Causal Learning With Occam's Razor

Oliver Schulte[a,*]

[a]School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada

Abstract

Occam's razor directs us to adopt the simplest hypothesis consistent with the evidence. Learning theory provides a precise definition of the inductive simplicity of a hypothesis for a given learning problem. This definition specifies a learning method that implements an inductive version of Occam's razor. As a case study, we apply Occam's inductive razor to causal learning. We consider two causal learning problems: learning a causal graph structure that presents global causal connections among a set of domain variables, and learning context-sensitive causal relationships that hold not globally, but only relative to a context. For causal graph learning, Occam's inductive razor directs us to adopt the model that explains the observed correlations with a minimum number of direct causal connections. For expanding a causal graph structure to include context-sensitive relationships, Occam's inductive razor directs us to adopt the expansion that explains the observed correlations with a minimum number of free parameters. The paper provides a gentle introduction to the learning-theoretic definition of inductive simplicity and the derivation of Occam's razor.

1. Introduction

An inductive version of Occam's razor directs us to adopt the simplest hypothesis consistent with the evidence. The question is how to define simplicity. In this paper we describe a learning-theoretic topological concept of inductive simplicity. As a case study, we apply the concept to an important and challenging inductive problem: inferring causal relationships from observed correlations among variables. Causal graphs are a widely used model class for representing such relationships (also known as Bayesian networks). We show that the inductive simplicity rank of a causal graph is measured exactly by the number of edges in the graph: The smaller this number is, the greater is the model's inductive simplicity. Thus the disconnected graph is simplest, and the fully connected graph the most complex. Occam's razor for learning causal graph therefore directs us to adopt a graph that explains the observed correlations with a minimum number of direct causal links.

As a second case study in causal learning, we examine context-sensitive

[*]Corresponding author

causal relationships, that hold only in a given context. For example, different subpopulations may exhibit different causal connections. A common approach to learning context-sensitive causal relationships is to first learn a causal graph that describes general causal relationships, then expand the causal graph with a model of context-sensitive relationships. Context-sensitive causal relationships can be represented in graphs whose edges are labelled with specific values of variables. These graphical representations can be converted to probabilistic logical rules. Therefore a method for learning context-sensitive relationships can also be used to learn probabilistic logical rules that represent statistical patterns in the data. We show that the inductive simplicity rank of a context-sensitive causal model is measured exactly by the number of free parameters in the model: The smaller this number is, the greater is the model's inductive simplicity. In this case, the learning-theoretic concept agrees with widely used statistical model selection scores (e.g., BIC and AIC [6]), which also use the number of free parameters.

The paper presents a gentle introduction to the learning-theoretic definitions and theorems that lead to an inductive version of Occam's razor. We begin with the concept of a learning problem. A learning problem has three components: i) A space of alternative hypotheses, ii) a set of possible evidence items that learning uses to decide among different hypotheses, and iii) a definition of which hypotheses are consistent with which evidence items. We discuss in detail how causal learning can be framed as a learning problem in this sense.

We present a general definition of Occam's razor for a given learning problem with a finite space of alternative hypotheses. Finite hypothesis spaces are sufficient for our causal learning case study, and the finiteness assumption simplies technical definitions and theorems. Learning theorists have developed the concept of inductive simplicity more generally for infinite hypothesis spaces. In a finite hypothesis space, inductive simplicity is defined in terms of the branching structure of the hypothesis space as follows. The inductive simplicity rank of a hypothesis H is the length of the longest chain of nested hypotheses, starting with the hypothesis H. One hypothesis H_1 is nested within another H_2 if all evidence items that are consistent with H_1 are also consistent with H_2, but not vice versa. This simplicity concept has several noteworthy features.

1. Although in practice a hypothesis is described within a hypothesis language, its inductive simplicity ranking is entirely a function of its observational content. Inductive simplicity therefore is completely *independent of the choice of language* for describing hypotheses.
2. Inductive simplicity depends on the learning context; it is *relative to the entire hypothesis space*. A hypothesis does not have an intrinsic inductive simplicity rank, but only a rank relative to alternatives under consideration. If the set of alternative under consideration changes, the inductive complexity of a hypothesis can change as well [31].
3. A precise definition of inductive simplicity leads to a precise definition of Occam's razor for inductive inference. The inductive version of Occam's razor that corresponds to the topological definition can be rigorously jus-

> tified in terms of learning performance: We show that Occam's inductive razor is the only ~~learner~~ that achieves reliable, steady, and fast convergence to a correct hypothesis ~~is Occam's razor.~~

This theorem provides a justification of Occam's razor in terms of learning performance that is independent of whether the inferences underwritten by Occam's inductive razor are intuitively plausible. Nonetheless, it is interesting to ask whether in cases of interest, Occam's inductive razor directs us towards plausible inferences. The answer is generally yes: We have already described the results for causal learning. Other case studies include Goodman's New Riddle of Induction, where Occam's inductive razor selects "all emeralds are green" over "all emeralds are grue" [33], and inferring conservation laws in particle physics, where the Occam method rediscovers the laws in the important Standard Model of particle physics [34].

Paper Organization. We introduce the concept of a learning problem using an abstract toy example. Then we discuss how causal graph learning can be framed as a learning problem. The toy example serves to introduce performance criteria for a learning method. We give the general definition of inductive simplicity for a finite hypothesis space, then prove the relationship between inductive simplicity and optimal learning performance. Causal learning illustrates these results in a concrete setting. We first apply the general theory to the problem of learning causal graph structures, then to the problem of learning context-senstive causal relationships.

2. Definition of Learning Problems

We first introduce general concepts from learning theory and present some general results concerning inductive simplicity. These concepts have been used with different nomenclature depending on the intended application [15, 16, 25]. We present a framework that is as simple as possible, while expressive enough to model causal learning. Our nomenclature is intended to be appropriate for discussing learning in general (rather than, e.g., language learning problems, in particular). After introducing the general concepts, we present in detail a model for applying them to causal graph learning.

Learning begins with observations. The first part of a learning problem is therefore a set of **evidence items**. This set is part of the specification of the learning problem. There is no assumption that the evidence items are in some sense "basic data", e.g., sensor readings. Evidence items may result from extensive post-processing of sensor readings, or may be aggregates of sensor readings. In the terminology of [1], an evidence item may be a lower-level datum, or a higher-level phenomenon. For learning-theoretic analysis, a sequence of evidence items is the basis for drawing inductive conclusions; more formally, a sequence of evidence items is the input to a learning algorithm.

Examples. In modelling high-energy physics, an evidence item may be a single reaction that physicists report as having observed in a particle accelerator [32]. In cognitive psychology, an evidence item may be a reaction by an experimental subject to an input stimulus that is to be explained by a model of cognitive architecture [13]. In inductive problems often discussed in philosophy, an evidence item may be the color of a swan, the color of an emerald, or the observation of a sunrise on a given day. In language learning, an evidence item may be a single sentence heard by a child [14].

The second part of a learning problem is a space of **hypotheses**, possible explanations/models of the evidence. This space represents the background knowledge on which learning is based.

Examples. In high-energy physics, a set of conservation laws defines which particle reactions are possible. In cognitive architectures, a specification of cognitive models and their interconnections can explain behavior. In language learning, a grammar defines a set of well-formed sentences.

The third part of a learning problem is a specification of which evidence items are **consistent** with which hypotheses. This specifies the prediction made by a hypothesis, or its empirical content.

Examples. In high-energy physics, a reaction is consistent with a set of conservation laws if it conserves all quantities posit~~i~~ed by the conservation laws. A cognitive computational architecture is consistent with a subject's reaction to an input stimulus if the reaction is the same as the output computed by the hypothesized system. In language learning, a sentence is consistent with a grammar if it can be generated by the grammar.

In sum, we have the following definition of a learning problem.

