Interpretability and Causality in Machine Learning for Human and Nonhuman Decision-making

Logan Graham

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2020

Acknowledgements

First and foremost, my deep thanks to my supervisor, Dr. Michael Osborne. He allowed me to be a sort of black sheep in the Bayesian Exploration Lab, coming from an unusual background and pursuing unusual research. I appreciate the opportunities you put before me and your mentorship in my development as a scientist and practitioner. I also thank Dr. Carl Benedikt Frey, for your guidance in the early days of this DPhil. I enjoyed getting to read your thrilling book before it was out.

Thanks to all the leaders, post-docs, and visiting members of the Machine Learning Research Group, for creating a very interesting place to be for several years. Thanks to Dr. Stephen Roberts in particular for challenging philosophical and technical questions. Thanks to Dr. Paul Duckworth – I feel lucky to have collaborated with you during your post-doc.

Thank you to lab colleagues for personal encouragement, stirring up interesting ideas, and lightening the mood when needed. To Dr. Jonathan Downing, for being a consistent companion in good moments and bad ones; Dr. Ali Rizvi for being an early mentor and friend; and to Dr. Justin Bewsher and Dr. Elmarie van Heerden for being friends I continue to look up to. To the rest – there are too many to name - I count myself very lucky to have been there at the same time.

Thank you to those I have worked with outside of the lab that have shaped how I think and what I believe. To Dr. Brad Zamft for investing in me, for teaching me about big ideas, and for attempting something ambitious. Thanks to the rest of the team at X. Thanks to Dr. Ciarán Lee in particular, Yura Perov, and Dr. Saurabh Johri, for your mentorship and collaboration at Babylon; thanks to the rest of the AI Research team. To Dr. Brody Foy, friend and plucky tinkerer; building RAIL with you was a perspective-shaping experience.

To other supporters: the Rhodes Trust for your financial support, and the entire Rhodes community for teaching me what being a good human means; the Oxford-Man Institute for your financial support; to reviewers, summer school friends, and others in the academic community for giving me feedback. To Finn and Bex, whose mark will never leave me. To every friend and family member I have made along the way, and every conversation, big or small, that has assembled the neurons in my brain which now make up my field of consciousness and perception.

Most of all, my thanks and unending love to my family. To my mom, Pam, and dad, John, for knowing me better than I do, and for always being in my corner. To my sister Tookie, for being the best friend I will ever have. I am truly lucky, every day, to live life with you. I will visit and call more.

Abstract

In order to integrate machine learning into human decision-making in a useful way, we must trust machine learning systems enough in our reasoning processes. To evaluate a system's trustworthiness, humans naturally seek interpretable causal systems to understand outcomes, make decisions, and integrate feedback. This thesis presents four explorations into interpretable causal systems, progressing from associational interpretability up the "Ladder of Causation" to counterfactual representation. In the first contribution, I introduce a Bayesian nonparametric method for calculating the expected value and volatility of gradients in the data; this helps inform further experiments for deriving causal effects and interpreting changes when faced with decision-making scenarios. In the second, I show how latent expert knowledge can be collected to produce an interpretable model with policy-relevant consequences. In the third, I interpret time series treatment effect inference as a problem naturally handled by multitask Gaussian processes, addressing several shortcomings in a popular approach. In the fourth, I consider twin networks, a novel but understudied method of representing counterfactual questions; I show that one can sometimes improve on computational cost, error, and insight by representing counterfactual problems as twin networks. I then introduce a prototype probabilistic programming engine that natively leverages the efficiencies gained from twin networks. I close by considering the future of how interpretability and causality might integrate into machine learning-driven decision-making and speculate that causality is a key component in achieving efficient, generalisable intelligence in a variety of applications.

Contents

"Happy is the person who has been able to learn the causes of things."

— Virgil, 1st Century BCE

1

Introduction

1.1 Searching for understanding

What are the effects of a carbon tax on CO_2 per capita? Should I sleep 9 hours or study all night to perform best on the exam? What would have happened to lymphoid enhancer-binding factor 1 expression had I 2×-upregulated glycogen synthase kinase-3 beta?

We ask these questions partly because we're curious, but mostly because we hope the answers will allow us to take actions beneficial to us. We decide on which drug to give a patient, which policy to use to stave off a recession, or which airplane wing

design to fly with. Intelligent beings are actors with desires, making decisions in worlds that help or harm our ability to satisfy our desires. Machines – I use this term to refer to any computational substrate that performs machine learning – are similar. Machines may be tasked to answer one subproblem to a larger problem, but most of the time, the ultimate desire is a guide for how to take optimal actions. Knowing the likely response to an action is desirable if we are going to make wise decisions.

These questions are so useful that we identify our uniquely human intelligence with them. No other organism has yet produced an academia of science, and very few use tools. We might grant that other animals also exhibit curiousity or make plans based on intuitions of cause-and-effect, but humans seem do it much more significantly. In turn, it seems, asking these questions, and having good answers to them, rewards us when it comes to making decisions.

The core problem of this quest is that the universe does not make it easy to acquire this knowledge. There are myriad problems we run into. Some are easy to conquer and some are so pernicious they have consumed the lives of philosophers and scientists for millennia. The universe's data generating process, comprised of all the causal laws that generate the world we live in, is complex. Most of it is hidden to us. We may also not have have complex enough perceptual faculties to even sense the important variables. Or, even more catastrophically, some unknown force – a *malicious demon*, as Descartes said – might actively conspire against us, presenting us sights and sounds and other types of data that create the illusion of one causal story, while hiding data that describes the true causal story.

Consequently, the more we can observe the true "data generating process" at the heart of the system in which our tasks take place, the better we are able to predict the consequences of acting within that system. The better these predictions, the more we can choose preferable actions. The more we can choose preferable actions,

the more we can benefit ourselves.

We have formalised this quest as the search for "causal knowledge", and over millennia have created and refined language to aid us in our search. We seek to understand *causes* and their *effects;* seek to know how much one component *changes* with respect to another; ask *why* a component changes so we can form *explanations;* predict what will happen if I *intervene* in some way; retrospectively ask *what if* a different change had occurred.

However, even if we use this terminology, we are not guaranteed the right answer. Extrapolating from what we see to create theories of what we know (in terms of a causal explanation) is a difficult and risky process. First, we need to have tools that help us see patterns, and interpret what we're seeing so that we can form explanations at the right level of detail. Thousands of years of theories of biology and physics were roughly useful, but only when we had access to a microscope could we illuminate how they actually worked (at smaller scales, at least).

Second, even if we can see compelling stories, our eyes may deceive us. We need tools that verify we are reproducing what our terminology is asking for. To take a famous example, suppose you are a farmer. Every morning of your life, you have observed that the rooster crows and then the sun rises. You make the explanatory leap: *roosters cause the sun to rise!* While you have observed a strong correlation, you have in reality mistaken the direction of causality – and your attempts to raise the sun in the winter months by making roosters crow will be for naught. We will formalise why this happens in the next chapter, but it suffices to say for now that we need more than just storymaking. We must have tools to verify the narratives we produce.

1.2 Reasoning with machines, or the motivation of this thesis

We sit at an inflection point in the history of using machines to perform tasks in conjunction with human reasoning. As we increasingly automate humans out of tasks — something we explore in chapter 4 — we will increasingly combine human and machine reasoning to make decisions. We already are. Each day, machine learning provides business-leaders forecasts on which they make strategic decisions; doctors use image processing techniques to reason about diagnoses; models suggest targeted educational content for personalised learning plans. This incomplete list shows the reach our tools have, and the promise they bring. Speeding up their integration, in a safe way, would be a good thing.

However, the key barrier to their integration into our most sensitive, important decisions seems to be that they lack the satisfying qualities we seek in our search for knowledge. We are just forming our ideas of how to interpret what machine learning models do. We are not yet sure how to train machines to extract and use causal information. Because these two characteristics are core to our thinking, it is easy to understand why we feel uncomfortable with abdicating decision-making to machines that do not have them.

And yet, machine learning is valuable often because it can do what humans do not. Machines can often learn from more data than humans can; they can find patterns in high dimensional settings that would be impractical or impossible for humans to search for; they can often make predictions faster once trained; they can operate 24/7 without the need to sleep; they can scale inference to billions of users at the same time; machines are relatively inexpensive.

This thesis is inspired by this disconnect. It is motivated by the belief that to best exploit the value of machines, we should make them interpretable, and endow them with a sense of cause and effect, because humans seem to naturally use these in their reasoning.

1.3 Overview of this thesis

The goal of this thesis is to contribute towards our understanding of using interpretable causal tools and machine learning together for making good decisions. I introduce three general methods, and one context-specific application. In the next chapter, I give a brief overview that connects machine learning, interpretability, causality, and decision-making. In particular, I introduce the "Ladder" of causal reasoning, describing the hierarchical ordering of associational, interventional, and counterfactual reasoning. Following this chapter, I describe a contribution on each step of the Ladder, moving from associational through to counterfactual reasoning.

The aims and structure of these chapters are as follows. As a background, **Chapter 2** introduces our increasing tendency to make decisions with machine inputs, the nature of those decisions, and our need for interpretability. It then reviews machine learning fundamentals, contextualising most machine learning tasks as *associational*. It describes how the correlation vs. causation dilemma leads associational reasoning to fall short of causal reasoning, and thus how machine learning is still ill-equipped to perform causal reasoning; this motivates a tour through approaches to causal reasoning, describing causal inference and the language of expression we use for it; last, it connects interpretable causal reasoning with human psychology, decision-making, and modern machine learning tasks.

The next four chapters make at least one research contribution each to a rung of the

Ladder:

1. We begin on the first step – *association*. **Chapter 3** describes PIGEBaQ, a Bayesian modelling toolkit for probabilistically characterising black box model gradients for decision makers who wish to make an *intervention* but without experimental data. We designed PIGEBaQ for the decision-maker that is beginning to make decisions with support from machine learning models.

2. **Chapter 4** introduces a survey on task automatability, where we collect new expert opinion on what tasks we can automate using intelligent technology. Of particular focus here is how to aggregate expert knowledge, evaluate predictions based on that knowledge, and interpret these predictions. We model expert knowledge aggregation as a Bayesian classifier combination problem, explain predictions through gradients, and argue for greater granularity of data collection in order to improve explanatory power.

3. **Chapter 5** addresses the *interventional* step of the Ladder and introduces Multisynth, a Bayesian treatment effect estimation model that extends the synthetic control method. Multisynth argues that temporal treatment effect estimation shares significant similarities with multitask modelling. We introduce a modified coregionalisation model that allows a modeller significant flexibility while relaxing a key restricting identification assumption with most synthetic control adaptations.

4. **Chapter 6** addresses the *counterfactual* step of the Ladder. We evaluate twin networks, a method originally introduced in Balke and Pearl (1994b) that performs counterfactual inference by modifying the structure of a causal model rather than the procedure of inference. Given the increasing use of counterfactuals in

machine learning, efficient methods of counterfactual inference are needed; we analyse and show that twin networks are a desirable modification, but not for the reasons originally proposed.

5. in **Chapter 7**, I review promising areas for future work and briefly chart a vision for full integration of interpretable causal tools in machine learning.

A defining characteristic of extracting interpretable causal insights is that it is hard to do. In practice, having causal information – such as data from controlled experiments, or a sufficient and fully-specified causal model – can be costly or impossible to collect. Regardless, intuitions from causality are still useful. The people who seek to use causal intuitions often find themselves in important decision-making scenarios. This thesis is motivated by the practical value of causal reasoning. As a result, it introduces novel research where causal reasoning is useful with or without a full treatment of causal identification.

Underlying this motivation is a simple hope: the more we treat causal reasoning as an engineering problem the more we'll be successful at augmenting intelligence. This is a blasphemous statement within the causal inference world.[1] I make it in service of the pursuit of general machine intelligence.

[1] And it is worth stirring the discussion, at least to ensure that this pursuit does not give license to bad and possibly dangerous non-causal conclusions.

1.4 Technical results of this thesis

1.4.1 A Bayesian model for interpreting the distribution and volatility of the expected gradient

Interpreting data, especially with an eye towards action, requires some notion of "change' of an output wiht respect to an output. It would be good to get an understanding of how this change occurs in ones data. In Chapter 3, we introduce PIGEBaQ, a "sit on top" model that can be applied to any data and give a probabilistic estimate of the average gradient in the data. PIGEBaQ (*Principled Interpretability for Gradient Evaluation using Bayesian Quadrature*) is motivated by the key insight that the expectation of the gradient – a quantity of interest itself – is an integration task that can be modelled approximately with Bayesian Quadrature. Bayesian Quadrature is an integration technique which implies a posterior over the integral by placing a prior over the integrand. By treating the expectation of the gradient probabilistically, a decision-maker receives several quantities of interest that are not possible with traditional gradient expectation-only methods. First, we can receive a true distribution over the expected gradient, weighted by a prior. From this distribution we can calibrate uncertainty and even choose new sampling points for further study. Second, we can characterise the volatility of the gradient to get an understanding for how it changes across feature space, whereas otherwise we might lose valuable information due to the expectation of the gradient. This also allows us to characterise the probability that the data displays a monotonic pattern, a quantity of interest to a modeller. By its nonparametric nature, we can do all this without making limiting functional form assumptions that would affect our gradient structure. We derive each result, show gradient recovery across synthetic data, discuss comparative benefits, and apply to

an example case within education policy.

1.4.2 Aggregating expert knowledge for subjective interpretation

Despite the attention around the role of intelligent systems in displacing or augmenting human labour, we have neither good data nor good models for how this may happen. We collected the first dataset of its kind to do so. We surveyed 156 intelligent systems experts from academia and industry across the world to create a dataset of $4,599$ task-level ratings of automatability. We consider the challenge of eliciting this expert knowledge, representing it, combining it, and interpreting it. To strike the balance between informativeness and user-friendliness, we opt for an ordinal representation of task ratings. We then aggregate task ratings using a weighted aggregation model into a level of activity categorisation that maps to feature space. We then use the Independent Bayesian Combination of Classifiers model to elicit an essentially agreement-weighted ground truth label for each activity. Last, we use a Gaussian Process Ordinal Likelihood model from which we discussions interpretations. We then discuss various applied consequences of the results, including using the expected gradients from the Gaussian Process to analyse the relationship between features and automatability. We conclude with a discussion on the role of data collection and granularity of data measurement to provide policy-ready interpretation, shifting the focus from models to data for interpretability.

1.4.3 A multitask model for temporal treatment effect estimation

We consider the temporal treatment effect estimation problem, where a treatment effect is to be estimated over time. To do so, we need to construct a counterfactual history of the treated unit(s). We consider the Synthetic Control Method, *Synth*, that translates this counterfactual construction task into an optimisation task that finds an optimal linear mapping from unaffected donor units to the effected treated units. We make the case that this problem is a missing data and prediction problem that is well handled by multitask machine learning. We consider the needs of likely causal inference scenarios, and settle on extending multitask Gaussian processes (MTGP) to do so. We develop a modified Linear Coregionalisation Model that allows for arbitrarily flexible accounting of unit-specific trends to reduce error. We show that multitask Gaussian processes have several advantages over *Synth*, including the ability to extrapolate, relax the stable relationship identifying assumption by accounting for post-intervention data, provide probabilistic estimates, and incorporate self-trends. There is research that develops models that address one or more of these, but each of these are not yet performed by a single model. A modified MTGP is naturally able to do so, making them (accidentally) a powerful tool for temporal causal inference.

1.4.4 Evaluation of twin networks for counterfactual inference

As counterfactuals become increasingly widespread in machine learning, it would be useful to have clearer, more efficient methods for estimating them. Here we perform, to our knowledge, the first empirical evaluation of twin networks for counterfactual inference. Twin networks enable full counterfactual inference within the graphical

Structural Causal Model framework via a simple modification. We show that they do indeed outperform standard counterfactual inference when using exact inference. We argue that theoretical analysis of twin network performance should focus on how characteristics of a graph, particular its substructures and motifs, can be taken advantage of by whichever exact inference method is used. We then standard and twin network counterfactual inference are equivalent when using sampling or variational inference-based approximate inference. However, there are some considerations to make when using approximate inference, which if not judiciously followed may make twin networks more useful. We show that under certain approximate inference contexts, twin networks can be beneficial. In particular, they can outperform if there is a large cost to sampling from the standard posterior, if there is counterfactual evidence, and if there is bootstrapping error in the resampling step. Finally, we introduce a probabilistic programming engine for performing counterfactual inference and exploiting the insights of twin networks in inference. We then consider future work, advocating for a greater study of how interaction between graph structures and inference algorithms make twin networks more or less efficient.

"Man is so intelligent that he feels impelled to invent theories to account for what happens in the world. Unfortunately, he is not quite intelligent enough, in most cases, to find correct explanations. So that when he acts on his theories, he behaves very often like a lunatic."

— Aldous Huxley

2

Concepts

The purpose of this chapter is to introduce the basic challenges of using machine learning to assist in decision-making. I give an overview of what I mean by what machine learning is, what causality and interpretability are, and how they can fit into machine learning to help decision-making. It is clear that interpretable, causal notions are important for tasks humans wish computers would perform, and this chapter describes the language we cast them in and how I think about them in the research in this thesis.

Critically, I introduce Pearl's Ladder of Causation, motivating each chapter by making a contribution to one rung of the ladder.

This is not a comprehensive introduction to all the technicalities of the problems and

details of expressing causal models and interpreting them. It should be seen as a first order overview of the essential concepts. It assumes a basic understanding probability statements and terminology of graphs. For further details of the implications of the challenges and systems of expression described herein, I point the curious reader to the subsections of Pearl (2009), Peters, Janzing, and Schölkopf (2017), and Rubin (1974) that expand on the details.

2.1 The origins of causality: why we care

As we increasingly co-exist with machines, it seems we will naturally look to them to help us answer many of the questions we instinctively ask. Whether in scientific discovery, economic policy, medical treatment, game-playing, generative art, or more, we naturally express questions that ask *what can and should we do, and why?*

One of the most general questions we tend to ask is the **relation** question: how are variables X and Y related? By that, we might (nonexhaustively) mean: how do they depend on one another? How do their extremities align? (Which subset $X_S \subseteq X$ is related to the most extreme values of Y?) What are naturally informative groups within X? (How can X-space be partitioned, subject to some regularising function ρ, to minimise global entropy?) We express the **relation** mechanism usually in statistical terms: what is the statistical relationship between X and Y? We use relation information usually as an input into decision making. For example, one of the canonical common relation mechanisms is the least squares estimate of the coefficients of a linear model on $P(Y \mid X) = N(\boldsymbol{\beta}^T X, \boldsymbol{\sigma}^2)$, with $\boldsymbol{\beta} = (X^T X)^{-1} X^T Y$. We use the values $\beta_i \in \boldsymbol{\beta}$ as the linear effect of X_i on Y.

But what if we do not know if X is related to Y? This question is known as the **discovery** question: how do we discover the appropriate structure of relations? In

the linear least squares example, X is on the right hand side; it is an *independent* variable, *covariate,* or *feature* and Y is the *dependent,* or *response* variable. This structural knowledge may not yet be known. Naturally, we ask: does X cause Y, or Y cause X, or neither? Does a drug cause a recovery, or does an economic policy cure a recession? Imagine all variables X as nodes in a graph with edge connections denoted by $\leftarrow, \rightarrow$, and where each arrow denotes a causal influence flowing from one variable to another. We ask which relation holds: $\{X_i \rightarrow X_j, X_j \rightarrow X_i, X_j \not\leftrightarrow X_i\}$. Sometimes, in machine learning, we use the **relation** mechanism to approximate the answer to the discovery question – which usually takes place in the forms of feature relevance and feature engineering.

Suppose we have the structural knowledge about the directions of relationships. Or, at least, we have a *model* of those relationships, that may or may not be the true model. Given a model (e.g. $X \rightarrow Y$), we are next interested in the **identification** question: what is the causal effect of X on Y? For reasons explained in the following questions, the relation mechanism may not be the causal mechanism; intuitively, causation is not correlation. This question is better framed as what happens to Y if *we force* $X_i = x_i$? This is known as an **intervention,** and should be remembered as the most important wrench in the causal toolbox.

Last, we are interested in a very special, and complicated question, but one that is surprisingly common: the **counterfactual.** A counterfactual question asks: *what would have happened to* Y *if* X_i *had been* x_i*?* In more formal language, $X_i = x_i$ may be an *antecedent* that is false — it is an event that did not actually happen — supposing that some $X_i = x_{real}$ did actually happen, and we are intervening on X_i to take a particular value, all else equal, given the non-intervened values of what did take place. As will be explored, the counterfactual notion is used extensively in our daily lives: *would I have arrived late had I taken a different route? Would this person be dead had the accused not*

shot them? Would the patient's cancer have metastasised had we found it 2 weeks later than we did? As will be formalised, a counterfactual is a combination of the **relation** and the **identification** question – more directly, **association** and **intervention.**

> **Definition 1 (Intuitive definition of Intervention)** *An intervention on X in some system S is a forced change to the way in which the values of X are generated, regardless of the ways in which the values of other elements of S are generated. (A more formal definition follows in the section on Structural Causal Models.)*

Only the first question is a solely statistical question. The latter questions involve two new tools: *structure* and *intervention.* These two tools are directly a result of the causal nature of these questions. Indeed, we frame them as causal questions, where we can force some relationships to values we desire and we can change contexts as we wish. This immediately presents a problem, because as soon as we encapsulate this notion of *intervention* on a world or system, we move outside of the classical context on which machine learning has been developed: statistical relationships, rather than causal ones.

2.2 Intrepreting models when we want to act on them

The context of this thesis is one where decision-makers and machines are increasingly interacting to make decisions that matter to things we care about. This brings two important questions: what decisions are we making, and what do we need in our machines in order to make good decisions?

We have begun employing machines in a wide variety of decisions, from frequent small decisions to infrequent very large decisions. Some of the most widely used

machine decisions operate at massive scale: internet advertising and search, content recommendation, compute resource allocation. But with better machine learning systems, we have increasingly integrated machines into more complex and valuable decisions. Medicine – diagnosis, prescription, preventative actions – are a significant focus in machine learning. Economic policy increasingly relies on machine learning models to capture complex patterns. Drug discovery will prioritise candidate compounds generated by machine learning models.

This thesis is motivated by the extrapolation of this phenomenon. Machine learning is likely to be increasingly used in highly consequential decisions. They are a core component of the as yet unachieved dreams of personalised medicine and education, autonomous driving, intelligent warfare, advanced materials and fabrication, and automated policy-making. For the most part, integrating machine learning into our decisions will be useful, especially when guided judiciously. As a result, it is worth asking the question: what do we need to safely integrate machine learning models into our decisions?

To that end, we have increasingly sought for ways to interpret machine learning models. This search has formalised the field of **interpretability**. Research in this field usually focuses on one of two areas, both of which are used to improve the outcomes of decisions:

1. Understanding *models*: illuminating a machine learning model's mechanisms, representations, or outputs. These are used for three main model-related tasks: debugging models, validating model behaviour, or developing better models.

2. Understanding *phenomena*: using a representation or output to understand a phenomenon of interest. This is used mainly to derive a plan, such as taking a beneficial action, or seeking further knowledge.

In this thesis, we specifically focus on the latter. We are interested in interpreting what models have to say about phenomena using feature spaces that represent feasible interventions. For this reason, we are open to whichever type of interpretability approach proves useful. However, it is worth illustrating a brief taxonomy of interpretability approaches so that we can place the research of this thesis in context.

2.2.1 Types of interpretability models

First, a model may be **intrinsically interpretable** in that the structure model itself lends itself to an easy interpretation; a decision tree, or linear regression, are examples of intrinsically interpretable models. Alternatively, the model may be **post hoc interpretable**; that is, some additional process is used to find patterns in the outputs of an uninterpretable model. In either case, if an additional model is used to interpret any model's internals or its outputs, this additional model is a **model-agnostic method**.

A model may give **local interpretations**, meaning an interpretation of an outcome for one part of feature space (usually a single observation). LIME and SHAP are popular examples of such a model (Ribeiro, Singh, and Guestrin, 2016b; Lundberg and Lee, 2017). Alternatively, it may give **global interpretations**, summarising some behaviour that occurs over feature space. A model may give **complete interpretations**, which are comprehensive mechanistic explanations, even if not conventionally understood (such as the values of weights and neurons in a neural network for a given datapoint); more commonly, and most usefully, a model may give **practical interpretations**, forgoing complete mechanistic explanations in favour of human-usable interpretations.

This is a basic but not comprehensive overview of interpretability. An unanswered question is what makes for a satisfying explanation in the first place. For a more detailed overview of philosophical considerations, definitions, and examples of models and

outputs of the types described above, see Gilpin et al. (2018) and Du, Liu, and Hu (2019).

In particular, in this thesis, each chapter uses different subsets of the interpretability field. In chapter 3, we develop a model-agnostic method for global interpretation. In chapter 4, we combine this with some local interpretations. In chapter 5, we use both a post hoc and intrinsically interpretable method. In chapter 6, we use a model that is theoretically intrinsically interpretable, post hoc interpretable, local, global, complete, and practical. Each contribution is united in using model outputs or characteristics as aids to decisions about what to change. Some might automate the decision, while others might only inform human theories that affect decisions.

