Causal Discovery with Adaptable AI Agents

Matteo Ceriscioli, Karthika Mohan

School of Electrical Engineering and Computer Science (EECS)
Oregon State University
Corvallis, OR 97331 USA
{ceriscim, karthika.mohan}@oregonstate.edu

Abstract

Understanding the connection between robustness to distribution shifts and learning the causal model of an environment is an important area of study in AI. While previous work has established this link for single agents in unmediated decision tasks, many real-world scenarios involve mediated settings where agents influence their environment. We demonstrate that agents capable of adapting to distribution shifts can recover the underlying causal structure even in these more dynamic settings. Our contributions include an algorithm for learning Causal Influence Diagrams (CIDs) using optimal policy oracles, with the flexibility to incorporate prior causal knowledge. We illustrate the algorithm's application in a mediated single-agent decision task and in multi-agent settings. We also show that the presence of a single robust agent is sufficient to recover the complete causal model and derive optimal policies for all the other agents operating in the same environment.

Introduction

A defining characteristic of human intelligence is the ability to adapt seamlessly to new environments and inputs (Piaget 1936). In AI, the goal is to design agents capable of adapting effectively to environments that differ from their initial training environment. This problem has been extensively studied through diverse methodologies including domain adaptation (Ben-David et al. 2006), transfer learning (Pan and Yang 2010; Zhuang et al. 2021), federated learning (Konečný et al. 2016), and transportability (Pearl and Bareinboim 2011), each addressing distinct flavors of the problem. Recent research (Richens and Everitt 2024) has demonstrated that for agents to adapt seamlessly to new domains, they must learn causal models i.e., understand how the world operates. However, the results in Richens and Everitt (2024) rely on the strong assumption of no mediation (Pearl 2009), meaning the agent's actions cannot have an effect on the utility via environment states. In contrast, many real-world AI applications involve tasks where mediation exists. For example, an autonomous car navigating from point A to point B, may affect lane occupancy and, in turn, traffic flow and the behavior of other drivers. Similarly, a robot in an industrial plant might interact with tools, move through space, and transform products to complete its task. In this work, we demonstrate that the assumption of unmediated tasks is unnecessary.

We show that agents robust to domain shifts can learn and encode the causal model even in the presence of mediation. Additionally, we present an algorithm to learn the causal model of the environment using an agent that is adaptable to domain shifts. We also outline how to incorporate prior knowledge into the causal model and offer insights into the implications of our findings for multi-agent environments.

Problem setup

We denote the set of parents of a node X as Pa_X, the set of children as Ch_X, and instantiations of random variables in lower-case. To model the causal relationships in the environment where agents operate, we use Causal Influence Diagrams (CIDs) (Heckerman 1995; Everitt et al. 2021). Similar to Influence Diagrams (Howard and Matheson 1984), CIDs are commonly used to reason about decision-making tasks. CIDs further assume that the graph encodes the causal relationships between the nodes. A CID is a tuple $M = (G = \{V, E\}, P)$, where P is a joint probability distribution compatible with the conditional independences encoded in G. The set of nodes V is partitioned into chance nodes C, decision nodes D, and utility nodes U. There is a real function $U(pa_U) \mapsto \mathbb{R}$ associated with every utility node U. An example of CID can be found in Figure 1. Throughout the paper, we assume there not exists $d \in \mathrm{Im}(D)$ s.t. $d \in \arg\max_d U(d, x)$ for all $x \in \mathrm{Im}(Pa_U \setminus \{D\})$, this implies domain dependence (Richens and Everitt 2024), meaning that for the tasks and environments we consider, no single policy can be optimal across all possible distribution shifts. Domain dependence excludes trivial cases where distribution shifts do not affect the optimal policy. In such trivial scenarios, any optimal agent in one domain remains optimal under any shift, so optimality automatically implies robustness. In these cases, the agent does not need to learn a causal model of the environment to be robust against distribution shifts. Following (Richens and Everitt 2024), we represent domain shifts as mixtures of local interventions. Given a random variable X with $x_1, \ldots, x_n$ as possible outcomes, a local intervention on X is a function $\sigma : x_i \mapsto f(x_i)$ that maps each outcome x_i to a new outcome $f(x_i)$. A mixture of local interventions is a convex combination $\sigma^* = \sum_i p_i \sigma_i$ of local interventions σ_i, where each coefficient p_i repre-

