CSCE 590-1: From Data to Decisions with Open Data: A Practical Introduction to AI

Lecture 18: Agents That Learn

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

16TH MAR, 2021

Carolinian Creed: "I will practice personal and academic integrity."

Organization of Lecture 18

- Introduction Segment
  - Recap/ Discussion of Lecture 17

- Main Segment
  - Reinforcement learning
  - Bayesian Optimization

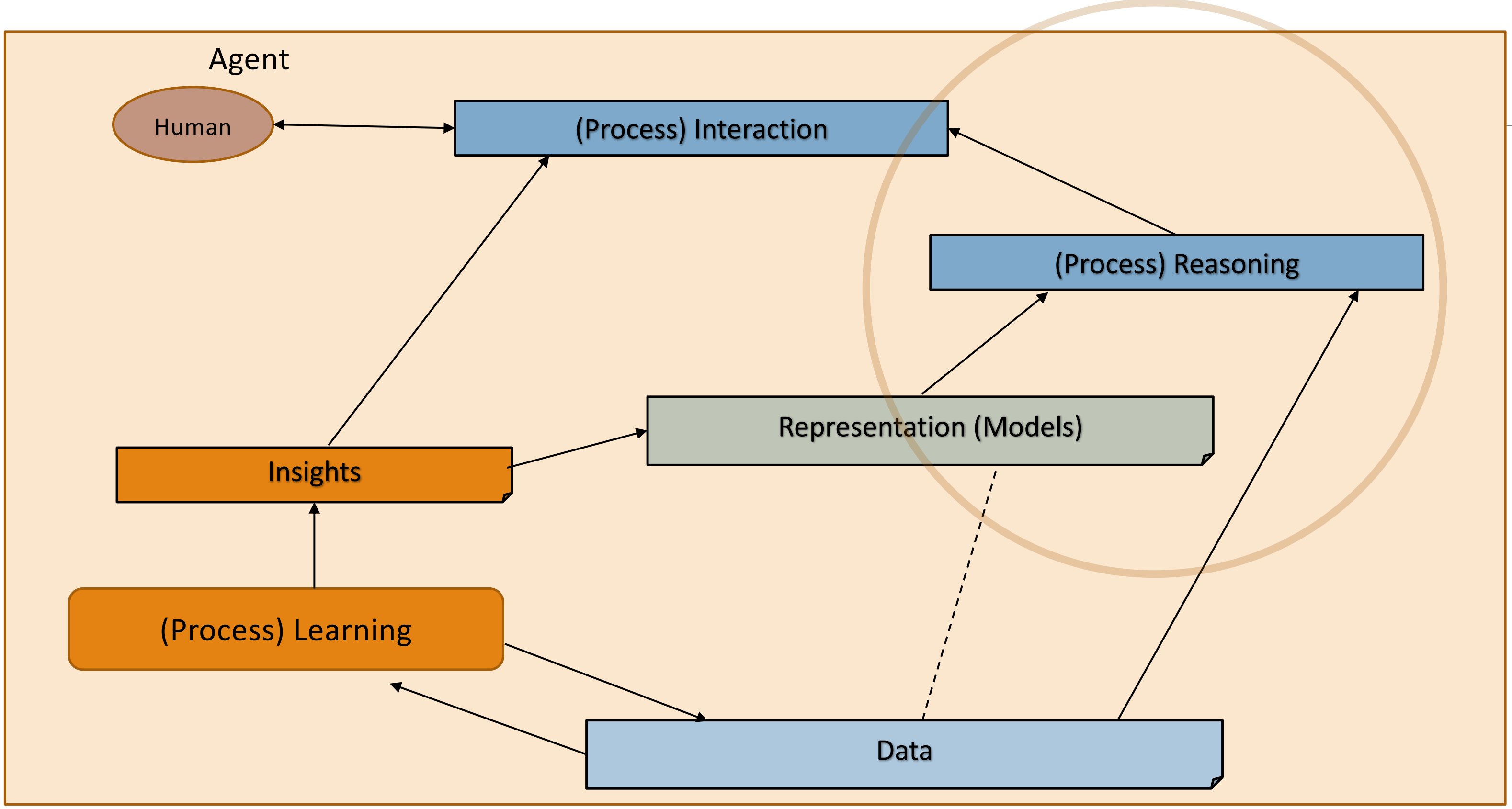
- Concluding Segment
  - About Next Lecture – Lecture 19
  - Ask me anything

