systems are moved from ***static***, ***accessible***, and above all ***simple*** environments to ***dynamic***, ***inaccessible***, ***complex*** environments that are closer to the real world.
- The DDN provides a ***general***, ***concise representation*** for large POMDP, so they can be used as inputs for any POMDP algorithm including value and policy iteration methods.

Perspective of DDN to reduce complexity

- Combined with a heuristic estimate for utility of the remaining steps
- Many approximation techniques:
  - Using less detailed state variables for states in the distant future.
  - Using a greedy heuristic search through the space of decision sequences.
  - Assuming “most likely” values for future percept sequences rather than considering all possible values

…