01DRO1
Decision Making under Uncertainty

2017/2018

Bayesian Learning.
Introduction to Markov Decision Processes.

Lecture 3

6.3.2018

Readings:

- Martin Puterman, *Markov decision processes*, John Wiley & Sons, 1994
- D.P. Bertsekas, *Dynamic Programming*, Prentice Hall, 1987
- L.P.Kaelbling, M. L. Littman, A.R.Cassandra, Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101,99–134, 1998.
- K.J. Aström, Optimal control of Markov decision processes with incomplete state information, J. Math. Anal. Appl. 10, 174–205, 1985.

Announcements:

Mid-term assignment – Apr 3 (Lectures 1-6)

Homework topic approval due March 15.

1.5 and 8.5 are holidays – no class

Possible topics for coursework on 01DRO1

‘Research’ project (examples of possible topics)

1. Dynamic programming for MDP with two-dimensional action, whose entries exploit different knowledge.
2. Design of on/off stabilization of a room temperature.
3. Design of target-tracking policy based on MDP for known linear-Gaussian environment model and quadratic reward.
4. Formulation and solution of sequential estimation as MDP. Inspection of influence of measurement cost.
5. Bayesian estimation of unobserved state in Markov model with linear Gaussian state evolution.

… any topic related to your BSc, Mgr or research project

Critical literature survey (examples of possible topics)

1. Survey of knowledge elicitation techniques.
2. Survey of preference elicitation techniques.
3. Survey of knowledge transfer methodologies.
4. Survey of state of the art in Bayesian networks.
5. Survey of possible multi-agents’ scenario and interactions.

Implementation of existing approach (examples of possible topics)

1. Implementation of MDP with known envir. model,discrete states and actions. Inspecting influence of decision horizon.
2. Implementation of discounted MDP with known envir. model, discrete states and actions. Inspection of influence of discounted factor.
3. Implementation of MDP for known linear-Gaussian environment model and quadratic reward. Inspecting influence of decision horizon.
4. Implementation of discounted MDP for known linear-Gaussian environment model. Inspection of influence of discounted factor.

Where are we?

Last time.. Decision theory = probability theory + utility theory

Probability theory deals with degrees of belief (about env. states, action effects,..)

Utility theory is used to represent and reason about DM preferences.

An agent is *rational* iff it chooses the action yielding the *highest* expected utility, averaged over all *possible* outcomes.

The Maximum Expected Utility (MEU) principle:

$$EU(s,a,K) = \Sigma_i\, P(result_i(a)|a,s)\; U(result_i(a),a,K),$$

$$a^{best} = \mathrm{arg}max_a\, EU(s,a,K),$$ where

$result_i(a)$ is i-th possible outcome of action a

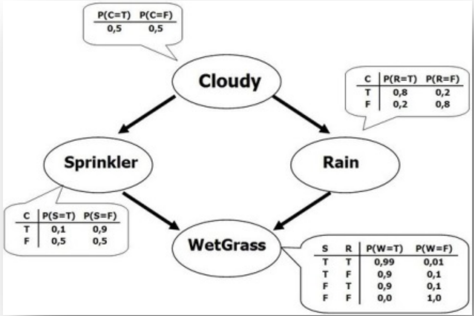
K is other knowledge available for choice a

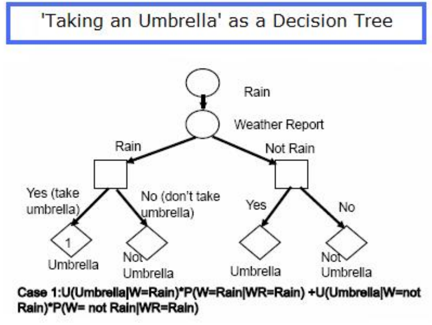
Last time.. BN, DT, DN

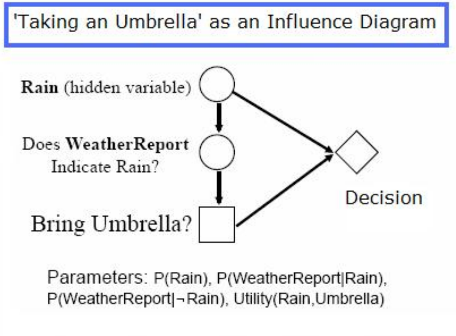
- Bayesian nets (BN) represent dependencies over a set of random vars.
- Decision trees (DT) represent all possible decision sequences.
- Decision nets (DN) represent a finite sequential decision problem. DN extend BN to include *decision variables* (actions) and utility.