sents the probability that σ_i is used to map the outcome for X. We use optimal policy oracles to formalize the agent's understanding of optimal behavior under distribution shifts. Given a set of interventions Σ, an optimal policy oracle is a map $\Pi^*_\Sigma : \sigma \mapsto \pi_\sigma(d|pa_D)$ for $\sigma \in \Sigma$, where $\pi_\sigma(d|pa_D)$ is the optimal policy under the distribution shift induced by the intervention σ.

Main results

In this section, we describe an algorithm for learning the CID using the optimal policy oracle along with a sketch of the proof establishing its soundness[1]. In Example 1, we show this algorithm can be applied to learn a simple environment with a single agent. Subsequently, in Example 2, we demonstrate how to adapt this approach to handle multi-agent environments.

LearnCID Algorithm

The LearnCID algorithm operates under the following assumptions: The CID is faithful and sufficient, it contains a single decision node D and a single utility node U. The Markov blanket of the decision node D is known, and D is a parent of U. We also assume all chance nodes are ancestors of U or D, because otherwise, they do not play a role in the decision task and can be pruned. The utility function is fully specified i.e., all parents of U are known. Additionally, prior knowledge in the form of the causes of a subset of chance nodes is available. Specifically, prior knowledge about direct causes (parents) is available for a subset of chance nodes, denoted as V_{kwn} i.e., for every $C_i \in V_{\text{kwn}}$, the set of parent nodes of C_i is known.

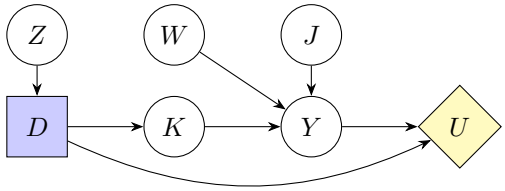


Figure 1: A CID that represents a mediated decision task.

The algorithm performs a breadth-first traversal on chance nodes that are not children of D and are not in the set V_{kwn}. The traversal starts from the parents of U, then we enqueue the remaining parents of D after visiting all nodes with a path to U that does not include D. In the CID in Figure 1, the traversal starts from Y and once the queue is empty, Z is enqueued as a parent of D. For each chance node X encountered during the search, and for each possible instantiation x, we define a local intervention $f(X) \leftarrow \begin{cases} x, & \text{if } X = x \\ x', & \text{otherwise} \end{cases}$ where x' is an arbitrary outcome for X different from x. We also define the following family of local interventions:

$$\sigma_Y(c) \leftarrow \{do(Y = y, C_X = c, X = f(X))|y \in \text{Im}(Y)\} \quad (1)$$

[1]The extended version of this paper will contain the full proof with details.

where C_X represents the set of all chance nodes except X and those along a directed path from X to either U or D. Y is a variable in C_X, and c is an outcome for the variables in $C_X \setminus \{Y\}$. For any local intervention $\sigma \in \sigma_Y$, let d be the deterministic optimal decision under the shift induced by σ. By assuming there not exists $d \in \text{Im}(D)$ s.t. $d \in \arg\max_d U(d, x)$ for all $x \in \text{Im}(Pa_U \setminus \{D\})$, there is a hard intervention σ' such that d is no longer optimal. Let d_2 be the deterministic optimal decision under σ'. Considering the mixture $\sigma(q) \coloneqq q\sigma + (1-q)\sigma'$, there exist a value q_{crit} for q such that d_2 and another deterministic decision d_1 are both optimal. Using Algorithm 1 from (Richens and Everitt 2024), referred to here as ALGq_{crit}, we can compute q_{crit}, d_1, d_2, and pa'_U, the value of U's parents under the hard intervention σ'. Let $C_1, \ldots, C_k$ be the chance nodes in the directed internal path from X to U or D we are considering. If $C_1 \in Pa_U$ let $\mathcal{C} \coloneqq \{C_1, \ldots, C_k\}$ otherwise let $\mathcal{C} \coloneqq \{C_2, \ldots, C_k\}$. For both x and x', we compute:

$$\beta(x) \coloneqq \sum_{c \in \text{Im}(\mathcal{C})} \prod_{i=1}^{k} P(c_i|pa_{C_i})[U(d_2, c) - U(d_1, c)] \quad (2)$$

