Probabilistic Active Learning of Functions in Structural Causal Models

Paul K. Rubenstein*, **Ilya Tolstikhin, Philipp Hennig, Bernhard Schölkopf**
Max Planck Institute for Intelligent Systems, Tübingen
*Machine Learning Group, University of Cambridge

Abstract

We consider the problem of learning the functions computing children from parents in a Structural Causal Model once the underlying causal graph has been identified. This is in some sense the second step after causal discovery. Taking a probabilistic approach to estimating these functions, we derive a natural myopic active learning scheme that identifies the intervention which is optimally informative about all of the unknown functions jointly, given previously observed data. We test the derived algorithms on simple examples, to demonstrate that they produce a structured exploration policy that significantly improves on unstructured base-lines.

1 Introduction

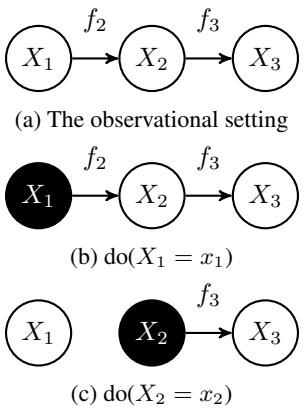
Large parts of the literature on causality are concerned with learning the causal graph of a system of random variables [Tong and Koller, 2001, Eberhardt, 2010, Hyttinen et al., 2013, Mooij et al., 2016]. Also known as *causal discovery* or *causal inference*, this problem is motivated by realistic problems in science: a biologist may wish to discover genes responsible for regulating other genes in a cell; a public health researcher may wish to know whether certain habits in a population (e. g. smoking) influence certain health outcomes (e. g. probability of developing lung cancer).

The starting point of this paper is to consider what should be done *after* the causal graph of a system of variables has already been identified. That is, it is known which variables are functions of which other variables, but the precise functional relationships are still unknown. Thus, although we understand the causal relationships in a coarse sense, we will not be able to accurately predict the result of an intervention to the system, possibly with implications for decisions made based on this uncertainty.

For instance, suppose that in a cell, Gene A up-regulates Gene B and Gene B down-regulates Gene C. We know that reducing expression of Gene A will lead to a decrease in expression of Gene B which, in turn, will lead to an increase in expression of Gene C. However, without more precise knowledge of the relationships between genes, we will be unable to quantitatively predict the effect of applying a drug that reduces expression of Gene A by 20%. Similarly, if our goal is to reduce overall levels of lung cancer in the population, then knowing only that smoking causes cancer is insufficient to know what the best public health policy should be: would it be better if we could persuade 50% of smokers to stop smoking completely, or persuade every smoker to reduce their consumption by 50%?

For a *passive* agent supplied with data generated by the system, learning the functional relationships between parents and children reduces simply to separate regression problems—one for each unknown function—once the causal graph is known. Many knowledge acquisition problems, however, can be phrased as a sequential decision making process in which the data that is received at the next point in time is affected by a decision made based on the data that has already been observed. A biologist does not blindly perform a series of costly and time-consuming experiments, only looking at the generated data once the last is over; the data from each experiment would be analysed before the next is performed, thus informing which experiment would be best to perform next.

Below, we formalise this problem using the language of Structural Causal Models (SCMs), also known as Structural Equation Models [Pearl, 2009, Bollen, 2014]. An SCM, in essence, consists of functions connecting child variables with their causal parents, and is equipped with a notion of *intervention* in which a variable (or subset of variables) is externally forced to take a particular value



(a) The observational setting

(b) do($X_1 = x_1$)

(c) do($X_2 = x_2$)

Figure 1: Even in the simple setting of three variables whose graph is a chain, there is a non-trivial trade-off between the information one expects to gain by performing different interventions.

(or values). We use these interventions as an idealised mathematical representation of performing an experiment. We take a Bayesian approach to estimating the functional relationships between parents and children, which naturally gives rise to an active learning algorithm to decide on the next 'experiment' to perform.

While our approach works with causal graphs that are arbitrary DAGs, the non-triviality of this problem is apparent even in the simple case of three variables whose causal graph is a chain (Figure 1). Our goal in this situation is, in a sense to be made precise later, to learn the functions f_2 and f_3. At each point in time, we must decide whether to perform one of the possible interventions or to passively observe the system, with each different action having some cost. If we make a passive observation, we will learn *something* about both f_2 and f_3, though only in areas where the distributions over X_1 and X_2 put probability mass. In the finite sample setting, we are very unlikely to learn anything about the functions in areas of low probability of their inputs. If we intervene on X_1 and choose what value to set it to, we can decide precisely *where* we want to learn about f_2 and we will also learn about f_3 in some region, although we would be uncertain about where. If we intervene on X_2, we can learn a precise aspect of f_3, but will learn nothing about f_2. How should we decide which action to pick?

The problem considered in this paper and our approach to solving it have a close connection to ideas in Bayesian optimization [Osborne et al., 2009, Shahriari et al., 2016]. In Bayesian optimization, the goal is to find the extremum of a function that is expensive to evaluate, possibly exploiting known structure to speed up the search. Due to this expense, the information obtained from each evaluation should be used efficiently. In contrast, however, in our setting we are not simply interested in finding the extremum of an unknown function, but rather to learn the entire function (or set of functions) in some sense. Tong and Koller [2000] considered a similar setting, but only treated discrete variables.

Below, we begin with a formal definition of Structural Causal Models (Section 2). Section 3 states the precise problem we are tackling. Section 4 provides a Bayesian formulation for inference in this setting, which is then complemented by an active learning scheme to guide exploration (Section 5). Section 6 provides empirical evaluations on synthetic toy examples.

2 Structural Causal Models

We now formally define Structural Causal Models (SCMs), the learning of which we will study in the remainder. For notational simplicity, our definition deviates from the wider literature by including the set of interventions modelled by the SCM.

Definition 1. *Suppose that $X_1, \dots, X_N, E_1, \dots, E_N$ are variables with each X_n and E_n taking value in $\mathcal{X}_n = \mathbb{R}$ and $\mathcal{E}_n = \mathbb{R}$ respectively. We write $X = (X_n)_{n=1,\dots,N}$ for the vector of variables, and $\mathcal{X} = \prod_{n=1,\dots,N} \mathcal{X}_n$ for their domain, similarly for E and $\mathcal{E}$. A Structural Causal Model (SCM) $\mathcal{M} = (\mathcal{S}, \mathcal{I}, \mathbb{P}_E)$ is a tuple consisting of the following three quantities:*

- *$\mathcal{S}$ is a set of* structural equations $X_n = f_n(X_{pa(n)}, E_n)$*, where $pa(n) \subset \{1, \dots, N\}$ and $X_{pa(n)}$ is the vector of variables $(X_m)_{m \in pa(n)}$. We call the $X_{pa(n)}$ the* causal parents *of X_n.*
- *$\mathcal{I}$ is a set of* interventions*. An intervention is a mathematical operation that replaces a subset of the equations in $\mathcal{S}$ with equations setting the variables to specific constants (e. g. $X_n = 3$). We write the intervention $i \in \mathcal{I}$ that intervenes on the subset of variables $X_{var(i)}$, $var(i) \subseteq \{1, \dots, n\}$, setting them to values $x_{val(i)}$, as $\mathrm{do}(X_{var(i)} = x_{val(i)})$ (we will often abuse notation by writing $\mathrm{do}(i)$ instead). We write $\mathcal{S}^{\mathrm{do}(i)}$ for the resulting equations.*
- *$\mathbb{P}_E$ is a distribution over the* exogenous *(aka. noise, unexplained) variables E taking value in $\mathcal{E} = \mathbb{R}^N$. This distribution is fixed and does not change due to interventions.*

We will only consider acyclic[1] SCMs. In this case, given any fixed value e of the variables E, there is a unique value $x \in \mathcal{X}$ such that each structural equation in $\mathcal{S}$ is satisfied;[2] thus $\mathbb{P}_E$ induces a distribution over $\mathcal{X}$ via these

[1] That is, the directed graph with nodes $\{1, \dots, N\}$ with edges $n \to m$ if and only if $n \in \mathrm{pa}(m)$ is a DAG.