It should be mentioned that humans can be inputs to models rather than only using models or their outputs in their decisions. Human subjective knowledge is valuable information about human intuitions or structures of phenomena. These are valuable because they capture latent knowledge of phenomena that may be unmeasured, and because they help shape machine intelligence to replicate human desires. In this case, we equally need interpretability tools to verify that these two characteristics are being picked up on. We term this *human-in-the-loop*. We illustrate this with an example in Chapter 4.

In sum, we are interested in using interpretability to shed light onto a model so that we can make better decisions. Inherent in this concept is an *action* from which we expect an outcome to result. This action-response framework is inherently a causal notion, hence the need to interpret this task from a causal lens. Before moving onto a causal framework, it is illustrative first to survey what is *not* causal.

2.3 The traditional machine learning paradigm

Machine learning has, by and large, developed its results on modelling statistical relationships. We usually frame machine learning tasks probabilistically, where we desire to estimate a joint probability $P(X, Y)$, a conditional probability $P(Y|X)$, or a marginal probability $P(X)$. If possible, a joint probability is a strictly more powerful expression than either, given the ability to integrate; by the sum and product rules, we can express the joint as a product of a conditional and a marginal:

$$P(X, Y) = P(Y \mid X)P(X)\,,$$

and the marginal as an integration of the joint:

$$P(Y) = \int P(X, Y)dX\,.$$

We cannot recover the joint $P(X, Y)$ from $P(Y)$ as the integration "destroys" information; we can recover the joint $P(X, Y)$ from the conditional $P(Y \mid X)$, provided we can model $P(X)$. That is: everything one wants to do with a conditional probability, one can do by having the full joint.

A model that estimates this joint probability is known as a **generative model**; this is in contrast to a **discriminative model**, which only estimates a conditional $P(Y \mid X)$ or $P(X \mid Y)$ (Ng and Jordan, 2002). While, as argued above, a joint (generative) model is strictly more desirable than a conditional (discriminative) alone, there may be practical limitations that make a conditional more desirable to estimate. For example, modelling the marginal $P(X)$ for the denominator may be prohibitive in practice—requiring a potentially computationally expensive integration operation—and it might be simpler to develop a functional form to estimate $P(Y \mid X)$ for

classification or regression purposes.

On the other hand, if one does have a full joint model, one has more capabilities than in the conditional case alone. Having a generative model affords extra advantages, including:

- *Sampling from the joint distribution.* Learning a full joint $P(X_1, X_2, ...X_n)$ allows us to sample from it and, for example, synthesise new images (with X_i representing the value of a particular pixel)
- *Answering any conditional or marginal question.* We need no extra information to answer any probability statement comprised of one or all of the variables if we have the joint; we would need to seek this information otherwise.
- *Naturally express probabilistic systems.* We can encode known independencies into our model by factorising the joint probabillity into a a product of probabilities. For example, a model of Y inheriting only from W (a mediating variable) and W inheriting only from X, we can write $P(X, W, Y) = P(Y \mid W)P(W \mid X)P(X)$. Doing so allows us to naturally express known independencies which describe the *generative process*. Sampling from this generative process requires sampling from the factorised joint.

Learning, in this paradigm, is the process of estimating $P(X, Y)$ or $P(Y \mid X)$, with some notion of "optimal" estimation, given some data D. For simplicity, we'll concentrate solely estimating the joint, $P(X, Y)$. First, we describe many different *generative models* $M = \{M_1, M_2, ..., M_n\}$ each describing a different factorisation of $P(X, Y)$. Each model $M_i \in M$ is parameterised by parameters $\boldsymbol{\theta}_i$. Within the Bayesian paradigm, we can write the joint posterior of the models and their parameters given the data by Bayes' formula:

$$P(M, \boldsymbol{\theta} \mid D) \propto P(D \mid \boldsymbol{\theta}, M) P(\boldsymbol{\theta} \mid M) P(M)$$

Given full access to every model M_i and parameters $\boldsymbol{\theta}_i$ we can apply one or another optimality criterion and find the $\{M_i, \boldsymbol{\theta}_i\}$ that maximises it. Two commonly used optimisation paradigms are:

- *Maximum Likelihood Estimate (MLE):* maximise $L(M, \boldsymbol{\theta}) = \sum_{i=1}^{n} p(D \mid \boldsymbol{\theta}, M)$
- *Maximum A Priori (MAP) Estimate:* maximise $P(D \mid \boldsymbol{\theta}, M) P(\boldsymbol{\theta} \mid M) P(M)$, with $P(M)$ forming the *prior over models* and $P(\boldsymbol{\theta} \mid M)$ forming the posterior probability of the parameters with respect to the model. Note that often a model family is defined, that is, $P(M) = \delta(M = M_i)$, in which case $P(\boldsymbol{\theta} \mid M) P(M) = P(\boldsymbol{\theta})$, forming the prior.

It is worth distinguishing vocabulary here. We define **learning** as optimising a belief over models and/or parameters with respect to data and a notion of optimality, and **estimation** as the process of extracting a quantity of interest from data, models, and parameters. **Inference** is a superset of these definitions: it is the process of deriving a posterior from a likelihood and a prior. Within the Bayesian paradigm, estimation and learning are a type of inference. They are the "backward pass" from data to the model. The "forward pass", from the model to the data (a prediction), may also be (but need not be) a type of inference.

2.3.1 What we can do with this paradigm

Let's stop and ask what this paradigm gives us. Given this paradigm, we can ask any probabilistic question given a probabilistic model. In standard machine learning, we frame our canonical tasks as answering this probabilistic question, which uses

inference to perform learning and estimation.

In **supervised machine learning,** we are given some dataset D comprised of variables $\boldsymbol{x}$ with *labels* $\boldsymbol{y}$. That is, $D = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N \sim P(X, Y)$. Here, the task is to predict $\boldsymbol{y}_i$ given some $\boldsymbol{x}_i$. (For example, a classic task is to **classify** an image of an animal, $\boldsymbol{x}_i$, into its proper categorical label $\boldsymbol{y}_i$. On the other hand, if the label is not categorical, the task is called a **regression** task.) $\boldsymbol{x}_i$ is some m-dimensional vector of attributes, each of which we call a **feature** or **covariate.** Using the formulation above, we can formalise this statistical model: given some $D = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N \sim P(X, Y)$, calculate $p(Y \mid X = \boldsymbol{x})$.

It is natural to ask what regions of X-space are most related to Y. We might then be interested in the **gradient** of Y with respect to X. Suppose we have modelled this task as $P(Y \mid X; \theta)$ such that $\mathbb{E}[Y \mid X; \theta] = f_\theta(X)$, with $\theta \sim P(\theta \mid X, Y)$; it is natural to evaluate, in other words, $\nabla_X f_\theta = \frac{\partial f_\theta(X)}{\partial X}$. The notion of a gradient occurs frequently in both machine learning (as a guide for the learning process) and in applications (as a notion of change with respect to an input). Gradients are increasingly heavily used to interpret models, to evaluate model fairness, and to imagine counterfactuals. Because gradients encapsulate the notion of change, we are often interested in what we can *do* with gradient information. In Chapter 4, I introduce a model-agnostic tool for using gradients for interpretability and decision making.

In **unsupervised machine learning,** we have the above context, without labels Y. Here, the task is to find "interesting characteristics" of the data, often involving **density estimation** by modelling $p(\boldsymbol{x})$. For example, we might speculate the data is structured into unseen clusters $\boldsymbol{z}$, and we write a generative model of the joint $p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})$. The task is to estimate the true distribution $P(X)$ with the empirical density $p(\boldsymbol{x})$, while accounting for this cluster structure. We intergate over possible clusters $\boldsymbol{z}$, writing the generative model as: $p(\boldsymbol{x}) = \int p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$.

When we extend a joint by a hypothesised hierarchy of generative processes, we have developed a **hierarchical model.**

We might here be interested in the **structure** uncovered by the model during inference. For example, how many clusters best explain the data? What feature transformation $\boldsymbol{x}' = g(\boldsymbol{x})$ is appropriate, and are there patterns within this new feature space? Once the desirable insights are uncovered, we tend to use them to take some action: how does this structure help us in making a decision?

2.3.2 Limits of this paradigm

Critically, in this paradigm, D is usually assumed to be drawn from the distribution $P(X, Y)$ to be estimated. Further, it is assumed that all $\{x_i, y_i\} \in D \sim P(X, Y)$. Last, it is assumed that $\{x_i, y_i\} \perp\!\!\!\perp \{x_j, y_j\}$; that is, all samples from $P(X, Y)$ are independent and identically distributed (IID). Intuitively, this says that your data comes from the same "world", and the *generating mechanism* of the data stays the same.

The IID assumption is more than just a handy intuitive assumption about the nature of the system we are trying to make inferences about. In addition, the IID assumption leads to universal consistency guarantees that imply that given enough data $D \sim P(X, Y)$ and computational power, we can develop a learning algorithm that provably converges to the lowest risk, or error, estimator (Schölkopf, 2019).

The IID assumption is pernicious because the modelling paradigms we have built on top of it fall down when we change the distribution from which X or Y or drawn. We are, in essence, hopeless, as we have a model for only the distribution we have observed. This presents us with a problem: a modeller needs some way to make inferences under a new distribution. The process of taking a model trained on one distribution and applying to another distribution generally falls under the categories of **generalisation**

or **transfer.** Sometimes we might describe it as **domain** or **covariate shift.** In essence, we are leaping from "seeing" to "doing", the latter of which involves changing the data generating mechanism of a distribution, thus changing the distribution.

Indeed, most complex real world systems involve a change in distribution. This problem is so thorny and famous in economics, for example, that the *Lucas Critique* – which says that as soon as choices are made with respect to a model designed to output advantageous choices given an equilibrium (IID) domain, the equilibrium shifts – remains an unsolved and challenging barrier 40 years later (Lucas, 1976). Intervening in an economy, giving a new drug treatment, or shipping a self-driving car to a new country, for example, all induce a change in distribution.

This challenge is a large problem because the questions introduced at the beginning of this section mostly involve a change in distribution. Technically, question (1) is an *associational question,* which we can address with our given paradigm. But as soon as we want to use a causal notion — forcing a variable to take a certain value, independent of other variables — we need tools to express this distributional shift.

If we are to understand these questions robustly, we need to build tools that allow us to make inferences under changes in distribution given causal actions. In order to do so, we typically look for some extra information, intuition, or inductive bias that guides our ability to do so.

2.4 Introducing causality into the paradigm

The questions we pursue (as listed above) introduce a distributional shift due to an **intervention.** Given a system, we seek to act on it. Through this action we can discover the structure of a system, the effect of an action, or the counterfactual outcome. Because we introduce a distributional shift, we are no longer in the realm of modelling

only observed statistical relationships; we are changing the relationship and inducing a causal one. For this reason, to address this challenge we have to move from *statistical dependence* to *causation,* the latter of which is a strictly more powerful concept.

Why might a causal relation be different than a statistical one? We have already alluded to its core difference: under a causal relationship, only a particular *direction* describes the causal influence from one variable to another. We can observe this via a factorisation of the system in question. Imagine describing a generative model of prices and house size measured in square footage. A statistical treatment may factorise the joint in (at least) two ways: $P(\text{price}, \text{sqft}) = P(\text{price} \mid \text{sqft})P(\text{sqft}) = P(\text{sqft} \mid \text{price})P(\text{price})$. Intuitively, the former describes a classic prediction problem for home sellers, while the latter might be used by home buyers to prioritise possible target houses that do not have publicly available size information. However, only one direction is intuitively causal: the square footage changes the price, but changing the price will not change the square footage of a house.

Let's say we have a system like this: how do we know the causal direction? This is where the **intervention** is useful. While it will be defined more formally later, intuitively the intervention gives a modeller full control over a variable's value (or the way its value is generated), regardless of how the other variables are generated. We can use this tool to find causal directionality by observing the effect of interventions and whether or not it affects other variables. If it does affect those variables, it is a cause of those variables; if it does not, it is not a cause of those variables. In this example, by increasing a house's square footage, we will (likely) increase its price; on the other hand, increasing the price of a house on the market will not change the square footage. This intervention is the key to disentangling the causal directionality and, in turn, the causal factorisation of the system.

On the other hand, it may not be that two statistically dependent variables are

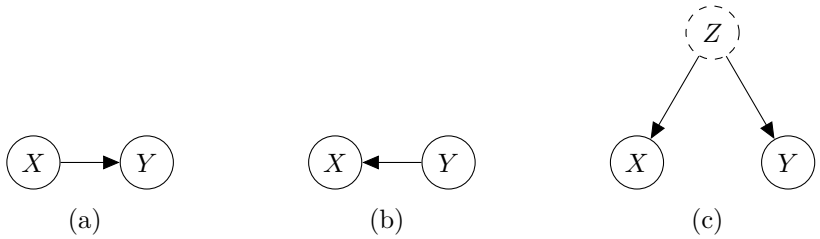


Fig. 2.1: Illustration of Reichenbach's Common Cause Principle. If $X \not\perp\!\!\!\perp Y$, three causal structures are possible. Either (a) X causes Y due to coinciding exactly with Z, (b) Y causes X for exactly the same reason, or (c) both share a common ancestor.

causally related in the first place, in the sense that they cause each other (where dependence between X and Y is measured such that $P(X,Y) = P(X)P(Y)$ if X and Y are independent, and $P(X,Y) \neq P(X)P(Y)$ if they are dependent). There may exist a third variable that causes them to be dependent, but where intervening on either variable may not affect the other.

This directionality is formalised in **Reichenbach's Common Cause Principle**, which states that for two variables X and Y if $X \not\perp\!\!\!\perp Y$ (i.e. X and Y are not independent), then there is a common cause Z that causally influences both. In a special case, Z coincides exactly with X or Y, leading to three possible causal structures.

The key takeaway of the Principle is that without additional information, we cannot decide which structure is the true structure. To be sure, there are other reasons for variables to be dependent, such as having common observed descendants. The salient point is that additional information – namely a proposition of a causal model – is needed before moving from dependence to causation; further, we can accord probabilistic dependence with causal influence.[1]

The core idea within causality that separates it from statistical relation, as measured by mutual information, is this property of directionality. Typically, we call this

[1]A fourth structure is possible, with $X \leftarrow Z \rightarrow Y$ and one of $X \rightarrow Y$ or $X \leftarrow Y$. We take this as a composite of the above three.

property **asymmetry** and it is the core explanation behind the common aphorism that "correlation is not causation." Correlation is a statistical artefact with no direction; causation is a mechanistic artefact with direction and also has statistical artefacts. A statistical artefact can make no causal conclusions without additional information such as the directionality, or the presence of Z.

This asymmetry is codified in the **principle of Independent Causal Mechanisms** (Peters, Janzing, and Schölkopf, 2017). The Principle states that if a system is factorised in the true causal way, the system can be expressed as a combination of mechanisms that do not "inform or influence each other". For a system $P(Y, X)$ factorised causally as $P(Y \mid X)P(X)$, a shift in $P(X)$ to $P'(X)$ should not change $P(Y \mid X)$; if $P(Y \mid X)$ were thought of as a function $f : X \to Y$, the parameters θ of f do not change if $P(X)$ were to change. In this sense, the causal components of the system $P(Y, X)$ are mechanisms that are independent of each other. In this principle, the elements of factorisations are **components** that are autonomous and modular.

A consequence of the ICM is that a fully independent model should be easier to *transport* to another context, where *transportation* is the application of a causal model to a domain where one or more causal mechanisms has changed. Because of the proposal that causal systems are independent, autonomous, and modular, some mechanisms are *invariant* to the context while others change. Adapting the model to a new context can focus on updating only the mechanism that has changed, rather than every mechanism.

Further, the ICM principle argues the causal direction is simpler to describe, in some cases, than the non-causal direction. An intervention in the causal factorisation does not need additional information to describe changes in other components. There is an analogous proposal in algorithmic information theory that states that a factorisation that can be described more simply is more likely to be the causal one (Peters, Janzing, and Schölkopf, 2017). For that reason, the ICM principle can be used to discover the

causal directionality of a system. Between two representations of equal modular size, the one that maintains this independence is more likely the causal one and can be exploited for predictive performance under certain assumptions (Peters, Bühlmann, and Meinshausen, 2016; Bloebaum et al., 2018; Schölkopf et al., 2012).

There is some debate over which systems are actually causal and described by Principle of Independent Causal Mechanisms. For example, in machine learning, it might be argued that processes that are modelled at some point with a lower-dimensional representation – as often happens in deep learning, such as in an autoencoder model, for example – implicitly describe the data generating process as a high-dimensional outcome generated by lower-dimensional mechanisms. The value of this representation can be tested by taking representations where the lower-dimensional representations satisfy some independence criteria, and comparing their performance to ones that do not satisfy the same criteria (Bengio et al., 2017). This may or may not be always useful in practice. This point is often studied as one of **entanglement**, which describes an increasing focus in machine learning on discovering representations with independent components. It is an open debate if finding disentangled representations is a causal process (Schölkopf, 2019). This is part of a larger debate about whether probabilistic factorisations – e.g. graphical models – ought to be causal.

In the examples above, we have used **interventions** to illuminate causal directionality and influence. Let's now formalise our definition of an intervention:

Definition 2 (Intervention) *An intervention* $do(X = x)$ *forces the variable* X *to take a value* x*.*

That is, an intervention forces a variable to take a particular value, *without depending on the values of other variables.* We use the *do*-operator to denote this. Critically, this

is different than conditioning, as explained below. However, there is an extension of an intervention that modifies the generating mechanism of X instead of its value, called a **soft intervention:**

Definition 3 (Soft Intervention) *A soft intervention* $do(X = f'(PA_x))$ *forces the variable* X *to take the value of* $f'(PA_x)$*, where* f' *is some function, and* PA_x *are the variables that are parent variables of* X*.*

In turn, we can formally define a **cause:**

Definition 4 (Cause) *Given variables* X, Y *and an intervention* $do(X = x)$*, we say* X *is a cause of* Y *if* $P(Y) \neq P(Y \mid do(X = x))$ *for at least one intervention on* X*.*

Note that as we talk about the distribution of Y conditional on the intervention, we assume that the effect of an intervention $\text{do}(X = x)$ has propagated to its descendants. That is, the distribution of a descendant of X is updated to account for the new value of X.

For exactly the same reason as illustrated in the examples above, conditioning is *not* the same as intervening. Conditioning on $X = x$ constrains a joint distribution $P(X, Y)$ to the region where $X = x$. Intervening, on the other hand, changes the nature of the distribution, and induces a new **interventional distribution**. The interventional distribution may be the same as the conditional one, but need not be. When they are the same will be highlighted below, but let's examine a situation where they are different. Take, for example, $P(Y = \text{high} \mid T = \text{intensive})$, the probability that an asthma patient will have a top-50% post-hospital outcome $Y = \text{high}$ given a very intensive

treatment $T = \text{intensive}$ (such as an invasive corrective surgery, or a heavy biological drug). We then run a randomised controlled trial ("RCT") that randomises treatments given to asthma patients; as explained this represents the interventional distribution $P(Y = \text{high} \mid \text{do}(T = \text{intensive}))$. In this case, however, we find that not only are the distributions different, but they are the inverse: $P(Y = \text{high} \mid T = \text{intensive}) = 0.2$ while $P(Y = \text{high} \mid \text{do}(T = \text{intensive})) = 0.8$. Why? In this case, there is a selection bias: in historical data, only patients with severe, hard-to-treat chronic asthma received intensive treatments. Their condition made them predisposed to worse long-term outcomes. In the randomised case, patients received an intensive treatment regardless of their profile, and thus both easy and difficult cases received intensive treatment, leading to better long-term outcomes on average. This selection effect is an example of **confounding** and **model misspecification,** which we touch on below. A key takeaway here is that modelling a causal system without causal tools can cause benign or disastrous outcomes. For example, making treatment decisions on the conditional model might lead one to conclude they should only prescribe light treatments, as the average outcome was always better.

2.5 Challenges in causal inference

Causal reasoning is not easy in practice, because the tools that we have to perform it are often difficult – and sometimes impossible – to have in real life. The first, and most common mistake, is to interpret a non-causal statement as causal. The second mistake is to misidentify a causal effect due to a poor causal model or poor inference procedure. We'll discuss these two types of mistakes and frame why they occur.

2.5.1 Spurious relationships, or "correlation is not causation"

The most basic, and most common, mistake in causal inference is confusing correlation for causation. This stems from two sources:

1. Interpreting a claim that two variables are dependent as a causal relationship, when it was intended by the claimant to only be statistical. For example, if $cov(X, Y) \neq 0$, the statement alone does not carry causal information. What is missing is a causal model and an argument for why $cov(X, Y) \neq 0$. A model $X \to Y$ with a believable justification would imply that the dependence is the result of a causal model.
2. Claiming a causal effect, of a certain size, even though the underlying causal model is wrong. We've seen this in the Common Cause Principle: it could be that two correlated variables both inherit from a third variable but do not have a causal link from one to another. Alternatively, given observed variables X and Y and latent variable Z, with $X \to Y$ and $Z \to Y$ (but not $Z \to X$), $Z = X$, and $Y = X + Z$, then if Z is not observed X will be mistakenly attributed with twice as much effect as it actually has on Y.

However, there is a case yet for dependence, which will be important in chapter 4. We humans tend to use dependence as a "first guess" at a causal model. Taking Reichenbach's Common Cause Principle as an example, assuming the observed dependence is not by chance, dependence makes the case that one variable is a cause of another variable, or that there is a common cause variable a modeller should investigate. Further, in the case that the causal link $X \to Y$ and for $X \not\perp\!\!\!\perp Y$ are both true, there would be a variable set S such that the causal effect of S on Y is exactly that to neutralise the effect of X on Y. For example, suppose S is a single variable and

$Y = X + S$, and a statistical analysis of data sampled from this system shows $X \perp\!\!\!\perp Y$. S must then equal $Y - X + \epsilon$, where ϵ is some factor uncorrelated with Y. This is possible, but in real world systems, it is intuitively unlikely. As a result, dependence is a first indicator of a possibly relevant causal connection. In the realms of causal inference, it can be used to prioritise variables for further study so that a modeller can refine their causal model or seek new data. This is very useful, for example, in high-dimensional cases or in cases where one wants to use flexible black-box models. This is precisely the deployment we consider in chapter 3.

2.5.2 Confounding and causal sufficiency

Simpson's paradox (Simpson, 1951) illustrates why non-causal estimates can be dangerous if we interpret them as having more information than they do. Simpson's paradox is a case of confounding where a marginal or conditional distribution shows one pattern of covariation but conditioning or de-conditioning the same distribution reverses the pattern (Julious and Mullee, 1994). As a result, causal claims on only statistical information can lead to fragile, and potentially dangerous, insights.

A clear case is group-subgroup reversal. Take the left scatterplot in Figure 2.2, which shows a sample of individuals, the size of a one-time unconditional cash transfer given to them, *transfer*, and the change in their long-term annual income after the transfer, *response*. (The higher the response, the more that value of cash transfer raises the individual's long term income.) The graph shows the least squares linear model with the assumed form:

$$\text{response} \mid \text{transfer}, \beta, \sigma^2 \sim \mathcal{N}(\text{transfer} \cdot \beta, \sigma^2)\,,$$

where $\mathcal{N}$ denotes the Normal distribution, σ^2 is a variance parameter, and β is

the linear slope parameter. The slope is interpreted as the effect of the cash transfer, with a positive slope at 0.15: a $15 increase in permanent long-term income for every $100 increase in cash transfer. A policymaker may conclude that the larger the unconditional cash transfer, the better the outcome. She eagerly reports back to her government that they should increase the size of the cash transfer. (Note that our savvy policymaker is careful to say cash transfers are *associated* with an increase in income, but also urges for their expansion.)

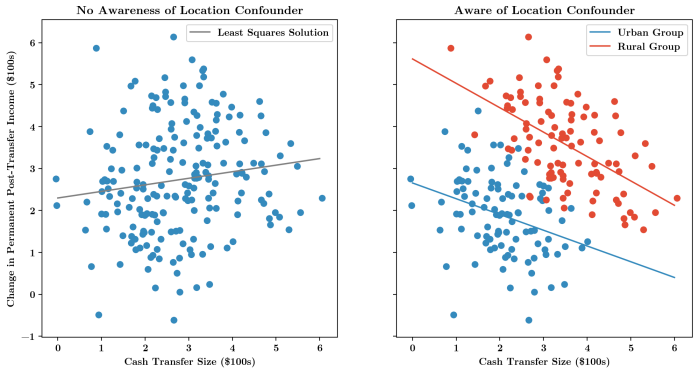


Fig. 2.2: (left:) A scatterplot and regression of the effect of a cash transfer on long-term income, without modelling recipient location. Without disaggregating by location, we see a weakly positive association of the cash transfer and long-term income. (right:) Separate regressions for the urban and rural groups. When disaggregated, we see that the associations appear strongly negative in both groups.

