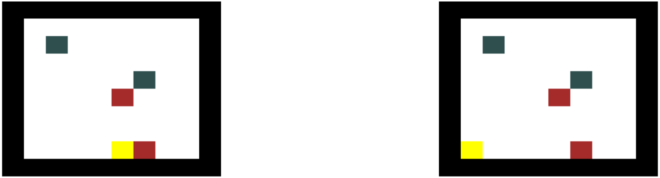
Causal Structure Learning

June 24, 2020

Motivation

- Human beings learn rich causal models and use abstract causal concepts to transfer knowledge in the similar domains.
- For example, human beings can play two variants of a video game with similar causal dynamics but different perceptual features. Abstract causal concepts can help in the transfer of knowledge in this scenario.
- Abstraction and disentanglement of dynamics from variant perceptual features is important for out-of-distribution generalization to novel but similar domains.

Motivating Example: Triggers Game[1]



(a) Source domain

(b) Target domain

Figure: Example for Proof of concept

[1] Self-Attentional Credit Assignment for Transfer in Reinforcement Learning

Motivating Example: Triggers Game [1]

- Colors (perceptual features) characterize the causal behavior: Yellow box denotes the agent, white boxes denote the free space, black boxes denote the wall, two gray boxes denote the switches, ,and two red boxes are the doors.
- Goal: Activate the two gray switches and then open the two red doors. Red doors are only source of rewards $\{+1, -1\}$.

Motivating Example: Triggers Game [1] Contd.

Causal Concepts of the Game

- Agent can take one of the four possible actions $\{north, south, east\ or\ west\}$
- Hitting black walls doesn't change the agent's position.
- Hitting free space increases or decreases the agent's x-position or y-position by 1, depending on the action.
- Gray boxes disappear after the agent hits them and behave similarly to white boxes.
- Red doors disappear and provides $+1$ reward only after the agent has activated both of the gray switches.
- If any one of the gray box is not hit, hitting on the red box gives -1 reward and agent's position remain unchanged.

Markov Decision Processes: Preliminaries and Background

- An MDP is a five-tuple $(\mathcal{S}, \mathcal{A}, T, \mathcal{R}, \gamma)$
- $\mathcal{S}$ is a set of states
- $\mathcal{A}$ is the set of actions
- $T(s^{(t+1)}|s^{(t)}, a^{(t)})$ is the probability of transforming from state $s^{(t)} \in \mathcal{S}$ to $s^{(t+1)} \in \mathcal{S}$ after action $a^{(t)} \in \mathcal{A}$
- $\mathcal{R}(r^{(t+1)}|s^{(t)}, a^{(t)})$ is the probability of receiving the reward $r^{(t+1)} \in \mathbb{R}$ after executing action $a^{(t)}$ while in state $s^{(t)}$
- $\gamma \in [0, 1]$ is the rate at which future rewards are exponentially discounted

Object-oriented representation: OO - MDP

- To construct the causal graph depicting causal relationships between different objects, it is more convenient to represent state $s^{(t)}$ as set of n objects $\{o_1^{(t)}, o_2^{(t)}, o_3^{(t)} \dots o_n^{(t)}\}$ each with m attributes $\{\alpha_{i1}, \alpha_{i2}, \alpha_{i3}, \dots \alpha_{im}\}$.
- Example of an attribute α_{ij} is {color, position etc}. For yellow object o_1 in 7b, attributes are {color: yellow, x:1, y:1}.

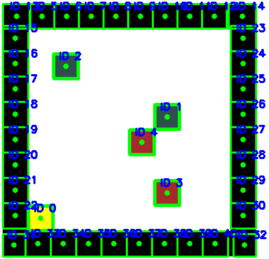


Figure: Object Oriented Representation

Causal Interactions between objects

We assume that the causal effect of interaction resulting from an action taken by the agent (denoted by object o_i) leads to change in the j^{th} attribute (α_{ij}^{t}) of object o_i, defined by the following structural causal model.

$$\alpha_{ij}^{(t)} = f_{a^{(t)}}(\alpha_i^{(t-1)}, \alpha_r^{(t-1)})$$

where $\alpha_i^{(t-1)}$ are the past attributes of the object o_i, $\alpha_r^{(t-1)}$ are the attributes of neighbors of i^{th} object within radius of r and $a^{(t)}$ is the action taken at time t.

Causal Interactions between the objects

Example of causal interaction in trigger game:
Let x-position of yellow object is $\alpha_{00} = 1$, y-position of yellow object is $\alpha_{01} = 1$.