Definition 1. *A learning problem consists of the following three components.*

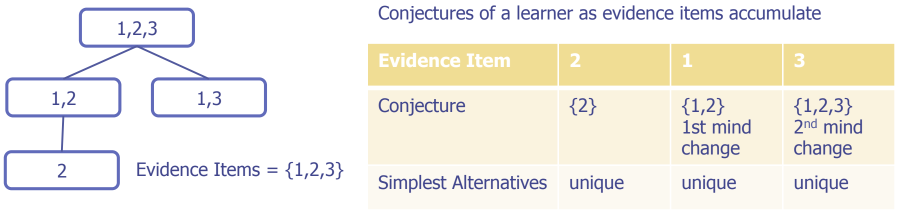
Evidence Items *A countable set of evidence items $\mathbb{E}$.*

Hypotheses *A finite set of hypotheses $\mathcal{H}$.*

Consistency *A consistency relation between a hypothesis and an evidence item that specifies which hypotheses are consistent with which evidence items. Thus the consistency relation is a subset of the Cartesian product $\mathbb{E} \times \mathcal{H}$.*

This definition is equivalent to the concept of a language learning problem used in formal learning theory [15]. Learning-theoretic analysis applies to infinite hypothesis sets as well as finite ones. We assume finiteness only to simplify the definitions. We will indicate below how the results can be generalized to infinite hypothesis spaces. Another generalization allows consistency to be a matter of degree, as is often the case with statistical hypotheses [19].

Global Underdetermination. Some issues with defining a hypothesis space arise when different hypotheses are **empirically equivalent**, meaning that they are consistent with exactly the same evidence items. This occurs in many practical hypothesis spaces, because hypotheses are described using a hypothesis language, and just like natural language, a hypothesis language typically allows us to express the same content in different ways. For example one set of conservation laws is empirically equivalent to another if they both span the same linear subspace [32]. We discuss empirical equivalence for causal graphs in Section 3.3



Conjectures of a learner as evidence items accumulate

Evidence Item	2	1	3
Conjecture	{2}	{1,2} 1st mind change	{1,2,3} 2nd mind change
Simplest Alternatives	unique	unique	unique

Figure 1: Left: A set of abstract evidence items (three) and a simple abstract hypothesis space. A hypothesis is represented as the set of evidence items that are consistent with it. Links between hypotheses correspond to set inclusion. Right: A learning sequence for this learning problem. Three evidence items are presented in sequence (2,1,3). After each new evidence items, the conjecture of the learner is shown.

below. In philosophical discussions of inductive inference, this phenomenon is often described as global underdetermination [16]. In statistical terminology, it is referred to as the identifiability problem, where even an infinitely large sample does not entail a uniquely correct model and/or parameter values for the model. Global underdetermination raises the problem of defining which hypothesis is the correct one for a complete set of observations when there are empirically equivalent alternatives. A simple approach to global underdetermination is to have learners output an equivalence class of hypotheses. In effect, this changes the hypothesis space such that a hypothesis in the new space is an equivalence class of hypotheses in the original space. A benefit of focusing on empirical equivalence classes is that in many problems, determining which hypotheses are empirically equivalent is a problem of independent interest. In the following we assume that learners output equivalence classes of hypotheses. As we show in the causal graph example, an equivalence class of original hypotheses can be often be described compactly.

3. The Causal Graph Structure Learning Problem

We present a view of causal graph learning as a learning problem in the sense of Definition 1, following [18, 36]. In this paper, we use the term “causal graph” as essentially synonymous with “Bayesian network”. The approaches for learning Bayesian networks are similar to those for causal graphs. The main difference is in the interpretation: causal graphs are viewed as representing the effects of actions or interventions. For further discussion see [37, 29].

3.1. Hypothesis Space

A **causal graph structure** is a directed acyclic graph (DAG). The hypothesis space is the set of causal graphs that share a common fixed set of nodes $\mathbf{V} = \{X_1, X_2, \ldots, X_n\}$. The nodes are also called its **variables**. Every node X has a possible **domain of values**. In this paper we consider only discrete variables with finite domains. We write $X = x$ for an assignment of value to variable X, and use boldface vector notation such as $\mathbf{X} = \mathbf{x}$ for an assignment of values to a set of variables. Figure 2 shows a causal graph from [29, p.15].

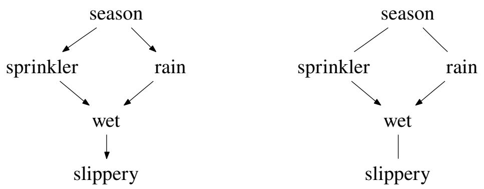


Figure 2: The sprinkler network and its pattern. Sprinkler and Rain share Season as a common cause, and Wetness as a common effect. The wetness of the pavement is a cause of its being slippery.

3.2. Evidence Items

An evidence item is a **conditional dependence assertion** of the form

$$\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{Z} \tag{1}$$

where **X**, **Y**, and **Z** are disjoint sets of variables. Intuitively, a dependence assertion can be read as "the variables in the set **X** are relevant to the variables in the set **Y**, given an assignment of values to the variables in the set **Z**."

Examples. The most common interpretation of a dependence assertion is as expressing a *probabilistic dependence.* Probabilistic dependencies are defined with respect to a **joint distribution** over the nodes in the graph. A joint distribution specifies a probability

$$P(\mathbf{V} = \mathbf{v})$$

for each complete assignment of values to the variables. From a joint distribution we obtain marginal distributions over any subset **X** of variables

$$P(\mathbf{Y} = \mathbf{y}) \equiv \sum_{\mathbf{x}} P(\mathbf{Y} = \mathbf{y}, \mathbf{X} = \mathbf{x})$$

where **X** is the set of variables $\mathbf{V} - \mathbf{Y}$ other than **Y**.

A joint distribution also specifies conditional distributions via the definition

$$P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) \equiv \frac{P(\mathbf{Y} = \mathbf{y}, \mathbf{X} = \mathbf{x})}{P(\mathbf{X} = \mathbf{x})}$$

where **X** and **Y** are disjoint, and $P(\mathbf{X} = \mathbf{x}) > 0$. In what follows we assume that all joint probabilities are positive so that conditional probabilities are well defined. For a discussion of learning causal graphs with 0 joint probabilities, which may occur with deterministic causal relationships, please see [22].

The meaning of a conditional dependence assertion can be defined in terms of a **probabilistic inequality** as follows:

$$\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{Z} \equiv \exists \mathbf{x}, \mathbf{y}, \mathbf{z}. P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}) \neq P(\mathbf{Y} = \mathbf{y} | \mathbf{Z} = \mathbf{z}). \tag{2}$$

For example, in the graph of Figure 2, a joint distribution may specify a probability of 0.1 that it is summer, the sprinker is on, and all other variables are simultaneously true:

$$P(\texttt{season} = \textit{summer}, \texttt{sprinkler} = \textit{on}, \texttt{rain} = T, \texttt{wet} = T, \texttt{slippery} = T) = \textit{0.1}.$$

The joint distribution may entail the claim that the probability that the pavement is wet is affected by the probability that the season is summer, even given that the sprinkler is on:

$$P(\texttt{wet} = T|\texttt{season} = \textit{summer}, \texttt{sprinkler} = \textit{on}) \neq P(\texttt{wet} = T|\texttt{sprinkler} = \textit{on})$$

which witnesses the dependence assertion that

$$\texttt{wet} \not\perp\!\!\!\perp \texttt{season}|\texttt{sprinkler}.$$

3.2.1. Discussion

A dependence assertion existentially quantifies over possible values of variables: it asserts that there exist specific values $\mathbf{x}, \mathbf{y}, \mathbf{z}$ such that if the conditioning variable set $\mathbf{Z}$ takes on the values $\mathbf{z}$, then the values $\mathbf{x}$ for $\mathbf{X}$ ~~are do not~~ affect the probability that $\mathbf{Y}$ takes on value $\mathbf{y}$. This existentially quantified assertion can be derived from basic unquantified probabilistic inequalities of the form

$$P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}) \neq P(\mathbf{Y} = \mathbf{y}|\mathbf{Z} = \mathbf{z}) \tag{3}$$

for fixed values $\mathbf{x}, \mathbf{y}, \mathbf{z}$. Such basic probabilistic inequalities can be ascertained using statistical tests. A statistical test can be applied to a given data sample to decide whether there is a statistically significant difference in two (conditional) distribution over the same set of variables ($\mathbf{Y}$) in the equation above~~)~~ [37], [39, Sec.4]. Dependence assertions illustrate therefore the point of Section 2 that evidence items can be higher-level aggregates of the original data: First ~~a~~ statistical tests are applied to discover basic probabilistic inequalities for specific values, of the form (3). Second, these inequalities are aggregated into an existence assertion that constrains only the relationships between *variables* of the form (2), not between specific values. In Section 7 below we consider learning from probabilistic inequalities for specific values.