- DT ☺ : clear ☹: multiple paths/nodes; large memory.
- DN ☺: concise and clear.

  ☹: assume memorizing all past observations/ decisions(actions)







Last time.. Preferences and their ordering

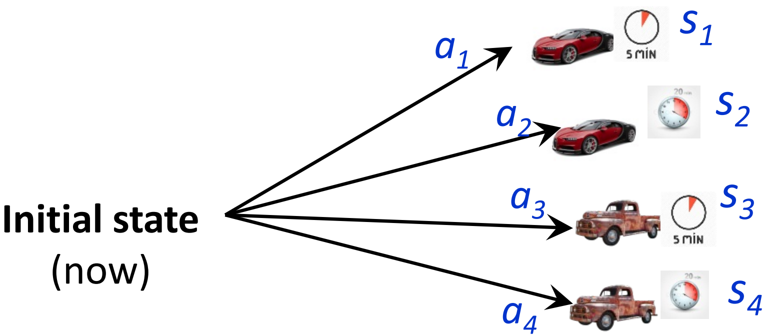
To perform any DM you need preference ordering over states/actions:

- $x \succ y$: x is strictly preferred over y
- $x \succcurlyeq y$: x is preferred over y (incl. indifference x and y)

Ordering must be *transitive* and $(x \succcurlyeq y \wedge y \succcurlyeq x) \Rightarrow x = y$; $(x \succ y \wedge y \succ x) \Rightarrow \emptyset$ *

For e.g: for agent $s_1 \succ s_2 \succ s_3 \succ s_4$;

for taxi service (possible loss, if a selection is not accepted) $a_4 \succ a_2 \succ a_3 \succ a_1$



But if *there is*:

- uncertainty in the next state (environ.)
- stochastic actions (agent)
- uncertainty about initial state (environ.)

Then no deterministic solution exists!

* *This is only part of constrains on preference ordering. Utility theory axioms require other constrains.*

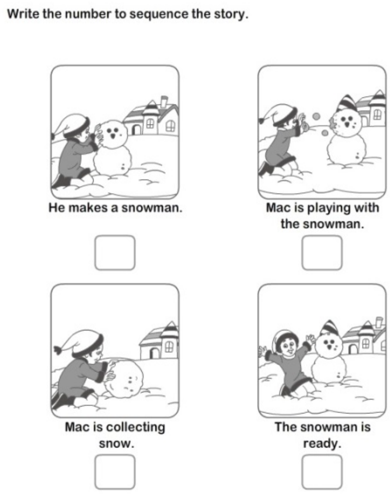
Last time.. Evaluation of DM Policy

sequence of actions (plan) ≠ policy

Plan: a_1, a_2, a_3, a_4

Policy: if *state* = s_i then action a_i else a_k

- *Policy* assigns a decision (action) to *each* reachable state.
- Policy can generate *more* state trajectories than plan.
- Do *not* compare values at states but policies
- *Indistinguishable* policies: policies differing at a number of unreachable decision nodes (states)



Today..

Learning

Different fields gave birth to similar ideas and (with different emphasize)

- Computer Science: AI, computer vision, information retrieval, ...
- Statistics: learning theory, learning and inference from data, ...
- Economics: game theory, operational research, decision theory,...
- Psychology: perception, reinforcement learning,...
- Computational Neuroscience: neuronal networks,..
- Engineering: signal processing, system identification, adaptive and optimal control, information theory, robotics, ...
- others..

DM and learning as inference

Inference task: given the knowledge (observation) what is the implication on non-observed variable?

You have (any of):
knowledge about the environment, observations; assumed type of model; prior probability over model parameters, etc.

You can obtain decision about:

- optimal actions to influence environment (*knowledge& prefer. elicitation*)
- a particular model (model structure *estimation*)
- posterior on model parameter (*identification*)
- data classification (*non-typical behaviour*, fault, malware *detection*)
- future data (*prediction*)

Inference

How can we reason about the environment from *observations*?

