Approximately counting and sampling knowledge states

Jeffrey Matayoshi

McGraw Hill ALEKS
jeffrey.matayoshi@aleks.com

Abstract

Approximately counting and sampling knowledge states from a knowledge space is a problem that is of interest for both applied and theoretical reasons. However, many knowledge spaces used in practice are far too large for standard statistical counting and estimation techniques to be useful. Thus, in this work we use an alternative technique for counting and sampling knowledge states from a knowledge space. This technique is based on a procedure variously known as subset simulation, the Holmes-Diaconis-Ross method, or multilevel splitting. We make extensive use of Markov chain Monte Carlo methods and, in particular, Gibbs sampling, and we analyze and test the accuracy of our results in numerical experiments.

AMS 2020 Mathematics Subject Classification: Primary 06A07; Secondary 91E45.
Keywords: Knowledge space theory, Subset simulation, Markov chain Monte Carlo, Gibbs sampling

1 Introduction

Knowledge spaces are combinatorial structures that are used to model the knowledge of learners in various academic fields of study (Doignon and Falmagne, 1985; Falmagne et al., 2013; Falmagne and Doignon, 1988, 2011). Given a set of items from such a field, a knowledge space contains all the subsets of these items that can realistically be known by a learner at a given time; these subsets of items are known as knowledge states. Because of the combinatorial nature of knowledge spaces, the number of knowledge states can grow exponentially with the number of items—this can lead to knowledge spaces of immense size, even when the number of items is relatively small. Due to these issues, many knowledge spaces used in practice are far too large for standard statistical counting and estimation techniques to be applied. Thus, accurately estimating the size of a given knowledge space is a challenging problem that requires the use of specialized techniques.

Both sampling and counting knowledge states have several uses in educational assessment. For example, as mentioned in Eppstein (2013a), knowing the distribution of the knowledge states gives information on the distribution of the items throughout the knowledge space, which can be taken advantage of by an adaptive assessment algorithm. A related observation is that a sample of knowledge states can also be used to assess how accurately a knowledge space fits empirical data, thus leading to more accurate procedures for building knowledge spaces. As yet another practical example, being able to compare the sizes of two versions of a knowledge space can be useful since, all else being equal, a smaller knowledge space is more efficient for an adaptive assessment algorithm. Next, from a more theoretical standpoint, having a procedure for estimating the size of a knowledge space would allow one to compare this value with other characteristics of the space, possibly giving further insight into the combinatorial nature of the structure. Finally, given the enormous numbers of states contained in many knowledge spaces, an argument could be made that having a reasonable estimate for the size of a knowledge space—or even a relatively accurate lower bound—would be of interest for no other reason than simply satisfying intellectual curiosity.

However, the problem of accurately estimating the size of a knowledge space is complicated by two somewhat contradictory issues. On the one hand, many knowledge spaces that are used in practice are far too large to allow a direct computation of the number of knowledge states; no existing computer would be able to complete such a computation in any reasonable amount of time. Knowing that, a different approach would be to estimate the size of a knowledge space by drawing samples from a larger family that contains the knowledge space, a family for which an accurate estimate of the size exists; an example of a larger such family would be the power set of the items in the knowledge space. Then, by finding the proportion of sets from this larger family that belong to the knowledge space, we would have our estimate for the size of the knowledge space. As before, however, this technique is ineffective for many knowledge spaces that are used in practice—the proportion of sets belonging to the knowledge space would be too small for standard estimation techniques to accurately estimate.

Thus, to estimate the size of a knowledge space we employ a technique that has been used in various areas of probability, combinatorics, and computer science; furthermore, it has also appeared in the analysis of the reliability of engineering systems. In the probability and combinatorics literature it is sometimes called the *Holmes-Diaconis-Ross method* (Diaconis and Holmes, 1995; Ross, 2002), in the engineering field it is known as *subset simulation* (Au and Beck, 2001), and in the rare event literature it is called *multilevel splitting* (Cérou et al., 2012; Glasserman et al., 1999). The key idea of the method, that of using a nested sequence of events, is attributed to John von Neumann, where it first appeared in the work of Kahn and Harris (1951).

The outline of the paper is as follows. We begin with an introduction to knowledge space theory (KST), where we review the basic concepts and existing results that are needed for this work. In subsequent sections we then discuss Markov chain Monte Carlo (MCMC) methods in detail, as these methods form

a core component of our procedure for estimating the size of a knowledge space. In addition to providing background information, we also prove a couple of results that show the Gibbs sampler, a particular MCMC method, satisfies the necessary conditions to be properly applied to the current problem. In the remainder of the work we then outline the specific details of our algorithm for estimating the size of a knowledge space, and we then evaluate its effectiveness in a set of numerical experiments.

2 Background

In this section we give a brief background of knowledge space theory (KST), where we introduce the relevant material that is necessary for developing our later methods and experiments. For a more thorough treatment of KST, we refer the reader to Falmagne and Doignon (2011). We begin by introducing the related notions of a *knowledge structure* and a *knowledge space*.

Definition 2.1. A *knowledge structure* is a pair $(Q, \mathcal{K})$ in which Q is a nonempty set, and $\mathcal{K}$ is a family of subsets of Q, containing at least Q and the empty set $\varnothing$. The set Q is called the *domain* of the knowledge structure. Its elements are referred to as *questions* or *items* and the subsets in the family $\mathcal{K}$ are labeled *(knowledge) states*. Since $\cup\mathcal{K} = Q$, we shall sometimes simply say that $\mathcal{K}$ is the knowledge structure when reference to the underlying domain is not necessary. A family of sets $\mathcal{F}$ is *closed under union* if for any sets $A, B \in \mathcal{F}$, we have $A \cup B \in \mathcal{F}$. If a knowledge structure $\mathcal{K}$ is closed under union, we say that $\mathcal{K}$ is a *knowledge space*.

Throughout this work we assume that Q is a finite set. Two useful concepts associated with families of sets are well-gradedness and 1-connectedness, which we define as in Doignon and Falmagne (1997) and Doble et al. (2001).

Definition 2.2. Let Δ denote the standard symmetric difference operation between sets. Given a family of sets, $\mathcal{F}$, a finite sequence of sets

$$A = K_0, K_1, \ldots, K_n = B$$

in $\mathcal{F}$ is called a *(stepwise) path* between A and B if $|K_{i-1}\Delta K_i| = 1$ for all $i = 1, \ldots, n$. If, additionally, $|A\Delta B| = n$, the sequence of sets is called a *tight path* between A and B. The family $\mathcal{F}$ is *1-connected* if there exists a stepwise path between any $A, B \in \mathcal{F}$, and it is *well-graded* if there exists a tight path between any $A, B \in \mathcal{F}$.

Example 2.3. Consider the following family of sets.

$$\mathcal{F} = \{\{a\}, \{c\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}$$

The only path from $\{a\}$ to $\{c\}$ in $\mathcal{F}$ is given by

$$\{a\}, \{a, b\}, \{a, b, c\}, \{b, c\}, \{c\}. \tag{2.1}$$

Note, however, this is not a tight path as it contains four total steps, but $|\{a\}\Delta\{c\}| = 2$. So, $\mathcal{F}$ is not well-graded, but it is 1-connected; this is easily seen by noting that (2.1) contains a (sub)path between each possible pair of sets. On the other hand, suppose we were to add the set $\{a, c\}$ to $\mathcal{F}$, resulting in a new family $\mathcal{F}'$:

$$\mathcal{F}' = \{\{a\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}.$$

There now exists a tight path from $\{a\}$ to $\{c\}$ in $\mathcal{F}'$ of the form

$$\{a\}, \{a, c\}, \{c\}.$$

Additionally, it is straightforward to check that a tight path exists between each pair of sets in $\mathcal{F}'$; thus, it follows that $\mathcal{F}'$ is well-graded.

A *learning space* is a particular type of knowledge space whose properties are motivated by the following pedagogical assumptions, as they appear in Falmagne and Doignon (2011).

Definition 2.4. A knowledge structure $(Q, \mathcal{K})$ is called a learning space if it satisfies the following conditions.

[L1] *Learning smoothness.* For any two knowledge states K, L such that $K \subset L$, there exists a finite chain of knowledge states

$$K = K_0 \subset K_1 \subset \cdots \subset K_p = L$$

such that $|K_i \setminus K_{i-1}| = 1$ for $1 \leq i \leq p$ and so $|L \setminus K| = p$.

[L2] *Learning consistency.* If K, L are two knowledge states satisfying $K \subset L$ and q is an item such that $K \cup \{q\} \in \mathcal{K}$, then $L \cup \{q\} \in \mathcal{K}$.

Note that, while originally introduced for pedagogical reasons, learning smoothness is actually a special case of the well-graded property. In fact, it was shown in Cosyn and Uzun (2009) that a well-graded knowledge space is equivalent to a learning space. Additionally, a well-graded knowledge space that is closed under intersection is called an *ordinal knowledge space*. Ordinal knowledge spaces have several properties that make them computationally appealing. In particular, Birkhoff's theorem (Birkhoff, 1937) allows us to define a partial order on the items in an ordinal knowledge space $\mathcal{K}$; furthermore, this partial order uniquely defines the ordinal knowledge space (see, for example, Eppstein, 2013a or Section 3.8 in Falmagne and Doignon, 2011). Having this partial order gives us a concise way to represent the ordinal knowledge space and to efficiently check if a set of items is part of the knowledge space.

To see this, we can derive the partial order for an ordinal knowledge space $\mathcal{K}$ as follows. Let q, r be items in our domain of knowledge, Q, and suppose that whenever a knowledge state $K \in \mathcal{K}$ contains r it also contains q; in this case, we have the ordering $q < r$. More formally, for any $q \in Q$ we can define

$$\mathcal{K}_q = \{K \in \mathcal{K} \,|\, q \in K\},$$

the family of all knowledge states containing q. Then, $q < r$ if and only if $\mathcal{K}_r \subset \mathcal{K}_q$. Furthermore, given an arbitrary set $A \subseteq Q$, we now have a simple procedure for checking if $A \in \mathcal{K}$. That is, it can be shown that $A \in \mathcal{K}$ if and only if for any $q \in A$ we have $\cap\mathcal{K}_q \subseteq A$ (for example, see Sections 3.4 and 3.6 in Falmagne and Doignon, 2011).

Most likely due to the computational advantages afforded by ordinal knowledge spaces, many knowledge spaces used in practice are ordinal (Desmarais and Pu, 2005; Doignon and Falmagne, 2016; Lynch and Howlin, 2014). Furthermore, while technically outside the field of KST proper, in related areas of education research the idea of having a partial order among the topics in a domain of knowledge is becoming more prevalent (Chaplot et al., 2016; Chen et al., 2018; Liang et al., 2015). Thus, while we prove our theoretical results for the general class of well-graded knowledge spaces whenever possible, for all of the above reasons we focus exclusively on ordinal knowledge spaces in our numerical experiments.

