Making Simple Decisions

decision-theoretic agent – an agent capable of making decisions in the face of uncertainty and conflicting goals via a continuous measure of state quality.

Combining Belief and Desire under Uncertainty

- **utility function** – describes the desirability of each state. Combined with the probability of each action's outcome these give expected utility of the action.
  - **expected utility** (*A* is the action, *E* is the evidence):
    $$EU[A|E] = \sum_i P(\text{Result}_i(A) | Do(A), E) U(\text{Result}_i(A))$$
  - **principle of maximum expected utility (MEU)** – a rational agent should choose the action that maximizes it's expected utility.
  - *If an agent maximizes a utility function that correctly reflects the performance measure for behavior, it will achieve the highest possible performance measure in averaging over all environments possible.*
- **one-shot decision** – agent only chooses the next action to make.
- **sequential decision** – agent must choose best possible sequence of actions.

Utility Theory

1. $A \succ B$ *A* is preferred to *B*.
2. $A \sim B$ agent is indifferent between *A* and *B*.
3. $A \succsim B$ agent prefers *A* to *B* or is indifferent.

- **Lottery** – a set of outcomes C_i with a probability p_i: $L = [p_1, C_1; p_2, C_2; \ldots; p_n, C_n]$
- **Axioms of Utility Theory**
  1. **Orderability** – for any two states, an agent must prefer one to the other or else be indifferent between them.
     $$(A \succ B) \vee (A \prec B) \vee (A \sim B)$$
  2. **Transitivity** –A preferred to B, & B preferred to C, then A preferred to C.
     $$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$$
  3. **Continuity** – If B is between A and C in preference, there exists probability *p* for which the agent is indifferent between getting B for sure and a lottery that yields A with probability *p* and C with probability *1-p*.
     $$A \succ B \succ C \quad \Rightarrow \quad \exists p \quad [p, A; 1-p, C] \sim B$$
  4. **Substitutability** – an agent indifferent to A and B is indifferent to 2 more complex lotteries, 1 with each A and B.
     $$A \sim B \quad \Rightarrow \quad [p, A; 1-p, C] \Rightarrow [p, B; 1-p, C]$$
  5. **Monotonicity** – If 2 lotteries have the same outcomes, A and B, and agent prefers A to B, then it also prefers the lottery with higher probability of A.
     $$A \succ B \quad \Rightarrow \quad p \geq q \Leftrightarrow [p, A; 1-p, B] \succsim [q, A; 1-p, B]$$
  6. **Decomposability** – Compound lotteries can be decomposed:
     $$[p, A; 1-p, [q, B; 1-q, C]] \sim [p, A; (1-p)q, B; (1-p)(1-q), C]$$

- Utility
  1. **Utility Principle**

     $$U(A) > U(B) \quad \Leftrightarrow \quad A \succ B$$

     $$U(A) = U(B) \quad \Leftrightarrow \quad A \sim B$$

  2. **Maximum Expected Utility Principle**

     $$U([p_1, S_1; \ldots; p_n, S_n]) = \sum_i p_i U(S_i)$$

- By observing a rational agent's preference, it is possible to construct the utility function representing what the agent's actions attempt to achieve.

Utility Functions

- the utility of money
  - monotonic preference – agent prefers more money to less
  - true utility of positive money is more *logarithmic*… given only a small amount of money, agent is willing to risk it all, whereas the rich need more incentive since less gain is not worth the risk of having nothing.
  - in considering negative money, utility becomes an S-curve… the deeper in debt one goes the more risk one is willing to take to eliminate it.
- *Insurance Premium*
  - insurance premium – the difference between the expected monetary value of a lottery and its certainty equivalent: $IP = U(L) - U\left(S_{EMV(L)}\right)$ where $S_{EMV(L)}$ is the state of having the expected monetary value of lottery *L*.
    - $IP > 0$ *risk adverse*
    - $IP = 0$ *risk neutral*
    - $IP < 0$ *risk seeking*
- utility scales and assessment
  - Consider transformation $U'(S) = k_1 + k_2 U(S)$ where k_1 is any constant and k_2 is any positive constant. Then the agent's behavior is the same for utility U and U'.
  - In a deterministic context, agent's behavior is unchanged by any monotonic transformation ➔ **value function** – a function that provides a ranking of states rather than meaningful numeric values.
    - best possible prize: $U(S_*) = u_\uparrow$
    - worst possible prize: $U(S) = u_\downarrow$
    - normalized utility - $u_\downarrow = 0$ and $u_\uparrow = 1$.
- **normative theory** – how a rational agent should act.