Types of variables considered:

- observations, measurement, any evidence
- unknown parameters
- auxiliary variables, noises

Given the observations and prior probabilities, what are the probabilities of the unknown parameters?

Basics of Bayesian learning

- $P(\theta)$ - the *prior* probability of a parameter $\theta \in \theta^*$

  Reflects background knowledge; before data is observed.

  If no information available, use uniform distribution.

- $P(d)$ - the probability that data d is observed (no knowledge of the parameter $\theta \in \theta^*$ is at disposal) - *evidence*
- $P(d|\theta)$ – *likelihood* of the data, i.e. the probability of observing the data d, given parameter θ
- $P(\theta|d)$- the *posterior* probability of θ. The probability of θ given that d has been observed.

  Bayes rule: $P(\theta|d) = P(d|\theta)\, P(\theta) / P(d)$

here $P(d)$ serves as scaling factor that guarantees the posterior is sum up to 1

Bayesian learning

Addresses problem of inverse probabilities: knowing the conditional probability (cp) of x given y, compute cp of y given x

Example: 70% AMSM students attend DROS while only 25% of MI students attend DROS. Given a random student that attends DROS is he/she from AMSM? (59% of all students are from AMSM, the rest from MI).

Frequentist probabilities are defined as limit of infinite number of trials

(0.05% of banknotes in circulation are fakes)

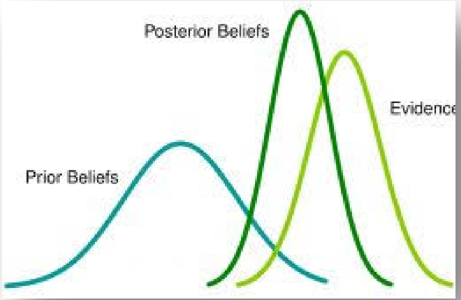
Bayesian (subjective) probabilities quantify degrees of belief based on experience

(what is the probability belief that my 100Kc is fake?)

Recap: Bayesian Learning

Given:

data observed $d=\{x_1,\ldots,x_{n-1}\}$,
parameterised model $p(x_i|x_{i-1}, \theta)$ with parameter $\theta\in\theta^*$,
likelihood model $p(d|\theta)$, evidence $p(d)$,
Find: $p(x_i|x_{i-1})$ – prediction model



- if d is Markov $p(d|\theta)= \Pi_{i=1}p(x_i|x_{i-1}, \theta)$ - *chain rule, Markov assumption*
- posterior probability of model parameters $p(\theta|d)= p(d|\theta)\, p(\theta)/p(d)$ *Bayes rule*
- prediction model of interest

$$p(x_n|x_{n-1}, d)=\int_{\theta\in\theta^*} p(x_n|x_{n-1}, \theta, d)\, p(\theta|d)\, d\theta$$

more details about learning, see 01HBM

Learning: testing hypotheses

The coin can be *fair* or *biased* 55% in favor of tails. Find bias of the coin.
H-heads, T-tails.

Hypothesis		Prior
h1: "fair"	$p(H)=0.5;\ p(T)=0.5$	$p(h1)=0.9$
h2: "biased"	$p(H)=0.45;\ p(T)=0.55$	$p(h2)=0.1$

Observed data: $d=\{H\}$
Goal: find *the most probable hypothesis* given the observed (or training) data

$$p(d='H')=p(d|h1)p(h1)+p(d|h2)p(h2)=0.5*0.9+0.45*0.1=0.545$$

Testing hypotheses (learning):

$p(h1|d)= p(d|h1)p(h1)/p(d)=0.5*0.9/0.545\approx0.826$ - *more likely variant*

$p(h2|d)= p(d|h2)p(h2)/p(d)=0.45*0.1/0.545\approx0.082$

Learning: testing hypotheses

- Maximum A Posteriori hypothesis $h_{MAP} = \operatorname{argmax}_{h=\{h1,h2\}} p(h|d)$ was used and after one data (head) the coin is more likely to be fair.

  MAP assumes a prior over the hypotheses $p(h)$ and finds hypothesis maximising the *posterior* $p(h|d)$
- if priors were the same, one can use Maximum Likelihood hypothesis

  $h_{ML} = \operatorname{argmax}_{h=\{h1,h2\}} p(d|h)$
- ML does not assume a prior over the hypotheses and finds hypothesis maximising the *likelihood* of the data $p(d|h)$. It coincides with MAP for uniform prior.