To that end, in our analyses we need to make use of the following definition, as it appears in Eppstein (2013b).

Definition 2.5. A *chain* in a partial order is a set of elements in which each pair of elements is comparable; equivalently, it is a sequence of items $q_0, q_1, \ldots$ such that $q_i < q_j$ if and only if $i < j$. A *chain cover* is a set of chains that together include all the items in the order. An *antichain* is a set of items, no two of which are comparable to each other. The *width* of a partial order is the maximum cardinality of any of its antichains, or equivalently—by Dilworth's Theorem; see below—the minimum number of chains in a chain cover.

Given a chain cover of the items in Q, it is straightforward to remove items from each chain until we are left with a partition of the items in Q; thus, in what follows, we assume that we are always dealing with a chain cover that is also a partition.

For completeness, we have included the statement of Dilworth's theorem below (Dilworth, 1950).

Theorem 2.6 (Dilworth's Theorem)**.** Let Q be a partially ordered set (e.g., the items in an ordinal knowledge space). There exists an antichain A, and a partition of the set into a family of chains, $\mathcal{C}$, where the size of A equals the number of chains in $\mathcal{C}$. In this case, $\mathcal{C}$ is a minimal chain covering of Q, and the width of $\mathcal{K}$ is defined to be the sizes of A and $\mathcal{C}$.

Following the procedure outlined in Eppstein (2013b), we can represent an ordinal knowledge space $\mathcal{K}$ as a bipartite graph, which then allows us to use the Hopcroft-Karp algorithm (Hopcroft and Karp, 1973) to efficiently compute a minimal chain covering $\mathcal{C}$ of Q; in our numerical experiments, we use the implementation from Eppstein (2002) to perform these specific computations.

We conclude this section with the following example illustrating several of the concepts from the previous paragraphs.

Example 2.7. Consider the following knowledge space on $Q = \{a, b, c, d\}$.

$$\mathcal{K} = \{\varnothing, \{a\}, \{b\}, \{a, b\}, \{a, c\}, \{b, d\}, \{a, b, c\}, \{a, b, d\}, \{a, b, c, d\}\} \tag{2.2}$$

Note that $\mathcal{K}$ is an ordinal knowledge space, as it is both well-graded and intersection-closed. From (2.2), we obtain the following partial order representing $\mathcal{K}$:

$$a < c \text{ and } b < d. \tag{2.3}$$

That is, a is in every state containing c, while b is in every state containing d. Defining the chains

$$\mathcal{C}_1 = \{a, c\}$$
$$\mathcal{C}_2 = \{b, d\},$$

we can see that, based on (2.3), $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2\}$ is a chain cover of the items in Q. Additionally, as a is not comparable to b and c is not comparable to d, $A = \{a, b\}$, $B = \{c, d\}$, $C = \{a, d\}$, and $D = \{b, c\}$ are all antichains of the same size as $\mathcal{C}$; hence, by Dilworth's Theorem, it follows that $\mathcal{C}$ is a minimal chain cover of the items in Q.

3 Estimating the size of a knowledge space

In this section we give a brief overview of the main technique we employ to estimate the size of a knowledge space. As mentioned previously, this method has been applied in many areas of probability, combinatorics, computer science, and engineering. It appears as the *Holmes-Diaconis-Ross method* (Diaconis and Holmes, 1995; Ross, 2002) in the probability and combinatorics literature, *subset simulation* (Au and Beck, 2001) in the engineering field, and *multilevel splitting* (Cérou et al., 2012; Glasserman et al., 1999) in the rare event literature. The underlying idea of the method is to use a nested sequence of events that terminates at the (rare) event of interest—this insight first appeared in the work of Kahn and Harris (1951), where it is attributed to John von Neumann.

The intuition is the following. Let $\mathcal{F}$ be an event for which we want to estimate the probability of occurrence. By assumption, $\mathcal{F}$ is a rare event, which means $P(\mathcal{F})$ is very small and, as such, directly estimating its value is not feasible. Following the procedure as it is described in Au and Beck (2001), we can instead construct a decreasing sequence of events, $\mathcal{E}_i$, which terminates at $\mathcal{F}$:

$$\mathcal{E}_n \supset \mathcal{E}_{n-1} \supset \cdots \supset \mathcal{E}_1 \supset \mathcal{E}_0 = \mathcal{F}.$$

Since the events $\mathcal{E}_i$ are nested, we then have

$$\begin{aligned} P(\mathcal{F}) = P(\mathcal{E}_0) &= P\left(\bigcap_{i=0}^{n} \mathcal{E}_i\right) \\ &= P\left(\mathcal{E}_0 \,\middle|\, \bigcap_{i=1}^{n} \mathcal{E}_i\right) P\left(\bigcap_{i=1}^{n} \mathcal{E}_i\right) \\ &= P\left(\mathcal{E}_0 \,|\, \mathcal{E}_1\right) P\left(\bigcap_{i=1}^{n} \mathcal{E}_i\right) \\ &\;\;\vdots \\ &= P\left(\mathcal{E}_n\right) \prod_{i=0}^{n-1} P\left(\mathcal{E}_i \,|\, \mathcal{E}_{i+1}\right). \end{aligned} \tag{3.1}$$

Without loss of generality, we can assume that $P(\mathcal{E}_n) = 1$. Thus, our problem of estimating $P(\mathcal{F})$ has now been transformed into one of estimating a sequence of conditional probabilities

$$P(\mathcal{E}_i \,|\, \mathcal{E}_{i+1}), \quad i = 0, \ldots, n-1, \tag{3.2}$$

where it is assumed that each conditional probability is easier to estimate than $P(\mathcal{F})$.

The next step in the process is to generate samples from each of the $\mathcal{E}_i$'s; doing this allows us to accurately estimate the conditional probabilities given by (3.2). To generate these samples, we use Markov chain Monte Carlo (MCMC) simulation. This connection between sampling and approximate counting has been known for some time (Jerrum et al., 1986; Sinclair and Jerrum, 1989), and such techniques have found widespread use in combinatorics to solve various counting and estimation problems; examples include estimating the permanent of a matrix (Jerrum et al., 2004) and approximately counting the number of solutions to the 0-1 knapsack problem (Morris and Sinclair, 2004). In the next section we give a brief introduction to Markov chain theory and the machinery that we need for our MCMC simulations. For more thorough introductions to these concepts, we refer the reader to Brémaud (1999) and Norris (1998) (for general Markov chain theory), or Liu (2001) and Robert and Casella (2004) (for more on MCMC methods).

4 Markov chain Monte Carlo

Let $X_1, X_2, \ldots$ be a sequence of random variables taking values in a set S. Such a sequence is called a *Markov chain* if the value of the sequence at time $n+1$ only depends on the value at time n. Formally, for any $i_0, i_1, \ldots, i_n, i_{n+1} \in S$,

this is written as

$$P(X_{n+1} = i_{n+1} \mid X_n = i_n, X_{n-1} = i_{n-1}, \ldots, X_1 = i_1, X_0 = i_0) = P(X_{n+1} = i_{n+1} \mid X_n = i_n), \forall n \in \mathbb{N}^0. \quad (4.1)$$

At this point, we must say a few words about terminology. In the standard Markov chain literature, any $i \in S$ is typically referred to as being a *state* of the Markov chain, with S then being called the *state-space* of the Markov chain. Thus, to avoid confusion with standard knowledge space terminology, we explicitly use the term "knowledge state" whenever we are referring to a set of items coming from a knowledge space, and we reserve the shortened version "state" for the values of a Markov chain.

For our purposes, a Markov chain state is composed of a subset of the items in a domain of knowledge Q. Since we are assuming that Q is a finite set, the state-space of any Markov chain we encounter is also finite. We can then define the *transition matrix* $\mathbf{P}$ of a Markov chain X_n as follows:

$$\begin{aligned} \mathbf{P} &= \{p_{ij}\}_{i,j \in S} \\ &= P(X_{n+1} = j \mid X_n = i). \end{aligned}$$

Thus, $\mathbf{P}$ is a matrix of size $|S| \times |S|$, where the ij-th entry gives the probability of the Markov chain jumping from state i to state j after one time step. Using the finiteness of the state-space once again, we can define a probability distribution π on the states of S as a row vector of length $|S|$ whose entries sum to one. Furthermore, such a distribution is said to be *stationary* with respect to $\mathbf{P}$ if

$$\pi^T = \pi^T \mathbf{P}. \quad (4.2)$$

One property of Markov chains that is important for our later results is the following.

Definition 4.1. A Markov chain is said to be *irreducible* if any state can be reached from any other state. That is, given any two states A and B, the Markov chain has a non-zero probability of transitioning from state A to state B after a finite number of steps.

The main idea of Markov chain Monte Carlo is to construct a Markov chain that converges to the distribution in which we are interested in sampling from. One general procedure for constructing such a chain is the following. Let P be a probability distribution on a set S, and let $\mathbf{X}_n = \left(\mathbf{x}_n^{(1)}, \ldots, \mathbf{x}_n^{(J)}\right)$ be a value from S, where each component $\mathbf{x}_n^{(j)}$ refers to a group of one or more variables (or, in our case, items). To generate a new value $\mathbf{X}_{n+1} = \left(\mathbf{x}_{n+1}^{(1)}, \ldots, \mathbf{x}_{n+1}^{(J)}\right)$, we sample the components individually as follows. For each value $\mathbf{x}_n^{(j)}$, we sample the new value, $\mathbf{x}_{n+1}^{(j)}$, conditioned on the newly updated values of $\mathbf{x}_{n+1}^{(1)}, \ldots, \mathbf{x}_{n+1}^{(j-1)}$. However, at the same time, we also condition on the remaining values that have yet to

be updated, $\mathbf{x}_n^{(j+1)}, \ldots, \mathbf{x}_n^{(J)}$. Thus, the j-th component is updated based on the conditional distribution given by $P\left(\mathbf{x}_{n+1}^{(j)} \,\middle|\, \mathbf{x}_{n+1}^{(1)}, \ldots, \mathbf{x}_{n+1}^{(j-1)}, \mathbf{x}_n^{(j+1)}, \ldots, \mathbf{x}_n^{(J)}\right)$. This general procedure is known as Gibbs sampling.