It is also possible to take as evidence items *in*dependence assertions among variables, which are just the negations of probabilistic dependence assertions among variables. Many methods for learning causal graphs are based on observed independencies rather than dependencies [37, 38, 5, 39]. Schulte *et al.* provide a learning-theoretic analysis with independence assertions as evidence items [36]. One reason for taking dependencies rather than independencies as learning input ~~is motivated~~ is that many statisticians hold that no independence conclusion should be drawn when a statistical significance test fails to reject an independence hypothesis [12].

Many BN learning systems follow the “search and score” paradigm, seeking a structure that optimizes some numeric scoring function [9]. Our learning model follows the alternative constraint-based paradigm; see [9], [27, Ch.10]. The term reflects the fact that observed dependencies are treated as constraints that a learned model must satisfy.

3.3. Consistency of a Causal Graph With Dependency Statements.

To define consistency with dependence assertions for causal graphs, we need to review the semantics of causal graphs, that is, what a graph entails about observations. A causal graph defines a joint distribution over its nodes if it is extended with **conditional probability parameters**. These parameters specify the probability of each child node value, given each assignment of values to its parents. A joint distribution can be derived from the local conditional probabilities via the product formula: multiply together all the conditional probabilities defined by each child-parent value assignment. For instance, for the graph in Figure 2(left), the joint probability above would defined by the product

$$\begin{aligned} & P(\texttt{season} = \mathit{summer}, \texttt{sprinkler} = \mathit{on}, \texttt{rain} = T, \texttt{wet} = T, \texttt{slippery} = T) \\ = \; & P(\texttt{season} = \mathit{summer}) \times P(\texttt{sprinkler} = \mathit{on} | \texttt{season} = \mathit{summer}) \times P(\texttt{rain} = T | \texttt{season} = \cancel{T}) \\ \times \; & P(\texttt{wet} = T | \texttt{sprinkler} = \mathit{on}, \texttt{rain} = T) \times P(\texttt{slippery} = T | \texttt{wet} = T) \end{aligned}$$

where the conditional probabilities that appear in this expression are specified as parameter values for the causal graph. Since the combination of (graph structure + parameters) defines a joint distribution P, we can take such a combination to be consistent with a set of observed dependencies if the dependencies are entailed by the joint distribution P. Figure 3 illustrates the logic of this definition. A causal graph structure G (without parameter values set) is then **consistent** with a set of dependence assertions of the form (1) if *there exists* a parameter assignment for the graph G that is consistent with the observed dependencies.

It may appear that determining whether a graph structure G is consistent with a given set of dependencies is difficult because searching through the set of all possible parameter assignments is difficult. However, causal graph theory has developed a graph-based criterion, known as *d-separation*, that facilitates an efficient check whether a graph structure is consistent with given dependencies based on the links only, without reference to parameter values. For readers unfamiliar with this criterion, we provide a review in the appendix. In terms of d-separation, a causal graph structure is consistent with observed dependencies if it is an I-map of the given dependencies. This means that if any node set $\mathbf{X}$ is d-separated from another $\mathbf{Y}$ by the nodes $\mathbf{Z}$, then the dependence assertion $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$ is not included in the given dependencies.

The d-separation criterion also makes it possible to provide a graphical characterization of causal graphs that are empirically equivalent, that is, that are consistent with exactly the same dependence assertions. Say that two nodes X, Y are **adjacent** in a graph G if G contains an edge $X \to Y$ or $Y \to X$.

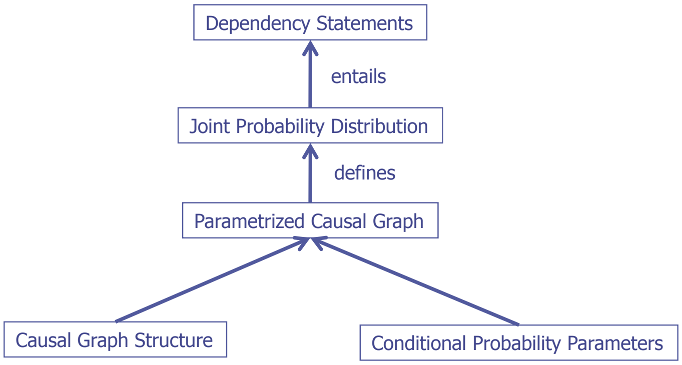

Dependency Statements
entails
Joint Probability Distribution
defines
Parametrized Causal Graph
Causal Graph Structure
Conditional Probability Parameters

Figure 3: Consistency of Causal Graph Structure with Observed Dependency Statements. A parametrized causal graph is consistent with observed dependencies if its joint distribution entails the dependencies. A causal graph structure is consistent with observed dependencies if there is a parametrization of the structure that is consistent with the dependencies.

The **pattern** of DAG G is the partially directed graph that has the same adjacencies as G, and contains an arrowhead $X \to Y$ if and only if G contains a triple $X \to Y \leftarrow Z$ where X and Z are not adjacent. Figure 2 illustrates the concept. Verma and Pearl proved that two graphs G_1 and G_2 are consistent with the same dependence assertions if and only if they lead to the same pattern [40, Thm. 1]). Thus we can use a pattern as a syntactic representation of an empirical equivalence class of graphs.

As usual in a learning problem, we assume that a complete infinite data stream enumerates exactly those dependencies that are consistent with a correct causal graph structure. The assumption that the observed dependencies can be represented exactly by a causal graph is known as *faithfulness* in causal graph theory. For discussions of the faithfulness assumption, see [29, Ch.2.4], [41], [38, Ch.8.1].

As the causal graph example shows, defining the three components of a learning problem for a realistic scenario can be a substantial task. Once this is accomplished, we can apply powerful results from general learning theory to determine optimal learning algorithms for the learning problem. We review some of these general results and apply them to causal graph structure learning in the next section.

4. Performance Criteria and Optimal Learning

Intuitively, a learner takes as input a sequence of evidence items—called the **data sequence**, and produces as output a member of the hypothesis space—called the **conjecture**. It is often convenient to slightly generalize this learning model: First, we allow the data sequence to contain the special nonevidence symbol $\{\#\}$ to model pauses in data presentation. Second, we allow the learner to output $\{?\}$, where ? corresponds to the vacuous conjecture "no guess". We generically denote learners by upper-case Greek letters such as Ψ, Φ.

Figure 1 illustrates these concepts. There are three evidence sequences,

$$(2), (2, 1), (2, 1, 3).$$

The learner maps these to a sequence of conjectures

$$\{2\}, \{1, 2\}, \{1, 2, 3\}.$$

Each conjecture is consistent with exactly the observed items.

A **data stream** is an *infinite* sequence of evidence items (or #). The content of a data stream is the set of evidence items that appear in the sequence. A hypothesis is **correct** for a data stream the evidence items in the data stream comprise all and only those consistent with the hypothesis. A data stream is **possible** for a learning problem if some hypothesis is correct for it. In other words, we assume that every possible complete sequence of evidence items can be explained by some hypothesis under consideration. ~~i~~t is however possible that for a finite evidence sequence, no hypothesis is consistent with all and only the evidence items observed. For instance in the toy problem above, if the first evidence sequence is (1), there is no hypothesis in the space is consistent with item 1 and only item 1.

4.1. Performance Criteria for Learning

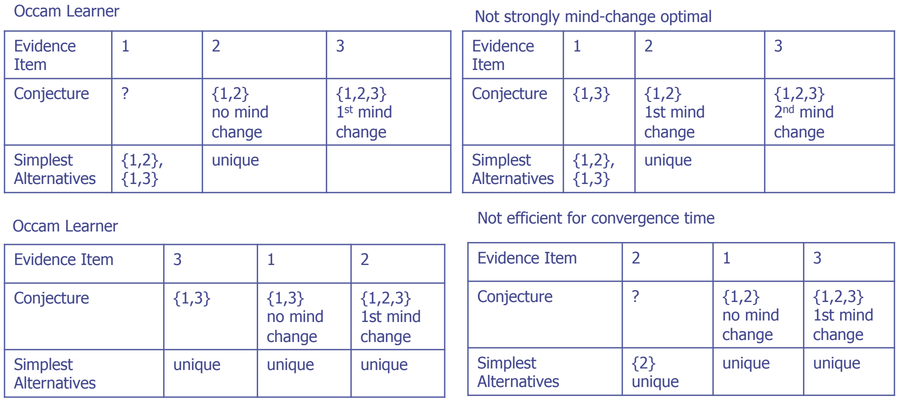
Learning theorists have studied a number of performance criteria for succesful learning (often referred to as identification criteria [4]). In this paper we consider three criteria: i) reliable identification of a correct hypothesis in the limit, ii) fast identification of a correct hypothesis~~:~~ minimizing time to convergence, and iii) steady identification of a correct hypothesis~~:~~ minimizing hypothesis changes.

