

CSCI 400 Artificial Intelligence
David Keil, Framingham State University

Topic 6: Adaptation and reinforcement learning

1. Interaction and intelligent behavior
2. Decision theory and expected utility
3. MDPs, policies, and utility
4. POMDPs and reinforcement learning
5. Robotics and embodied intelligence

Inquiry

- Is *adaptation* a higher form of intelligent behavior than generalization?
- Are situated and embodied forms of intelligence more robust than reasoning-based systems?
- [PIX OF ALG, INTER]

Objectives

- 6a. Identify problems that require interaction or adaptation
- 6b. Describe bounded rationality
- 6c. Describe methods of reinforcement learning
- 6d. Explain features of robotic systems

Reading:

Ch. 17, 21, 24-25

1. Interaction and intelligent behavior

- *Intelligence* is associated with the ability to adapt to an interactive, changing environment
- *Exploration* of an environment is another interactive way to learn
- *Adaptation* is learning that improves *interactive* behavior
- Different instances of adaptive intelligence are shaped by different environments

Online search

- Percepts and actions are interleaved
- Agent is informed of state when it enters it
- Online search is necessary for exploration
- “Online” is as opposed to algorithmic [note confusion in RN]
- Only nodes that an agent occupies may be expanded
- Backtracking requires that actions be reversible

Interactive environments

- An environment is *persistent* if its outputs depend on inputs received prior to the most recent; otherwise the environment is *amnesic* (episodic)
- An environment E is *dynamic* w.r.t. an agent or MAS A if its outputs to A are *strictly dependent* on its previous inputs from A ; otherwise *static*
- A *dynamic* environment may change autonomously with respect to an agent
- A *physical* environment is observable by analog sensors; a *virtual* environment is digital;

The paradigm shift to interaction

Algorithmic

Structured design

Logic and search in AI

Rule-based reasoning

Closed systems

Compositional behavior

Transforming input
to output

Interactive

Object-oriented design

Agent-oriented AI

Planning, simulation, control

Open systems

Emergent behavior

Providing a service
over time by agents

Three kinds of adaptation

- Interactive learning of three kinds can occur:
 - Adaptation by individuals (ontogenetic)
 - Competition in a population (sociogenetic)
 - Evolution of a species (phylogenetic)
- Living and artificial agents interact with their dynamic environments through streams of percepts and actions

Game strategies

- *Nash equilibrium*: combination of two dominant strategies for opposing players, or situation where neither player could improve chances with different strategy
- *Zero-sum game*: one in which outcomes of the two players are arithmetic inverses; e.g., win/lose
- Strategies for iterated prisoners' dilemma: perpetual punishment for defecting, tit-for-tat, ...
- *Partial-information games* are solved using *belief states*

2. Decision theory and expected utility

- Whereas goal-directed agent partitions states as good or bad, a decision-theoretic one has a continuous utility function for actions and states
- Expected utility of an action A given evidence E :

$$EU(A | E) = \sum_i P(\text{Result}_i(A) | \text{Do}(A), E) \cup (\text{Result}_i(A))$$
 [EXPLAIN]
- Principle of maximum expected utility recommends choice of action that maximizes utility
- Probabilities and utilities are hard to compute

Axioms of utility theory

- A *partial ordering* of states by utility exists
- *Transitivity* of ordering
- *Continuity*: if $A > B > C$, then some bet on B or A with a certain probability is rational
- *Substitutibility* in lotteries
- *Monotonicity*:
- *Decomposibility*: complex lotteries are reducible to simple ones

Utility principles

- *Principle*: if an agent obeys utility axioms, then there is a utility function U for it, s.t.
 $(A > B) \Rightarrow (U(A) > U(B))$
 $(A \sim B) \Rightarrow (U(A) = U(B))$ [EXPLAIN A, B]
- *Maximal expected utility principle*:
Utility of a lottery is sum of products of probabilities and utilities of states
- Money may measure utility
- Some agents are risk-averse, giving extra utility to sure-things over bets

Decision theory

- *Utility theory* joins with *probability theory* in *decision theory*
- *Rational agent*: one that chooses actions that yield maximum expected utility averaged over all outcomes
- Agent using decision theory is like the logical agent, except that it acts on a *belief state*:
 - Update belief state using percept
 - Calculate outcome probability for actions
 - Select action with maximum probable utility

Value of information

- Actions may include obtaining information
- Value of information is difference between expected values of best actions with and without the information
- Information has value depending on likelihood of changing to a plan that is better than the current one
- When no information observation is worth its cost, “real” (non-information-obtaining) actions are taken

Sequential decision problems

- Utility depends on a series of actions
- Sensing between actions may matter
- Typically stochastic
- *Example*: reach goal state, avoid death state, on a 3 x 4 grid, with unreliable actions [see pic p. 614]
- Transitions are Markovian
- Rewards are (negative) additive
- This is a *Markov decision process*
- Assume environment is fully observable