Algorithm 1 Gibbs sampler

Start with an initial sample $\mathbf{X}_0 = \left(\mathbf{x}_0^{(1)}, \ldots, \mathbf{x}_0^{(J)}\right)$
for $n = 0$ to $N - 1$ **do**
  Sample $\mathbf{x}_{n+1}^{(1)}$ from $P\left(\mathbf{x}_{n+1}^{(1)} \,\middle|\, \mathbf{x}_n^{(2)}, \ldots, \mathbf{x}_n^{(J)}\right)$
  for $j = 2$ to $J - 1$ **do**
    Sample $\mathbf{x}_{n+1}^{(j)}$ from $P\left(\mathbf{x}_{n+1}^{(j)} \,\middle|\, \mathbf{x}_{n+1}^{(1)}, \ldots, \mathbf{x}_{n+1}^{(j-1)}, \mathbf{x}_n^{(j+1)}, \ldots, \mathbf{x}_n^{(J)}\right)$
  end for
  Sample $\mathbf{x}_{n+1}^{(J)}$ from $P\left(\mathbf{x}_{n+1}^{(J)} \,\middle|\, \mathbf{x}_{n+1}^{(1)} \ldots, \mathbf{x}_{n+1}^{(J-1)}\right)$
  $\mathbf{X}_{n+1} \leftarrow \left(\mathbf{x}_{n+1}^{(1)}, \ldots, \mathbf{x}_{n+1}^{(J)}\right)$
end for

The Gibbs sampler is a Markov chain Monte Carlo method that first appeared in Geman and Geman (1984) and was later introduced to the statistics literature in Gelfand and Smith (1990). Because it samples from the conditional distributions of groups of one or more variables, rather than the full distribution, the Gibbs sampler is well-suited to problems where the number of dimensions is high. It has been observed that grouping the components into blocks—known as blocked Gibbs sampling—can help to increase the convergence speed of the chain (Amit and Grenander, 1991; Roberts and Sahu, 1997). As we discuss in Section 6, using the blocked components gives us an efficient way to generate states from a Markov chain using the partial order properties of an ordinal knowledge space. Thus, for these reasons, all of our numerical experiments use the blocked Gibbs sampler. (For more information on the Gibbs sampler see Brooks et al., 2011; Casella and George, 1992; Kroese et al., 2011).

5 Gibbs sampler on a knowledge space

Let $\mathbf{X}_n = \left(\mathbf{x}_n^{(1)}, \ldots, \mathbf{x}_n^{(J)}\right)$ represent a sample value of items in Q, where each $\mathbf{x}_n^{(j)}$ is a vector of indicator functions for a group of one or more items. In this section we show that, under certain conditions, the Gibbs sampler has the necessary convergence properties. In other words, given a probability distribution P on some family of sets, the distribution of the samples generated by the Gibbs sampler converges to P. To start, we need the following definition.

Definition 5.1. Let $\mathcal{F}$ be an arbitrary family of sets with $\cup\mathcal{F} = Q$. For any subset of items $A \subseteq Q$, we can define the function

$$d_{\mathcal{F}}(A) := \min_{\{B \in \mathcal{F} \,|\, A \subseteq B\}} |B \setminus A|. \tag{5.1}$$

For a set $A \subseteq Q$, $d_{\mathcal{F}}(A)$ gives the minimum number of items that would need to be added to A in order to obtain a set in $\mathcal{F}$; in the specific case when $\mathcal{F}$ is a knowledge structure, it gives the distance from the smallest knowledge state containing A. Alternatively, we could have defined $d_{\mathcal{F}}$ as the distance from the largest set contained in A; admittedly, this would be easier to work with when dealing with a knowledge space that is not closed under intersections. However, since we focus exclusively on ordinal knowledge spaces in our experiments, Definition 5.1 gives a slight improvement in computational efficiency. Yet another option would have been to define $d_{\mathcal{F}}$ as the distance from the closest set to A, regardless of whether that set contains A or is contained in A. However, such a computation is less efficient than either of the previous options, as it would require computing both the smallest set containing A *and* the largest set contained in A, and then taking the minimum distance to A between the two.

Next, we can define our sequence of nested events as follows.

Definition 5.2. Let $\mathcal{F}$ be as in Definition 5.1, and let $\mathcal{E}$ be a family of sets such that $\mathcal{F} \subseteq \mathcal{E}$ and $\cup\mathcal{E} = Q$. For any $i \in \mathbb{N}^0$, let

$$\mathcal{E}_i^{\mathcal{F}} = \mathcal{E}_i := \{A \in \mathcal{E} \,|\, d_{\mathcal{F}}(A) \leq i\}. \tag{5.2}$$

We are now ready to prove our first result, which shows that for a well-graded knowledge space $\mathcal{K}$ and a learning smooth family $\mathcal{E}$, where $\mathcal{K} \subseteq \mathcal{E}$, the family $\mathcal{E}_i$ is 1-connected.

Lemma 5.3. Let $\mathcal{K}$ be a well-graded knowledge space with $\cup\mathcal{K} = Q$, and let $\mathcal{E}$ be a learning smooth family such that $\mathcal{K} \subseteq \mathcal{E}$. For any $i \in \mathbb{N}^0$, let $\mathcal{E}_i = \mathcal{E}_i^{\mathcal{K}}$ be defined as in (5.2). Then, for any sets $A, B \in \mathcal{E}_i$, there exists a sequence $D_j \in \mathcal{E}_i$, $j = 0, \ldots, m$, such that $A = D_0$, $B = D_m$, and $|D_{j-1} \Delta D_j| = 1$, for any $j = 1, \ldots, m$; in other words, the family $\mathcal{E}_i$ is 1-connected.

Proof. Let $A, B \in \mathcal{E}_i$. By the definition of $\mathcal{E}_i$, there exist $\widetilde{A}, \widetilde{B} \in \mathcal{K}$ such that $A \subseteq \widetilde{A}$, $B \subseteq \widetilde{B}$, $|\widetilde{A} \setminus A| \leq i$ and $|\widetilde{B} \setminus B| \leq i$. Since $\mathcal{K}$ is well-graded, there exists a tight path in $\mathcal{K}$ (and, hence, in $\mathcal{E}_i$) from $\widetilde{A}$ to $\widetilde{B}$. Thus, if we can show that there exist paths in $\mathcal{E}_i$ from A to $\widetilde{A}$ and B to $\widetilde{B}$, the result then follows. Note that since A and B are arbitrary sets in $\mathcal{E}_i$, we only need to show this for one of them; so, without loss of generality, we next show that A and $\widetilde{A}$ are connected by a path in $\mathcal{E}_i$.

Since $A, \widetilde{A} \in \mathcal{E}$ and $A \subseteq \widetilde{A}$, by the learning smoothness of $\mathcal{E}$ there exists a tight path, $C_1, \ldots, C_n$, from A to $\widetilde{A}$. Note that for any C_j we have $A \subseteq C_j \subseteq \widetilde{A}$; thus, since $|\widetilde{A} \setminus A| \leq i$ it is clear that $|\widetilde{A} \setminus C_j| \leq i$ as well, which in turn implies that $C_j \in \mathcal{E}_i$. As discussed in the previous paragraph, this same argument holds for B and $\widetilde{B}$, and the result then follows. □

Let $\mathbf{x}_n^{(1)}, \ldots, \mathbf{x}_n^{(J)}$ be the components of the blocked Gibbs sampler. Then, for any two sets $A = \left(A^{(1)}, \ldots, A^{(J)}\right), B = \left(B^{(1)}, \ldots, B^{(J)}\right) \in \mathcal{E}_i$, we say that $A \sim_k B$ if $A^{(l)} = B^{(l)}$ for any $l \in \{1, \ldots, J\} \setminus \{k\}$. We are now ready to prove our main result.

Theorem 5.4. Let $\mathcal{K}$ and $\mathcal{E}$ be as in Lemma 5.3. For any $i \in \mathbb{N}^0$, let π_i be a probability distribution on $\mathcal{E}_i$, where $\pi_i(X) > 0$ for any $X \in \mathcal{E}_i$. Then, computing the conditional probability distribution in Algorithm 1 based on π_i, it follows that the resulting Markov chain $\mathbf{X}_n$ converges to its (unique) stationary distribution π_i.

Proof. The proof consists of two parts. We start by showing that π_i is a stationary distribution for the Markov chain $\mathbf{X}_n$. The next part of the proof then shows that $\mathbf{X}_n$ is irreducible; since $\mathbf{X}_n$ is defined on a finite family, by standard results from Markov chain theory (see Sections 3.3 and 3.4 in Brémaud, 1999, for example) it then follows that $\mathbf{X}_n$ converges to a unique stationary distribution which, by the first part of the proof, must be equal to π_i.

The transition matrix $\mathbf{P}$ of the sequence $\mathbf{X}_n$ is given by

$$\mathbf{P} = \mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_J,$$

where each $\mathbf{P}_l$ is the transition matrix of the l-th component of $\mathbf{X}_n$. To show that π_i is a stationary distribution of $\mathbf{X}_n$, we must have

$$\pi_i^T \mathbf{P} = \pi_i^T.$$

We claim that it is enough to show that

$$\pi_i^T \mathbf{P}_l = \pi_i^T, \tag{5.3}$$

for each $l = 1, \dots, J$; that is, if the transition matrix for each component preserves the distribution π_i, then the matrix $\mathbf{P}$ preserves π_i as well. To see this, we can repeatedly apply (5.3) as follows:

$$\begin{aligned}
\pi_i^T \mathbf{P} &= \pi_i^T \mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_J \\
&= \left(\pi_i^T \mathbf{P}_1\right) \mathbf{P}_2 \dots \mathbf{P}_J \\
&= \pi_i^T \mathbf{P}_2 \dots \mathbf{P}_J \qquad \text{by (5.3)} \\
&\;\;\vdots \\
&= \pi_i^T \mathbf{P}_J \\
&= \pi_i^T \qquad \text{by (5.3).}
\end{aligned}$$

Thus, the main result follows if we can show that (5.3) holds. To that end, let $U, V \in \mathcal{E}_i$. For each transition matrix $\mathbf{P}_l$, we can define the probability of a jump from U to V as

$$p_{UV,l} = \begin{cases} \dfrac{\pi_i(V)}{\sum_{W \sim_l V} \pi_i(W)}, & \text{if } U \sim_l V \\ 0, & \text{otherwise.} \end{cases}$$

We then have

$$\begin{aligned}\sum_{U\in\mathcal{E}_i}\pi_i(U)p_{UV,l} &= \sum_{U\sim_l V}\pi_i(U)p_{UV,l}\\ &= \sum_{U\sim_l V}\pi_i(U)\frac{\pi_i(V)}{\sum_{W\sim_l V}\pi_i(W)}\\ &= \pi_i(V)\frac{\sum_{U\sim_l V}\pi_i(U)}{\sum_{W\sim_l V}\pi_i(W)}\\ &= \pi_i(V).\end{aligned}$$

Thus, for each component, the update $\mathbf{P}_l$ preserves the distribution π_i, and it follows that π_i is a stationary distribution of $\mathbf{X}_n$.