Introduction Segment

Recap of Lecture 17

- Kind of uncertainties
- What is the best decision possible: Maximize Expectation
- Some methods
  - Bayesian methods
  - Utility theory
  - Markov Decision Processes

Relationship Between AI Processes



Machine Learning – Insights from Data

- Descriptive analysis
  - Describe a past phenomenon
  - **Methods**: **classification, clustering**, dimensionality reduction, anomaly detection, *neural methods*
- Predictive analysis
  - Predict about a new situation
  - **Methods**: **time-series**, *neural networks*
- **Prescriptive analysis**
  - What an agent should do
  - **Methods**: simulation, *reinforcement learning, Bayesian optimization*, **reasoning**

- New areas
  - Counterfactual analysis
  - Causal Inferencing
  - Scenario planning

Example Situation – Course Selection

- A person wants to pass an academic program in two majors: A and B
- There are three subjects: A, B and C, each with three levels (*1, *2, *3). There are thus 9 courses: A1, A2, A3, B1, B2, B3, C1, C2, C3
- To graduate, at least one course at beginner (*1) level is needed in major(s) of choice(s), and two courses at intermediate levels (*2) are needed
- **Learning Agent**: *The student learns from their performance in earlier (e.g., level-1) courses, from others who have take courses or graduated, switches courses mid-semester*
- Answer questions
  - Q1: Should I switch my course in the middle of the program ?
  - Q2: Should I major in all the courses that the program has?
  - Q3: Should I drop dual major and focus on one? Which one?
  - …

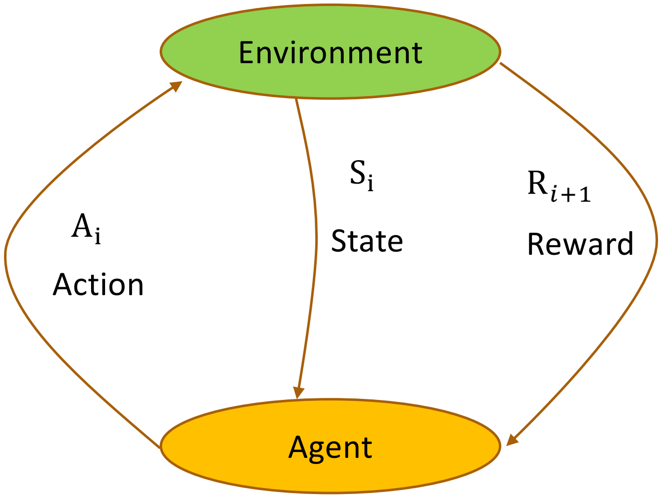
Main Segment

Reinforcement Learning Setting

- An agent in an environment
- Agent
  - Can see state
  - Can take action
  - Will get rewards

- Precisely, at each time step i
  - In state $\mathrm{S_i}$, agent takes action $\mathrm{A_i}$
  - Based on state s_i and action $\mathrm{a_i}$, the environment transitions to state $\mathrm{S_{i+1}}$ and outputs reward R_{i+1}

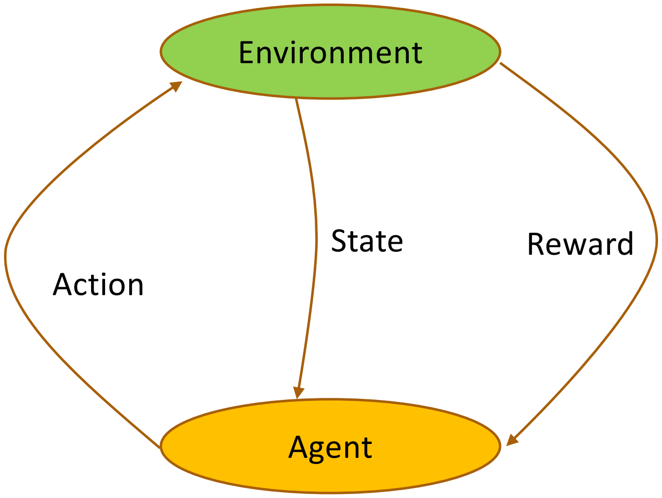
- **Objective**: learn mapping of states to actions so that the agent maximizes the reward from the environment.



Reinforcement Learning

- **Objective**: learn mapping of states to actions so that the agent maximizes the reward from the environment.