- $$\alpha_{00}^{t} = f_{left}(\alpha_{0}^{(t-1)}, \alpha_{black}^{(t-1)}) = \alpha_{00}^{(t-1)}$$

  [Taking left action and hitting the wall doesn't change the x or y position of the agent]

- $$\alpha_{00}^{t} = f_{right}(\alpha_{0}^{(t-1)}, \alpha_{white}^{(t-1)}) = \alpha_{00}^{(t-1)} + 1$$

  [Taking right action and hitting the wall increments the x-position of the agent by 1]

Causal Structure Learning [Linear Case]

To learn the causal structure, we need to first learn the causal graph between attributes of object i at position t and object i and neighboring objects at $t-1$. Basically which attributes at $t-1$ are *preconditions* for the change in the attribute $\alpha_{ij}^{(t)}$.

- We formulate the structure of the toy problem as linear structural equation as below.
- Assume there are N objects, each with M attributes and $d = M \times N$. Then, $\alpha^t, \mathbf{z} \in \mathbb{R}^d$ and $\mathbf{w} \in \mathbb{R}^{d \times d}$.

$$\hat{\alpha}^{\mathbf{t}} = \mathbf{w}\alpha^{t-1} + \mathbf{z}_{t-1}$$

, where $w \in \mathbb{R}^{M \times N}$ is the adjacency matrix of the directed graph and z is the noise vector.

Step 1: Bayesian Structure Learning as Continuous Optimization

Recent work in formulating bayesian structure learning problem as continuous optimization problem [1].
Objective function:

$$\min_{\mathbf{w} \in \mathbb{R}^{d \times d}} f(\mathbf{w}) + \lambda ||\mathbf{w}||_1$$

where

$$f(\mathbf{w}) = l(\mathbf{w}; \alpha) + \rho |h(\mathbf{w})|^2 + \alpha h(\mathbf{w})$$

$$l(\mathbf{w}; \alpha) = ||\alpha^t - \hat{\alpha^t}||_2$$

$$h(\mathbf{w}) \textit{enforces the acyclic constraint on the graph}$$

[1] DAGs with NO TEARS: Continuous Optimization for Structure Learning

Step 1: Bayesian Structure Learning as Continuous Optimization (Contd.)

- α : $[bias, ax^{t-1}, ay^{t-1}, ac^{t-1}, ux^{t-1}, uy^{t-1}, uc^{t-1}, dx^{t-1}, dy^{t-1}, dc^{t-1}, \mathrm{lx}^{t-1}, ly^{t-1}, lc^{t-1}, rx^{t-1}, ry^{t-1}, rc^{t-1}, ax^{t}, ay^{t}]$

True Bayesian Graph

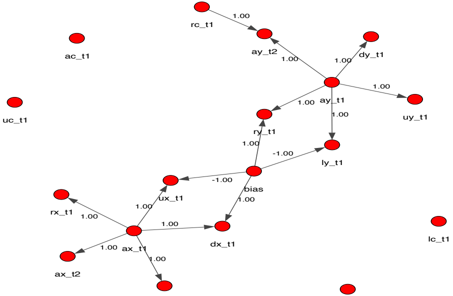


Figure: True Bayesian Graph for action = down

Step 1: Bayesian Structure Learning as Continuous Optimization (Contd.)

- α : $[bias, ax^{t-1}, ay^{t-1}, ac^{t-1}, ux^{t-1}, uy^{t-1}, uc^{t-1}, dx^{t-1}, dy^{t-1}, dc^{t-1}, \mathrm{lx}^{t-1}, ly^{t-1}, lc^{t-1}, rx^{t-1}, ry^{t-1}, rc^{t-1}, ax^{t}, ay^{t}]$

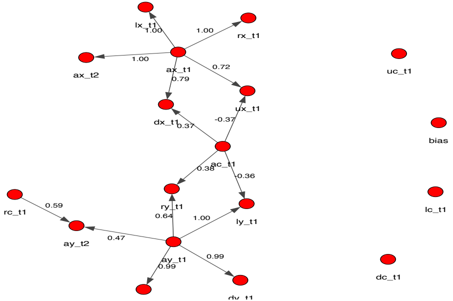


Figure: Learned Dynamic Bayesian Graph for action = right

Existing Approaches for Causal Structure Learning

- Existing Methods for (bayesian) structure learning: Structure learning based on conditional independence, Score-based structure learning, Bayesian Model Averaging
- **Current Focus:** Use linear programming [2] based method for structure learning: Also used by [3] for zero-shot transfer learning but no explicit structure mapping across domains.

[2] Learning Bayesian Network Structure using LP Relaxations

[3] Schema Networks: Zero-shot Transfer with a Generative Causal Model of Intuitive Physics

Schema Learning using LP Relaxations

$$y = f_W(X) = \overline{\overline{X}W}\vec{1}$$

Loss:

$$\min_{W\in\{0,1\}^{D'\times L}} \frac{1}{D}|y - f_W(X)| + C|W|$$

- $X \in \{0,1\}^{D\times D'}$ is binary matrix, where $D = NT$ and $D' = MR$. N is total umber of objects over an episode of time T, M is number of attributes and R is maximum number of neighbors of each object.
- $y \in \{0,1\}^{D\times M}$ is the binary matrix containing actual attributes at time t which we want to predict using object interactions from X^{t-1}.