We next show that $\mathbf{X}_n$ is irreducible. Let $A, B \in \mathcal{E}_i$. By Lemma 5.3, there exists a path $A = D_0, \ldots, D_m = B$ in $\mathcal{E}_i$ connecting A and B. Suppose we can show that for any $j = 0, \ldots, m-1$ and $n \in \mathbb{N}^0$ we have $\pi_i\left(\mathbf{X}_{n+1} = D_{j+1} \,|\, \mathbf{X}_n = D_j\right) > 0$. It would then follow that

$$\begin{aligned}\pi_i\left(\mathbf{X}_{n+m} = B \,|\, \mathbf{X}_n = A\right) &= \pi_i\left(\mathbf{X}_{n+m} = D_m \,|\, \mathbf{X}_n = D_0\right)\\ &\geq \prod_{j=0}^{m-1}\pi_i\left(\mathbf{X}_{n+j+1} = D_{j+1} \,|\, \mathbf{X}_{n+j} = D_j\right)\\ &> 0,\end{aligned}$$

thus proving that there is a non-zero probability of transitioning from A to B.

It remains to show that for any $j = 0, \ldots, m-1$ there is a non-zero probability of transitioning from D_j to D_{j+1} after one time step. Let $\{q\} = D_j \Delta D_{j+1}$, and let $\mathbf{X}_n$ be the current state of the Markov chain. Assume that $\mathbf{x}_n^{(k)}$ is the component of $\mathbf{X}_n$ containing the variable representing q. We have

$$\pi_i\left(\mathbf{X}_{n+1} = D_{j+1} \,|\, \mathbf{X}_n = D_j\right) = \prod_{l=0}^{J-1}\widetilde{p}_l,$$

where

$$\widetilde{p}_l = \begin{cases} p_{D_jD_j,l} & \text{if } l < k\\ p_{D_jD_{j+1},l} & \text{if } l = k\\ p_{D_{j+1}D_{j+1},l} & \text{if } l > k.\end{cases}$$

Notice that for any $l < k$ we have

$$\widetilde{p}_l = \frac{\pi_i\left(D_j\right)}{\sum_{W\sim_l D_j}\pi_i\left(W\right)} > 0,$$

where the inequality follows from the fact that $D_j \in \mathcal{E}_i$. Similarly, for $l \geq k$ we have

$$\widetilde{p}_l = \frac{\pi_i\left(D_{j+1}\right)}{\sum_{W\sim_l D_{j+1}}\pi_i\left(W\right)} > 0,$$

where the inequality follows from the fact that $D_{j+1} \in \mathcal{E}_i$. Thus, we have shown that

$$\prod_{l=0}^{J-1} \widetilde{p}_l > 0,$$

from which the claimed result follows. □

For our problem of estimating the size of a knowledge space, we assume that π_i is the uniform distribution on the sets in $\mathcal{E}_i$, for any $i \geq 0$; however, the above result is more general and holds for any probability distribution on $\mathcal{E}_i$, at least in theory (in practice, the Gibbs sampler may run into convergence issues if the distribution is highly non-uniform). Also, note that the above result makes it straightforward to sample knowledge states from a knowledge space—we simply need to run the Gibbs sampler with a uniform distribution on $\mathcal{E}_0$.

6 Bounds on the size of an ordinal knowledge space

Given a knowledge space $\mathcal{K}$ with $|Q| = n$, there are 2^n possible combinations of items. Taking $n = 300$, for example, gives a total possible number of knowledge states on the order of 10^{90}. In the specific case of an ordinal knowledge space, we can make use of the techniques developed in Eppstein (2013a,b) to derive upper and lower bounds for the number of knowledge states.

Following the procedure outlined in Eppstein (2013b), we can represent an ordinal knowledge space $\mathcal{K}$ as a bipartite graph, which then allows us to use the Hopcroft-Karp algorithm (Hopcroft and Karp, 1973) to efficiently compute a minimal chain covering $\mathcal{C}$ of Q. Let J be the size of the chain covering. From Theorem 2.6 we know the largest antichain A has size J as well. Because the elements of A are not comparable to each other, any possible combination of elements from A must be contained in at least one knowledge state; thus, as a lower bound we have $|\mathcal{K}| \geq 2^J$.

Next, we can use $\mathcal{C}$ to derive an upper bound on the size of $\mathcal{K}$. Let $C_1, C_2, \ldots, C_J$ be the J chains in $\mathcal{C}$. Since each C_j is a chain, there are $|C_j| + 1$ subsets of C_j that follow the partial order, which means that any knowledge state must contain one of these $|C_j| + 1$ subsets. Thus, taking the Cartesian product of these subsets across all the chains in $\mathcal{C}$, we get an upper bound for the size of $\mathcal{K}$ of the form $|\mathcal{K}| \leq \prod_{j=1}^{J} (|C_j| + 1)$.

Regarding the grouping of variables in Algorithm 1, we can use the chain cover $\mathcal{C}$ to define $\mathbf{X} = \left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(J)}\right)$, where each $\mathbf{x}^{(j)}$ corresponds to the items in C_j. Using this representation gives a convenient and efficient way to implement a Gibbs sampler on an ordinal knowledge space. Furthermore, when using this form of the Gibbs sampler, the length of the decreasing sequence of events $\mathcal{E}_i$ is typically much smaller—this is because we are starting from the reduced family of $\prod_{i=1}^{J}(|C_i| + 1)$ sets, rather than the complete power set of Q, as we would using a Gibbs sampler with one item in each component. Our

next result shows that the requirements of Theorem 5.4 are satisfied using this representation.

Lemma 6.1. Let $\mathcal{K}$ be a well-graded knowledge space with a chain cover $\mathcal{C} = \{C_1, \ldots, C_J\}$. Define $\mathcal{E}$ to be the Cartesian product of the subsets of the chains in $\mathcal{C}$. Then, $\mathcal{E}$ is a learning smooth family.

Proof. Let $A, B \in \mathcal{E}$, where $A \subseteq B$. Let $q_1^1, \ldots, q_{n_1}^1$ be the items in $C_1 \cap (B \setminus A)$, with $q_j^1 < q_{j+1}^1$, $j = 1, \ldots, n_1 - 1$. Then, it follows that $A_j^1 = A \cup \{q_1^1, \ldots, q_j^1\} \in \mathcal{E}$, for any $j = 1, \ldots, n_1$.

Next, let $q_1^2, \ldots, q_{n_2}^2$ be the items in $C_2 \cap (B \setminus A_{n_1}^1)$, with $q_j^2 < q_{j+1}^2$, $j = 1, \ldots, n_2 - 1$. Then, $A_j^2 = A_{n_1}^1 \cup \{q_1^2, \ldots, q_j^2\} \in \mathcal{E}$, for any $j = 1, \ldots, n_2$. Continuing this process $J - 2$ more times, we get

$$A \subset A_1^1 \subset \cdots \subset A_{n_1}^1 \subset A_1^2 \subset \cdots \subset A_{n_J-1}^M \subset A_{n_J}^M = B,$$

where each consecutive pair of sets in the sequence differs by one item. Thus, we have constructed a tight path between A and B, and the claimed result follows. □

7 Algorithm for estimating the size of a knowledge space

Our algorithm for estimating the size of a knowledge space is a modified version of the subset simulation procedure described in Au and Beck (2001). The first change is that, as mentioned previously, we use the Gibbs sampler to generate our Markov chains, rather than the modified Metropolis algorithm given in Au and Beck (2001). In addition to performing well in all of our numerical experiments, the Gibbs sampler avoids technical complications caused by the dependence between the groups of items that make up the components of our Markov chain states.

The second change concerns the way in which we start our Markov chain samples in each $\mathcal{E}_i$, and it is specifically motivated by the connected properties of the sets that we sample from. As shown in Lemma 5.3, any sets $A, B \in \mathcal{E}_i$ are connected by a path of sets in $\mathcal{E}_i$; furthermore, based on the proof of Lemma 5.3, it seems reasonable to assume that in most cases there are numerous possible paths from A to B. The algorithm as described in Au and Beck (2001) is more general and tries to account for the possibility that the state-space of the Markov chain consists of multiple disconnected regions; thus, it estimates each conditional probability by using several Markov chains started from different points in the state-space. The intuition is that, by having multiple starting points, the Markov chains are able to "explore" the entire space effectively. In our experiments, however, we have the advantage of dealing with a specific type of state-space, and we obtain better results by using a single Markov chain to estimate each conditional probability. We hypothesize that this is due to the fact that each $\mathcal{E}_i$ is (as mentioned previously) relatively well-connected. (Also,

see 1.11.3 in Geyer, 2011, for a more general discussion of the possible benefits of using one long simulation run over multiple shorter runs.)

Given a set of samples $\mathbf{X}_1, \ldots, \mathbf{X}_M$ drawn from $\mathcal{E}_i$, for any $j < i$ we can estimate the conditional probability of a set from $\mathcal{E}_j$ by

$$P\left(\mathcal{E}_j \mid \mathcal{E}_i\right) = P_{j,i} \approx \hat{p}_{j,i} = \frac{1}{M} \sum_{m=1}^{M} I_{\mathcal{E}_j}\left(\mathbf{X}_m\right), \tag{7.1}$$

where $I_{\mathcal{E}_j}$ is an indicator function that is one if a set is in $\mathcal{E}_j$ and zero otherwise. To simplify notation, when the conditioning set is clear, we drop the second index and refer to the above as P_j and $\hat{p}_j$. Let $\mathcal{K}$ be an ordinal knowledge space, and let $\mathcal{C}$ be a chain cover of the items in Q. Define $\mathcal{E}$ to be the Cartesian product of the chains in $\mathcal{C}$. Combining (3.1) and (7.1), we can estimate the proportion of sets in $\mathcal{E}$ that are knowledge states in $\mathcal{K}$ as

$$P\left(\mathcal{K} \mid \mathcal{E}\right) = P\left(\mathcal{K}\right) \approx \hat{p} = \prod_{n=1}^{N-1} \hat{p}_{i_n, i_{n-1}}. \tag{7.2}$$

That is, $\hat{p}$ is an estimate of the probability that a randomly chosen set from $\mathcal{E}$ is a knowledge state in $\mathcal{K}$. Next, let $0 < \alpha_0 < 1$ be our chosen target conditional probability, a value that at each $\mathcal{E}_i$ determines the next level in the sequence. Then, our implementation of an algorithm for estimating the size of a knowledge space proceeds as follows.