- **Output**
  - Deterministic: $a = \pi(s)$
  - Stochastic: $\pi(a|s) = P(A_i = a|S_i = s)$



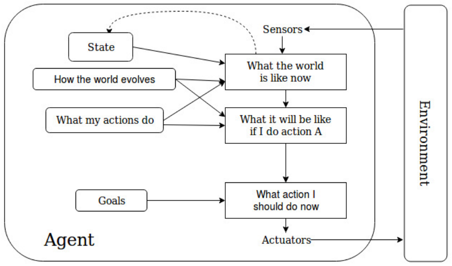
Comparison With Other Learning

- Supervised learning
  - Training information: labels
  - Objective: learn (input-label) mapping
  - Goodness criteria: Reduce error = (Predicted label – Actual label)
- Reinforcement learning
  - Training information: reward functions
  - Objective: learn policy
  - Goodness criteria: maximal reward
- These two forms of learning are orthogonal – for different tasks

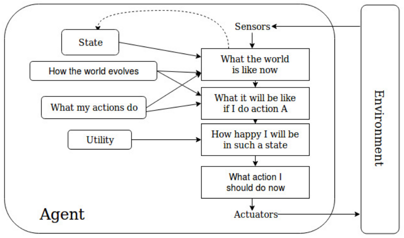
RL as a Learning-Based Agent

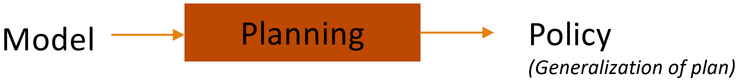
A general, alternative way of solving goal-based problems from just execution traces

Image Source: Russell & Norvig, AI: A Modern Approach


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

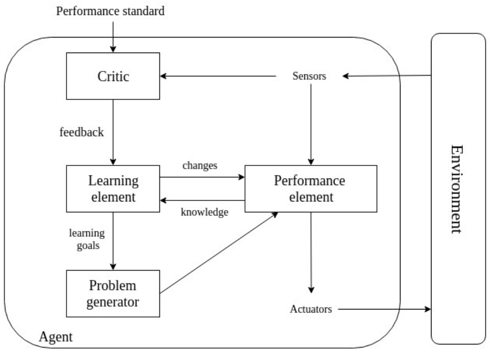
Goal- and Utility-based Intelligent Agent

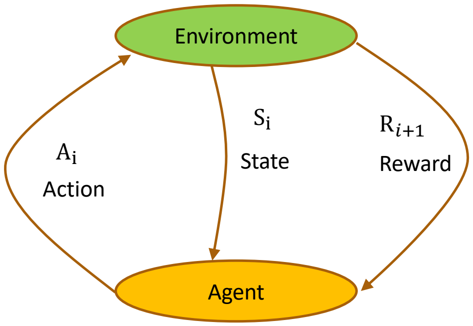

Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment



Model → Planning → Policy

(Generalization of plan)





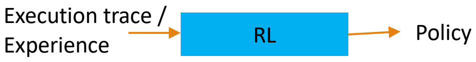
Environment

A_i Action

S_i State

R_{i+1} Reward

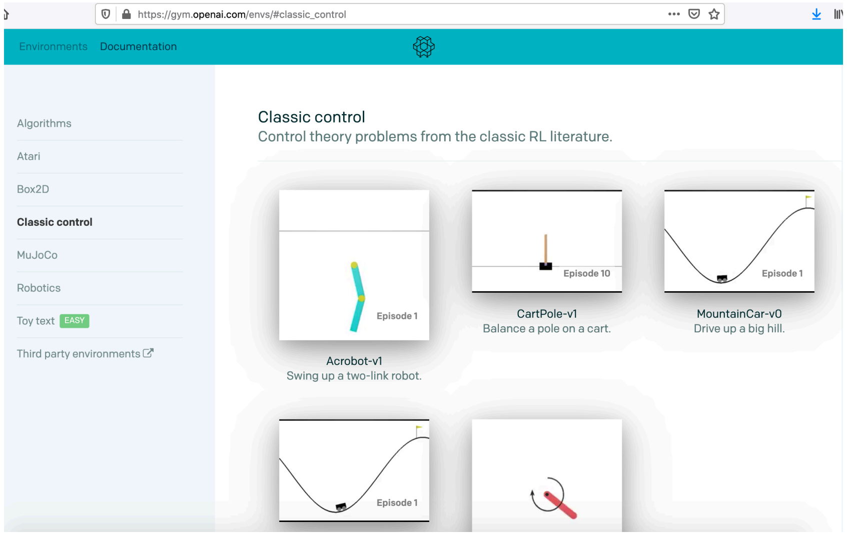
Agent



Execution trace / Experience → RL → Policy

Exercise and Code – Gym RL

- RL using Open AI’s Gym
  - https://gym.openai.com/
  - Environments: https://gym.openai.com/envs/#classic_control
- Exercise (5 mins):
  - Look at the various categories
  - Explore the videos



Classic control

Control theory problems from the classic RL literature.

