Trustworthy AI Autonomy
M5-2 Trustworthy RL-Interpretability

Ding Zhao
Assistant Professor
Carnegie Mellon University

2022 @ Ding Zhao
We are on the cusp to revolute the way to make machines

Connected
By complex structures

Evolving
In a self-supervised way

Sharing
With blackboxes and uncertainty
Contents

• Hierarchical AI structures

• Trees
  • Decision trees
  • Random tree/forests
  • Monte Carlo Tree search, Alpha Go

• Hierarchical RL
  • Manager-worker
  • Option/Semi-MDP

• Hierarchical structures in Meta learning
  • Neural Processes
Generalizations of concepts

• Human children learning names for object concepts routinely make strong generalizations from just a few examples. The same processes of rapid generalization can be studied in adults learning names for novel objects created with computer graphics.

• Given these alien objects and three examples (boxed in red) of “tufas” (a word in the alien language), which other objects are tufas? Almost everyone selects just the objects boxed in gray.

Generalizations of concepts

• Learning names for categories can be modeled as (Bayesian) inference over a tree-structured domain representation.

• Objects are placed at the leaves of the tree, and hypotheses about categories that words could label correspond to different branches.

• Branches at different depths pick out hypotheses at different levels of generality.
Some typical graph structures

Human may not always be able to generalize


https://www.medicalnewstoday.com/articles/baby-sticking-tongue-out
Graph grammars to describe knowledge

• A graph $G$ is a set of nodes (vertices) connected by directed/undirected edges.
• This is a very flexible data structure
  • If there are no edges, then it becomes a set.
  • A tree is an undirected graph in which any two vertices are connected by exactly one path.
  • A forest is an undirected graph in which any two vertices are connected by at most one path.

(A) Chain  (B) Linear bus  (C) Tree
(D) Ring  (E) Hub-and-spoke  (F) Fully connected mesh
(G) Partial mesh  (H) Multiple incomplete networks  (I) No connections
Decision Trees

https://medium.com/behavior-trees-for-path-planning-autonomous/behavior-trees-for-path-planning-autonomous-driving-8db1575fec2c
Decision Trees vs Random Forests

• Issues of decision trees: overfitting
• Random forests could avoid this by
  • training with a random subset of data (bootstrapping)
• Randomly select a subset of attributes
• Take an aggregation of results

* Bootstrapping the data using the aggregation to make a decision is called bagging

Table:

<table>
<thead>
<tr>
<th>Chest Pain</th>
<th>Good Blood Circ.</th>
<th>Blocked Arteries</th>
<th>Weight</th>
<th>Heart Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>125</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>180</td>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>210</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>167</td>
<td>Yes</td>
</tr>
</tbody>
</table>

https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d
https://youtu.be/J4Wdy0Wc_xQ
Decision Trees vs Random Forests

• Issues of decision trees: overfitting

• Random forests could avoid this by
  • training with a random subset of data (bootstrapping)
  • Randomly select a subset of attributes
  • Take an aggregation of results

* Bootstrapping the data using the aggregation to make a decision is called **bagging**
How to grow a tree to search: Random Tree

- A Random Tree selects a node at random from the tree and adds an edge in a random direction.

https://www.wolframcloud.com/objects/demonstrations/RapidlyExploringRandomTreeRRTAndRRT-source.nb
Rapid Random Tree (RRT)

• A RRT first selects a random goal point, then tries to add an edge from the closest node in the tree toward the goal point.

https://www.wolframcloud.com/objects/demonstrations/RapidlyExploringRandomTreeRRTAndRRT-source.nb
Rapid Random Tree Star (RRT*)

- RRT* improves this by rewiring the tree to form shortest paths.
A*: Random tree search with heuristic

- \( f(n) = g(n) + h(n) \)
  - Where \( f(n) \) = total estimated cost of path through node \( n \)
  - \( g(n) \) = cost so far to reach node \( n \)
  - \( h(n) \) = estimated cost from \( n \) to goal. This is the heuristic part of the cost function.

https://qiao.github.io/PathFinding.js/visual/
https://brilliant.org/wiki/a-star-search/
#:~:text=A*%20(pronounced%20as%20%22A,or%20points%20on%20the%20graph.&text=With%20A*%20a%20robot%20would,diagram%20on%20the%20right%20below.
Monte Carlo Tree Search

Selection → Expansion → Simulation → Backpropagation

Tree Policy

Default Policy

Monte Carlo Tree Search case study: Alpha Go

Contents

• Hierarchical AI structures

• Trees
  • Decision trees
  • Random tree/forests
  • Monte Carlo Tree search, Alpha Go

• Hierarchical RL
  • Manager-worker
  • Option/Semi-MDP

• Hierarchical structures in Meta learning

• Neural Processes
Hierarchical Decision-Making ( Autonomous Vehicles)
Hierarchical Decision-Making (home robots)

![Diagram showing the process of cleaning the house with a robot]

https://phillipi.github.io/6.882/2020/notes/The%20problem%20of%20learning%20for%20long%20horizons/FeudalRL.pdf
Hierarchical Reinforcement Learning

• Benefits
  • Efficiency/Scalability
  • Transfer/reusability of skills
  • Explainability/maintenance

• Different hierarchical frameworks
  • Manager-submanager: manager sets subgoals and rewards for sub-managers
    • Feudal RL; FeUdal Networks (FUNs)
  • Option: no explicit subgoals learn and discover options
    • Option-Critic; Meta Learning Shared Hierarchies (MLSH)

Higher level control
Sub-goals (where to drive)
Lower level control
Braking pressures commands
Feudal Reinforcement Learning

• Good concept
• Was not widely used
FeUdal Networks (FUNs)