Algorithm 2 Approximately counting knowledge states

Inputs:
$\mathcal{E}$, the Cartesian product of the chains in $\mathcal{C}$; M, the number of samples to generate for each conditional probability estimate; α_0, the target conditional probability

Initialization:
Set $i_0 = |Q|$
Generate a sequence of M samples $\mathbf{X}_{1,i_0}, \ldots, \mathbf{X}_{M,i_0}$ from $\mathcal{E}_{i_0} = \mathcal{E}$ by sampling randomly from $\mathcal{E}$ (or, equivalently, this can be done by using Algorithm 1 and an arbitrary starting set in $\mathcal{E}$)
Set $j = i_0 - 1$
while $j > 1$ and $\hat{p}_{j-1,i_0} > \alpha_0$ **do**
  $j = j - 1$
end while
Set $k = 1$ and $i_k = j$

Iterations:
while $i_k > 0$ **do**
  Pick $\mathbf{X}_{0,i_k}$ at random from $\left\{\mathbf{X}_{1,i_{k-1}}, \ldots, \mathbf{X}_{M,i_{k-1}}\right\} \cap \mathcal{E}_{i_k}$
  Starting from $\mathbf{X}_{0,i_k}$, draw M samples $\mathbf{X}_{1,i_k}, \ldots, \mathbf{X}_{M,i_k}$ from $\mathcal{E}_{i_k}$ using Algorithm 1
  Set $j = i_k - 1$
  while $j > 1$ and $\hat{p}_{j-1,i_k} > \alpha_0$ **do**
    $j = j - 1$
  end while
  Set $k = k + 1$ and $i_k = j$
end while

Output:
Estimated number of knowledge states given by

$$|\mathcal{E}| * \prod_{n=1}^{k-1} \hat{p}_{i_n,i_{n-1}}$$

Regarding the choice of probability threshold α_0, this was investigated in some detail in Zuev et al. (2012), where it was suggested that, for all practical purposes, values of α_0 such that $0.1 \leq \alpha_0 \leq 0.3$ give comparable results. Our initial numerical experiments supported this claim, where we did not observe any meaningful difference in accuracy between the threshold values in this range. Thus, since a lower threshold results in having to estimate fewer conditional probabilities, we use a value of $\alpha_0 = 0.1$ in our simulations in Section 8.

To get a sense of how much variability there is with the above procedure, we

can use the techniques from Au and Beck (2001) to derive an estimate for the standard error of the sample statistic $\hat{p}$ from (7.2).[1] We start by analyzing the variance of the probability estimate $\hat{p}_{i_n}$ from (7.1). To that end, observe that the indicator function of $\mathcal{E}_{i_n}$, given by $I_{\mathcal{E}_{i_n}}$, is a Bernoulli random variable with mean P_{i_n} when applied to a sequence of samples $\mathbf{X}_1, \ldots, \mathbf{X}_M$ from $\mathcal{E}_{i_{n-1}}$; thus, it has a variance of $P_{i_n}(1-P_{i_n})$. Using the fact that $\mathbf{X}_1, \ldots, \mathbf{X}_M$ is a stationary sequence, we can then directly compute the variance of $\hat{p}_{i_n}$ as

$$\begin{aligned}
E\left[\hat{p}_{i_n} - P_{i_n}\right]^2 &= \frac{1}{M^2}\sum_{j=1}^{M}\sum_{k=1}^{M} E\left[\left(I_{\mathcal{E}_{i_n}}(\mathbf{X}_j) - P_{i_n}\right)\left(I_{\mathcal{E}_{i_n}}(\mathbf{X}_k) - P_{i_n}\right)\right] \\
&= \frac{E\left[\left(I_{\mathcal{E}_{i_n}}(\mathbf{X}_1) - P_{i_n}\right)^2\right]}{M} \\
&\quad + \frac{2}{M^2}\sum_{m=2}^{N}(M-m)E\left[\left(I_{\mathcal{E}_{i_n}}(\mathbf{X}_1) - P_{i_n}\right)\left(I_{\mathcal{E}_{i_n}}(\mathbf{X}_m) - P_{i_n}\right)\right] \\
&= \frac{P_{i_n}(1-P_{i_n})}{M} \\
&\quad + \frac{2}{M^2}\sum_{m=2}^{N}(M-m)E\left[\left(I_{\mathcal{E}_{i_n}}(\mathbf{X}_1) - P_{i_n}\right)\left(I_{\mathcal{E}_{i_n}}(\mathbf{X}_m) - P_{i_n}\right)\right] \\
&= \frac{P_{i_n}(1-P_{i_n})}{M}\left(1 + \frac{2}{M}\sum_{m=2}^{M}(M-m)\cdot R(m-1)\right),
\end{aligned}$$

where

$$R(t) = \frac{E\left[\left(I_{\mathcal{E}_{i_n}}(\mathbf{X}_1) - P_{i_n}\right)\left(I_{\mathcal{E}_{i_n}}(\mathbf{X}_{t+1}) - P_{i_n}\right)\right]}{P_{i_n}(1-P_{i_n})} \tag{7.3}$$

is the autocorrelation function of the sequence $I_{\mathcal{E}_{i_n}}(\mathbf{X}_1), \ldots, I_{\mathcal{E}_{i_n}}(\mathbf{X}_M)$. Thus, the conditional probability estimate $\hat{p}_{i_n}$ has a standard error of

$$\sigma_{\hat{p}_{i_n}} = \sqrt{\frac{P_{i_n}(1-P_{i_n})}{M}\left(1 + \frac{2}{M}\sum_{m=2}^{M}(M-m)\cdot R(m-1)\right)}. \tag{7.4}$$

Assuming further that we have independence across the samples from all the $\mathcal{E}_i$'s—that is, for any $j \neq i$ the samples in $\mathcal{E}_i$ are independent of the samples in $\mathcal{E}_j$—it was shown in Au and Beck (2001) that the standard error for the product of the estimated conditional probabilities is given by

$$\sigma_{\hat{p}} = P\left(\mathcal{K}\right)\sqrt{\sum_{n=1}^{k-1}\frac{\sigma_{\hat{p}_{i_n}}^2}{P_{i_n}}}. \tag{7.5}$$

[1] The work of Au and Beck (2001) focused on estimates for the coefficient of variation. However, as the coefficient of variation is simply the ratio of the standard error to the mean, it is straightforward to adapt their results to the specific case of standard errors.

Furthermore, Au and Beck (2001) provided evidence that (7.5) can give a good approximation to the true standard error even when the estimators $\hat{p}_{i_n}$ are correlated. In the following section we run simulations to verify the validity of (7.5) for our specific numerical experiments.

8 Applications

For our numerical experiments[2] we use an ordinal knowledge space $\mathcal{K}$ composed of 300 items (i.e., $|Q|$=300) as our starting point. Based on the statistics given in Table 1 of Cosyn et al. (2021), this number of items is similar to that of several knowledge spaces used in an actual implementation of KST. Of the $300^2 = 90,000$ possible pairs of items, 18,284 of these pairs are comparable under the partial order given by $\mathcal{K}$. These pairs, along with the order of the items in each pair, were for the most part chosen at random. In the cases when they weren't chosen at random, the pairs of items were ordered to ensure that the resulting relation is a proper partial order—i.e., that the relation is reflexive, antisymmetric, and transitive.

Our first example uses the restriction of $\mathcal{K}$ to a subset of 100 items, $\widetilde{Q}$, for which we can explicitly count the number of knowledge states—this can be efficiently done, for example, by using the unfolding algorithm outlined in Eppstein (2013a). Formally, let $\widetilde{\mathcal{K}}$ be the ordinal knowledge space resulting from intersecting each knowledge state in $\mathcal{K}$ with the 100 items in $\widetilde{Q}$. After applying this process, there are 2191 distinct pairs (out of a possible 10,000) in the partial order given by $\widetilde{\mathcal{K}}$, and for $|\widetilde{\mathcal{K}}|$ we obtain an explicit value on the order of 6.1969×10^{11}. We next generate a minimal chain covering $\widetilde{\mathcal{C}}$ of $\widetilde{\mathcal{K}}$ using the procedure described in Eppstein (2013b); for this part, we use the implementation of the Hopcroft-Karp algorithm from Eppstein (2002). Since $\widetilde{\mathcal{C}}$ contains 32 chains, we get a lower bound of $2^{32} \approx 4.29 \times 10^9$ for the number of knowledge states. Letting $\widetilde{C}_1, \ldots, \widetilde{C}_{32}$ be the chains in $\widetilde{\mathcal{C}}$, our upper bound is

$$\prod_{i=1}^{32} \Big(|\widetilde{C}_i| + 1 \Big) \approx 2.9490 \times 10^{18}.$$

Notice that this upper bound is already a significant improvement on the total number of possible subsets of $\widetilde{Q}$, which is given by $2^{100} \approx 1.2677 \times 10^{30}$.

We next apply Algorithm 2 with $\mathcal{E}$ equal to the Cartesian product of the chains in $\widetilde{\mathcal{C}}$. Using $M = 10^7$ samples to estimate each conditional probability gives an overall estimate of

$$\begin{aligned} P(\widetilde{\mathcal{K}}) \approx \hat{p} &= \prod_{n=1}^{8} \hat{p}_{i_n} \\ &\approx 2.1056 \times 10^{-7}. \end{aligned} \tag{8.1}$$

[2]The code for running these experiments is available at `https://github.com/jmatayoshi/state-sampler`.

Our estimate for the number of knowledge states in $\widetilde{\mathcal{K}}$ is then given by

$$\begin{aligned} |\mathcal{E}| * \hat{p} &\approx 2.9490 \times 10^{18} * 2.1056 \times 10^{-7} \\ &\approx 6.2094 \times 10^{11}. \end{aligned} \tag{8.2}$$

Note that the absolute difference between the estimated value in (8.2) and the explicitly computed size of 6.1969×10^{11} is on the order of 1.25×10^{9}, a seemingly large number. However, in relative terms the difference is only about 0.2% of the actual value, and we submit that this level of accuracy would be sufficient for most practical purposes.

Table 1 gives further information on the computations for (8.1). Specifically, the table contains the estimated conditional probabilities based on the different families $\mathcal{E}_i$ that appear in the sampling procedure. For example, the first row shows the estimated probability of sampling a set from $\mathcal{E}_{12}$ given that the sets are being sampled from $\mathcal{E}_{100}$ (where $\mathcal{E}_{100} = \mathcal{E}$ is the Cartesian product of the chains in $\widetilde{\mathcal{C}}$); we can see that about 10% of the sampled sets from $\mathcal{E}_{100}$ are also contained in $\mathcal{E}_{12}$. Then, as another example, the last row shows that when sampling sets from $\mathcal{E}_1$, slightly under 8% of the time the obtained set is actually a knowledge state from $\widetilde{\mathcal{K}}$ (as $\mathcal{E}_0 = \widetilde{\mathcal{K}}$). Taking the product of all these conditional probabilities, and then multiplying the result by the size of $\mathcal{E}$, we arrive at our estimate (8.2).

Conditional probability	**Estimate**
$P(\mathcal{E}_{12} \mid \mathcal{E}_{100})$	0.101622
$P(\mathcal{E}_{8} \mid \mathcal{E}_{12})$	0.119788
$P(\mathcal{E}_{6} \mid \mathcal{E}_{8})$	0.209713
$P(\mathcal{E}_{4} \mid \mathcal{E}_{6})$	0.126765
$P(\mathcal{E}_{3} \mid \mathcal{E}_{4})$	0.273108
$P(\mathcal{E}_{2} \mid \mathcal{E}_{3})$	0.211882
$P(\mathcal{E}_{1} \mid \mathcal{E}_{2})$	0.146924
$P(\mathcal{E}_{0} \mid \mathcal{E}_{1})$	0.076529

Table 1: Conditional probability estimates for $\widetilde{\mathcal{K}}$ using $M = 10^7$ samples.