Bounded rationality

- Notion suggested by Herbert Simon, 1972, as alternative to classical rationality assumption of economic theory
- *Argument*: Humans have limited knowledge and resources for decision making
- Alternative goal to optimality: *satisficing* (good enough)
- *Rational agent*: one that chooses actions that yield maximum expected utility averaged over all outcomes

3. MDPs, policies, and utility

- A *Markov decision problem* (see topic 4) is defined by an initial state s_0 , a transition model $T(s, a, s')$, and a reward function $R(s)$
- A solution specifies a *policy* $\pi(s)$: what agent should do given any perceived state of environment
- Policies have *expected utilities*: utility of possible environment histories generated by it
- Optimal (maximal-utility) policy is called π^*
- *Proper policy*: one guaranteed to reach a terminal state

Environments and policies

- *Environment* is an MDP defined by a set S of states, a transition probability model P , an action set A , and a set R of reward values for each state

- *Example:*

0	0	+10
0	0	0
0	0	0

- Policy $\pi: S \rightarrow A$ is a mapping from perceived states to *actions*

Good policy:

→	→	↓
→	→	↑
→	↑	↑

Poor policy:

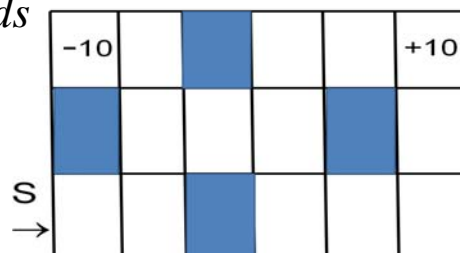
←	←	←
←	←	←
←	←	←

Reward vs. utility

- *Reward* is obtained immediately upon entering a state
- In chess, reward is win or lose; in ping-pong, reward is point won/lost; in animals, reward is pleasure/pain, hunger/food
- *Utility* of a state is the expected long-term cumulative reward for an agent that starts in that state
- Utility can provide a guide to rational decisions

Sample environment

- The following environment has $6 \times 3 = 18$ states, 14 of which are accessible, and two of which have *rewards*
- Possible actions are *up, down, left, right*
- *Learning problem:* choose a rational *policy* for this environment by assigning an action to each state



Policy search

- *Policy*, a mapping from states to actions, is as opposed to *action sequence*
- An agent that follows a policy incorporates percepts about state into action determination
- Future rewards may be discounted in deciding expected utility
- *Policy search* is search in the belief state space, which is now continuous because of P function
- *Value of information* counts in policy search

Value iteration algorithm

- Calculates utilities of states, allowing choice of optimal action for each state
- Utility of state s under policy π $U_{\pi}(s)$ depends on reward $R(s)$, but utility and reward are distinct, in that U is long-term expected reward
- *Utility of a state*: $R(s)$, plus expected discounted utility of the next state under optimal choice, using discount value γ
- Evolves a table U of estimated utilities of states

Value-iteration(E, γ) $U \leftarrow (0,0,0,0,0,0,0,0,0,0)$

repeat

 $U \leftarrow U'$ $\delta \leftarrow 0$ For each $s \in S$ $U'[s] \leftarrow E.R(s) + \gamma \max_{a \in A} E.P_{s'}(s' | s, a) U[s']$ if $|U'[s] - U[s]| > \delta$ $\delta \leftarrow |U'[s] - U[s]|$ until $\delta < \varepsilon(1 - \gamma) / \gamma$ return U U is utility estimate A is action set $R(s)$ is reward $P_{s'}(s' | s, a)$ is probability of transition from s' to s on action a γ is discount value

0	0	+10
0	0	0
0	0	0

 R

- After value iteration, *policy* may be set by choosing actions that favor high-valued (high-utility) states

Policy iteration

- Iterated steps:
 - *Policy evaluation*: calculate utility of each state U_i assuming π_i
 - *Policy improvement*: use 1-step lookahead to find π_{i+1} based on U_i
- Terminates when no improvements are made
- Works with any subset of states

Utilities of states under policies

- *Utility* U for a state, given a policy π , is long-term expected reward for an agent in that state
- Bellman equation:
$$U_i(s) = R(s) + \gamma \sum \mathbf{P}(s' | s, \pi_i(s)) U_i(s')$$
- *Algorithm*: repeatedly evaluate this formula for $i = 1$ to 10 or so, or until it converges

4. POMDPs and reinforcement learning

- The real world is an example of a *partially observable Markov decision problem* (POMDP)
- POMDPs have an *observation* model $O(s, o)$ that gives $P(\text{observe}(o), \text{in-state}(s))$
- *Belief state* is a probability distribution over all states; *optimal action* is a function of current belief state
- *Reinforcement learning* is an approach to solving POMDPs, guided by exploration of the environment rather than guided by a teacher