[2] That is, for any $e \in \mathcal{E}$, there is a unique $x \in \mathcal{X}$ such that

unique solutions. We refer to this as the *observational distribution* of the SCM, and write it as $\mathbb{P}_X^{\mathrm{do}(\varnothing)}$ (where $\varnothing$ signifies the 'empty' intervention). Under each intervention, the resulting intervened structural equations $\mathcal{S}^{\mathrm{do}(i)}$ are also acyclic. It follows by the same reasoning as above that the SCM implies a distribution $\mathbb{P}_X^{\mathrm{do}(i)}$ over $\mathcal{X}$. Therefore, once all of the parameters of the model $\mathcal{M}$ have been fixed, it implies a set of distributions indexed by intervention: $\{\mathbb{P}_X^{\mathrm{do}(i)} : i \in \mathcal{I}\}$. We assume that the empty intervention $\varnothing$ is an element of $\mathcal{I}$.

An important note is that once the graphical structure has been fixed, the parameters of the model $\mathcal{M}$ are the functions f_n and the distribution $\mathbb{P}_E$ over the variables E. We will make the non-trivial assumption that the structural equations are *non-linear with additive Gaussian noise*, meaning that each equation is of the form $X_n = f_n(X_{\mathrm{pa}(n)}) + E_n$, where each E_n is a zero-mean Gaussian random variable. This setting comes with results on the identifiability of the causal graph [Peters et al., 2011].

Example 1. *Consider the following SCM $\mathcal{M} = (\mathcal{S}, \mathcal{I}, \mathbb{P}_E)$ represented by Figure 1, where*

$$\mathcal{S} = \left\{ \begin{aligned} X_1 &= 3 + E_1, \\ X_2 &= X_1^2 + E_2, \\ X_3 &= 2X_2 + \sin(X_2) + E_3 \end{aligned} \right\},$$

$$\mathcal{I} = \{\varnothing\} \cup \{\mathrm{do}(X_i = x_i) \mid x_i \in \mathbb{R},\ i = 1, 2\},$$
$$E_n \sim \mathcal{N}(0, \sigma_n^2), \quad n = 1, 2, 3.$$

The observational distribution of $\mathcal{M}$ factorises as $\mathbb{P}_{X_1X_2X_3}^{\mathrm{do}(\varnothing)} = \mathbb{P}_{X_1}^{\mathrm{do}(\varnothing)} \mathbb{P}_{X_2|X_1}^{\mathrm{do}(\varnothing)} \mathbb{P}_{X_3|X_2}^{\mathrm{do}(\varnothing)}$. Under, for instance, the intervention $\mathrm{do}(X_2 = 0)$, the equation $X_2 = X_1^2 + E_2$ in $\mathcal{S}$ is replaced by $X_2 = 0$. Under the distribution $\mathbb{P}_{X_1X_2X_3}^{\mathrm{do}(X_2=0)}$, X_1 and X_3 are independent and X_2 is degenerate.

3 Problem setup

Suppose there exists an SCM $\mathcal{M} = (\mathcal{S}, \mathcal{I}, \mathbb{P}_E)$ with $\mathcal{S} = \{X_n = f_n(X_{\mathrm{pa}(n)}) + E_n : n = 1, \ldots, N\}$ and $\mathbb{P}_E \sim \mathcal{N}(0, \Lambda)$ where Λ is diagonal. We assume that the graphical structure is known, but the functions f_n themselves are not. For simplicity, we assume that Λ is known.[3] We are given data $\mathcal{D}$ drawn from the observational and a variety of interventional distributions of $\mathcal{M}$; each element of $\mathcal{D}$ is a tuple (i, x), where x is an independent draw from $\mathbb{P}_X^{\mathrm{do}(i)}$. We are interested in two separate tasks.

Problem 1: Estimating $\mathcal{M}$. Using $\mathcal{D}$, learn functions $\hat{f}_n$ such that the estimated model $\widehat{\mathcal{M}} = (\hat{\mathcal{S}}, \mathcal{I}, \mathbb{P}_E)$ with $\hat{\mathcal{S}} = \{X_n = \hat{f}_n(X_{\mathrm{pa}(n)}) + E_n : n = 1, \ldots, N\}$ is 'close' to the true model $\mathcal{M}$ in some sense. Part of this problem is to define a sensible notion of closeness between SCMs.

Problem 2: Active learning. We can select an intervention $i \in \mathcal{I}$ at some cost $c(i)$ and observe a single draw from $\mathbb{P}_X^{\mathrm{do}(i)}$. Which i should be selected to ensure the next estimation $\widehat{\mathcal{M}}$ after incorporating the new datum is as close to $\mathcal{M}$ as possible?

The rest of this section is devoted to defining a risk functional to provide a notion of closeness between $\widehat{\mathcal{M}}$ and $\mathcal{M}$. There may be no single best way to define this notion of closeness, as desirable properties may be dependent on particular use case.[4] Consider, for instance, the following scenarios in which the ultimate goal is to:

- Approximate each function f_n so that a practitioner can visually interpret the relationship between parents and children (e.g. identifying genes as 'excitatory' or 'inhibitory', or identifying a threshold dosage at which a drug under medical trial is considered toxic.).
- Predict the result of an intervention that cannot be practically carried out, or is potentially dangerous to do so (e.g. raising interest rates, or giving a patient a drug).
- Better control a system in a variety of environments or conditions (e.g. learning the dynamics of a complex vehicle to be employed in variable conditions)

We will suppose that for each function f_n with domain $\mathcal{X}_{\mathrm{pa}(n)} = \prod_{m \in \mathrm{pa}(n)} \mathcal{X}_m$ we are supplied with a probability measure Π_n over $\mathcal{X}_{\mathrm{pa}(n)}$ specifying the importance of learning the pointwise value of f_n at each input $x_{\mathrm{pa}(n)} \in \mathcal{X}_{\mathrm{pa}(n)}$. That is, Π_n puts large amounts of mass in areas that we should learn f_n precisely, small amounts of mass in areas that we should learn f_n only approximately, and zero mass in areas for which we do not care about learning f_n at all. For the (estimated) function $\hat{f}_n$, we use the risk functional L_n below. The weighted sum of these according to the importance of each function gives rise to the *total risk* L, where $f = (f_n)_{n=1,\ldots,N}$ and $\hat{f} = (\hat{f}_n)_{n=1,\ldots,N}$ are vectors of the functions f_n and

$x_n = f_n(x_{\mathrm{pa}(n)}, e_n)$ for each n. This can be seen to be true by explicitly writing each x_n as a function of the e_n by substituting the equations into one another.

[3] This assumption could be relaxed to a more Bayesian approach involving a prior over the covariance matrix, which would reduce to running the algorithm derived in the following, but averaging its results over the posterior.

