
UNIVERSAL DECISION MODELS

A PREPRINT

Sridhar Mahadevan *

Adobe Research and University of Massachusetts, Amherst
 smahadev@adobe.com, mahadeva@umass.edu

October 28, 2021

ABSTRACT

Humans are universal decision makers: we reason causally to understand the world; we act competitively to gain advantage in commerce, games, and war; and we are able to learn to make better decisions through trial and error. Whilst these individual modalities of decision making have been studied for decades in various subfields of AI and ML, there has been commensurately less effort in developing formalisms that unify these various modalities into common framework. In this paper, we propose Universal Decision Model (UDM), a mathematical formalism based on category theory, to address this challenge. Decision objects in a UDM correspond to instances of decision tasks, ranging from causal models and dynamical systems such as Markov decision processes and predictive state representations, to network multiplayer games and Witsenhausen’s intrinsic models, which generalizes all these previous formalisms. A UDM is a category of objects, which include decision objects, observation objects, and solution objects. Bisimulation morphisms map between decision objects that capture structure-preserving abstractions. We formulate universal properties of UDMs, including information integration, decision solvability, and hierarchical abstraction. Information integration consolidates data from heterogeneous sources by forming products or limits in the UDM category. Abstraction simulates complex decision processes by simpler processes through bisimulation morphisms by forming quotients, co-products and co-limits in the UDM category. Finally, solvability of a UDM decision object is defined by a fixed point equation, and it corresponds to an isotonic order-preserving morphism across the topology induced by UDM objects. We describe universal functorial representations of UDMs, and propose an algorithm for computing the minimal object in a UDM using algebraic topology. We sketch out an application of UDMs to causal inference in network economics, using a complex multiplayer producer-consumer two-sided marketplace.

Keywords Causal inference · Reinforcement Learning · Game Theory · Category Theory · Decision Making

1 Introduction

One of the singular aspects of human cognition is our universal decision capacity: we reason causally to interact with and understand the world from a young age (Sobel et al., 2004), and continue to do so into adulthood (Pearl, 2009; Imbens and Rubin, 2015). We act competitively when it benefits us in arms control negotiations, commerce, and games (Maschler et al., 2013; Shoham and Leyton-Brown, 2008). Since we almost always make sub-optimal decisions, due to incomplete information and computational limitations (Russell and Subramanian, 1995), we learn to make better decisions through trial and error (Sutton and Barto, 1998). Whilst these individual decision making modalities have been studied for decades in AI (Russell and Norvig, 2020), we nonetheless possess an inadequate understanding of how to integrate these disparate abilities. It appears we understand the parts of universal decision making far better than the whole! The main contribution of this paper is a novel theory of universal decision making, codified in a mathematical framework we call Universal Decision Model (UDM). The bulk of the paper is focused on understanding the *information structures* that guide decision making. In particular, our paper does not specifically address the algorithmic aspects of universal decision making, although we touch upon this topic towards the end. Furthermore, optimization plays a

*Draft under revision. Comments welcome

central role in much of the literature in *sequential* decision making. As the scope of UDM is far broader than sequential decision making, which is a very specialized information structure, optimization as traditionally conceived plays only a minor role in the UDM framework.

Our paper is also motivated by the growing need to understand how to scale existing formalisms to extremely large and complex decision systems, both to understand complex behavior in biology, and to control large decentralized computing systems. Our work is related to category-theoretic models of complex interconnected systems (Fong, 2016). Consider a group of computing elements that form a cloud AI implementation, which are tasked to make decisions on gathering and processing data from a heterogeneous set of sources (Lin et al., 2020). Similarly, consider the challenge faced by a group of honeybees that are scouting their environment for a new location for their hive (Seeley, 2011). Our framework provides a fresh perspective on these challenges, bringing a powerful formalism of categorial thinking to shed light on universal properties underlying decision making in these different realms. Our approach does not assume any a priori ordering on the agent structure, which must be discovered or designed to make the problem feasible. Each agent may be unaware if it should act first or last, or indeed, if it should act at all. The organization of agents into a linearly or partially ordered structure may vary, depending on the state of nature, randomness in observations, or the task at hand. To ensure unique solvability of complex decision making tasks by such large organizations of agents, the fundamental *information structures* that underlie decision making must be carefully designed.

2 Universal Decision Model: Informal Overview

Figure 1 illustrates the broad Universal Decision Model (UDM) framework studied in this paper, which seeks to elucidate the common information structures that underly a variety of decision making modalities that have been extensively studied in a broad swath of literature, including AI (Russell and Norvig, 2020), control theory (Witsenhausen, 1975), game theory (Maschler et al., 2013; Nisan et al., 2007) statistics (Imbens and Rubin, 2015), psychology Sobel et al. (2004), and network economics Nagurney (1999). Broadly, a UDM involves a collection of *elements* A (representing agents, causal variables, points in time, economic entities etc.), a decision space $U_\alpha \in A$ for each “actor” $\alpha \in A$ that has an associated measurable space $(U_\alpha, \mathcal{F}_\alpha)$ (over which a suitable probability space can be defined), and most importantly, an *information field* (Witsenhausen, 1973) that represents each agent’s state of knowledge regarding its decision. Each agent α makes a decision using a policy $\pi_\alpha : \prod_\beta U_\beta \rightarrow U_\alpha$ that defines a measurable function over $(\mathcal{I}_\alpha, \mathcal{F}_\alpha)$. The measurability condition imposes an abstract constraint on what an agent knows in making a decision. At the one extreme, if the agent’s information field $\mathcal{I}_\alpha \subset \mathcal{F}(\emptyset)$, that implies it can act without depending on any of the other decision makers. More generally, the information field $\mathcal{I}_\alpha \subset \mathcal{F}(B)$ for some subset $B \subset A$ of decision makers, which imposes a (pre or partial) ordering on the decision makers.

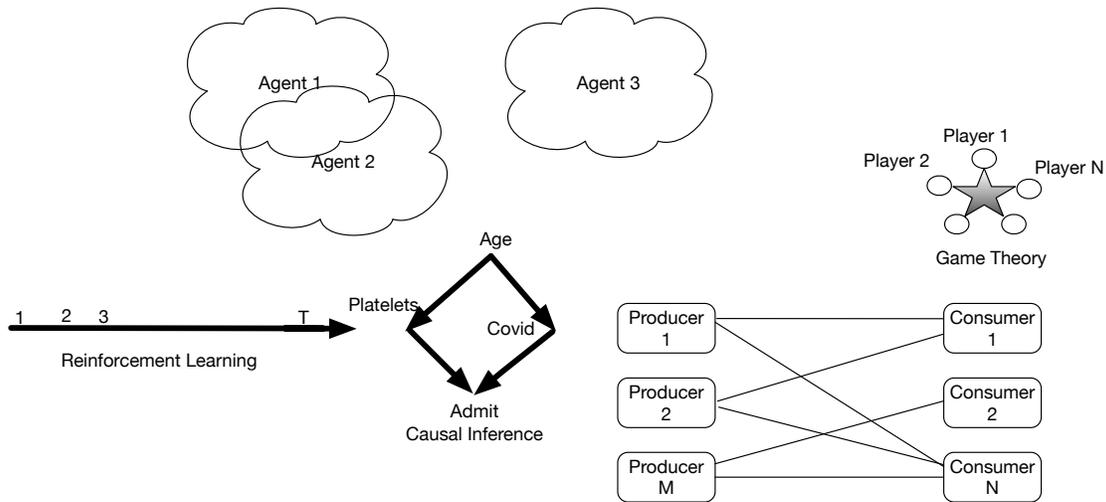


Figure 1: Universal decision model (UDM) is a unifying framework that integrates decision making across multiple modalities, from reinforcement learning (left) and causal inference (middle) to complex network games (right).



Figure 2: Left: a swarm of honeybees scouting for a new location are required to make a complex life-altering decision based on collecting information from surveillance flights (Seeley, 2011). Right: A generic cloud computing network can be decomposed into subsystems, comprised of computing nodes with varying degrees of information access.

2.1 Two Real-World Applications

To motivate the following theoretical development, we turn to two practical applications, one involving the design of cloud computing systems, and the second involving the computation of equilibria in network economics. Together, these real-world problems will illustrate the need to develop more sophisticated notions of agency that the usual formalisms in causal inference, game theory and RL currently enable.

Figure 2 illustrates challenges of decision making by complex groups of agents in biology and in technology. A swarm of thousands of honeybees are required to make a life-altering decision on where to relocate their hive based on reconnaissance flights, and lack of any a priori fixed coordination mechanism among the bees (Seeley, 2011). A generic cloud computing network, which is organized into subsystems with varying levels of information storage, compute power, and responsiveness. In our paper, we abstract from the specifics of such systems, and in fact, even potential applications of such networks. Our focus is primarily on understanding how to theoretically characterize the *information structures* underlying such systems. For example, computing elements at the lowest level of the cloud network may be individual IOT devices or smartphones. These have visibility into data collected at an individual level. In contrast, the subsystems at the fog or edge layer have greater visibility at the aggregate population level, but due to privacy concerns, may have access to only aggregate statistics of individual data.

2.2 Information Fields

Our work draws extensively on the idea of *information fields*, a key component of Witsenhausen’s intrinsic model (Witsenhausen, 1971a,b, 1973, 1988). Information fields provide a foundation to analyze decision making in a wide range of settings, from game playing, to decentralized decision making and multiagent stochastic control. A book length treatment of the intrinsic model is given in (Carpentier et al., 2015). The intrinsic model continues to be studied (Grover, 2015; Nayyar et al., 2011, 2013; Nayyar and Teneketzis, 2019; Nayyar and Basar, 2012), and was recently shown to generalize Pearl’s causal do-calculus (Heymann et al., 2021).

At its core, the *intrinsic model* is based on a measure-theoretic approach for representing *information fields* – the data available to make a decision at some point. Witsenhausen introduced the notion of a *subsystem* that defines a topology on the finite space of entities in the model, based on a (reflexive, transitive) *pre-ordering* relationships based on each element’s information fields. A key insight of his is the discovery that the subsystem relationship is intimately related to the nature of the overall decision-making problem. For example, a *team* decision making problem involves entities that can act without knowledge of each other’s information fields, which defines a topology where the subsystems correspond to singletons. In contrast, a T_0 (Kolmogorov) topology is defined by a *sequential* intrinsic model where there exists a fixed ordering $(\alpha_1, \dots, \alpha_n)$ of the (decision makers, variables) entities such that the information field for entity α_k is purely a function of the fields defined by entities that preceded it in this fixed ordering.

Our main contribution in this paper is to study the categorial foundations of the intrinsic model, namely elucidate the universal properties of information fields that underlie complex decision making. We focus on three universal properties: information integration, decision solvability and hierarchical abstraction. An agent is constantly required to make

decisions given partial information about its environment. The ability to act thus requires integration of information from multiple sources into a sufficient statistic for action. A decision problem must be solvable in a well-defined manner, which almost always can be shown to reduce to solving a fixed point equation. Finally, complex decision problems must be decomposable for decision making to scale: hierarchical abstraction that ignores details is an essential component for scalability.

Our paper uses two fundamental guiding principles. The first principle regards the definition of what is considered a *universal property*, which is based on category theory (Riehl, 2016). In category theory, objects are characterized not in terms of their internal structure, but rather the interactions they make with other objects. A universal property of an object, consequently, is a functorial representation of its *interaction* with other objects, which serves to define the object up to isomorphism. We are thus interested in answering the fundamental question: given a decision-making object, whether a structural causal model or a game or an MDP, how can we functorially characterize its universal properties up to isomorphism?

Rather than describe an object by enumerating its elements, such as commonly done in set theory, category theory builds on the principle of describing objects by their interaction with other objects in the category. This principle is embodied in the *Yoneda lemma*, one of the deepest and most influential results in category theory. This lemma formalizes precisely the notion that an object c in a category \mathcal{C} can be completely described (to within an isomorphism) by the set of all morphisms from the other objects in \mathcal{C} from c , or from other objects to c . Morphisms in a category are closed under composition, and satisfy an associative property. *Functors* are structure-preserving mappings from one category \mathcal{C} to another category \mathcal{D} , which map objects in \mathcal{C} to corresponding objects in \mathcal{D} , but also map morphisms $f : c \rightarrow d$ in \mathcal{C} to corresponding morphisms in \mathcal{D} .

We characterize the universal properties of information fields that play a foundational role in decision making using the tools of category theory. Information integration corresponds to the ability to form *products* of elements. The product object in a category is universal with respect to the property of having unique morphisms to its factors, such that every mapping to one of the factors must be uniquely decomposable through the product. This universal property is shown to underlie a range of decision making formalisms, from decision making to causal inference. Another key notion is that of a *bisimulation* between objects representing decision making processes (Arbib and Manes, 1974; Joyal et al., 1993), which underpins the widespread use of homomorphisms in MDPs and predictive state representations (PSRs) (Ravindran and Barto, 2003; Dean and Givan, 1997; Soni and Singh, 2007). We characterize bisimulation as a universal property through quotient spaces, defined by an equivalence relation \sim , where the quotient space X/\sim is uniquely characterized by the ability to map objects such that isomorphic objects have the same image. We will see that the notion of subsystems in intrinsic models is based on the universal property of quotient spaces, wherein agents form equivalence classes of subsystems based on shared information fields.

Decision solvability in causal inference, games and reinforcement learning all involve finding fixed points of a system of equations. For example, in recursive structural causal models $M = (U, V, F, P)$, where U is a set of exogenous variables, V is a set of endogenous variables where each $x_i \in V$ is a function $f_i \in F$ of some subset of variables $U \cup V - \{x_i\}$, and P is a probability over the exogenous variables U , recursive solvability implies there is a fixed ordering of the variables $U \cup V$ such that each exogenous variable takes on a unique value $X_i = f(\text{Pa}_i)$ (where Pa_i refers to the “parents” of x_i) defined for some particular probability $P(u)$ of the exogenous variables. This constraint imposes a fixed point requirement on solvability. Similarly, in game theory, each agent must be able to compute a best response behavior based on knowledge of the other agents’ actions. Finally, in reinforcement learning, the Bellman optimality condition imposes a fixed point solvability constraint. All of these constraints can be shown to follow the general causality principle elucidated by Witsenhausen (Witsenhausen, 1971a). We characterize the universal properties of information fields that induce solvable decision problems in all these specialized settings.

2.3 Information Fields

The concept of information fields had its origins in work on game theory (Aumann, 1976, 1961; Maschler et al., 2013), and we introduce it first in that setting where it can be described in a simpler way. In general, a group of decision makers only have partial knowledge of the true state of nature, referred to below by a (continuous or discrete) set Ω . At any point where an agent α is contemplating a decision, the true state of the world may be indicated by $\omega_0 \in \Omega$, but the agent may only be able to glimpse the true state of ω_0 with some uncertainty, e.g. knowing it belongs to some partition field of Ω .

Definition 1. A *partial information game* $\mathcal{G} = \langle A, (\Omega, \mathcal{B}, P), (U_\alpha, \mathcal{F}_\alpha)_{\alpha \in A} \rangle$, where A is a finite group of players, Ω defines the states of nature, \mathcal{B} is the usual Borel topology of sets closed under complementation and countable unions on Ω , so that (Ω, \mathcal{B}, P) forms a probability space. Each player can make decisions from a continuous or discrete set U_α , and their knowledge of the true state of the world is defined by \mathcal{F}_α , a partition of Ω .

For example, consider the ensemble of computing elements in a cloud computing network, or in an network economics problem, such as those shown in Figure 2, by the set A . Each computing element $a \in A$ can be thought of as an “actor” that makes decisions over the measurable space $(U_\alpha, \mathcal{F}_\alpha)$. Intuitively, this means that U_α could be a discrete or continuous set of choices, and \mathcal{F}_α is a partition of Ω . Consider a two unit network with units α and β , where the parameters of the game are defined as follows:

- $\Omega = \{1, 2, \dots, 9\}$, $\mathcal{B} = 2^\Omega$, $P\{i : i \in \Omega\} = \frac{1}{9}$.
- $\mathcal{F}_\alpha = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$.
- $\mathcal{F}_\beta = \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \{9\}\}$.

We would like to be able to update the partitions based on events. For example, if the computing elements observed the event $\{3, 4\}$, what would their posteriors look like?

2.3.1 Operations on Information Fields: Join and Meet

To update the priors based on evidence, we introduce the join and meet operations on partition fields, and more generally, on σ -algebras. Carpentier et al. (2015) contains an extensive discussion of partition fields, and its relation to σ -algebras. We now give a simple example of working with partition fields, which will later be generalized to σ -algebras. For agent α , its partition field \mathcal{F}_α contains a partition of the states of nature Ω . We want to introduce two operations on states of knowledge that will be useful in the remainder of the paper, namely join and meet. The set of partition fields, or their generalization, σ -algebras, form a partially ordered set, or even a lattice. This structure naturally allows computing the least upper bound and the greatest lower bound of a set of elements. We will use the join and meet operations to indicate these as follows.

Definition 2. The **meet** of two partition fields $\mathcal{F}_\alpha \wedge \mathcal{F}_\beta$ is defined as finest partition refined by both \mathcal{F}_α and \mathcal{F}_β . In contrast, the **join** of two partitions $\mathcal{F}_\alpha \vee \mathcal{F}_\beta$ is the coarsest common refinement of \mathcal{F}_α and \mathcal{F}_β . More formally, we say a partition \mathcal{F}_α is **finer** than another partition \mathcal{F}_γ , denoted as $\mathcal{F}_\gamma \leq \mathcal{F}_\alpha$, if every element of \mathcal{F}_γ is included as an element of \mathcal{F}_α .

In other words, to compute the join of two partition fields, we compute the intersection of every component of \mathcal{F}_α with that of \mathcal{F}_β . To compute the meet of two partition fields, we find the smallest set of subsets that can be composed as the union of partition elements from \mathcal{F}_α and \mathcal{F}_β . Using the above simple example of a game, we get:

- $\mathcal{F}_\alpha \vee \mathcal{F}_\beta = \{\{1, 2, 3\}, \{4\}, \{5, 6\}, \{7, 8\}, \{9\}\}$.
- $\mathcal{F}_\alpha \wedge \mathcal{F}_\beta = \{\{1, 2, 3, 4, 5, 6, 7, 8, 9\}\}$. Note here that there does not exist a smaller subset that can be constructed out of the subsets in both partition fields.

Consider α and β observing the event $A = \{3, 4\}$. In this case, the posteriors for each of them is given as:

- $\mathcal{F}_\alpha^A = \{\{1, 2, 3\}, \{4, 5, 6\}\}$, leading to its assessment of the probability of the state of nature being $F_\alpha(\omega) = \frac{P(A \cap P_\alpha(\omega))}{P_\alpha(\omega)} = \frac{1}{3}$.
- $\mathcal{F}_\alpha^B = \{\{1, 2, 3, 4\}\}$, leading to its assessment of the probability of the state of nature being $P_\beta(\omega) = \frac{P(A \cap P_\alpha(\omega))}{P_\alpha(\omega)} = \frac{1}{2}$.