Task to think:

Modify priors above, compare ML and MAP for 100 tosses and 77 tails.

Which method is more consistent with data observed?

Bernoulli distribution

$x \in \{0, 1\}$ $\quad$ $dom(x) = \{0, 1\}$

$p(x=1|\theta) = \theta$ $\quad$ $p(x=0|\theta) = 1-\theta$

$Bern(x|\theta) = \theta^x (1-\theta)^{1-x}$

Given data set $\quad D = \{x_1, \dots, x_n\}$, $x_i \in \{0, 1\}$

if $x_i \sim Bern(x_i|\theta)$ then

$$p(D|\theta) = \prod_{i=1}^{n} Bern(x_i|\theta) = \prod_{i=1}^{n} \theta^{x_i} (1-\theta)^{1-x_i}$$

→ special case of binomial distribution

→ can be used to represent random variables taking the values $\{0, 1\}$ (yes/no-question).

Inference example: Bayesian Learning the coin

The coin is either *fair* or *biased* 55% in favor of tails.

Assume:

- possible observations x = {'Tail', 'Head'} = {T,H}
- data set observed D = $\{x_1,\dots,x_n\}$, m tails
- i.i.d data $x_i \sim Bern(x_i \mid \theta)$ with $\theta \in \theta^* = \{0.5; 0.55\}$
  $p(x = \text{'T'} \mid \theta) = \theta$; $p(x = \text{'H'} \mid \theta) = 1-\theta$

Task: learn parameter $\theta \in \theta^*$.

Bayesian Learning assumes a prior over the model parameters $\theta \in \theta^*$ and computes the posterior distribution of the parameters: $P(\theta \mid D)$.

Learning the coin (cont.)

Likelihood: $p(x|\theta, D) = p(x|\theta) = \theta^{\delta_{x,T}}(1-\theta)^{\delta_{x,H}}$, where $\delta_{x,y} = \begin{cases} 0 & \text{if } x \neq y, \\ 1 & \text{if } x = y. \end{cases}$

Probability of data observed:

$$p(x_1 \dots x_n|\theta) = \prod_{i=1}^{n} p(x_i|x_{i-1},\theta) = \prod_{i=1}^{n} p(x_i|\theta) = \prod_{i=1}^{n} \theta^{\delta_{x_i,T}}(1-\theta)^{\delta_{x_i,H}}$$

$$= \theta^{\overbrace{\sum_{i=1} \delta_{x_i,T}}^{\text{NUMBER of 'T'}}} (1-\theta)^{\overbrace{\sum_{i=1} \delta_{x_i,H}}^{\text{NUMBER of 'H'}}}$$

Posterior probability:

$$p(\theta|x_1 \dots x_n) = \frac{p(x_1 \dots x_n|\theta) \cdot p_0(\theta)}{p(x_1 \dots x_n)} = \frac{p(x_1 \dots x_n|\theta) \cdot \overbrace{p_0(\theta)}^{\text{prior}}}{\underbrace{\int_{\theta^*} p(x_1 \dots x_n|\theta)\, p_0(\theta)\, d\theta}_{\text{normalizing factor}}}$$

$$\propto p(x_1, \dots x_n|\theta) \cdot p_0(\theta)$$

$$= \theta^{\sum_{i=1}^{n} \delta_{x_i,T}} (1-\theta)^{\sum_{i=1}^{n} \delta_{x_i,H}} \cdot p_0(\theta)$$

Normalising factor:

$$\int_{\theta^*} p(x_1 \dots x_n|\theta)\, p_0(\theta)\, d\theta = \underbrace{\theta^{\sum_{i=1} \delta_{x_i,T}} (1-\theta)^{\sum_{i=1} \delta_{x_i,H}} \cdot p_0(\theta)}_{\theta = 0{,}5} + \underbrace{\theta^{\sum_{i=1} \delta_{x_i,T}} (1-\theta)^{\sum \delta_{x_i,H}} \cdot p_0(\theta)}_{\theta = 0{,}55}$$