[4] The same is true in the case of causal graph learning. See e. g. de Jongh and Druzdzel [2009].

$\hat{f}_n$, respectively.

$$L_n(\hat{f}_n||f_n) = \int_{\mathcal{X}_{\text{pa}(n)}} \left(f_n(x) - \hat{f}_n(x)\right)^2 d\Pi_n(x),$$

$$L(\hat{f}||f) = \sum_{n=1}^{N} \alpha_n L_n(\hat{f}_n||f_n), \quad \alpha_n \geq 0.$$

We will assume for simplicity that $\alpha_n = 1$ for each n. L_n is also known as the Mean Integrated Squared Error [Tsybakov, 2009] and in the case that $\Pi_n = \mathbb{P}^{\text{do}(\varnothing)}_{X_{\text{pa}(n)}}$, this coincides with a typical objective that would be minimised in a classical non-parametric statistical learning setting [Györfi et al., 2006]. It is worth considering other possible risk functionals that could be used, since the presence of the measures Π_n is arguably somewhat arbitrary in the L that we consider. When learning the parameters of a statistical model, a commonly used objective with many separate justifications is to minimise the KL divergence $KL[\mathbb{P}_X||\hat{\mathbb{P}}_X]$ between the true data distribution $\mathbb{P}_X$ and that implied by the learned model, $\hat{\mathbb{P}}_X$. SCMs do not imply a single distribution over the variables X, but rather a family of distributions, one for each intervention: $\{\mathbb{P}^{\text{do}(i)}_X : i \in \mathcal{I}\}$. One may therefore wish to consider a separate loss for each interventional distribution and, for example, uniformly bound these losses over a subset of interventions $\mathcal{I}' \subseteq \mathcal{I}$ of interest.

$$L^i_{KL}(\hat{f}||f) = KL\left[\mathbb{P}^{\text{do}(i)}_X||\hat{\mathbb{P}}^{\text{do}(i)}_X\right],$$

$$L^{\mathcal{I}'}_{KL}(\hat{f}||f) = \sup_{i\in\mathcal{I}'} L^i_{KL}(\hat{f}||f).$$

Alternatively, one could replace the KL divergence with a different divergence measure or metric on distributions. For instance, one could use the Maximum Mean Discrepancy (MMD) corresponding to a characteristic kernel l Sriperumbudur et al. [2008]

$$L^i_{\text{MMD}_l}(\hat{f}||f) = \text{MMD}_l\left[\mathbb{P}^{\text{do}(i)}_X||\hat{\mathbb{P}}^{\text{do}(i)}_X\right],$$

$$L^{\mathcal{I}'}_{\text{MMD}_l}(\hat{f}||f) = \sup_{i\in\mathcal{I}'} L^i_{\text{MMD}_l}(\hat{f}||f).$$

Though we do not analyse or derive active learning schemes for these risk functionals, they will be used in Section 6 to evaluate our algorithm. We leave their consideration for future work.

4 A probabilistic approach to learning f

By taking a Bayesian approach to learning the vector of unknown functions f, Problem 1 can be reduced to a series of independent regression problems between input and output domains $\mathcal{X}_{\text{pa}(n)}$ and $\mathcal{X}_n$ for each n. A common choice of prior when learning functions is a Gaussian Process (GP) [Rasmussen and Williams, 2005]. For each function f_n, we will assume a zero mean GP prior with kernel k_n over the domain $\mathcal{X}_{\text{pa}(n)}$[5]

$$f_n \sim \mathcal{GP}(0, k_n).$$

Recall that we are given a dataset $\mathcal{D}$ consisting of elements (i, x) where $x \sim \mathbb{P}^{\text{do}(i)}_X$. Let $\mathcal{D}_n$ be the collection of marginal observations of $(X_{\text{pa}(n)}, X_n)$ drawn from any distribution in which X_n is not intervened upon.[6] Since by assumption $X_n \sim f_n(X_{\text{pa}(n)}) + E_n$ where the distribution of $E_n \sim \mathcal{N}(0, \sigma^2_n)$ is known, each element $(x_{\text{pa}(n)}, x_n)$ of $\mathcal{D}_n$ represents an evaluation of f_n at the input point $x_{\text{pa}(n)}$ corrupted by Gaussian noise of known variance. Performing GP regression using $\mathcal{D}_n$ as the data gives the posterior distribution over f_n. By properties of Gaussians, this is also a GP with distribution

$$f_n|\mathcal{D}_n \sim \mathcal{GP}(\mu_{f_n|\mathcal{D}_n}, k_{f_n|\mathcal{D}_n}),$$

where $\mu_{f_n|\mathcal{D}_n}$ and $k_{f_n|\mathcal{D}_n}(x, y)$ can be explicitly written in terms of k_n and the data $\mathcal{D}_n$ (see Appendix for details). The above procedure can be applied for each f_n independently, giving a posterior distribution over the vector of functions f.

Which $\hat{f}$ should be chosen, given the posterior over f? Problem 1 demands that a single choice $\hat{f}$ be made when making the estimation $\widehat{\mathcal{M}}$ of $\mathcal{M}$. For a fixed $\hat{f}$, the total risk $L(\hat{f}||f)$ is a random variable (the randomness coming from the uncertain belief over f). The expectation of this random variable can be calculated and expressed in terms of the posterior covariance and mean functions of each f_n.

Lemma 1.

$$\mathbb{E}_{f|\mathcal{D}}\left[L(\hat{f}||f)\right] = \sum_{n=1}^{N} \alpha_n \int_{\mathcal{X}_{pa(n)}} \left(\hat{f}_n(x) - \mu_{f_n|\mathcal{D}_n}(x)\right)^2 + k_{n|\mathcal{D}_n}(x, x) d\Pi_n(x)$$

See Appendix for proof. This immediately implies the following result.

Lemma 2. *Let $\mu_{f|\mathcal{D}}$ be the tuple of functions $(\mu_{f_n|\mathcal{D}_n})_{n=1,\ldots,N}$. Then*

$$\mu_{f|\mathcal{D}} = \arg\min_{\hat{f}} \mathbb{E}_{f|\mathcal{D}}\left[L(\hat{f}||f)\right]$$

That is, choosing $\hat{f}_n$ to be the posterior mean of f_n for each n minimises the expected total risk.

[5] No specific assumptions will be made on the choice of k_n, which can be freely chosen to incorporate prior knowledge about the functions (for instance, typical length scale of variation and magnitude). If X_n is parentless, f_n is an unknown *constant* for which we assume a 1-dimensional Gaussian prior with zero mean and variance k_n.

[6] That is, for any distribution $\mathbb{P}^{\text{do}(i)}_X$ for which $X_n \notin \text{var}(i)$.

The benefit of taking a Bayesian approach to learning f is that it directly yields an estimate of the total risk once the optimal $\hat{f} = \mu_{f|\mathcal{D}}$ is chosen which, once the prior has been fixed, is purely a function of the data $\mathcal{D}$. Denote by $\mathcal{R}(\mathcal{D}) = \mathbb{E}_{f|\mathcal{D}}\left[L(\mu_{f|\mathcal{D}}||f)\right]$ this *expected total risk*. Choosing the intervention i for which $\mathcal{R}(\mathcal{D} \cup \{(i,x)\})$ is expected to be smallest after making the new observation x from $\mathbb{P}_X^{\mathrm{do}(i)}$ forms the basis of the proposed active learning algorithm.