Our next analysis is an attempt to quantify the uncertainty in our probability estimate (8.1). As a start, to get an idea of the amount of correlation there is between our generated samples, we can look at the autocorrelation values for the indicator functions of the $\mathcal{E}_i$'s; these values correspond to the output of the autocorrelation function $R(t)$, defined in (7.3). As shown in Figure 1, at a time lag of one the autocorrelation is either close to zero or negative; in the latter case, as the lag increases the autocorrelation quickly converges to zero.

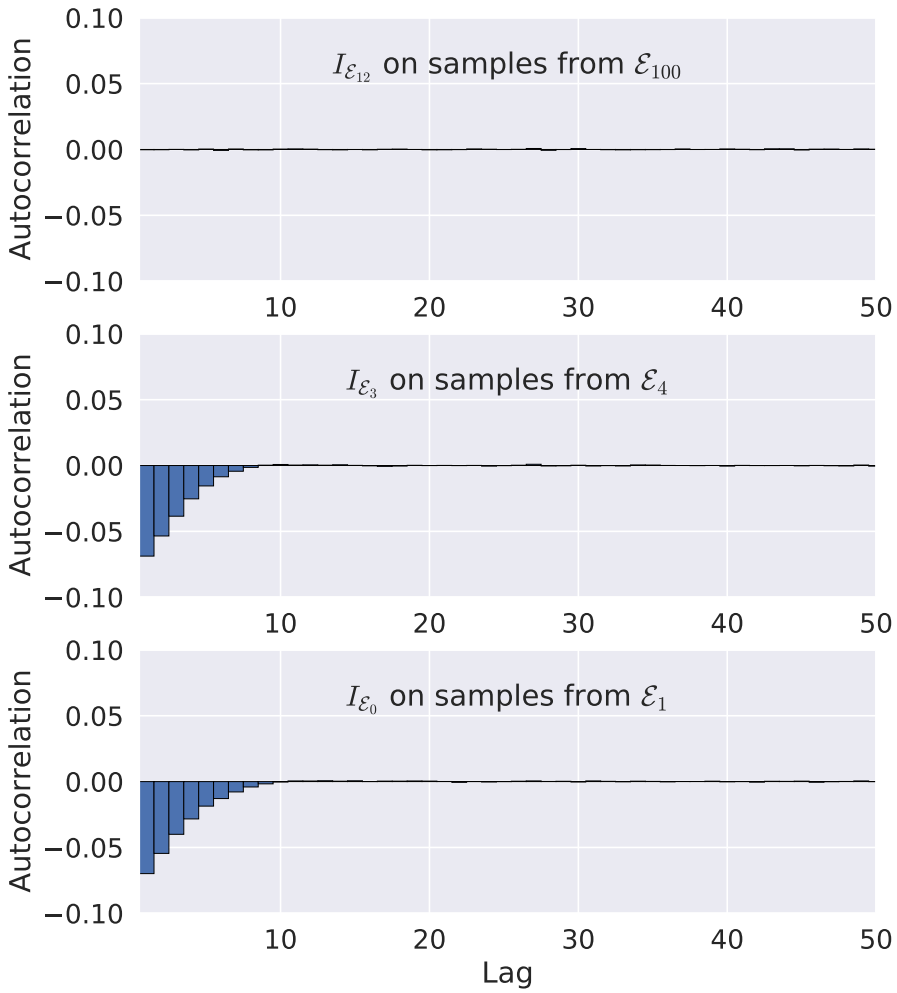

Figure 1: Autocorrelation plots for three different indicator functions using the subdomain of 100 items, $\widetilde{Q}$. Two of the examples initially show a negative correlation, with the values then converging towards zero as the lag increases.

Our next step is to compute $\sigma_{\hat{p}}$, the standard error of our estimate (8.1), by using formulas (7.4) and (7.5). The results in Figure 1 suggest there is some degree of correlation for nearby samples, but very little for samples further away from each other. Thus, in Figure 2 we compute the values of $\sigma_{\hat{p}}$ under two scenarios. In the first scenario—represented by the solid (orange) line—we assume there is no correlation between the samples generated within each $\mathcal{E}_i$ (which means the summation in (7.4) is zero). Then, in the second scenario—shown by the dashed (green) line—we compute (7.4) with the values of $R(t)$ estimated from the samples used to compute (8.1). In this latter case, we set $R(t) = 0$ for any $t > 100$; the decreasing autocorrelation values in Figure 1 suggest that this is a reasonable approximation. To evaluate the accuracy of our computed values of $\sigma_{\hat{p}}$, we use the following procedure. For a given value of M, we run Algorithm 2 to estimate the conditional probabilities in Table 1, using M samples for each estimate; based on these conditional probabilities, we compute $\hat{p}$ using (7.2). We then repeat this procedure until we have a sample of 10,000 different realizations of $\hat{p}$ for which we can compute the standard deviation—the results are given by the (blue) dots in Figure 2 for various values of M. As shown, the results from the simulations are aligned very closely with the curve using the estimated autocorrelation from the data. Thus, as suggested in Au and Beck (2001), (7.5) appears to be a reasonable estimate of the standard error when the autocorrelation among the samples is taken into account.

Now that we have validated the accuracy of our formula for computing the standard errors, we can use it to derive a confidence interval for our probability estimate. An inspection of the sampling distributions of the simulations in Figure 2 shows they are approximately normally distributed; thus, we compute our confidence intervals using a normal approximation interval. To test the validity of this procedure, we use the following approach. As discussed in the previous paragraph, for a given value of M we have 10,000 different estimates of $\hat{p}$. For each estimate we compute the confidence intervals for three different confidence levels and check whether the true value of $P(\widetilde{\mathcal{K}})$ is contained in the intervals. Based on the results, we can then check the coverage probabilities of these intervals over the 10,000 different simulation runs; that is, for each confidence level we compute the proportion of the time that the true value is contained in the associated interval.

The results are shown in Figure 8. While the actual coverage probabilities are slightly below the nominal coverage probabilities for the lower sample sizes—i.e., the true value is contained in the confidence intervals less than expected for the smaller sample sizes—the performance is better with the larger sample sizes. For example, focusing on the 95% confidence intervals, for any sample size larger than 25,000 the smallest coverage probability is 0.944, just slightly below the nominal coverage probability of 0.95. Based on these results, as well as the fact that our estimate (8.1) uses a sample size of 10^7—much larger than any of the sample sizes in Figure 8—it seems reasonable to use this procedure to generate a confidence interval around the point estimate (8.1).

Taking all of this into account, we compute a 95% confidence interval for our estimate (8.1) using a normal approximation interval and $\sigma_{\hat{p}}$, with the au-

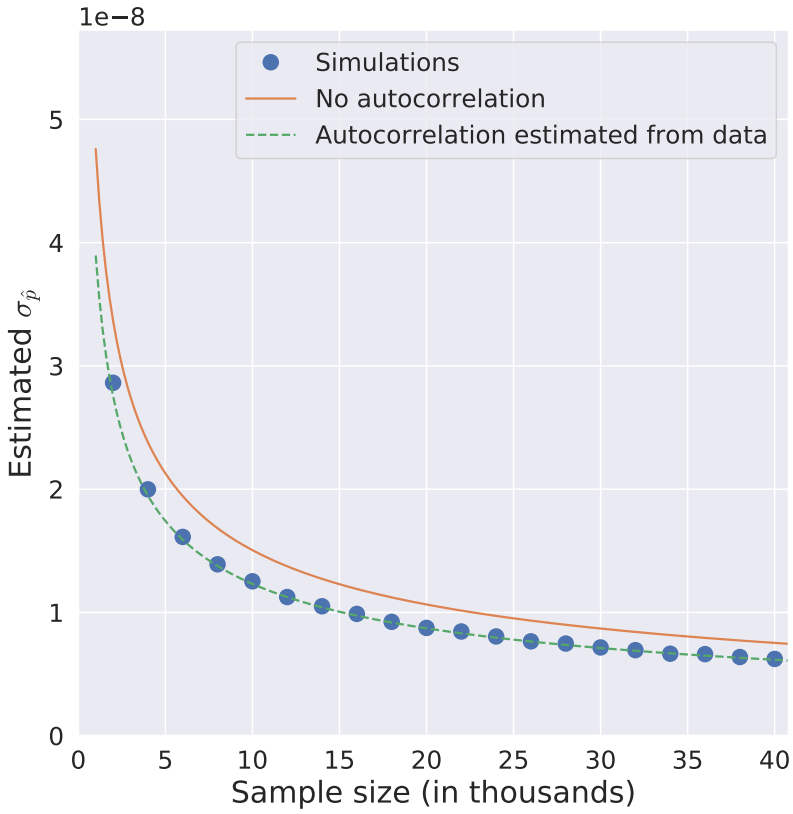

Figure 2: Estimated standard error as a function of the sample size of the conditional probability estimates, $\hat{p}_{i_n}$, using the subdomain of 100 items, $\widetilde{Q}$. The (blue) dots are estimates from 10,000 independent runs of Algorithm 2 at each of the different sample sizes. Note that using the autocorrelation estimated from the data results in a close match between the standard deviations of the repeated simulations (blue dots) and the standard errors from (7.5) (dashed green line).