Acrobot-v1
Swing up a two-link robot.

CartPole-v1
Balance a pole on a cart.

MountainCar-v0
Drive up a big hill.

Exercise and Code – Gym RL

- RL using Open AI’s Gym
  - https://gym.openai.com/

- Code: https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l18-learning-agent/RL%20using%20Gym.ipynb

Source: Russell & Norvig, AI: A Modern Approach

Diversity in RL Problems

- Environment - accessible or inaccessible
  - Accessible: states can be identified with percepts
  - Inaccessible environment: agent has to learn and maintain representation of state to track environment
- Knowledge of effects of action and utility, or learn
- Rewards
  - Available for all states or only terminal states
  - Actual utility or hints of increase/ decrease
- Ability to execute actions - Active learner or passive learner
  - A passive learner simply watches the world going by, and tries to learn the utility of being in various states
  - An active learner can actions to explore unknown environment

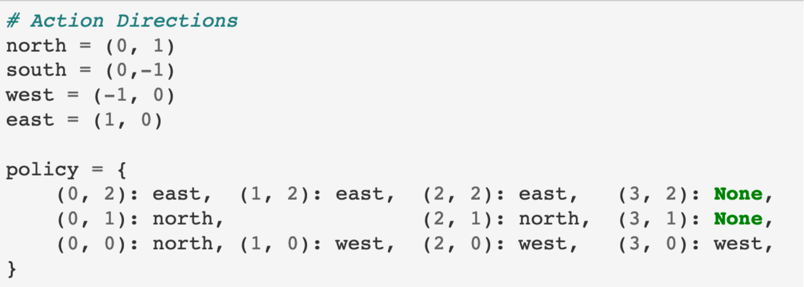
Source: Russell & Norvig, AI - A Modern Approach

Passive RL

- **Input**
  - policy: π_i
  - // Has no knowledge Reward R(s) and Transition function P(s' |s, a)
- **Output**
  - Expected utility for each state, U(s)
- **Procedure**:
  - Execute a sequence of runs
  - At any instant, the agent knows only its current state and current reward, and the action it must take next. This action may lead it to more than one state, with different probabilities.
- **Expected Utility**

$$U^{\pi}(s) = E(\sum_{t=0}^{\inf} \gamma^t R^t(s')$$

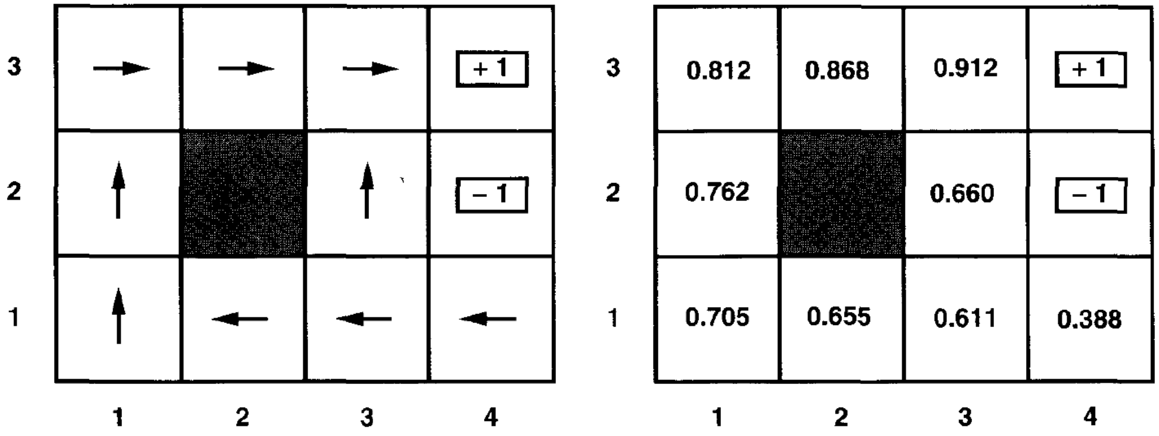
Illustration



```
# Action Directions
north = (0, 1)
south = (0,-1)
west = (-1, 0)
east = (1, 0)

policy = {
    (0, 2): east,  (1, 2): east,  (2, 2): east,   (3, 2): None,
    (0, 1): north,                (2, 1): north,  (3, 1): None,
    (0, 0): north, (1, 0): west,  (2, 0): west,   (3, 0): west,
}
```