5 Active learning

In this section a myopic active learning algorithm is derived based on the GP belief of the functions f_n and the expected total risk $\mathcal{R}(\mathcal{D})$ described above. At each step in time, we select an intervention i at cost $c(i)$ and observe a single draw from the distribution $\mathbb{P}_X^{\mathrm{do}(i)}$. The goal is to select the intervention $i \in \mathcal{I}$ which will reduce the expected total risk as much as possible, taking into account the cost $c(i)$. This problem is non-trivial for two main reasons.

1. The true distributions $\mathbb{P}_X^{\mathrm{do}(i)}$ are unknown and therefore it is not possible to calculate the true expected reduction in expected total risk given a proposed intervention.
2. There is a potentially large set of interventions that must be searched over.

Consider the first issue above. How will the expected total risk change if the intervention i is chosen and a single new observation is drawn from $\mathbb{P}_X^{\mathrm{do}(i)}$? If the new observation is $x \in \mathcal{X}$, the new expected total risk will be $\mathcal{R}(\mathcal{D} \cup \{(i,x)\})$. Define the *value* of the intervention i to be the expected reduction of $\mathcal{R}$ after performing the intervention i, divided by the cost of i:

$$V(i|\mathcal{D}) = \frac{\mathcal{R}(\mathcal{D}) - \mathbb{E}_{x\sim\mathbb{P}_X^{\mathrm{do}(i)}}\mathcal{R}\left(\mathcal{D} \cup \{(i,x)\}\right)}{c(i)} \qquad (\star)$$

The goal is to find the intervention with the largest value, but since $\mathbb{P}_X^{\mathrm{do}(i)}$ is unknown it is not possible to calculate the right-hand term in the numerator of $(\star)$. It is possible, however, to *estimate* this by replacing $\mathbb{P}_X^{\mathrm{do}(i)}$ in the expectation with the *belief* of the distribution based on the uncertain estimates of each f_n. In this next two parts of this section, two different methods are proposed that estimate the expected total risk after performing each intervention. The first uses sampling, and requires a brute-force search over the set of interventions. This may be appropriate when the set of possible interventions is small enough that this is feasible. The second uses a form of dynamic programming, enabling a search over a larger set of interventions more efficiently. The derived algorithm, however, makes specific assumptions on the graph of $\mathcal{M}$ and the set of interventions.

Algorithm 1 Sampling to estimate expected risk after intervention

1: **Input:** Previously observed data $\mathcal{D}$, GP kernels k_n for prior on f_n, number of samples T, proposed intervention i.
2: **for** $t = 1, \ldots, T$ **do**
3: Draw x^t from predictive distribution $\widehat{\mathbb{P}}_X^{\mathrm{do}(i)}$
4: $s_t = \mathcal{R}(\mathcal{D} \cup \{(i, x^t)\})$: expected loss given new x^t
5: **end for**
6: **return** $\frac{1}{T}\sum_{t=1}^{T} s_t$: estimated expected loss after intervention i.

5.1 Sampling

Write $\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}$ for the belief of $\mathbb{P}_X^{\mathrm{do}(i)}$, taking into account the full uncertainty over f.[7] Since the belief over each f_n at each input $x_{\mathrm{pa}(n)} \in \mathcal{X}_{\mathrm{pa}(n)}$ is Gaussian and the noise variables are additive and Gaussian, it is possible to efficiently sample from $\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}$. It is illustrated here how to draw from the estimated observational distribution $\widetilde{\mathbb{P}}_X^{\mathrm{do}(\varnothing)}$ for notational convenience, but the procedure for any other $\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}$ is essentially the same.

For any parentless variable, the structural equation is $X_n = f_n + E_n$ where $f_n \sim \mathcal{N}(\mu_{n|\mathcal{D}_n}, k_{n|\mathcal{D}_n})$ and $E_n \sim \mathcal{N}(0, \sigma_n^2)$, and therefore $X_n \sim \mathcal{N}(\mu_{n|\mathcal{D}_n}, k_{n|\mathcal{D}_n} + \sigma_n^2)$.

For any variable with parents, the structural equation is $X_n = f_n(X_{\mathrm{pa}(n)}) + E_n$ where $f_n \sim \mathcal{GP}(\mu_{n|\mathcal{D}_n}, k_{n|\mathcal{D}_n})$ and $E_n \sim \mathcal{N}(0, \sigma_n^2)$. Therefore the conditional distribution of a variable given its parents is $X_n|X_{\mathrm{pa}(n)} \sim \mathcal{N}(\mu_{n|\mathcal{D}_n}(X_{\mathrm{pa}(n)}), k_{n|\mathcal{D}_n}(X_{\mathrm{pa}(n)}, X_{\mathrm{pa}(n)}) + \sigma_n^2)$. Observe that the joint distribution $\widetilde{\mathbb{P}}_X^{\mathrm{do}(\varnothing)}$ factorises as

$$\widetilde{\mathbb{P}}_X^{\mathrm{do}(\varnothing)} = \prod_{n:\mathrm{pa}(n)=\emptyset} \widetilde{\mathbb{P}}_{X_n}^{\mathrm{do}(\varnothing)} \prod_{n:\mathrm{pa}(n)\neq\emptyset} \widetilde{\mathbb{P}}_{X_n|X_{\mathrm{pa}(n)}}^{\mathrm{do}(\varnothing)}. \qquad (*)$$

Since drawing from each of the above factors amounts to drawing from a Gaussian distribution, it is possible to efficiently sample from the entire joint distribution. By replacing $\mathbb{P}_X^{\mathrm{do}(i)}$ with $\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}$ in Equation $(\star)$ above, we arrive at an estimate of the expected total risk which can be estimated using samples drawn from $\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}$:

$$\mathbb{E}_{x\sim\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\mathcal{R}(\mathcal{D} \cup \{(i,x)\}) \approx \frac{1}{T}\sum_{t=1}^{T} \mathcal{R}(\mathcal{D} \cup \{(i,x^t)\})$$

where each $x^t \sim \widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}$ (see Algorithm 1). Finding the optimal intervention hence reduces to computing the above quantity for each $i \in \mathcal{I}$ from which it is possible to estimate each $V(i|\mathcal{D})$.

[7] In contrast to $\widehat{\mathbb{P}}_X^{\mathrm{do}(i)}$, which is the estimated distribution once a particular choice for $\hat{f}$ is made.

Algorithm 2 Dynamic programming to estimate expected risk after interventions (chain, interventions on all $X_{n\leq m}$, some m)

Input: Previously observed data $\mathcal{D}$, GP kernels k_n, discretisations $\widehat{\mathcal{X}}_n$ of each $\mathcal{X}_n$.
Pre-compute U_n vectors and discrete approximations to conditional probability distributions:
for $n = 1, \ldots, N$ **do**
    If $n = 1$: $P^1_{x_1} \propto P(x_1)$ for $x_1 \in \widehat{\mathcal{X}}_1$, else:
    $P^n_{x_{n-1},x_n} \propto P(x_n|x_{n-1})$ for $x_{n-1} \in \widehat{\mathcal{X}}_{n-1}, x_n \in \widehat{\mathcal{X}}_n$
    $U_n^{\text{curr}} = \mathcal{R}_n(\mathcal{D}_n)$
    $U_n(x_{n-1})$ for $x_{n-1} \in \widehat{\mathcal{X}}_{n-1}$
end for
Calculate new expected risk for all interventions:
for $n = 1, \ldots, N$ **do**
    $V = 0$
    for $m = N-1, \ldots, n+1$ **do**
        $V = P^m(V + U_{m+1})$
    end for
    Expected risk after intervention on variables $X_m, m \leq n$:
    $ER_n = V + U_1^{\text{old}} + \ldots + U_n^{\text{old}}$
end for
return Vectors ER_n giving estimated expected risks after all interventions $\mathrm{do}(X_n = x_n, X_{n-1} = \ldots)$

5.2 Dynamic programming

If the set of interventions under consideration exhibits structure that coincides with that of the causal graph appropriately, it is possible to estimate the value of many interventions simultaneously. A specific example is provided here of how this can be done in the case that the causal graph is a chain $X_1 \to \ldots \to X_N$, and any intervention intervenes on one variable and everything upstream of it.[8] A similar example for chains in which all interventions intervene on exactly one variable is provided in the Appendix.