Aumann (1976) studied an interesting class of problems where the players cannot agree to disagree if they share common knowledge about an event. A classic example of this problem is the “muddy children problem”, where a group of children are told they can go home if their foreheads are muddy. Each child cannot see his or her own forehead, but they can see the other foreheads, and no communication is allowed otherwise. For a group of N children, it turns out the teacher has to repeat the statement “At least one child has a muddy forehead” before all the children get up to leave the class. This is a simple but insightful example of the problem of reasoning with common knowledge. We return to this topic later in the paper, and pose it again in the context of information fields.

2.3.2 Sigma algebras and Cylindrical Extensions

We will work more generally with σ -algebras, but the underlying concepts are similar to partition fields (a detailed comparison of their properties is given in (Carpentier et al., 2015)). σ -algebras are defined on the states of nature Ω as a collection of subsets \mathcal{F} that are closed under complementation and countable union, which implies closure under intersection as well, and with the restriction that $\Omega \in \mathcal{F}$.

Definition 3. A measurable space (U, \mathcal{F}) is defined as a set U along with a σ -algebra \mathcal{F} of subsets of U , closed under complementation and (countable) union, along with the constraint that the complete set $U \in \mathcal{F}$.

Much of our discussion in this paper will be in the context of measurable functions on measurable spaces.

Definition 4. A measurable function $f : U \rightarrow V$ is defined to be any function defined over measurable spaces in its domain and range, namely if (U, \mathcal{F}_U) is the measurable space over its domain, and (V, \mathcal{F}_V) is the measurable space over the range, then every pre-image of a measurable set in the range is measurable in the domain, that is $f^{-1}(Y) \in \mathcal{F}_U, Y \in \mathcal{F}_V$.

An important special case is when the σ -algebras are finite, in which we can use the following theorem.

Theorem 1. For any finite measurable space (U, \mathcal{F}) , its σ -algebra \mathcal{F} can be generated purely from a partition of $U = \{P_1, \dots, P_k\}$, by forming the union of all possible subsets in the partition. That is, for any $X \in \mathcal{F}$, it follows that $X = \cup_{i \in I} P_i$, where $I \subset \{1, \dots, k\}$.

An important application of this theory is defining observations over information fields. We can state the general definition as follows:

Definition 5. The smallest σ -algebra $\sigma(C)$ generated by a family of sets $C = \{C_1, \dots, C_k\}$ is defined as the intersection of all σ -algebras containing C .

In particular, given a topological space the smallest σ -algebra generated by the topology is called the Borel σ -algebra.

Definition 6. The Borel σ -algebra is defined as the smallest σ -algebra defined by the topological space (X, \mathcal{O}) .

This leads naturally to the definition of a probability space (Ω, \mathcal{B}, P) . A detailed definition of probability measures is given in any textbook on measure theory (Halmos, 1974).

Definition 7. The probability space (Ω, \mathcal{B}, P) is defined as a measurable space (Ω, \mathcal{B}) with a measurable function $P : \Omega \rightarrow (0, 1)$, defined over it, such that $P(\Omega) = 1$, $P(A \cup B) = P(A) + P(B)$ for all disjoint events A, B , where P is a measurable function.

Consider for simplicity the case when a computing element represents an IOT sensor that can only measure two values, so in this case, $U_\alpha = \{0, 1\}$. We can choose \mathcal{F}_α in several ways, ranging from the power set or *discrete topology* $\mathcal{F}_\alpha = P(U_\alpha) = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$ to the *indiscrete topology* $\mathcal{F}_\alpha = \{\emptyset, \{0, 1\}\}$. Each computing unit also has some awareness of the “state of nature”, which could be represented as a set of noisy measurements of local and/or global information. The state of nature is modeled as a probability space (Ω, \mathcal{B}, P) , where Ω is the sample space of events, \mathcal{B} is a measurable space of subsets of the sample space, which is also endowed with a Borel topology, and P is the probability measure such that $P(\Omega) = 1$.

We can also use some simple properties of σ -algebras.

- If $D \subset C \subset B \subset A$, then $\mathcal{F}_B(D) \subset \mathcal{F}_B(C)$, $\mathcal{F}_B(C) \cup \mathcal{F}_B(D) = \mathcal{F}_B(C \cup D)$, $\mathcal{F}_B(C) \vee \mathcal{F}_B(D) = \mathcal{F}_B(C \cup D)$, which will be useful below in defining a topology over computing elements that share a common information field.
- Note $H_\emptyset = \Omega$, and $\mathcal{F}_B(\emptyset)$ is defined as the *cylindrical extension* of the σ -algebra \mathcal{B} over states of nature to H_B . In general, the cylindrical extension of a σ -algebra $\prod_{\alpha \in B} \mathcal{F}_\alpha$ for a subset $B \subset A$ to all of A is defined as $\prod_{\alpha \in B} \mathcal{F}_\alpha \times \prod_{\alpha \notin B} \{\emptyset, \mathcal{F}_\alpha\} \subset \mathcal{F}_A(A) \equiv \mathcal{F}$. In other words, the σ -algebra for elements $\alpha \in B$ remains the same, whereas for elements $\alpha \notin B$, we use the maximally uninformative σ -algebra of the indiscrete topology $\{\emptyset, \mathcal{F}_\alpha\}$.

3 Universal Decision Model

We now proceed to give a more formal introduction to the Universal Decision Model (UDM), which draws extensively on the concepts in category theory (Riehl, 2016), as well as Witsenhausen’s information field representation (Witsenhausen, 1971b), suitably generalized to the setting of category theory. Accordingly, we first give a brief review of category theory, and then proceed to describe UDM. Subsequent chapters will explore particular instantiations of UDM models in more concrete settings, such as causal inference, stochastic control and reinforcement learning, and network economics.

3.1 Category Theory

Over the past 70 odd years, a concerted effort by a large group of mathematicians has resulted in the development of a sweeping unification of large areas of mathematics using category theory (Riehl, 2016). Table 1 compares the basic

Set theory	Category theory
set	object
subset	subobject
truth values $\{0, 1\}$	subobject classifier Ω
power set $P(A) = 2^A$	power object $P(A) = \Omega^A$
bijection	isomorphisms
injection	monic arrow
surjection	epic arrow
singleton set $\{*\}$	terminal object 1
empty set \emptyset	initial object 0
elements of a set X	morphism $f : \mathbf{1} \rightarrow X$
-	non-global element $Y \rightarrow X$
-	functors, natural transformations
-	limits, colimits, adjunctions

Table 1: Comparison of notions from set theory and category theory.

notions in set theory vs. category theory. Briefly, a category is a collection of objects, and a collection of morphisms between pairs of objects, which are closed under composition, satisfy associativity, and include an identity morphism for every object. For example, sets form a category under the standard morphism of functions. Groups, modules, topological spaces and vector spaces all form categories in their own right, with suitable morphisms (e.g. for groups, we use group homomorphisms, and for vector spaces, we use linear transformations). We will illustrate the application of category theory to reinforcement learning by showing that it is relatively straightforward to define categories over MDPs and PSR models, based on the previously defined homomorphisms over these models. We then summarize previous work on open maps over machines, which generalizes these ideas.

A broad class of models used in optimal control, reinforcement learning, operations research and system identification can be characterized in terms of categories and the morphisms between them, including Markov decision processes (MDPs) and semi-MDPs (Puterman, 1994), predictive state representations (PSRs) (Soni and Singh, 2007) and subspace identification models in system identification (Overschee and Moor, 1993), as well as Witsenhausen’s intrinsic model of decentralized stochastic control based on information fields (Witsenhausen, 1973). Our work can also be viewed as a generalization of previous abstraction methods, such as *homomorphisms* used in model minimization in MDPs and SMDPs (Dean and Givan, 1997; Ravindran and Barto, 2003) and PSR’s (Soni and Singh, 2007), as well as related abstraction models used in algebraic automata theory (Hartmanis and Stearns, 1962).

Our presentation will follow the excellent treatments given in (Bradley et al., 2020; Goldblatt, 2006; Riehl, 2016). Intuitively, a category is simply a *collection of objects* X, Y, \dots , and a collection of *morphisms* f, g, \dots , where $f : X \rightarrow Y$ is the morphism whose domain is X and co-domain is Y . A basic principle of category theory is that objects have no discernable internal structure, and their identity up to isomorphism is revealed by their interaction with other objects in the category. To take a simple, but illustrative example, consider a set X with n elements. Rather than list the elements of the set, we define it simply as a collection of mappings from the category **1** to X , where **1** is the category with exactly one object, and one morphism (identity). Each mapping from **1** to X must by definition pick out one of its elements, and consequently the entire ensemble of elements in X is revealed by the ensemble of mappings from **1** to X . The Yoneda lemma described later generalizes this principle to mappings from an arbitrary category to the category of sets. Mappings between categories are known as *functors*, and will be defined below.

For each pair of morphisms f, g , such that the co-domain of f is the same as the domain of g , there is a composite morphism gf , simply defined as the composition of g and f (where f is applied first, followed by g), defined as $gf : X \rightarrow Z$. There are two additional requirements: each object X has associated with it an *identity* morphism $1_X : X \rightarrow X$, whose composition with any other morphism $f : X \rightarrow Y$ is defined as $1_Y f = f = f 1_X = f$. The second requirement is *associativity*, whereby given morphisms $f : X \rightarrow Y, g : Y \rightarrow Z, h : Z \rightarrow W$, the composite morphism $hgf : X \rightarrow W$ is associative.

Some examples of categories are illustrated below, which we will refer to in the remainder of the paper.

- **Set**: The canonical example of a category is **Set**, which has as its objects, sets, and morphisms are functions from one set to another. The **Set** category will play a central role in our framework, as it is fundamental to the universal representation constructed by Yoneda embeddings.

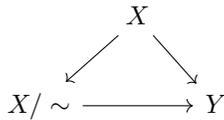
- **Top:** The category **Top** has topological spaces as its objects, and continuous functions as its morphisms. Recall that a topological space (X, Ξ) consists of a set X , and a collection of subsets Ξ of X closed under finite intersection and arbitrary unions.
- **Group:** The category **Group** has groups as its objects, and group homomorphisms as its morphisms.
- **Graph:** The category **Graph** has graphs (undirected) as its objects, and graph morphisms (mapping vertices to vertices, preserving adjacency properties) as its morphisms. The category **DirGraph** has directed graphs as its objects, and the morphisms must now preserve adjacency as defined by a directed edge.
- **Poset:** The category **Poset** has partially ordered sets as its objects and order-preserving functions as its morphisms.
- **Meas:** The category **Meas** has measurable spaces as its objects and measurable functions as its morphisms. Recall that a measurable space (Ω, \mathcal{B}) is defined by a set Ω and an associated σ -field of subsets \mathcal{B} that is closed under complementation, and arbitrary unions and intersections, where the empty set $\emptyset \in \mathcal{B}$.

3.2 Universal Properties

A core goal in category theory is to elucidate the universal properties of objects and morphisms. The motivation is understand the essence of what makes a particular concept unique. For example, in set theory, the cartesian product of two sets $A \times B = \{(a, b) | a \in A, b \in B\}$ is simply defined by listing the elements of the set representing the cartesian product. In category theory, a different approach is taken, one that involves articulating the universal property of objects that represent cartesian products, as we will see below. One of our primary contributions is the categorial formulation of Witsenhausen’s intrinsic model. The key principle underlying category theory is *universality*: this seemingly simple concept is somewhat difficult to grasp at first glance since some of its definitions involve a deeper definition of terms that we provide in later sections. Intuitively, let us for now consider the universal property of an object to be something that characterizes all morphisms into or out of the object. This philosophy of describing objects in terms of the interactions they make with other objects is a key characteristic of category theory.

3.2.1 Quotients:

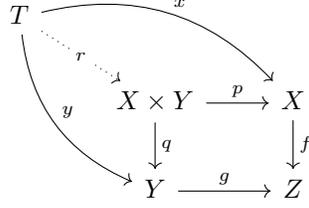
Quotient spaces induced by an equivalence relation \sim play a fundamental role in the UDM framework. Given a category \mathcal{C}_{UDM} of decision objects objects, the quotient $\mathcal{C}_{\text{UDM}} / \sim$ is the set of equivalence classes in \mathcal{C}_{UDM} , whereby an object $c \in \mathcal{C}_{\text{UDM}}$ is mapped to its equivalence class $[c]_f$ under the function f such that $f(c) = f(d)$ implies $[c] = [d]$. Reflexivity, symmetry, and transitivity easily follow from the definition. The canonical projection $\pi : C \rightarrow C / \sim$ is the unique map sending c to its equivalence class $[c]_f$. Quotients will play a key role in the UDM framework as we will see below.



The universal property of quotients is indicated in the above diagram whereby any map from object X to object Y that equates equivalent objects is uniquely factorizable through its quotient map, so that $f = g\pi$, and the diagram commutes. Quotients have played a central role in MDP homomorphisms (Dean and Givan, 1997; Ravindran and Barto, 2003) and PSR’s (Soni and Singh, 2007), as well as related abstraction models used in algebraic automata theory (Hartmanis and Stearns, 1962).

3.2.2 Product:

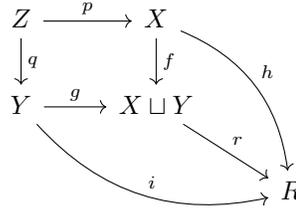
A central motif in much of the literature in decision making is the need to integrate information from multiple sources. In the RL literature, dynamical system models like MDPs, POMDPs and PSRs typically assume the notion of a *state*, which summarizes all the information from the past (or future) that is important for making optimal decisions. In structural causal models, an endogenous variable in the model is a function of exogenous and other endogenous variables, which requires integrating information from all these “parent” variables. In games, an agent needs to consider the potential responses of all other other actions. All of these involve the fundamental operation of a product. In category theory, products are defined as the following universal property:



The above figure shows a diagram, a standard construct in category theory, where objects are depicted by vertices with labels, and morphisms are indicated by labeled edges. This diagram asserts that there is an object labeled $X \times Y$ with morphisms $p : X \times Y \rightarrow X$ and $q : X \times Y \rightarrow Y$, which we recognize immediately as the canonical projection from a cartesian product to its components. Furthermore, the diagram asserts that given any morphism from an object T to X , there is in fact a unique way to factor that morphism through the product object, so that the diagram “commutes”, meaning the morphism $x = p r$. Similarly, any morphism from T to Y is also uniquely factored through r , so that $y = q r$. We have thus characterized the product object purely in terms of the morphisms into and out of the object. Our first claim is that for decision making, the ability to form products is a universal property, which is an essential ingredient in any framework. In the UDM framework, products play a key role in defining information fields, which are a subfield of the product space $(\prod_{\alpha} U_{\alpha}, \prod_{\alpha} \mathcal{F}_{\alpha})$. As we will see later, the ability to form products is essential in using information fields to specify structural causal models, as well as define states in sequential models.

3.2.3 Co-Product:

A related universal property to product is the *coproduct* property, which loosely translates to forming “disjoint” unions of sets. Coproducts refer to the universal property of abstracting a group of elements into a larger one. For example, information fields of multiple decision objects can be combined into one larger information field through co-products.



In the commutative diagram above, the coproduct object $X \sqcup Y$ uniquely factorizes any mapping $h : X \rightarrow R$ and any mapping $i : Y \rightarrow R$, so that $h = r f$, and furthermore $i = r g$.

3.2.4 Pullback and Pushforward Mappings

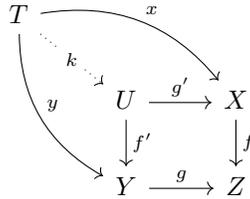


Figure 3: Universal Property of pullbacks and pushforward mappings.

Figure 3 illustrates the fundamental property of a *pullback*, which along with *pushforward*, is one of the core ideas in category theory. The pullback square with the objects U, X, Y and Z implies that the composite mappings $g f'$ must equal $g' f$. In this example, the morphisms f and g represent a *pullback* pair, as they share a common co-domain Z . The pair of morphisms f', g' emanating from U define a *cone*, because the pullback square “commutes” appropriately. Thus, the pullback of the pair of morphisms f, g with the common co-domain Z is the pair of morphisms f', g' with common domain U . Furthermore, to satisfy the universal property, given another pair of morphisms x, y with common domain T , there must exist another morphism $k : T \rightarrow U$ that “factorizes” x, y appropriately, so that the composite morphisms $f' k = y$ and $g' k = x$. Here, T and U are referred to as *cones*, where U is the limit of the set of all cones “above” Z . If we reverse arrow directions appropriately, we get the corresponding notion of pushforward. So, in this example, the pair of morphisms f', g' that share a common domain represent a pushforward pair.

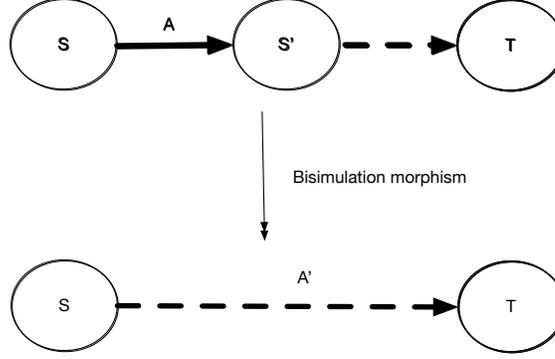


Figure 4: A simple example of a bisimulation morphism between labeled transition systems (Joyal et al., 1993). Our work generalizes this construction to UDM models, replacing states by information fields.

3.3 Universal Decision Model

Now, we introduce the Universal Decision Model (UDM) more formally. In the UDM category \mathcal{C}_{UDM} , as in any category, we are given a collection of *decision objects* \mathcal{D} , and a set of morphisms \mathcal{M}_{UDM} between UDM objects, where $f : c \rightarrow d$ is a morphism that maps from UDM object c to d . A morphism need not exist between every pair of UDM objects. In this paper, we restrict ourselves to *locally small* UDM categories, meaning that exists only a set's worth of morphisms between any pair of UDM objects. More general categories of UDMs are beyond the scope of this introductory paper.

Definition 8. A *Universal Decision Model (UDM)* is defined as a category \mathcal{C}_{UDM} , where each decision object is represented as a tuple $\langle (A, (\Omega, \mathcal{B}, P), U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)_{\alpha \in A} \rangle$, where A describes a finite universe of elements (e.g., random variable in a structural causal model, dynamical systems, such as linear dynamical systems, MDPs, PSRs etc., intrinsic models, or multiplayer network games), (Ω, \mathcal{B}, P) is a probability space representing the inherent stochastic state of nature due to randomness, U_α is a measurable space from which a decision $u \in U_\alpha$ is chosen by decision object α . Each element's policy in a decision object is any function $\pi_\alpha : \prod_{\beta} U_\beta \rightarrow U_\alpha$ that is measurable from its information field \mathcal{I}_α , a subfield of the overall product space $(\prod_{\alpha} U_\alpha, \prod_{\alpha} \mathcal{F}_\alpha)$, to the σ -algebra \mathcal{F}_α . The policy of decision object α can be any function $\pi_\alpha : \prod_{\beta} U_\beta \rightarrow U_\alpha$.

A UDM may also contain *observation objects* and *solution objects*, which we discuss later in the paper. Briefly, observation objects correspond to a "run-time" trace behavior of a decision object, whereas a solution object represents a "solution" of the decision problem. As mentioned at the outset, the traditional role of optimization in much of (sequential) decision making plays only a minor role in the UDM framework, as it is tailored to a particular information structure. We will discuss solution methodologies for particular information structures later in the paper.