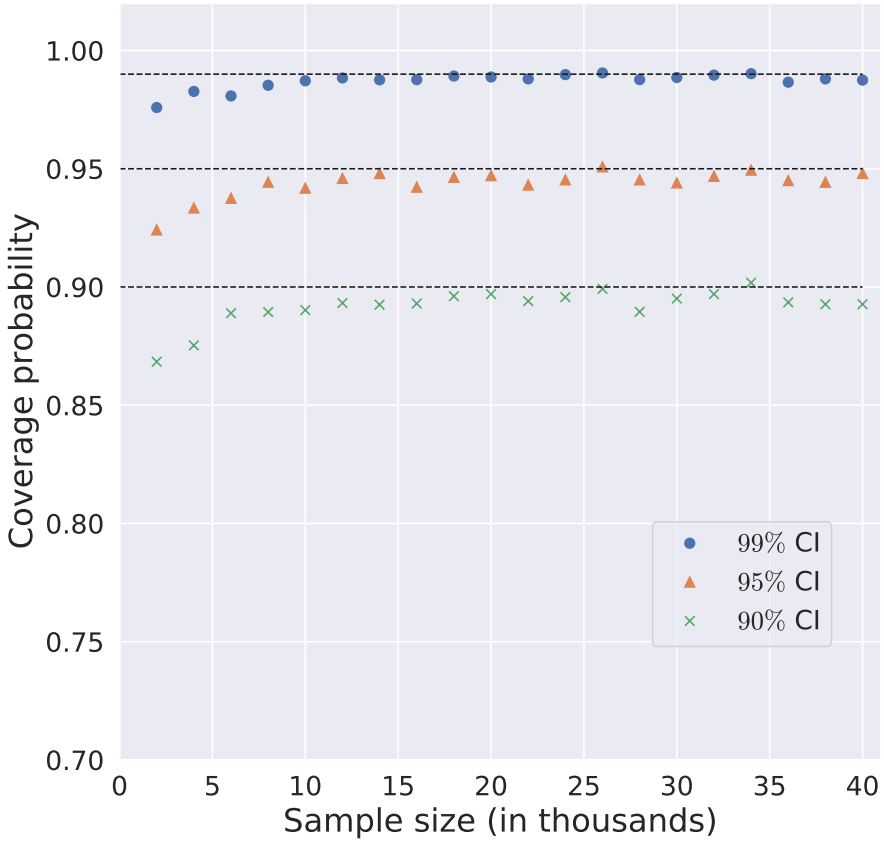

Figure 3: Coverage probabilities for various confidence intervals computed from 10,000 independent runs of Algorithm 2 at each of the different sample sizes.

tocorrelation function estimated from the data. After obtaining this interval, we multiply it by $|\widetilde{\mathcal{E}}|$ to obtain the following 95% confidence interval for the size of $\widetilde{\mathcal{K}}$:

$$(6.1869 \times 10^{11},\ 6.2319 \times 10^{11}).$$

We next turn our attention to the full domain of 300 items. The chain covering $\mathcal{C}$ of the items in Q contains 65 chains, which gives a lower bound of $2^{65} \approx 3.6893 \times 10^{19}$ for the number of knowledge states in $\mathcal{K}$. Letting $C_1, \ldots C_{65}$ be the chains in $\mathcal{C}$, our upper bound is

$$\prod_{i=1}^{65} (|\mathcal{C}_i| + 1) \approx 2.8606 \times 10^{44}.$$

Using one simulation run with $M = 10^7$ gives estimates for the 22 conditional probabilities in Table 2; based on these conditional probabilities, our estimate for the overall probability $\hat{p}$ is

$$\begin{aligned} P(\mathcal{K}) \approx \hat{p} &= \prod_{n=1}^{22} \hat{p}_{i_n} \\ &\approx 6.2627 \times 10^{-19}. \end{aligned} \tag{8.3}$$

Finally, our resulting estimate for the total number of knowledge states in $\mathcal{K}$ is given by

$$\begin{aligned} |\mathcal{E}| * \hat{p} &\approx 2.8606 \times 10^{44} * 6.2627 \times 10^{-19} \\ &\approx 1.7915 \times 10^{26}. \end{aligned} \tag{8.4}$$

We next quantify the uncertainty in this estimate by computing a confidence interval. As before, we begin by examining the indicator function autocorrelation values for three different examples; the results are shown in Figure 4. Note that in two of these examples the correlation is initially negative, with the values then converging towards zero as the lag increases; overall, this seems to indicate some level of correlation in the sampled sets. Using (7.4), (7.5) and the conditional probabilities in Table 2, in Figure 5 we show the estimated standard errors under the assumption of no correlation between the samples within each $\mathcal{E}_i$—represented by the solid (orange) line—and with the values of $R(t)$ estimated from the samples used to compute (8.3)—shown by the dashed (green) line. These estimates of the standard error are then compared to the standard deviations from 1,000 simulations each at various values of M, shown by the (blue) dots.[3] As can be seen in the plots, the actual standard deviations from the simulation values are roughly equivalent to the standard errors computed by using the autocorrelation estimated from the data. Thus, as with our previous

[3]Note that while the simulations in Figure 2 used a sample size of 10,000 to compute each standard deviation, due to the extra computational demands of the larger knowledge space, $\mathcal{K}$, a smaller sample size of 1,000 is used for the results in Figure 5.

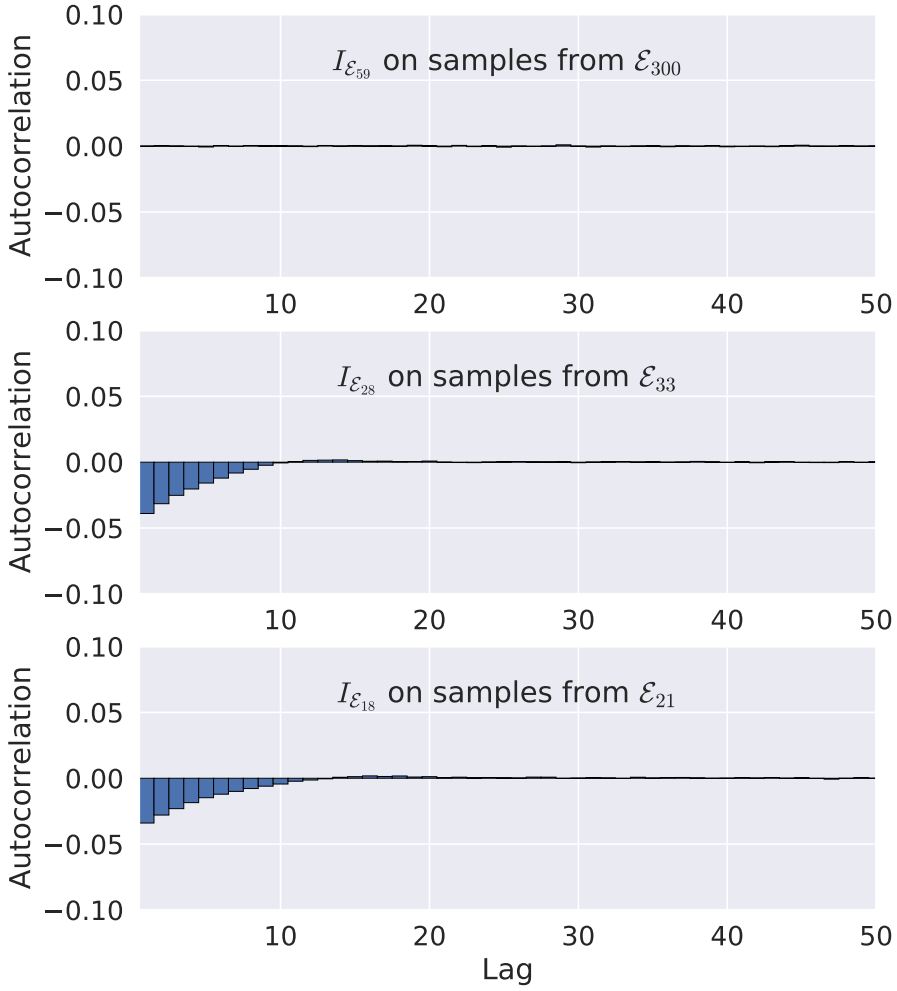

Figure 4: Autocorrelation plots for three different indicator functions using the full domain of 300 items, Q. As in Figure 1, two of the examples initially show a negative correlation, with the values then converging towards zero as the lag increases.

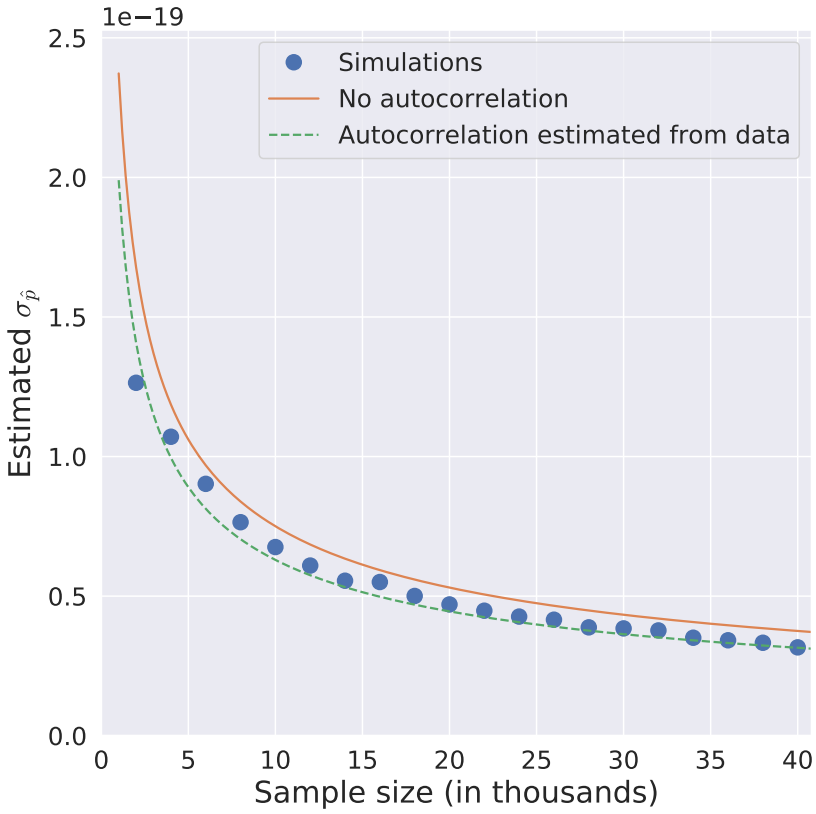

Figure 5: Estimated standard error as a function of the sample size of the conditional probability estimates, $\hat{p}_{i_n}$, using the full domain of 300 items, Q. The (blue) dots are estimates from 1,000 independent runs of Algorithm 2 at each of the different sample sizes. Note that using the autocorrelation estimated from the data results in a close match between the standard deviations of the repeated simulations (blue dots) and the standard errors from (7.5) (dashed green line).