We say that a learner **identifies** a correct hypothesis on an infinite data stream if after some finite time, the learner outputs a hypothesis that is correct for the entire data stream. Identification requires induction in the sense of going beyond the data: although the learner ususally has not observed all evidence items that may appear, it has to conjecture which items may appear in the future based on which have appeared in the past. A learner **reliably identifies** the correct hypothesis in a learning problem if it identifies a correct on every possible data stream. To illustrate these concepts in the example of Figure 1, consider the infinite data stream

$$2, 1, 3, \#, \#, \ldots.$$

The set of evidence items for this data stream is $\{1, 2, 3\}$, so the learner converges to a correct hypothesis after three evidence items have been observed. Figure 4 provides more examples of evidence and conjecture sequences.

~~Identifiability~~ requires only eventual convergence. We can compare the performance of different learners with respect to convergence speed by using the decision-theoretic criterion of weak dominance: A learner Ψ is **faster than** a learner Φ on a data stream if Ψ converges to a hypothesis before Φ does. A learner Ψ is **uniformly faster** than Φ if Ψ is faster than Φ on some possible



Occam Learner

Evidence Item	1	2	3
Conjecture	?	{1,2} no mind change	{1,2,3} 1st mind change
Simplest Alternatives	{1,2}, {1,3}	unique	

Not strongly mind-change optimal

Evidence Item	1	2	3
Conjecture	{1,3}	{1,2} 1st mind change	{1,2,3} 2nd mind change
Simplest Alternatives	{1,2}, {1,3}	unique	

Occam Learner

Evidence Item	3	1	2
Conjecture	{1,3}	{1,3} no mind change	{1,2,3} 1st mind change
Simplest Alternatives	unique	unique	unique

Not efficient for convergence time

Evidence Item	2	1	3
Conjecture	?	{1,2} no mind change	{1,2,3} 1st mind change
Simplest Alternatives	{2} unique	unique	unique

Figure 4: Examples of different learning methods. After each evidence item is received, a learner outputs a conjecture.

data stream, and Ψ is at least as fast as Φ on every possible data stream. A learner is **convergence-time efficient** if it is not uniformly slower than another learner. Time efficiency may be restricted to a class of learners, for example we may consider learners that are not uniformly slower than another learner that reliably identifies a correct hypothesis.

Our final criterion is steady convergence: minimizing the number of times that a learner changes its conjecture before convergence. A learner Ψ **changes its mind** at some nonempty finite sequence of evidence items $\langle e_1, \ldots, e_m, e_{m+1}\rangle$ if the conjecture of Ψ after observing evidence items $\langle e_1, \ldots, e_m\rangle$ is not vacuous (i.e., is not ?) and differs from the conjecture of Ψ after observing evidence items $\langle e_1, \ldots, e_m, e_{m+1}\rangle$[15, Ch. 12.2][30, 16]. For any possible data stream, we can count the number of mind changes that a learner undergoes on that data stream. This is a finite number assuming that the learner eventually settles on a correct hypothesis.

We can assess the performance of a learner with respect to mind changes by using the decision-theoretic minimax criterion, which considers worst-case performance. Say that a learner Ψ reliably identifies a correct hypothesis with at most k mind changes if it is reliable and changes its conjectures at most k times on every possible data stream. In the toy problem of Figure 1, a learner can reliably identify a correct hypothesis with at most two mind changes. This is illustrated by the Occam learner shown in Figure 1 and in the left two examples of Figure 4. The Occam learner outputs a hypothesis that accommodates a minimum number of evidence items, if there is a unique such hypothesis. Otherwise the learner outputs ?.

A global mind change bound is not quite good enough in many problems, because a learner may fail to take advantage of a lucky evidence sequence that makes it possible to learn with fewer mind changes than in the worst case. The top half of Figure 4 illustrates this possibility. After evidence item 1 is observed, it is possible to succeed with at most one further mind change: wait with ? until further evidence decides between the hypotheses $\{1,2\}$ or $\{1,3\}$.

Then at most one more mind change is required if the correct hypothesis turns out to be $\{1, 2, 3\}$. The learner in the top right box conjectures $\{1, 3\}$ right away and undergoes two mind changes in case the item 2 is observed before 3. Since the problem requires two mind changes in the worst case (see Figure 1), this behavior is consistent with a global mind change bound of two. We therefore refine the mind change criterion as follows [23]. Say that a learning problem can be solved with at most k mind changes starting with evidence sequence $\langle e_1, \ldots, e_m \rangle$ if for every data stream extending the evidence sequence, there is a learner that reliably identifies a correct hypothesis, and uses at most k mind changes, where mind changes are counted starting at time m. That is, mind changes are counted with the conjecture for $\langle e_1, \ldots, e_m \rangle$ as the initial conjecture. A learner Ψ is **strongly mind change optimal** if for every finite evidence sequence $\langle e_1, \ldots, e_m \rangle$, the learner Ψ solves the learning with the best possible mind change bound, when the count starts at $\langle e_1, \ldots, e_m \rangle$.

4.2. Optimal Learning

Performance criteria can be used to select learning methods. We can picture a performance criterion as a filter that is applied to learning methods: applying multiple performance criteria is like applying successive filters to learning. The criteria we examine in the rest of the paper are as follows. First, filter out all methods that do not reliably identify a correct hypothesis. Second, among these, filter out those that are not strongly mind change optimal. Third, eliminate among the remaining ones those methods that are uniformly slower than another remaining method. We refer to a method that meets these criteria simply as **optimal**. In decision-theoretic terms, the successive filters correspond to a lexicographic ordering of performance criteria: identifiability first, mind change optimality second, convergence time third. It is possible to combine performance criteria in different ways [33, 17], to arrive at different concepts of optimality. The lexicographic ordering is surprisingly powerful: we will show next that in many problems, including those with finite hypothesis spaces, there is only one optimal method, the Occam learner. This optimal method guides us towards interesting and plausible inductive inferences in many domains. Determining the conjectures of the optimal method is an investigation that leads to substantive insights into the methodological structure of a learning problem. We outline a general result that assists a theorist in determining the conjecture of the optimal method. Then we show how to apply this to causal graph search.

5. Optimal Learning and Inductive Simplicity

Our goals in this section are to prove the uniqueness of an optimal learner and to ~~analyze the~~ characterize optimal inferences in terms of the structure of the learning problem. This structure can be described in terms of a topological ranking of hypotheses in a given hypothesis space [24].

Definition 2. *Let $\mathcal{H}$ be a hypothesis space and H be a ~~language~~ in $\mathcal{H}$. We write $H \subset H'$ to denote that the evidence items consistent with hypothesis H*

*are a proper subset of those consistent with H'. An **inclusion chain** of length k starting with H is a sequence of the form*

$$H \subset H_1 \subset \cdots \subset H_i \subset \cdots \subset H_k$$

*where each hypothesis in the chain is contained in the hypothesis space $\mathcal{H}$. The **inclusion depth** of H is the maximum length of an inclusion chain starting with H.*

Kevin Kelly has developed the view that inclusion depth, or closely related concepts, can be viewed as a simplicity ranking of hypotheses [17]. Accordingly, we define the **inductive simplicity rank** of a hypothesis H in a learning problem as the inclusion depth of H. Occam's razor directs us to adopt a maximally simple hypothesis that is consistent with the evidence. For a learning problem, this leads to the following definition of the **Occam learner**

$$\Psi_{occam}(\langle e_1, \ldots, e_m \rangle) = \begin{cases} ? & \text{if there is no uniquely simplest hypothesis consistent with } \langle e_1, \ldots, e_m \rangle \\ H & \text{if } H \text{ is the uniquely simplest hypothesis consistent with } \langle e_1, \ldots, e_m \rangle. \end{cases}$$

where $\Psi_{occam}(\langle e_1, \ldots, e_m \rangle)$ is the output of the Occam learner after receiving the evidence sequence $\langle e_1, \ldots, e_m \rangle$. This definition assumes that there are no empirically equivalent hypotheses; otherwise it should be modified so that the Occam learner conjectures the maximally simple equivalence class if there is one.