Definition 9. The **information field** of an element $\alpha \in A$ in a decision object c in UDM category \mathcal{C}_{UDM} is denoted as $\mathcal{I}_\alpha \subset \mathcal{F}_A(A)$ characterizes the information available to decision object α for choosing a decision $u \in U_\alpha$.

As we will see below, the information field structure yields a surprisingly rich topological space that has many important consequences for how to organize the decision makers in a complex organization into subsystems. An element α in a decision object requires information from other elements or subsystems in the network. To formalize this notion, we use product decision fields and product σ -algebras, with their canonical projections.

Definition 10. Given a subset of nodes $B \subset A$, let $H_B = \Omega \times \prod_{\alpha \in B} U_\alpha$ be the **product space of decisions** of nodes in the subset B , where the **product σ -algebra** is $\mathcal{B} \times \prod_{\alpha \in B} \mathcal{F}_\alpha = \mathcal{F}_B(B)$. It is common to also denote the product σ -algebra by the notation $\otimes_{\alpha \in A} \mathcal{F}_\alpha$. If $C \subset B$, then the **induced σ -algebra** $\mathcal{F}_B(C)$ is a subfield of $\mathcal{F}_B(B)$, which can also be viewed as the inverse image of $\mathcal{F}_C(C)$ under the canonical projection of H_B onto H_C .²

3.4 Bisimulation Morphisms as Open Maps

In a UDM category, the morphisms between decision objects are represented using the concept of *open maps*, as proposed in (Joyal et al., 1993). This framework is based on defining a model of computation as a category. Figure 4

²Note that for any cartesian product of sets $\prod_i X_i$, we are always able to uniquely define a projection map into any component set X_i , which is a special case of the product universal property in a category.

illustrates a simple example of the concept of bisimulation in the category of labeled transition systems, which can be seen as a deterministic MDP (Joyal et al., 1993). A related notion has been proposed for probabilistic bisimulation (Larsen and Skou, 1991). The use of category theory to provide an algebraic characterization of machine models has a long and distinguished history (Arbib and Manes, 1974), which been studied at length in a number of different subfields of computer science. One fundamental notion is *bisimulation* between machines or processes using *open maps* in categories (Joyal et al., 1993, 1996). This definition can be seen as a generalization of the simpler bisimulation relationship that exists for the category of labeled transition systems (Joyal et al., 1993), which are specified as a relation of tuples (s, a, s') , which indicates a transition from state s to state s' , where $s, s' \in S$, and $a \in A$. Given a collection of labeled transition systems, each of which is represented as an object, morphisms are defined from one object to another that preserve the dynamics under the labeling function. For example, a surjective function $f : S \rightarrow S'$ maps states in object X to corresponding states in Y , where the labels are mapped as well, with the proviso that some transitions in X may be hidden in Y (i.e., cause no transition). If a morphism exists between objects X and Y , then Y is said to be a bisimulation of X .

Let \mathcal{M} denote a model of computation, where a morphism $m : X \rightarrow Y$ is to intuitively viewed as a simulation of X in Y . Within \mathcal{M} , we choose a subcategory of “observation objects” and “observation extension” morphisms between them. We can denote this category of observations by \mathcal{P} . Given an observation object $P \in \mathcal{P}$, and a model $X \in \mathcal{M}$, P is said to be an *observable behavior* of X if there is a morphism $o : P \rightarrow X$ in \mathcal{M} . We define morphisms $m : X \rightarrow Y$ that have the property that whenever an observable behavior of X can be extended via f in Y , that extension can be matched by an extension of the observable behavior in X .

Definition 11. (Joyal et al., 1993) A morphism $m : X \rightarrow Y$ in a model of computation \mathcal{M} is said to be \mathcal{P} -open if whenever $f : O_1 \rightarrow O_2$ in \mathcal{P} , $p : O_2 \rightarrow X$ in \mathcal{M} , and $q : O_2 \rightarrow Y$ in \mathcal{M} , the below diagram commutes, that is, $m \circ p = q \circ f$.

$$\begin{array}{ccc} O_1 & \xrightarrow{p} & X \\ \downarrow f & & \downarrow m \\ O_2 & \xrightarrow{q} & Y \end{array}$$

This definition means that whenever such a “square” in \mathcal{M} commutes, the path $f \circ p$ in X can be extended via m to a path q in Y , there is a “zig-zag” mediating morphism p' such that the two triangles in the diagram below

$$\begin{array}{ccc} O_1 & \xrightarrow{p} & X \\ \downarrow f & \nearrow p' & \downarrow m \\ O_2 & \xrightarrow{q} & Y \end{array}$$

commute, namely $p = p' \circ f$ and $q = m \circ p'$. We now define the abstract definition of *bisimulation* as follows:

Definition 12. Two models X and Y in \mathcal{M} are said to be \mathcal{P} -bisimilar (in \mathcal{M}) if there exists a span of open maps from a common object Z :

$$\begin{array}{ccc} & Z & \\ & \swarrow m & \searrow m' \\ X & & Y \end{array}$$

Note that if the category \mathcal{M} has pullbacks (see Figure 3 right), then the $\sim_{\mathcal{P}}$ is an equivalence relation, which induces a quotient mapping. Furthermore, pullbacks of open map bisimulation mappings are themselves bisimulation mappings. Many of the bisimulation mappings studied for MDPs and PSRs are special cases of the more general formalism above.

3.5 Bisimulation in UDMs

We introduce the concept of bisimulation morphisms between UDM objects, which builds on a longstanding theme in computer science on using category theory to understand machine behavior (Arbib and Manes, 1974). One fundamental notion is *bisimulation* between machines or processes using *open maps* in categories (Joyal et al., 1993, 1996).

Definition 13. The **bisimulation** relationship between two UDM objects $M = \langle A, (\Omega, \mathcal{B}, (U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)_{\alpha \in A}) \rangle$ and $M' = \langle A', (\Omega', \mathcal{B}', (U'_\alpha, \mathcal{F}'_\alpha, \mathcal{I}'_\alpha)_{\alpha \in A'}) \rangle$, denoted as $M \twoheadrightarrow M'$, is defined as is defined by a tuple of surjections as follows:

- A surjection $f : A \rightarrow A'$ that maps elements in A to corresponding elements in A' . As f is surjective, it induces an equivalence class in A such that $x \sim y, x, y \in A$ if and only if $f(x) = f(y)$.
- A surjection $g : H \rightarrow H'$, where $H = \Omega \times \prod_{\alpha \in A} U_\alpha$, with the product σ -algebra $\mathcal{B} \times \prod_{\alpha \in A} \mathcal{F}_\alpha$, and $H' = \Omega' \times \prod_{\alpha \in A'} U'_\alpha$, with the corresponding σ -algebra $\mathcal{B}' \times \prod_{\alpha \in A'} \mathcal{F}'_\alpha$.

This definition can be seen as a generalization of the simpler bisimulation relationship that exists for the category of labeled transition systems (Joyal et al., 1993), which are specified as a relation of tuples (s, a, s') , which indicates a transition from state s to state s' , where $s, s' \in S$, and $a \in A$. Given a collection of labeled transition systems, each of which is represented as an object, morphisms are defined from one object to another that preserve the dynamics under the labeling function. For example, a surjective function $f : S \rightarrow S'$ maps states in object X to corresponding states in Y , where the labels are mapped as well, with the proviso that some transitions in X may be hidden in Y (i.e., cause no transition). If a morphism exists between objects X and Y , then Y is said to be a bisimulation of X . Definition 4 can be seen as the generalization of the bisimulation relationship in labeled transition systems, MDPs, and related models like PSRs, to intrinsic models. The state-dependent action recoding in the MDP homomorphism definition is captured by the equivalent surjection g that maps the product space H with its associated σ -algebra to H' with its corresponding σ -algebra.

We can specialize the definition in a number of ways, depending on the exact form chosen for the surjection g between product spaces H and H' . Since the surjection f maps decision makers into equivalence classes, each decision maker in model M' corresponds to an equivalence class of decision makers in model M . Thus, we need to collapse their corresponding information fields. We can define the information field of an equivalence class of agents $[\alpha]_f$, meaning all β such that $f(\beta) = f(\alpha)$, by recalling that an information field is a subfield of the product field, and as it is a lattice, we can use the join operation, as defined below:

Definition 14. The **quotient information field** of a collection of agents $[\alpha]_f$ is defined as the join of the information fields of each agent:

$$\mathcal{I}_{[\alpha]} = \bigvee_{\beta \in [\alpha]_f} \mathcal{I}_\alpha \quad (1)$$

3.6 Observation Objects in UDM

We now briefly discuss observation objects in a UDM. Observation objects, as mentioned above, represent observable trace behavior of a decision object. We first define observation functions that underlie information fields.

Definition 15. For a UDM object $M = \langle A, (\Omega, \mathcal{B}, (U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)_{\alpha \in A}) \rangle$ over a finite σ -algebra, the observations Z_1, \dots, Z_k taking values in a measurable space (Z_i, \mathcal{Z}_i) , where $Z_i = \eta_i(\omega, U_1, \dots, U_{|A|})$ is an **observation generation map function** such that $\sigma(Z_{1:k})$ is the smallest σ -algebra contained in $\mathcal{B} \otimes \prod_{\alpha} \mathcal{F}_\alpha$ with respect to which the observation maps η_i are measurable functions. We say the observations Z_1, \dots, Z_k generate the information field \mathcal{I}_α if

$$\sigma(Z_1, \dots, Z_k) = \mathcal{I}_\alpha \quad (2)$$

We can then define an observation object associated with a UDM decision object as one equipped with an observation generation map that can generate the various information fields in the decision object.

Definition 16. A **UDM observation object** $O = \langle A, (\Omega, \mathcal{B}, (U_\alpha, \mathcal{F}_\alpha, \eta_\alpha)_{\alpha}) \rangle$ is such that each information field \mathcal{I}_α can be generated from the associated observation generation map η_α .

We can define an observation morphism between an observation object O and a decision object M to be one such that O represents an observable behavior of M , and extend the notion of \mathcal{P} -open morphisms from Definition 11 above.

3.7 Example: Network Economics

To illustrate the general UDM framework, we now give an example of a multiplayer producer consumer game from network economics. Consider the network economic model in Figure 5. The set of elements in this decision object can be represented as $(A, (\Omega, \mathcal{B}, P), U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)_{\alpha \in A}$, where A is defined by the set of vertices in this graph representing the decision makers. For example, service provider i chooses its actions from the set U_i , which can be defined as $\cup_{j,k} Q_{ijk}$. \mathcal{F}_i is the associated measurable space associated with U_i . \mathcal{I}_i represents the information field of service provider i , namely its visibility into the decisions made by other entities in the network at the current or past time steps.

Network economics (Nagurney, 1999) is the study of a rich class of equilibrium problems that occur in the real world, from traffic management to supply chains and two-sided online marketplaces. Consider a cloud based network economics model comprises of three tiers of agents: producer agents, who want to sell their goods, transport agents who

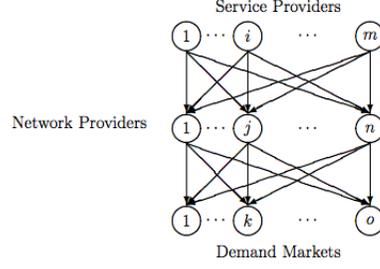


Figure 5: A multiplayer game network economic model (Nagurney and Wolf, 2014) as an example of a UDM. Each decision object in this UDM is a network economics model, where the top tier of producer agents is interested in selling merchandise (digital content, manufactured goods) to a set of demand market agents, but needs the cooperation of transport agents to deliver the merchandise. All the players in this network compete for the best price and quality.

ship merchandise from producers, and demand market agents interested in purchasing the products or services. The model applies both to electronic goods, such as video streaming, as well as physical goods, such as face masks and other PPEs. The model assumes m service providers, n network providers, and o demand markets. Each firm's utility function is defined in terms of the nonnegative service quantity (Q), quality (q), and price (π) delivered from service provider i by network provider j to consumer k . Production costs, demand functions, delivery costs, and delivery opportunity costs are designated by f , ρ , c , and oc respectively. Service provider i attempts to maximize its utility function $U_i^1(Q, q^*, \pi^*)$ by adjusting Q_{ijk} . Likewise, network provider j attempts to maximize its utility function $U_j^2(Q^*, q, \pi)$ by adjusting q_{ijk} and π_{ijk} .

$$U_i^1(Q, q^*, \pi^*) = \sum_{j=1}^n \sum_{k=1}^o \hat{\rho}_{ijk}(Q, q^*) Q_{ijk} - \hat{f}_i(Q) - \sum_{j=1}^n \sum_{k=1}^o \pi_{ijk}^* Q_{ijk}, \quad Q_{ijk} \geq 0$$

$$U_j^2(Q^*, q, \pi) = \sum_{i=1}^m \sum_{k=1}^o \pi_{ijk} Q_{ijk}^* - \sum_{i=1}^m \sum_{k=1}^o (c_{ijk}(Q^*, q) + oc_{ijk}(\pi_{ijk})), \quad q_{ijk}, \pi_{ijk} \geq 0$$

As a second example, consider a toy cloud computing network that is comprised of three elements $A = \{e_1, e_2, h\}$, where e_i is an edge node, and h is a hub node. Let us assume the decision space for the edge nodes $U_1 = U_2 = \{s, r\}$, representing "send" and "receive" modes, and for the hub $U_h = \{c, t\}$ representing "collect" and "transmit" modes. Let us also define the states of nature $\Omega = \{+, -\}$, indicating whether the environment is a "safe" or "unsafe" mode for information transmission or collection. Let us assume that the σ -algebras for both edge and hub devices is defined by the discrete topology given as $\mathcal{F}_{e_i} = \{\emptyset, \{s\}, \{r\}, \{s, r\}\}$, $\mathcal{F}_h = \{\emptyset, \{c\}, \{r\}, \{c, r\}\}$. The σ -algebra for states of nature is given as $\mathcal{B} = \{\emptyset, \{+\}, \{-\}, \{+, -\}\}$. The product decision space is given as $H = \Omega \times \prod_{\alpha \in A} U_{\alpha}$, the product σ -algebra is defined as $\mathcal{F}_A = \mathcal{B} \times \mathcal{F}_{e_1} \times \mathcal{F}_{e_2} \times \mathcal{F}_h$.

3.8 Causality and Solvability of UDM objects

Each decision maker α in a UDM object has associated with it a control law or policy $\pi_{\alpha} : H \rightarrow U_{\alpha}$, which is measurable from its information field \mathcal{I}_{α} to \mathcal{F}_{α} , the measurable space associated with its decision space U_{α} . Essentially, this means that any pre-image of $\pi_{\alpha}^{-1}(E)$, for any measurable subset $E \subset \mathcal{F}_{\alpha}$, is also measurable on its information field, that is $\pi_{\alpha}^{-1}(E) \subset \mathcal{I}_{\alpha}$. For any subsystem in the cloud computing network, the overall policy space $\pi_B = \prod_{\alpha \in B} \pi_{\alpha}$ is given by the product space of all individual control laws.

Definition 17. A UDM object $\langle A, (\Omega, \mathcal{B}, P), (U_{\alpha}, \mathcal{F}_{\alpha}, \mathcal{I}_{\alpha})_{\alpha \in A} \rangle$ is said to be **solvable** if for every state of nature $\omega \in \Omega$, and every control law $\pi \in \Pi_A$, the set of simultaneous equations given below has one and only one solution $u \in U$.

$$u_{\alpha} = \pi_{\alpha}(h) \equiv \pi_{\alpha}(\omega, u) \quad (4)$$

Here, π_{α} can be viewed as a projection from the joint decision h taken by the entire ensemble of decision makers in the intrinsic model. A UDM category \mathcal{C}_{UDM} is solvable if every object in it is solvable.

Intuitively, the solvability criterion states that a UDM object represents a solvable decision problem if each agent in the object can successfully compute its response, given access to its information field, and that its response is uniquely

determined for every state of nature. It is easy to construct unsolvable decision objects. Consider a simple network with two elements α and β , each of whose information fields includes the measurable space of the other. In this case, neither element can compute its function without knowing the other's response, hence both are waiting for the other to compute their response, and a deadlock ensues.

Given the notion of solvability above, we can now define solution objects in a UDM.

Definition 18. A UDM solution object $\langle A, (\Omega, \mathcal{B}, P), (U_\alpha, \pi_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)_{\alpha \in A} \rangle$ is defined as one for which for every state of nature $\omega \in \Omega$, the control law π_α uniquely defines a fixed point solution $u_\alpha = \pi_\alpha(h) \equiv \pi_\alpha(\omega, u)$ to the associated decision object.

We can straightforwardly define morphisms between solution objects and decision objects. To understand the causality condition, it is crucial to organize the decision makers into a partial order, such that for every total ordering that can be constructed from the partial ordering, the agents can successfully compute their functions based on the computations of agents that preceded them in the ordering.

Definition 19. A UDM object $\langle A, (\Omega, \mathcal{B}, P), (U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha) \rangle$ is said to be **causal** if there exists at least one function $\phi : H \rightarrow S$, where S is the set of total orderings of computing elements in A , satisfying the property that for any partial stage of the computation $1 \leq k \leq n$, and any ordered set $(\alpha_1, \dots, \alpha_k)$ of distinct elements from A , the set $E \subset H$ on which $\phi(h)$ begins with the same ordering $(\alpha_1, \dots, \alpha_k)$ satisfies the following causality condition:

$$\forall F \in \mathcal{F}_{\alpha_k}, E \cap F \in \mathcal{F}(\{\alpha_1, \dots, \alpha_{k-1}\}) \quad (5)$$

In other words, if at every step of the process, the k^{th} decision making element α_k can successfully compute its response based on the information fields of the past $k - 1$ elements, the system is then considered causal. Interestingly, it has been shown (see (Heymann et al., 2021)) that the causality condition as stated above generalizes the notion of causality in Pearl's structural causal models (Pearl, 2009).

4 UDMs for Causal Inference and Stochastic Control

We now show the general framework of UDMs transcends multiple decision making regimes, by illustrating how they can form the basis for "universal" decision making in two special cases: linear total ordering, which gives rise to stochastic control, and partial ordering, which gives rise to causal inference.

4.1 UDMs in RL and Stochastic Control

Witsenhausen (1973) himself showed the importance of information fields in stochastic control, in particular developing a canonical model of stochastic control (Witsenhausen, 1973). We summarize a more recent extension from (Nayyar and Teneketzis, 2019) that shows how to define common knowledge using information fields, an interesting contrast to the notion of common knowledge defined above in game theory using Aumann's framework. We define the abbreviation $U_{1:T} = (U_1, \dots, U_T)$ for the product decision space, and similarly $\mathcal{F}_{1:T} = \mathcal{F}_1 \times \dots \times \mathcal{F}_T$ for the product σ -algebra.

Definition 20. (Nayyar and Teneketzis, 2019) Given the probability model (Ω, \mathcal{B}, P) for the random states of nature $\omega \in \Omega$, the measurable decision spaces $(U_t, \mathcal{F}_t), t = 1, \dots, T$, the information field σ -algebras $\mathcal{I}_t \subset \mathcal{B} \times \mathcal{F}_1 \dots \mathcal{F}_T$, and the cost function $c : (\Omega \times U_{1:T} \times \mathcal{B} \times \mathcal{F}_{1:T} \rightarrow (\mathbb{R}, \mathbb{B}))$, find a (generally non-stationary) policy $\pi = (\pi_1, \dots, \pi_T)$, with each policy at time t defined as the mapping $g_t : (\Omega \times U_{1:T}, \mathcal{I}_t \rightarrow (U_t, \mathcal{F}_t)$, that minimizes the cost function $\inf_\pi E[c(\omega, U_1, \dots, U_T)]$ exactly, or to within ϵ .