Using this, we can compute $P(x|pa_X; \sigma)$ as:

$$P(x|pa_X; \sigma) = \frac{(1 - \frac{1}{q_{crit}})[U(d_2, pa'_U) - U(d_1, pa'_U)] - \beta(x')}{\beta(x) - \beta(x')} \quad (3)$$

If for some $\sigma_1, \sigma_2 \in \sigma_Y$, we find $P(x|pa_X; \sigma_1) \neq P(x|pa_X; \sigma_2)$, then Y must be a parent of X. Due to the specific form of interventions defined in Equation 1, it follows that $P(x|pa_X; \sigma) = P(x|pa_X)$. For example, for the CID in Figure 1, when considering $P(y|pa_Y; \sigma)$, $C_Y = \{Z, K, W, J\}$, assuming we are verifying if J is a parent of Y then the intervention σ would be $do(J = j, Z = z, K = k, W = w, Y = f(Y))$, and therefore $P(y|pa_Y; \sigma) = P(y|J = j, W = w, K = k)$ by the rules of do-calculus (Pearl 2009). Returning to the general case, since the parents and Conditional Probability Tables (CPTs) of all nodes $C_1, \ldots, C_k$ along the path from X to U or D are already known, all terms on the right side of Equation 2 are computable. By the end of the traversal, all chance nodes that are not children of D have been visited, and we have learned the parents and CPTs of each, completing the causal model. As shown in Example 2, this algorithm can also be applied in multi-agent settings. In such cases, we can assign each decision node a policy that preserves the faithfulness of the CID (i.e., the policy depends on the node's parents) and treat these nodes as chance nodes in V_{kwn}, then we can use the optimal policy oracle for the remaining decision node.

Example 1 - Single agent environment

Consider the CID in Figure 2, assume we know $X \in Ch_D$, $Y \in Pa_U$, and all the variables are binary. We also know $U \coloneqq 1$ if $D = Y$ and 0 otherwise, and an optimal policy oracle Π^*_Σ where Σ is the set of all mixtures of local interventions. The CPT for Y can be found in the table on the right side of Figure 2, but let us assume it is unknown. We

Algorithm 1: LearnCID

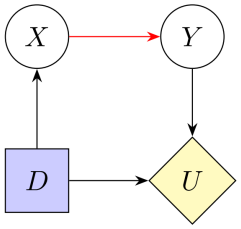
Input:
1. Nodes $V = \{\{D\}, \{U\}, C\}$
2. Known set of edges $\hat{E}$
3. Set of chance nodes with all known parents V_{kwn}
4. Number of samples N to estimate q_{crit}

Output: The CID's structure E', and the set of CPTs P for all the nodes in $C \setminus Ch_D$