The next proposition shows a strong connection between inductive simplicity and required mind changes: The worst case number of mind changes required is exactly the maximum of the inductive simplicity ranks in the hypothesis space.

Proposition 3. *For any learning problem with a finite hypothesis space, there is a learner that reliably identifies a correct hypothesis with at most k mind changes $\iff$ the maximum inductive simplicity rank of any hypothesis is k.*

Proof Outline. ($\Leftarrow$) Suppose that the maximum inductive simplicity rank of any hypothesis is k. Consider a mind change by the Occam learner Ψ_{occam} on a sequence of evidence items $\langle e_1, \ldots, e_m, e_{m+1} \rangle$. Let H be the conjecture of Ψ_{occam} on the previous sequence $\langle e_1, \ldots, e_m$. Since a mind change occurred at stage $m + 1$, the hypothesis H is not vacuous. By the definition of the Occam learner, H is therefore the uniquely most simple consistent with the evidence $\langle e_1, \ldots, e_m$. Thus any other hypothesis consistent with the further evidence $\langle e_1, \ldots, e_m, e_{m+1} \rangle$ must have lower simplicity rank than H. Therefore any time that the Occam learner changes its mind, the maximum simplicity rank of the remaining hypotheses decreases by at least 1. Since the maximum rank over all is k, there can be at most k mind changes by the Occam learner.

($\Rightarrow$) Let Ψ be an arbitrary learner. Suppose that there exists an inclusion chain

$$H \subset H_1 \subset \cdots \subset H_i \subset \cdots \subset H_k.$$

There is a possible data stream that enumerates all and only evidence items that are consistent with H. On this data stream, the learner Ψ must conjecture H at some finite stage m to converge to the correct hypothesis. At this point, there is a data stream that extends the finite evidence sequence and enumerates all and only evidence items that are consistent with H_1. Again the learner Ψ must conjecture H_1 at some finite stage m_1 to converge to the correct hypothesis. This leads to at least one mind change by the learner. Repeating this argument, we can extend the data streams after the mind change occurs consistent with hypotheses $H_2, \ldots, H_i, \ldots, H_k$, in such a way that a mind change occurs for each hypothesis in the chain. The result is a data stream on which the learner Ψ requires at least k mind changes. Since Ψ is an arbitrary learner, the construction shows that in the worst case, every learner requires at least k mind changes. ■

A formal proof is provided by Luo and Schulte [24]. This result can be extended to infinite hypothesis spaces using transfinite mind change bounds and a topological generalization of the concept of simplicity rank. The concept of inductive simplicity developed applies therefore in a wide class of problems. The main restriction is that learning with a mind change bound requires that some hypothesis be conclusively verifiable, in the sense that there is some evidence that is consistent only with that hypothesis. Kelly presents a generalization of mind change optimality that relaxes this assumption [17].

Proposition 3 characterizes the inductive complexity of an entire learning problem. The next proposition concerns the properties of optimal learners. The main result is that the Occam learner is the *only* learner that achieves reliable, steady, and fast convergence.

Proposition 4. *For a finite hypothesis space, the Occam learner is the only learner that reliably identifies a correct hypothesis, is strongly mind change optimal, and convergence-time efficient.*

Proof Outline. Any reliable learner is strongly mind change optimal if and only if whenever it produces a nonvacuous conjecture, the conjecture is the uniquely simplest consistent with the evidence. For otherwise the learner may incur one more mind change than necessary given the evidence, as illustrated in the top right box of Figure 4. What distinguishes the Occam learner from other strongly mind change optimal learners is that the Occam learner does not wait: it immediately conjectures the uniquely simplest hypothesis as soon as there is one, whereas other strongly mind change optimal may output ? instead. It is easy to see that the Occam learner is uniformly faster than all such learners: Since the output ? is not correct for any data stream, whenever a non-Occam learner outputs ? on an evidence sequence $\langle e_1, \ldots, e_m \rangle$, its convergence time is later than stage m. The Occam learner by contrast converges to the uniquely simplest hypothesis H by time m on any data stream extending the evidence $\langle e_1, \ldots, e_m \rangle$ for which H is correct. Therefore the Occam learner possibly converges sooner than the mind-change optimal non-Occam learner, and never slower. ■

Schulte and Luo provide a formal proof that includes the general case of infinite hypothesis spaces. Implementing the Occam learner requires determining the simplicity rank of a hypothesis. Next we describe how to determine the simplicity rank of causal graphs.

6. Inductive Simplicity for Learning Causal Graphs

In this section we characterize the inductive simplicity rank of a causal graph and the Occam learner for causal graphs. Figure 5 illustrates an inclusion chain of causal graph patterns. A fundamental result in causal graph theory characterizes the inclusion relation between two causal graphs $G_1 \subset G_2$, meaning that G_2 is consistent with all dependence assertions that are consistent with G_1. The Meek-Chickering theorem shows that the inclusion holds just in case G_2 can be transformed into G_1 by a sequence of two types of transformations: (i) deleting edges, and (ii) reversing a covered arc [26],[7, Thm.4]. For precise definitions, please see [23]. The Meek-Chickering theorem is the basis for the following characterization of inductive simplicity for causal graphs.

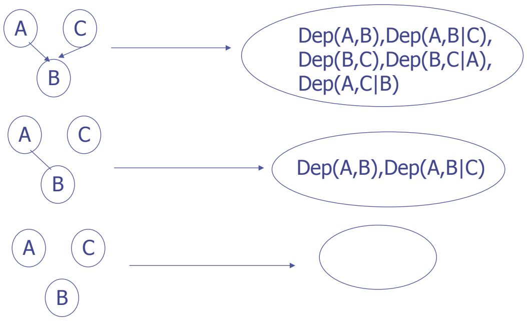


Figure 5: Left: An inclusion chain among causal graph patterns. The dependencies of the bottom pattern are included in those for the middle pattern which are included in those for the top patterns. Right: Representative dependence assertions for each pattern. In figures we use the notation $Dep(A,B|C)$ for $A \not\perp\!\!\!\perp |C$.

Proposition 5. *The inductive simplicity rank of a causal graph or pattern containing edges* $\mathbf{E}$ *is* $|\mathbf{V}| - |\mathbf{E}|$*, the number of edges that are* not *included in the graph.*

The formal proof can be found in [36]. So the simplest graph is the empty one, and the most complex graph is the complete one that contains all possible adjacencies. It is not surprising that the inductive complexity of a graph increases with the number of edges in the graph. The surprising aspect of the proposition is that this is all that matters: the direction of the adjacencies does not affect the simplicity rank.

The Occam learner for causal graph learning is therefore defined by

$$\Psi_{occam}(\mathcal{D}) = \begin{cases} ? & \text{if there is no uniquely simplest pattern consistent with the dependencies } \mathcal{D} \\ G & \text{if } G \text{ is the uniquely simplest pattern consistent with } \mathcal{D}. \end{cases}$$

where $\mathcal{D}$ is a list of observed dependencies. Figure 6 illustrates some of the conjectures of the Occam learner. Schulte and Luo [36] provide a more elaborate example for a graph with four nodes.

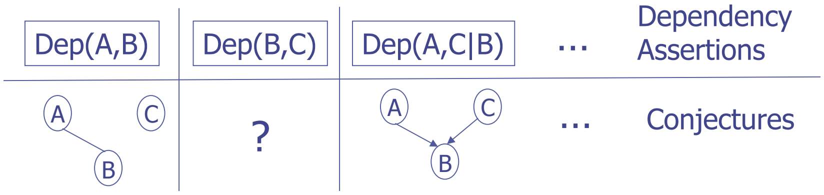


Figure 6: The conjectures of the Occam learner on a sequence of three observed dependence assertions.

As the number of variables increases, it becomes challenging to determine whether there is a uniquely optimal causal graph for a given list of observed correlations. Luo and Schulte [36, Th.23] show that computing the conjectures of an Occam learner is NP-hard. From this we can conclude that there is no algorithm that implements the Occam learner exactly in reasonable (polynomial) computation time. Researchers in causal graph learning have developed a number of heuristic search algorithms that can be seen as approximating the Occam learner [35].

So far we have considered the problem of learning causal relationships among *variables*. However, some causal relationships involve specific *values* of variables. In the next section, we examine learning with Occam's razor for such relationships.