Note that the above definition of stochastic decision making is just a special case of the UDM model in Definition 8. In particular, the agents in stochastic control are labeled $1, \dots, T$, their temporal ordering is fixed a priori, and each agent's information field is generally defined over the entire horizon $(1, \dots, T)$. To define the special case of finite horizon *sequential* stochastic control, we must impose further conditions on the information fields available at each instant of time $t \in (1, \dots, T)$.

Definition 21. An information structure in the stochastic control model in Definition 20 is **sequential** if there exists a permutation $p : \{1, \dots, T\} \rightarrow \{1, \dots, T\}$ such that for $t = 1, \dots, T$, the information field $\mathcal{I}_t \subset \mathcal{B} \times \mathcal{F}_{p(1)} \times \mathcal{F}_{p(2)} \times \dots \times \mathcal{F}_{p(t-1)} \times \{\emptyset, \mathcal{F}_{p(t)}\} \times \dots \times \{\emptyset, \mathcal{F}_{p(T)}\}$.

In terms of the terminology we have introduced earlier, note that the information field I_t at time t is a cylindrical extension from the field over $1, \dots, t - 1$ to all of $1, \dots, T$. Note that for the sequential case, the permutation ordering p is fixed a priori, and does not vary over the different states of nature $\omega \in \Omega$.

We now define the notion of *common knowledge* in information fields based on the definition in (Nayyar and Teneketzis, 2019). Recall that the information field $\mathcal{I}_t \subset \mathcal{F} \times \mathcal{F}_1 \times \dots \mathcal{F}_{t-1} \times \{\emptyset, U_t\} \times \dots \{\emptyset, U_T\}$.

Definition 22. (Nayyar and Teneketzis, 2019) The **common knowledge** for the t^{th} decision maker in a sequential intrinsic model is defined as

$$\mathcal{C}_t = \bigcap_{s=t}^T \mathcal{I}_s \quad (6)$$

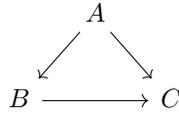
That is, the common knowledge \mathcal{C}_t is defined as the intersection of all information fields from time t till the end of the decision process.

Some simple properties of common knowledge can be readily shown:

- Coarsening property: $\mathcal{C}_t \subset \mathcal{I}_t$: immediate from definition.
- Nestedness property: $\mathcal{C}_t \subset \mathcal{C}_{t+1}$: immediate from definition.
- Common observations: There exist observations Z_1, \dots, Z_T with Z_t taking values in a finite measurable space $(Z_t, 2^{Z_t})$, and $Z_t = \eta_t(\omega, U_1, \dots, U_{t-1})$ such that $\sigma(Z_{1:t}) = \mathcal{C}_t$: for a detailed proof, see (Nayyar and Teneketzis, 2019)). The basic idea exploits the fact that finite σ -algebras can be generated from partitions.

4.2 The Category of Causal UDMs

In the above, we assume that temporal ordering is given a priori as a total ordering $(1, \dots, T)$, and in the sequential case, each decision maker’s information field is a subset of the product decision and information fields of all agents that have acted prior to it. We now generalize from the requirement of imposing a strict linear ordering, and consider more general partially ordered temporal structures. This relaxation from linear to partial ordering allows us to formalize a particular case of the intrinsic model that in fact exactly corresponds to causal inference, as shown recently in Heymann et al. (2021). We briefly review how information fields can formalize causal inference, referring the interested reader to (Heymann et al., 2021) for additional details. Consider a simple causal model shown below, where variable A is a “common” cause of variables B and C . In a structural causal model (Pearl, 2009), we consider the universe of variables $\{A, B, C\}$ to be subdivided into “exogenous” variables U with no parents in the model, below $U = \{A\}$, and “endogenous” variables $V = \{B, C\}$ whose parents include exogenous and endogenous variables.



Let us illustrate how information fields can be used to represent such structural causal models. Let the three variables above all be binary, so each variable can be viewed as a decision maker whose decision space $U_A = U_B = U_C = \{0, 1\}$. Let the associated σ -algebras be defined as by the discrete topology $\mathcal{F}_A = \mathcal{F}_B = \mathcal{F}_C = \{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$. Let the states of nature be defined as $\Omega = \{0, 1\}^3$, with the associated Borel topology $\mathcal{B} = 2^\Omega$. We can think of $\Omega = \Omega_A \times \Omega_B \times \Omega_C$, and $\mathcal{B} = \mathcal{B}_A \times \mathcal{B}_B \times \mathcal{B}_C$. To specify the causal DAG model fully, we need to specify the conditional probability distributions, which we can do using information fields for each variable.

Consider the exogenous variable A . Since it has no parent in the model, its value depends only on the measure of uncertainty from the external environment, hence we can write its information field $\mathcal{I}_A \subset \mathcal{B}_A \times \{\emptyset, \Omega_B\} \times \{\emptyset, \Omega_C\} \times \{\emptyset, U_A\}$. Note that a variable in a structural causal model cannot be “self-aware”, that is, its value cannot depend on its own value! Hence, the condition $\{\emptyset, U_A\}$ is imposed. On the other hand, the information field for variable C depends on the values of the other two variables, so its information field can be written as $\mathcal{I}_C \subset \{\emptyset, \Omega_A\} \times \{\emptyset, \Omega_B\} \times \mathcal{F}_A \times \mathcal{B}_C \times \mathcal{F}_B \times \{\emptyset, U_C\}$. That is, the value taken by C depends on the values taken by A and B and its own uncertainty.

Definition 23. A **causal UDM** is defined as one where each object $\mathcal{M} = (U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha, (\Omega, \mathcal{B}, P))$, where $\alpha \in X$, a finite space of variables. U_α is a non-empty set that defines the range of values that variable α can take. \mathcal{F}_α is a σ -algebra of measurable sets for variable α . The triple (Ω, \mathcal{B}, P) is a probability space, where \mathcal{B} is a σ -algebra of measurable subsets of sample space Ω . The **information field** $\mathcal{I}_\alpha \subset \mathcal{F}$ represents the “receptive field” of an element $\alpha \in X$, namely the set of other elements $\beta \in X$ whose values α must consult in determining its own value. We impose the restriction that the information field \mathcal{I}_α respect the Alexandroff topology on X , so that $\mathcal{I}_\alpha \subset \mathcal{F}(U_\alpha)$, where U_α is the minimal basic open set associated with element $\alpha \in X$.

Following structural causal models (Pearl, 2009), we can decompose the elements of a causal UDM object into disjoint subsets $X = U \sqcup V$, where U represents “exogenous” variables that have no parents, namely α is exogenous precisely

when $\mathcal{I}_\alpha \subset \mathcal{F}(\emptyset)$, and V are “endogenous” variables whose values are defined by measurable functions over exogenous and endogenous variables. Note that the probability space can be defined over the “exogenous” variables $\alpha \in U$, in which case it is convenient to attach a local probability space $(\Omega_\alpha, \mathcal{B}_\alpha, P)$ to each exogenous variable, where $\mathcal{B}_\alpha \subset \mathcal{B}$. We define conditional independence with respect to the induced information fields over the open sets of the Alexandroff space.

Definition 24. Given the induced probability space over information fields in a causal UDM object, a stochastic basis is a sequence of information fields $\mathcal{G} = \mathcal{I}_1, \dots, \mathcal{I}_n$ such that for $1 \leq i \leq n-1$, $\mathcal{I}_i \subset \mathcal{I}_{i+1}$, and $\cup_{i=1}^n \mathcal{I}_i = \mathcal{F}$. Two such sequences \mathcal{G}_1 and \mathcal{G}_2 are **conditionally independent** given the base σ -algebra \mathcal{F} , if for all subsets $A \in \mathcal{G}_1$, $B \in \mathcal{G}_2$, it follows that $P(A|B|\mathcal{F}) = P(A|\mathcal{F})P(B|\mathcal{F})$.

Definition 25. The **decision field** $U = \prod_{\alpha \in X} U_\alpha$ defines the space of all possible values of the variables in a causal UDM object, where the cartesian product is interpreted as a map $u : X \rightarrow \cup_{\alpha \in X} U_\alpha$ such that $u(\alpha) \equiv u_\alpha \in U_\alpha$.

Definition 26. For any subset of elements $B \in X$, let P_B denote the projection of the product $\prod_{\alpha \in X} U_\alpha$ upon the product $\prod_{\beta \in B} U_\beta$, that is $P_B(u)$ is simply the restriction of u to the domain B .

Definition 27. The product σ -algebra is defined as $\prod_{\alpha \in B} \mathcal{F}_\alpha$ over $\prod_{\alpha \in B} U_\alpha$, where $\mathcal{F}(B)$ is the smallest sigma-field such that P_B is measurable. Note that if $B_1 \subset B_2$, then $\mathcal{F}_{B_1} \subset \mathcal{F}_{B_2}$. The finest sigma-field $\mathcal{F}(X) = \prod_{\alpha \in X} \mathcal{F}_\alpha$.

Definition 28. A causal UDM object \mathcal{M} is **causally faithful** with respect to the probability distribution P over \mathcal{M} if every conditional independence in the topology, as defined in Definition 24, is satisfied by the distribution P , and vice-versa, every conditional independence property of the P is satisfied by the topology.

We can now formally define what it means to “solve” a causal UDM object \mathcal{M} . We impose the requirement that each variable $\alpha \in X$ must compute its value using a function measurable on its own information field.

Definition 29. Let the policy function f_α of each element $\alpha \in X$ be constrained so that $f_\alpha : U \times \Omega \rightarrow U_\alpha$ is measurable on the product σ -algebra $\mathcal{I}_\alpha \times \mathcal{B}_\alpha$, namely $f_\alpha^{-1}(\mathcal{F}_\alpha) \subset \mathcal{I}_\alpha \times \mathcal{B}_\alpha$.

Definition 30. The causal UDM object $\mathcal{M} = (X, U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)$ is **measurably solvable** if for every $\omega \in \Omega$, the closed loop equations $P_\alpha(u) = f_\alpha(u, \omega)$ have a unique solution for all $\alpha \in X$, where for a fixed $\omega \in \Omega$, the induced map $\mathcal{M}^\omega : \Omega \rightarrow U$ is a measurable function from the measurable space (Ω, \mathcal{B}) into (U, \mathcal{F}) .

Definition 31. The causal UDM object $\mathcal{M} = (X, U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)$ is **stable** if for every $\omega \in \Omega$, the closed loop equations $P_\alpha(u) = f_\alpha(u, \omega)$ are solvable by a fixed constant ordering Ξ that does not depend on $\omega \in \Omega$.

Measurably solvable causal UDM objects generalize the corresponding property in a structural causal model (U, V, F, P) , which states that for any fixed probability distribution P defined over the exogenous variables U , each function f_i computes the value of variable $x_i \in V$, given the value of its parents $Pa(x_i)$ uniquely as a function of $u \in U$. This allows defining the induced distribution $P_u(V)$ over exogenous variables in a unique functional manner depending on some particular instantiation of the random exogenous variables U . Stable models are those where the ordering of variables is fixed. We now extend the notion of *recursive* causal models in DAGs (Pearl, 2009) to finite topological spaces.

Definition 32. The causal UDM object $\mathcal{M} = (X, U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)$ is a **recursively causal model** if there exists an ordering function $\psi : X \rightarrow \Xi_n$, where Ξ_n is the set of all injective (1-1) mappings of $(1, \dots, n)$ to the set X , such that for any $1 \leq k \leq n$, the information field of variable α_k in the ordering Ξ_n is contained in the joint information fields of the variables preceding it:

$$\mathcal{I}_{\alpha_k} \subset \mathcal{F}(\alpha_1, \dots, \alpha_{k-1}) \quad (7)$$

In other words, the ordering ψ essentially proves a filtration of the σ -algebras over the previous variables to make the causal UDM object solvable. Note this property generalizes the recursive property in DAG models. What recursively causal means in the above definition is that element α_k has the information needed to compute its value based on the values of the variables that preceded it in the ordering given by ψ , and crucially, this ordering need not be the same for every element $\omega \in \Omega$ in the sample space. That is, for some setting of the exogenous variables, it may very well turn out that the ordering changes. This variability is not the case in DAG models, where there is an assumption of a fixed ordering on the DAG induced by the partial ordering, which is independent of any randomness in the exogenous variables. Finally, we define causal interventions in finite topological spaces prior to describing algorithms for learning causal finite space models.

Definition 33. A **causal intervention** $do(\beta=u_\beta)$ in a causal UDM object $\mathcal{M} = (X, U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)$ is defined as the subobject \mathcal{M}_β whose information fields \mathcal{I}_α are exactly the same as in \mathcal{M} for all elements $\alpha \neq \beta$, and the information field of the intervened element β is defined to be $\mathcal{I}_\beta \subset \mathcal{F}(\emptyset) \times \mathcal{B}_\beta$. Note that since the only measurable function on

$\mathcal{F}(\emptyset)$ is the constant function, whose value depends on a random sample space element $\omega \in \Omega_\beta$, this generalizes the notion of causal intervention in DAGs, where an intervened node has all its incoming edges deleted.³

4.3 UDMs based on Markov Decision Processes

We now briefly describe the (sub) category of UDMs, where each object represents a (finite) Markov decision process (MDP) (Puterman, 1994). Recall that an MDP is defined by a tuple $\langle S, A, \Psi, P, R \rangle$, where S is a discrete set of states, A is the discrete set of actions, $\Psi \subset S \times A$ is the set of admissible state-action pairs, $P : \Psi \times S \rightarrow [0, 1]$ is the transition probability function specifying the one-step dynamics of the model, where $P(s, a, s')$ is the transition probability of moving from state s to state s' in one step under action a , and $R : \Psi \rightarrow \mathbb{R}$ is the expected reward function, where $R(s, a)$ is the expected reward for executing action a in state s . MDP homomorphisms can be viewed as a principled way of abstracting the state (action) set of an MDP into a “simpler” MDP that nonetheless preserves some important properties, usually referred to as the stochastic substitution property (SSP).

Definition 34. A UDM MDP homomorphism (Ravindran and Barto, 2003) from object $M = \langle S, A, \Psi, P, R \rangle$ to $M' = \langle S', A', \Psi', P', R' \rangle$, denoted $h : M \rightarrow M'$, is defined by a tuple of surjections $\langle f, \{g_s | s \in S\} \rangle$, where $f : S \rightarrow S', g_s : A_s \rightarrow A'_{f(s)}$, where $h((s, a)) = \langle f(s), g_s(a) \rangle$, for $s \in S$, such that the stochastic substitution property and reward respecting properties below are respected:

$$P'(f(s), g_s(a), f(s')) = \sum_{s'' \in [s']_f} P(s, a, s'') \quad (8)$$

$$R'(f(s), g_s(a)) = R(s, a) \quad (9)$$

Given this definition, the following result is straightforward to prove.

Theorem 2. The UDM category \mathcal{C}_{MDP} is defined as one where each object c is defined by an MDP, and morphisms are given by MDP homomorphisms defined by Equation 8.

Proof: Note that the composition of two MDP homomorphisms $h : M_1 \rightarrow M_2$ and $h' : M_2 \rightarrow M_3$ is once again an MDP homomorphism $h' \circ h : M_1 \rightarrow M_3$. The identity homomorphism is easy to define, and MDP homomorphisms, being surjective mappings, obey associative properties. \square

4.4 UDM Category of Predictive State Representations

We now define the UDM (sub)category \mathcal{C}_{PSR} of predictive state representations (Thon and Jaeger, 2015), based on the notion of homomorphism defined for PSRs proposed in (Soni and Singh, 2007). Recall that a PSR is (in the simplest case) a discrete controlled dynamical system, characterized by a finite set of actions A , and observations O . At each clock tick t , the agent takes an action a_t and receives an observation $o_t \in O$. A *history* is defined as a sequence of actions and observations $h = a_1 o_1 \dots a_k o_k$. A *test* is a possible sequence of future actions and observations $t = a_1 o_1 \dots a_n o_n$. A test is successful if the observations $o_1 \dots o_n$ are observed in that order, upon execution of actions $a_1 \dots a_n$. The probability $P(t|h)$ is a prediction of that a test t will succeed from history h .

A state ψ in a PSR is a vector of predictions of a suite of *core tests* $\{q_1, \dots, q_k\}$. The prediction vector $\psi_h = \langle P(q_1|h) \dots P(q_k|h) \rangle$ is a sufficient statistic, in that it can be used to make predictions for any test. More precisely, for every test t , there is a $1 \times k$ projection vector m_t such that $P(t|h) = \psi_h \cdot m_t$ for all histories h . The entire predictive state of a PSR can be denoted Ψ .

Definition 35. In the UDM category \mathcal{C}_{PSR} defined by PSR objects, the morphism from object Ψ to another Ψ' is defined by a tuple of surjections $\langle f, v_\psi(a) \rangle$, where $f : \Psi \rightarrow \Psi'$ and $v_\psi : A \rightarrow A'$ for all prediction vectors $\psi \in \Psi$ such that

$$P(\psi' | f(\psi), v_\psi(a)) = P(f^{-1}(\psi') | \psi, a) \quad (10)$$

for all $\psi' \in \Psi', \psi \in \Psi, a \in A$.

Theorem 3. The UDM category \mathcal{C}_{PSR} is defined by making each object c represent a PSR, where the morphisms between two PSRs $h : c \rightarrow d$ is defined by the PSR homomorphism defined in (Soni and Singh, 2007).

Proof: Once again, given the homomorphism definition in Definition 35, the UDM category \mathcal{P}_{PSR} is easy to define, given the surjectivity of the associated mappings f and v_ψ . \square

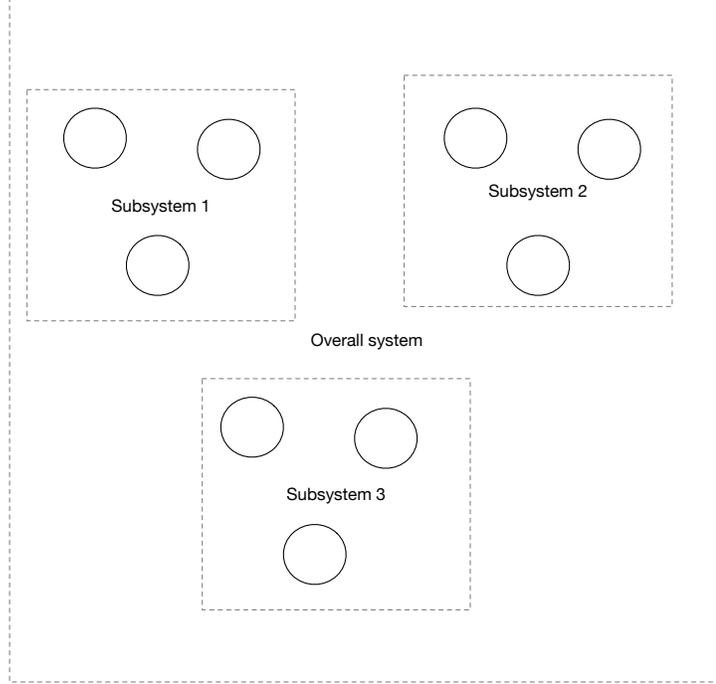


Figure 6: Organization of a UDM object into subsystems is based on a finite space topology induced by the information field structure.