Conditional probability	**Estimate**
$P(\mathcal{E}_{59} \mid \mathcal{E}_{300})$	0.112414
$P(\mathcal{E}_{47} \mid \mathcal{E}_{59})$	0.101169
$P(\mathcal{E}_{39} \mid \mathcal{E}_{47})$	0.106080
$P(\mathcal{E}_{33} \mid \mathcal{E}_{39})$	0.114070
$P(\mathcal{E}_{28} \mid \mathcal{E}_{33})$	0.109741
$P(\mathcal{E}_{24} \mid \mathcal{E}_{28})$	0.123257
$P(\mathcal{E}_{21} \mid \mathcal{E}_{24})$	0.165288
$P(\mathcal{E}_{18} \mid \mathcal{E}_{21})$	0.130088
$P(\mathcal{E}_{16} \mid \mathcal{E}_{18})$	0.219147
$P(\mathcal{E}_{14} \mid \mathcal{E}_{16})$	0.188575
$P(\mathcal{E}_{12} \mid \mathcal{E}_{14})$	0.157812
$P(\mathcal{E}_{10} \mid \mathcal{E}_{12})$	0.127093
$P(\mathcal{E}_{9} \mid \mathcal{E}_{10})$	0.323435
$P(\mathcal{E}_{8} \mid \mathcal{E}_{9})$	0.299725
$P(\mathcal{E}_{7} \mid \mathcal{E}_{8})$	0.274500
$P(\mathcal{E}_{6} \mid \mathcal{E}_{7})$	0.247633
$P(\mathcal{E}_{5} \mid \mathcal{E}_{6})$	0.219288
$P(\mathcal{E}_{4} \mid \mathcal{E}_{5})$	0.188764
$P(\mathcal{E}_{3} \mid \mathcal{E}_{4})$	0.156212
$P(\mathcal{E}_{2} \mid \mathcal{E}_{3})$	0.121349
$P(\mathcal{E}_{1} \mid \mathcal{E}_{2})$	0.083901
$P(\mathcal{E}_{0} \mid \mathcal{E}_{1})$	0.043514

Table 2: Conditional probability estimates for $\mathcal{K}$ using $M = 10^7$ samples.

experiments on the restricted knowledge space $\widetilde{\mathcal{K}}$, it appears that we can accurately estimate $\sigma_{\hat{p}}$—i.e., the standard error in our estimate (8.3)—using (7.5) with $R(t)$ estimated from the data. Once again using a normal approximation interval, we first construct a 95% confidence interval around our probability estimate (8.3); we then multiply this resulting interval by $|\mathcal{E}|$ to obtain the following 95% confidence interval for the estimated size of $\mathcal{K}$:

$$(1.7804 \times 10^{26},\ 1.8027 \times 10^{26}).$$

9 Discussion

In this work we introduced and evaluated a method for estimating the size of a knowledge space. This method requires the drawing of samples from specific families of sets that are related to the knowledge space of interest. To handle the generation of these samples, we discussed how Markov chain Monte Carlo methods can be used. In particular, we showed that the Gibbs sampler is well-suited to this problem, especially in the case of an ordinal knowledge space, where it can take advantage of the partial order properties of the space to efficiently generate sample sets. Furthermore, these same sampling techniques can also be used to sample states directly from the knowledge space itself.

As an application of these techniques, we computed an estimate for the size of a (relatively) small knowledge space for which we could directly count the number of knowledge states. We then performed an analysis of the uncertainty around this estimate, where simulation results suggested that the proposed procedure for deriving standard errors and confidence intervals is valid. Based on these results, we then computed an estimate of the size of a much larger knowledge space, along with the associated confidence interval, where a direct counting of the knowledge states is not feasible.

In the case of this larger knowledge space composed of 300 items, we estimated the number of knowledge states to be on the order of 1.7915×10^{26}. As discussed at the beginning of Section 8, the number of items in this knowledge space is a reasonable approximation of many knowledge spaces used in practical applications—as such, it is worth reflecting on the scale of the numbers involved. Given that the the maximum number of possible knowledge states on a set of 300 items is $2^{300} \approx 2.0370 \times 10^{90}$, this estimated number of knowledge states represents an exceedingly small proportion of the maximum possible, with this proportion being on the order of 10^{-64}. However, in absolute terms it is a staggeringly large number, and it thus emphasizes the difficulties inherent in adaptive assessment procedures that attempt to identify a small set of very likely knowledge states (see, for example, Matayoshi et al., 2021, for further background on these issues). Furthermore, examples exist of knowledge spaces defined on sets of items as large as 1500 or more; applying the techniques in this paper to knowledge spaces containing such a large number of items, our estimates for the number of knowledge states have been as high as 10^{140}.

Knowing the scale of the numbers involved could help guide further development of adaptive assessments in KST. For example, in assessments that involve large knowledge spaces, it may no longer be desirable to focus on precisely identifying the entire knowledge state of a student—instead, perhaps the goal could be to focus on certain subsets of the domain of knowledge that contain the most important and core topics. Related to this, some previous algorithms for constructing knowledge spaces have been motivated by the desire to identify a knowledge space that is as small as possible (see, for example, Section 11 in Doignon and Falmagne, 2016); as mentioned in the introduction, all else being equal a smaller knowledge space makes it easier to run an adaptive assessment. However, in light of the estimated sizes seen in this work, such an objective may

not be as important as previously thought. That is, with a knowledge space containing roughly 10^{26} (or even more) knowledge states, reducing this number by a half, a tenth, or even a thousandth, would likely have little practical effect on an adaptive assessment. Thus, with insights such as these, hopefully the results of this study can be of some use to researchers working in KST, on adaptive assessments, or in other related areas.

10 Acknowledgements

The author is grateful to Hasan Uzun, Eric Cosyn, and Christopher Doble for many useful discussions during the course of this work.

References

Amit, Y. and Grenander, U. (1991). Comparing sweep strategies for stochastic relaxation. *Journal of Multivariate Analysis*, 37:197–222.

Au, S.-K. and Beck, J. L. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277.

Birkhoff, G. (1937). Rings of sets. *Duke Mathematical Journal*, 3(3):443–454.

Brémaud, P. (1999). *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, New York.

Brooks, S., Gelman, A., Jones, G., and Meng, X., editors (2011). *Handbook of Monte Carlo Methods*. CRC Press, New York.

Casella, G. and George, E. (1992). Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174.

Cérou, F., Moral, P. D., Furon, T., and Guyader, A. (2012). Sequential Monte Carlo for rare event estimation. *Statistics and Computing*, 22(3):795–808.

Chaplot, D. S., Yang, Y., Carbonell, J., and Koedinger, K. R. (2016). Data-driven automated induction of prerequisite structure graphs. *International Educational Data Mining Society*.

Chen, P., Lu, Y., Zheng, V. W., and Pian, Y. (2018). Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 39–48. IEEE.

Cosyn, E. and Uzun, H. (2009). Note on two sufficient axioms for a well-graded knowledge space. *Journal of Mathematical Psychology*, 53(1):40–42.

Cosyn, E., Uzun, H., Doble, C., and Matayoshi, J. (2021). A practical perspective on knowledge space theory: ALEKS and its data. *Journal of Mathematical Psychology*, 101:102512.

Desmarais, M. C. and Pu, X. (2005). A Bayesian student model without hidden nodes and its comparison with item response theory. *International Journal of Artificial Intelligence in Education*, 15(4):291–323.

Diaconis, P. and Holmes, S. (1995). Three examples of Monte Carlo Markov chains: At the interface between statistical computing, computer science, and statistical mechanics. In Aldous, D., Saloff-Coste, L., Spencer, J., and Steele, J., editors, *Discrete Probability and Algorithms*, volume 72, pages 43–56. Springer, New York, NY.

Dilworth, R. (1950). A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51(1):161–166.

Doble, C., Doignon, J.-P., Falmagne, J.-C., and Fishburn, P. (2001). Almost connected orders. *Order*, 18:295–311.

Doignon, J.-P. and Falmagne, J. (2016). Knowledge spaces and learning spaces. *arXiv: Combinatorics*, pages 1–51.

Doignon, J.-P. and Falmagne, J.-C. (1985). Spaces for the assessment of knowledge. *International Journal of Man-Machine Studies*, 23:175–196.

Doignon, J.-P. and Falmagne, J.-C. (1997). Well-graded families of relations. *Discrete Mathematics*, 173:35–44.

Eppstein, D. (2002). PADS, a library of Python Algorithms and Data Structures. `http://www.ics.uci.edu/~eppstein/PADS`.

Eppstein, D. (2013a). Learning sequences: An efficient data structure for learning spaces. In Falmagne, J.-C., Albert, D., Doble, C., Eppstein, D., and Hu, X., editors, *Knowledge Spaces: Applications in Education*, chapter 13, pages 287–304. Springer-Verlag, Heidelberg.

Eppstein, D. (2013b). Projection, decomposition, and adaption of learning spaces. In Falmagne, J.-C., Albert, D., Doble, C., Eppstein, D., and Hu, X., editors, *Knowledge Spaces: Applications in Education*, chapter 14, pages 305–322. Springer-Verlag, Heidelberg.

Falmagne, J.-C., Albert, D., Doble, C., Eppstein, D., and Hu, X., editors (2013). *Knowledge Spaces: Applications in Education*. Springer-Verlag, Heidelberg.

Falmagne, J.-C. and Doignon, J.-P. (1988). A class of stochastic procedures for the assessment of knowledge. *British Journal of Mathematical and Statistical Psychology*, 41:1–23.

Falmagne, J.-C. and Doignon, J.-P. (2011). *Learning Spaces*. Springer-Verlag, Heidelberg.

Gelfand, A. and Smith, A. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Geyer, C. (2011). Introduction to Markov chain Monte Carlo. In Brooks, S., Gelman, A., Jones, G., and Meng, X.-L., editors, *Handbook of Markov Chain Monte Carlo*, chapter 1, pages 3–48. Chapman & Hall/CRC, New York.

Glasserman, P., Heidelberger, P., Shahabuddin, P., and Zajic, T. (1999). Multilevel splitting for estimating rare event probabilities. *Operations Research*, 47(4):585–600.

Hopcroft, J. and Karp, R. (1973). An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231.

Jerrum, M., Sinclair, A., and Vigoda, E. (2004). A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51(4):671–697.

Jerrum, M., Valiant, L., and Vazirani, V. (1986). Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188.

Kahn, H. and Harris, T. (1951). Estimation of particle transmission by random sampling. *National Bureau of Standards: Applied Mathematics Series*, 12:27–30.

Kroese, D., Taimre, T., and Botev, Z. (2011). *Handbook of Monte Carlo Methods*. Wiley, Hoboken, New Jersey.

Liang, C., Wu, Z., Huang, W., and Giles, C. L. (2015). Measuring prerequisite relations among concepts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1668–1674.

Liu, J. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer, New York.

Lynch, D. and Howlin, C. P. (2014). Real world usage of an adaptive testing algorithm to uncover latent knowledge. In *Proceedings of the 7th International Conference of Education, Research and Innovation*, pages 504–511.

Matayoshi, J., Cosyn, E., and Uzun, H. (2021). Are we there yet? Evaluating the effectiveness of a recurrent neural network-based stopping algorithm for an adaptive assessment. *International Journal of Artificial Intelligence in Education*, 31(2):304–336.

Morris, B. and Sinclair, A. (2004). Random walks on truncated cubes and sampling 0-1 knapsack solutions. *SIAM Journal on Computing*, 34:195–226.

Norris, J. (1998). *Markov Chains*. Cambridge University Press, Cambridge.

Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer, New York.

Roberts, G. and Sahu, S. (1997). Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *Journal of the Royal Statistical Society*, 59:291–317.

Ross, S., editor (2002). *Simulation*. Academic Press, New York.

Sinclair, A. and Jerrum, M. (1989). Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and computation*, 82:93–133.

Zuev, K., Beck, J., Au, S., and Katafygiotis, L. (2012). Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. *Computers and Structures*, 92-93:283–296.