FeUdal Networks (FUNs) Empirical Results

Hierarchical Reinforcement Learning

• Benefits
  • Efficiency/Scalability
  • Transfer/reusability of skills
  • Explainability/maintenance
• Different hierarchical frameworks
  • Manager-submanager: manager sets subgoals and rewards for sub-managers
    • Feudal RL; FeUdal Networks (FUNs)
  • Option: no explicit subgoals learn and discover options
    • Option-Critic; Meta Learning Shared Hierarchies (MLSH)

Higher level control

Sub-goals (where to drive)

Lower level control

Braking pressures commands
Meta Learning Shared Hierarchies

Options: Temporal abstraction in RL

- MDP + Options = Semi-MDP

- Semi-Markovian
  - Transition probability:
    - \( p(s', \tau | s) = p(s' | s) p(\tau | s) \)
    - where \( \tau \) indicates the time to transition

Example 1: Traffic Primitives

Traffic primitive is referred to the representation of fundamental building blocks of the traffic environment in spatiotemporal space.

Toyota (PI) “Extracting Traffic Primitives from Millions of Naturalistic Driving Encounters -- A Synthesized Method based on Nonparametric Bayesian and Deep Unsupervised Learning”

Previous methods:
- Subjectively-selected scenarios

Traffic Primitive:
- Segment/cluster similar traffic scenes automatically using unsupervised learning
- Objectively-selected scenarios

[Wang, Zhao, 'Extracting Traffic Primitives Directly from Naturalistically Logged Data for Self-Driving Applications, ICRA, 2018]
Extracting driving primitives

Nonparametric Bayesian learning (HDP-HMM)

[Wang, Zhang, Zhao, 'Understanding V2V Driving Scenarios through Traffic Primitives', IEEE ITS, 2020]
Example 2: DPGP-MBRL

- Use Model-based RL with an infinite mixture of Gaussian Processes as the learned dynamics model.

- Do not require pre-training <- GP.

- Handle substantially different tasks <- mixture model.

- Online setting with streaming data <- streaming variational inference
Contents

- Hierarchical AI structures
- Trees
  - Decision trees
  - Random tree/forests
  - Monte Carlo Tree search, Alpha Go
- Hierarchical RL
  - Manager-worker
  - Option/Semi-MDP

- Hierarchical structures in Meta learning
- Neural Processes
Meta learning

- Meta-learning (aka “learning to learn”) consists in training a model on various different tasks so that it can solve new learning tasks more efficiently using only a small number of training samples.

- Meta-RL is meta-learning on reinforcement learning tasks. After trained over a distribution of tasks, the agent is able to solve a new task by developing a new RL algorithm with its internal activity dynamics.
Why Meta reinforcement learning

- Task imbalanced environment is prevalent for safety critical applications

Why do we care about generative models

* Data source: California Department of Motor Vehicle 2019 disengagement report
Machine learning with big data
Machine learning with small/imbalanced dataset

Inductive biases
- Preprocessing
- Feature engineering
- Data augmentation
- Model architectures
- Regularisation
Meta-learning, learning-to-learn

Inductive biases
Single-task learning

Training Data \( \{(x_i, y_i)\}_{i=1}^{n} \)

Parameter \( \theta \)

Test Data \( \{(x_{n+i}, y_{n+i})\}_{i=1}^{m} \)

Loss

\(20\%\)
Multi-task learning

Training Data \( \{(x_i, y_i)\}_{i=1}^{n} \)

Shared parameter \( \eta \)

Parameter \( \theta \)

Test Data \( \{(x_{n+i}, y_{n+i})\}_{i=1}^{m} \)

Loss

Task 1

Task 2

Task 3

20%

0%

20%
Meta-Learning

Meta-Learning — an idiosyncratic tutorial, Yee Whye Teh
Neural Processes

- Learn function approximation from data directly
- Can model complex functions with few functional restrictions
- Fast evaluation at test time

Regression algorithms

Neural Networks
- Learn distribution over functions > Flexible at test time
- Have a measure of uncertainty given observations at test-time

Gaussian Processes

Neural Processes
Neural Processes
Generate Coherent Samples
Generate Coherent Samples

Saturday, July 14 - 16:10
TADGM Workshop
Skepticism on Hierarchical RL

• “Surprisingly, we find that most of the empirical benefit of hierarchy in our considered settings can be attributed to improved exploration.”

• “These proposed exploration methods enable non-hierarchical RL agents to achieve performance competitive with state-of-the-art HRL. Although our analysis is empirical and thus our conclusions are limited to the tasks we consider, we believe that our findings are important to the field of HRL.”

• “Our findings reveal that only a subset of the claimed benefits of hierarchy are achievable by current state-of-the-art methods, even on tasks that were previously believed to be approachable only by HRL methods.”
Summary

• Hierarchical AI structures

• Trees
  • Decision trees
  • Random tree/forests
  • Monte Carlo Tree search, Alpha Go

• Hierarchical RL
  • Manager-worker
  • Option/Semi-MDP

• Hierarchical structures in Meta learning
  • Neural Processes
Additional reading materials

• High level intro to HDP

• Frans, Kevin, Jonathan Ho, Xi Chen, Pieter Abbeel, and John Schulman. 2017. “Meta Learning Shared Hierarchies.” ICLR 2018
Trustworthy AI for Safety-Critical Applications

Rare event learning (IS, Cross Entropy)
Adversarial machine learning (FGSM)
GAN
Imitation learning (DAGAR)
Model-based Markovian Decision making (CEM, iLQR, RNN, LQR, MPC, Meta RL)
Model-free Markovian Decision making (DQN, REINFORCE, A2C, DDPG)

Summary of TAIAT:
12 lectures
+25 papers
+Final project