5 Topology associated with UDM objects

Figure 6 illustrates a simple way to decompose a UDM object into sub-objects. The information field structure induces a finite space topology that enables decomposing a complex UDM object into subobjects. We define the closure of an element $\alpha \in A$ as the set of elements on whom it depends for information. These closure sets will define a finite topology on the space of decision makers, enabling the decomposition of complex objects into more manageable pieces. The induced topology has a rich structure, and has many consequences for organizing computation.

Definition 36. A subset of decision makers $B \subset A$ in a UDM object form a **subsystem** if the data requirements of members of the set only depend on the actions of nature, and the actions of the members of the set, and is independent of the actions of the non-members. More precisely, $B \subset A$ is a subsystem if for all $\alpha \in B$, $\mathcal{I}_\alpha \subset \mathcal{F}(B)$. If B is a subsystem, the induced UDM object $\langle B, (\Omega, \mathcal{B}, P), (U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_{\alpha B})_{\alpha \in B} \rangle$ is also a valid UDM object by itself, where the induced information subfield $\mathcal{I}_{\alpha B}$ is the canonical projection of \mathcal{I}_B upon H_B .

Definition 37. The **closure** of a decision maker $\alpha \in A$ in a UDM object $\langle A, (\Omega, \mathcal{B}, P), (U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)_{\alpha \in A} \rangle$ is the smallest subsystem containing α , denoted by $\{\alpha\}$.

Definition 38. The **preorder** relationship between decision makers, denoted $\alpha \leftarrow \beta$ is defined by the containment between the closure sets, namely $\alpha \leftarrow \beta$ if and only if $\{\alpha\} \subset \{\beta\}$.

Note that the \rightarrow relation defined above is a preorder because it is clearly reflexive and transitive. To explore more interesting special cases of this relationships, we need to introduce some additional notions from the topology of finite spaces.

Theorem 4. The subsystems of a UDM object $\langle A, (\Omega, \mathcal{B}, P), (U_\alpha, \mathcal{F}_\alpha, \mathcal{I}_\alpha)_{\alpha \in A} \rangle$ induce a finite space topology on the space A of decision makers.

Proof: (adapted from (Witsenhausen, 1975)): Recall that in a finite space topology (Barmak, 2011), the collection of subsets of A termed “open” sets are closed under arbitrary unions and intersections (it’s worth pointing out that in the general case, topological spaces require finite intersections). As the complement of an open set is a closed set, the set of closed sets is also closed under intersection and union. Given two subsystems S_1 and S_2 , if element $\alpha \in S_1 \cup S_2$, then

³Our definition of causal intervention differs from that proposed in causal information fields (Heymann et al., 2021), where additional intervention nodes were added to the model.

either $\mathcal{I}_\alpha \subset \mathcal{F}(S_1)$ or $\mathcal{I}_\alpha \subset \mathcal{F}(S_2)$. It follows that $\mathcal{I}_\alpha \subset \mathcal{F}(S_1) \cup \mathcal{F}(S_2) = \mathcal{F}(S_1 \cup S_2)$. The proof for closure under intersection is similar. \square

Given this theorem, we can immediately bring to bear the powerful tools of algebraic topology (Barmak, 2011) of finite topological spaces, also called Alexandroff spaces (Alexandroff, 1937), to analyze the topological properties of UDM objects. Essentially, we are showing that UDMs form a subcategory in the category of all topological spaces (as each UDM object is a topological space in its own right). We briefly review some of the key properties that we will use below.

Definition 39. *The neighborhood of an element x in a finite space X is a subset $V \subset X$ such that $x \in U$ for some open set $U \subset V$.*

- X is a Kolmogorov (or T_0) finite space X if each pair of points $x, y \in X$ is distinguishable in the space, namely for each $x, y \in X$, there is an open set $U \in \mathcal{U}$ such that $x \in U$ and $y \notin U$. Alternatively, if $x \in U$ if and only if $y \in U$, $\forall U \in \mathcal{U}$ implies that $x = y$.
- X is a T_1 finite space if element $x \in X$ defines a closed set $\{x\}$.
- X is a T_2 finite space or a Hausdorff space if any two points have distinct neighborhoods.

Lemma 1. *If X is a T_2 space, then it is a T_1 space. If X is a T_1 space, then it is a T_0 space.*

The key concept that gives finite (Alexandroff) spaces its power is the definition of the minimal open basis. First, we introduce the concept of a basis in a topological space.

Definition 40. *A basis for the topological space X is a collection \mathcal{B} of subsets of X such that*

- For each $x \in X$, there is at least one $B \in \mathcal{B}$ such that $x \in B$.
- If $x \in B' \cap B''$, where $B, B'' \in \mathcal{B}$, then there is at least one $B \in \mathcal{B}$ such that $x \in B \subset B' \cap B''$.

The topology \mathcal{U} generated by the basis \mathcal{B} is the set of subsets U such that for every $x \in U$, there is a $B \in \mathcal{B}$ such that $x \in B \subset U$. In other words, $U \in \mathcal{U}$ if and only if U can be generated by taking unions of the sets in the basis \mathcal{B} . Now, we turn to giving the most important definition in Alexandroff spaces, namely the *unique minimal basis*.

Lemma 2. *Let X be a finite Alexandroff space. For each $x \in X$, define the open set U_x to be the intersection of all open sets that contain x . Define the relationship \leq on X by $x \leq y$ if $x \in U_y$, or equivalently, $U_x \subset U_y$ (where $x < y$ if the inclusion is strict). The open sets U_x constitute a **unique minimal basis** \mathcal{B} for X in that if \mathcal{C} is another basis for X , then $\mathcal{B} \subset \mathcal{C}$. Alternatively, define the closed sets $F_x = \{y \mid y \geq x\}$, which provide an equivalent characterization of finite Alexandroff spaces.⁴*

Note that the relation \leq defined above is a *preorder* because it is reflexive (clearly, $x \in U_x$) and transitive (if $x \in U_y$, and $y \in U_z$, then $x \in U_z$). However, in the special case where the finite space X has a T_0 topology, then the relation \leq becomes a partial ordering.

5.1 Classes of UDMs

We now describe a way to decompose UDM objects based on information fields. Witsenhausen (1975) defines the following 10 classes of information structures, each of which leads to a distinct type of UDM object. This decomposition shows the importance of the topology induced on a UDM object based on information structures.

1. **Monic:** A monic UDM object has only decision maker $A = \{\alpha\}$, and its information field $\mathcal{I}_\alpha \subset \mathcal{F}(\emptyset)$. In other words, the decision maker α only requires access to the state of nature, and does not obviously need information from any other decision maker, including itself!
2. **Team:** A team UDM object can be viewed as an independent set of decision makers, all of whom only need access to the state of nature, that is $\mathcal{I}_\alpha \subset \mathcal{F}(\emptyset)$.
3. **Sequential:** A sequential UDM object is one where there exists a fixed ordering $\{\alpha_1, \dots, \alpha_n\}$ of decision makers from A such that for any $1 \leq k \leq n$, it holds that $\mathcal{I}_{\alpha_k} \subset \mathcal{F}(\{\alpha_1, \dots, \alpha_{k-1}\})$. Sequential systems satisfy the causality condition with a constant ordering function ϕ .
4. **Classical:** A UDM object is called classical if it is sequential, and furthermore, $\mathcal{I}_0 \in \mathcal{F}(\emptyset)$, $\mathcal{I}_{k-1} \subset \mathcal{I}_k$, for all $k = 2 \dots, n$.

⁴The minimal basic closed sets in a T_0 finite Alexandroff space correspond to the *ancestral sets* in a DAG graphical model.

5. **Strictly classical:** A UDM object is strictly classical if it is classical, and $[\mathcal{F}_{\alpha_k}] \subset \mathcal{I}_{k+1}$, where $[\mathcal{F}_{\alpha_k}]$ is the cylindrical extension of \mathcal{F}_{α_k} to all of H .
6. **Strictly quasiclassical:** In a strictly quasiclassical UDM object, if $\alpha \leftarrow \beta$, $\alpha \neq \beta$ implies that $\mathcal{I}_\alpha \cup [\mathcal{F}_\alpha] \subset \mathcal{I}_\beta$.
7. **Quasi-classical:** A UDM object is quasi-classical if it is sequential, and if $\alpha \leftarrow \beta$, then $\mathcal{I}_\alpha \subset \mathcal{I}_\beta$.
8. **Causal:** See Definition 19.
9. **Solvable:** See Definition 17.
10. **Without self-information:** A UDM object has no self-information if for all its decision elements $\alpha \in A$, it holds that $\mathcal{I}_\alpha \subset \mathcal{F}(A - \{\alpha\})$.

A detailed study of the properties ensuing from this classification can be found in (Witsenhausen, 1975), for example, systems with T_0 topologies are precisely those that induce a partial ordering on computing elements, and also define a sequential system. We will discuss some of these properties based on our generalization of the intrinsic model using category theory below.

6 Functors, Natural Transformations, and the Yoneda Lemma

We now introduce some additional terminology from category theory, including the important idea of *functors* that map from one category to another, preserving the underlying structure of morphisms, *natural transformations* that map from one functor to another, and finally one of most important results in category theory, the *Yoneda lemma* and how it can be used to construct representations of functors and associated universal representations. Our goal in the subsequent section is to use this machinery to construct universal representations of intrinsic models.

Definition 41. A **covariant functor** $F : \mathcal{C} \rightarrow \mathcal{D}$ from category \mathcal{C} to category \mathcal{D} is defined as the following:

- An object $\mathcal{F}X$ of the category \mathcal{D} for each object X in category \mathcal{C} .
- A morphism $\mathcal{F}f : \mathcal{F}X \rightarrow \mathcal{F}Y$ in category \mathcal{D} for every morphism $f : X \rightarrow Y$ in category \mathcal{C} .
- The preservation of identity and composition: $\mathcal{F} id_X = id_{\mathcal{F}X}$ and $(\mathcal{F}g)(\mathcal{F}f) = \mathcal{F}(fg)$ for any composable morphisms $f : X \rightarrow Y, g : Y \rightarrow Z$.

Definition 42. A **contravariant functor** $F : \mathcal{C} \rightarrow \mathcal{D}$ from category \mathcal{C} to category \mathcal{D} is defined exactly like the covariant functor, except all the mappings are reversed. In the contravariant functor $\mathcal{F} : \mathcal{C}^{\text{op}} \rightarrow \mathcal{D}$, every morphism $f : X \rightarrow Y$ is assigned the reverse morphism $\mathcal{F}f : \mathcal{F}Y \in \mathcal{F}X$ in category \mathcal{D} .

Our goal is to construct covariant and contravariant functorial representations of intrinsic models. To this end, we introduce the following functors that will prove of value below:

- For every object X in a category \mathcal{C} , there exists a covariant functor $\mathcal{C}(X, -) : \mathcal{C} \rightarrow \mathbf{Set}$ that assigns to each object Z in \mathcal{C} the set of morphisms $\mathcal{C}(X, Z)$, and to each morphism $f : Y \rightarrow Z$, the pushforward mapping $f_* : \mathcal{C}(X, Y) \rightarrow \mathcal{C}(X, Z)$.
- For every object X in a category \mathcal{C} , there exists a contravariant functor $\mathcal{C}(-, X) : \mathcal{C}^{\text{op}} \rightarrow \mathbf{Set}$ that assigns to each object Z in \mathcal{C} the set of morphisms $\mathcal{C}(X, Z)$, and to each morphism $f : Y \rightarrow Z$, the pullback mapping $f^* : \mathcal{C}(Z, X) \rightarrow \mathcal{C}(Y, X)$.

From the above examples, it is now relatively straightforward to see how to define covariant and contravariant functors from the category of intrinsic models to the category of sets, but we need to develop a bit more machinery to understand the significance of these functorial representations.

Definition 43. Let $\mathcal{F} : \mathcal{C} \rightarrow \mathcal{D}$ be a functor from category \mathcal{C} to category \mathcal{D} . If for all objects X and Y in \mathcal{C} , the map $\mathcal{C}(X, Y) \rightarrow \mathcal{D}(\mathcal{F}X, \mathcal{F}Y)$, denoted as $f \mapsto \mathcal{F}f$ is

- injective, then the functor \mathcal{F} is defined to be **faithful**.
- surjective, then the functor \mathcal{F} is defined to be **full**.
- bijective, then the functor \mathcal{F} is defined to be **fully faithful**.

Our goal is to construct fully faithful functorial embeddings of intrinsic models, which gives us an embedding of intrinsic models into the category of sets.

6.1 Natural Transformations and the Yoneda Lemma

Definition 44. Given two functors $\mathcal{F}, \mathcal{G} : \mathcal{C} \rightarrow \mathcal{D}$ that map from category \mathcal{C} to category \mathcal{D} , a natural transformation $\eta : \mathcal{F} \rightarrow \mathcal{G}$ consists of a morphism $\eta_X : \mathcal{F}X \rightarrow \mathcal{G}X$ for each object X in \mathcal{C} . Moreover, these morphisms should satisfy the following property, that is the diagram below should commute:

$$\begin{array}{ccc} \mathcal{F}X & \xrightarrow{\mathcal{F}f} & \mathcal{F}Y \\ \downarrow \eta_X & & \downarrow \eta_Y \\ \mathcal{G}X & \xrightarrow{\mathcal{G}f} & \mathcal{G}Y \end{array}$$

Definition 45. For any two functors $\mathcal{F}, \mathcal{G} : \mathcal{C} \rightarrow \mathcal{D}$, let $\text{Nat}(\mathcal{F}, \mathcal{G})$ denote the natural transformations from \mathcal{F} to \mathcal{G} . If $\eta_X : \mathcal{F}X \rightarrow \mathcal{G}X$ is an isomorphism for each X in category \mathcal{C} , then the natural transformation η is called a **natural isomorphism** and \mathcal{F} and \mathcal{G} are naturally isomorphic, denoted as $\mathcal{F} \cong \mathcal{G}$.

The machinery of natural transformations between functors enables making concrete the central philosophy underlying category theory, which is construct representations of objects in terms of their interactions with other objects. Unlike set theory, where an object like a set is defined by listing its elements, in category theory objects have no explicit internal structure, but rather are defined through the morphisms that they define with respect to other objects. The celebrated Yoneda lemma makes this philosophical statement more precise.

Theorem 5. Yoneda Lemma: For every object X in category \mathcal{C} , and every contravariant functor $\mathcal{F} : \mathcal{C}^{\text{Op}} \rightarrow \text{Set}$, the set of natural transformations from $\mathcal{C}(-, X)$ to \mathcal{F} is isomorphic to $\mathcal{F}X$.

That is, the natural transformations from $\mathcal{C}(-, X)$ to \mathcal{F} serve to fully characterize the object $\mathcal{F}X$ up to isomorphism. In the special circumstance when the set-valued functor $\mathcal{F} = \mathcal{C}(-, Y)$, the Yoneda lemma asserts that $\text{Nat}(\mathcal{C}(-, X), \mathcal{C}(-, Y)) \cong \mathcal{C}(X, Y)$. In other words, a pair of objects are isomorphic $X \cong Y$ if and only if the corresponding contravariant functors are isomorphic, namely $\mathcal{C}(-, X) \cong \mathcal{C}(-, Y)$.

6.2 Presheaf Representations

A very important class of representations that follow from the Yoneda lemma are *presheafs* $\mathcal{C}(-, X)$. Given any two categories \mathcal{C}, \mathcal{D} , we can always define the new category $\mathcal{D}^{\mathcal{C}}$, whose objects are functors $\mathcal{C} \rightarrow \mathcal{D}$, and whose morphisms are natural transformations. If we take $\mathcal{D} = \text{Set}$, and consider the contravariant version $\text{Set}^{\mathcal{C}^{\text{Op}}}$, we obtain a category whose objects are presheafs. Presheafs have some very nice properties, which makes them a *topos* (Goldblatt, 2006).

Given a category of intrinsic models \mathcal{C}_I , or in particular a category of MDPs \mathcal{C}_{MDP} with bisimulation homomorphism or a category of PSRs \mathcal{C}_{PSR} with the defined PSR homomorphism, we can clearly apply the Yoneda lemma to construct presheaf representations of these decision making objects. A detailed study of each individual case is outside the scope of this introductory paper, and is a topic for future research.

7 Homotopical Representations of UDMs

We have seen that information fields induce a finite topological space over a UDM, enabling the decomposition of the UDM object into subsystems. In this section, we explore the topological ramifications of this idea further. Homotopy is a fundamental idea in algebraic topology, and we build on the use of homotopical constructions over finite topological spaces (Barmak, 2011). A fundamental idea throughout mathematics is that of gleaning insight into the structure of one space by probing it with objects from another space. Thus, a fundamental way to understand the category of groups is to map it to the category of group representations. Similarly, in a topological space \mathbf{Top} , two objects X and Y are considered isomorphic if the corresponding sets $\mathbf{Top}(\mathbf{Z}, X)$ and $\mathbf{Top}(\mathbf{Z}, Y)$ are isomorphic. The Yoneda lemma described above allows us to construct functors from any category \mathcal{C} to the category Set . Our goal is to be able to compute invariant representations of UDM objects, such as their homotopies, and understand how to compute the *fundamental group* associated with a UDM object. We begin by reviewing some basic material on connectivity in finite topological spaces, and then show how various UDM objects can be compared in terms of their subsystem topologies.

7.1 Connectivity in UDMs

Since UDMs define a finite topological space, we can build on the core ideas of connectivity in such spaces (Barmak, 2011). Every concept in a topological space must be defined in terms of the open (or closed) set topology, and that

Table 2: Examples of UDM objects $A = \{a, b, c\}$ with different subsystem topologies. A proper open set is any set other than \emptyset or A (which are in any topology). P_n is a topology on a set of size n with only one proper open set. D_n is the discrete topology over n elements. $P_{m,n}$ are topologies where the proper open sets are all non-empty subsets of a subset of size m . The \cong equivalence relation is homotopy equivalence. See text for explanation.

Proper Open Sets	Name	T_0 ?	Connected?	Equivalent graphical model
All	D_3	yes	no	HEDG (hyper-edge over (a,b,c))
b, c		yes	yes	DAG $b \rightarrow a, c \rightarrow a$ (collider over a)
$a, b, (a, b)$	$P_{2,3} \cong \mathbb{C}D_2$	yes	yes	Chain graph: $a \rightarrow c, b \rightarrow c, a - b$
$a, b, (a, b), (b, c)$	$D_1 \sqcup P_2$	yes	no	DAG with node a disconnected, $b \rightarrow c$
a	P_3	no	yes	Chain graph: $a \rightarrow b, a \rightarrow c$, and $b - c$

includes (path) connectivity. The crucial idea here is that connectivity is defined in terms of a continuous mapping from the unit interval $I = (0, 1)$ to a topological space X .