The crux of this approach is the fact that the posterior covariance function of a Gaussian Process is only a function of the inputs of the conditioning data, not of the outputs. That is, writing $\mathcal{R}_n(\mathcal{D}_n) = \mathbb{E}_{f_n|\mathcal{D}_n}[L_n(\mu_{f_n|\mathcal{D}_n}, f_n)] = \int_{\mathcal{X}_{\text{pa}(n)}} k_{n|\mathcal{D}_n}(x,x)d\Pi_n(x)$ for the contribution to the expected total risk due to estimating function f_n, and writing $\mathcal{D}_n = \{(x^s_{\text{pa}(n)}, x^s_n) : s = 1 \ldots, |\mathcal{D}_n|\}$, it follows that $\mathcal{R}_n$ is only actually a function of the $x^s_{pa(n)}$ (or, for parentless variables, just the size of the dataset $|\mathcal{D}_n|$). Consider the intervention $i = \mathrm{do}(X_m = x_m, X_{m-1} = \ldots)$ that sets $X_m = x_m$ and all variables upstream of X_m to arbitrary values. When a new observation $x \sim \mathbb{P}_X^{\mathrm{do}(i)}$ is made, this only provides new information about the functions f_n for $n > m$, since i intervenes on X_n for $n \leq m$. Let $U_n^{\text{curr}} = \mathcal{R}_n(\mathcal{D}_n)$ for $n \leq m$ be the current contributions to the expected total risk. Define, for $n > m$, the following shorthand for the contribution to the new expected total risk function made by f_n if a new observation of f_n at the input point x_{n-1} is made, for any value x_n, $n > m$:

$$U_n(x_{n-1}) = \mathcal{R}_n(\mathcal{D}_n \cup \{(x_{n-1}, x_n)\})$$

It follows that the estimated expected total risk after performing the intervention i decomposes thus:

$$\mathbb{E}_{x\sim\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\left[\mathcal{R}(\mathcal{D}\cup\{(i,x)\})\right]$$
$$= \sum_{n=1}^{m} U_n^{\text{curr}} + \mathbb{E}_{x\sim\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\left[\sum_{n=m+1}^{N} U_n(x_{n-1})\right]. \qquad (\dagger)$$

By exploiting a factorisation of $\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}$ similar to $(*)$ and discretely approximating the continuous domains $\mathcal{X}_n$, it is possible to reduce evaluating the right hand side of (†) to a series of matrix multiplications and additions (see Appendix for details). This series of operations can be vectorised to allow calculation of (†) for many x_m simultaneously. Moreover, many of the initial calculations can be cached and used to speed up calculation over different values of m. See Algorithm 2.

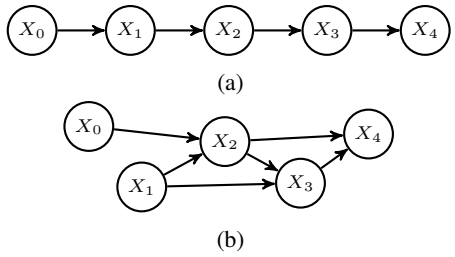


Figure 2: The causal graphs of the SCMs (a) $\mathcal{M}_1$ and (b) $\mathcal{M}_2$ used in the experiments (Section 6).

6 Experiments

Figure 3 shows the results of running the proposed methods on two synthetic example SCMs $\mathcal{M}_1$ and $\mathcal{M}_2$, for which the graphs are given by Figures 2a and 2b respectively. In each case, structural equations consisting of sines and cosines of the parent variables were fixed, giving two sets of structural equations $\mathcal{S}_1$ and $\mathcal{S}_2$. All noise variables in both SCMs were fixed to have variance 0.1. The interventions $\mathcal{I}_1$ for the chain SCM $\mathcal{M}_1$ were chosen to be all interventions of the form $\mathrm{do}(X_m = x_m, X_{m-1} = \ldots)$ with $x_m \in [-6, 6]$, such that $\mathcal{M}_1$ satisfies the conditions set out in the description of the dynamic programming algorithm. The interventions $\mathcal{I}_2$ for the non-chain SCM $\mathcal{M}_2$ were defined to be all interventions on single variables $\mathrm{do}(X_n = x_n)$ with $x_n \in [-6, 6]$. The total risk function for each experiment was chosen by setting Π_n to

[8] That is, any intervention is of the form $\mathrm{do}(X_n = x_n,\ n \leq m)$ for some m. This is equivalent to the case that all interventions act on a single variable, but only variables downstream of this are observable.