Prior: $p_0(\theta) = \cancel{p}\,\frac{1}{2}\delta_{\theta,0.5} + \frac{1}{2}\delta_{\theta,0.55}$ or $p_0(\theta) = \theta^{\alpha}\cdot\theta^{\beta}$

Prediction

$$\underbrace{p(x_{n+1}=T\,|\,x_1\dots x_n)}_{\text{WHAT is the probability of next 'T'}} = \int_{\theta^*} p(x_n=T\,|\,\theta)\,p(\theta\,|\,x_1\dots x_n)\,d\theta$$

$$\propto \int p(x_n=T\,|\,\theta)\,p(x_1\dots x_n\,|\,\theta)\cdot p_0(\theta)\,d\theta$$

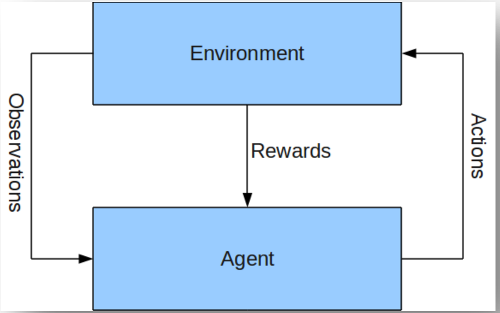
$$= \int_{\theta} \theta^{\delta_{x_n,T}}\cdot\theta^{\sum_i^n \delta_{x_i,T}}\,(1-\theta)^{\sum_{i=1}^n \delta_{x_i,H}}\cdot p_0(\theta)\,d\theta.$$

Continue the example, define prior and compute $p(x_{n+1}=\text{'H'}\,|\,x_1, \dots x_n)$

Recap: Bayesian learning

- Assigns probabilities to hypotheses
- Combines prior knowledge (prior probabilities) and observations
- Provides practical (feasible) learning algorithms for Markov chains and linear gaussian models
- Serves as a basis for machine learning
- Serves for evaluating other learning algorithms
- …

Recall: MDP formalisation

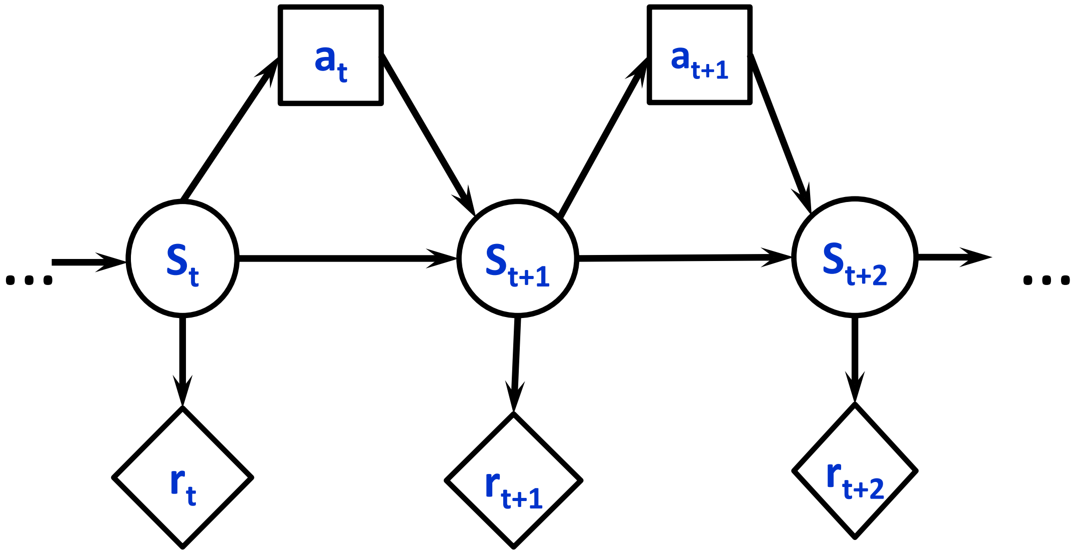


MDP is defined by (T, S, A, R, Pr):