However, if we had one additional piece of information – the location of each individual receiving the cash transfer – the insights change. Divided into rural and urban we notice two distinct subgroups within the sample. When we run separate linear models for them both, we see the effect for both is actually negative. The true effect of the cash transfer is -0.37 in the urban group, and -0.58 in in the urban group. In otherwords, the *smaller* the transfer, the better the long-term outcome.

That is, the location of the recipient appears to have an effect both on the transfer size (rural and urban communities have different physical and human capital endowments, and pre-transfer incomes are different) and the response to the transfer

(cash transfers may be differentially complementary to these endowments). The original assumed model was *transfer* $\rightarrow$ *response* but the true model is *transfer* $\leftarrow$ *location* $\rightarrow$ *response*, with *transfer* $\rightarrow$ *response*. Conditioning on both reveals that the subgroup effects are negative, while the overall effect is positive, hence the paradox. The eager policy analyst might then enact a policy with the exact opposite effect than desired. Had we intervened on *transfer*, *response* would have been negative.[2]

2.5.3 The fundamental problem of causal inference

As we've discussed, we try to avoid these problems by intervening on a system and observing the result. Intervening, for example, allows us to eliminate confounding if done properly. However, observing the result of an intervention involves a *comparison* between the intervened and non-intervened state. The so-called Fundamental Problem of Causal Inference refers to the fact that in order to know the true causal effect of an intervention on a variable, we need to simultaneously observe the value of a variable under an intervention *and* not under the intervention. The difference between the two we describe as the **treatment effect,** and will be elaborated on below. However, there is an inherent contradiction: only under extremely restrictive assumptions can you observe both. For example, to observe the causal effect of a particular chemotherapy treatment on a patient's tumor, a doctor has to have that patient take the treatment *and* not take the treatment *at the same time.* A doctor could perform both trials in sequence, but the difference in time would likely cause a difference in other biometric variables relevant to the patient's tumor; functionally, each intervention is happening on a different patient, but the same human.

The consequence of the Fundamental Problem of Causal Inference is that we are

[2]This is just an example for intuition, and it does not necessarily reflect the state of understanding of the effect of unconditional cash transfers.

forced to *estimate* the treatment effect or causal model under consideration. This estimation is the heart of causal reasoning. In order to estimate the answer to a causal query, we need either an **estimand** (the quantity desired to be calculated) or a **model** (a fully generative model of the data where inheritance is causal). The process by which we move from an estimand or model to the estimate is the process of **causal inference.** Our estimates are only as good as our inference process, and as described above, causal inference is sensitive to the specification of the problem and the information available.

For that reason, the causal inference community has endeavoured to create techniques for performing causal inference designed for the situation at hand. For the purposes of this thesis, we are uninterested in surveying these methods; a good survey, focusing on applications in machine learning, can be found in Guo et al. (2018). However, what is relevant is how to choose the appropriate inference method. To do so, we need to first describe the causal inference problem. There are myriad methods for performing causal inference, but choosing the appropriate method depends on how you *express* your causal query in the first place.

2.6 Expressing causality

We still need a notational language with which to express and answer causal queries. Over the past century, several frameworks have emerged. Here we discuss the dominant two; it is useful to frame some of the language and approaches we see later in this thesis.

2.6.1 Potential outcomes

The first is the potential outcomes framework, developed first in Splawa-Neyman (1923) and Fisher (1926), and notably extended in Rubin (1974). The potential outcomes framework is centred on the concept of the **potential outcome**: given some treatment

t (e.g. a drug), and some unit i (e.g. a patient), the potential outcome of variable Y for that unit is denoted $Y_i(t)$. The potential outcome describes what would happen to unit i if they received treatment t. The treatment variable t is typically within $\{0, 1\}$, where 0 represents no treatment and 1 represents a treatment. Conceptually, a potential outcome is an attribute of a unit that exists whether or not that unit is treated; it is not (yet) an observational outcome.

Of interest is some **estimand**: a quantity, such as $Y_i(1) - Y_i(0)$, which is the difference between potential outcomes, or the **individual treatment effect**. The task is to first define the estimand and then compute it. But computing the estimand is not straightforward. First, we encounter the **Fundamental Problem of Causal Inference**: we can only ever observe one of $(Y_i(1), Y_i(0))$; we cannot simultaneously give and not give a drug to a specific patient. One approach to this problem is reframe the task from an individual-level to a population-level task, estimating the average treatment effect, i.e.:

$$\mathbb{E}\Big[Y(1) - Y(0)\Big] = \bar{Y}(1) - \bar{Y}(0) \tag{2.1}$$

where $\bar{Y}(t)$ is the *sample* mean of observed outcomes of all individuals sharing treatment status t and $Y(t)$ is the random variable representing the potential outcomes for all units if they were to take treatment status t. In other words, the difference of sample averages of observed outcomes is an unbiased estimator of the differences in the potential outcomes.

However, we must reassure ourselves that any quantity of interest, like the above, is computable using only observed data. This task is known as **identification**. While the right hand side of the above equation does not rely on any unobserved data, it is not obvious that it is an unbiased estimator of the true population treatment effect.

As a result, in order to estimate causal quantities using only statistical information, we have to make assumptions.

The above formulation requires several assumptions to be true. First, we require **ignorability**. Ignorability states that the assignment of treatment is independent of the potential outcomes. Let $Y(1)$ and $Y(0)$ be random variables representing potential outcomes across units of treatments $t = 1$ and $t = 0$, respectively, and let a new random variable T represent the assignment of treatments. For ignorability to hold, then:

$$P(Y(t)) = P(Y(t) \mid T = t) = P(Y \mid T = t) \tag{2.2}$$

where $P(Y(t)) = P(Y(t) \mid T = t)$ because the assumption states $Y(t) \perp\!\!\!\perp T$. The key insight of this assumption is that the distributions of potential outcomes, $P(Y(1))$ and $P(Y(0))$, can be reliably estimated using the observed conditional distributions $P(Y \mid T = 1)$ and $P(Y \mid T = 0)$, and thus we can recover the distributional counterpart of (2.1) by subtracting the two conditional distributions. This would not be the case if there were confounding between treatment assignment and outcomes. Intuitively, if ignorability did not hold, then whether or not a unit is treated would not be independent of how a unit would respond to the treatment; thus, the potential outcomes $Y(t)$ would not equal the observed outcomes and their treatment statuses, $Y \mid T$. A more general formulation of ignorability is known as strong ignorability, which is satisfied if there is some observed set of variables S such that $T \perp\!\!\!\perp Y(1), Y(0) \mid S$. In order for the average treatment effect as calculated in Equation 2.1 to be identified, then the modeller must first argue that the ignorability assumption is valid.

Two further assumptions are implicit in our formulation. First, by this formulation, $Y_i(t_i) \perp\!\!\!\perp Y_j(t_j)$ for any pair of units i, j and treatments t_i, t_j – that is, the potential outcome of one unit is independent of the potential outcome of another unit. Combined

with the assumption that treatments are exactly the same across units, this is known as the Stable Unit Treatment Values Assumption, or SUTVA. A final assumption that is required is the positivity assumption, which states that $0 < P(T \mid X) < 1$ for covariates X. This states that every treatment has a non-zero, but not certain, chance of being assigned a treatment.

These assumptions, taken together, allow us to **identify** causal estimands of interest from observational data using conditional distributions. Some quantities we may be interested in, for example, are the Conditional Average Treatment Effect (CATE), defined as $\mathbb{E}\Big[Y(1) - Y(0) \mid X = x\Big]$ where X is some covariate of all units, or the Average Treatment Effect on the Treated (ATT): $\mathbb{E}\Big[Y(1) - Y(0) \mid T = 1\Big]$, which measures the treatment effect of only the units that were actually treated.

Note that these assumptions naturally hold in properly-done trials where treatment assignment is randomised across units, which is where this framework originated. However, as mentioned, trials can be impractical, or impossible, to perform. If one is lucky, one might discover a "natural experiment", where a phenomenon has randomised assignment without an experimenter's intention. Those, however, are also rare. As a result, the search for valid causal inferences is often primarily occupied with developing an argument that given certain variables and an estimator, the causal effect is **identified**.

2.6.2 Probabilistic graphical framework

In contrast to the potential outcomes framework focus on the estimand and assumptions, the Structural Causal Model (SCM) framework, developed in its modern form largely by Pearl (Pearl, 2009) and Spirtes, Glymour, and Scheines (Spirtes, Glymour, and Scheines, 2000), is built around using a causal *model.* The causal model describes

a fully generative (hypothesised) causal process that generates data. This model is represented by a graph with functions that generate nodes, as well as an assumption that allows this graph to entail probability distributions. The task remains the same as in potential outcomes: estimate a causal quantity using only observed data. However, how we express a causal quantity, and how we make assurances that we can (or cannot) estimate it, are different.

A Structural Causal Model M is comprised of four components:

- a set of unobservable, or *exogenous*, variables U, distributed according to $P(U)$,
- a set of observable, or *endogenous*, variables $V = \{v_1, \ldots, v_n\}$,
- a directed acyclic graph G, called the *causal structure* of the model, whose nodes are the variables $U \cup V$,
- a collection of functions $F = \{f_1, \ldots, f_n\}$, where f_i is a mapping from $U \cup V \setminus \{v_i\}$ to v_i. This is symbolically represented as

$$v_i = f_i(PA_i, u_i), \text{ for } i = 1, \ldots, n,$$

where PA_i denotes the parent endogenous nodes of the ith endogenous variable in G.

In other words, we compose a graph where all edges are interpreted causally, and where every endogenous node is deterministically generated as a function of its parents. This looks very similar the causal relations notation we used earlier in this chapter. The deterministic functional mappings operate as the *causal mechanisms* referred to earlier, borrowed from the Structural Equations Modelling paradigm (Bollen, 2005).

Before we describe how to ask and answer queries of a Structural Causal Model, we first have to describe what it means to write a model. Writing a model is the

first step towards causal inference in the graphical approach, whereas writing the estimand is the first step in the potential outcomes approach. First, we note that the model is indeed a generative model of the data, in that we can express a distribution over variables. Given a Structural Causal Model M, let $Y \subseteq V$ be a set of variables of interest, and let $Y(\boldsymbol{u})$ be defined as the values of nodes Y when the vector $\boldsymbol{u}$ of exogenous variable values is passed to the model and the resulting values of V are evaluated. Then the probability that Y takes values $\boldsymbol{y}$ is:

$$P(Y = \boldsymbol{y}) := \sum_{\{\boldsymbol{u}|Y(\boldsymbol{u})=\boldsymbol{y}\}} P(\boldsymbol{u})\,.$$

That is, despite endogenous variables being deterministically generated as a function of their parents, we can express a distribution over endogenous variables due to the stochasticity of exogenous variables. We note that this allows us to express the notion that variables are causally generated by deterministic mechanisms.

Because the model entails a distribution, the edges in the graph imply statements of dependency. More accurately, they imply statements of *independence* (Koller and Friedman, 2009). The most natural question to ask given a graphical model is if different variables are dependent. Pearl (2009) shows that if two nodes are **d-separated** in the graph, their corresponding variables are independent in the distribution entailed by the model. Two nodes $X, Y \subseteq V$ are said be to be d-separated by a set of nodes S not containing X or Y if S *blocks all paths* between X and Y. A path p between X and Y is the set of nodes and edges that connect X and Y, including X and Y themselves. S blocks a path p if:

1. there is a collider (a node with two or more incoming edges from nodes in p) in p that is not in S and has no descendants in S, or,
2. p contains a node in S which is a parent of another node in p.

This test of independence has been popularised as the "Bayes-ball" criterion for its deployment in Bayesian networks. The ability to find independencies within the graphical model will be crucial in answering whether or not a particular causal query can be estimated or not. A further assumption, the **causal Markov** assumption, is usually made in this case, which is also from Bayesian networks: that a node is independent of its nondescedents given its parents. As a result, one could factorise the joint distribution into independent components:

$$P(V) = \prod_{i}^{n} P(v_i \mid PA_i, u_i)\,. \tag{2.3}$$

A final assumption that is usually made is the **faithfulness** assumption. The faithfulness assumption assumes that the independencies encoded by the graph, and no more, are represented in the distribution the graph entails, and vice versa. These two assumptions – the causal Markov assumption, and the faithfulness assumption – give us the power to state that dependency relations in the graph correspond to dependency relations in the distribution, and thus allow us to estimate graph quantities via probabilistic ones. This is important when using interventions.

In sum, a Structural Causal Model describes a generative model (and thus a distribution over variables), as well as statements of independence, and a natural factorisation of modular mechanisms. However, so far, these are only *statistical* conclusions. To answer causal queries, we reintroduce the intervention. Whereas in the potential outcomes framework, an intervention is described in binary form as the treatment, in the graphical model, an intervention is seen as a change in a mechanism f_j that generates any endogenous node v_j. The most common change to f_j is to make it a constant function: $f_j(PA_j, u_j) = c$, written $\text{do}(v_j = c)$, for some value c. Similarly we can now describe an **interventional distribution**, which

is the joint distribution of variables if we make one or more interventions, say by setting variable $v_j \in V$ to equal a constant c:

$$P(V \setminus \{v_j\} \mid \text{do}(v_j = c)) = \prod_{i \neq j}^{n} P(v_i \mid PA_i, u_i) \mid_{v_j = c} .$$

As a result, an SCM entails more than a joint distribution over variables, but also a distribution over variables given interventions.

Now that we have a model, and a way of writing conditional and causal queries, we can now ask how inference of either type is performed in an SCM. If we want to perform regular inference, i.e. estimate a conditional distribution, we can simply perform inference as we would in any graphical model. This approach will be more familiar to those from a machine learning and Bayesian statistics background. Inference over endogenous variables involves implicitly computing a posterior over (at maximum, all) latent variables. Let v_i be an endogenous node of interest, V_o be a set of observed endogenous nodes, and let $Z = V_l \cup U$ be the set of unobserved (latent) endogenous nodes V_l as well as exogenous nodes U. Then posterior inference over V_i can be calculated as:

$$P(v_i \mid V_o) = \int P(v_i \mid Z, V_o) P(Z \mid V_o) dZ$$

Depending on the independencies implied by the graph, integration may not need to be performed over all nodes in Z for a particular query.

Causal inference, on the other hand, is trickier. This query involves a node of interest v_i, observed nodes $V_o \subset V$, and node $v_j \in V \setminus V_o$ which is intervened upon and set to c. The query is written as $P(v_i \mid V_o, \text{do}(v_j = c))$. This defines an **interventional distribution.** Letting unobserved variables $Z = V_l \cup U$, it might then seem straightforward to apply the exact same inference query as above, but

including the *do*-operator:

$$P(v_i \mid V_o, \text{do}(v_j = c)) \\ = \int P(v_i \mid Z, V_o, \text{do}(v_j = c)) P(Z \mid V_o, \text{do}(v_j = c)) dZ \,.$$

But, as we saw in the potential outcomes case, an interventional response is only a theoretical idea, and we only have data on what we observed. As a result, inference in a graphical model must again face identification, where a causal query (any query involving the *do*-operator, like $P(v_i \mid V_o, \text{do}(v_j = c))$ is turned into one of only conditional probability statements without a *do*-operator.

In a graphical model, the way identification is achieved is by applying a simple procedure called the *do*-calculus. The *do*-calculus is a series of steps and checks applied to a graph that relies only on the set up of the graph. If, via the *do*-calculus, all *do* statements can be replaced by a conditional probability, then a causal quantity is identified. The key advantage this has over the potential outcomes framework is that it essentially makes identification automatic and algorithmic. This reduces the identification problem to positing a usable causal graph that allows for identification, rather than thinking about potentially complex confounding relationships between treatment and outcome variables. In fact, the graph makes visible the potential outcomes assumptions: strong ignorability is easily inspected given the graph and observed data, and violations of SUTVA and positivity are the result of a misspecified causal graph.

It should then stand out that the first core task of the graphical model approach is to find the correct structure of the causal graph (or, at least one that admits the same joint distribution relevant to a pre-defined query of interest as the full graph). This is

usually not easy, as the number of possible graphs is superexponential in the number of nodes. As a result, the assumptions one makes in the graphical framework is in the *graph*; we do not get causal inference without assumptions, no matter how hard we try.

However, the graphical model does have several advantages. First, it is intuitively simple: simply specify the causal relationships you believe are and are not operating. In essence, this is what the assumptions of potential outcomes do. Second, identification becomes algorithmic. Third, we can reuse the graph for other queries, which is something that becomes important in Chapter 6. If causal reasoning is to be integrated into general agent reasoning, then a reusable graph as a world model seems much better than defining estimands and thinking about their assumptions for every query. And last, researchers can test their causal models explicitly with other researchers. By being forced to put your assumptions on the page, one can identify critical shortcomings much more transparently.

2.7 The Ladder of Causation: working towards causal reasoning in practice

We have now explained the goal of causal inference, explained the traditional paradigm of machine learning and its shortcomings, and outlined the motivations and details behind different frameworks for causal reasoning. What, then, is our guiding research aim that motivates the contributions of this thesis? In short, each contribution is a contribution towards solving a problem on what Pearl calls the "Ladder of Causation" (Pearl and Mackenzie, 2018). The Ladder describes three types of queries that we ask of a system when we have a causal question in mind. It has three rungs: association, intervention, and counterfactual.

The first rung, the associational question, asks a conditional probability query without any causal tool like an intervention. For our purposes, we are interested in the posterior defined by the query so that we can learn from it. Without causal assumptions or data, the best we can do is ask better associational questions that we believe better model the outcome of our actions, or use such associational questions to inform future experiments or causal hypotheses. For example, we might build a hierarchical model that models the conditional subject to structure we believe ought to be modelled. Alternatively, the conditional can be an imperfect input to human decision-making by providing a semblance of interpretability. And as we saw above, depending on the causal assumptions we make and structure of our SCM, our conditional query does *identify* the true causal effect – something we can show using the do-calculus by proving equivalency between an interventional and associational question.

The second rung, the interventional question, asks a *do* query. As illustrated, it cannot be answered by information from the first rung unless the do-calculus *identifies* the effect. If it does not, then an intervention needs to take place. In the real world, this takes the form of a randomised controlled trial or a randomness-driven natural shock, for example. In a simulation, one can simply force the relevant variable on their structural causal model to take a value.

The third rung, the counterfactual questions, asks a *do* query in the context of a particular associational question. In doing so, it absorbs the interventional power of the second rung, and adds to it the contextual power of the first rung. Given no fixed context, a counterfactual is an interventional question; however, an interventional question requires accounting for a context which cannot be expressed in a single do-query without additional counterfactual assumptions.

As we move from one rung to the next, we gain expressibility that cannot be achieved only with information from the rung below. Pearl argues that as a result,

the traditional paradigm is fundamentally limited because it lacks interventional and counterfactual reasoning, even though both are integral parts of human reasoning.

As a result, contributing to each ladder on the rung is to contribute to the advance of causal reasoning in machine learning systems. This thesis exists to make a contribution to each: one motivated by its particular medical context, and three motivated by their need to make associational, interventional, and counterfactual reasoning easier in causal queries or the search for causal answers.

2.8 Does causality exist?

Before continuing to this thesis' research advancements, it is worth saying that for practicality's sake, we will leave behind myriad difficult but very interesting questions about whether some thing called "causality" even exists. Unfortunately, we have not reached a stage where this question does not prematurely hamstring useful research into causality. I often find that when I present work around causality in the machine learning community, those that do not already work on causality find themselves pondering the question of causality's existence. If it exists, is it a physical thing? Does it imply determinism and the lack of free will? Is it represented in the way particles in the universe interact? Is it implied by the use of the = sign when we write the laws of nature? Is it a convenient human psychological construct to describe a uniquely human perception that *things happen as a result of other things happening?*

Whatever the answer is – that is beyond my purview, and I suggest it be tackled over drinks in a pub late into the night – we do not know the answer. Causality is at least a formalism defined within the scope of the implications of the Structural Causal Model we have described. It is an idea that allows us to make powerful conclusions, upon which modern science rests. It is a language with which we express

human desires of action in this world.

Therefore: causality is useful, whether or not it exists. I propose that in order to make machines more intelligent, we put aside these questions for now and see causality first as being comprised of a useful set of tools. Causality seems, at least, a useful tool for *engineering* machine intelligence. Somewhere between "causality does not exist and is therefore pointless" and "identification or bust" lies all the useful tools from the causality toolbox that we can bring to making machines more intelligent.

2.9 Connections to machine learning

Here we highlight the increasing role of causal reasoning in traditional machine learning tasks and challenges. As discussed, causal tools are useful for answering questions particularly of association, intervention, and counterfactuals; in turn, these often relate to notions of accountability, generalisation, and explanation. We have seen advances integrating the tools of causal reasoning make their way into machine learning, in particular increasingly over the past 5 years.

Fairness, Accountability, and Transparency – in recent years, a community has emerged under the FAccT* umbrella. This community has searched for frameworks to understand potential harm to humans from machine learning systems, and attempted to develop tools to detect and mitigate it. Several of these tools have embodied notions of causality: interventional and counterfactual fairness (Kusner, Loftus, et al., 2017; Kusner, Russell, et al., 2018; Louizos et al., 2016; Loftus et al., 2018; Madras et al., 2019), and causal model criticism, interpretation and explanation (Tran, Ruiz, et al., 2016; Schwab and Karlen, 2019; Mothilal, Sharma, and Tan, 2020; Janzing, Minorics, and Bloebaum, 2020). As governments and companies increasingly deploy machine learning systems, it seems that government, users, and civil society will

increasingly demand these systems be fair, accountable, and transparent (Graham et al., 2020; Brundage et al., 2020).

Interpretability – interpretability in machine learning has roughly divided interpretation methods into two groups: models that are *intrinsically* interpretable, and models that can be interpreted *post hoc*. We find causality in both groups. In the first, a learned Structural Causal Model can be interrogated with any associational, interventional, or counterfactual question; if the SCM is identified, then any practical analysis method will recover the desired explanation. If the task is to discover the SCM, the structure of the SCM itself acts as an explanation for phenomena. This equally applies to models that imply causal reasoning, such as some rule-based models. Such a model appears in Chapter 6. A second type of intrinsically causal model for interpretability is a model that attempts to discover, but not identify, causal relationships (Carvalho, Pereira, and Cardoso, 2019; Murdoch et al., 2019; Moraffah et al., 2020).

Post hoc interpretation treats each model as a causal system, where we have full control over a model's parameters and hyperparameters once it has been trained. An interventional query holds, for example, the values of all neurons but one fixed in a deep neural network, while varying the value of another neuron, evaluating the outcome across a range of examples. A counterfactual query calculates neuron values for a single value, and intervenes on another neuron. It should be noted that this method does *not* identify the true treatment effect of input features on the output in the real-world system the model is modelling, but rather the effect of input features on the prediction output.

Improving generalisation, efficiency, and performance – we can incorporate the conclusions we have found in the search for interpretable causal systems. In particular, we have increasing evidence that incorporating causal information helps us generalise insights as well as learn more efficiently. Accounting for confounding, for example by introducing randomness where possible, can assist with generalisation

and transfer (Zhang and Bareinboim, 2017; Kallus and Zhou, 2018; Parbhoo, Wieser, and Roth, 2018); incorporating causal structures or interventional data can improve learning efficiency (Bareinboim, Forney, and Pearl, 2015; Lemeire and Dirkx, 2006; Bickel, Brückner, and Scheffer, 2009; Santoro et al., 2016; Lake et al., 2017; Hill et al., 2019; Peters, Bühlmann, and Meinshausen, 2016; Rojas-Carulla et al., 2018; Kansky et al., 2017; Besserve et al., 2018; Schölkopf et al., 2012; Zech et al., 2018; Silva, 2016; Bahadori et al., 2017; Amos, 2008); and, by using causal structure, we can answer naturally interventional questions ("*sample a photo of a woman from my GAN, and give her a moustache*") that cannot be answered without it (Kocaoglu et al., 2018; Goudet et al., 2017).

"The difficulty is to detach the framework of fact—of absolute fact—from the embellishments of theorists..."

— Sherlock Holmes

3

PIGEBaQ: Interpretation via the Gradient Posterior

This is joint work with Dr. Jonathan Downing, Dr. Justin Bewsher, Dr. Michael Osborne and Dr. Stephen Roberts. JD and I conceived of the original idea. JB, initiated derivations and completed with support from JD, and myself. JD, JB, and I led experimental set up and analysis; JD wrote most of the architectural code. I led the first version of validation experiments and JD led the updated second version, which is reproduced here. MO and SR oversaw progress.

3.1 Introduction: associational inference for characterising changes

The first step on the Ladder of Causation is *assocation*. Associational questions enquire how variables covary. Hidden within this search for covariation is usually an implicit causal question: how does Y change if I modify X? The outcome is to construct desired actions on X.

However, we cannot use associational questions to answer causal questions, unless we have further assumptions. Without these assumptions, at best associational questions can be an early step in a chain of analyses that will allow a modeller to move closer to answering causal questions of interest. Here we focus this case, which is frequently observed in the real world: when we have data from a phenomenon we want to study, and we want some insight into the observed relationships between inputs and outcomes of interest.