Multiattribute Utility Functions

- **multiattribute utility theory** – utility theory for outcomes involving two or more attributes: $\mathbf{X} = X_1, \ldots, X_n$.
- **strict dominance** – option 1 has higher value on all attributes than another option 2. Clearly the 1st option is chosen.
- **stochastic dominance** – if two actions A_1 and A_2 lead to probability distributions $p_1(x)$ and $p_2(x)$ on attribute X, then A_1 stochastically dominates A_2 on X if,

$$\forall x \quad \int_{-\infty}^{x} p_1(y)\,dy \le \int_{-\infty}^{x} p_2(y)\,dy$$

  - *If A_1 stochastically dominates A_2, then for any monotonically nondecreasing utility function U(x), the expected utility of A_1 is at least as high as the expected utility of A_2.*
  - **qualitative probabilistic networks** – algorithms for making rational decisions based on stochastic dominance alone.
- **representation theorems** – theorems that identify regularities in preference behavior; $U(x_1, \ldots, x_n) = f\left[f_1(x_1), \ldots, f_n(x_n)\right]$
- **preference independence** – attributes X_1 and X_2 are preferentially independent of X_3 if the preference between outcomes $\langle x_1, x_2, x_3 \rangle$ and $\langle x_1', x_2', x_3 \rangle$ doesn't depend on the value x_3.
- **mutual preferential independence (MPI)** – no attributes affect the way in which one trades off to the other attributes against each other
  - If attributes $X_1, \ldots, X_n$ are mutually preferentially independent, then the agent's preference behavior can be described as maximizing the function

$$V(x_1, \ldots, x_n) = \sum_i V_i(x_i)$$

    where each V_i is a value function referring only to the attribute X_i.
  - **additive value function** – a multiattribute value function that is the sum of value functions for individual attributes.
  - Even in situations where additive value functions are not valid, they often serve as good approximations to the actual value functions.
- **utility-independence** – an extension of preference independence to lotteries. A set of attributes **X** is utility-independent of a set of attributes **Y** if preferences between lotteries on the attributes in **X** are independent of the particular values of the attributes in **Y**.
- **mutually utility-independent (MUI)** – each subset of a set of attributes is utility-independent of the remaining attributes.
  - **multiplicative utility function** – a function that can express the behavior of any agent exhibiting MUI in only *n* single-attribute utilities and *n* constants for *n* attributes.

Decision Networks

- **decision network** – a Bayesian network with additional node types for actions and utilities. Contains information about the agent's current state, its possible actions, the state resulting from the agent's action, and the utility of the state.
  - Structure
    - Chance nodes (ovals) – represent random variables each with a conditional distribution indexed by parent states. Parents can be other chance nodes or decision nodes.
    - Decision nodes (rectangles) – represent points where agent has a choice to make.
    - Utility nodes (diamonds) – represent the agent's utility function. Its parents are all variables directly affecting utility.
  - **action-utility tables** – A simplified form in which the action is connected directly to the utility thus making the utility node represent the expected utility… a compiled version.