- S - finite set of all possible *states*, $|S| = n$
- A_s - finite set of allowable *actions* (decisions) in state s.
  $A = \cup_{s \in S} A_s$ - the set of all possible actions, $|A| = m$
- $Pr(s_{t+1}|s_t, a_t)$ - *state transition function*
  - represented by set of n x n probability matrices for each a_t
  - each $Pr(s_{t+1}|s_t, a_t)$ is a distribution over S
- bounded, real-valued *reward function* $R(s)$
  - represented by an n-vector
  - can be generalised to include action costs $R(s,a)$
  - can be negative to reflect the cost incurred
  - generally can be stochastic (replaceable by expectation)

Recall: Decision Epochs

- Times at which decisions are made
  (analogous to period start times in Markov Process)
- The set *T* of decisions epochs can be either a discrete set or a continuum.
- The set *T* can be finite (*finite horizon problem*) or infinite (*infinite horizon*).



Recall: States and Transition Functions

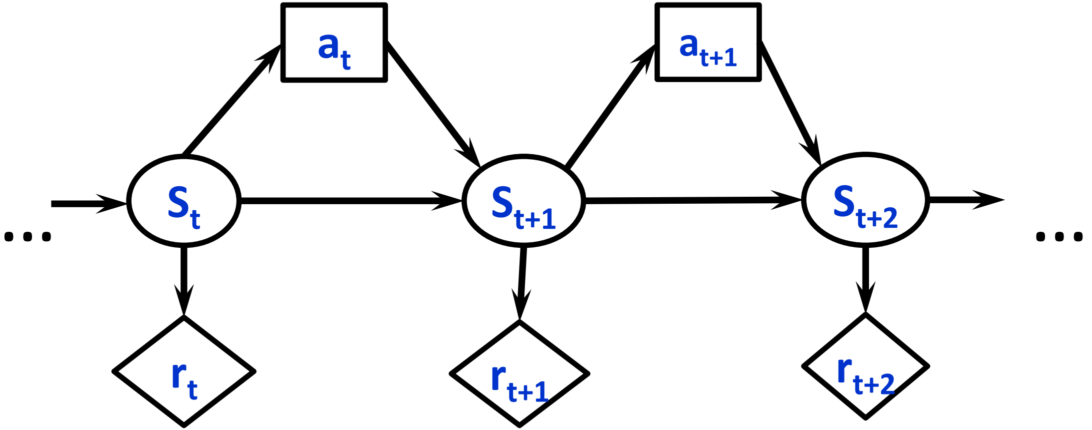
States s are analogous to states in Markov Processes and include all info from the past relevant to the future.

Transition function is a distribution that governs how the state changes as actions are taken over time.

As a result of choosing action $a \in A_s$ in state s at decision epoch t, the system state at $t+1$ is determined by the probability distribution $p_t(. | s, a)$.

For each state s and a

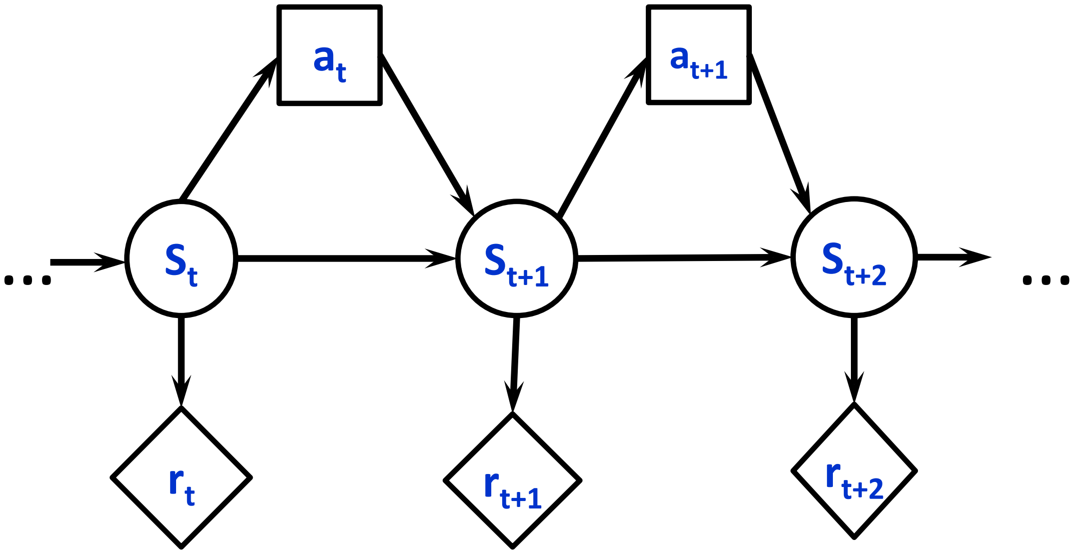
$\sum_{s_{t+1} \in S} Pr(s_{t+1} | s_t, a_t) = 1$