What, then, does a decision-maker do with associational relationships? Importantly, it should be an aide to further steps that help to establish causal relationships. Having a sense of the size, direction, and uncertainty of relationships of input variables to outcome variables allows one to prioritise some variables for further study, to begin thinking of potential explanations, or to compare to known relationships, for example.

In this chapter, we are specifically interested in one type of associational relationship: the notion of the *gradient*, which describes how changes in the inputs are associated with changes in the outputs. In the sense that the gradient captures a notion of change, it is the associational counterpart of the treatment effect (which is usually the true value of interest when a decision-maker has an intervention they are considering making.)

In this chapter, we study the problem of extracting associational insights of how

outcome variables change over feature space. This can equally be used to estimate the gradient (with respect to features) of measured outcomes as well as of predicted ones from a model. We focus on the distributional characteristics of gradients using Bayesian quadrature, which casts integration as a probabilistic problem. This allows us a greater degree of information about the expected gradient, how the gradient varies, and allows us to identify highly informative new sample points. This also has implications for decision-makers who want to quantify uncertainty of insights, as well as for decision-makers who want to optimally collect new data.

3.1.1 Interpreting models via their gradients

Here, we consider the case of how to make an arbitrary machine learning model interpretable for a real-world policy-maker (politician, CEO, economist, etc.) that wants to understand three questions:

1. **Direction:** Are (potentially) controllable factors positively or negatively associated with outcomes I am interested in understanding?
2. **Impact:** Which factors are most (and least) associated with changes in outcomes?
3. **Confidence:** How certain am I about these relationships? What data do I need to be more confident?

In machine learning terms, we care about the *gradient* of a desirable outcome – that is, the vector of derivatives of the outcome with respect to possible policy actions, of which each component is represented as $\frac{\partial\, outcome}{\partial\, action_i}$. For example, an education minister may be interested in how changing actionable variables such as *{student-teacher ratio, enrollment in extracurricular programs, science funding}* relates to the the outcome of graduation rates at individual high schools. Before deciding what policy to enact,

the minister would want to know the direction and impact of the actions, and how confident they can be about each action.

However, a fundamental problem in gradient evaluation is that the direction, magnitude, and confidence of the gradient *changes* across the input space. For many non-trivial problems, the common assumption of a linear function from input to output space with uniform variance is a poor answer to a likely nonlinear problem with varying degrees of confidence and regions of space that matter more or less. Dividing the space into regions and approximating the derivative within them requires strong a priori knowledge that might be more flexibly addressed with smooth probabilistic priors.

We address the problem of recovering average gradients by casting it as a probabilistic integration task, giving us a pipeline that can take observed data or model outputs and (a) give dimension-wise posteriors of the average derivative, (b) use priors to allow some regions of space to matter more or less, (c) provide Bayesian uncertainty of the average gradient, (d) provide a natural measure of gradient volatility across the space, and (e) allow decision-makers to identify optimal datapoints to collect to maximally reduce uncertainty of the average gradient.

A foundational insight of this paper is that Bayesian quadrature, as an integration technique, is a natural fit to probabilistically average gradients, characterise their distribution, and choose new data points to observe. We introduce PIGEBaQ, an end-to-end BQ pipeline for this task. By treating our prior-weighted average gradient as a random variable, we recover a posterior over the expected gradient. We use the mean of this distribution as our gradient estimate but also use its uncertainty and calculate a closed-form expression for its volatility across the space of integration.

The contributions we make in this paper are to (a) introduce PIGEBaQ and Bayesian quadrature as a natural single tool for this interpretability task, (b) explain how to analytically recover the gradient mean, uncertainty, and volatility (c) show that

it performs comparably to other models used in the interpretability literature. We also discuss an applied example, working thorugh the usefulness of associational inference in the first step towards trustable estimates of causal effects in real-world applications.

3.2 Related Work

3.2.1 Interpretability in machine learning

As machine learning techniques become more integrated in the real world, being able to interpet models and their application becomes more important (Vellido, Martin-Guerroro, and Lisboa, 2012). Governing bodies have looked into enforcing the need for interpretability (Goodman and Flaxman, 2017). Machine learning researchers are increasingly building interpretability into their models from the beginning (Kim, Shah, and Doshi-Velez, 2015). There is no standard framework for interpretability, though some have been proposed (Lipton, 2018; Doshi-Velez and Kim, 2017). Generally, these frameworks emphasise understanding concepts such as *trustworthiness, causality, generalisability, meaningfulness, fairness* and *informativeness*. There are commonly-used techniques, from bespoke visualisation to post-hoc model simplification (Lakkaraju, Bach, and Leskovec, 2016; Hara and Hayashi, 2016), to instance- or model-based approaches relating inputs to parameters or outputs (Kumar, Wong, and Taylor, 2017; Barratt, 2017). A good overview of explaining black box models is provided in (Guidotti et al., 2018).

3.2.2 Gradient methods

Gradient methods are often a desirable interpretability tool, as the change in a value of interest (model parameter, outcome value, evaluation metric, or otherwise) can be

useful information for decision makers. Analysing the gradient is not a new method (see Engelbrecht, Cloete, and Zurada (1995)) and is sometimes categorised as *sensitivity analysis*, *perturbation methods*, *conditional expectations analysis*, and in one sense *influence function analysis* (Koh and Liang, 2017; Ribeiro, Singh, and Guestrin, 2016a), where individual features are altered (or all of them, at the same time, to produce point-specific gradients). Gradient methods for interpretability vary by characteristic. Some are fundamentally *local*, in that they examine gradients within regions of interest, such as in (Ribeiro, Singh, and Guestrin, 2016b; Baehrens et al., 2010; Babibker and Goebel, 2017). Other gradient analyses vary by unit (as one might be interested in *relative rank* rather than *magnitude*, or vice versa). Using gradients to interpret machine learning models has been successfully applied in health outcome prediction (Choi et al., 2016), recommender systems (Hechtlinger, 2016), visual decision making (Simonyan, Vedaldi, and Zisserman, 2014), local visual decision analysis (Ribeiro, Singh, and Guestrin, 2016b), and elsewhere. Gradient methods are often powerful for interpretability, yet the difficulties of using a flexible model while retaining interpretable gradients, quantifying uncertainty and volatility, and subsetting in feature space, remain. We address this gap using novel averaging of gradients using Bayesian quadrature.

3.2.3 Bayesian quadrature for principled integration

Bayesian quadrature (O'Hagan, 1991; Rasmussen and Ghahramani, 2003) is a fully Bayesian approach to integration. Bayesian quadrature operates by treating an integral as a random variable. By placing a Bayesian prior, like a Gaussian process (Rasmussen and Ghahramani, 2003), over the integrand, the linear operation of integration implies a posterior over the integral. The advantages of Bayesian Quadrature for integration are threefold. First, in an active learning or sampling-based integration setting, BQ

can lead to faster convergence than standard Monte Carlo approaches (Gunter et al., 2014; Osborne, 2010; Osborne, Garnett, Roberts, et al., 2012). Because Bayesian Quadrature provides the variance over the posterior of the integral, the posterior variance becomes a natural part of an acquisition function for sampling, or an indicator of convergence. Second, the resulting posterior gives distributional information about the integral that can be useful to quantify uncertainty, if the uncertainty of the integral is of interest. In a human decision-making context, this can be very valuable for decision-makers to calibrate their confidence. Third, the Bayesian approach allows one to use a prior density to weight values if the modeller has knowledge of the relevance of different parts of feature space. For that reason, Bayesian quadrature has been successfully used in applications like sensitivity analysis for electronic circuit design (Pfingsten, 2006), decision making in reinforcement learning (Ghavamzadeh, Engel, and Valko, 2016), or predicting storm surges from computationally-intensive hurricane simulations (Toro et al., 2010).

3.3 PIGEBaQ: Principled Interpretation of Gradient Evaluation using Bayesian Quadrature

Here we introduce how we can use Bayesian Quadrature to recover a posterior over the expected gradient and the volatility of the gradient. In Bayesian Quadrature, the integrand requires a Bayesian prior, for which we use a Gaussian process prior over the derivative of a function $f : X \to Y$, with X, Y defining some input and output space respectively. While f can be any function (so long as we have access to samples of its derivative), we illustrate the method by placing a Gaussian process prior over f so that $\frac{\delta f(\boldsymbol{x})}{\delta \boldsymbol{x}}$, its derivative with respect to any point $\boldsymbol{x}$, is a Gaussian

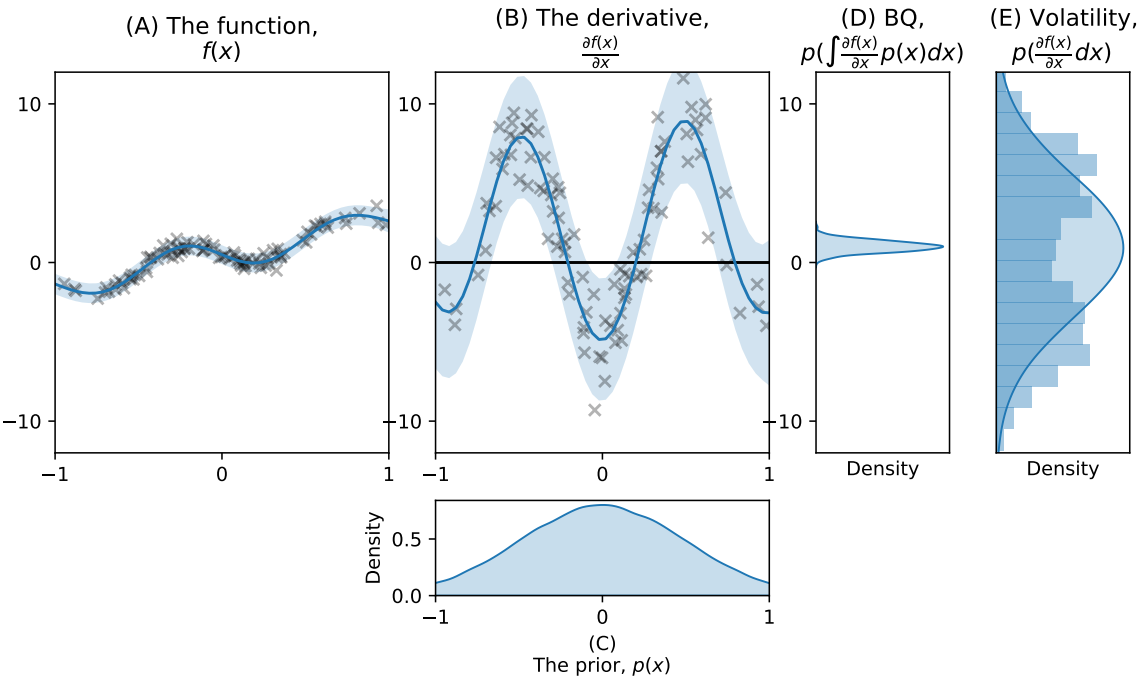


Fig. 3.1: The PIGEBaQ pipeline. (A) the data is fit with some model $f(\boldsymbol{x})$; in this work, we use a Gaussian process; (B) the model is differentiated; if the model is not a Gaussian process, we fit a GP to samples of the derivative, both resulting in $\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}$, which is then weighted by the prior from (C). This differentiated Gaussian process allows us to calculate the volatility of the gradient. In step (D), Bayesian Quadrature gives us the distribution of $\int \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} p(\boldsymbol{x}) d\boldsymbol{x}$, our distribution of the average gradient; (E) we calculate the volatility of $\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}$ as a measure of gradient change over the space (histogram is the empirical distribution of the gradient, and the smooth density is our first- and second-moment matched volatility).

process and can be analytically derived. We then perform Bayesian Quadrature on this derivative by using a Gaussian mixture prior.

3.3.1 Gaussian processes and their derivatives

In PIGEBaQ, we are interested in a posterior over the gradient, and its integral. In order to recover this, we need a posterior over the function of which we take the gradient. Naturally, we want to express uncertainty over what this function is, and what its values $f(\boldsymbol{x}_*)$ are at some unobserved point $\boldsymbol{x}_*$. We can then project from

function space to output space and recover a predicted output value y_*:

$$p(y_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \int p(y_* \mid f, \boldsymbol{x}_*) p(f \mid \boldsymbol{X}, \boldsymbol{y}) df \tag{3.1}$$

where $\boldsymbol{X}$ is a collection of observed points $\boldsymbol{x}$, and $\boldsymbol{y}$ is a vector of their corresponding observations. Note that the output can also be a vector $\boldsymbol{y}_*$ in the multi-output case, which we visit in Chapter 5. Intuitively, $p(y_* \mid f, \boldsymbol{x}_*)$ defines a likelihood function, and $p(f \mid \boldsymbol{X}, \boldsymbol{y})$ defines a distribution over functions that generate function values $\boldsymbol{y}$.

Gaussian processes

A Gaussian process, GP, defines a prior over functions (Rasmussen and Williams, 2006). The key assumption a GP makes is that any finite subset of function values are jointly (multivariate) Gaussian. That is, $[f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)] \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$ where $\boldsymbol{\mu}$ is a mean vector $[\mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_n)]$ defined by some *mean function* μ which operates on $\boldsymbol{x}$, and $\boldsymbol{K}$ defines a positive semi-definite *covariance matrix* where each entry $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is defined by a *kernel function* κ.[1]

An important intuition here is that a GP is fully characterised by its mean function μ and kernel κ. Typically, we assume $\mu(\boldsymbol{x}) = 0$. The behaviour of the GP is then relegated in full to the structure of K as defined by κ.

A conventional κ, and the one we used to derive closed form expressions in this work, is the Exponentiated Quadratic Kernel:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{x}')^T \Lambda^{-1} (\boldsymbol{x} - \boldsymbol{x}')}{2}\right) \tag{3.2}$$

where σ^2 is an output variance hyperparameter and Λ is a diagonal matrix of

[1] Note that a GP defines a function as a distribution over its *values*. In our case, values can occur at infinitely many locations in input space $\boldsymbol{X}$. As a result, a GP can be infinitely-dimensional, scaling with the number of datapoints. For this reason, a GP is a nonparametric model.

lengthscale hyperparameters. Of interest is often the posterior predictive distribution $p(y_* \mid \boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y})$ at some new point $\boldsymbol{x}_*$, which can be calculated in closed form:

$$p(y_* \mid \boldsymbol{x}_*, \boldsymbol{X}, y) = \mathcal{N}(y_*; \mu_*, \boldsymbol{\Sigma}_*) \tag{3.3}$$

$$\mu_* = \mu(\boldsymbol{x}_*) + \boldsymbol{K}_*^T \boldsymbol{K}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}) \tag{3.4}$$

$$\Sigma_* = \kappa(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{K}_*^T \boldsymbol{K}^{-1} \boldsymbol{K}_* \tag{3.5}$$

where $\boldsymbol{K}$ is the covariance matrix of entries $\kappa(\boldsymbol{x}, \boldsymbol{x})$ for all observed $\boldsymbol{x}$, $\boldsymbol{K}_*$ is the vector of $\kappa(\boldsymbol{x}, \boldsymbol{x}_*)$ for all observed $\boldsymbol{x}$. Intuitively, prediction in a GP is defined by how similar one point is to another, as given by the kernel function.

The final question is how to choose $\boldsymbol{\theta}$, the set of hyperparameters. These hyperparameters depend on the kernel; in this case, $\boldsymbol{\theta} = \{\sigma^2, \Lambda\}$. In practice, and in this work, they are learned through maximising the marginal log likelihood, $\int_{\boldsymbol{\theta}} p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$. In our work, we do this through gradient descent using GPflow (de G. Matthews et al., 2017).

Importantly, the performance of the model can be sensitive to the choice of the K and the values of $\boldsymbol{\theta}$. In practice, kernel choice depends on what covariance relationship the modeller believes should be modelled in the data. In our case, we use the Exponentiated Quadratic Kernel because we model the scenario where we are using some arbitrary f of which we have highly confident point-wise derivative information, and this kernel enables reliable recovery of arbitrary functions with smooth interpolation over regions of uncertainty. Further, in this work, we place wide priors over hyperparameters based on what seems reasonable given the data, and we then use cross-validation with random restarts to confirm their suitability. A more general approach would use a more sophisticated sampling of hyperparameters, such as Bayesian optimisation.

There are several reasons why a GP is useful in our application:

1. it gives a principled posterior distribution;
2. it requires no parametric or structural assumptions, apart from the kernel and mean functions;
3. with conventional kernels, uncertainty grows as the model extrapolates away from regions of space for which it has observations;
4. the kernel function can be composed of several kernel functions, allowing for flexible covariance structure.
5. a GP is closed under differentiation if its kernel is differentiable.

Derivatives of Gaussian processes

The function we are truly interested in is the derivative of the function that generates the data. The derivative is the function that expresses our associational measure of change with respect to input space. However, because it is much rarer to have a large amount of true derivative information (and sometimes difficult to get gradient information from a model), we choose to first fit a function on observed space which we then hope to differentiate. (We note that if point-wise gradients from a different function f are available, we can fit a Gaussian process to those gradients and then use the PIGEBaQ approach, too.)[2] Here, we differentiate a Gaussian process with an exponentiated quadratic kernel, and receive $\frac{\partial f_*}{\partial \boldsymbol{x}_*}$, which is also a Gaussian process.

Noting that differentiation is a linear operator, the derivative of the mean function can be performed as normal. Assuming a zero mean prior (that is, $\mu(\boldsymbol{x}) = 0$), we remind ourselves that the posterior mean function is:

[2]As discussed below, we can incorporate observations of gradients into a derivative of a Gaussian process if we like.

$$\mu_* = \boldsymbol{K}_*^T \boldsymbol{K}^{-1} \boldsymbol{y} \tag{3.6}$$

Noting that only $\boldsymbol{K}_*$ depends on the unobserved test points $\boldsymbol{x}_*$ that we will differentiate with respect to, we can let $\boldsymbol{\alpha} = \boldsymbol{K}^{-1}\boldsymbol{y}$, and just differentiate $\boldsymbol{K}_*$:

$$\frac{\partial \boldsymbol{K}_*}{\partial \boldsymbol{x}_*} = \frac{\partial K(\boldsymbol{x}_*, X)}{\partial \boldsymbol{x}_*} \tag{3.7}$$

$$= \frac{\partial}{\partial \boldsymbol{x}_*} \left\{ \sigma^2 \exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{x}_*)^T \Lambda^{-1} (\boldsymbol{x} - \boldsymbol{x}_*)}{2} \right) \right\} \tag{3.8}$$

$$= -\Lambda^{-1}(\boldsymbol{x}_* - X)\boldsymbol{K}_* \tag{3.9}$$

where $\odot$ denotes the element-wise product, implying:

$$\mathbb{E}\left[\frac{\partial f_*}{\partial \boldsymbol{x}_*}\right] = -\Lambda^{-1}(\boldsymbol{x}_* - \boldsymbol{X})(\boldsymbol{K}_* \odot \boldsymbol{\alpha})\,. \tag{3.10}$$

Omitting laborious details, we find the variance of the derivative of $\frac{\partial f_*}{\partial \boldsymbol{x}_*}$ as:

$$\mathbb{V}\left[\frac{\partial f_*}{\partial \boldsymbol{x}_*}\right] = \frac{\partial^2 K_*}{\partial \boldsymbol{x}_* \partial \boldsymbol{x}_*} - \frac{\partial K\left(\boldsymbol{x}_*, X\right)}{\partial \boldsymbol{x}_*} K^{-1} \frac{\partial K_*}{\partial \boldsymbol{x}_*} \tag{3.11}$$

In sum, we can recover the posterior of the derivative of the Gaussian process so long as we can twice differentiate the kernel. We use these conclusions to instead find the distribution of the integral of this derivative.

3.3.2 Bayesian quadrature

Bayesian Quadrature involves recovering $p(\boldsymbol{I} \mid f(\boldsymbol{x}))$, the posterior over the value of an integral $\boldsymbol{I}$, by placing a Bayesian prior over the integrand $f(\boldsymbol{x})$. To do so, we need a prior over $f(\boldsymbol{x})$ that is implies a distribution under linear transformation, because

integration is a linear operation. Because GP s are closed under a linear transformation, we can use them as a prior over $f(\boldsymbol{x})$ to recover a posterior over the integral. With a GP prior over $f(\boldsymbol{x})$ such that $f(\boldsymbol{x}) \sim GP\left(f(\boldsymbol{x}); \mu(\boldsymbol{x}), K(\boldsymbol{X}, \boldsymbol{X})\right)$, we have:

$$\int_{\mathcal{X}} f(\boldsymbol{x}) d\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{I}; \int_{\mathcal{X}} \mu(\boldsymbol{x}), \int_{\mathcal{X}} \int_{\mathcal{X}'} K(\boldsymbol{x}, \boldsymbol{x}') \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{x}'\right) \tag{3.12}$$

Bayesian Quadrature can be used to perform any integration task. Most frequently, it provides a probabilistic acquisition function as an alternative to Monte Carlo methods. Alternatively, one can take it out of the active learning context and use it to compute integrals of interest, like a marginal distribution.

3.3.3 PIGEBaQ: extending BQ for derivatives

The core motivating insight behind PIGEBaQ is that the expected gradient, as used for interpretability, is an integral that benefits from Bayesian Quadrature's flexible probabilistic nature.

We note that the expected gradient of a function $f(\boldsymbol{x})$ is calculated as:

$$\mathbb{E}[\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}] = \int_{\mathcal{X}} \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{3.13}$$

PIGEBaQ uses Bayesian Quadrature to calculate and interpret this integral by:

1. placing a GP prior over $f(\boldsymbol{x})$, letting $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$ be observations corresponding to observed values in the real world or from a potentially black box function handle;
2. analytically calculating $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$;
3. introducing a prior $p(\boldsymbol{x})$ and weighting $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$;
4. integrating the integrand $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} p(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ across input space;

5. calculating the volatility of $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$.

An important distinction between PIGEBaQ and vanilla Bayesian Quadrature is that the integrand is the derivative of $f(\boldsymbol{x})$, not $f(\boldsymbol{x})$ itself. PIGEBaQ is motivated by the need to analytically calculate the posterior of the integral when we are interested in the derivative of the function over which the prior is placed.

Model

First, we need to choose a prior to compose $p(\boldsymbol{x})$. Most commonly, a single Gaussian is chosen such that $p(\boldsymbol{x}) := \mathcal{N}(\boldsymbol{x}; \boldsymbol{\nu}, \boldsymbol{\Psi})$, though in practice any prior can be used by following an importance reweighting scheme (Gunter et al., 2014). For illustrative purposes, we choose a mixture of Gaussians prior:

$$p(\boldsymbol{x}) = \sum_{m=1}^{M} \pi_m p(\boldsymbol{x}|\theta_m)\mathrm{d}\boldsymbol{x}, \tag{3.14}$$

For this derivation, we give the GP a zero-mean prior use the exponentiated quadratic kernel, again:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{x}')^T \Lambda^{-1} (\boldsymbol{x} - \boldsymbol{x}')}{2}\right), \tag{3.15}$$

where σ is the output variance hyperparameter and Λ is a matrix of lengthscale hyperparameters. Derivatives of the kernel are given in Appendix A.1.

The posterior distribution of $\boldsymbol{I}$ can now be computed using properties of the GP. We instead replace the integrand with $f(\boldsymbol{x})$. The posterior mean and variance is:

$$\mathbb{E}[\boldsymbol{I}] = \sum_{i=1}^{N} \boldsymbol{z}^T \frac{\partial K}{\partial \mathbf{x_i}}^{-1} \frac{\partial f(\mathbf{x_i})}{\partial \mathbf{x_i}}, \tag{3.16}$$

$$\mathbb{V}[\boldsymbol{I}] = \int_{\mathcal{X}} \int_{\mathcal{X}'} K_{\boldsymbol{x},\boldsymbol{x}'}(\boldsymbol{x}, \boldsymbol{x}') p(\boldsymbol{x}) p(\boldsymbol{x}') \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{x}' - \boldsymbol{z}^T \frac{\partial K}{\partial \mathbf{x}}^{-1} \boldsymbol{z}. \tag{3.17}$$

where N is the number of model observations and $K_{\boldsymbol{x},\boldsymbol{x}'}$ is the second derivative of the kernel. The components of $\boldsymbol{z}$ are given by:

$$\boldsymbol{z}_n = \sum_{m=1}^{M} \frac{\pi_m \sigma^2 |\Lambda|^{1/2}}{|\Lambda + \Sigma_m|^{\frac{1}{2}}} (\Lambda + \Sigma_m)^{-1} (\boldsymbol{\mu}_m - \boldsymbol{x}_n) \exp\left(-\frac{(\boldsymbol{x}_n - \boldsymbol{\mu}_m)^T (\Lambda + \Sigma_m)^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_m)}{2} \right). \tag{3.18}$$

The second term of the variance be computed using $\boldsymbol{z}$. The first term is a combination of the lengthscale along with the prior means and variances. Note that the variance is independent of the function outputs. Full derivations of both terms are provided in the Appendix A.1.