Policy: https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l15-l16-l17-l18-agents/reinforcement_learning.ipynb



(a)

(b)

Input Policy and Output Optimal Utility

Source: Russell & Norvig, AI - A Modern Approach

The Markov Property – True of Many Domains

- **Our policy at timepoint t is only dependent on the current state s**
  - $\pi(a|s) = P(A_t = a|S_t = s)$
- Although the agent has a history up until S_t
  - $\mathrm{H_t} = S_0, A_0, R_1 S_1, A_1, R_2 \ldots S_{t-1}, A_{t-1}, R_t, S_t$
- One may assume that all relevant information about the future is contained in the current state and action
  - $P(S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a) = P(S_{t+1} = s', R_{t+1} = r|H_t = h_{t+1}, A_t = a)$
- This is a generalization of the Markov property to sequential decision problems
  - $P(S_{t+1}|S_t) = P(S_{t+1}|S_t, S_{t-1}, \ldots S_0)$

Source: Forest A.'s RL Course

RL with Finite States

Solving a Finite MDP

- **States:** A discrete and finite set $\mathcal{S}$
- **Actions:** A discrete and finite set $\mathcal{A}$
- **Transition Probabilities:** $P(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a)$
  - Defines the dynamics of the MDP
- The state-transition probabilities can be obtained from the transition probabilities
  - $p(s'|s, a) = \sum_{r \in \mathcal{R}} p(s', r|s, a)$ // Estimating state-transition by looking at reward of samples
- The **expected reward** can be obtained from the transition probabilities
  - $r(s, a\,) = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s, a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$
    // Estimating reward from transitions seen

Adapted from: Forest A.'s RL Course

Model-free RL: Q-learning

- Learning action-value functions
- Q(a,i): value of doing action a in state i
- Relationship between utility U of state and Q value
  - U(i) = max Q(a, i)
- Finding Q value based on whether transition probability is known
  - When M (transition is known)

$$Q(a,i) = R(i) + \sum_{j} M_{ij}^{a} \max_{a'} Q(a',j)$$

  - Estimating with TD method

$$Q(a, i) \leftarrow Q(a, i) + a\,(R(i) + \max_{a'} Q(a',j) - Q(a,i))$$

Source: Russell & Norvig, AI - A Modern Approach

RL with Deep Learning

- For small problems, like games, state-value function (U), action- utility value (Q), and transition functions (M), and policy functions are represented using a table

- But for large and realistic problems, number of states are countably large/ practically infinite
- Deep learning are excellent function approximators
  - Estimate Q-value i.e., action-value

- Not covered in this class

Exercise and Code – RL

- RL settings and solution methods

- Code: https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l18-learning-agent/RL%20using%20Gym.ipynb

Source: Russell & Norvig, AI: A Modern Approach

Inverse Reinforcement Learning

- Given π*and transition function M,
  - can we recover R

- Or, given execution traces corresponding to π*
  - can we recover R?

- Applications
  - Path planning
  - Automated-driving

- Reference: Pieter Abbel’s course slides: https://people.eecs.berkeley.edu/~pabbeel/cs287-fa12/slides/inverseRL.pdf

RL References

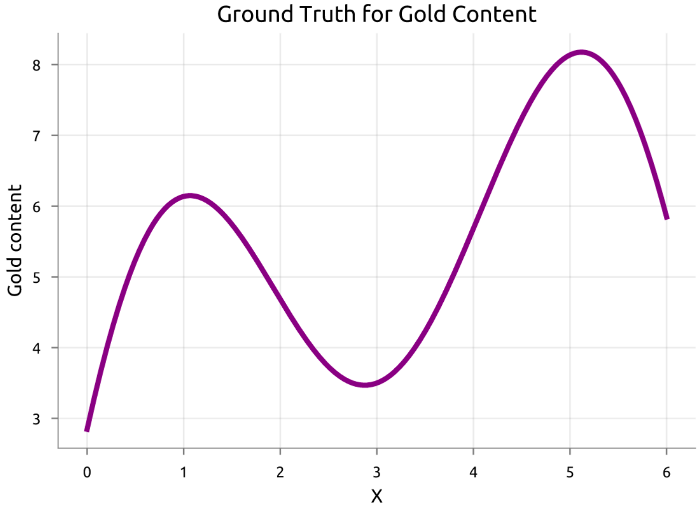
- Sutton and Barto’s Book: http://incompleteideas.net/book/the-book.html
- Russell and Norvig, AI – A modern Approach
- David Silver’s RL course, https://www.davidsilver.uk/teaching/