Reinforcement learning

- Agents that precompute action sequences with an algorithm cannot interactively respond to new sensory information
- Action-response experiments provide information about environment
- But to help make sure that actions are useful, *reward* (reinforcement) is needed in this exploration
- RL searches for a reward-maximizing *policy* without starting with a model of environment

Passive RL

- *Objective*: to learn utility of states given a policy π ; i.e., to learn $U^\pi(s)$, by value iteration
- *Utility*: expected sum of discounted rewards under π
$$U^\pi(s) = \mathbb{E} [\sum_{t \in \mathbb{N}} \gamma^t R(s_t) \mid \pi, s_0 = s]$$
- *Example*: run trials until in final state; record accumulated rewards
- This reduces RL to inductive learning, but omits interdependence of states' utilities and learns nothing until end of trial

Passive vs. active RL

- A *passive-learning* agent has fixed policy
- It learns the utilities of states of the environment, or state-action pairs
- An *active-learning* agent adapts its policy to the environment
- [Example]

Adaptive dynamic programming

- Learns transition model of the environment as it goes along
- Uses dynamic programming (tables) to solve MDP calculating states' utilities from model and from observed reward function
- For large state spaces, is intractable

Temporal difference learning

- Uses observed transitions to adjust state values
- Updates $U^\pi(s)$ with difference in utilities between successive states
- Acts by moving utility estimates locally toward equilibrium equation
- Does not require a model of environment
- Update rule, based on transition from state i to j :
 $U(i) \leftarrow U(i) + \alpha(R(i) + U(j) - U(i))$, where R is reward, α is learning rate

Active reinforcement learning

- Learning by policy iteration; policy not fixed as in passive RL
- *Greedy agent* leaves a good policy, fails to explore further, usually ends with suboptimal policy for actual environment even if it is optimal for learned model
- *Exploration* and *exploitation* are tradeoffs
- *Exploration function* determines tradeoff between greed, curiosity

Q-learning policy search

- Policy π : states \rightarrow actions
- $\pi(s) = \max_a Q_\theta(a, s)$ [CHECK]
- $Q(a, s) =$ estimated utility of action a in state s
- Policies are discontinuous functions, because policy may shift response on infinitesimal change of percept
- Hence policy search uses stochastic representation of policy, giving probability of choice of action a in state s

Summary of reinforcement learning

- *Approaches to design:*
 - Model based, using model T , utility function Q
 - Reflex, using policy π
- Exploration vs. exploitation trade-offs are encountered
- RL may eliminate hand coding of control strategies
- Applications in robotics are inviting
- *Environments:* continuous, high-dimensional, partially observable

4. Robotics and embodied intelligence

- *Robot*: a physical agent that acts in a physical environment by exerting physical force
- *Effectors*: wheels, grippers, joints, legs
- *Sensors* enable perception: cameras, sound, accelerometers gyroscopes
- *Types*:
 - Manipulator (robot arm), 1 million worldwide
 - Mobile (land, water, air)
 - Hybrid: mobile with manipulators

Robust methods in robotics

- Robust methods handle uncertainty by assuming boundedness, working regardless of values within an interval
- *Fine-motion planning* consists of guarded motions (command, termination condition), e.g., motion to fit a peg in a hole
- Plans are designed for worst-case successful outcomes

Robotics and Markov processes

- State transition of robot is often probabilistic, hence modeled by *Markov decision process* (MDP)
- Solution to MDP is a *policy* (robotic navigation function), i.e., a state-action mapping
- Partially observable environments are POMDPs, where robot maintains internal belief state
- Uncertainty about environment state may trigger *information gathering actions*

Perception

- *Definition*: conversion of sensor input to internal representation of environment
- *Difficult because of noise, dynamism, unpredictability, hiddenness*
- Good internal representations
 - Enable decisions
 - Are efficiently updatable
 - Correspond naturally to the world
- *Updating representation* is updating of belief state via filters

Robotic perception

- Is initiated by sensors
- Provides information about an agent's environment
- Active sensing: sending signal to get response
- *Modalities:*
 - *Hearing*
 - *Touch*
 - *Vision*
- *Approaches:*
 - Feature extraction
 - Model based, constructing representation of world

Sensors

- *Passive sensors* capture signals originating in environment
- *Active sensors* capture reflected energy sent to environment
- *Range finders* measure distance
- *Tactile sensors* work at close range
- *Imaging sensors* use camera vision
- *Proprioceptive sensors* detect state of robot

Image processing

- Light from image is captured and processed
- Early processing:
 - Smoothing image to remove noise, e.g., by averaging neighbor values of a pixel
 - Edge detection
 - Segmentation: breaking image into groups