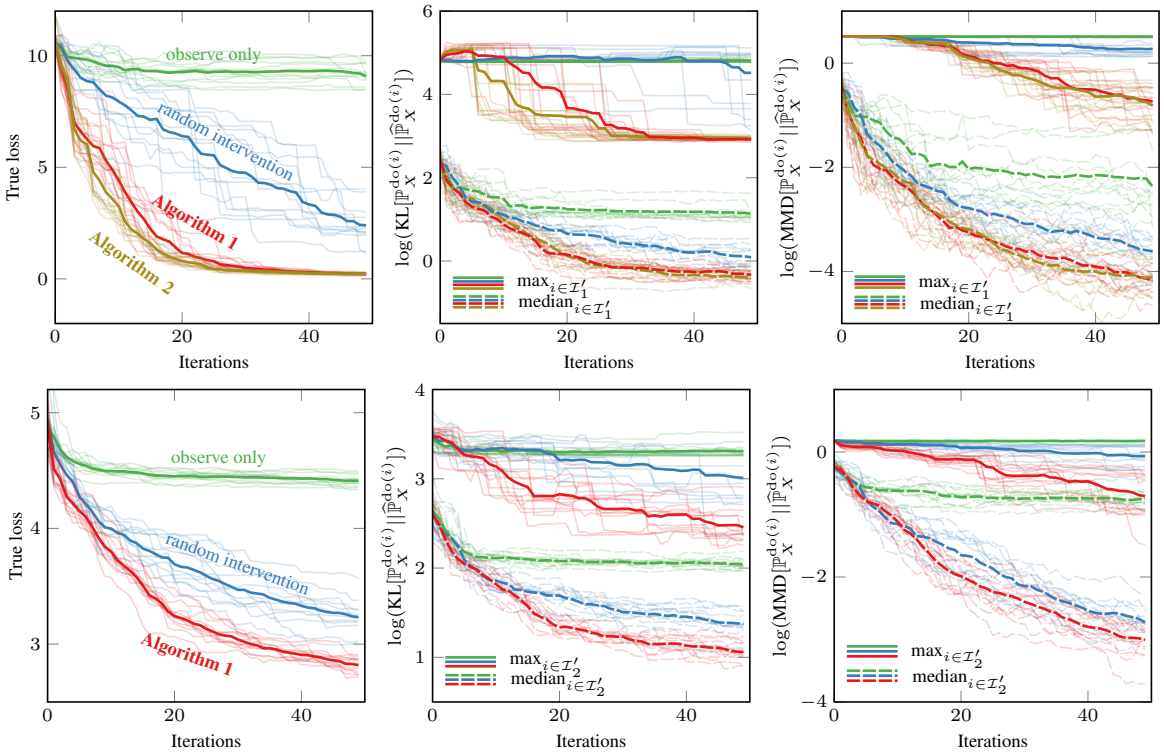


Figure 3: **Actively choosing informative interventions speeds up learning**. The proposed methods (**Algorithm 1**, **Algorithm 2**) outperform **only observing** and **randomly intervening** by each metric of evaluation. Top and bottom row show results of experiments on learning $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively. Experimental details described in Section 6. Each experiment was performed many times; faded lines represent results from single trials, bold lines represent averages of these single trials.

be the uniform distribution on the domain $[-6, 6]^{|\operatorname{pa}(n)|}$ and $\alpha_n = 1$ for each n. All GP kernels used were Radial Basis Function kernels with bandwidth parameter 1. All costs $c(i)$ were assumed to be equal.

In each experiment, learning was initiated with no data. For learning $\mathcal{M}_1$, the proposed algorithms were compared against the strategies of only drawing from the observational distribution of $\mathcal{M}_1$ and of selecting an intervention uniformly at random. Since the dynamic programming algorithm could not be used for learning $\mathcal{M}_2$, we could only test the sampling algorithm. Uniform discretisations $\mathcal{I}'_1$ and $\mathcal{I}'_2$ of the intervention sets were made of total size 250 each. These were used as the sets of interventions to search over using the sampling algorithm.

The following quantities were used to evaluate the performance of the algorithms at each point in time: the true total risk incurred by choosing $\hat{f}$ to be the vector of GP posterior means; the maximum and median of the set of KL divergences $KL[\mathbb{P}_X^{\mathrm{do}(i)} || \widehat{\mathbb{P}}_X^{\mathrm{do}(i)}]$ calculated for each intervention $i \in \mathcal{I}'_*$ in the discretised intervention sets; the maximum and median of the set of MMDs (corresponding to a Radial Basis Function kernel l with bandwidth 1) $\mathrm{MMD}_l[\mathbb{P}_X^{\mathrm{do}(i)} || \widehat{\mathbb{P}}_X^{\mathrm{do}(i)}]$ calculated for each intervention $i \in \mathcal{I}'_*$ in the discretised intervention sets.

7 Discussion and future directions

There are many ways in which the work presented here could be incrementally furthered: it may be possible to find efficient ways to search over the set of interventions subject to less restrictive assumptions than those made for Algorithm 2; different distributions over the noise variables could be considered, in which case an approximation may need to be made when regressing to find the posterior distribution over each f_n; one could try to relax the additive noise assumption altogether. Other natural extensions include estimating the value of an intervention based on reasoning multiple steps into the future, or considering the implications of a constrained budget.

Although the proposed algorithms seem to perform well on the synthetic toy examples considered, it remains to be seen whether this method, suitably extended, would similarly perform well on a convincing real-world problem. A perhaps more fundamental issue that was raised and *not* tackled is the fact that it is not clear how best one should even define what it means to learn an SCM, or the parameters thereof. We proposed $\sup_{i \in \mathcal{I}'} L^i_{KL}(\hat{f} || f)$ and $\sup_{i \in \mathcal{I}'} L^i_{\mathrm{MMD}_l}(\hat{f} || f)$ for a suitable set of interventions $\mathcal{I}'$ as potentially more principled objectives to minimise. Interestingly, the derived algorithms do reduce these quantities in the experiments considered, though this was in no way an explicit objective. Future directions of research include trying to understand whether these, or other objectives, give rise to tractable methods for parameter estimation in causal models and for selecting interventions in active settings, and under what assumptions mathematical guarantees can be made.

References

K. A. Bollen. *Structural equations with latent variables*. John Wiley & Sons, 2014.

M. de Jongh and M. J. Druzdzel. A comparison of structural distance measures for causal Bayesian network models. *Recent Advances in Intelligent Information Systems, Challenging Problems of Science, Computer Science series*, pages 443–456, 2009.

F. Eberhardt. Causal discovery as a game. In *NIPS Causality: Objectives and Assessment*, pages 87–96, 2010.

L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

A. Hyttinen, F. Eberhardt, and P. O. Hoyer. Experiment selection for causal discovery. *Journal of Machine Learning Research*, 14(1):3041–3071, 2013.

J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research*, 2016.

M. A. Osborne, R. Garnett, and S. J. Roberts. Gaussian processes for global optimization. In *in LION*. Citeseer, 2009.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, 2nd edition, 2009.

J. Peters, J. Mooij, D. Janzing, and B. Schölkopf. Identifiability of causal graphs using functional models. In F. Cozman and A. Pfeffer, editors, *27th Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, pages 589–598, Corvallis, OR, USA, 2011. AUAI Press.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf. Injective Hilbert Space Embeddings of Probability Measures. In R. Servedio and T. Zhang, editors, *Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008)*, pages 111–122, Madison, WI, USA, 2008. Omnipress.

S. Tong and D. Koller. Active learning for parameter estimation in Bayesian networks. In *NIPS*, volume 13, pages 647–653, 2000.

S. Tong and D. Koller. Active learning for structure in Bayesian networks. In *International joint conference on artificial intelligence*, volume 17, pages 863–869. Lawrence Erlbaum Associates LDT, 2001.

A. B. Tsybakov. Introduction to nonparametric estimation. Revised and extended from the 2004 French original. Translated by Vladimir Zaiats, 2009.

A Appendix

A.1 Posterior distribution of f_n given $\mathcal{D}_n$

The prior over f_n is a Gaussian Process with zero mean and covariance function k_n

$$f_n \sim \mathcal{GP}(0, k_n).$$

Let $\mathcal{D}_n = \{(x^s_{\text{pa}(n)}, x^s_n) \,:\, s = 1 \ldots, |\mathcal{D}_n|\}$ and let K be the matrix with entries

$$K_{st} = k_i(x^s_{\text{pa}(n)}, x^t_{\text{pa}(n)}).$$

Suppose that the Gaussian noise variable E_n has variance σ_n^2. Then the posterior mean function $\mu_{f_n|\mathcal{D}_n}$ and posterior covariance function $k_{f_n|\mathcal{D}_n}$ can be written in closed form as

$$\mu_{f_n|\mathcal{D}_n}(x) = \sum_{s,t=1}^{|\mathcal{D}_n|} k_n(x, x^s_{\text{pa}(n)})(K + \sigma_n^2)^{-1}_{st} x^t_n,$$

$$k_{f_n|\mathcal{D}_n}(x, y) = k_n(x, y) - \sum_{s,t=1}^{|\mathcal{D}_n|} k_n(x, x^s_{\text{pa}(n)})(K + \sigma_n^2 I)^{-1}_{st} k_n(y, x^t_{\text{pa}(n)}).$$