- Inverse RL
  - A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress, https://arxiv.org/abs/1806.06877, 2018
  - Pieter Abbel’s course slides: https://people.eecs.berkeley.edu/~pabbeel/cs287-fa12/slides/inverseRL.pdf

Bayesian Optimization (BO)

BO Problem Setting

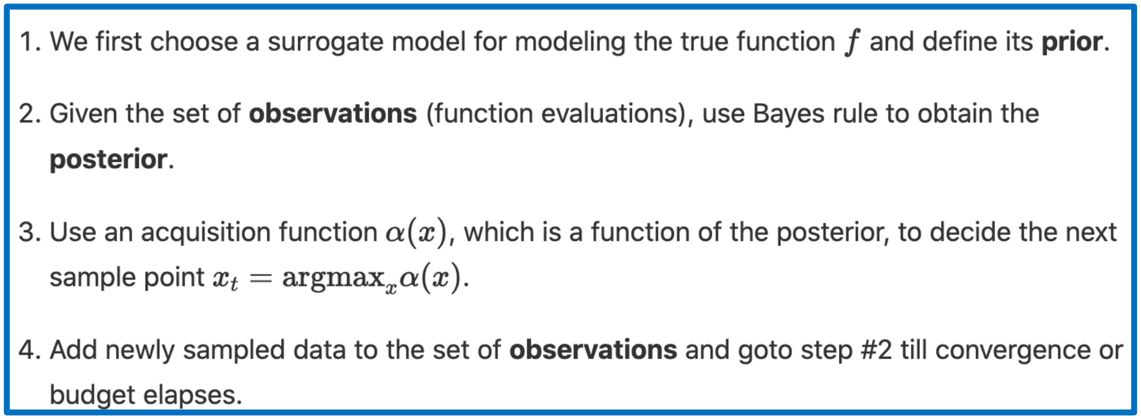
- Input
  - A continuous variable: x $x \in \mathrm{R}^d$
  - A function: f(x) $f: \mathrm{R}^d \rightarrow R$
    - f is a black box for which detail, closed form or gradient is unknown
    - f is expensive to evaluate
    - Evaluations of f(x) = y may be noisy
- Output:
  - $x^* = \arg\min_x f(x)$



Source: Agnihotri & Batra, "Exploring Bayesian Optimization", Distill, 2020.

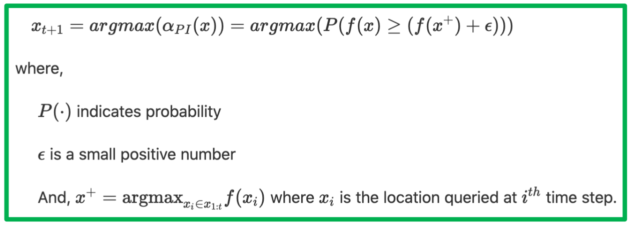
Details of Solving

- Acquisition function: heuristics about how desirable it is to evaluate a data point x_i, based on our present model
- Update by Bayes rules
  - At every step, a model of estimates and uncertainty at each point is updated using Bayes' rule



1. We first choose a surrogate model for modeling the true function f and define its **prior**.
2. Given the set of **observations** (function evaluations), use Bayes rule to obtain the **posterior**.
3. Use an acquisition function $\alpha(x)$, which is a function of the posterior, to decide the next sample point $x_t = \text{argmax}_x \alpha(x)$.
4. Add newly sampled data to the set of **observations** and goto step #2 till convergence or budget elapses.

Acquisition function



$$x_{t+1} = argmax(\alpha_{PI}(x)) = argmax(P(f(x) \geq (f(x^+) + \epsilon)))$$

where,

$P(\cdot)$ indicates probability

ϵ is a small positive number

And, $x^+ = \text{argmax}_{x_i \in x_{1:t}} f(x_i)$ where x_i is the location queried at i^{th} time step.