Lemma 3. *A function $f : X \rightarrow Y$ between two finite spaces is continuous if and only if it is order-preserving, meaning if $x \leq x'$ for $x, x' \in X$, this implies $f(x) \leq f(x')$.*

Definition 46. *We call two points $x, y \in X$ **comparable** if there is a sequence of elements x_0, \dots, x_n , where $x_0 = x, x_n = y$ and for each pair x_i, x_{i+1} either $x_i \leq x_{i+1}$ or $x_i \geq x_{i+1}$. A **fence** in X is a sequence x_0, x_1, \dots, x_n of elements such that any two consecutive elements are comparable. X is **order connected** if for any two elements $x, y \in X$, there exists a fence starting in x and ending in y .*

Lemma 4. *Let x, y be two comparable points in a finite space X . Then, there exists a path from x to y in X , that is, a continuous map $\alpha : (0, 1) \rightarrow X$ such that $\alpha(0) = x$ and $\alpha(1) = y$.*

Lemma 5. *Let X be a finite space. The following are equivalent: (i) X is a connected topological space. (ii) X is an order-connected topological space (iii) X is a path-connected topological space.*

A crucial strength of the topological perspective is the ability to combine two UDM objects X and Y into a new space, which can generate a rich panoply of new objects. Here are a few of the myriad ways in which UDM objects can be combined Munkres (1984). Table 2 illustrates some of these ways of combining spaces for a small UDM object whose space of elements X is comprised of just three agents.

- Subspaces: The **subspace** topology on $A \subset X$ is defined by the set of all intersections $A \cap U$ for open sets U over X .
- Quotient topology: The **quotient topology** on U defined by a surjective mapping $q : X \rightarrow Y$ is the set of subsets U such that $q^{-1}(U)$ is open on X .
- Union: The **topology of the union** of two spaces X and Y is given by their disjoint union $X \sqcup Y$, which has as its open sets the unions of the open sets of X and that of Y .
- Product of two spaces: The **product topology** on the cartesian product $X \times Y$ is the topology with basis the "rectangles" $U \times V$ of an open set U in X with an open set V in Y .
- Wedge sum of two spaces: The **wedge sum** is the "one point" union of two "pointed" spaces (X, x_0) with (Y, y_0) , defined by $X \vee Y / x_0 \sim y_0$, the quotient space of the disjoint union of X and Y , where x_0 and y_0 are identified.
- Smash product: The **smash product** topology is defined as the quotient topology $X \wedge Y = X \times Y / X \vee Y$.
- Non-Hausdorff cone: The **non-Hausdorff cone** of topological space X with $Y = \{*\}$ yields the new space $\mathbb{C}(X)$, whose open sets are now $\mathcal{O}_{\mathbb{C}(X)} = \mathcal{O}_X \cup \{X \cup \{*\}\}$.
- Non-Hausdorff suspension: The **non-Hausdorff suspension** of topological space X with $Y = \{+, -\}$ yields the new space $\mathbb{S}(X)$, whose open sets are now $\mathcal{O}_{\mathbb{S}(X)} = \mathcal{O}_X \cup \{X \cup \{+, -\}\}$.

7.2 UDM Homotopies over Finite Topological Spaces

Definition 47. *Let $f, g : X \rightarrow Y$ be two continuous maps between finite space topologies X and Y . We say f is homotopic to g , denoted as $f \cong g$ if there exists a continuous map $h : X \times [0, 1] \rightarrow Y$ such that $h(x, 0) = f(x)$ and $h(x, 1) = g(x)$. In other words, there is a smooth "deformation" between f and g , so we can visualize f being slowly warped into g . Note that \cong is an equivalence relation, since $f \cong f$ (reflexivity), and if $f \cong g$, then $g \cong f$ (symmetry), and finally $f \cong g, g \cong h \implies f \cong h$ (transitivity).*

Definition 48. A map $f : X \rightarrow Y$ is a homotopy equivalence if there exists another map $g : Y \rightarrow X$ such that $g \circ f \cong id_X$ and $f \circ g \cong id_Y$, where id_X and id_Y are the identity mappings on X and Y , respectively.

Definition 49. A topological space X is **contractible** if the identity map $id_X : X \rightarrow X$ is homotopically equivalent to the constant map $f(x) = c$ for some $c \in X$.

For example, any convex subset $A \subset \mathbb{R}^n$ is contractible. Let $f(x) = c, c \in A$ be the constant map. Define the homotopy $H : A \times I \rightarrow X$ as equal to $H(x, t) = tc + (1 - t)x$. Note that at $t = 0$, we have $H(x, 0) = x$, and that at $t = 1$, we have $H(x, 1) = c$, and since A is a convex subset, the convex combination $tc + (1 - t)x \in A$ for any $t \in [0, 1]$.

Theorem 6. If X is a finite topological space containing a point y such that the only open (or closed) subset of X containing y is X itself, then X is contractible. In particular, the non-Hausdorff cone $\mathbb{C}(X)$ is contractible for any X .

Proof: Let $Y = \{*\}$ denote the space with a single element, $*$. Define the retraction mapping $r : X \rightarrow *$ by $r(x) = *$ for all $x \in X$, and define the inclusion mapping $i : Y \rightarrow X$ by $i(*) = y$. Clearly, $r \circ i = id_Y$. Define the homotopy $h : X \rightarrow I \rightarrow X$ by $h(x, t) = x$ if $t < 1$, and $h(x, 1) = y$. Then, h is continuous, because for any open set U in X , if $y \in U$, then clearly $U = X$ (as X is the only open set containing y), and hence $h^{-1}(U) = X \times I$, which is open. If on the other hand, $y \notin U$, then $h^{-1}(U) = U \times [0, 1)$. It follows that h is a homotopy $h \cong id_X = i \circ r$. \square

The following lemma is of crucial importance, showing how elements of a topological space that can be removed, reducing model size.

Definition 50. A point x in a finite topological space X is **maximal** if there is no $y > x$, and **minimal** if there is no $y < x$.

Lemma 6. If X is an finite space, then U_x is contractible. In particular, if X has a unique maximal point or unique minimal point, then X is contractible.

Definition 51. Let $f, g : X \rightarrow Y$ be two continuous maps between finite space topologies X and Y . We say f is homotopic to g , denoted as $f \cong g$ if there exists a continuous map $h : X \times [0, 1] \rightarrow Y$ such that $h(x, 0) = f(x)$ and $h(x, 1) = g(x)$. In other words, there is a smooth “deformation” between f and g , so we can visualize f being slowly warped into g . Note that \cong is an equivalence relation, since $f \cong f$ (reflexivity), and if $f \cong g$, then $g \cong f$ (symmetry), and finally $f \cong g, g \cong h \implies f \cong h$ (transitivity).

Definition 52. A map $f : X \rightarrow Y$ is a homotopy equivalence if there exists another map $g : Y \rightarrow X$ such that $g \circ f \cong id_X$ and $f \circ g \cong id_Y$, where id_X and id_Y are the identity mappings on X and Y , respectively.

7.3 Efficient Enumeration of Homeomorphically Distinct UDMs

Next, we turn to the fundamental problem of how to construct homomorphically distinct UDMs. In the definitions below, we focus purely on the topological structure of a UDM, namely the subsystem topology as defined in Definition 36.

Definition 53. For every object \mathcal{M} in a UDM with T_0 subsystem topology that defines a partial ordering \leq , define its associated **Hasse diagram** $H_{\mathcal{M}}$ as a directed graph which captures all the relevant order information of \mathcal{M} . More precisely, the vertices of $H_{\mathcal{M}}$ are the elements of \mathcal{M} , and the edges of $H_{\mathcal{M}}$ are such that there is a directed edge from x to y whenever $y \leq x$, but there is no other vertex z such that $y \leq z \leq x$.

General pre-ordered UDM objects can be reduced to partially ordered UDM objects with T_0 topologies up to homeomorphic equivalence.

Theorem 7. (Stong, 1966) Let (X, \mathcal{T}) be an arbitrary UDM object with a subsystem topology defining an associated preordering \leq . Let X_0 represent the quotient topological space X / \sim , where $x \sim y$ if $x \leq y$ and $y \leq x$. Then X_0 is a homotopically equivalent intrinsic model with T_0 separability, and the quotient map $q : X \rightarrow X_0$ is a homotopy equivalence. Furthermore, X_0 induces a partial ordering on the elements $x \in X_0$.

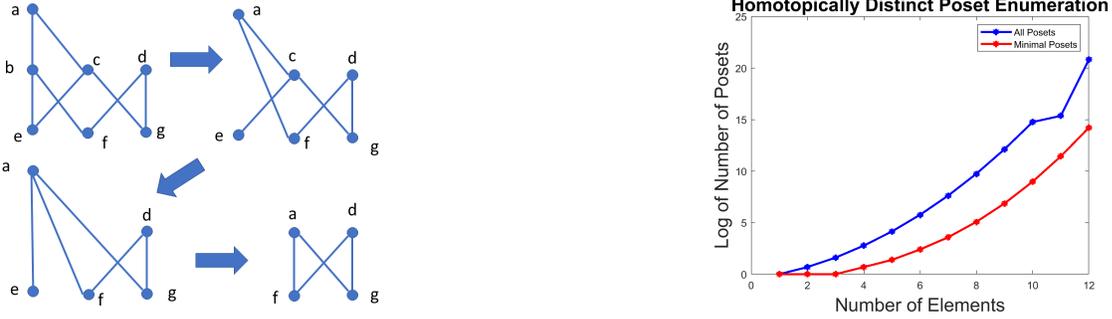
A key idea in the enumeration is to assume that each element in the Hasse diagram of the poset does not have an in-degree or out-degree of 1.

Definition 54. (Stong, 1966) An element $x \in X$ in a UDM object with T_0 subsystem topology X is a **down beat point** if x covers one and only one element of X . Alternatively, the set $\hat{U}_x = U_x \setminus \{x\}$ has a (unique) maximum. Similarly, $x \in X$ is an **up beat point** if x is covered by a unique element, or equivalently if $\hat{F}_x = F_x \setminus \{x\}$ has a (unique) minimum. A **beat point** is either a down beat or up beat point.

Definition 55. A subspace $A \subset X$ is called a strong deformation retract of X if there is a homotopy $F : X \times [0, 1] \rightarrow A$ such that $F(x, 0) = x, F(x, 1) \in A, F(a, t) = a$ for all $x \in X, t \in [0, 1], a \in A$.

Theorem 8. (Stong, 1966) Let X be a UDM object with T_0 subsystem topological model, and let $x \in X$ be a (down, up) beat point. Then the reduced object $X \setminus \{x\}$ is a **strong deformation retract** of X . An element x in a UDM object

Figure 7: Left: Constructing minimal UDM objects by removing beat points (Barmak, 2011; Stong, 1966). Right: Efficiently enumerating minimal UDM objects (based on the enumeration method in (Fix and Partias, 2008)).



Algorithm 1: Find Topologically Minimal UDM object with T_0 Subsystem Topology.

Input: General UDM object \mathcal{M} with a reflexive transitive pre-ordered structure induced by the subsystem topology.

Output: Minimal UDM object with T_0 subsystem topology homotopically equivalent to original pre-ordered object.

The algorithm uses homotopy theory to find the *core* T_0 description of a pre-ordered UDM object.

begin

Define the topology (X, \mathcal{U}) where $X = V$ and the open sets in \mathcal{U} are constructed from the induced pre-order \leq from \mathcal{M} . Define the minimal object (X_0, \mathcal{U}') , and set $X_0 = X$.

repeat

for $x, y \in X_0$ s.t. $x \leq y, y \leq x$ **do**

Remove x, y from X_0 , and replace them with a new variable $z = x \sim y$.

Set $X_0 \leftarrow X_0 \setminus \{x, y\} \cup \{z\}$. z represents the equivalence class that includes x and y .

end

for $x \in X_0$ **do**

Remove down beat points: If $\hat{U}_x = U_x \setminus \{x\}$ has a maximum, then $X_0 \leftarrow X_0 \setminus \{x\}$.

Remove up beat points: If $\hat{F}_x = F \setminus \{x\}$ has a minimum, then $X_0 \leftarrow X_0 \setminus \{x\}$.

end

until convergence;

Define the open sets $U_x \in \mathcal{U}'$ as $U_x = \{y \mid y \leq x\}$ for $x \in X_0$.

end

\mathcal{M} is an up beat point if and only if it has in-degree one in the associated Hasse diagram $H_{\mathcal{M}}$, i.e., it has only one incoming edge). Similarly, x is down beat if and only if it has out-degree one (it has only one outgoing edge).

Definition 56. A UDM object with a T_0 subsystem topological space is a **minimal** if it has no beat points. A **core** of a UDM object X is a strong deformation retract, which is a minimal finite space. The **minimal graph** of a minimal UDM object is its equivalent Hasse diagram.

Theorem 9. (Stong, 1966) **Classification Theorem:** A homotopy equivalence between minimal UDM objects is a homeomorphism. In particular, the core of a UDM object is unique up to homeomorphism and two UDM objects are homotopy equivalent if and only if they have homeomorphic cores.

Figure 7 illustrates the process of removing beat points to construct the minimal UDM object. b is an up beat point of X , c is an up beat point of $X \setminus \{b\}$, and e is an up beat point of $X \setminus \{b, c\}$. Similarly, points c and e are removed, resulting in the minimal object. The figure also shows that homeomorphic equivalences greatly reduces the search space of possible object structures. Note the plot is on log scale. For example, for 12 variables, the number of minimal objects is $< 0.1\%$ of the number of possible objects, a savings of three orders of magnitude.

Algorithm 1 determines a reduced UDM object based on discovering a quotient T_0 subsystem topology that is homotopically equivalent to original non-reduced object with a non- T_0 topology. Second, the algorithm further reduces the object to its core by removing *beat points* (Barmak, 2011; Stong, 1966).

8 Equilibration in UDMs

At the outset, we cautioned that the bulk of this paper is devoted to a study of information structures that underly decision making broadly, which precludes introducing particular solution methodologies, such as dynamic programming (Bertsekas, 2005), which is specific to sequential information structures. In this section, however, we discuss a broad solution methodology called *equilibration*, which applies to multiplayer network games, reinforcement learning, and to causal inference, based on a generalization of optimization called variational inequalities (VIs) (Nagurney, 1999). We use a running example from earlier in the paper of a producer consumer multiplayer game, as shown in Figure 5, and analyze its solution in depth in this section. VIs can be seen as a generalization of optimization, and frequently used to solve complex network games. This section is a condensed version of a recent paper on causal VIs (Mahadevan, 2021), which the reader is encouraged to read for further details.

8.1 Causal Variational Inequalities

Our variational formulation of causal inference is a synthesis of classical variational inequalities (Facchinei and Pang, 2003) and causal models (Imbens and Rubin, 2015; Pearl, 2009). More precisely, a causal variational inequality model $\mathcal{M} = \text{CVI}(F, K)$, where F is a collection of modular vector-valued functions defined as F_i , where $F_i : K_i \subset \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_i}$, with each K_i being a convex domain such that $\prod_i K_i = K$. We assume that the domains of each F_i range over a collection V of endogenous variables, and a set U of exogenous variables, where only the endogenous variables are subject to causal manipulation. We model each intervention as a submodel F_w , and each component of F_w reflects the effect of some manipulation of a subset $V_w \subseteq V$ of endogenous variables.

Definition 57. *The category \mathcal{C}_{CVI} of causal VIs is defined as one where each object is defined as a finite-dimensional causal variational inequality problem $\mathcal{M} = \text{CVI}(F, K)$, where the vector-valued mapping F depend on both deterministic and stochastic elements, namely $F(x) = E[F(x, \eta)]$, where η is a random variable defined over the probability space (Ω, \mathcal{F}, P) , $E[\cdot]$ denotes expectation with respect to the probability distribution P over the random variable η , and $F : K \rightarrow \mathbb{R}^n$ is a given continuous function, K is a given closed convex set, and $\langle \cdot, \cdot \rangle$ is the standard inner product in \mathbb{R}^n . A causal intervention is modeled as a submodel $\mathcal{M}_w = \text{CVI}(F_w, K)$, where $F_w(x) = E_w[F(x, \eta|\hat{w})]$, where \hat{w} denotes the intervention of setting of variable w to a specific non-random value, and where $E_w[\cdot]$ now denotes expectation with respect to the intervention probability distribution P_w . Solving a causal VI is defined as finding a vector $x^* = (x_1^*, \dots, x_n^*) \in K \subset \mathbb{R}^n$ such that*

$$\langle F_w(x^*), (y - x^*) \rangle \geq 0, \forall y \in K$$

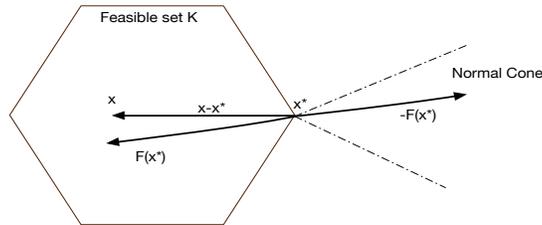


Figure 8: This figure provides a geometric interpretation of a causal variational inequality $\text{CVI}(F_w, K)$. The mapping F_w defines a vector field over the feasible set K and a probability space, where $E_w(F(x, \eta)|\hat{w})$ is the conditional mean vector field (denoted in the figure by F), computed over the intervention distribution P_w . At the solution point x^* , the vector field $F(x^*)$ is directed inwards at the boundary, and $-F(x^*)$ is an element of the normal cone $C(x^*)$ of K at x^* where the normal cone $C(x^*)$ at the vector x^* of a convex set K is defined as $C(x^*) = \{y \in \mathbb{R}^n | \langle y, x - x^* \rangle \leq 0, \forall x \in K\}$.

8.2 Properties of Mappings

The solution to a (causal) VI depends on the properties satisfied by the mapping F and the feasible space K . If K is compact and F is continuous, it is straightforward to prove using Brouwer's fixed point theorem that there is always at least one solution to any VI (see Theorem 11). However, to obtain a unique solution, a stricter condition is necessary.

Definition 58. $F(x)$ is monotone if $\langle F(x) - F(y), x - y \rangle \geq 0, \forall x, y \in K$.

Definition 59. $F(x)$ is strongly monotone if $\langle F(x) - F(y), x - y \rangle \geq \mu \|x - y\|_2^2, \mu > 0, \forall x, y \in K$.

Definition 60. $F(x)$ is Lipschitz if $\|F(x) - F(y)\|_2 \leq L\|x - y\|_2, \forall x, y \in K$.

Crucially, VI problems can only be converted into equivalent optimization problems when a very restrictive condition is met on the Jacobian of the mapping F , namely that it be symmetric. Most often, real-world applications of VIs, such as the example in Section 5, do not induce symmetric Jacobians.

Theorem 10. Assume $F(x)$ is continuously differentiable on K and that the Jacobian matrix $\nabla F(x)$ of partial derivatives of $F_i(x)$ with respect to (w.r.t) each x_j is symmetric and positive semidefinite. Then there exists a real-valued convex function $f : K \rightarrow \mathbb{R}$ satisfying $\nabla f(x) = F(x)$ with x^* , the solution of $VI(F, K)$, also being the mathematical programming problem of minimizing $f(x)$ subject to $x \in K$.

The algorithmic development of methods for solving VIs begins with noticing their connection to fixed point problems.

Theorem 11. The vector x^* is the solution of $VI(F, K)$ if and only if, for any $\alpha > 0$, x^* is also a fixed point of the map $x^* = P_K(x^* - \alpha F(x^*))$, where P_K is the projector onto convex set K .