A.2 Proof of Lemma 1

Proof. The expected total risk can be written in full form as

$$\begin{aligned}
\mathbb{E}_{f|\mathcal{D}}\left[L(\hat{f}||f)\right] &= \mathbb{E}_{f|\mathcal{D}}\left[\sum_{n=1}^{N} \alpha_n \int_{\mathcal{X}_{pa(n)}} \left(\hat{f}_n(x) - f_n(x)\right)^2 d\Pi_n(x)\right] \\
&= \sum_{n=1}^{N} \alpha_n \int_{\mathcal{X}_{pa(n)}} \mathbb{E}_{f|\mathcal{D}}\left[\left(\hat{f}_n(x) - f_n(x)\right)^2\right] d\Pi_n(x) \\
&= \sum_{n=1}^{N} \alpha_n \int_{\mathcal{X}_{pa(n)}} \mathbb{E}_{f_n|\mathcal{D}_n}\left[\left(\hat{f}_n(x) - f_n(x)\right)^2\right] d\Pi_n(x).
\end{aligned}$$

Under the posterior distribution given $\mathcal{D}_n$, we can decompose f_n as the sum of its posterior mean and a zero mean Gaussian Process:

$$f_n|\mathcal{D}_n = \mu_{f_n|\mathcal{D}_n} + g_n, \qquad g_n \sim \mathcal{GP}(0, k_{f_n|\mathcal{D}_n}).$$

We can therefore rewrite the expected total risk as

$$\begin{aligned}
&\mathbb{E}_{f|\mathcal{D}}\left[L(\hat{f}||f)\right] = \sum_{n=1}^{N} \alpha_n \int_{\mathcal{X}_{pa(n)}} \mathbb{E}_{g_n|\mathcal{D}_n}\left[\left(\hat{f}_n(x) - \mu_{f_n|\mathcal{D}_n}(x) - g_n(x)\right)^2\right] d\Pi_n(x) \\
&= \sum_{n=1}^{N} \alpha_n \int_{\mathcal{X}_{pa(n)}} \mathbb{E}_{g_n|\mathcal{D}_n}\left[\left(\hat{f}_n(x) - \mu_{f_n|\mathcal{D}_n}(x)\right)^2 - \left(\hat{f}_n(x) - \mu_{f_n|\mathcal{D}_n}(x)\right) g_n(x) + g_n(x)^2\right] d\Pi_n(x) \\
&= \sum_{n=1}^{N} \alpha_n \int_{\mathcal{X}_{pa(n)}} \left(\hat{f}_n(x) - \mu_{f_n|\mathcal{D}_n}(x)\right)^2 - \left(\hat{f}_n(x) - \mu_{f_n|\mathcal{D}_n}(x)\right) \mathbb{E}_{g_n|\mathcal{D}_n}\left[g_n(x)\right] + \mathbb{E}_{g_n|\mathcal{D}_n}\left[g_n(x)^2\right] d\Pi_n(x) \\
&= \sum_{n=1}^{N} \alpha_n \int_{\mathcal{X}_{pa(n)}} \left(\hat{f}_n(x) - \mu_{f_n|\mathcal{D}_n}(x)\right)^2 + k_{f_n|\mathcal{D}_n}(x, x) d\Pi_n(x).
\end{aligned}$$

□

A.3 Derivation of Algorithm 2

Under the intervention $i = \mathrm{do}(X_m = x_m^*, X_{m-1} = \ldots)$, the distribution $\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}$ factorises as:

$$\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)} = \prod_{n \leq m} \delta_{X_n = x_n^*} \prod_{n \geq m+1} \widetilde{\mathbb{P}}_{X_n | X_{n-1}},$$

where

$$\begin{aligned} X_n | X_{n-1} &\sim \mathcal{N}\left(f_n(X_{n-1}), \sigma_n^2\right) \\ &\sim \mathcal{N}\left(\mu_{f_n|\mathcal{D}_n}(X_{n-1}), k_{f_n|\mathcal{D}_n}(X_{n-1}, X_{n-1}) + \sigma_n^2\right). \end{aligned}$$

The quantity we are trying to calculate can hence be written

$$\begin{aligned} \mathbb{E}_{x \sim \widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\left[\mathcal{R}(\mathcal{D} \cup \{(i,x)\})\right] &= \sum_{n=1}^{m} U_n^{\text{curr}} + \mathbb{E}_{x \sim \widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\left[\sum_{n=m+1}^{N} U_n(x_{n-1})\right] \\ &= \sum_{n=1}^{m} U_n^{\text{curr}} + \int_{\mathcal{X}} \sum_{n=m+1}^{N} U_n(x_{n-1})\, d\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}(x) \\ &= \sum_{n=1}^{m} U_n^{\text{curr}} + \sum_{n=m}^{N-1} \int_{\mathcal{X}_n} U_{n+1}(x_n)\, d\widetilde{\mathbb{P}}_{X_n}^{\mathrm{do}(i)}(x_n). \end{aligned}$$

Now, since $\widetilde{\mathbb{P}}_{X_m}^{\mathrm{do}(i)} = \delta_{X_m = x_m^*}$, observe that we can write

$$\int_{\mathcal{X}_m} U_{m+1}(x_m)\, d\widetilde{\mathbb{P}}_{X_m}^{\mathrm{do}(i)}(x_m) = U_{m+1}(x_m^*).$$

For any $n > m$, we have

$$\begin{aligned} &\int_{\mathcal{X}_n} U_{n+1}(x_n)\, d\widetilde{\mathbb{P}}_{X_n}^{\mathrm{do}(i)}(x_n) \\ &\qquad = \int_{\mathcal{X}_m} \ldots \int_{\mathcal{X}_n} U_{n+1}(x_n)\, d\widetilde{\mathbb{P}}_{X_n|X_{n-1}}^{\mathrm{do}(i)}(x_n|x_{n-1}) \ldots d\widetilde{\mathbb{P}}_{X_{m+1}|X_m}^{\mathrm{do}(i)}(x_{m+1}|x_m) d\widetilde{\mathbb{P}}_{X_m}^{\mathrm{do}(i)}(x_m). \end{aligned}$$

Hence if we define the following quantities recursively

$$\begin{aligned} V_{N-1}(x_{N-2}) &= \int_{\mathcal{X}_{N-1}} U_N(x_{N-1})\, d\widetilde{\mathbb{P}}_{X_{N-1}|X_{N-2}}^{\mathrm{do}(i)}(x_{N-1}|x_{N-2}), \\ V_n(x_{n-1}) &= \int_{\mathcal{X}_n} V_{n+1}(x_n) + U_{n+1}(x_n)\, d\widetilde{\mathbb{P}}_{X_n|X_{n-1}}^{\mathrm{do}(i)}(x_n|x_{n-1}) \qquad n = N-2, \ldots m+1, \end{aligned}$$

it follows that

$$V_{m+1}(x_m^*) = \sum_{n=m}^{N-1} \int_{\mathcal{X}_n} U_{n+1}(x_n)\, d\widetilde{\mathbb{P}}_{X_n}^{\mathrm{do}(i)}(x_n).$$

We can approximate calculation of $V_{m+1}(x_m^*)$ for many x_m^* simultaneously by discretising each $\mathcal{X}_n$ into a set of points $x_n^1, x_n^2, \ldots, x_n^{D_n}$. Define P^n to be the matrix with normalised rows such that $P_{ij}^n \propto \widetilde{p}_{X_n|X_{n-1}}^{\mathrm{do}(i)}(x_n^j | x_{n-1}^i)$, where $\widetilde{p}^{\mathrm{do}(i)}$ is the density of $\widetilde{\mathbb{P}}^{\mathrm{do}(i)}$ with respect to the Lebesgue measure on $\mathcal{X}$. Define $\mathbf{u}^n$ to be the vector with entries $\mathbf{u}_i^n = U_n(x_{n-1}^i)$. Then, if we recursively define

$$\begin{aligned} \mathbf{v}^{N-1} &= P^{N-1}\mathbf{u}_N, \\ \mathbf{v}^n &= P^n(\mathbf{v}^{n+1} + \mathbf{u}_{n+1}) \qquad n = N-2, \ldots m+1, \end{aligned}$$

it follows that $\mathbf{v}_i^{m+1} \approx V_{m+1}(x_m^i)$.

