Intelligent Agents

Agents

- **agent** – anything viewed as perceiving its *environment* through *sensors* and acting upon the environment through *actuators*.
    - general assumption: an agent can perceive their actions but necessarily the effect they have.
- **percept** – an agent's percetual inputs at a given instance.
- **percept sequence** – the history of all that an agent has perceived.
- *In general, an agent's choice of action at any given time can depend on the entire percept sequence thus far observed.*
- **agent function** - a map from the precept sequence to an action

Performance Measure

- **performance measure** – a *subjective* criterion to measure the success of an agent's behavior typically stipulated by the designer of the agent.
- *As a general rule, it is better to design a performance measure according to what one wants in the environment rather than according to how one thinks the agent should behave.*

Rationality

- **Rationality** – behavior depending on 4 factors:
    1. The performance measure defining "success"
    2. The agent's prior knowledge about the environment
    3. The actions the agent can perform
    4. The agent's percept sequence to date
- **Rationality vs. Perfection** – rationality maximizes expected performance while perfection maximizes actual performance.
    - **omniscience agent** – an agent that knows the actual outcome of its actions and can act accordingly.
- **rational agent** – For each possible percept sequence, a rational agent should select an action from its set of allowable actions that is expected to maximize its performance measure given the evidence of the percept sequence and the agent's prior knowledge about the environment.
    - **Information Gathering (Exploration)** – actions taken specifically to modify future percepts.
    - **learning** – the process of modifying prior knowledge based on experience.
    - **autonomy** – the ability to compensate for partial or incorrect prior knowledge thereby becoming independent of prior knowledge eventually.
    - Computing the agent function:
        1. <u>design</u> – prior knowledge incorporated by designers.
        2. <u>deliberation</u> – choosing the next action based on agent function.
        3. <u>learning</u> – using experience to modify behavior.

Task Environments

- **task environment** – the problem the agent is solving as characterized by 1) Performance Measure 2) Environment 3) Actuators 4) Sensors – PEAS.
- Properties of task environments
    - *Fully Observable vs. Partially Observable*
        - **fully observable** – sensor's detect all relevant aspects for choosing an action
    - *Deterministic vs. Stochastic*
        - **deterministic** – environment completely determined by current state and agent's action
        - **stochastic** – from the point of the view of the agent, the environment is a random variable dependent on state and action.
        - **strategic** – environment deterministic except for other agent's actions.
    - *Episodic vs. Sequential*
        - **episodic** – experience divided into atomic independent episodes.
        - **sequential** – current decision could affect all future decisions.
    - *Static vs. Dynamic*
        - **static** – environment doesn't change until agent makes a decision.
        - **dynamic** – environment doesn't wait for agent.
        - **semi**-dynamic – environment is static, but performance measure may change during course of the decision.
    - *Discrete vs. Continuous*
        - distinction that can be applied to state, time, percepts, or actions.
    - *Single agent vs. Multiagent*
        - **Multiagent**
            - competitive vs. cooperative
            - communication is often rational decision
            - stochastic behavior can avoid pitfalls of predictability.
- Hardest case: *partially observable, stochastic, sequential, dynamic, continuous, multiagent.*
- **Environment class** – the set of environments on which an agent is to act.
    - **Environment Generator** – selects an environment from the class with a certain likelihood.
    - We are interested in the average performance of the agent on the class.

Agent Structure

- Agent = Architecture + Agent Program
- **Architecture** – the machinery that an agent executes on.
- **Agent Program** - a concrete implementation of an agent function.
    - Table-driven approach is doomed to failure
        - size of table is too large for any feasible implementation
        - designer could never go through the enumerations
        - no agent could learn these table entries from experience
        - the designer has no guide to fill the table.

- **Simple Reflex Agents** – an agent that chooses actions only based on the current percept.
    - **condition-action rule** – a rule that maps a state (condition) to an action.
    - *Rational only if the correct decision can be made solely on the basis of the current precept… the environment is fully observable.*
    - *Randomization* can help an agent escape from infinite loops, but randomization in single-agent environments is usually not rational.
- **Model-Based Reflex Agents** – an agent that uses a model of the world to choose actions.
    - In order to handle partial observablity, agent must have state.
    - **internal state** – a representation of unobserved aspects of current state dependent on percept history.
    - Updating state requires
        1. Information about how the world evolves.
        2. Information about how the agent's actions affect the world.
    - **model** – knowledge about "how the world works".
- **Goal-Based Agents** – an agent that chooses actions in order to achieve goals.
    - **goal** – description of situations that are desirable.
    - *planning* and *search* address the problem of finding long sequences of actions use to find goals.
    - Goal-based approach more flexible than reflex agent's implicit encoding of goals in rules since the knowledge supporting a decision is explicitly modeled, thereby allowing for modifications.
- **Utility-Based Agents** – agent chooses actions based on a preference (*utility*) for each state.
    - Goals are inadequate when…
        - There are conflicting goals only some of which can be achieved.
        - Goals have some uncertainty of being achieved and one needs to weigh likelihood of success against the importance of a goal.
    - **utility function** – a mapping of a (sequence of) state(s) to a real number describing the "degree of happiness" of that state.

Learning – the process of modification of each component of an agent to make the components agree closer with the available feedback thereby improving the agent's performance.

- **learning element** – responsible for making improvements
- **performance element** – responsible for selecting external actions… the agent being modified.
- **critic** – provides feedback on the agent's performance and suggests improvements.
    - **performance standard** – a *fixed* measure of agent's performance.
        - distinguishes the *reward* in the percept by providing direct feedback on quality of agent's performance.
- **problem generator** – suggests actions that will lead to exploration.

Problem Solving

- **problem**
  - initial state – the initial configuration given as input to the agent.
  - actions – the set of actions (currently) available to the agent. This can be formulated using a successor function.
    - successor function *succ(s)* – returns the set of $\langle action, state \rangle$ tuples that define the resulting state achieved by taking that action from state *s*.
  - goal test – determines whether a state is a goal state.
    - goal (state) – a state that is desirable to the agent.
  - path cost – the numeric cost of a path reflecting a performance measure. Typically defined as the sum of costs of each step on the path.
    - **step cost** $c(x, a, y)$ - the cost of taking action *a* to go from state *x* to state *y*.
- **state space** – the set of all possible states an agent could be in.
  - the initial state and a successor function implicitly define state space as the set of states reachable from the initial state.
- **path** – a sequence of states connected by a sequence of actions.
- **solution** – a path from the initial state to a goal state.
  - **optimal solution** – the solution with minimal path cost.
- **problem solving agent** – an agent that attempts to discover a *solution* to a problem so that following the actions in that *solution* will lead to a goal. Typically agent design is based on "formulate, search, execute".
  - ***formulation*** – building a representation of the agent's world.
    - **abstraction** – the process of removing details about the world.
      - *valid* – any abstract solution can be expanded into a solution in the real world.
      - *useful* – carrying out and analyzing actions in the abstract world is easier than in the real world.
    - **goal formulation** – defining goals to limit the agent's objectives.
    - **problem formulation** – defining the set of actions and states relevant to achieving a goal.
  - ***search*** – takes a *problem* as input and returns a *solution* to the problem.
    - **Premise:** *An agent with several immediate options of unknown value can decide what to do by examining different possible sequences of actions that lead to states of known value and choosing the best such sequence.*
  - ***execute*** – carrying out the actions recommended in the *solution*.
- **incremental formulation** – a state is modified by operators that augment the state (add new components without altering old ones).
- **complete-state formulation** – every state contains all objects and are modified by operators that alter the arrangement of those objects.