In terms of the geometric picture of a VI illustrated in Figure 8, this property means that the solution of a VI occurs at a vector x^* where the vector field $F(x^*)$ induced by F on K is normal to the boundary of K and directed inwards, so that the projection of $x^* - \alpha F(x^*)$ is the vector x^* itself. This property forms the basis for the projection class of methods that solve for the fixed point.

8.3 Causal Variational Inequalities and Games

VIs are a mathematically elegant approach to modeling and solving equilibrium problems in game theory (Maschler et al., 2013). A *Nash game* consists of m players, where player i chooses a strategy x_i belonging to a closed convex set $X_i \subset \mathbb{R}^n$. After executing the joint action, each player is penalized (or rewarded) by the amount $f_i(x_1, \dots, x_m)$, where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function. A set of strategies $x^* = (x_1^*, \dots, x_m^*) \in \prod_{i=1}^m X_i$ is said to be in equilibrium if no player can reduce the incurred penalty (or increase the incurred reward) by unilaterally deviating from the chosen strategy. If each f_i is convex on the set X_i , then the set of strategies x^* is in equilibrium if and only if $\langle \nabla_i f_i(x_i^*), (x_i - x_i^*) \rangle \geq 0$. In other words, x^* needs to be a solution of the VI $\langle F(x^*), (x - x^*) \rangle \geq 0$, where $F(x) = (\nabla f_1(x), \dots, \nabla f_m(x))$.

Complementarity problems provide the foundation for a number of Nash equilibrium algorithms. The class of complementarity problems can also be reduced to solving a VI. When the feasible set K is a cone, meaning that if $x \in K$, then $\alpha x \in K, \alpha \geq 0$, then the VI becomes a CP.

Definition 61. Given a cone $K \subset \mathbb{R}^n$ and mapping $F : K \rightarrow \mathbb{R}^n$, the complementarity problem $CP(F, K)$ is to find an $x \in K$ such that $F(x) \in K^*$, the dual cone to K , and $\langle F(x), x \rangle \geq 0$.⁵

The nonlinear complementarity problem (NCP) is to find $x^* \in \mathbb{R}_+^n$ (the non-negative orthant) such that $F(x^*) \geq 0$ and $\langle F(x^*), x^* \rangle = 0$. The solution to an NCP and the corresponding $VI(F, \mathbb{R}_+^n)$ are the same, showing that NCPs reduce to VIs. In an NCP, whenever the mapping function F is affine, that is $F(x) = Mx + b$, where M is an $n \times n$ matrix, the corresponding NCP is called a linear complementarity problem (LCP) (Murty, 1988).

8.4 Causal Network Economics

We now describe how to model causal inference in a network economics problem, which will be useful in illustrating the abstract definitions from the previous section. The model in Figure 5 is drawn from (Nagurney, 1999; Nagurney and Wolf, 2014). which were deterministic, and included no analysis of causal interventions. This network economics model comprises of three tiers of agents: producer agents, who want to sell their goods, transport agents who ship merchandise from producers, and demand market agents interested in purchasing the products or services. The model applies both to electronic goods, such as video streaming, as well as physical goods, such as face masks and other PPEs. Note that the design of such an economic network requires specifying the information fields for every producer, transporter and consumer. For the sake of brevity, we assume that the definition of these information fields are implicit in the equations defined below, but a fuller discussion of this topic will be studied in a subsequent paper.

The model assumes m service providers, n network providers, and o demand markets. Each firm's utility function is defined in terms of the nonnegative service quantity (Q), quality (q), and price (π) delivered from service provider i by network provider j to consumer k . Production costs, demand functions, delivery costs, and delivery opportunity costs are designated by f, ρ, c , and oc respectively. Service provider i attempts to maximize its utility function $U_i^1(Q, q, \pi^*)$

⁵Given a cone K , the dual cone K^* is defined as $K^* = \{y \in \mathbb{R}^n | \langle y, x \rangle \geq 0, \forall x \in K\}$.

by adjusting Q_{ijk} (eqn. 11a). Likewise, network provider j attempts to maximize its utility function $U_j^2(Q^*, q, \pi)$ by adjusting q_{ijk} and π_{ijk} (eqn. 11b).

$$U_i^1(Q, q^*, \pi^*) = \sum_{j=1}^n \sum_{k=1}^o \hat{\rho}_{ijk}(Q, q^*) Q_{ijk} - \hat{f}_i(Q) - \sum_{j=1}^n \sum_{k=1}^o \pi_{ijk}^* Q_{ijk}, \quad Q_{ijk} \geq 0 \quad (11a)$$

$$U_j^2(Q^*, q, \pi) = \sum_{i=1}^m \sum_{k=1}^o \pi_{ijk} Q_{ijk}^* - \sum_{i=1}^m \sum_{k=1}^o (c_{ijk}(Q^*, q) + oc_{ijk}(\pi_{ijk})), \quad q_{ijk}, \pi_{ijk} \geq 0 \quad (11b)$$

We assume the governing equilibrium is Cournot-Bertrand-Nash and the utility functions are all concave and fully differentiable. This establishes the equivalence between the equilibrium state we are searching for and the variational inequality to be solved where the F mapping is a vector consisting of the negative gradients of the utility functions for each firm. Since F is essentially a concatenation of gradients arising from multiple independent, conflicting objective functions, it does not correspond to the gradient of any single objective function.

$$\langle F(X^*), X - X^* \rangle \geq 0, \forall X \in \mathcal{K}, \quad (12a)$$

where $X = (Q, q, \pi) \in \mathbb{R}^{3mno+}$

and $F(X) = (F_{ijk}^1(X), F_{ijk}^2(X), F_{ijk}^3(X))$

$$F_{ijk}^1(X) = \frac{\partial f_i(Q)}{\partial Q_{ijk}} + \pi_{ijk} - \rho_{ijk} - \sum_{h=1}^n \sum_{l=1}^o \frac{\partial \rho_{ihl}(Q, q)}{\partial Q_{ijk}} \times Q_{ihl} \quad (12b)$$

$$F_{ijk}^2(X) = \sum_{h=1}^m \sum_{l=1}^o \frac{\partial c_{hjl}(Q, q)}{\partial q_{ijk}} \quad (12c)$$

$$F_{ijk}^3(X) = -Q_{ijk} + \frac{\partial oc_{ijk}(\pi_{ijk})}{\partial \pi_{ijk}} \quad (12d)$$

The variational inequality in Equations 12a represents the result of combining the utility functions of each firm into standard form. F_{ijk}^1 is derived by taking the negative gradient of U_i^1 with respect to Q_{ijk} . F_{ijk}^2 is derived by taking the negative gradient of U_j^2 with respect to q_{ijk} . And F_{ijk}^3 is derived by taking the negative gradient of U_j^2 with respect to π_{ijk} .

8.4.1 Numerical Example

We extend the simplified numerical example in (Nagurney and Wolf, 2014) by adding stochasticity to illustrate our causal variational formalism. Let us assume that there are two service providers, one transport agent, and two demand markets. Define the production cost functions:

$$f_1(Q) = q_{111}^2 + Q_{111} + \eta_{f_1}, \quad f_2(Q) = 2Q_{111}^2 + Q_{211} + \eta_{f_2}$$

where η_{f_1}, η_{f_2} are random variables indicating errors in the model. Similarly, define the demand price functions as:

$$\begin{aligned} \rho_{111}(Q, q) &= -Q_{111} - 0.5Q_{211} + 0.5q_{111} + 100 + \eta_{\rho_{111}} \\ \rho_{211}(Q, q) &= -Q_{211} - 0.5Q_{111} + 0.5q_{211} + 200 + \eta_{\rho_{211}} \end{aligned}$$

Finally, define the transportation cost functions as:

$$\begin{aligned} c_{111}(Q, q) &= 0.5(q_{111} - 20)^2 + \eta_{c_{111}} \\ c_{211}(Q, q) &= 0.5(q_{211} - 10)^2 + \eta_{c_{211}} \end{aligned}$$

and the opportunity cost functions as:

$$oc_{111}(\pi_{111}) = \pi_{111}^2 + \eta_{oc_{111}}, oc_{211}(\pi_{211}) = \pi_{211}^2 + \eta_{oc_{211}}$$

Using the above equations, we can easily compute the component mappings F_i as follows:

$$\begin{aligned} F_{111}^1(X) &= 4Q_{111} + 0.5Q_{211} - 0.5q_{111} - 99 \\ F_{211}^1(X) &= 6Q_{211} + \pi_{211} - 0.5Q_{111} - 0.5q_{211} - 199 \\ F_{111}^2(X) &= q_{111} - 20, F_{211}^2(X) = q_{211} - 10 \\ F_{111}^3(X) &= -Q_{111} + 2\pi_{111}, F_{211}^3(X) = -Q_{211} + 2\pi_{211} \end{aligned}$$

For simplicity, we have not indicated the noise terms above, but assume each component mapping F_i has an extra noise term η_i . It is also clear that we can now give precise semantics to causal intervention in this system, following the principles laid out in (Pearl, 2009). For example, if we set the network service cost q_{111} of network provider 1 serving the content producer 1 to destination market 1 to 0, then the production cost function under the intervention distribution is given by

$$E_{q_{111}=0}(f_1(Q)|\hat{q}_{111}) = Q_{111} + E_{q_{111}=0}(\eta_{f_1}|\hat{q}_{111})$$

Finally, the Jacobian matrix associated with $F(X)$ is given by the partial derivatives of each F_i mapping with respect to $(Q_{111}, Q_{211}, q_{111}, q_{211}, \pi_{111}, \pi_{211})$ is given as:

$$-\nabla U(Q, q, \pi) = \begin{pmatrix} 4 & .5 & -.5 & 0 & 1 & 0 \\ 0.5 & 6 & 0 & -.5 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 2 & 0 \\ 0 & -1 & 0 & 0 & 0 & 2 \end{pmatrix}$$

Note this Jacobian is non-symmetric, but positive definite, as it is diagonally dominant. Hence the induced vector field F can be shown to be strongly monotone, and the induced VI has exactly one solution.

8.5 Causal VI Algorithms

We now discuss algorithms for solving causal VI's. There are a wealth of existing methods for deterministic VI's (Facchinei and Pang, 2003; Nagurney, 1999), which can be adapted to solving causal VI's. The simplest method for solving a causal VI is the well-known projection algorithm (Facchinei and Pang, 2003):

$$x_{k+1} = \Pi_K[x_k - \alpha_k F_w(x_k)]$$

where F_w is the vector field induced by some causal intervention, which can be viewed as a modification of the classical projection method for deterministic VI's. The algorithm follows the direction of the negative vector field at a point x_k , and if the iterate falls outside the feasible space K , it projects back into K . If F_w is strongly monotone, and Lipschitz, and the learning rate α_k is suitably designed, then the projection algorithm is guaranteed to find the solution to a causal VI.

Understanding the convergence of the projection method will give us insight into how to analyze causal interventions in VI's. At the heart of convergence analysis of any VI method is bounding the iterates of the algorithm. In the below derivation, x^* represents the final solution to a causal VI, and x_{k+1}, x_k are successive iterates:

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|P_K[x_k - \alpha_k F_w(x_k)] - P_K[x^* - \alpha_k F_w(x^*)]\|^2 \\ &\leq \|(x_k - \alpha_k F_w(x_k)) - (x^* - \alpha_k F_w(x^*))\|^2 \\ &= \|(x_k - x^*) - \alpha_k (F_w(x_k) - F_w(x^*))\|^2 \\ &= \|x_k - x^*\|^2 - 2\alpha_k \langle (F_w(x_k) - F_w(x^*)), x_k - x^* \rangle \\ &\quad + \alpha_k^2 \|F_w(x_k) - F_w(x^*)\|^2 \\ &\leq (1 - 2\mu\alpha_k + \alpha_k^2 L^2) \|x_k - x^*\|^2 \end{aligned}$$

Here, the first inequality follows from the nonexpansive property of projections, and the last inequality follows from strong monotonicity and Lipschitz property of the F_w mapping. Bounding the term $\langle (F_w(x_k) - F_w(x^*)), x_k - x^* \rangle$ is central to the design of any VI method. As we show in the next section, in modeling causal interventions a similar term will arise, except under different mappings, representing the "untreated" and "treated" cases.

Korpelevich (1977) extended the projection algorithm with the well-known "extragradient" method, which requires two projections, but is able to solve VI's for which the mapping F is only monotone. If projections are expensive, particularly in large network economy models, these algorithms may be less attractive than incremental stochastic projection methods, which we turn to next.

8.6 Incremental Projection Methods

We now describe an incremental two-step projection method for solving causal VI's, based on work by Wang and Bertsekas (2015). Their algorithm adapted to causal VI's can be written as follows:

$$z_k = x_k - \alpha_k F_w(x_k, v_k), \quad x_{k+1} = z_k - \beta_k(z_k - P_{w_k} z_k) \quad (13)$$

where $\{v_k\}$ and $\{w_k\}$ are sequences of random variables, generated by sampling the causal VI model, and $\{\alpha_k\}$ and $\{\beta_k\}$ are sequences of positive scalar step sizes. Note that an interesting feature of this algorithm is that the sequence of iterates x_k is not guaranteed to remain within the feasible space K at each iterate. Indeed, P_{w_k} represents the projection onto a randomly sampled constraint w_k .

The analysis of convergence of this algorithm is somewhat intricate, and we refer the reader to (Mahadevan, 2021) for more details. It can be shown that two-step stochastic algorithm given in Equation 13 converges to the solution of a causal VI, namely:

Theorem 12. *Given the category of \mathcal{C}_{CVI} causal VIs, the solution associated with any decision object representing a finite-dimensional causal variational inequality problem $\mathcal{M} = CVI(F, K)$, and a causal intervention, defined by the sub-object $\mathcal{M}_w = CVI(F_w, K)$, where $F_w(x) = E_w[F(x, \eta|\hat{w})]$, where \hat{w} denotes the intervention of setting of variable w to a specific non-random value, and where $E_w[\cdot]$ now denotes expectation with respect to the intervention probability distribution P_w , the two-step algorithm given by Equation 13 produces a sequence of iterates x_k that converges almost surely to x^* , where*

$$\langle F_w(x^*), (y - x^*) \rangle \geq 0, \quad \forall y \in K$$

Proof: The proof of this theorem is given in (Mahadevan, 2021), and largely follows the derivation given in (Wang and Bertsekas, 2015), where the only difference is that in a causal VI problem, we are conditioning the stochastic VI on the intervention distribution P_w . \square

8.7 Decomposition of Causal VI UDMs

Finally, we discuss the issue of how to exploit structure in solving complex UDMs, specifically for the type of network economics problems described above. We also discuss "sensitivity analysis" of UDMs, in particular, how to measure the effect of some intervention, by comparing potential outcomes across the "treated" units with the "untreated" units (Imbens and Rubin, 2015). We characterize treatment effects in causal variational inequalities under interventions, building on the existing results on sensitivity analysis of classical variational inequalities (Nagurney, 1999).

We can apply the more general machinery of finite space topologies introduced above, but for simplicity, we focus on the case when a UDM object is defined by a cartesian product operation over the set of feasible solutions, and the monotone operator F decomposes additively over these individual subsets.

Definition 62. *A partitioned CVI is defined as the causal variational inequality problem of finding a vector $x^* = (x_1^*, \dots, x_n^*) \in K \subset \mathbb{R}^n$ such that*

$$\langle E_w[F(x, \eta|\hat{w})], (x - y) \rangle \geq 0, \quad \forall y \in K$$

where the function F is partitionable function of order m , meaning that

$$\langle E_w[F(x, \eta|\hat{w})], (x - y) \rangle = \sum_{i=1}^m \langle E_w[F_i(x, \eta|\hat{w})], (x_i - y_i) \rangle$$

where each $F_i : K_i \subset \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{n_i}$, with each K_i being a convex domain such that $\prod_i K_i = K$.

We can further simplify the solution of causal VIs by doing sensitivity analysis of a UDM subject to an intervention. The use of interventions to probe a structure is very common in many engineering domains, as well as more abstractly in many areas of math.

Theorem 13. *If Y is probabilistically causally irrelevant to X , given Z , then $CVI(F_{\hat{y}, \hat{z}}(x), K)$ has the same solution as $CVI(F_{\hat{y}', \hat{z}}(x), K)$.*

Proof: The proof is straightforward given the axioms of causal irrelevance (Pearl, 2009). If Y is causally irrelevant to X given Z , then it follows that $P(x|\hat{y}, \hat{z}) = P(x|\hat{y}', \hat{z})$ for all y, y', x, z , namely if \hat{z} is fixed, then changing the value of y has no influence on the distribution of x . In this case, the mapping F under the two intervention distributions remains identical. \square

Now we examine the case when interventions do alter the solution to a causal VI, where our goal is to measure the change in solution in terms of properties of the "untreated mapping F_0 and the "treated" mapping F_1 .

Theorem 14. *Let the solution of the original “untreated” CVI(F_0, K) be denoted by x_0 , where F_0 is assumed to be strongly monotone, and (stochastically) Lipschitz, with μ being the coefficient in the strong monotonicity property. Given a causal intervention, the “treated” CVI(F_1, K) results in the modified solution vector x_1 . Then it follows that*

$$\|x_1 - x_0\| \leq \frac{1}{\mu} \|F_1(x_1) - F_0(x_1)\| \quad (14)$$

Proof: Since x_0 and x_1 solve the “untreated” and “treated” causal VI’s, respectively, it must follow that:

$$\begin{aligned} \langle F_0(x_0), y - x_0 \rangle &\geq 0, \quad \forall y \in K \\ \langle F_1(x_1), y - x_1 \rangle &\geq 0, \quad \forall y \in K \end{aligned}$$

Substituting $y = x_1$ in the first equation above, and $y = x_0$ in the second equation, it follows that:

$$\langle (F_1(x_1) - F_0(x_0)), x_1 - x_0 \rangle \leq 0$$

Equivalently, we get

$$\langle (F_1(x_1) - F_0(x_0) + F_0(x_1) - F_0(x_1)), x_1 - x_0 \rangle \leq 0$$

Using the monotonicity property of F_0 , we get:

$$\begin{aligned} \langle (F_1(x_1) - F_0(x_1)), x_0 - x_1 \rangle &\geq \langle (F_0(x_0) - F_0(x_1)), x_0 - x_1 \rangle \\ &\geq \mu \|x_0 - x_1\|^2 \end{aligned}$$

from which the theorem follows immediately \square

Interestingly, in the above analysis, we did not assume any property of the intervened causal VI F_1 , other than it has a solution (meaning that F_1 should be continuous). The following corollaries follow directly from Theorem 14.

Theorem 15. *Given the original “untreated” causal VI CVI(F_0, K), where F_0 is strictly monotone, and the intervened “treated” causal VI CVI(F_1, K), where the intervened mapping F_1 is continuous, but not necessarily monotone, if x_0 and x_1 denote the solutions to the original “untreated” and causally intervened CVI, where $x_0 \neq x_1$, then it follows that:*

$$\langle (F_1(x_1) - F_0(x_1)), x_1 - x_0 \rangle < 0 \quad (15)$$

$$\langle (F_1(x_1) - F_0(x_0)), x_1 - x_0 \rangle \leq 0 \quad (16)$$

Here, we are bounding the causal intervention effect of $F_1 - F_0$ of the “treated” vs. “untreated” operator, whereas previously in the convergence analysis of Equation 13, we were trying to bound the same operator’s effect on two different parameter values. The following theorem extends Theorem 15 in showing that for partitionable CVI’s, the effects induced by local causal interventions can be isolated.