A final benefit of Bayesian Quadrature is that it can be used for active learning to identify new points to sample that reduce uncertainty over the integral (Osborne, Garnett, Ghahramani, et al., 2012). In our context, this might mean a decision-maker choosing to prioritise one data-collection approach over another, run one intervention over another, or submit a particular query to an expensive function. We do not expand on this in this work but we remark on it as another source of support to a decision-maker. If they aren't confident in the recovered expected gradient relationship, they can instead use PIGEBaQ's output to choose to observe another data-point to become more so.

Volatility of the derivative

Bayesian quadrature allows us to compute a principled average derivative and an uncertainty estimate of the average. However, this uncertainty quantifies only our uncertainty around the estimate of the average derivative, not the variation in the average derivative across the input space. Imagine a situation when the model is completely certain the average derivative is zero over a domain, however, the derivative could vary drastically (for example, a sine curve). This can help a decision-maker, for example, understand the monotonicity of the underlying function. Thus, we consider the volatility of the derivative, which we understand as the variation of the derivative over the input space. We visualise the intuition behind this in Figure 3.2.

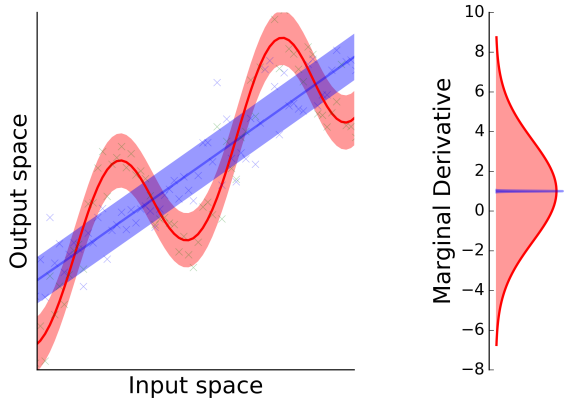


Fig. 3.2: An example of when volatility is useful information. (left:) Two underlying functions with the same average gradient. (right:) The distribution of their gradients in output space. Note that while the gradients have the same mean, the sinusoidal gradient process is more variable (even negative), implying different policy effects depending on the location of a policy change in the input space.

To compute the volatility of the derivative, we take the attention weighted derivative GP and marginalise out $\boldsymbol{x}_*$. As standard in GP regression we now deal with noise corrupted observations, $\boldsymbol{y}$, of the true function $\boldsymbol{f}$. The distribution of the volatility is non-Gaussian:

$$p(\boldsymbol{y'}) = \int_{\mathcal{X}} p\left(\frac{\partial y}{\partial \boldsymbol{x}_*}\middle| \boldsymbol{X}, y, \boldsymbol{x}_*\right) p\left(\boldsymbol{x}_*\right) d\boldsymbol{x}_* \tag{3.19}$$

$$= \int_{\mathcal{X}} p(\boldsymbol{y'} \mid \boldsymbol{X}, y, \boldsymbol{x}_*) p\left(\boldsymbol{x}_*\right) d\boldsymbol{x}_* \tag{3.20}$$

Where $p(\boldsymbol{y'} \mid \boldsymbol{X}, y, \boldsymbol{x}_*) = \mathcal{N}(\boldsymbol{y'} \mid \boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$, with:

$$\boldsymbol{\mu}_{\boldsymbol{x}} = \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}_*)} \boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})} \boldsymbol{y} \tag{3.21}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{x}} = {} & \boldsymbol{K}_{\boldsymbol{y'}(\boldsymbol{x}_*),\boldsymbol{y'}(\boldsymbol{x}_*)} + \\ & - \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}_*)} \boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})} \boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}_*)}\end{aligned} \tag{3.22}$$

Where, to ease exposition, we have introduced the kernel notation: $\boldsymbol{K}_{A(\mathcal{C}),B(\mathcal{D})} = \operatorname{cov}(A_{\mathcal{C}}, B_{\mathcal{D}})$. This is read as: the covariance between functions A and B with input points to A being $\mathcal{C}$ and input points to B being $\mathcal{D}$.

One can empirically approximate Equation (3.20) if desired, as it is likely intractable. However, because we are mostly interested in the amount amount of feature space that the derivative is above and below zero, we analytically approximate Equation (3.20) with a Gaussian distribution $q(\boldsymbol{y'}) = \mathcal{N}(\boldsymbol{y'} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$, found by minimising the Kullback-Leibler divergence between $p(\boldsymbol{y'})$ and $q(\boldsymbol{y'})$:

$$\boldsymbol{\mu} = \mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}] \tag{3.23}$$

$$\boldsymbol{\Sigma} = \mathbb{V}_{p(\boldsymbol{y'})}[\boldsymbol{y'}] \tag{3.24}$$

$$= \mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}\boldsymbol{y'}^T] - \mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}]\mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}]^T. \tag{3.25}$$

Full derivations of all terms are given in Appendices A.2 and A.3.

We see the recovery of volatility as a critical component of the model for interpretability. For that reason, it is important to have a distribution over functions of the derivative (via a GP), rather than point wise-estimates that may be highly locally variable. This is also a key reason why, in this work, we differentiate f and then integrate; we need the derivative function to calculate volatility. If f does not take the form of a GP but can be differentiated, then a GP prior can be placed over $\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$ and Bayesian Quadrature then applied.

3.3.4 PIGEBaQ in applied decision-making

The first step of PIGEBaQ places a GP prior over $f(\boldsymbol{x})$. We are considering the case where decision-makers use the output of $f(\boldsymbol{x})$, and its gradient $f(\boldsymbol{x})$ information, to make decisions. It is then worth considering the intuition behind why one would bother with putting a GP prior over $f(\boldsymbol{x})$ in the first place.

First, if $f(\boldsymbol{x})$ is hard to know and expensive to query, a probabilistic interpolation can be useful. For example, $f(\boldsymbol{x})$ might be a likelihood function or a very complex and costly simulation. As previously described, BQ can speed up convergence to the true posterior. This approach can be valuable, for example, where $\boldsymbol{x}$ defines a vector of input parameters to which the output may be highly sensitive.

Second, take the case where $f(\boldsymbol{x})$ is some non-probabilistic black box machine learning model such as a neural network (as are increasingly used as tools in decision-making). It is often the case that the data collected by a decision-maker is more confident about a model's performance in some regions of feature space, such as those where data is collected. A GP prior has the benefit of confidently interpolating in that region of input space while expressing uncertainty in other regions of input

space. In this case, we can use the GP prior to take advantage of where *any* model is confident while also representing where it is less confident. This is valuable especially in sensitive applications, where uncertainty may encourage a decision-maker to not use the model for their decision. It can also enable making decisions with respect to Bayesian decision theory.

Third, Gaussian processes have been shown to be useful priors for decision-making scenarios where there is interventional as well as observational data (Silva, 2016). For a policy-maker that strives for identification of the true response curve while learning from observational data, the value of probabilistically combining both types of data is compelling.

Last, because derivatives of Gaussian processes are themselves Gaussian processes, if derivative information is known – such as prior information about the treatment effect of one variable on another – we can also incorporate that information to inform the derivative process (Solak et al., 2003).

3.4 Validation

Here, we compare PIGEBaQ to a variety of gradient-based interpretation models which are flexible, inflexible, probabilistic, and not probabilistic. We show that PIGEBaQ tends to outperform comparable models, at least for this sample size and dimension range. We discuss PIGEBaQ's limitations going forward.[3]

[3]I designed and completed the first version of the validation experiments. However, several years later, Dr. Jonathan Downing (co-author on this work) updated the validation experiments with more recent interpretability comparisons with additional visualisations. These results have been gratefully reproduced here with Dr. Downing's permission, but the design of the experiments remains the same as the original. For a slightly different discussion, see Dr. Downing's (very enjoyable) thesis.

3.4.1 Data

We synthesise several functions for which we know the gradient analytically and which represent ranges of behaviour across feature space. The functions are specified in Table 3.1. We set hyperparameters $\beta_i \sim \mathcal{U}(0,1)$ and $\gamma \sim \mathcal{U}(0,1)$. We draw n points $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ in d dimensions where $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]$ with $\mu_i \sim \mathcal{U}[-1,1]$ and $\boldsymbol{\sigma}^2 = [\sigma_1^2, \ldots, \sigma_d^2]$ with $\sigma_i^2 \sim \mathcal{N}(0,1)$.

Table 3.1: Three synthetic functional forms chosen to test PIGEBaQ.

FN.	FORM
f_1	$\boldsymbol{\beta}^T\boldsymbol{x}$
f_2	$\sum_i^M \sin(\beta_i x_i) + \sin(\gamma \prod_i^M x_i)$
f_3	$\sum_i^M \sin(\gamma_i x_i) + \beta_i x_i$

For each function, four experiments were conducted for each combination of dimensionality (8 or 16) and number of samples (256 or 512) from the true function. Each experiment was repeated 20 times with different samples.

3.4.2 Comparison models

We compare the PIGEBaQ model (labelled PQ in figures 3.3 and 3.4) to three benchmark models. The first is a 3-layer Deep Neural network (DNN) with layer-wise L_2-regularisation. The second model is a standard ordinary least squares linear model (Lin). The final model is the LIME Python package, which internally uses ridge regression. These three models are popular within the gradient interpretation literature. The first is more commonly both the model and gradient model, while the latter two are more often model-agnostic surrogate models. The first two models are fit on the observed data and differentiated at observations, while the last provides

point-wise gradient approximations natively.[4]

3.4.3 Metrics

We compare models by the root-mean-squared error (RMSE) of the average gradient relative to the true gradient ($\mathcal{L}_\nabla$), and RMSE of the average volatility of the gradient relative to the true volatility ($\mathcal{L}_{\sigma^2}$), evaluated at every observation:

$$RMSE = \sqrt{\Sigma_i^n \left(\hat{y}_i - y_i\right)^2}, \tag{3.26}$$

where $\hat{y}_i$ is the outputted value of the model (gradient or volatility) and y_i is the true value for observation i.

Of interest is also the relative gradient of the features, ranking features from most positive and most negative effect on the output. To evaluate this, we use Kendall's τ rank correlation measure, a measure of the ordinal agreement between all possible pairs of points $(\hat{y}_i, y_i)$; a τ of 1 indicates a model perfectly ranks points compared to the true ranking, and a τ of -1 indicates a model never ranks points correctly. In this case, the points are the estimated gradients and volatilities compared to the ground truth gradients and volatilities.

3.4.4 Results

PIGEBAaQ performs well, with its distributions of gradient recovery and ranking errors being consistently higher performing (in terms of location and variance) than the other comparison models. In f_1, the linear function, PIGEBaQ matches the linear model (which should exactly recover the true gradient); the complex nature of the

[4]Notably, we did not include SHapley Additive exPlanations (SHAP) as SHAP is not a gradient model per se, but we note its increasingly common usage for feature effect and ranking analysis.

neural network makes it prone to initialisation randomness; counterintuitively, despite being a linear model, LIME suffers because it stochastically samples features and computes gradients on that subset of features many times for each observation. As functions become more complex in f_2 and f_3, the neural network suffers, likely due to small data; LIME, as an essentially piece-wise linear approximation of the gradient, is likely heavily influenced by local gradients while it ignores global model structure. The linear model's performance is relatively unchanged.

This is pronounced in the error with respect to recovering the true average gradient and volatility. It is worth considering why this is the case. Compared to to the linear estimator, PIGEBaQ allows for the essentially structure-agnostic GP to model variations in the gradient. This is useful for any nonlinear function, as in f_2 and f_3. Alternatively, LIME, which uses a piecewise gradient explanation, is highly influenced by local gradients while ignoring global model structure. Last, a DNN is very susceptible to high variation across input space; it requires strong modelling assumptions to overcome this challenge.

PIGEBaQ also performs well in recovering the the relative rank of features. While PIGEBaQ may not strictly outperform the other models, it is consistently the highest performing.

A challenge we observed is expanding the task to higher dimensions (Duvenaud, 2012). Empirically, we observed PIGEBaQ perform worse once we pushed beyond the original validation experiments. This was expected, as it has been observed that Bayesian Quadrature struggles in high dimensions. This is partly due to the prior, which suffers the curse of dimensionality. It is mostly due to the Gaussian process prior, which in higher dimensions with fewer samples is prone to overfitting due to the need to learn dimension-wise lengthscales. Of course, the curse of dimensionality affects all models, but is less likely to affect ones that make limiting but sensible

structure assumptions than ones that are highly flexible.

It is also worth noting that our model is the only one to produce an analytical posterior of the average gradient and volatility. While measures of uncertainty can be approximated, the approximation scheme introduces an additional source of uncertainty. We propose that decision-maker would favour a model whose uncertainty estimates are verifiably produced, even in (and especially in) the case where it is highly uncertain.

In summary, PIGEBaQ provides a good balance between low average gradient error and low volatility error. At the same time, it provides additional information about uncertainty, which can calibrate a decision-maker's sense of confidence and allow for guiding sampling for decreased uncertainty.

3.4.5 Limitations

However, there are situations when one should not use PIGEBaQ due to inherent limitations in its design. First, it is computationally costly, in that fitting a Gaussian process has $\mathcal{O}(n^3)$ complexity, and then further hyperoptimisation is generally desirable. The benefit of this complexity is an analytical distribution over gradient values. The modeller should think carefully about whether an approximate expected gradient estimate – for example, a method like LIME, or an ensemble of linear models fit to partitions in feature space – would satisfy the modelling criteria.

Further, the output of the model – especially the derivatives – can be sensitive to the choice of hyperparameters. A sufficiently short lengthscale for a exponentiated quadratic kernel, for example, may have the effect of overfitting to noise in the data. In an extreme case, while the true underlying gradient is monotonic and positive (variations around which are only due to noise), a small lengthscale squared exponential kernel may overstate the volatility of the gradient. In this case, it would give the gradient a

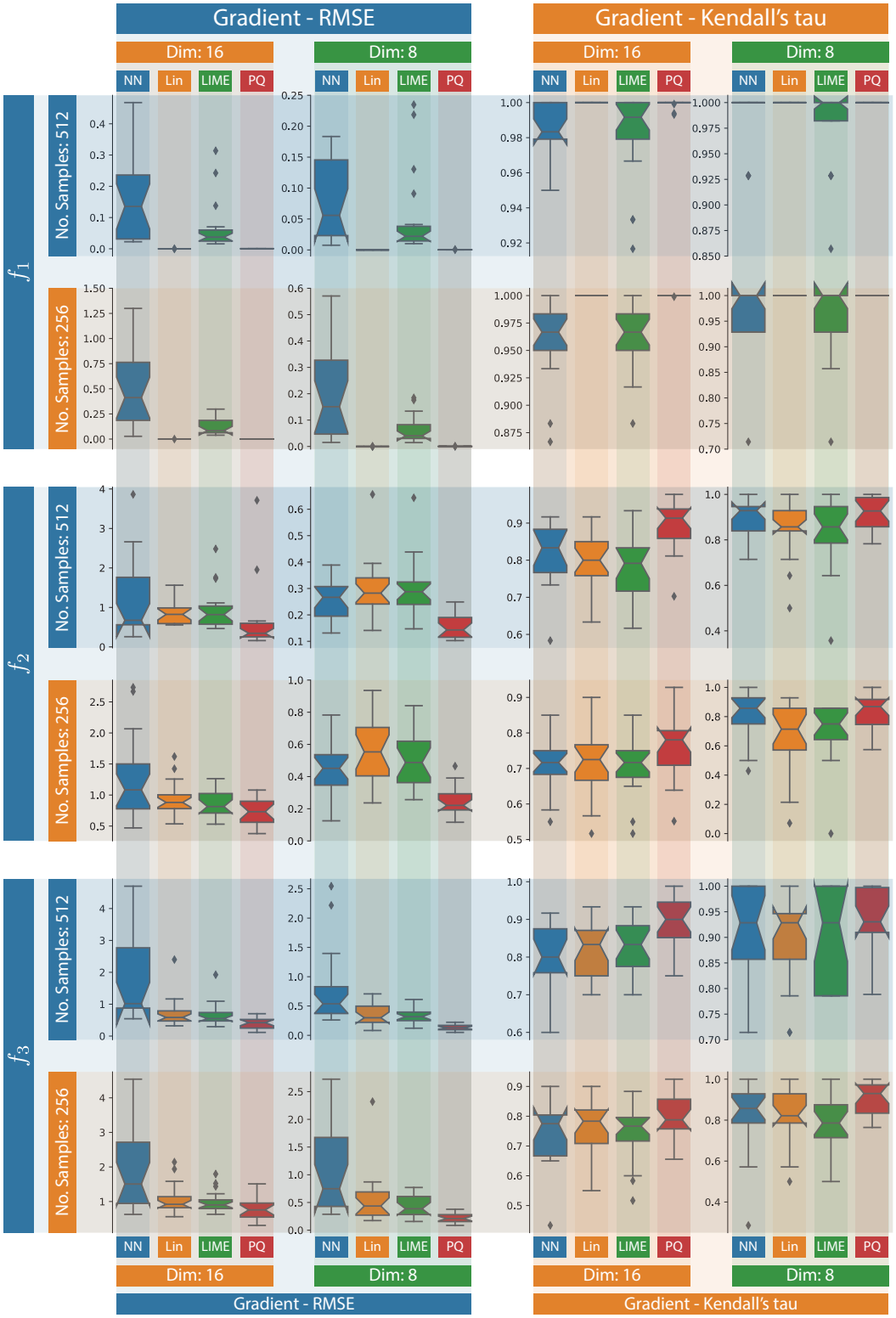


Fig. 3.3: Boxplot comparison of gradient interpretability method performance on synthetic data, measuring how well they recover the average gradient. Average gradient RMSE and Kendall's τ reported.

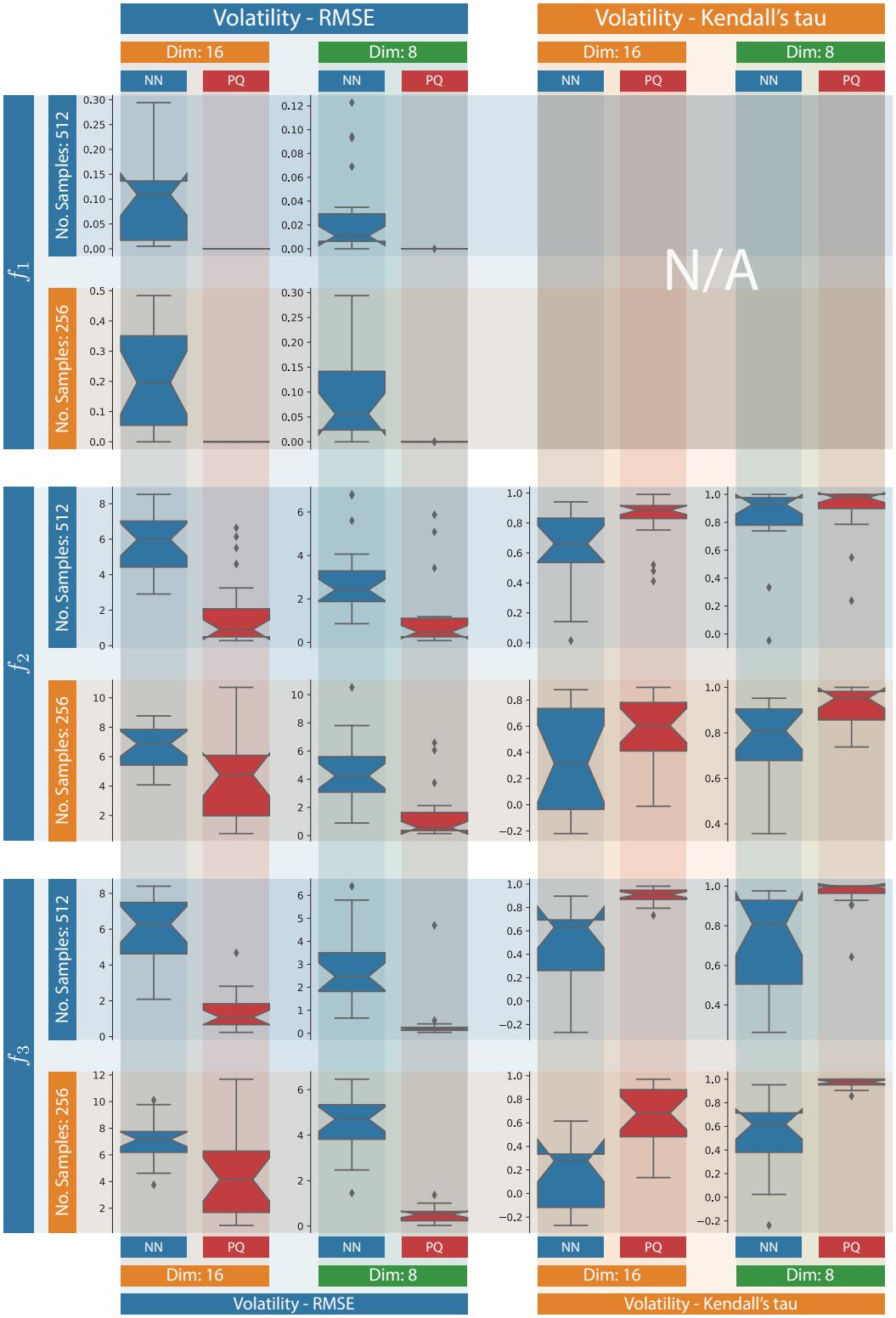


Fig. 3.4: Boxplot comparison of gradient interpretability method performance on synthetic data, measuring how well they recover volatility. RMSE and Kendall's τ reported.

large probability of being negative as well as positive, and it would identify the gradient as being highly variable around observed points. As a result, hyperparameter choice is critical. The standard guidelines, as described in Section 3.3.1, apply, but the modeller should think carefully about kernel choice in the context of how $f(\boldsymbol{x})$ changes.

3.5 Application: educational outcomes in the United States

To show PIGEBaQ in practice, and demonstrate the value of its features when making a decision, we consider the case of improving post-college economic performance. With almost $1.5 trillion U.S. dollars in student debt held today (Reserve, 2018), and a declining wage premium of college attendance, increasing post-college economic success is an important policy problem. We consider the case of a national education policymaker with the desire to incentivise college-level academic or financial changes to improve economic performance.

3.5.1 Data

We use data on 4,148 colleges in 2016 by provided by the College Scorecard from the U.S. Department of Education (D.O.E., 2017). We use the median student income 6 years after college as our target variable. We use 18 actionable, non-sparse features (having trained with an additional 14 control features for a total of 32 features) that describe the academic and financial makeup of a post-secondary education. Our underlying function f is drawn from a Gaussian process such that $f : \boldsymbol{X} \to \boldsymbol{Y}$, where $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{32}\}$ is our set of feature descriptors and $\boldsymbol{Y} = \boldsymbol{y}_{\boldsymbol{income}}$ is our economic performance outcome variable.

Table 3.2: The action derivatives with the largest magnitudes

	AVG. $\frac{\partial f}{\partial \boldsymbol{x}_i}$
% Degrees in STEM	6.70
% Degrees in Business	6.07
% Degrees in Health	1.84
Instructional expenditures / student	0.38
% Degrees in Law	0.26
Net Tuition/Student	0.19
% Degrees in Arts/Soc Sci	−3.85
% Undergrads on a Loan	−15.54
% Undergrads on Pell Grant	−34.98
% Degrees in Trades	−40.45

3.5.2 Prior

In this case, we choose a prior $p(\boldsymbol{x})$ that weights the gradient process by the size of the college within it. This reflects that we would like our average gradient to be more informed by regions in the space with more students, as the policy goal is to affect the largest number of students. We first sample from the input space proportionally to the number of students in each college. For each college c, we draw $n_c \propto n_c^s$ samples where n_c^s is the number of students in college c. College samples $\boldsymbol{x}_c$ are drawn such that $\boldsymbol{x}_c^{'} \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu_c}$ is college c's feature vector and $\boldsymbol{\Sigma}$ is the diagonal covariance matrix of features over all colleges. We then fit a 10-mixture Gaussian Mixture Model over the sampled space as our prior.

3.5.3 Evaluating actions

We consider the top dimension-wise average derivatives largest in magnitude. Results are shown in Table 3.2. We see that an understanding of the volatility and the uncertainty allows the policy maker to show that while a lower *% Degrees in Trades* is associated with the most positive increase in income, it is highly volatile relative to

a lower *% of Undergraduates who receive a Pell Grant* but not much more strongly associated. Alternatively, a policy maker may examine the effect of *Instructional expenditure per student* rather than the proportion of health-related students, as the model is more certain about the average derivative of the former.