visited $\leftarrow$ Dictionary(), $Q \leftarrow$ Queue()
visited[X]$\leftarrow$ True $\iff$ X is a chance node child of D
Enqueue in Q the parents of U not children of D.
while Q is not empty **do**
  $X \leftarrow Q$.dequeue()
  Path $\leftarrow$Set of chance nodes on a directed internal path from X to U, or to D if U is unreachable.
  $C_X \leftarrow$ Chance nodes that are not in "Path", and that are not known to be parents of X.
  $Z \coloneqq Pa_X$ **if** $X \in V_{knw}$ **else** $Z \coloneqq C_X$
  for each $x \in \text{Im}(X)$ **do**
    $x' \leftarrow$ A possible outcome for X different from x.
    $f(X) \leftarrow \begin{cases} x, & \text{if } X = x \\ x', & \text{otherwise} \end{cases}$
    for each $Y \in Z$ and each $c \in \text{Im}(C_X \setminus Y)$ **do**
      $\sigma_Y(c) \leftarrow$ As in Equation 1
      for each $\sigma \in \sigma_Y$ **do**
        $q_{crit}, d_1, d_2, pa'_U \leftarrow ALGq_{crit}(U, \Pi^*_\Sigma, N, \sigma)$
        $\beta(x), \beta(x') \leftarrow$ As in Equation 2
        $P(x|pa_X; \sigma) \leftarrow$ As in Equation 3
      if $\exists \sigma, \sigma' \in \sigma_Y$ s.t. $P(x|pa_X; \sigma) \neq P(x|pa_X; \sigma')$ **then**
        $\hat{E} \leftarrow \hat{E} \cup \{(Y, X)\}$
        if visited[Y]=False and $Y \notin Q$ **then**
          Q.enqueue(Y)
    for each pa_X outcome of parents of X **do**
      $P(x|pa_x) \leftarrow P(x, pa_x; \sigma)$ for any hard intervention σ compatible with pa_X
  visited[X]$\leftarrow$ True
  if Q is empty and there are still unvisited nodes **then**
    Enqueue the parents of D in Q
Return $E' \leftarrow \hat{E}$, P the set of CPTs

want to use Algorithm 1 to learn whether there is an edge between X and Y, and the CPT for Y. For ease of comprehension we summarize Algorithm 1, in the following steps:

1. Perform a breadth-first traversal starting with nodes that are parents of U and not children of D.
2. Define local interventions on X as in Equation 1, used to obtain both the CPT for Y and the set of its parents.
3. Estimate q_{crit} using ALGq_{crit} (Algorithm 1 in Richens and Everitt (2024))
4. Compute $P(Y = y_i|do(X = x_i))$
5. Repeat steps 2 to 4 for all configurations of y_i and x_i.
6. Deduce the set of parents of Y and its CPT.



X	Y	$P(Y\|X)$
0	0	0.7
0	1	0.3
1	0	0.1
1	1	0.9

Figure 2: An example of a single-decision/single-utility CID. On the right, the CPT for variable Y. The edge marked in red is unknown.

Following the aforementioned steps:

Step 1: In this example Pa_D is empty and $Pa_U = \{Y\}$, so we start the traversal from node Y. We also assume V_{knw} is empty. Observe that since Y is the only chance node that is not children of D the traversal will stop after processing Y. Since $Y \in Pa_U$, "Path" is empty and $C_Y = \{X\}$.

Step 2: Set $\sigma_0 \coloneqq do(X = 0)$, then in ALG$q_{crit}$ we use the oracle $\Pi^*_\Sigma(\sigma_0)$ to find the optimal decision $d_1 \coloneqq 0$. This is evident from the (unknown) CPT, because for $X = 0$ the probability that $Y = 0$ is higher than the one for $Y = 1$, since the utility is an AND operation between Y and D, the oracle returns the optimal decision $D = 0$ which will be equal to Y more often than $D = 0$.

Step 3: We obtain q_{crit} using ALGq_{crit}. It works by finding σ' such that the optimal decision is no longer $D = 0$. Since in this example D is binary, we already know the new optimal decision must be $D = 1$, in general, for this we can use the optimal policy oracle. We can find σ' by hard intervening on the parents of the utility node U, which is Y in this case, such that d_1 is no longer optimal. Specifically, we can define σ' as a hard intervention that sets Y to 1, therefore the new optimal decision is $d_2 \coloneqq 1$. Then, we define the mixture of local interventions $\sigma(q) \coloneqq q\sigma_0 + (1 - q)\sigma'$. We can sample q uniformly in the interval $[0, 1]$ N times and each time query the optimal policy oracle. Each time the oracle returns an optimal decision for the intervention σ' we increment a counter θ. Then $\frac{\theta}{N}$ is an unbiased estimate for q_{crit}. For this example, $q_{crit} = \frac{5}{7}$.