Theorem 16. *If a partitioned causal VI CVI(F, K) is defined, where each component F_i is a strongly monotone partitionable function, F_i^1 denotes the causally intervened component function, and F_i^0 is the “untreated” function, x^0 denotes the solution to the original “untreated” CVI(F_0, K) and x^1 denotes the solution to the “treated” causally intervened CVI(F_1, K) defined by the manipulated F_i^1 component functions, then*

$$\sum_{i=1}^m \langle (F_i^1(x_i^1) - F_i^0(x_i^1)), x_i^1 - x_i^0 \rangle < 0 \quad (17)$$

Proof: The proof follows readily from Theorem 14, Theorem 15, and Definition 62. In particular, if x^1 is the solution of the intervened CVI(F_1, K) and x^0 is the solution of the original CVI(F_0, K), it follows that:

$$\langle F_1(x^1) - F_0(x^1), x^1 - x^0 \rangle = \sum_{i=1}^m \langle F_i^1(x_i^1) - F_i^0(x_i^1), x_i^1 - x_i^0 \rangle$$

Since the component functions F_i are strongly monotone, the overall function F is as well, and by applying Theorem 14, it follows that:

$$\langle (F^1(x^1) - F^0(x^1)), x^1 - x^0 \rangle < 0$$

which immediately yields that

$$\sum_{i=1}^m \langle (F_i^1(x_i^1) - F_i^0(x_i^1)), x_i^1 - x_i^0 \rangle < 0 \quad \square$$

If only a single component function F_i is treated, then:

$$\langle (F_i^1(x_i^1) - F_i^0(x_i^1)), x_i^1 - x_i^0 \rangle < 0$$

We can use these insights into designing an improved version of the two-step stochastic approximation algorithm given by Equation 13. Instead of selecting random iterates to project on, we can instead prioritize those components F_i^1 that have been modified by the intervention.

9 Summary and Future Work

In this paper we proposed the Universal Decision Model (UDM) framework, building on the core concept of information fields, suitably generalized to the formalism of category theory. We showed how information fields defined by decision objects in a UDM are associated with a finite topology, which can be exploited to facilitate hierarchical decomposition, as well as build homotopically invariant representations. We described a specific UDM category of causal variational inequalities, and showed how it can be used to solve causal inference problems in real-world complex network games. We identified several universal properties, including information integration, decision solvability, and hierarchical abstraction. Information integration is the process of consolidating data from heterogeneous sources, and its categorical foundation is built on forming products or limits in a category. Abstraction enables simulating complex decision process by simpler processes through bisimulation morphisms, and its categorical foundation rests on forming quotients, co-products and co-limits. Finally, solvability refers to the requirement that a decision problem must have a unique solution defined by a fixed point equation, and it requires an order-preserving morphism across objects. Much remains to be done in this research paradigm on universal decision making. We summarize a few topics for further research that extend the current paper.

9.1 Presheaf Representations

We briefly described the Yoneda lemma, which specified how to construct universal representations for any object $c \in \mathcal{C}$ in a category based on covariant or contravariant functors. We can apply this approach to construct particular pre-sheaf $\mathcal{C}(-, c)$ representations of objects in the UDM category of MDPs, POMDPs, PSRS, and more generally intrinsic models. A detailed study of presheaf UDM representations is an important topic for future research.

9.2 Giri Monads

In category theory, the usual way to model probability distributions is through *monads*, a topic we did not get into as it would take us far afield into category theory. Briefly, a Giri monad is the canonical monad structure for the category of all measurable spaces. A **monoidal category** \mathcal{C} is a category that has a bifunctor $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$, along with an identity mapping, and several natural isomorphisms that define the associativity of the tensor product. More formally, a monoid (M, μ, η) in a monoidal category \mathcal{C} is an object M in \mathcal{C} together with two morphisms (obeying the standard associativity and identity properties) that make use of the category's monoidal structure: the associative binary operator $\mu : M \otimes M \rightarrow M$, and the identity $\eta : I \rightarrow M$. A **monad** is often termed a "monoid in the category of endofunctors", namely functors that map a category into itself. That is, consider the category of endofunctors whose objects are endofunctors and whose morphisms are natural transformations between them. This can be shown to define a monoidal category.

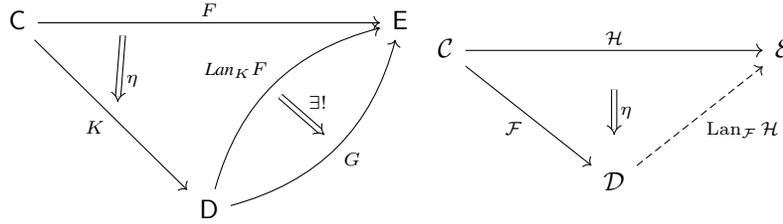
To link monads to probability distributions, recall that a measurable space (X, \mathcal{F}) is a set X equipped with a σ -algebra \mathcal{F} . Recall also that a measure $\nu : X \rightarrow \mathbb{R}$ is a particular kind of set function from the σ -algebra to nonnegative real numbers. A measurable space completed with a measure (X, \mathcal{F}, ν) is called a measure space, and a measurable space completed with a probability measure is called a probability space. We have already previously defined measurable functions. We can now define the category **Meas** of measurable spaces, where the morphisms are simply the measurable mappings between them. For any specific measurable space M , we can define the space of all possible probability measures that could be placed on it as $\Xi(M)$. Note that $\Xi(M)$ is itself a space of measures - that is, a space in which the points themselves are probability measures. As a probability measure, any element of $\Xi(M)$ is a function from measurable subsets of M to the interval $[0, 1]$ in \mathbb{R} . A key area for future work is to study UDMs defined over Giri monads.

9.3 Kan Extensions of Intrinsic Models

It is well known in category theory that ultimately every concept, from products and co-products, limits and co-limits, and ultimately even the Yoneda embeddings, can be derived as special cases of the Kan extension (Riehl, 2016). Another topic for future work is to apply this powerful technique in the analysis of intrinsic models. We briefly define the concept of Kan extensions below. Kan extensions intuitively are a way to approximate a functor \mathcal{F} so that its domain can be extended from \mathcal{C} to \mathcal{D} . In other words, Kan extensions are a way of taking two functors and constructing a third functor to make a diagram commute. Because it may be impossible to make commutativity work in general, Kan extensions rely on natural transformations to make the extension be the best possible approximation to \mathcal{F} along \mathcal{K} .

Definition 63. A **left Kan extension** of a functor $H : \mathcal{C} \rightarrow \mathcal{E}$ along F , another functor $F : \mathcal{C} \rightarrow \mathcal{D}$, is a functor $\text{Lan}_F H : \mathcal{D} \rightarrow \mathcal{E}$ with a natural transformation $\eta : H \rightarrow \text{Lan}_F H \circ F$ such that for any other such pair $(G : \mathcal{D} \rightarrow \mathcal{E}, \gamma : H \rightarrow GK)$, γ factors uniquely through η . In other words, there is a unique natural transformation

$\alpha : Lan_F \Rightarrow G.$



A key challenge for future work is to explore approximation of functors from the UDM category to other categories using Kan extensions.

9.4 Applications

We introduced the Universal Decision Model (UDM), a broad overarching framework for decision making that integrates a number of well-studied modalities, including causal inference, decentralized stochastic control and reinforcement learning, and multiplayer games in network economics. The UDM model uses category theory to construct a universe of decision making objects, which are related by bisimulation morphisms. The information field representation defines the knowledge available to each decision maker, and induces a finite topology on the space of agents. The topology of subsystems allows hierarchical decomposition of complex networks of agents, and we showed how homotopically equivalent systems can be formulated using algebraic topology. The next step is to articulate how specific applications can be solved in this paradigm, and design effective algorithms for this purpose. Applications naturally would require making concrete choices on the particular types of problem classes involved (e.g., team, classical, sequential etc.), and particular types of temporal ordering (e.g., linear, partial), and subsystem design using specific information fields.

References

- David M. Sobel, Joshua B. Tenenbaum, and Alison Gopnik. Children’s causal inferences from indirect evidence: Backwards blocking and bayesian reasoning in preschoolers. *Cogn. Sci.*, 28(3):303–333, 2004. doi:10.1016/j.cogsci.2003.11.001. URL <https://doi.org/10.1016/j.cogsci.2003.11.001>.
- Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.
- Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, USA, 2015. ISBN 0521885884.
- Michael Maschler, Eilon Solan, and Shmuel Zamir. *Game Theory*. Cambridge University Press, 2013. doi:10.1017/CBO9780511794216.
- Y. Shoham and K. Leyton-Brown. *Multiagent systems: algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.
- Stuart J. Russell and Devika Subramanian. Provably bounded-optimal agents. *J. Artif. Intell. Res.*, 2:575–609, 1995. doi:10.1613/jair.133. URL <https://doi.org/10.1613/jair.133>.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL <https://www.worldcat.org/oclc/37293240>.
- Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020. ISBN 9780134610993. URL <http://aima.cs.berkeley.edu/>.
- Brendan Fong. *The algebra of open and interconnected systems*, 2016.
- Sen Lin, Zhi Zhou, Zhaofeng Zhang, Xu Chen, and Junshan Zhang. *Edge Intelligence in the Making: Optimization, Deep Learning, and Applications*. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers, 2020. doi:10.2200/S01054ED1V01Y202009LNA025. URL <https://doi.org/10.2200/S01054ED1V01Y202009LNA025>.
- Thomas Seeley. *Honeybee Democracy*. Princeton University Press, 2011.
- H. S. Witsenhausen. The intrinsic model for discrete stochastic control: Some open problems. In A. Bensoussan and J. L. Lions, editors, *Control Theory, Numerical Methods and Computer Systems Modelling*, pages 322–335, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg. ISBN 978-3-642-46317-4.

- N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.
- A. Nagurney. *Network Economics: A Variational Inequality Approach*. Kluwer Academic Press, 1999.
- Hans S. Witsenhausen. A standard form for sequential stochastic control. *Math. Syst. Theory*, 7(1):5–11, 1973. doi:10.1007/BF01824800. URL <https://doi.org/10.1007/BF01824800>.
- Hans S. Witsenhausen. On information structures, feedback and causality. *SIAM Journal of Control*, 9(2):149–160, 1971a.
- Hans S. Witsenhausen. On the relations between the value of a game and its information structure. *Inf. Control.*, 19(3):204–215, 1971b. doi:10.1016/S0019-9958(71)90085-4. URL [https://doi.org/10.1016/S0019-9958\(71\)90085-4](https://doi.org/10.1016/S0019-9958(71)90085-4).
- Hans S. Witsenhausen. Equivalent stochastic control problems. *Math. Control. Signals Syst.*, 1(1):3–11, 1988. doi:10.1007/BF02551232. URL <https://doi.org/10.1007/BF02551232>.
- Pierre Carpentier, Guy Cohen, Jean-Philippe Chancelier, and Michel De Lara. *Stochastic Multi-Stage Optimization*, volume 75 of *Probability Theory and Stochastic Modelling*. Springer, 2015. doi:10.1007/978-3-319-18138-7. URL <https://hal.archives-ouvertes.fr/hal-01165572>.
- Pulkit Grover. Information structures, the witsenhausen counterexample, and communicating using actions. In John Baillieul and Tariq Samad, editors, *Encyclopedia of Systems and Control*. Springer, 2015. doi:10.1007/978-1-4471-5102-9_148-1. URL https://doi.org/10.1007/978-1-4471-5102-9_148-1.
- Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis. Optimal control strategies in delayed sharing information structures. *IEEE Trans. Autom. Control.*, 56(7):1606–1620, 2011. doi:10.1109/TAC.2010.2089381. URL <https://doi.org/10.1109/TAC.2010.2089381>.
- Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Trans. Autom. Control.*, 58(7):1644–1658, 2013. doi:10.1109/TAC.2013.2239000. URL <https://doi.org/10.1109/TAC.2013.2239000>.
- Ashutosh Nayyar and Demosthenis Teneketzis. Common knowledge and sequential team problems. *IEEE Trans. Autom. Control.*, 64(12):5108–5115, 2019. doi:10.1109/TAC.2019.2912536. URL <https://doi.org/10.1109/TAC.2019.2912536>.
- Ashutosh Nayyar and Tamer Basar. Dynamic stochastic games with asymmetric information. In *Proceedings of the 51th IEEE Conference on Decision and Control, CDC 2012, December 10-13, 2012, Maui, HI, USA*, pages 7145–7150. IEEE, 2012. doi:10.1109/CDC.2012.6426857. URL <https://doi.org/10.1109/CDC.2012.6426857>.
- Benjamin Heymann, Michel De Lara, and Jean-Philippe Chancelier. Causal inference theory with information dependency models. *CoRR*, abs/2108.03099, 2021. URL <https://arxiv.org/abs/2108.03099>.
- Emily Riehl. *Category Theory in Context*. Dover, 2016.
- Michael A. Arbib and Ernest G. Manes. Basic concepts of category theory applicable to computation and control. In Ernest G. Manes, editor, *Category Theory Applied to Computation and Control, Proceedings of the First International Symposium, San Francisco, CA, USA, February 25-26, 1974, Proceedings*, volume 25 of *Lecture Notes in Computer Science*, pages 1–34. Springer, 1974. doi:10.1007/3-540-07142-3_59. URL https://doi.org/10.1007/3-540-07142-3_59.
- André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation and open maps. In *Proceedings of the Eighth Annual Symposium on Logic in Computer Science (LICS '93), Montreal, Canada, June 19-23, 1993*, pages 418–427. IEEE Computer Society, 1993. doi:10.1109/LICS.1993.287566. URL <https://doi.org/10.1109/LICS.1993.287566>.
- Balaraman Ravindran and Andrew G. Barto. SMDP homomorphisms: An algebraic approach to abstraction in semi-markov decision processes. In Georg Gottlob and Toby Walsh, editors, *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 1011–1018. Morgan Kaufmann, 2003. URL <http://ijcai.org/Proceedings/03/Papers/145.pdf>.
- Thomas L. Dean and Robert Givan. Model minimization in markov decision processes. In Benjamin Kuipers and Bonnie L. Webber, editors, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA*, pages 106–111. AAAI Press / The MIT Press, 1997. URL <http://www.aaai.org/Library/AAAI/1997/aaai97-017.php>.
- Vishal Soni and Satinder P. Singh. Abstraction in predictive state representations. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 639–644. AAAI Press, 2007. URL <http://www.aaai.org/Library/AAAI/2007/aaai07-101.php>.

- Robert J. Aumann. Agreeing to Disagree. *The Annals of Statistics*, 4(6):1236–1239, 1976. doi:10.1214/aos/1176343654. URL <https://doi.org/10.1214/aos/1176343654>.
- Robert J. Aumann. Borel structures for function spaces. *Illinois Journal of Mathematics*, 5(4):614–630, 1961. doi:10.1215/ijm/1255631584. URL <https://doi.org/10.1215/ijm/1255631584>.
- Paul R. Halmos. *Measure theory* / [by] Paul R. Halmos. Springer-Verlag New York, 1974. ISBN 0387900888. URL <http://www.loc.gov/catdir/enhancements/fy0814/74010690-t.html>.
- Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. ISBN 978-0-47161977-2. doi:10.1002/9780470316887. URL <https://doi.org/10.1002/9780470316887>.
- Peter Van Overschee and Bart De Moor. Subspace algorithms for the stochastic identification problem. *Autom.*, 29(3):649–660, 1993. doi:10.1016/0005-1098(93)90061-W. URL [https://doi.org/10.1016/0005-1098\(93\)90061-W](https://doi.org/10.1016/0005-1098(93)90061-W).
- Juris Hartmanis and Richard Edwin Stearns. Some dangers in state reduction of sequential machines. *Inf. Control.*, 5(3):252–260, 1962. doi:10.1016/S0019-9958(62)90588-0. URL [https://doi.org/10.1016/S0019-9958\(62\)90588-0](https://doi.org/10.1016/S0019-9958(62)90588-0).
- Tai-Danae Bradley, Tyler Bryson, and John Terilla. *Topology A Categorical Approach*. MIT Press, 2020.
- Robert Goldblatt. *Topoi: The Categorical Analysis of Logic*. Dover, 2006.
- Kim Guldstrand Larsen and Arne Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94(1):1–28, 1991. doi:10.1016/0890-5401(91)90030-6. URL [https://doi.org/10.1016/0890-5401\(91\)90030-6](https://doi.org/10.1016/0890-5401(91)90030-6).
- André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. *Inf. Comput.*, 127(2):164–185, 1996. doi:10.1006/inco.1996.0057. URL <https://doi.org/10.1006/inco.1996.0057>.
- Anna Nagurny and Tilman Wolf. A cournot–nash–bertrand game theory model of a service-oriented internet with price and quality competition among network transport providers. *Computational Management Science*, 11(4):475–502, 2014.
- Michael R. Thon and Herbert Jaeger. Links between multiplicity automata, observable operator models and predictive state representations: a unified learning framework. *J. Mach. Learn. Res.*, 16:103–147, 2015. URL <http://dl.acm.org/citation.cfm?id=2789276>.
- Jonathan A. Barmak. *Algebraic topology of finite topological spaces and applications*. Lecture notes in mathematics <Berlin>. Springer, Heidelberg ; Berlin u.a., 2011. URL <http://deposit.d-nb.de/cgi-bin/dokserv?id=3826587&prov=M&dok%5Fvar=1&dok%5Fext=htm>.
- P. S. Alexandroff. *Diskrete Räume*. *Rec. Math. [Mat. Sbornik] N.S.*, 2:501–518, 1937.
- James R. Munkres. *Elements of algebraic topology*. Addison-Wesley, 1984. ISBN 978-0-201-04586-4.
- Patrick Forre and Joris M. Mooij. Markov properties for graphical models with cycles and latent variables, 2017.
- R. E. Stong. Finite topological spaces. *Trans. Amer. Math. Soc.*, 123:325–340, 1966.
- Alex Fix and Stephen Partias. Enumeration of homotopy classes of finite t0 topological spaces. <http://math.uchicago.edu/~may/FINITE/REUPapers/Fix.pdf>, 2008.
- Dimitri P. Bertsekas. *Dynamic programming and optimal control, 3rd Edition*. Athena Scientific, 2005. ISBN 1886529264. URL <https://www.worldcat.org/oclc/314894080>.
- Sridhar Mahadevan. Causal inference in network economics. *CoRR*, abs/2109.11344, 2021. URL <https://arxiv.org/abs/2109.11344>.
- F. Facchinei and J. Pang. *Finite-Dimensional Variational Inequalities and Complimentarity Problems*. Springer, 2003.
- K. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, 1988.
- G. Korpelevich. The extragradient method for finding saddle points and other problems. *Matekon*, 13:35–49, 1977.
- Mengdi Wang and Dimitri P. Bertsekas. Incremental constraint projection methods for variational inequalities. *Math. Program.*, 150(2):321–363, 2015. doi:10.1007/s10107-014-0769-x. URL <https://doi.org/10.1007/s10107-014-0769-x>.