Schema Learning using LP Relaxations

Algorithm for Schema Learning:

$$y = f_W(x_n) = \overline{\overline{X}W\vec{1}} = (1 - x_i)w = singleoutput$$

Input: Input vectors $\{x_n\}$ for which $f_W(x_n) = 0$ (current schema network predicts 0), and the corresponding output scalars y_n. In first iteration, input X will be $(T, N, M * 5)$ matrix and Y will be (T, N, M) matrix.

1. **Find a cluster of input samples** that can be solved with a single (relaxed) schema while keeping perfect precision (no false alarms, (e.g. output 1 when it needs to be zero)). Select an input sample and put it in the set "solved", then solve the LP.

   $$min_{w \in [0,1]^{M*R}} \sum_{n:y_n=1} (1 - x_n)w$$

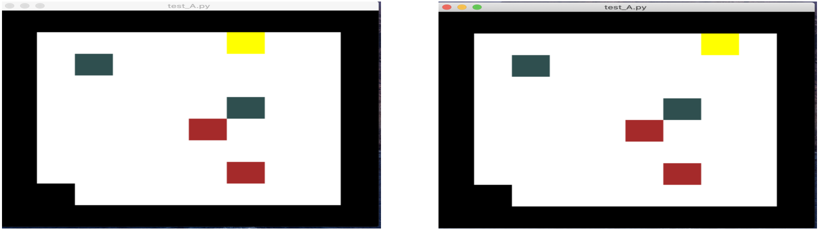
   1. prediction for non-active attribute remains zero (¿1). No false alarms.

      $$s.t.(1 - x_n)w > 1 \forall_{n:y_n=0}$$

   2. For solved ones the prediction remains zero (don't mess up with the previous predictions, basically) Perfect precision.

Drawbacks of Schema Learning using LP Relaxations

1. The input representation is binarized and position-specific. The causal graph is specific to each binarized position of the objects, and thus difficult for learning high-level abstract structure of the causal graphs for similar spatial configurations of the objects. For example, assuming that the transition dynamics follow the markov property, below two spatial configurations which should have similar causal structure but the model learns position-specific rules.



(a) Source domain

(b) Target domain

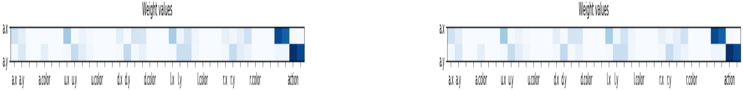
Figure: Configurations with similar causal structure for taking different actions.

Drawbacks of Schema Learning using LP Relaxations

1. Not all causal parents are captured for learning the causal structure. For Example, white color of the bottom object is not considered as the cause of change in the position of yellow object.
   Input: [’down’, ’yellow’, 7.0, 2.0, ’white’, 7.0, 1.0, ’white’, 7.0, 3.0, ’white’, 8.0, 2.0, ’white’, 6.0, 2.0]
   Output: [’yellow’, ’8’, ’2’]
   Learned causes: [’down’, ‘’, 0.0, 0.0, ‘’, 0.0, 0.0, ‘’, 0.0, 0.0, ‘’, 8.0, 0.0, ‘’, 0.0, 2.0]
   Ground Truth: [’down’, ’yellow’, 7.0, 0, ‘’, 0, 0, ‘’, 0, 0, ’white’, 8, 0, ”, 0, 0]

Linear Prediction of next state and feature selection

1. Linear non-sparse prediction model for all data



(a) Source domain

(b) Target domain

Figure: Configurations with similar causal structure for taking different actions.

Desired characteristics of causal structure learning model

1. State and Temporal Abstraction in learning the causal structure.
2. Captures all the possible causes, and thus learning the causal characteristics of the objects based on learning the behavior.
3. Causal structure should be learned from the effect calculated using absence and presence of the cause, rather than associational relationship between the potential cause and effect.

Next To Dos

1. Understand bayesian causal structure induction.
2. Understand general relationship between Reinforcement learning and model based RL and causal models.
3. Understand Factored MDPs. How OO MDPs are related to Factored MDPs?

State Abstraction

Causal Structure Mapping

Can we use W^{source} learned from the source domain to learn W^{target}. We assume that we are using random policy in the source domain and limited interventions in target domain. Can we infer W^{target} by mapping f_{red}^{source} against f_{gray}^{target} based on structure similarity?

[4] Structure Mapping Theory

[5] Manifold Alignment

Next Steps

- Finish implementation of structure learning module
- Formalize structure mapping
- Iterate between (1) and (2).
- Comparison against Vanilla RL/ Progressive Neural Networks benchmarking. Please note that the transfer learning in RL is still a new area with limited benchmarks.

Appendix

Dynamics of pendulum system with different masses.

theta_transition.png

(a) Angular position

theta_dot_transition.png

(b) Angular Velocity

Figure: Dynamics of pendulum with different mass

Inputs and Feedback