Recall: Bayes Theorem

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}$$

H: Hypothesis, E: Evidence

BO Applications

- Hyperparameter tuning – Auto-AI
- Mining industry
- Sensor placement
- …

Exercise and Code

- Bayesian Optimization

  - Code: https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l18-learning-agent/bayesian-optimization.ipynb

Source: Russell & Norvig, AI: A Modern Approach

References: Bayesian Optimization

- Description:
  - https://distill.pub/2020/bayesian-optimization/
  - https://machinelearningmastery.com/what-is-bayesian-optimization/

- Papers
  - A Tutorial on Bayesian Optimization, Peter I. Frazier, https://arxiv.org/abs/1807.02811, 2018
  - Taking the Human Out of the Loop: A Review of Bayesian Optimization
    B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N.d. Freitas.
    Proceedings of the IEEE, Vol 104(1), pp. 148-175. 2016.
    DOI: 10.1109/JPROC.2015.2494218
  - A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning
    E. Brochu, V. M. Cora, N. De Freitas.
    CoRR, Vol abs/1012.2599. 2010.

- Scikit: https://scikit-optimize.github.io/stable/auto_examples/bayesian-optimization.html

Lecture 18: Concluding Comments

- We looked at a learning agent
- Reinforcement learning method
  - Various variations
- Bayesian Optimization