Vision in manipulation and navigation

- *Example task*: automated vehicle driving on freeway
- *Tasks*:
 - Lateral control – stay in lane
 - Longitudinal control – keep distance
 - Obstacle avoidance by evasive maneuvers
- *Actions*: steer, accelerate, brake

Extracting 3D information

- Goal is to interact with objects in the world
- *Object recognition*
 - Segmenting scene into objects
 - Determining shape (what remains unchanged under transformations like rotation, transposition)
 - Determining position and orientation of each
 - *Cues*: motion, stereopsis, texture, shading, contour

Object recognition

- *Segmentation*: image must be partitioned into groups of pixels that represent distinct objects
- *Segmentation* may be *top-down* or *bottom-up*, looking for face patterns or finding objects and trying to build a face out of them
- *Brightness-based recognition* enables face recognition
- *Feature-based recognition* detects and marks regions and edges; *shape similarity* may be detected
- *Pose estimation* determines position and orientation w.r.t. the viewer

Localization

- *Definition*: the problem of determining where things are
- *Tracking* localizes object over time when original location is known
- *Global localization* assumes previous location is unknown
- Sensor models may assume use of landmarks: stable recognizable feature of environment
- Localization may be combined with *mapping* of environment

Motion

- *Kinds*:
 - *Point to point* with target location
 - *Compliant* while in contact with an object
- *Configuration space*: a way to describe possible state of robot defined by location, orientation, and angles of joints
- *Workspace*: description of robot configuration in same coordinate system as environment

Robot motion

- *Dynamic state* (change of position and angle in 3D) changes under influence of forces applied
- *Differential equations*, relating quantity to change of quantity over time, model dynamics
- Alternative technique to kinematic planning is use of *controllers* to keep robot on track via feedback

Reactive control

- Some environments are hard to model
- *Reactive control* for reflex agents is an alternative to modeling of environment
- *Example*: Control rules may be to rotate contact with ground among six legs
- Finite-state machines may model reactive controllers
- Feedback from environment plays crucial role, generating *emergent behavior*

Robot software architectures

- Deliberate techniques (planning based on model of environment) are used at higher global levels
- Reactive techniques are used at lower levels
- *Subsumption architecture* (Brooks) assembles reactive controllers in layers, bottom up, using FSMs augmented by clocks
- Hybrid architectures are widely used

Subsumption architecture

- Developed by Rodney Brooks, MIT robotics researcher
- Challenge to notion of explicitly centralized representation
- Intelligent behavior is seen as emerging from
 - Interaction of system and environment
 - Interactions among layers of system with simpler and simpler behaviors
- “Use the world as its own model”

Robotics application domains

- *Industry and agriculture*, e.g., mining, harvesting, excavation, assembly, welding, painting
- *Transportation* – helicopters, wheelchairs, container loaders, hospital gofers
- *Hazardous environments*: nuclear waste cleanup
- *Exploration*: Mars, undersea, volcanoes
- *Health care (surgery), personal services, entertainment, human augmentation*

Limitations of robotics today

- Has failed in comparison with expectation of fifty years ago
- Robotics comes up short in processing power
- Robots perform at the level of insects
- “our descendants will cease to work in the sense that we do now”

Potential of robotics

- Brain is special-purpose, navigation and recognition oriented organ
- Language introduces universal-machine capabilities
- Current desktop computing is 1000 MIPS; monkeys have about 5M MIPS
- 100M-MIPS robots will exceed human brain's power

Concepts

accessible environment	game theory	persistent environment
adaptation	genetic algorithm	phylogenetic learning
adaptive dynamic programming	genetic operator	physical environment
amnesic environment	genetic programming	policy
decision network	greedy agent	policy iteration
decision theory	interactive computation	policy search
deterministic problem	iterated prisoners' dilemma	POMDP
dominant strategy	Markov decision problem	reinforcement learning
dynamic persistent environment	model-based learning	reward
emergent behavior	model-free learning	sociogenetic learning
evolutionary computation	Nash equilibrium	static environment
expected utility	No Free Lunch theorem	temporal difference learning
exploitation	online search	utility
exploration	ontogenetic learning	value function
fitness function	open system	value iteration
function-optimization problem	partial-information game	value of information
	partially observable Markov decision process	virtual environment
		zero-sum game

Concepts (robotics)

active sensor	object recognition
effector	passive sensor
feature extraction	reactive control
image processing	robot
image segmentation	sensor
imaging sensor	situatedness
information-	subsumption
gathering action	architecture
locality	tactile sensor
localization	tracking
manipulator	
mobile robot	

References

- D. Keil. Dissertation proposal, 2006.
- George Luger. *Artificial Intelligence*. Addison Wesley, 2005.
- Stuart Russell and Peter Norvig. *AI: A Modern Approach, 2nd ed.* Prentice Hall, 2003.
- R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT, 1998