7. The Occam Learner for Context-Sensitive Causal Relationships

For discrete variables, it is common that a causal relationship holds not in general, but only conditional on the values of some variables [2, 11, 10]. These values establish a *context* that may reveal additional causal relationships. For a simple example, there may be a causal relationship between the intelligence of a student and their grade in a course. But this relationship holds only conditional on the fact that the student has actually registered in the course. Geiger and Heckerman discuss the following example in detail [11].

> A guard of a secured building expects three types of persons to approach the building-s entrance: workers in the building, approved visitors, and spies. As a person approaches the building, the guard can note its gender and whether or not the person wears a badge. Spies are mostly men. Spies always wear badges in an attempt to fool the guard. Visitors dont wear badges because they dont have one. Female workers tend to wear badges more often than do male workers. The task of the guard is to identify the type of person approaching the building.

Figure 7 shows a causal graph for this example.

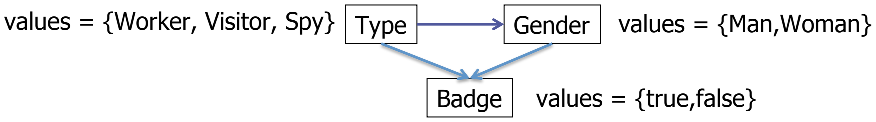


Figure 7: A causal graph for Geiger and Heckerman's security guard example. The graph is consistent with every dependence assertion.

One approach to representing context-sensitive (in)dependencies is to employ different causal graphs for different contexts, as in multinets or similarity networks. Various structures have been used to represent structure in the conditional distribution of a child node given values of its parents [21]. A widely used and especially intuitive model for discrete variables is a probability estimation tree (PET) [2, 11, 10]. A more expressive formalism replaces trees with directed acyclic graphs, known as *probability estimation diagrams* PEDs [3]. In this section we examine the Occam learner for identifying a probability estimation diagram. Our main result is that the inclusion depth complexity of a PED is given by the number of its terminal nodes (with no children). The number of terminal nodes is equivalent to the number of parameters required to specify a joint distribution.

7.1. Probability Estimation Diagrams

In what follows we consider a fixed graph structure G and a node Y in the graph. The parents of Y are denoted $X_1, \ldots, X_m$. We can think of the child node Y as a dependent or outcome variable, and as the parent nodes X_i as independent or treatment variables. An assignment of m values to each independent variable is denoted by boldface notation such as $\mathbf{x}, \mathbf{x}'$. The parameters of a causal graph specify conditional probabilities of the form

$$P(Y = y|\mathbf{X} = \mathbf{x}).$$

One possibility for specifying the required conditional probabilities is simply to enumerate them. The result is a flat table of conditional probabilities, abbreviated CPT. This approach fails to capture additional structure in the conditional distribution of the child variable given parent variable values. In particular, a flat CPT fails to represent context-sensitive independencies. A **probability estimation diagram** is a DAG D whose nonterminal nodes are labelled with the parents of Y. Each terminal node (with no children) is labelled with a probability distribution over the possible values of Y. An edge from a node labelled X to a child is labelled with a set of possible values from the domain of X. The edges originating from a node labelled X partition the domain of X, meaning that every possible value is assigned to one and only one edge. This entails that for any complete assignment of parent values $\mathbf{x}$, following the appropriate edges in the diagram leads to a unique terminal, which we denote as $terminal_D(\mathbf{x})$. Figure 8 illustrates two probability estimation trees for the security guard problem.

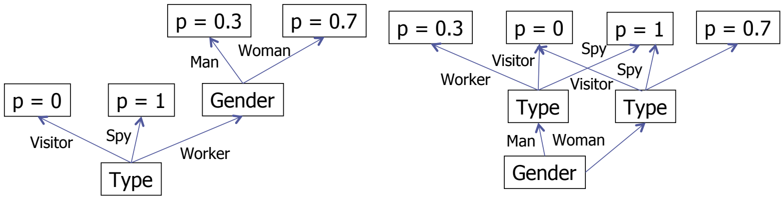


Figure 8: Two equivalent probability estimation trees for the security guard graph of Figure 7, for the child node *Badge*. We define $p = P(Badge = T)$.

7.2. The Diagram Learning Problem

The learning problem is to identify a probability estimation diagram structure that is sufficiently complex to represent the true conditional probability distribution of the child variable conditional on the parent variables. The evidence items for this learning problem are conditional inequalities of the form

$$P(Y|\mathbf{X} = \mathbf{x}) \neq P(Y|\mathbf{X} = \mathbf{x}')$$

where $\mathbf{X}$ denotes the parent variables and $\mathbf{x}$ is an assignment of values to the parents. Thus an evidence item asserts that the distribution of the child variable, conditional on one assignment of values to the parent variables, differs from the distribution conditional on another assignment of values to the parent variables.

To complete the definition of the learning problem, we need to specify which evidence items that are consistent with a diagram. A diagram D entails a conditional *equality* constraint

$$P(Y|\mathbf{X} = \mathbf{x}) = P(Y|\mathbf{X} = \mathbf{x}')$$

if $terminal_D(\mathbf{x}) = terminal_D(\mathbf{x}')$. A diagram is **consistent with** a conditional *in*equality constraint $P(Y|\mathbf{X} = \mathbf{x}) \neq P(Y|\mathbf{X} = \mathbf{x}')$ if the diagram does not entail its negation. Note that whether a diagram is consistent with an (in)equality constraint depends only on the qualitative graph structure of the diagram, not on the quantitative probability estimates in its terminals. In sum, the components of the learning problem in this section are as follows.

Hypothesis Space The set of probability estimation diagram structures for a child variable Y with parent variables $\mathbf{X}$.

Evidence Items Conditional distribution inequalities of the form

$$P(Y|\mathbf{X} = \mathbf{x}) \neq P(Y|\mathbf{X} = \mathbf{x}').$$

Consistency Relation A diagram structure ~~if~~ consistent with an inequality constraint $P(Y|\mathbf{X} = \mathbf{x}) \neq P(Y|\mathbf{X} = \mathbf{x}')$ if the assignments $\mathbf{x}, \mathbf{x}'$ are mapped to different terminals.

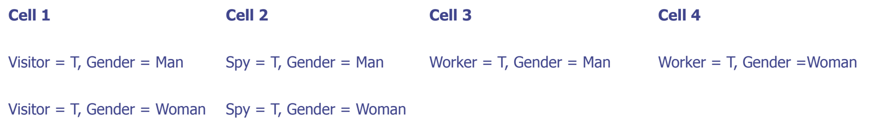
The Occam learner for this problem selects the inductively simplest diagram. In the next section we characterize the inductive simplicity rank of a probability estimation diagram.

8. The Inductive Simplicity of a Probability Estimation Diagram

Different diagram structures may be empirically equivalent in the sense of being consistent with exactly the same probabilistic inequalities. An equivalence class can be characterized by a partition of parent value assignments. More precisely, the terminals of a diagram induce a partition of the parent value assignments into equivalence classes, where two assignments are equivalent if they are mapped to the same terminal node:

$$\mathbf{x} \equiv \mathbf{x}' \iff \mathit{terminal}_D(\mathbf{x}) = \mathit{terminal}_D(\mathbf{x}'). \tag{4}$$

In general, if Equation (4) holds for an equivalence partition $\equiv$ and a diagram D, we say that the partition **corresponds** to the diagram. It is easy to see that two diagram structures are consistent with exactly the same probabilistic inequalities just in case they corresond to the same partition. Figure 9 shows the partition that corresponds to the diagrams for the security guard problem. Note that *the number of terminals in a diagram equals the size of any partition that corresponds to it.* Therefore equivalent diagrams have the same number of terminals, even if their internal structure is different.



Cell 1	Cell 2	Cell 3	Cell 4
Visitor = T, Gender = Man	Spy = T, Gender = Man	Worker = T, Gender = Man	Worker = T, Gender =Woman
Visitor = T, Gender = Woman	Spy = T, Gender = Woman		

Figure 9: The partition of parent assigments corresponding to each of the diagrams in Figure 8.

It is easy to see that for every partition, there is a corresponding diagram: We can build a tree structure whose branches corresond to the possible assignment of parent values. If two assignments are equivalent, the corresponding branches end in the same terminal node. We thus have the following lemma.

Lemma 6. *Consider the space of complete parent value assignments* $\mathbf{X} = \mathbf{x}$ *for a child node* Y.