The Value of Information

- **information value theory** – theory describing what information is best to acquire in order to make a decision… *one of the most important parts of decision making is know what questions to ask.*
  - sensing actions – actions preformed in order to acquire information
  - **value of information** – the value of a piece of information is the difference between the expected utility between the best possible actions before and after information is acquired.
    - *Information has value to the extent that it is likely to cause a change of plan and to the extent that the new plan will be significantly better than the old one.*
- **value of perfect information (VPI)** – value of information assuming exact evidence E_j of some random variable is obtained:

$$VPI_E(E_j) = \left(\sum_k P(E_j = e_{jk}) EU(\alpha_{e_{jk}} \mid E, E_j = e_{jk}) \right) - EU(\alpha \mid E)$$

  - Properties
    - VPI is non-negative: $\forall j, E \quad VPI_E(E_j) \geq 0$
    - VPI is not additive (in general): $VPI_E(E_j, E_k) \neq VPI_E(E_j) + VPI_E(E_k)$
    - VPI is order-independent: $VPI_E(E_j, E_k) = VPI_E(E_j) + VPI_{E,E_j}(E_k) = VPI_{E,E_k}(E_j) + VPI_E(E_k)$
- **Information-Gathering Agent**
  - agent is **myopic** since the VPI formulation only accounts for the effect of evidence E_j given that only that E_j is observed without including the possibility that future evidence may make the observation of E_j more valueable.

Making Complex Decisions

Sequential Decision Problems – utility depends on a sequence of decisions.

- **transition model *T(s,a,s')*** – probability of going from state *s* to *s'* via action *a*.
  - **Markovian** – the probability of reaching *s'* from *s* depends only on state *s* and not on the entire history of earlier states.
- **environment history** – the sequence of states on which utility depends. In state *s*, the agent receives a **reward** of *R(s)* so we simply ***sum*** the rewards received.
- **Markov Decision Process (MDP)** – a fully observable environment with a Markovian transition model and additive rewards.
  - initial state S_0, transition model *T(s,a,s')*, & reward function *R(s)*
- **policy π** - a plan of what action to take in a given state: $a_t = \pi(s_t)$
- **optimal policy π^*** - a policy that yields the highest expected utility.
- Optimality for a sequential decision process
  - Is the task episodic or continual?
    - **finite horizon** – the decision process goes on for a fixed time *N* (optimal policy is **nonstationary**).
    - **infinite horizon** – process continues forever (**stationary** policy)
  - How to calculate the utility of state sequences?
    - **stationary preference assumption** – if two state sequences, $[s_0, s_1, s_2, \ldots]$ and $[s_0', s_1', s_2', \ldots]$, begin with the same state, $s_0 = s_0'$, then the preference order of the two sequences should be the as sequences $[s_1, s_2, \ldots]$ and $[s_1', s_2', \ldots]$ are ordered.
    - Under stationarity, there are only two possible utilities:
      - **Additive Rewards** $U_h([s_0, s_1, s_2, \ldots]) = \sum_{t=0}^{T} R(s_t)$
      - **Discounted Rewards** $U_h([s_0, s_1, s_2, \ldots]) = \sum_{t=0}^{T} \gamma^t R(s_t)$
        - $\gamma \in [0,1]$ is a discount factor equivalent to an interest rate of $(1/\gamma) - 1$.
    - How to calculate utility when history is infinite.
      1. For discounted rewards with a maximum reward $R_{\max}$ and $\gamma < 1$, utility is still finite:
         $$U_h([s_0, s_1, s_2, \ldots]) \le R_{\max} / (1-\gamma)$$
      2. **Proper policy** –guaranteed to always reach terminal state.
      3. Compare infinite sequences by mean reward per time step.
  - How to choose between policies?
    - A policy π generates a whole range of possible state sequences, each with a certain probability determined by the transition model.
    - Value of policy is the expected sum of discounted rewards.
    - optimal policy:
      $$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi\right]$$

Value Iteration – an algorithm to calculate the optimal policy by calculating the utility of each state and using state utilities to select an optimal action in each state.