What should a policymaker do with this information? First, they should not treat these associations as interventional outcomes. The above statements do not equate to the statements *"incomes will rise if I decrease the number of students in trades or who receive a Pell Grant."* In fact, the real-world outcomes of intervening in such a way might be a *decrease* in incomes. For example, it might be that students from lower income families and geographies pursue trades, leading to the negative association, not the value of trades themselves (much has been said about the increasing value of trades occupations.) Alternatively, students who receive a Pell Grant by definition have financial need. This socioeconomic background may depress students' long-term earnings potential, not the Pell Grant. If the policymaker made the intervention, the non-existence of Pell Grants would probably make college unachievable for many students – these students would be removed from the sample, increasing average incomes in the sample but (probably) decreasing these removed students' incomes. In other words, the model might technically be right, but in reality the policymaker is deceiving themselves.

Instead, the policymaker should be encouraged to use these model outputs only as a first step in a series of steps towards better decision-making. The next step should be to evaluate the uncertainty of the outputs, and decide whether and where to collect more data. Next, they should seek to explain and understand the phenomena – thinking of possible confounders to measure (e.g. socioeconomic status, geography) is easier when the above associations are presented and prioritised. Last, they should seek to form causal estimates of the effect of the desired action: making assumptions,

finding or creating past interventional data, or finding other variables that allow them to estimate the effect of an action.

3.6 Conclusion and future work

In this chapter we address the problem of principled gradient averaging. We show that Bayesian quadrature is a convenient all-in-one approach for obtaining full posteriors over average gradients in feature space with further decision-relevant information. Our method, PIGEBaQ, provides a fully probabilistic way to average function derivatives. Closed form expressions for the distribution of the average gradient and the volatility are provided.

Synthetic experiments demonstrate that PIGEBaQ recovers the gradient while providing advantages over other common gradient models – namely, a fully Bayesian estimate of the average gradient and an analytic measure of its volatility across the space. We argue that Bayesian quadrature is a natural tool for these tasks. We also believe it may serve as a useful basis for future interpretability tools, such as second derivatives and other informative metrics. While PIGEBaQ models the data rather the gradients of an underlying model, in application we propose that PIGEBaQ would be well-suited to provide probabilistic estimates of the gradients of models that are particularly expensive to run. PIGEBaQ could be naturally extended to help select points that are most informative for calculating the average gradient.

Using a case study of improving post-college incomes in the United States, we show how PIGEBaQ might be applied in practice. We show that its additional measures may influence a decisionamker beyond relying solely on the gradient. The framework we present provides a methodology for policy makers to understand the three questions of *direction*, *impact* and *confidence* highlighted in the introduction.

"When the centuries behind me like a fruitful land reposed;
When I clung to all the present for the promise that it closed:
When I dipt into the future far as human eye could see;
Saw the Vision of the world and all the wonder that would be.
Now my own Time approaches, and then the ending of human stuff:
Then, suddenly, as always, that which was must be and must end;
There is no, one term more to the weary World: go with him who says,
This is the last age of Man!"

— Four lines from *Locksley Hall*, Tennyson. Italicised generated by GPT-2.

4

What can be automated? Integrating expert knowledge to infer task automatability

This is joint work with Dr. Paul Duckworth and Dr. Michael Osborne. Originally published at the 2019 AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society as well as the NeurIPS 2018 AI for Social Good workshop. MO originated the survey idea; I developed the survey and collected data; PD and I developed the model and conducted analysis in equal fashion. We thank Carl Benedikt Frey for early scoping conversations; Jonathan Downing, Katja Grace, Allan Dafoe, John Salvatier, Matthew Willis for survey and analysis discussions; The Health Foundation (award #7559), the Oxford Martin Programme on Technology and Employment.

4.1 Introduction

Many phenomena are too complex or novel to have clear, intrinsically interpretable, modelling strategies. We often use machine learning to model such phenomena, such as medical diagnosis, dynamic robotics (like self-driving cars), and financial trading. We rely on flexible models to learn from data that contains hidden patterns. We then use the outputs of these models, or of a different model that interprets this model, to shape our reactions to these phenomena.

However, what happens if we have a deficit of real-world data, but a surplus of expert opinion? Here we take the case of estimating automatability: the extent to which functions of jobs can be automated. In this chapter, we consider the challenge of eliciting latent knowledge on automatability from human experts, modelling their expertise, and interpreting our model to extrapolate to insights not provided by experts. Our approach is interactive: human experts generate novel data, from which new insights are extracted. Naturally, a tool that does this, reliably combines expert knowledge, and extracts new insights, would be useful for decision-makers in areas that are intuition-rich and data-poor.

We focus on the association step of the Ladder. There are two implicit causal notions within our task. The first is the same as in the last chapter: associational relationships in our model are causal with respect to the model (but not necessarily the process). The second is that if the predictions of our model are true, they inform policy that implies interventions that may reshape the labour force. This latter is the most important. Useful, interpretable insights at the association step are mostly useful as inputs to form theories of what intervention should be made. It's only when these insights are combined with human causal intuition that they are given the causal

assumptions that afford interventional prediction.

This task is not simple. We have to decide what eliciting and combining expert knowledge looks like. (What data should we capture? How granular should it be? How should it be represented? How do we weight different expert opinions?) We have changing requirements in a model of human-data to human-decisions. (Do we need uncertainty measures? How smooth should the model be?) And we need to define what it means to interpret the model in such a way as to be useful for real-world decision-making.

In sum, we face a real-world policy question that lacks good data, and we need to interpret our model of latent expert knowledge of the policy question. In doing so, we need to:

1. **Interact:** human experts provide their subjective beliefs as data;
2. **Aggregate:** choose an appropriate way to combine expert belief into ground truth data;
3. **Model:** model this ground truth data;
4. **Interpret:** extract useful insights from the model.
5. **Act:** guide some action based on the insights.

Using the first survey of its kind collected to assess economic automatability, we develop an approach to these questions. We use this study both as an illustrative example, as well as one of the most comprehensive studies to date intended to illuminate the phenomenon for real-world decision-makers.

4.1.1 The state of automatability

Machine learning (ML), in combination with complementary technologies such as robotics and software-based standardisation, have rapidly become real substitutes and complements to human labor. This work aims to better understand that automation, and its effects on work. One example is Amazon Go, a recently opened grocery store that uses computer vision to replace cashiers, of which over 3.5 million are employed in the United States (Grewal, Roggeveen, and Nordfaelt, 2017; Bureau of Labor Statistics, 2017). Further, the 500 000 designers in the US are beginning to use constraint-based generative design to automate creative designs of buildings, industrial components, and more (Autodesk, 2017; Bureau of Labor Statistics, 2017). As a result, we as researchers in these fields often confront examples of media and public concern about technologies we develop.

What remains uncertain is the magnitude and direction of impact on employment of machine learning technologies. While recent advances in technology seem able to automate *intelligent* work, we lack good data on the scope of such automation. However, this field is notoriously opinion-rich, within which we believe is hidden, latent information about automatability. Knowing the answer to these questions allows a decision-maker to take phenomena-relevant actions, such adapting reskilling programs, industrial policy, and unemployment benefits in a highly targeted way.

We collected a detailed *task*-based survey of 150+ machine learning, robotics, and automation researchers. This is the first dataset of its kind with over 4,500 datapoints about what specific tasks are automatable according to *current* technology. In this "nowcasting" exercise, technologists provide knowledge of the extent to which a task can or cannot be automated with technology that exists *today*. We use a probabilistic model to infer the automatability of thousands of activities for which it would be

prohibitively difficult to collect reliable data. By collecting task-specific data to model automatability, we believe we can develop richer, more accurate frameworks about what can be automated by the current state-of-the-art in intelligent technology. We believe this allows society to better understand and prepare for automation.

The contributions of this chapter are a novel dataset of the automatability of workplace activities; a probabilistic method for inferring automatability of unmeasured activities by combining expert knowledge; activity-level analysis of automatability; and consequent patterns across worktypes, occupations, income, and education. We use more detailed numeric attributes and incorporate more expert knowledge than used in previous studies.

We demonstrate that we can accurately model expert opinions regarding the current state of automation, and introduce a methodology that goes beyond the limitations of the literature. We call for more and better measurement of automation at the activity level. We discuss how this is needed to better prepare governments, employees, and businesses for the effects of automation. Adjusting how we measure, integrate expert opinion, and interpret results can allow us to take micro-level policy decisions to prevent job loss, create an adaptable workforce, and increase long-term economic growth.

4.2 Related work

Traditionally, approaches to predicting what is automatable have developed frameworks based on the "types" of occupations and the skills they require (Autor, 2013; Acemoglu and Restrepo, 2018; Frey and Osborne, 2017). One popular framework places occupations on a manual-cognitive spectrum and a standardisable-dynamic spectrum. These frameworks are broad and do not best reflect that *tasks* (or groups of tasks), not *entire occupations*, are the unit of automation. As a result, public perception is

conflicted about the true effect of automation on work (Smith, 2016).

Recent work assumes that occupations are better analysed as evolving combinations of detailed tasks, skills, and/or environments (Arntz, Gregory, and Zierahn, 2016; Manyika et al., 2017a; Acemoglu and Restrepo, 2018). With increasingly granular job data available from sources such as from O*NET (National Center for O*NET Development, 2018), that break down occupations into hundreds of continuously updated numerical components, we believe there is an opportunity to evaluate occupations by focusing on their tasks, skills, and environments. Two recent reports use a task-first approach. A recent report by McKinsey (Manyika et al., 2017a) uses an unclear approach to model the opinions of automation potential of an unknown number of industry-based potentially non-technical experts. Another recent analysis by the OECD (Arntz, Gregory, and Zierahn, 2016) derives high-level task-level estimates from occupation-level estimates, and uses worker (not task) characteristics in their inference procedure.

We differ from previous approaches in three key aspects: first, we seek expert knowledge at the most granular task level, similar to (Manyika et al., 2017a) and (Grace et al., 2018). Second, we ask what is automatable *today* and do not make speculative assumptions about future developments or uptake of future technological advancements. Third, we present a robust and openly available probabilistic methodology and dataset.

4.3 Constructing ground truth from expert opinion

4.3.1 Making human knowledge elicitation easy

We conducted an online survey of 156 academic and industry experts in machine learning, robotics and intelligent systems about how automatable specific tasks are using technology available today. We sourced these experts via academic networks and mailing lists, intending to receive an international group of experts with expertise in different areas related to intelligent systems. The distributions of field-relevant academic experience, and the geographic location are shown in Figure 3 in Appendix B.2. 97% of participants had field-relevant academic experience: most participants coming from computer science, ML, robotic or AI backgrounds, and 52% of responses from the US, UK and Germany, with the rest from 30 other countries.

A key question is how to receive the largest quantity of high-value information from these experts. However, humans face a bandwidth constraint: absent other incentives, they have limited energy to answer many complex questions. This problem is highly sensitive. If the knowledge-elicitation exercise is too exacting, the quality of information is also likely to suffer.

A further constraint on this problem is that expert opinion is subjective and high-variance. Human subjective surveys result in quite different data than readings from a pressure gauge accurate to within $\pm 1\%$ of scale. Not only is the underlying process much more uncertain (a complex technoeconomic phenomenon vs. a stand-alone pressurised natural gas tank), but the medium of expression for the reading is more complex.

We opted to design the survey to address the above issues as follows:

1. **Low-effort questions:** maximise responses by making it easy for each respondent to understand the questions, and complete the full set of questions.

2. **Discretised ordinal scale:** discretise automatability into 4 easy-to-understand groups of increasing automatability

3. **Combine experts:** rely on combined expert judgements for final ground truth, rather than any one expert

4.3.2 Survey design

Each expert was presented with 5 randomly-selected occupations and those occupations' 5 "most important" tasks, taken from the Occupational Network (O*NET) 2016 database (National Center for O*NET Development, 2018). The complete list of 70 occupations whose tasks are annotated is shown in Appendix B.1, with occupations chosen to be representative of the feature space, with an emphasis on high-employment and hence familiar occupations.

Each expert answered the following question: *"Do you believe that technology exists today that could automate these tasks?"*, then labeled each task as either: *Not automatable today* (score of 1.0), *Mostly not automatable today (human does most of it)* (2.0), *Could be mostly automated today (human still needed)* (3.0), *Completely automatable today.* (4.0), or *Unsure*. Respondents also reported overall confidence in their answers (distribution shown in Appendix B.2).

Our dataset contains 4 599 task level responses from 156 academic and industrial experts from around the world, and across various scientific and industrial fields. Due to the mix of fields, we broadly label these as experts in artificial intelligence and recognise experts offer varying perspectives.

Table 4.1: Five randomly selected survey occupations and their surveyed tasks. *Importance* is a subjective rating supplied by O*NET.

Title	Task	Importance
Chief Executives	Direct or coordinate an organization's financial or budget activities to fund operations, maximize investments, or increase efficiency.	4.54
	Appoint department heads or managers and assign or delegate responsibilities to them.	4.48
	Analyze operations to evaluate performance of a company or its staff in meeting objectives or to determine areas of potential cost reduction, program improvement, or policy change.	4.40
	Direct, plan, or implement policies, objectives, or activities of organizations or businesses to ensure continuing operations, to maximize returns on investments, or to increase productivity.	4.39
	Prepare budgets for approval, including those for funding or implementation of programs.	4.17
Lawyers	Represent clients in court or before government agencies.	4.59
	Present evidence to defend clients or prosecute defendants in criminal or civil litigation.	4.50
	Select jurors, argue motions, meet with judges, and question witnesses during the course of a trial.	4.50
	Study Constitution, statutes, decisions, regulations, and ordinances of quasi-judicial bodies to determine ramifications for cases.	4.47
	Interpret laws, rulings and regulations for individuals and businesses.	4.47
Diagnostic Medical Sonographers	Observe screen during scan to ensure that image produced is satisfactory for diagnostic purposes, making adjustments to equipment as required.	4.87
	Observe and care for patients throughout examinations to ensure their safety and comfort.	4.85
	Provide sonogram and oral or written summary of technical findings to physician for use in medical diagnosis.	4.84
	Select appropriate equipment settings and adjust patient positions to obtain the best sites and angles.	4.83
	Operate ultrasound equipment to produce and record images of the motion, shape, and composition of blood, organs, tissues, or bodily masses, such as fluid accumulations.	4.83
Cooks, Institution And Cafeteria	Clean, cut, and cook meat, fish, or poultry.	4.64
	Cook foodstuffs according to menus, special dietary or nutritional restrictions, or numbers of portions to be served.	4.61
	Clean and inspect galley equipment, kitchen appliances, and work areas to ensure cleanliness and functional operation.	4.61
	Apportion and serve food to facility residents, employees, or patrons.	4.58

	Direct activities of one or more workers who assist in preparing and serving meals.	4.27
Brickmasons And Blockmasons	Remove excess mortar with trowels and hand tools, and finish mortar joints with jointing tools, for a sealed, uniform appearance.	4.63
	Construct corners by fastening in plumb position a corner pole or building a corner pyramid of bricks, and filling in between the corners using a line from corner to corner to guide each course, or layer, of brick.	4.60
	Measure distance from reference points and mark guidelines to lay out work, using plumb bobs and levels.	4.47
	Break or cut bricks, tiles, or blocks to size, using trowel edge, hammer, or power saw.	4.39
	Interpret blueprints and drawings to determine specifications and to calculate the materials required.	4.31

4.3.3 Combining expert knowledge

Each occupation's 5 separate tasks has, on average, 10 different expert subjective automatability ratings. The challenge now becomes one of combining 10 different subject ratings into a single ground truth consensus for us to model. The challenge has two core characteristics: we do not know how to trust each individual expert nor how to identify the true label; en masse, however, the experts exhibit patterns of agreement and disagreement, despite rating the task independently. To represent the first characteristic, we should express a belief over expert accuracies and consequently over the final label; to represent the second, we should use agreement and disagreement to combine expert ratings. As a result, we use Independent Bayesian Classifier Combination to combine multiple classifiers (Kim and Ghahramani, 2012; Simpson et al., 2013) because it is designed specifically with these characteristics in mind.

IBCC assumes a label c_i^k for a task i by expert ("classifier") k is generated by a multinomial distribution where $p(c_i^k \mid t_i = j) = \pi^k_{j,c_i^k}$, where t_i is the true label for task i. A core part of this model is the assumption that $\boldsymbol{\pi}_j^k$ is a *vector* $[\pi^k_{j,1}, \ldots, \pi^k_{j,1}]$ representing a row in a confusion matrix. Each row is Dirichlet distributed and implicitly

represents the expert's propensity to exchange labels. An additional assumption is that all expert $k = 1, 2, \ldots, K$ are independent. These two assumptions upweight experts that are more likely to discern between label classes, and more likely to label similarly to how other experts classify.

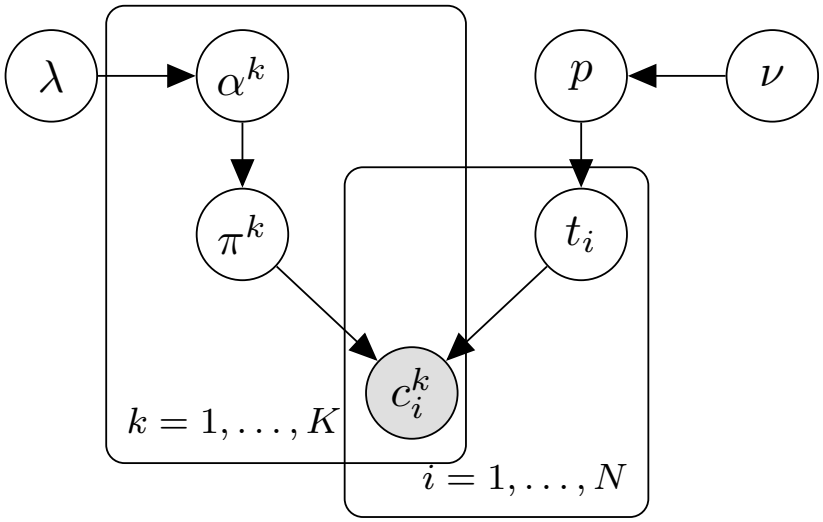


Fig. 4.1: The graphical model for Independent Bayesian Classifier Combination. c_i^k is the ith label given by expert k. True label t_i is what we aim to infer. π^k is expert k's confusion matrix for their given labels $\boldsymbol{c}^k$ and the final labels $\boldsymbol{t}$. α^k is a prior over expert k's confusion matrix, λ a global confusion hyperprior; p is a prior over the true label probabilities, and ν a hyperprior over p.

We follow Kim and Ghahramani (2012) to perform inference via variational Bayes to obtain the posterior $p(\boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{t}, \boldsymbol{\alpha} \mid c)$, where $\boldsymbol{p}$ is the multinomial probability parameter matrix for $\boldsymbol{t}$, $\boldsymbol{\pi}$ represents cross-expert cross-label confusion, $\boldsymbol{t}$ is the vector of true labels, $\boldsymbol{\alpha}$ represents priors for expert confusion matrices, and $\boldsymbol{c}$ is the matrix of expert labels. Following this, we are interested in $\gamma_{i,j} = P(t_i = j \mid \boldsymbol{p}, \boldsymbol{\pi}, \boldsymbol{\alpha}, \boldsymbol{c}) \propto p_j \Pi_{k=1}^{K} \pi_{j,c_i^k}$. Our final ground truth label is taken as $\sum_j^J \gamma_{i,j} \cdot j$, or the probability-weighted label.

We then face the challenge of reflecting our ground truth in feature space. We do not have feature-space representations at the task level, but we do at the work-activity level (described below). Task ground truths are then collected into tasks

that share work activities, and combined using a custom weighted average (described below). We found that labels concentrate around whole and half values; we round the final values to the closest 0.5 (a half-class).[1]

The advantage of this combination-of-experts approach is that we can use expert opinions across tasks to influence how we weight each expert's opinion. This approach is a principled way of doing this while making minimal assumptions. Especially as it comes to complex, uncertain problems with little ground truth, good expert aggregation methods are desirable. Ideally, they require essentially no contextual information. IBCC is one method that does so. We imagine it would be useful beyond this automation question and extend into social scientific policy questions, as it has in applications like crowdsourced science (Simpson et al., 2013).

As it comes to the subject at hand, it also reduces the complexity of classification needed. It requires no forecasting or prediction by the participants. We are able to exploit consensus and variability as reliable signals. The distribution of task-level expert responses, and the IBCC combined task-label distribution are shown in Figure 1 in Appendix B.2.

4.3.4 Occupations, activities and tasks

An occupation o is represented as a set of discrete *tasks* an employee may be required to perform, $o = \{t_1, t_2, \ldots, t_n\}$. Each occupation may have a different number of tasks, and all tasks are detailed enough to be unique to one occupation.

Similar tasks are grouped into a *work activity* w such that $w = \{t_i, \ldots, t_j\}$.

[1] A side-effect of this approach is that the ground truth labels are more polarised at the extreme values (one and four), when compared with simply (mean) aggregating the task-level responses together. A complication of this is that when comparing two models based upon their Root Mean Squared Error (RMSE), lower absolute error is achieved by not using the (more polarised) IBCC combined labels.

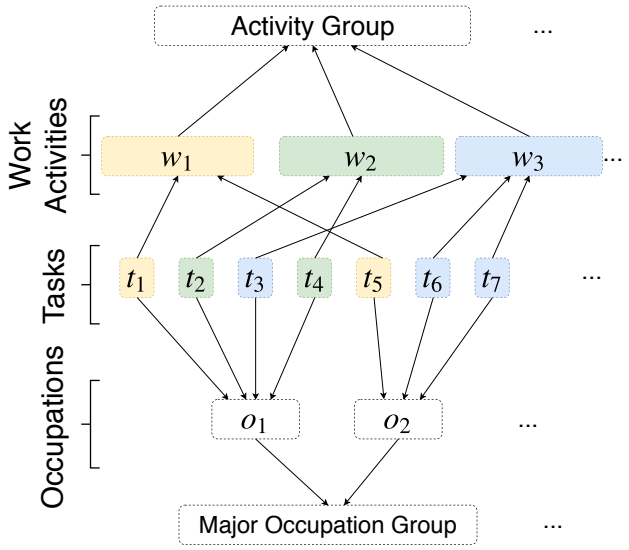


Fig. 4.2: The O*NET Database Taxonomy including occupations o, work activities w, tasks t, and high-level groupings. Tasks belong to both work activities and occupations; activities and occupations each have multiple higher-level groupings.

While a task is occupation specific, work activities are generic activities performed across multiple occupations. We further aggregate results to major occupational groups and high-level activity groups. A diagram demonstrating this hierarchy is shown in Figure 4.2.

4.3.5 Automation by work activity

To create activity-specific feature vectors, we aggregate the roughly $20,000$ tasks into their $2,067$ work activities (which are called Detailed Work Activities in O*NET) to create a feature vector $\boldsymbol{x}_w$ for specific work activity w. Each occupation is represented as a vector $\boldsymbol{x}_o$, comprised of the numerical ratings of its skills, ($\boldsymbol{x}_o^s$), knowledge ($\boldsymbol{x}_o^k$) and abilities ($\boldsymbol{x}_o^a$), i.e. $\boldsymbol{x}_o = \left[\boldsymbol{x}_o^s, \boldsymbol{x}_o^k, \boldsymbol{x}_o^a\right]$. We represent each task t of occupation o with the feature vector of occupation o, because we assume that an occupation's

skills, knowledge, and abilities informs those needed to perform its constituent tasks. These features are measured quantitatively on a 1 to 5 scale by dozens of employees and experts in the O*NET database.

The activity feature vector is a weighted average of its constituent task vectors: $\boldsymbol{x}_w = \sum_{t \in w} w_{(t,w)} \boldsymbol{x}_o$. $w_{(t,w)}$ is a normalised weight of the task's relative importance to its occupation and work activity, i.e. $w_{(t,w)} = I_{(t,o)} I_{(t,w)} / \sum_{t \in w} I_{(t,o)} I_{(t,w)}$. The relative importance of the task to its occupation, $I_{(t,o)}$, and the relative importance of a task to its work activity, $I_{(t,w)}$, are calculated as

$$I_{(t,o)} = {}^{I_t}/_{\sum_{t \in o} I_t} \tag{4.1}$$

$$I_{(t,w)} = {}^{I_t}/_{\sum_{t \in w} I_t}\,, \tag{4.2}$$

where task importance I_t is a numeric measure also supplied by O*NET.

4.3.6 Automation by occupation

We also explore what the automatability of activities implies about the automatability of the occupations that perform them. First, we infer automation scores $\hat{\boldsymbol{y}}_w$ for all work activities (including unlabeled ones), as will be described in the next section. We construct an occupation automation score $\hat{y}_o$ for occupation o using the importance-weighted average of its constituent work activities: $\hat{y}_o = \sum_{w \in \Omega_o} I_{(w,o)} \hat{y}_w$, where Ω_o, is the set of all work activities performed by occupation o, and $I_{(w,o)}$ is the importance score of work activity w normalised over this set.

We present automatability over *major occupation groups* which are the highest level occupation categorisation provided by the Bureau of Labor Statistics' SOC system. We represent a major occupation group G as a set of occupations $\{o_1, o_2, \dots\}$ and

construct the automation score $\hat{y}_G$ by taking the employment-weighted average of the automatability scores of its constituent occupations, $\hat{\boldsymbol{y}}_o$ for all occupations in the group.