1. *For every diagram structure* D*, there is a corresponding partition of the parent value assignments.*

2. *For every such partition $\equiv$, there is a corresponding diagram structure.*

A diagram semantically includes another if it is consistent with more inequality constraints. Therefore diagram D' includes D if and only if the partition of D' refines the partition of D; we denote this relationship by $D \leq D'$. The benefit of the partition representation is that inclusion depth in partition space is simply characterized by the size of the partition. Consider partitions of a finite set S elements. So the coarsest partition has size 1, and the maximally refined partition has size $|S|$. A refinement chain starting with partition $\equiv$ is a sequence

$$\equiv \leq \equiv_1 \leq \equiv_i \cdots \equiv_k,$$

where each element in the chain refines its predecessor but not vice versa. The **refinement depth** of partition $\equiv$ is the length k of a refinement chain starting with partition $\equiv$. The starting partition $\equiv$ is not included in the length count. The next proposition asserts that the refinement depth of a partition is simply the size $|S|$ of the most refined partition, minus the size of the partition. The basic reason for this is that to form a maximally long refinement chain, each partition in the chain should split exactly one cell of its predecessor into exactly two cells. Thus each partition in the chain contains exactly one more cell than its predecessor. Any other way to construct a refined partition is equivalent to merging two-way splits and therefore shortens the chain unnecessarily.

Proposition 7. *The refinement depth of a partition $\equiv$ on a finite set with N elements is the $N - |\equiv|$, where $|\equiv|$ is the size of the partition.*

By the correspondence Lemma 6, the inclusion depth of a diagram is the refinement depth of its corresponding partition. By Proposition 7, the refinement depth is the size of the partition, which equals the number of terminals in the diagram. Therefore the inclusion depth of a diagram equals the number of terminals in the diagram. We summarize our results in the following corollary.

Corollary 8. *The inclusion depth of a diagram is the number of its terminals. Therefore the Occam learner conjectures a diagram with the minimum number of terminals, if all such diagram structures are equivalent (i.e., they are consistent with exactly the same probabilistic inequalities). Otherwise it outputs ?.*

Figure 10 provides an example of the Occam learner. We leave for future work the analysis of the computational complexity of implementing the Occam learner.

Discussion. Combining the results of learning causal graphs and diagrams suggests a two-part approach: first, employ an Occam learner to find a maximally simple graph with a minimum number of edges, then another to find a maximally simple representation of context-sensitive causal relationships between a child node and its parents. Since every terminal in a diagram contains the same number of parameters, the number of terminals is equivalent to the number of free conditional probability parameters in a probability estimation diagram.

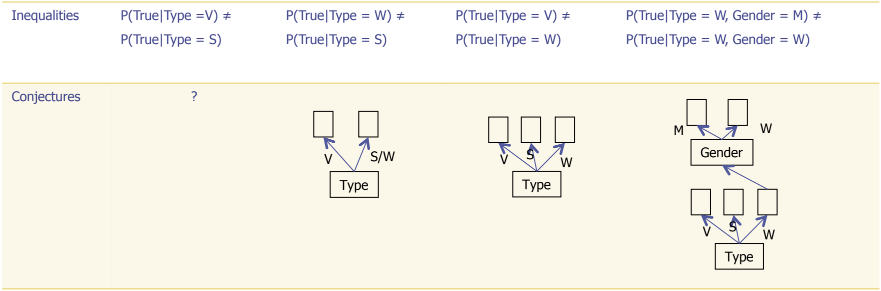


Figure 10: To illustrate the Occam learner for the security guard problem. We have used obvious abbreviations for variable values. The conditional probabilities are all of the form $P(Badge = T|\cdot)$, abbreviated as $P(T|\cdot)$.

Inductive simplicity as defined in terms of inclusion chain therefore agrees with standard statistical model selection criteria such as BIC and AIC that measure the complexity of a model by the number of parameters in the model. Such model selection criteria do not usually take into account the number of edges in a causal graph. However, as the number of parameters increases with the number of edges, for many graphs ranking by the number of parameters leads to very similar to results as our two-part scheme. The minimum message length criterion explicitly includes both the number of edges and the number of parameters in an additive overall complexity measure [8].

Given a probability estimation diagram, it is easy to translate paths in the diagram into probabilistic clauses: logical rules with probabilities attached. For instance, one of the paths in the left diagram of Figure 8 corresponds to the probabilistic clause

$$Badge = T \leftarrow Type = Worker, Gender = Woman; p = 0.7.$$

Such clauses provide a way to combine the formalism of first-order logic with probabilistic reasoning [28]. Khosravi and Schulte show that learning decision trees to augment causal graph structures is a scalable way to discover probabilistic clauses that provides accurate predictions on test cases [20].

9. Conclusion

An application of Occam's razor to inductive inference directs us to choose the simplest hypothesis consistent with the evidence. This principle is plausible but vague to the extent that the concept of simplicity is undefined. Learning theorist have developed a precise definition of inducive simplicity based on the topology of the space of alternative hypotheses. This leads to a learning-theoretic version of Occam's razor. This version can be justified in terms of learning performance: it is the only learning method that guarantees reliable, steady, and fast convergence to a correct hypothesis. As a case study, we applied Occam's inductive razor to learning causal relationships from observed correlations, an important and challenging practical problem. The Occam learner selects the causal graph that represents the observed dependencies among variables with a minimum number of edges. For context-sensitive dependencies, that may hold only in a given context, the Occam learner augments the causal graph with probability estimation diagrams that suffice to explain the observed correlations with a minimum number of free parameters. Probability estimation diagrams are easily converted to probabilistic logical rules, so causal learning can be used to discover logical rules that represent statistical patterns in the data.

The learning-theoretic version of Occam's razor has a clear justification in terms of learning performance guarantees. In many learning problems, including causal modelling, it leads to plausible inferences and to substantive insights into the methodological structure of the problem.

Acknowledgements

This research was supported by an NSERC discovery grant to the author. Preliminary results were presented at the Center for Formal Epistemology at Carnegie Mellon University. The author is grateful to the audience at the Center for helpful comments.

10. Proofs

Proposition 9. *The refinement depth of a partition $\equiv$ on a finite set with N elements is the $N - |\equiv|$, where $|\equiv|$ is the size of the partition.*

Proof. The proof is by downward induction on partition size $|\equiv|$. Base case: $|\equiv| = N$. Then the partition is maximally refined and hence a maximal refinement chain contains only $\equiv$, so its length is counted as 0. Inductive Step: Assume the hypothesis for $n+1$ and consider a starting partition $\equiv$ of size n. We can split one cell of $\equiv$ into two subcells to obtain a partition $\equiv'$ that contains exactly one more cell than $\equiv$. So $\equiv'$ contains $n+1$ cells. By inductive hypothesis, there is a refinement chain

$$\equiv' \leq \equiv_1 \cdots \equiv_{N-(n+1)}$$

that extends $\equiv'$ with $N-(n+1)$ elements. Hence the refinement depth of the starting partition $\equiv$ is *at least* $N-(n+1)+1 = N-n$.

To show that the refinement depth if $\equiv$ is *at most* $N-n$, consider any maximal refinement chain

$$\equiv \leq \equiv_1 \leq \equiv_i \cdots \equiv^N \tag{5}$$

where $\equiv^N$ denotes the maximally refined partition. We argue that (*) the partition $\equiv_1$ splits exactly one cell of the starting partition $\equiv$ in exactly two subcells. For suppose otherwise for contradiction. If $\equiv_1$ splits two or more cells of the starting partition, form a partition $\equiv_{0.5}$ that contains the first cell as in the starting partition $\equiv$, but splits the other cells as in partition $\equiv_1$. Then $\equiv_{0.5}$ refines $\equiv$ and is refined by $\equiv_1$. Therefore the chain

$$\equiv \leq \equiv_{0.5} \leq \equiv_1 \leq \equiv_i \cdots \equiv^N$$

refines the starting partition and is longer than the chain (5). So the original chain is not maximally long, contrary to assumption. This establishes that $\equiv_1$ splits at most one cell of the starting partition $\equiv$. If $\equiv_1$ splits the starting partition cell into more than two members, we can again construct an intermediate partition $\equiv_{0.5}$ by following only one of the splits and not the others. Then there exists a longer refinement chain than that in Equation (5), contrary to assumption. This establishes the claim (*) that the partition $\equiv_1$ splits exactly one cell of the starting partition $\equiv$ in exactly two subcells. Therefore $\equiv_1$ contains exactly one more cell than the starting partition:

$$| \equiv_1 | = | \equiv_1 | + 1 = n + 1.$$

Applying the inductive hypothesis to $\equiv_1$, it follows that the chain

$$\equiv_1 \leq \equiv_i \cdots \equiv^N$$

has length at most $N-(n+1)$, so the chain (5) has length at most $N-(n+1)+1 = n$ members. Since this chain was chosen to have maximum length, the refinement depth of the starting partition $\equiv$ is n. ■

Appendix: d-separation

An (undirected) **path** in G is a sequence of nodes such that every two consecutive nodes in the sequence are adjacent in G and no node occurs more than once in the sequence. A node Y is a **collider on undirected path** p in DAG G if p contains a triple $X \to Y \leftarrow Z$. If X and Z are adjacent in G, the collider Y is **shielded**, otherwise **unshielded**. Every BN structure defines a separability relation between a pair of nodes X, Y relative to a set of nodes $\mathbf{S}$, called **d-separation**: if X, Y are two variables and $\mathbf{S}$ is a set of variables disjoint from $\{X, Y\}$, then $\mathbf{S}$ d-separates X and Y if along every (undirected) path between X and Y there is a node W satisfying one of the following conditions:

1. W is a collider on the path and neither W nor any of its descendants is in $\mathbf{S}$, or
2. W is not a collider on the path and W is in $\mathbf{S}$.