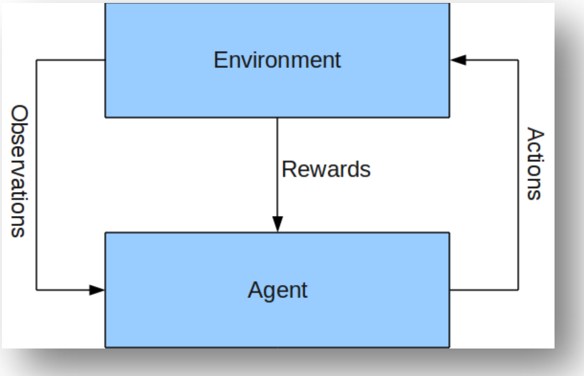


Actions

Actions *a* are means by which the agent interacts with the environment

- permissible actions can be *state dependent*
- *no* exact analogy to Markov Processes
- way of selecting decisions is usually modelled outside MDP





Decision rule

A *decision rule* prescribes a procedure for action selection in each state s at a specified decision epoch t.

A decision rule $d_t(s)$ can be either:

- *Markovian* (memoryless) if the selection of action a_t is based only on the current state s_t;
- *History dependent* if the action selection depends on the past history, i.e. the sequence of state/actions $h_t = (s_1, a_1, \ldots, s_{t-1}, a_{t-1}, s_t)$
- *Deterministic* if the decision rule $d_t(s)$ selects one action with certainty
- *Randomised* if the decision rule $d_t(s)$ specifies a probability distribution on the set of actions

Policy

Policy (strategy) is a collection of decision rules for all states.

$$\pi = (d_1(s), d_2(s), \ldots, , d_N(s)) \text{ or } \pi = (d_1(s), d_2(s), \ldots)$$

Note: expession above is simplified as $d_i(s)$ and $d_j(s)$, can generally operate on *different* states

What does a policy look like?

You can pick action based on states visited & actions used so far, i.e.

$s(1)\ a(1), s(2)\ a(2), \ldots$ or pick actions randomly using decision rule

Policy π is a mapping from each state $s \in S$ s to an action $a \in A_s$

For Markov fully observable process and deterministic decision rule:

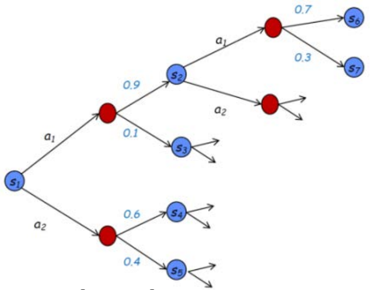
- *Non-stationary policy*

  $\pi : S \times T \rightarrow A$

  $\pi(s,t)$ is action to do at state s with t-stages-to-go
- *Stationary policy* $\pi : S \rightarrow A$

  $\pi(s) = (d(s), d(s), \ldots)$ is action to do at state s (independent of time)

Policy (cont.)



- MDP trying to find the minimum cost path
- fixed paths won't suffice for MDPs, because we don't know which states the random environment will take.
- **policy** specifies an action for *every* single state, not just the states along some path. This cover all paths and advises what action to take no matter which state is.
- no need to take different actions at a given state, i.e the state contains all information needed to act optimally for the future. Every time we are in s we have the same DM problem and hence should take the same optimal action (recall: the transitions and rewards satisfy the Markov property).