Step 4: We now compute $P(Y = 0|pa_Y; \sigma_0)$. We start with $Y = 0$ and $X = 0$, we first need to compute $\beta(Y = 0)$ and $\beta(Y = 1)$. This is the case where the node for which we are computing the CPT is the first on the directed path to the utility node. So the beta expressions are:

$$\beta(Y = 0) = U(d_2, 0) - U(d_1, 0) = -1 \tag{4}$$

$$\beta(Y = 1) = U(d_2, 1) - U(d_1, 1) = 1 \tag{5}$$

Following Equation 3:

$$P(y|pa_Y; \sigma) = \frac{(1 - \frac{7}{5})[U(d_2, 1) - U(d_1, 1)] - \beta(y')}{\beta(y) - \beta(y')} \tag{6}$$

$$= \frac{\frac{2}{5}[U(1, 1) - U(0, 1)] + 1}{2} = \frac{7}{10} \tag{7}$$

Which corresponds to the value in the table of Figure 2

Step 5: We can repeat the same procedure for $\sigma := do(X = 1)$ and find that $P(Y = 0|do(X = 1)) = P(Y = 0|X = 1) = 0.1$. The equivalence between intervention and conditioning follows the specific family of interventions we are using (i.e., hard interventions on all nodes that are not on the directed path to the utility node).

Step 6: Since $P(Y = 0|do(X = 0)) \neq P(Y = 0|do(X = 1))$, we can conclude that X is a parent of Y. In the general case, this approach ensures that X is not just an ancestor but indeed a parent of Y, because the intervention blocks all other paths from X to Y. Thus, $P(Y = 0|(X = 0), pa_Y)$ would equal $P(Y = 0|pa_Y)$ if X were not a parent.

As expected, this process allows us to learn both the correct graph structure and the CPT for Y (and, more generally, for all chance nodes that are not children of D).

Example 2 - Cooperative multi-agent environment

Examine the multi-decision CID in Figure 3. It represents a cooperative game between two agents, each controlling a distinct decision variable. Both agents aim to maximize a shared utility function, U, and operate in different contexts defined by the parent sets of their decision nodes ($Pa_{D_1} = \emptyset \neq \{Z\} = Pa_{D_2}$). Similarly to before, we assume knowledge of the Markov blanket for the decision node D_1, the set of parents of the decision node D_2, and the utility node U.

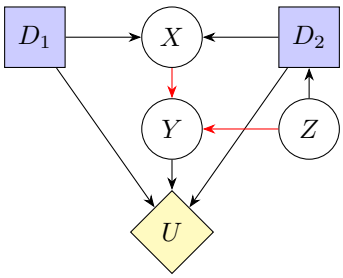


Figure 3: A multi-decision CID that represents an environment where two agents cooperate to maximize the utility U. The edges marked in red are unknown. Example 2 demonstrates how to adapt this CID to apply Algorithm 1 and recover the missing edges and CPTs for chance nodes.

Overview of Methodology In the multi-decision case, having access to an optimal policy oracle for at least one decision node allows us to learn the entire CID. Moreover, for the remaining decision nodes, we can define policies that preserve the faithfulness of the causal model i.e., a policy that effectively depends on the decision node's parents. This approach is feasible because we know the parent set for each decision node. We proceed by treating these decision nodes as chance nodes by assigning their respective policies as CPTs. Algorithm 1 is then applied to learn the CID structure and CPTs for all the chance nodes except those that are children of the decision node for which the optimal policy oracle was used. Once the CPTs for all chance nodes are learned, it becomes possible to determine the optimal policy for every decision node in the original graph under any distribution shift (Shachter 1986; Bareinboim et al. 2022).