We write $(X \perp\!\!\!\perp Y|\mathbf{S})_G$ if X and Y are d-separated by $\mathbf{S}$ in graph G. If two nodes X and Y are not d-separated by $\mathbf{S}$ in graph G, then X and Y are **d-connected** by $\mathbf{S}$ in G, written $(X \not\perp\!\!\!\perp Y|\mathbf{S})_G$.

Example. In the graph of Figure 2, the node `wet` is an unshielded collider on the path `sprinkler`$-$`wet`$-$`rain`; node `wet` is not a collider on the path `sprinkler`$-$`wet`$-$`slippery`. The pattern of the network has the same skeleton, but contains only two edges that induce the collider `wet`. The variables `sprinkler` and `rain` are d-separated given the set {`season`}, written (`sprinkler`$\perp\!\!\!\perp$ `rain`|`season`)$_G$, which can be seen as follows. There are two undirected paths from `sprinkler` to `rain`, namely `sprinkler` $-$ `wet` $-$ `rain` and `sprinkler` $-$ `season` $-$ `rain`.

For the first path, clause (1) of the definition of d-separation applies, since wet is a collider on the path sprinkler − wet − rain and neither wet nor its descendant slippery is contained in the conditioning set {season}. For the second path, clause (2) applies, since season is not a collider on the path sprinkler − season − rain and season is a member of the conditioning set {season}. The variables sprinkler and rain are *not* d-separated given the set {season, wet}, written $(\texttt{sprinkler} \not\perp\!\!\!\perp \texttt{rain}|\texttt{season})_G$, because wet is a collider on the path sprinkler − wet − rain contained in the conditioning set, which violates clause (1) of the definition of d-separation.

A fundamental theorem of causal graph theory entails that a causal graph structure over discrete variables is consistent with a set of observed dependencies if and only if for each observed dependency, the corresponding d-connection relation holds in the causal graph structure.

References

[1] James Bogen and James Woodward. Saving the phenomena. *The Philosophical Review*, 97:3:303–352, 1988.

[2] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in bayesian networks. In *UAI*, pages 115–123, 1996.

[3] Craig Boutilier, Thomas L. Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *J. Artif. Intell. Res. (JAIR)*, 11:1–94, 1999.

[4] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[5] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu. Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, 137:43–90, 2002.

[6] D. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2003.

[7] D. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2003.

[8] Joshua W Comley and David L Dowe. General bayesian networks and asymmetric languages. In *Proc. 2nd Hawaii International Conference on Statistics and Related Fields*, pages 5–8, 2003.

[9] G. Cooper. An overview of the representation and discovery of causal relationships using Bayesian networks. In C. Glymour and G. Cooper, editors, *Computation, Causation, and Discovery*, pages 4–62. AAAI Press/The MIT Press, 1999.

[10] Nir Friedman and Moises Goldszmidt. Learning Bayesian networks with local structure. In *NATO ASI on Learning in graphical models*, pages 421–459, 1998.

[11] Dan Geiger and David Heckerman. Knowledge representation and inference in similarity networks and bayesian multinets. *Artificial Intelligence*, 82(1-2):45–74, 1996.

[12] R. N. Giere. The significance test controversy. *The British Journal for the Philosophy of Science*, 23(2):170–181, 1972.

[13] C. Glymour. On the methods of cognitive neuropsychology. *British Journal for the Philosophy of Science*, 45:815–835, 1994.

[14] E. M. Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[15] S. Jain, D. Osherson, J. S. Royer, and A. Sharma. *Systems That Learn.* MIT Press, 2 edition, 1999.

[16] K. Kelly. *The Logic of Reliable Inquiry.* Oxford University Press, 1996.

[17] K. Kelly. Justification as truth-finding efficiency: How ockham's razor works. *Minds and Machines*, 14(4):485–505, 2004.

[18] K. Kelly. Why probability does not capture the logic of scientific justification. In C. Hitchcock, editor, *Contemporary Debates in the philosophy of Science*, pages 94–114. Wiley-Blackwell, London, 2004.

[19] Kevin T. Kelly and Conor Mayo-Wilson. Causal conclusions that flip repeatedly and their justification. In *UAI*, pages 277–285, 2010.

[20] Hassan Khosravi, Oliver Schulte, Jianfeng Hu, and Tianxing Gao. Learning compact markov logic networks with decision trees. *Machine Learning*, 89(3):257–277, 2012.

[21] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. Royal Statistics Society B*, 50(2):157–194, 1988.

[22] W. Luo. Learning Bayesian networks in semi-deterministic systems. In *Canadian AI 2006*, number 4013 in LNAI, pages 230–241. Springer-Verlag, 2006.

[23] W. Luo and O. Schulte. Mind change efficient learning. *Information and Computation*, 204:989–1011, 2006.

[24] W. Luo and O. Schulte. Mind change efficient learning. *Information and Computation*, 204:989–1011, 2006.

[25] E. Martin and D. N. Osherson. *Elements of Scientific Inquiry.* The MIT Press, Cambridge, Massachusetts, 1998.

[26] C. Meek. *Graphical Models: Selecting causal and statistical models.* PhD thesis, Carnegie Mellon University, 1997.

[27] R. E. Neapolitan. *Learning Bayesian Networks.* Pearson Education, 2004.

[28] Liem Ngo and Peter Haddawy. Answering queries from context-sensitive probabilistic knowledge bases. *Theor. Comput. Sci.*, 171(1-2):147–177, 1997.

[29] J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge university press, 2000.

[30] H. Putnam. Trial and error predicates and the solution to a problem of Mostowski. *The Journal of Symbolic Logic*, 30(1):49–57, 1965.

[31] O. Schulte. Discussion. What to believe and what to take seriously: a reply to David Chart concerning the riddle of induction. *The British Journal for the Philosophy of Science*, 51(1):151–153, 2000.

[32] O. Schulte. Inferring conservation laws in particle physics: A case study in the problem of induction. *The British Journal for the Philosophy of Science*, 51(4):771–806, December 2000.

[33] Oliver Schulte. Means-ends epistemology. *The British Journal for the Philosophy of Science*, 79(1):141–147, 1996.

[34] Oliver Schulte. The co-discovery of conservation laws and particle families. *Studies in the History and Philosophy of Modern Physics*, 39(2):288–314, 2008.

[35] Oliver Schulte, Gustavo Frigo, Russell Greiner, and Hassan Khosravi. A new hybrid method for Bayesian network learning with dependency constraints. In *Proceedings IEEE CIDM Symposium*, pages 53–60, 2009. Symposium on Computational Intelligence and Data Mining.

[36] Oliver Schulte, Wei Luo, and Russell Greiner. Mind-change optimal learning of Bayes net structure from dependency and independency data. *Information and Computation*, 208:63–82, 2010.

[37] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search.* MIT Press, 2000.

[38] M. Studeny. *Probabilistic Conditional Independence Structures.* Springer, 2005.

[39] I. Tsamardinos, L. E. Brown, and C. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

[40] T. S. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence (UAI 1990)*, pages 220–227, 1990.

[41] Y. Xiang, S. K. Wong, and N. Cercone. Critical remarks on single link search in learning belief networks. In *Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI 1996)*, pages 564–57, 1996.