A.4 Another Dynamic Programming Algorithm: chain, single variable interventions.

Similar reasoning to the above can be used to derive a dynamic programming scheme to calculate the estimated total risk after a proposed intervention $i = \mathrm{do}(X_m = x_m^*)$ (i. e. intervening on a *single* variable) for many x_m^* simultaneously. This is summarised by Algorithm 3.

Under this intervention, the joint distribution factorises as

$$\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)} = \widetilde{\mathbb{P}}_{X_1} \prod_{n=2}^{m-1} \widetilde{\mathbb{P}}_{X_n|X_{n-1}} \delta_{X_m = x_m^*} \prod_{n=m+1}^{N} \widetilde{\mathbb{P}}_{X_n|X_{n-1}}.$$

When we intervene on a single variable X_m, we learn something new about all functions except f_m. We can write the expected new total risk, the quantity we want to evaluate, as

$$\mathbb{E}_{x\sim\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\left[\mathcal{R}(\mathcal{D}\cup\{(i,x)\})\right] = U_1 + \mathbb{E}_{x\sim\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\left[\sum_{n=2}^{m-1} U_n(x_{n-1})\right] + U_m^{\text{curr}} + \mathbb{E}_{x\sim\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\left[\sum_{n=m+1}^{N} U_n(x_{n-1})\right].$$

Each of the expectations above can be calculated recursively in a similar fashion to the strategy employed above. If we define

$$\begin{aligned}
V_{N-1}(x_{N-2}) &= \int_{\mathcal{X}_{N-1}} U_N(x_{N-1})\, d\widetilde{\mathbb{P}}^{\mathrm{do}(i)}_{X_{N-1}|X_{N-2}}(x_{N-1}|x_{N-2}),\\
V_n(x_{n-1}) &= \int_{\mathcal{X}_n} V_{n+1}(x_n) + U_{n+1}(x_n)\, d\widetilde{\mathbb{P}}^{\mathrm{do}(i)}_{X_n|X_{n-1}}(x_n|x_{n-1}), \qquad n = N-2,\dots m+1,
\end{aligned}$$

it follows that

$$V_{m+1}(x_m^*) = \mathbb{E}_{x\sim\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\left[\sum_{n=m+1}^{N} U_n(x_{n-1})\right].$$

Similarly, defining

$$\begin{aligned}
V_{m-2}(x_{m-3}) &= \int_{\mathcal{X}_{m-2}} U_{m-1}(x_{m-2})\, d\widetilde{\mathbb{P}}^{\mathrm{do}(i)}_{X_{m-2}|X_{m-3}}(x_{m-2}|x_{m-3}),\\
V_n(x_{n-1}) &= \int_{\mathcal{X}_n} V_{n+1}(x_n) + U_{n+1}(x_n)\, d\widetilde{\mathbb{P}}^{\mathrm{do}(i)}_{X_n|X_{n-1}}(x_n|x_{n-1}), \qquad n = m-3,\dots 2,\\
V_1 &= \int_{\mathcal{X}_1} V_2(x_1) + U_2(x_1)\, d\widetilde{\mathbb{P}}^{\mathrm{do}(i)}_{X_1}(x_1),
\end{aligned}$$

it follows that

$$V_1 = \mathbb{E}_{x\sim\widetilde{\mathbb{P}}_X^{\mathrm{do}(i)}}\left[\sum_{n=2}^{m-1} U_n(x_{n-1})\right].$$

As before, we can approximate calculation of $V_{m+1}(x_m^*)$ for many x_m^* simultaneously by discretising each $\mathcal{X}_n$ into a set of points $x_n^1, x_n^2, \dots, x_n^{D_n}$. Define P^n to be the matrix with normalised rows such that $P^n_{ij} \propto \widetilde{p}^{\mathrm{do}(i)}_{X_n|X_{n-1}}(x_n^j|x_{n-1}^i)$ for $n > 1$, where $\widetilde{p}^{\mathrm{do}(i)}$ is the density of $\widetilde{\mathbb{P}}^{\mathrm{do}(i)}$ with respect to the Lebesgue measure on $\mathcal{X}$. Define P^1 to be the normalised *vector* with $P^1_i \propto \widetilde{p}_{X_1}(x_1^i)$. Define $\mathbf{u}^n$ to be the vector with entries $\mathbf{u}^n_i = U_n(x^i_{n-1})$. Then, if we recursively define

$$\begin{aligned}
\mathbf{v}^{N-1} &= P^{N-1}\mathbf{u}_N,\\
\mathbf{v}^n &= P^n(\mathbf{v}^{n+1} + \mathbf{u}_{n+1}) \qquad n = N-2,\dots m+1,
\end{aligned}$$

it follows that $\mathbf{v}_i^{m+1} \approx V_{m+1}(x_m^i)$. We must also estimate V_1. If we define

$$\begin{aligned}
\mathbf{v}^{m-2} &= P^{m-2}\mathbf{u}_{m-2},\\
\mathbf{v}^n &= P^n(\mathbf{v}^{n+1} + \mathbf{u}_{n+1}) \qquad n = m-3,\dots 2,\\
\mathbf{v}^1 &= P^{1\intercal}(\mathbf{v}^2 + \mathbf{u}_2)
\end{aligned}$$

then it follows that $\mathbf{v}^1 \approx V_1$.

Algorithm 3 Dynamic programming to estimate expected risk after interventions (chain, interventions on single variable X_m)

Input: Previously observed data $\mathcal{D}$, GP kernels k_n, discretisations $\widehat{\mathcal{X}}_n$ of each $\mathcal{X}_n$.
Pre-compute U_n vectors and discrete approximations to conditional probability distributions:
for $n = 1, \ldots, N$ **do**
    $P^n_{x_{n-1}, x_n} \propto P(x_n | x_{n-1})$ for $x_{n-1} \in \widehat{\mathcal{X}}_{n-1}, x_n \in \widehat{\mathcal{X}}_n$
    $U_n^{\text{curr}} = \mathcal{R}_n(\mathcal{D}_n)$
    $U_n(x_{n-1})$ for $x_{n-1} \in \widehat{\mathcal{X}}_{n-1}$
end for
Calculate expected loss for all interventions on each variable in turn:
for $m = 1, \ldots, N$ **do**
    $V = 0$
    for $n = N-1, \ldots, m+1$ **do**
        $V = P^n(V + U_n)$
    end for
    $V' = 0$
    for $n = m-1, \ldots, 2$ **do**
        $V' = P^n(V' + U_n)$
    end for
    Expected risk after intervention on variable m: $ER_m = V + V' + U_1 + U_m^{\text{curr}}$
end for
return Vectors ER_n giving estimated expected risks after interventions for all interventions on X_n.