Example Analysis Let Π^*_Σ be the optimal policy oracle for D_1 and $\pi(D_2 \mid Z)$ be any given policy that governs D_2. The nodes for which we need to learn the parents are Y and Z. Node Y's potential parents are X and Z, whereas node Z's only potential parent is Y. Using Algorithm 1, we determine parental relationships as follows: consider the instantiation $Y = 0$. With Algorithm 1 we can compute $P(Y = 0|pa_Y; \sigma_0)$ and $P(Y = 0|pa_Y; \sigma'_0)$ using $\sigma_0 := do(X = 0, Z = 0)$, and $\sigma'_0 := do(X = 0, Z = 1)$ respectively. We observe that these two probabilities are equal. We repeat this process with $\sigma_1 := do(X = 1, Z = 0)$ and $\sigma'_1 := do(X = 1, Z = 1)$, and again, $P(Y = 0|pa_Y; \sigma_1) = P(Y = 0|pa_Y; \sigma'_1)$. Performing the same procedure for $Y = 1$, we find that all pairs of interventions yield the same probabilities. Therefore, Z is not a parent of Y. Next, we check whether X is a parent of Y. Comparing $P(Y = y|pa_Y; \sigma_0)$ with $P(Y = y|pa_Y; \sigma_1)$, and $P(Y = y|pa_Y; \sigma'_0)$ with $P(Y = y|pa_Y; \sigma'_1)$ for all $y \in \{0, 1\}$, we find that at least one of these pairs of probabilities differs. Given the faithfulness of the CID. This confirms that X is a parent of Y. Finally, we consider Z. Since the only potential parent of Z was Y, and Y was found to be a child of X, Y can not be a child of Z because this would introduce a cycle in the graph. Z has no other potential parents and therefore we have learned the full CID. Therefore, a distribution shift-robust agent knows that Z is not relevant for this task. This insight can then be used to determine the optimal policy for D_2.

Conclusions

In this work, we addressed the challenge of understanding the relationship between robustness to distribution shifts and an agent's causal understanding of the environment in which it operates. While previous work established that robust agents encode the causal model in single-agent, unmediated tasks, we demonstrated that this connection also holds in mediated tasks and multi-agent settings. We presented an algorithm for learning CIDs using optimal policy oracles, which allows the integration of prior causal knowledge. Our results show that even for mediated tasks, where agents' actions affect the environment, it is possible to recover the underlying causal structure. Moreover, in multi-agent systems, we showed how a single robust agent enables the discovery of the complete causal model, making it possible to learn optimal policies for all the other agents under any distribution shift. These findings provide a solid foundation for the development of AI systems that can adapt and learn in complex settings. To bridge the gap between theory and practical applications, we are actively extending our research to approximate settings, where regret-bounded policies are employed instead of optimal ones.

References

Bareinboim, E.; Correa, J. D.; Ibeling, D.; and Icard, T. 2022. *On Pearl's Hierarchy and the Foundations of Causal Inference*, 507–556. New York, NY, USA: Association for Computing Machinery, 1 edition. ISBN 9781450395861.

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2006. Analysis of representations for domain adaptation. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, 137–144. Cambridge, MA, USA: MIT Press.

Everitt, T.; Carey, R.; Langlois, E. D.; Ortega, P. A.; and Legg, S. 2021. Agent Incentives: A Causal Perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11487–11495.

Heckerman, D. 1995. A Bayesian approach to learning causal networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, 285–295. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Howard, R. A.; and Matheson, J. E. 1984. Influence Diagrams. *Readings on the Principles and Applications of Decision Analysis, Vol. II*.

Konečný, J.; McMahan, H. B.; Yu, F. X.; Richtarik, P.; Suresh, A. T.; and Bacon, D. 2016. Federated Learning: Strategies for Improving Communication Efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*.

Pan, S. J.; and Yang, Q. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10): 1345–1359.

Pearl, J. 2009. *Causality: Models, Reasoning and Inference*. USA: Cambridge University Press, 2nd edition. ISBN 052189560X.

Pearl, J.; and Bareinboim, E. 2011. Transportability of Causal and Statistical Relations: A Formal Approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, 540–547.

Piaget, J. 1936. *La naissance de l'intelligence chez l'enfant*. Neuchâtel, Switzerland: Delachaux et Niestlé.

Richens, J.; and Everitt, T. 2024. Robust agents learn causal world models. In *The Twelfth International Conference on Learning Representations*.

Shachter, R. D. 1986. Evaluating Influence Diagrams. *Oper. Res.*, 34(6): 871–882.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2021. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1): 43–76.