- *Utility* of a state *s* by following policy π: $U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi, s_0 = s\right]$
- *True Utility* of state *s*: $U(s) = U^{\pi}(s)$
- *Maximum Expected Utility (MEU)* principle: $\pi^* = \arg\max_a \sum_{s'} T(s,a,s')U(s')$
- **Bellman Equation**

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

  - *The utility of a state is the immediate reward for that state plus the utility of the next state, assuming that the agent chooses the optimal action.*
  - For *n* possible states, there will be *n* Bellman equations in *n* unknowns. Unfortunately they are nonlinear.
- **Iterative Approach** – calculates the utility of each state via the utility of their neighbors → propagates information through the state space via local updates.
  - **Bellman Update**: $U_{i+1}(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U_i(s')$
  - Converges to a unique solution whose corresponding policy is optimal.
    - **contraction** – a unary function that, when applied to two different values in turn, causes their output values to be "closer together".
      - it can be shown, the function has a single fixed point
    - The Bellman update can be viewed as an operator *B* applied to the set of utilities: $U_{i+1} = BU_i$
    - **max norm**: $\|U\|_{\max} = \max_s |U(s)|$
    - The Bellman update is a contraction by a factor γ on the space of utility vectors. That is, let U_i and U_j be two utility vectors, then

$$\|BU_i - BU_j\|_{\max} \le \gamma \|U_i - U_j\|_{\max}$$

    - if $\|U_i - U\|_{\max}$ is the *error* in estimate U_i.
      - If R_{max} is the bound on the rewards, then the number of iterations required to reach an error of at most ε is,

$$N = \left\lceil \frac{\log(2R_{\max}) - \log(\varepsilon(1-\gamma))}{-\log(\gamma)} \right\rceil$$

    - If the update is small, then the corresponding error is small

$$\|U_{i+1} - U_i\|_{\max} < \varepsilon(1-\gamma)/\gamma \quad \Rightarrow \quad \|U_{i+1} - U\|_{\max} < \varepsilon$$

    - *What the agent really cares about is how well it will do if it makes decisions based on the current utility function.*
      - policy loss $\|U^{\pi_i} - U\|_{\max}$ - the most the agent can lose by executing policy π_i instead of the optimal policy.

$$\|U_i - U\|_{\max} < \varepsilon \quad \Rightarrow \quad \|U^{\pi_i} - U\|_{\max} < 2\varepsilon\gamma/(1-\gamma)$$

Policy Iteration – an alternative way to find optimal policies by alternating between 2 steps: policy evaluation and policy iteration.

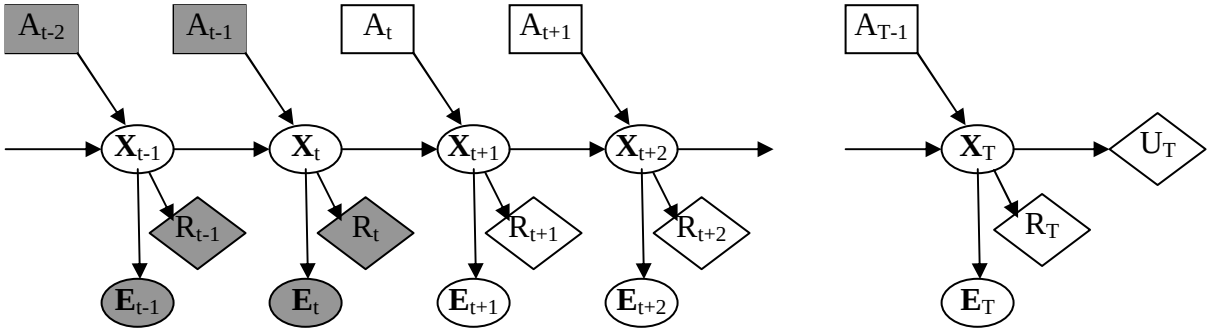
- **Policy Evaluation** – given a policy π_i, calculate $U_i = U^{\pi_i}$.
  - since policy is chosen, Bellman equations become linear:
    $$U_i(s) = R(s) + \gamma \sum_{s'} T(s, \pi_i(s), s') U(s')$$
  - Thus, given *n* states, this can be solved using linear algebra in $O(n^3)$.
- **Policy Iteration** – calculate a new MEU policy π_{i+1} based on maximizing U_i.
- **Modified Policy Iteration**
  - Use simplified Bellman updates repeated *k* times for the evaluation step:
    $$U_{i+1}(s) = R(s) + \gamma \sum_{s'} T(s, \pi_i(s), s') U_i(s')$$
  - Often more efficient than either value iteration or policy iteration
- **Asynchronous Policy Iteration** – pick any subset of states and apply either policy evaluation or policy iteration to that subset.
  - Under certain conditions on the initial policy and utility function, will still converge to optimal policy
  - Allows freedom to choose what states to work on.