4.4 Model comparison and validation

We seek a flexible function estimation capable of modelling complex, non-linear relationships between the features (skills, abilities, knowledge) and (perceived) automatability in high-dimensional space. Given the social scientific nature of the study, we also desire a measure of model uncertainty. We compare models based on their "tolerance accuracy" score – the percent of posterior prediction means, $\hat{\boldsymbol{y}}_w$, that are within 0.5 of the ground truth post-IBCC survey value $\boldsymbol{y}_w$. This is a sensible score for our task, and allows more flexibility in our multiclass ordinal setting than strict accuracy or average error. This score also takes into account that we compare models with heterogenous output types – some output discrete labels, and some output continuous values. We optimised the hyperparameters of all models using 10-fold cross-validation.

Our first candidate model class is the Gaussian process Rasmussen and Williams (2006), which abbreviate as GP, which have previously been applied to *occupation*-based data in (Frey and Osborne, 2017) and (Bakhshi et al., 2017). We specifically use the ordinal likelihood function introduced in Chu and Ghahramani (2005) to reflect the nature of having discrete labels but with an ordinal interpretation (*not at all* to *completely* automatable):

$$P\left(y_i \mid f\left(x_i\right)\right) = \int P_{\text{ideal}}\ \left(y_i \mid f\left(x_i\right) + \delta_i\right) \mathcal{N}\left(\delta_i; 0, \sigma^2\right) d\delta_i = \Phi\left(z_1^i\right) - \Phi\left(z_2^i\right) \quad (4.3)$$

where $f(x_i)$ is the GP latent function, δ_i is a Gaussian random variable that defines

a soft boundary between ordinal classes, with

$$P_{\text{ideal}}(y_i \mid f(x_i)) = \begin{cases} 1, & b_{y_{i-1}} < f(x_i) \leq b_{y_i} \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

for ordinal class "bins" defined by lower and upper limits $(b_{y_{i-1}}, b_{y_i})$. We denote $\Phi\left(z_1^i\right)$ as the cumulative density function of the z-score of the upper class boundary in a Gaussian distribution with mean $f(x_i)$ and a learned variance σ^2. Intuitively, this likelihood function divides the real-line into partitions where the likelihood is 1 or 0. Then, it assumes Gaussian noise that assigns non-zero likelihood when the GP prediction is outside of the true class bin, decreasing the further the prediction is from the true class bin.

We use the squared exponential, or RBF, kernel, as it consistently performed well compared to other kernels and was less likely to overfit. We optimise the kernel hyperparameters by minimising the negative log marginal likelihood $\log p(\boldsymbol{y}_w \mid \boldsymbol{x}_w)$ as described in Rasmussen and Williams (2006), using the open source software GPFlow (Matthews et al., 2017).

Table 4.2: Model tolerance accuracy & negative log likelihood.

Model	Accuracy (std)	−Log-likelihood
Ordinal RBF GP	**0.645** (0.028)	385.6
Gaussian RBF GP	0.643 (0.102)	382.3
Ordinal DNN	0.604 (0.071)	–
Random Forest	0.517 (0.081)	–
Ordinal Regression	0.451 (0.036)	–
$\hat{\boldsymbol{y}}_w = \text{avg}(\boldsymbol{y}_w)$	0.575 (0.065)	–
Proportional Random	0.374 (0.049)	–

For other candidate models, we consider ordinal logistic regression (Pedregosa-Izquierdo, 2015), a random forest, and a neural network with an ordinal loss function (Hart, 2017), with a 4-layer (120-60-120-7) fully connected layer architecture

and 10% layer-wise dropout. A proportional random assignment and constant mean predictor are compared as a naive baseline on predictive performance. Notably, just predicting with the value of the mean achieves fourth-best performance. This is due to the concentration of the values, as the mean-predictor offers no useful information. While we believe our metric is the best for assessing models, this highlights the challenges of modelling subjective data with the ordinal classification interpretation we've taken. Results for each model are displayed in Table 4.2 (standard deviation in brackets). The ordinal GP model consistently outperforms comparative methods at prediction of posterior mean values of automatability over the space of work activities. While the non-ordinal GP model and the optimised deep neural network perform on average similarly to the ordinal GP model, they do so much less reliably. We note that the variance of model performance cause models to overlap in performance. We choose the GP ordinal model because it performs well with minimal performance variance.

4.5 Experiments and results

4.5.1 Question 1: What is automatable?

Using the best performing GP model to infer the automatability score for all 2,067 work activities (including the training set), we examine examples of automatable and not-automatable activities. Appendix B.3 presents a sample of work activities with the highest and lowest automatability from the unlabeled data (with uncertainties).

We observe that activities such as "route mail to correct destinations" (3.82) or "cut fabrics" (3.93) have a high automation potential (and indeed, are widely automated). However, we also notice that white collar activities are also highly automatable with current technology: "Operate digital imaging equipment" (3.63),"Send information,

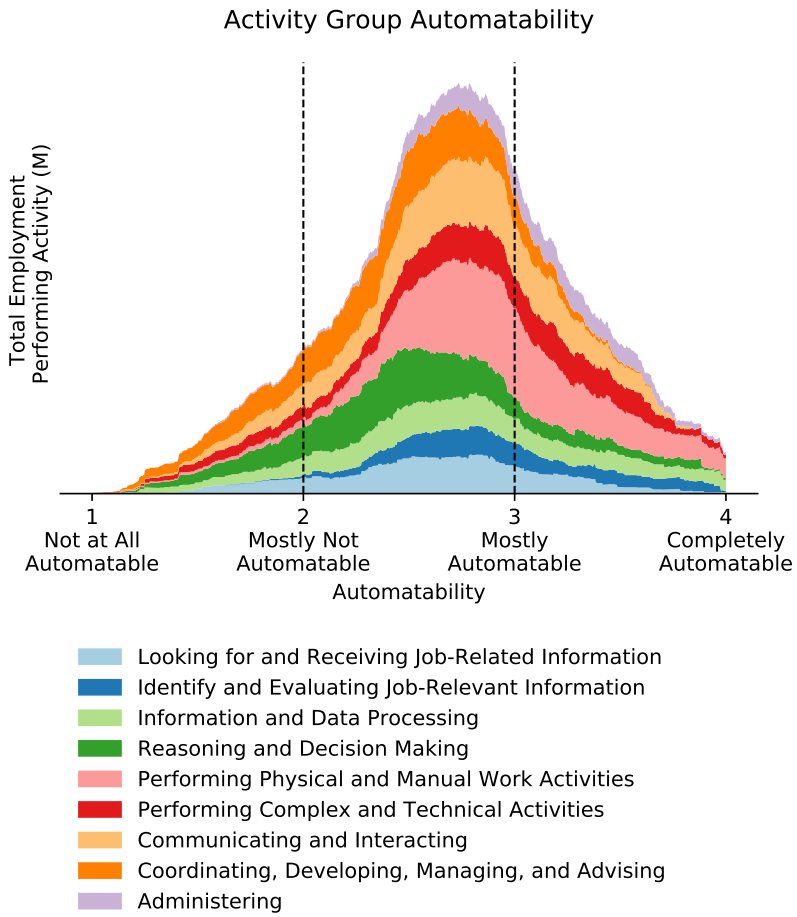


Fig. 4.3: Amount of activity, in terms of the number of employed people who perform them, and their automatability scores, divided into 9 high level activity groups. The more mass to the right hand side of the graph, the more activities are automatable. Note that most of the mass sits above the agnostic median of 2.5. Some groups are more automatable than others, but on the whole, there is somewhat even distribution across activity groups.

materials or documentation" (3.39), and "Advise others on ways to improve processes or products" (3.38). Insights such as these propose likely future automatable areas, where automation could be achieved in the real world with relatively little further attention to the underlying technology. The mean of automatability scores is 2.65, indicating that that the model, learned from expert estimates, believes that tasks are on average more likely to be more automatable than not.

In lieu of profiling the long list of activities by their automatability here, we consider

instead what *groups* of activities are automatable, and the implications for occupations when using an activity-first approach. In Figure 4.3 we plot the automatability of activities by the number of currently-employed individuals who perform them, and classify into nine high-level activity groupings. It becomes evident that while most activities are between mostly and mostly not automatable, work tends to lie closer to "mostly automatable". Eight times as much work lies between "mostly" and "completely" automatable than between "mostly not" and "not at all" automatable, when weighted by employment. Activities classified as "reasoning and decision making" and "coordinating, developing, managing, and advising" are less likely than others to be automatable. However, "administering", "information and data processing" and (perhaps surprisingly) "performing complex and technical activities" are more likely to be automatable.

Additionally, we average the automatability scores of an occupation's activity automatabilities to create an occupation-level automatability score. (How to properly aggregate activities for an occupation-level score is a subject of further research, but we use this as preliminary exploration.) We classify occupations into 12 high-level "Major Occupation Groups", as in Figure 4.4. We see that the model predicts very high automation potential in office, administrative support (orange), and sales occupations (red), which together employ about 38 million people in the United States. This stands in contrast to the popular emphasis on the automation of physical processes such as production (yellow), farming, fishing and forestry (dark orange), and transportation and material moving (teal), which employ about 20 million people in total. In contrast, two Major Occupation Groups appear robust to automation: education, legal, community service, arts, and media occupations (light green), as well as management, business, and financial occupations (light blue).

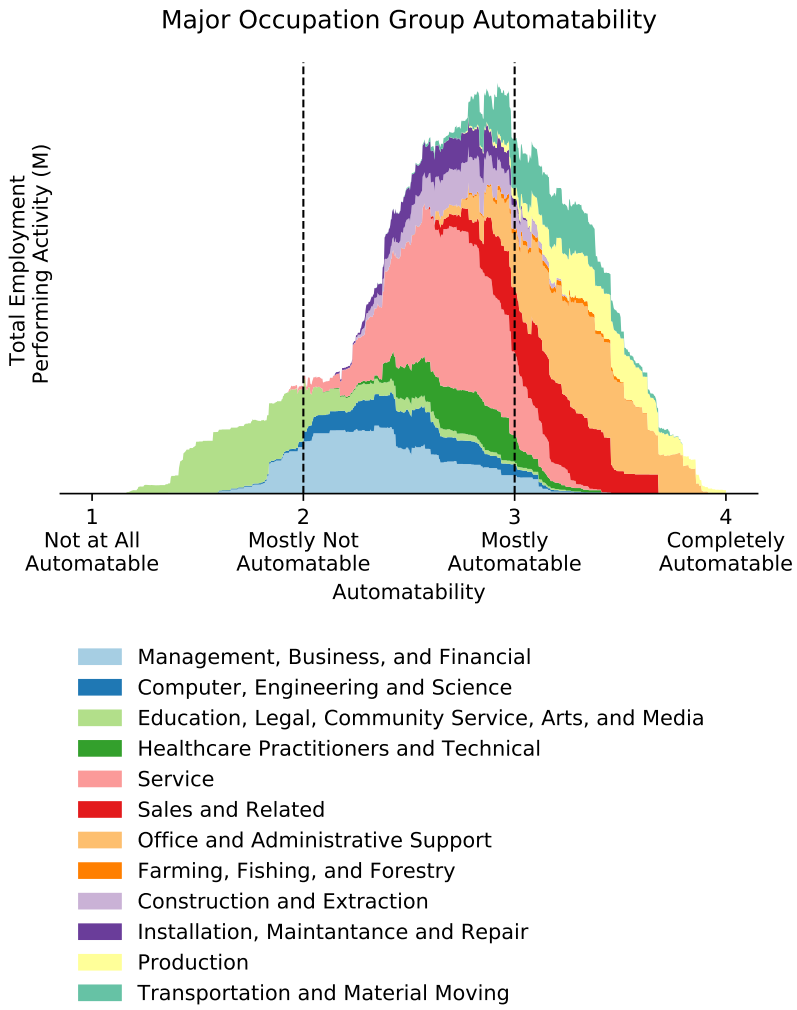


Fig. 4.4: Amount of activity, in terms of the number of employed people who perform them, and their automatability scores, divided into 9 high level activity groups. The more mass to the right hand side of the graph, the more activities are automatable. In contrast to the activity group analysis, there are more disparities in the amount of automatable work by occupation group. This suggests that occupation groups tend to cluster more by automatability than activity groups do.

Trends across income and education

Using our occupation-level mappings, we present a preliminary analysis of how automatable activities cluster across income and education, to understand the likely impact on employees. We use median annual income data from the Occupational

Employment Statistics from the Bureau of Labor Statistics (Bureau of Labor Statistics, 2017), and the average expert estimate that one requires at least a bachelor's degree to perform an occupation from O*NET (National Center for O*NET Development, 2018).

The impact is perhaps expected. The highest paid, most educated occupations tend to be the least automatable. They tend to be much smaller occupations. However, it is worth noting that even being paid well and having a bachelor's degree does guarantee an occupation's activities are not automatable. "Air Traffic Controllers" make about $125,000 a year, yet are deemed mostly automatable (2.93). "Cytogenetic Technologists", for example, require a Bachelor's degree (with a 100% likelihood from O*NET), with an estimated occupation automatability of 3.03.

Nor is the opposite always true. "Preschool Teachers" and "Teacher Assistants" make just under $30,000 a year, yet are mostly non-automatable (1.71 and 1.87, respectively). O*NET experts estimate a 5% chance of needing at least a bachelor's degree to perform successfully as a "Heating and Air Conditioning Mechanics and Installer", an occupation which is also predicted mostly not automatable (2.38).

Model disagreements with ground truth

It is useful to consider where our model disagrees most with our ground truth labels. While, as discussed, it is difficult to confirm which estimate is true, when the model disagrees with the ground truth, it is using all information learned from all other ratings. As a result, a disagreement represents where a local expert synthesis (point) disagrees with a general synthesis (model). For the sake of building explanatory theories, contradictory points might reveal shortcomings in the model, biases in the experts, or exemptions that require separate explanations than other points. Appendix B.3 lists the 50 tasks with most disagreement between ground truth and model.

In the cases where the model *overpredicts* relative to the ground truth, we see

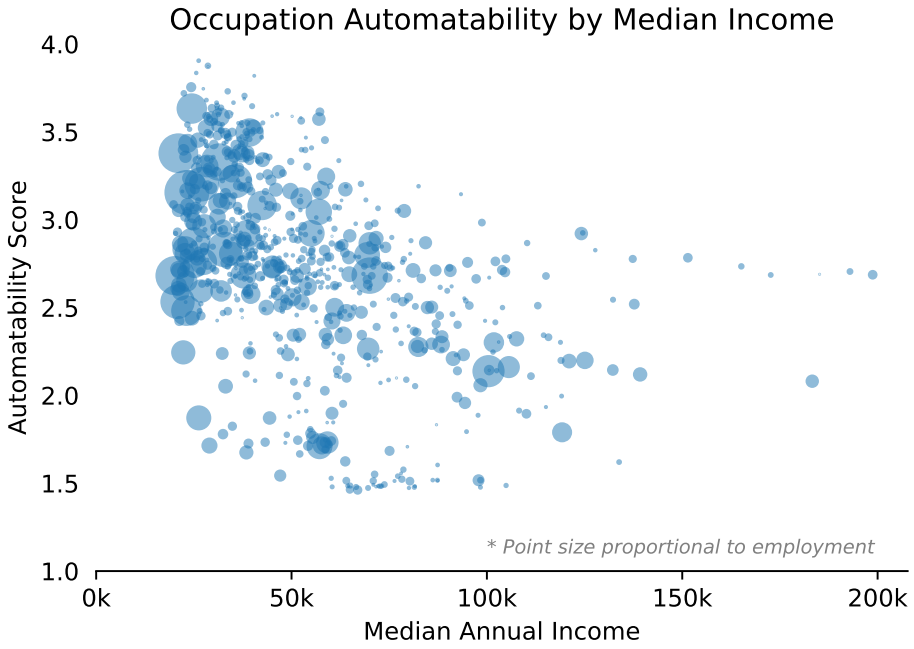


Fig. 4.5: Occupation-level automatability scores by annual median income. Point size is proportional to total employment in that occupation. Note that a large number of occupations, and large employing occupations, that pay less are highly automatable; in contrast, a small number of high-paying jobs with fewer employees are much less automatable.

common themes. Some activities, such as "connect electrical components or equipment" ($\boldsymbol{y}_w$: 1.0; $\hat{\boldsymbol{y}}_w$: 2.69) are characterised by dextrous physical action (the difficulty of automating of which has been famously proposed as "Moravec's paradox" in Moravec (1990)). Others involve information exchange in a situation with a clear goal, such as "communicate with customers to resolve complaints or ensure satisfaction" ($\boldsymbol{y}_w$: 1.0; $\hat{\boldsymbol{y}}_w$: 2.31). There are also cases involving complex process monitoring. Across these activities, it seems conceivable that while intelligent technology may not automate the *entire* activity, some combination of activity simplification, custom data gathering, and intelligent technology could automate a non-trivial amount of the activity.

The cases where the model *underpredicts* are more varied in interpretation. Some model underpredictions could be a case of downweighting unreasonable expert beliefs.

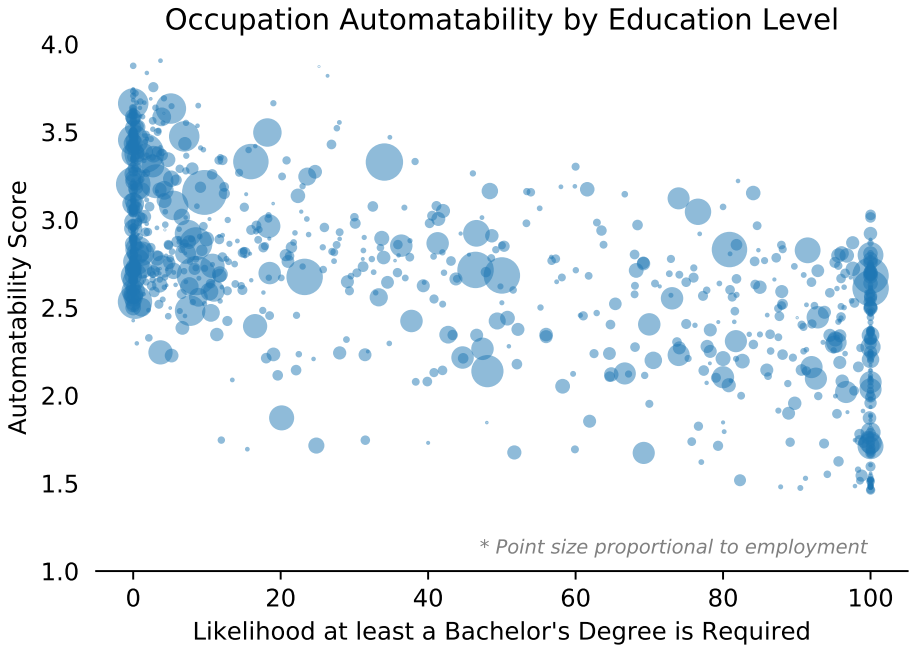


Fig. 4.6: Occupation-level automatability scores by education attainment. Note a relatively linear negative relationship between the likelihood a bachelor's degree is required and an occupation's automatability.

For example, "position construction forms or molds" ($\boldsymbol{y}_w$: 4.0; $\hat{\boldsymbol{y}}_w$: 2.80) – simple in theory, but influenced by many dynamic environmental factors to render it complex in practice. Other cases of underprediction might also be a reflection of a lack of data around the activity in feature space, leading to limited model learning. For example, there are some activities which are clearly automatable (and already automated) for which the model predicts low automatability (e.g. "Process customer bills or payments" and "Create electronic data backup to prevent loss of information"). Alternatively, the more questionable predictions may be artifacts of the O*NET taxonomy, in which the work activity title does not accurately reflect a critical nuance of its constituent tasks.

4.5.2 Question 2: What makes work automatable?

We now consider what increases or decreases the automatability of some activities. We compute the average derivative of automatability with respect to each feature as described in (Baehrens et al., 2010), over the space of work activities. For the nth feature, this is

$$AD(n) := \mathbb{E}(\frac{\partial m(\boldsymbol{x})}{\partial \boldsymbol{x}_n}), \tag{4.5}$$

where $m(x)$ is the posterior mean distribution. This measures the expected increase in automatability for a unit increase in the feature. Table 4.3 shows the highest and lowest average derivatives of the posterior mean function per feature.

These gradients seem to reflect what intelligent technology increasingly offers: work that is clerical, repetitive, precise, and perceptual can increasingly be automated. Increases in the features *Clerical, Number Facility, Depth Perception, Control Precision* and *Production and Processing* tend to increase an activity's automatability. Perhaps surprisingly, increases in *Economics and Accounting* and *Sales and Marketing* knowledge and ability also increase automatability, which reflects that we saw business-oriented sales & administrative work types having higher than average automatability.

On the other hand, work that is more creative, dynamic, and human oriented tends to be less automatable. While variable, the three strongest features driving decreased activity automatability are *Installation, Programming* and *Technology Design*. That is to say, the experts who answered our survey are relatively safe, or misperceive themselves to be.

The gradients might, for example, be used by employers/employees and policy makers to skill themselves differently, proactively change the characteristics of work

activities, or set policy to incentivise the development of particular skills, knowledge, abilities, or occupations.

Table 4.3: The 5 most positive and 5 most negative "local" feature gradients by activity group.

Activity Group	**Feature**	**Average Gradient**
Administering	Clerical	0.14
	Depth Perception	0.11
	Number Facility	0.10
	Control Precision	0.10
	Mathematical Reasoning	0.09
	English Language	−0.09
	Instructing	−0.09
	Management of Financial Resources	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Communicating and Interacting	Clerical	0.14
	Depth Perception	0.11
	Number Facility	0.11
	Control Precision	0.10
	Mathematical Reasoning	0.09
	Management of Financial Resources	−0.09
	Instructing	−0.09
	English Language	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Coordinating, Developing, Managing and Advising	Clerical	0.14
	Number Facility	0.10
	Control Precision	0.10
	Depth Perception	0.10
	Mathematical Reasoning	0.09
	English Language	−0.09
	Management of Financial Resources	−0.09
	Instructing	−0.09
	Management of Personnel Resources	−0.11
	Learning Strategies	−0.12
Identify and Evaluating Job-Relevant Information	Clerical	0.14
	Number Facility	0.10
	Depth Perception	0.09
	Mathematical Reasoning	0.09
	Mathematics (skill)	0.08
	Installation	−0.09
	Customer and Personal Service	−0.09
	Mechanical	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.11

Information and Data Processing	Clerical	0.14
	Depth Perception	0.10
	Number Facility	0.10
	Control Precision	0.10
	Mathematical Reasoning	0.09
	English Language	−0.09
	Instructing	−0.09
	Management of Financial Resources	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Looking for and Receiving Job-Related Information	Clerical	0.14
	Number Facility	0.10
	Depth Perception	0.10
	Control Precision	0.10
	Mathematical Reasoning	0.09
	English Language	−0.08
	Management of Financial Resources	−0.08
	Instructing	−0.00
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Performing Complex and Technical Activities	Clerical	0.14
	Depth Perception	0.11
	Number Facility	0.10
	Control Precision	0.10
	Mathematical Reasoning	0.09
	Management of Financial Resources	−0.09
	Instructing	−0.09
	English Language	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Performing Physical and Manual Work Activities	Clerical	0.14
	Number Facility	0.10
	Mathematical Reasoning	0.09
	Depth Perception	0.09
	Mathematics (skill)	0.09
	Mechanical	−0.09
	Installation	−0.09
	Customer and Personal Service	−0.09
	Management of Personnel Resources	−0.09
	Learning Strategies	−0.11
Reasoning and Decision Making	Clerical	0.14
	Number Facility	0.10
	Depth Perception	0.10
	Control Precision	0.10
	Mathematical Reasoning	0.02
	Coordination	−0.02
	Instructing	−0.02
	Management of Financial Resources	−0.09
	Management of Personnel Resources	−0.11

Learning Strategies −0.12

Further tables and figures are given in the Appendix. Appendix B.4 shows global average gradients for features. It reveals that it does not take a minor change in a small number of features for an activity's automatability to increase by a class. Appendix B.5 gives an example of local gradients by activity group. Appendix B.6 visualises examples of the effect of normalised feature values on automatability.

Uncovering the drivers of automation

In our own analysis, we found that using activity-level ratings allows us to hypothesise richer frameworks about the drivers of automation than the previous occupation-level frameworks. For example, one particularly useful analysis is to examine clusters of similar activities, which score significantly high in a subset of influential features but vary substantially in inferred automatability. Some of these activity clusters across features such as *Dynamic Strength, Persuasion, Critical Thinking,* and *Production and Processing* seemed to imply the predictive importance of (a) task-standardisation versus dynamism; (b) a well-specified performance metric versus open-ended thinking; (c) single-party versus multi-party goal satisfaction; and (d) active thinking versus active physical interaction. We reserve final conclusions from this approach for further expanded and validated research. In summary, we believe this suggests that more activity-level data likely allows us to find richer frameworks for automatability prediction, and that contradictions and variable contrasts serve as useful tools for explanation-making.