Concluding Segment

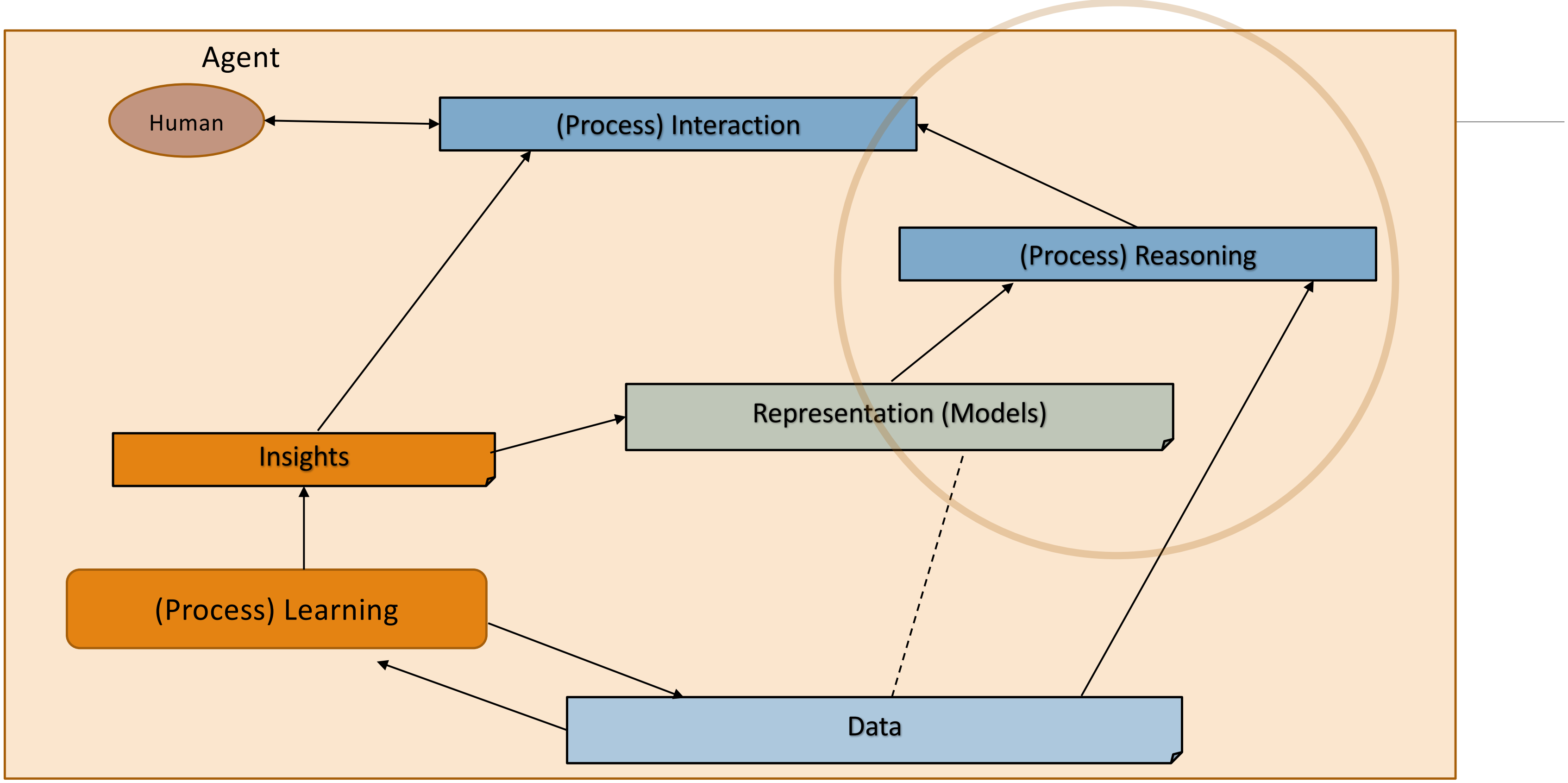
Upcoming Classes

15	**Mar 4 (Th)**	**Reasoning and Search**	**Semester - Midpoint**
16	Mar 9 (Tu)	Agent – Optimization	
17	Mar 11 (Th)	Agent – Handling Uncertain World	
18	Mar 16 (Tu)	Agent – Learning	
19	Mar 18 (Th)	Text: Data Prep (NLP)	Quiz 3
20	Mar 23 (Tu)	Text: Analysis - Supervised (NLP)_	
21	Mar 25 (Th)	Review, Paper presentations, Discussion	
22	Mar 30 (Tu)	Text: Advanced – Summarization, Sentiment	
23	Apr 1 (Th)	Text: Visualization, Explanation	
24	Apr 6 (Tu)	Multimodal Agents: Structured+Text: Examples	
25	Apr 8 (Th)	Case Study 1: Water	Quiz 4
26	Apr 13 (Tu)	Case Study 2: Finance	

Focus on Integrated Agent Behavior (Lectures 17, 18)

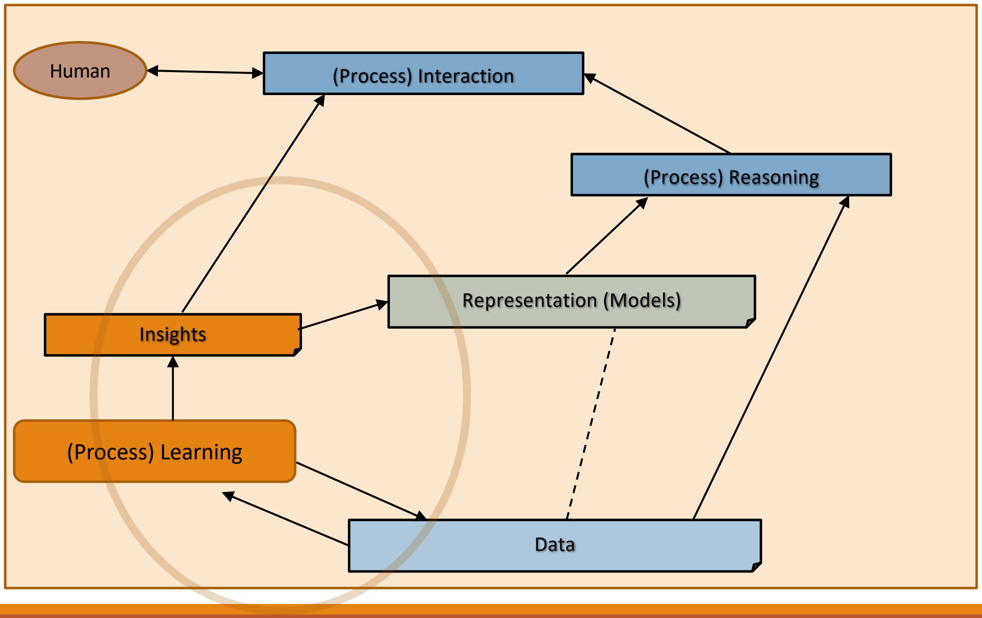
About Next Lecture – Lecture 18

Relationship Between AI Processes



Lecture 18: Text Analysis

- What is text ?
  - Multi-lingual
- Nature of analysis possible
- How it complements numerical analysis
- Quiz 3


Human
(Process) Interaction
(Process) Reasoning
Representation (Models)
Insights
(Process) Learning
Data