Partially Observable MDPs (POMDP) – an MDP agent operating in a partially observable environment where the optimal action is state *s* also depends on how much the agent knows in state *s*. Defined in terms of a *transition model* $T(s,a,s')$, a *reward function* $R(s)$, and an *observation model* $O(s,o)$ that specifies the probability of perceiving observation *o* in state *s*.

- **belief-state *b*** – the set of actual states the agent might be in, represented by a probability distribution over all states.
  - If *b(s)* was the previous belief state when the agent executes action *a* and observes observation *o*, the new belief state is
    $$b'(s') \propto O(s', o) \sum_{s} T(s, a, s') b(s)$$
- *The optimal action depends only on the agent's current belief state* → a mapping $\pi^*(b)$ from belief states to actions.
- *Solving a POMDP on a physical state space can be reduced to solving an MDP on the corresponding belief state space with transition model* τ *and rewards* ρ.
  - The probability of an observation *o* given action *a* in belief state *b* is,
    $$P(o \mid a, b) = \sum_{s'} O(s', o) \sum_{s} T(s, a, s') b(s)$$
  - The *probability of transitioning* from belief state *b* to belief state *b'* via action *a* is,
    $$\tau(b, a, b') = \sum_{o} P(b' \mid o, a, b) \sum_{s'} O(s, o) \sum_{s} T(s, a, s') b(s)$$
  - The *reward function* for belief states is, $\rho(b) = \sum_{s} b(s) R(s)$
  - Finding even approximately optimal POMDPs is difficult – PSPACE-hard

Decision-Theoretic Agents – an approach to designing agents for partially observable stochastic environments

- **dynamic decision network** – a dynamic Bayesian network (for transition and observation models) augmented with decision and utility nodes.
  - $\mathbf{X}_t$ - set of state variables at t – transition $T(s,a,s') \equiv P(\mathbf{X}_{t+1} \mid \mathbf{X}_t, A_t)$
  - $\mathbf{E}_t$ - set of evidence variables at t – observation $O(s,o) = P(\mathbf{E}_t \mid \mathbf{X}_t)$
  - A_t - action made at time t.
  - R_t - reward received at time t.
  - U_t - utility of the state at time t.



  - current and future actions as well as future rewards and future observations are all unknown.
- A filtering algorithm is used to incorporate new actions and percepts and thereby update the new belief state via a forward update.
  - By marginalizing future observations, the decision theoretic agent accounts for value of information thereby allowing for information-gathering actions where appropriate.
  - Similar to ExpectMinimax algorithm except
    1. rewards can be non-leaf states
    2. decision nodes correspond to belief states
  - Time complexity for exhaustive search to depth d: $O(|D|^d|\boldsymbol{E}|^d)$ where $|D|$ is the number of available actions and $|\boldsymbol{E}|$ is the number of possible observations.
- Decisions are made by forward projecting possible action sequences and choosing the best one.
  - graceful degradation – can easily revise plan to handle unexpected observations.