Goals and Rewards

Goal:

- should specify **what** we want to achieve **not how**
- is **not** the path to a specific state but reaching a specific state
- must be outside the agent's direct influence

Reward:

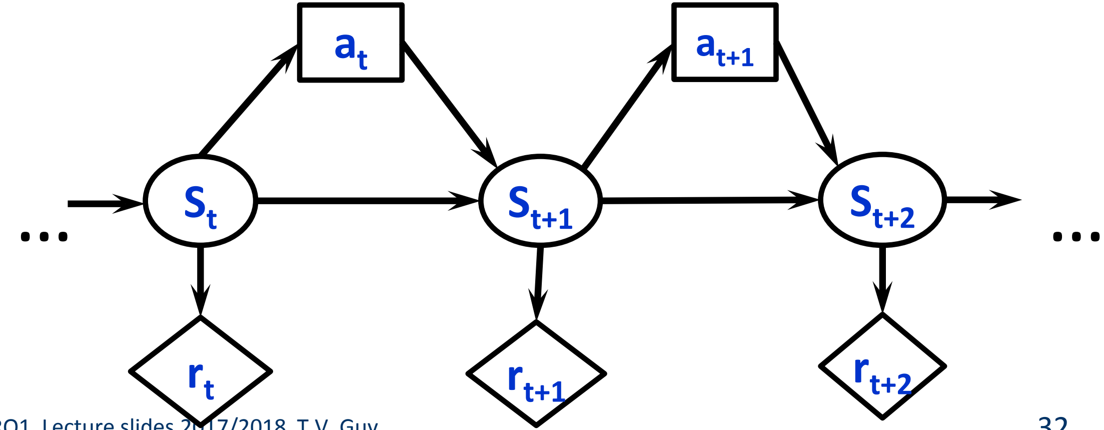
should serve an agent **to measure** success of reaching the goal explicitly and frequently during all decision epochs.

Reward

As a result of choosing action $a \in A_s$ in state s at decision epoch t,

- the agent considers a reward $R_t(s, a)$ that is expected immediate income/gain associated with taking a particular action at state s.
  (analogous to state utilities in Markov Processes)
- If the reward depends on the state at next decision epoch, then

  $R_t(s, a) = \sum_{j \in S} r_t(s, a, j)\, p_t(j|s, a)$, where $r_t(s, a, j)$ is the immediate reward if the next state is j.



Utility function

Executing a policy yields a sequence of rewards $R_1, R_2, \ldots$

How good is a policy π in a state s?

Define *utility function* $U(R_1, R_2, \ldots)$ to be some "quality measure" of a reward sequence

(The utility of a policy is the sum of the rewards on the path i.e utility is a random quantity).

- For *deterministic* actions criterion is sum of rewards obtained

  problem: infinite horizon => *infinite* result

- For *stochastic* actions, criterion is expected total reward obtained– again typically yields *infinite* value.

How do we compare policies of **infinite** value?

Utility function (cont.)

- Assume *stationary* agent preferences = *agent's preferences do not change with time* , i.e.

  $[s_0,s_1,s_2,\ldots] \succ [s_0,s_1',s_2',\ldots]$ iff $[s_1,s_2,\ldots] \succ [s_1',s_2',\ldots]$

  Note: a way how state s was reached does not affect the best policy from s

- Stationarity assumption allows to define utilities of state sequences
  - *Additive rewards* : $U(s_0,s_1,s_2,\ldots) = \Sigma_t R(s_t)$
  - *Discounted rewards* : $U(s_0,s_1,s_2,\ldots) = \Sigma_t \gamma^t R(s_t)$,

    with discount factor $0 \leq \gamma \leq 1$

    Intuitive interpretation: prefer utility sooner than later;

    γ indicates degree of agent's preference for the current over future reward

    $\gamma = 0$ future rewards are considered insignificant; $\gamma = 1$ future reward are important as the current one

Utility function (cont.)

Consider *no* terminal state (or if the agent never reaches it); *additive* utilities and reward is *bounded* by $R_{max.}$

Then total utility of an infinite action sequence is finite:

$$U(s_0,s_1,s_2,\ldots) = \Sigma_{t=0,1,\ldots}\ \gamma^t R(s_t\ ,a_t) \leq \Sigma_{t=0,1,..}\ \gamma^t R_{max} = \gamma^t R_{max}\ (1-\gamma).$$

Notes:

An *utility* function maps infinite sequences of rewards to single real numbers.

Discounting is the most analytically tractable approach

With a proper policy (with guaranteed terminal state) no discounting is needed.

An alternative to discounting in infinite-horizon problems is to optimize the *average reward over the long run* that is more complicated computationally (beyond the scope of this course)