4.6 Implications & limitations

4.6.1 Societal impact

Automation is a notably data-sparse yet opinion-heavy area of study (National Academies of Sciences and Medicine, 2017; Mitchell and Brynjolfsson, 2017). We need more clarity about what can be automated, at the *actual* level of automation (tasks), and more granular frameworks based on more granular data. Policymakers would be able to design better, more targeted policy responses, such as incentives to preemptively modify occupations or programs to reskill workers. Workers would be able to upskill or retrain in a more targeted way (towards certain tasks or away from certain tasks) to be robust to automation, instead of abandoning entire occupations. Further, we note the large *psychological* burden of fear, uncertainty, and doubt that comes with uncertain predictions; we hope to replace that with the optimism, clarity, and confidence that comes from every worker having better predictions to enable more effective responses. Last, we would be able to better spot ethically-challenging cases of activity automation, especially in healthcare and social services, *before* they happen, so that we can hold preparatory, informed ethical discussion.

We believe this necessitates collecting more data on automation. Governments, researchers, businesses, and employees would *all* benefit from uniting to do so. Despite automation being one of the dominant themes of work in the coming decades – a perhaps irreversible shift in how work is done in the future – we are only just taking our first steps towards granular, activity-level measurement. In this work we offer our dataset, consider the value of activity-level data, and present preliminary results that reinforce previous ones and provide more fidelity and opportunities for deeper research.

4.6.2 Limitations

While this research is a first attempt at deriving task-based automatability scores for every task, it faces structural limitations that we hope will be resolved over time. We offer these limitations as a potential guide for an organisation with the time and capital to take seriously the question of collecting the right data on automatability.

First, activities should have their own measured feature representations, instead of having representations constructed via aggregating the occupation feature vectors of the tasks in an activity. However, this is not a small exercise. To do so for all activities would require developing an activity-level and consistent schema of feature representation, as ONET provides for occupations. Then it would need to be measured for all activities – of which there are about twice as many as occupations.

Second, our approach uses subjective expert nowcasts, rather than validated nowcasts. The alternative is to form a measurable ground truth of the degree of automation of an activity, and model this ground truth data. Unfortunately, what automation is and how to measure it is still an open question.

Last, we are conscious that spurious inferences about the consequences of automatability on jobs and work are attractive to make. Among many measures worthy of further study, we note that our work does not consider the *net* effect of automation. For example, increased automatability may push costs down of a particular activity, and the elasticity of activity demand relative to cost may be such that this increases the amount the activity is performed. If there is still human involvement in the activity, the net effect on human involvement is unclear.

4.7 Conclusion & future work

In this work, we studied an opinion-rich, data-poor economic phenomenon with very real policy implications. We introduced an approach that extracts latent expert knowledge from opinions, probabilistically combines expert knowledge, models this knowledge with limited assumptions, and provides insights that can inform policy. We particularly focus on the average gradient as an expected measure of feature influence for an arbitrary model. We show the type of insights that can be derived from such a model. As a result, we recover and verify many expected results, as well as counterintuitive results. We also use model disagreement as a way to expose potential areas of expert blindness, or sensitive regions in feature space worth exploring further. By using a more granular approach to "now-casting" task-level automation, we can unlock more nuanced frameworks about what *actually* can be automated. Using task and activity-level data, we can likely go further to better understand the drivers of automatability.

However, our approach is a first step. We propose to the community six applied research gaps that our dataset and approach should be useful for answering:

1. **Real-world validation**: Does our model accurately identify activities that are already automated?

2. **Surprising automation**: Which activities are more, or less, automatable than previous models would have predicted?

3. **Automatable vs. automated**: *Why* are some automatable activities not automated while others are?

4. **Multivariate interaction patterns**: How does one feature modify a different feature's effect on a task's automatability?

5. **Economic value**: What is the monetary value of automation potential for highly automatable activities? (See Manyika et al., 2017b.)

6. **Employee characteristics**: What are the patterns of demographics, industry, technology use, and other characteristics of employees performing activities with low- and high-automatability activities?

In order to solve the question of how to "nowcast" automatability, we need to solve several problems at the heart of modelling, expert-model interaction, and insight interpretability. In that sense, it is a good model question for many problems where decision-making, interpretability, and policy meet. In particular, in this work, several valuable questions have arisen that may inform future research in this meta-approach:

1. **Integrating subjective & objective data**: if a phenomenon has both types of data as measurable quantities, how can they be both be used?

2. **Interacting with models**: how can experts interact with models, providing multiple rounds of feedback to combine model insights and expert theory?

3. **A framework for choosing models & insights**: can we automate how to decide which model(s) we use and what insights we want from it, given a question?

4. **Granularity of data**: what level of data granularity do we need to answer the relevant question?

5. **Trust**: how much can we trust model insights in complex, highly subjective phenomena, especially ones with policy implications? (Especially when extrapolating to unobserved regions in featurespace, as in this example.)

6. **Causal calibration for decision-makers**: given decision-makers have to use

causal and non-causal models, how can we ensure they are adequately calibrated to understand and weight the use of non-causal models in their decisions?

The question of what is automatable is at a very early stage. To answer it, we likely require many rounds of generating insights, defining, theory-forming, conducting natural and interventional experiments, and collecting data with large geographic coverage. It has very real policy consequences, is a significant source of concern for citizens, and is highly complex. Increasingly interactive, integrative, and interpretable models may help in this early stage of exploration.

"All buddhas know all phenomena come from interdependent origination. They know all world systems exhaustively. They know all the different phenomena in all worlds, interrelated..."

— the Avatamsaka Sutra

5

Multisynth: multitask Gaussian processes for Temporal Causal Inference

This work was produced entirely by myself.

5.1 Multisynth: multitask Gaussian processes for synthetic controls

Estimating treatment effects – the effect of an intervention on some outcome, such as a drug treatment on a patient's blood pressure or a financial stimulus package on an economy's unemployment rate – is the unique task of causal inference. Typically, this is estimated as the difference in real outcome under the intervention, and the counterfactual outcome – what *would have* happened had the treatment not been applied. One of the ways in which this task can be viewed is as a missing data problem: one can only observe one or the other, but not both at the same time. This is termed the *Fundamental Problem of Causal Inference,* referring to the need to estimate the unobserved scenario to calculate the treatment effect. Subject to the specifics of the estimation task and the assumptions required therein, machine learning tools are being increasingly used to develop these estimators or solve persistent shortcomings, especially functional form limitations and the curse of dimensionality.

Here we consider evaluating treatment effects as they evolve over time. Given some intervention (e.g. a drug), to a unit δ (e.g. a patient) at time τ, we are interested in the difference between the observed outcome $Y_{\delta,t}$ and the *counterfactual* outcome $y^*_{\delta,t}$ at every time $t > \tau$, measured as $y_{i,t} - y^*_{\delta,t}$. For example, this situation could arise naturally when evaluating the effect of randomly assigning one drug over another to a patient, or changing the trade laws between two countries. We might also be interested in the cumulative individual treatment effect over time, $\sum_{t=\tau}^{T} \left(y_{\delta,t} - y^*_{\delta,t}\right)$. In reality, $y^*_{\delta,t}$ is unknown and we would like to express our uncertainty over its values. To do so we represent our belief over $y^*_{\delta,t}$ as $P(y^*_{\delta,t} \mid D)$, where D is any data we use to develop an estimator of $y^*_{\delta,t}$.

How to develop an estimator for this distribution depends on the context and data available. Here I focus on the Synthetic Control method (which I will refer to as *Synth* to avoid confusion with Structural Causal Models), an extension of the *Difference-in-Differences* (DiD) methodology.

However, this approach is limited by its simplicity. Synth is required to make simplifying assumptions, and to throw out useful information. In this chapter we propose how to use multitask Gaussian processes (MTGPs) to reconcile a number of the shortcomings of current approaches to time-varying treatment effect estimation.

We begin by reviewing the synthetic control method and recent attempts to reconcile its shortcomings. We also review attempts to apply MTGPs to imputation in real-world settings, including in treatment effect estimation. The contribution of this chapter is threefold:

1. We motivate the synthetic control context as a prediction and imputation problem and highlight its structural causal characteristics;
2. We introduce a novel modification of the multitask Gaussian process framework that uniquely addresses a large number of the shortcomings of the Synthetic Control approach;
3. We empirically validate the MTGP approach and show it is robust to extrapolation, temporal relationships, and the validity of an identifying assumption, while retaining interpretability.

5.2 The synthetic control method

5.2.1 Motivation and approaches

The Synthetic Control Method is motivated by the need to estimate treatment effects over time. Given a treatment to unit i at time τ, a typical approach is to evaluate the individual treatment effect of the outcome variable Y at time $t > \tau$ evaluated as $y_{\delta,t} - y^*_{\delta,t}$, where $y_{\delta,t}$ is the observed value of the outcome of the treated unit δ at time t, and $y^*_{\delta,t}$ is the counterfactual value of the outcome of a treated unit δ at time t.

A popular estimation method for this problem is the *Difference-in-Differences* approach, which estimates $y_{\delta,t} - y^*_{\delta,t}$ as the difference between the change in outcome from τ to t for the treated unit minus the same change in the control group. Suppose there are m units in total, with one treated unit, and thus $m - 1$ untreated units. For notational simplicity, we let $D = m - 1$. Then:

$$y_{\delta,t} - y^*_{\delta,t} = (y_{\delta,t} - y_{\delta,\tau}) - \frac{1}{D}\sum_{d}^{D}(y_{d,t} - y_{d,\tau})\,, \tag{5.1}$$

where $d \in [1, \ldots, D]$ denotes the index of one of D untreated units. The rightmost term denotes the average change in outcome value for control units between treatment time τ and evaluation time t. If there are multiple treated units, one can equally perform the same averaging for treated units.

The Difference-in-Differences (DiD) approach requires a critical assumption, called the Parallel Paths assumption: that the baseline change in the control units would be experienced by the treated unit had the treated unit not been treated. Further, DiD limits evaluation to a single time point t. Last, it should be evident that the

counterfactual estimate is based on an equal-weighted average of control units.

The Synthetic Control Method (Abadie and Gardeazabal, 2003), which we term *Synth*, is an expansion on this intuition that relaxes these conditions. It allows for treatment effect inference at any time $t > \tau$. It also allows for the control units to have different weights, recognising that different units may have different relevance to the counterfactual estimation.

Synth adjusts for this representation problem by learning a pre-treatment estimate of the treated unit that is differentially informed by control units. It estimates the counterfactual as a linear weighted combination of control units, termed "donor" units, with simplex constrained weights:

$$\min_{\beta} \|\boldsymbol{y}_{\delta,1:\tau} - \boldsymbol{\beta} Y_{D,1:\tau}\|_2 \,, \tag{5.2}$$

having simplex constraints:

$$\beta_d \in [0,1] \tag{5.3}$$

$$\sum_{d}^{D} \beta_d = 1 \tag{5.4}$$

where $\boldsymbol{y}_{\delta,1:\tau}$ is the vector of pre-treatment outcomes for the treated unit and $Y_{D,1:\tau}$ is a $D \times \tau$ matrix of outcome values for the D donors. Thus forming the counterfactual estimate at time t as:

$$y^*_{\delta t} = \sum_{d}^{D} \beta_d y_{d,t} \,. \tag{5.5}$$

this formulation of Synth is known as the *constrained regression* formulation of the synthetic control method. The extended version of Synth incorporates not just outcomes data, but also data on other covariates, and finds a qualifying estimator:

$$\min_{\beta} \sqrt{(X_{\delta,1:\tau} - \boldsymbol{\beta} X_{D,1:\tau})^T V (X_{\delta,1:\tau} - \boldsymbol{\beta} X_{D,1:\tau})}\,, \tag{5.6}$$

where X is now a covariate matrix which may or may not include the outcomes data, and V is a diagonal matrix containing learned weight parameters representing covariate importance. Because both V and $\boldsymbol{\beta}$ are learned, and $\boldsymbol{\beta}$ is depends on V, there is a double optimisation problem. This is usually treated by jointly learning V, $\boldsymbol{\beta}$ that minimises mean squared error of the outcome variable and its prediction.

Intuitively, Synth finds a linear estimator of the pre-treatment treated unit that is within the interpolation region defined by the the donor units. One of the implications of this model is that it is presumed the pre-treatment outcome values and post-treatment counterfactual outcome values of the treated unit lie within the range defined by the convex combination. This is due to the constraints on the donor weights. If, for example, the treated unit had outcomes $y_{\delta,t} < 0 \,\forall\, t \geq 0$ but $y_{d,t} > 0 \,\forall\, t \geq 0$, then there does not exist $\boldsymbol{\beta}$ *s.t.* $\boldsymbol{\beta} Y_{D,t} < 0$. This constraint is both practical (otherwise extrapolation would be required) and for interpretability reasons (one can say a counterfactual is "composed of" $\beta_d \times 100$ percent donor unit d). However, one can relax this constraint to allow extrapolation (Abadie and Gardeazabal, 2003; Doudchenko and Imbens, 2016), removing interpretability. The estimation task essentially reduces to one of finding a good linear predictor of the pre-treatment treated unit. It is up to the modeller to decide how to conclude that a predictor is "good". Questions like covariate balance and temporal weighting are resolved by the modeller, and each

resolutions implicitly encodes different assumptions.

For the purpose of this chapter, we consider only the constrained regression context. That is, "goodness" is measured as pre-treatment outcome prediction. More specifically, we frame the problem probabilistically, where "goodness" is defined as finding a model that minimises the negative log likelihood. However, the model we use can easily be extended to generate synthetic covariates as well, thus generalising to the extended case.

For the synthetic control method's relative simplicity and principled improvement on standard imputation approaches, Athey and Imbens (2017) praised it as "arguably the most important innovation in the evaluation literature in the last fifteen years."

5.2.2 Identifying assumptions

Synth requires an identification strategy – the set of assumptions and estimator(s) that, taken together, compose an argument that an estimand is identified and an estimate meets desirable criteria (e.g. unbiasedness). There are two explicit *identifying assumptions* in the Synth method and one implicit one.

First, the relationship between the donor units and the treated ones – estimated as described above – is assumed to stay stable over time. That is, a donor weight β_d does not change.

Secondly, no donor unit ought to have experienced a similar treatment shock as the one the treated unit did. Further, a modeller should (but in practice rarely does) argue that any treatment or regime change experienced by a donor unit, after the treatment of the treated unit, could have counterfactually occurred in the treated unit – at least approximately.

Last, the implicit assumption is the Stable Unit Treatment Value Assumption (SUTVA), a core component of the potential outcomes model, that states that the

covariates and outcomes of the donor units should be unaffected by the treatment on the treated unit. Violation of SUTVA would violate the requirement that a donor unit not have experienced a similar treatment.

In practice, these last two requirements can be difficult to satisfy. By the nature of donor units being related to the treated unit, many real-world scenarios arise where units interact. The fragility of these assumptions is a shortcoming we address in our model.

To cast Synth as a Structural Causal Model, we need an SCM that explains the core assumption that donor units are *related* to the treated unit. Assuming that a donor unit is not a causal ancestor of the treated unit (and vice versa) Reichenbach's Common Cause principle states that donor and treated units share a latent parent process. It is these latent parents that give the units similar dynamics that are exploited to estimate the synthetic control.

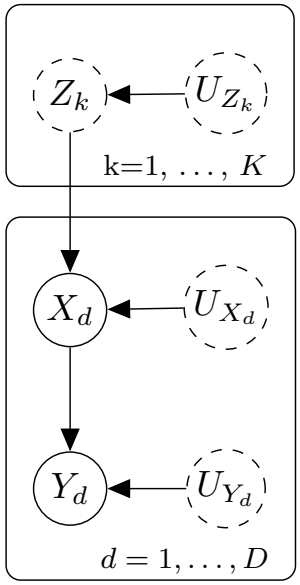


Fig. 5.1: An SCM model of the implied graphical model of Synth's assumed relationship between units. Z represents latent processes that cause units to share information. This affects the input processes X_d of a unit that then affects the outcome Y_d. In practice, Z is likely highly difficult to measure or model, requiring a simplifying assumption (like Synth's linearity assumption) or a model of the latent processes (as in this work).

5.2.3 Shortcomings of the synthetic control method

It should be clear, now, that Synth reduces to a prediction task with added assumptions and criteria for "goodness". In this framework, immediately several limitations emerge when compared to more flexible predictors than a single, non-probabilistic linear estimator with simplex-constrained donor weights.

First, the functional form assumption of the linear estimator is highly restrictive. Only the donor units as they are observed can serve as inputs for estimation. This is restrictive in that a donor unit may not represent underlying dynamic relationships, and which would likely lead to poor estimation when donors are noisy.

Further, this model uses *no information post-intervention* to inform the estimator, even though we have post-intervention information that is, theoretically, interesting; captures no temporal dynamics, either within units or across them (addressed by Brodersen et al., 2015; Poulos, 2017; Modi and Seljak, 2019; Ding and Toulis, 2020); the relationship stays stable over time (addressed by Brodersen et al., 2015); it offers no probabilistic information even if we have it (addressed by Brodersen et al., 2015; Kinn, 2018; Viviano and Bradic, 2019); it cannot handle missing data (addressed by Amjad, Shah, and Shen, 2018); and the extended regression context can make it hard to achieve balance in high dimensions (addressed by Ben-Michael, Feller, and Rothstein, 2018).

Each of these cited proposals attempt to reconcile one of those shortcomings, but no proposal reconciles all of the above. The motivation for this chapter is twofold:

1. the synthetic control task can be cast as a prediction and imputation problem (which are familiar tasks in the machine learning literature);
2. the aforementioned shortcomings can be elegantly reconciled in a single modelling approach using multitask Gaussian processes.

5.2.4 A note on terminology

We have used the term "counterfactual" here liberally. We use it in the same way the Rubin causal model uses it: the unobserved outcome of a specific unit under an intervention (or non-intervention). However, we should be clear to distinguish this from the Pearlian definition, wherein a Structural Causal Model, an intervention, and a distribution over latent variables define a counterfactual. Without this full treatment, the term "counterfactual" used in this chapter is actually closer to "interventional" in the Pearl framework. This distinction is important, as in the following chapter we give a full Pearl treatment of counterfactuals.

5.3 Multitask Gaussian processes

The multitask Gaussian process (MTGP) framework (Bonilla, Chai, and Williams, 2009) is an extension of the Gaussian process framework to model multiple tasks simultaneously. The advantage of the MTGP framework is that it is able to exploit correlations between task, if they exist, in order to better model each individual task. If tasks are uncorrelated, the MTGP reduces to multiple single task GPs. In this way, the motivation behind the MTGP for the synthetic control use case becomes clear: the MTGP can flexibly use cross-task correlations (the core insight of the synthetic control case) for better prediction, with minimal additional assumptions.

5.3.1 Formulation

The essential motivation of multitask models is that they allows for tasks to share mutual information. Multitask models exploit cross-task correlations in order to better inform models for at least one task – usually all tasks. There are many

ways to write multitask models – a Bayesian hierarchical model, with a task-shared hyperprior, is one example. In our application, we focus on the multitask Gaussian process framework, which is actually composed of a family of multitask models that construct the formulation of the latent function and the covariance of a Gaussian process differently depending on different assumptions.

Standard multitask Gaussian processes

We consider a setting where $\boldsymbol{X} = \{x_{d,t} \mid d = 1, \ldots, m, t = 1, \ldots, T_d\}$ represents the set of input observations for m tasks at time indices t, and $\boldsymbol{Y} = \{y_{d,t} \mid d = 1, \ldots, m, t = 1, \ldots, T_d\}$ for the output observations. Note that T_d may be different for each task d, that is each task may have a different number of observations. Also note that as opposed to the indexing notation in the *Synth* framework, d is now used to refer to all tasks (thus $d \in \{1, \ldots, m\}$), including the treated task. This is because the treated task is a donor unit in the multitask setting as it is modelled jointly.

A multitask Gaussian process models outputs for task d as normally distributed given a latent function f_d:

$$y_{d,t} \sim \mathcal{N}(f_d(\boldsymbol{x}_t), \sigma_d^2)\,, \tag{5.7}$$

where σ_d^2 is the noise variance for task d, and where f_d is drawn from a Gaussian process:

$$f_d(\boldsymbol{x}) \sim GP(\mu_d(\boldsymbol{x}), k_d(\boldsymbol{x}, \boldsymbol{x}')). \tag{5.8}$$

For two tasks d and d', k_d is decomposed into two covariance functions:

$$k_{d,d'}(\boldsymbol{x}, \boldsymbol{x}') = k_T(\boldsymbol{x}, \boldsymbol{x}') \times k_c(d, d')\,, \tag{5.9}$$

with k_c representing a cross-task similarity covariance function and k_T representing a covariance function modelling similarity in feature space. This implies a general kernel:

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{K}_T \otimes \boldsymbol{K}_c \,, \tag{5.10}$$

where $\otimes$ is the Kronecker product, and $\boldsymbol{K}_c$ is an $m \times m$ cross-task correlation matrix. While $k_T(\boldsymbol{x}, \boldsymbol{x}')$ can be easily constructed according to how kernels are usually constructed for a Gaussian process, it is less clear how to construct the cross-task covariance kernel $\boldsymbol{K}_c$. Bonilla (Bonilla, Chai, and Williams, 2009) constructs $\boldsymbol{K_c}$ as the Cholesky decomposition approximation of the covariance matrix:

$$\boldsymbol{K}_c = \boldsymbol{L}\boldsymbol{L}^\top \,, \tag{5.11}$$

where $\boldsymbol{L}$ is the lower triangular matrix formed in the Cholesky decompostion of $\boldsymbol{K_c}$, parameterised by $\boldsymbol{\theta}_c = \{\theta_{d,d'}\}$ for all combinations of d and d'. That is, $\boldsymbol{\theta}_c$ is a vector of "free" parameters representing covariances for all 2-combinations of tasks. These parameters are learned through optimisation. In sum, the multitask Gaussian process approach constructs a single Gaussian process that uses a unique covariance structure to model multiple tasks.

The linear coregionalisation model

In our application, we use a modification of the Multitask Gaussian Process that originates in the geostatistics literature called the *Linear Coregionalisation Model* (LCM). The LCM approach expands on each function f_d associated with task d and models it as a sum of Q latent processes with their own kernel:

$$f_d(\boldsymbol{x}) = \sum_q^Q w_d^q f_d^q(\boldsymbol{x}) \,, \tag{5.12}$$

where w_d^q is a scalar learned through optimisation. Secondly, we let:

$$f_d^q(\boldsymbol{x}) \sim GP_d(\mu_d^q(\boldsymbol{x}), k_d^q(\boldsymbol{x}, \boldsymbol{x'})) \,, \tag{5.13}$$

where we set $\mu_d^q(\boldsymbol{x}) = \mathbf{0}$. By virtue of Gaussian processes being closed under addition, we can then show that $f_d(\boldsymbol{x})$ is also drawn from a Gaussian process with some function-specific *basis* kernel $k_d^q(\boldsymbol{x}, \boldsymbol{x'})$. The LCM assumes a linear combination of random functions, leading to the covariance:

$$cov[f_d(\boldsymbol{x}), f_{d'}(\boldsymbol{x'})] = \sum_q^Q b_{d,d'}^q k^q(\boldsymbol{x}, \boldsymbol{x'}) \,, \tag{5.14}$$

where it assumes $k^q(\boldsymbol{x}, \boldsymbol{x'}) = k_d^q(\boldsymbol{x}, \boldsymbol{x'}) \, \forall \, d = 1, \ldots, m$. In this covariance structure, $b_{d,d'}^q = w_d^q w_{d'}^q$ and scales the q-th covariance kernel. We can then form the whole data kernel $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X'})$ for all functions:

$$\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X'}) = \sum_q^Q \boldsymbol{B}^q \otimes k^q(\boldsymbol{X}, \boldsymbol{X'}) \,, \tag{5.15}$$

where $\boldsymbol{B}^q$ collapses all $b_{d,d'}^q$ into a symmetric positive definite matrix such that $\boldsymbol{B}^q \in \mathbb{R}^{D \times D}$ where entry d, d' is $b_{d,d'}^q$. This weight matrix represents our task covariance matrix, and for inference we parameterise it using the Cholesky decomposition as above, with $\boldsymbol{B}^q$ adopting the role of the cross-task covariance kernel $\boldsymbol{K}_c^q$. However, we make one additional assumption following Álvarez, Rosasco, and Lawrence (2012), where $\boldsymbol{B}^q = \boldsymbol{w}^q \boldsymbol{w}^{q\intercal} + \kappa \boldsymbol{I}$, where $\boldsymbol{w}^q = \{w_1^q, \ldots, w_m^q\}$ and κ is a scaling parameter ≥ 0 for stability.

The coregionalisation model is a natural fit for our inference task. By nature, it makes explicit the fact that donor units are related to treated units by virtue

of latent processes; this is not made explicit in other models. Additionally, the LCM model allows for model building when not all outputs are observed at all data points. This is by definition the case in the counterfactual scenario: we have no access to the true counterfactual, and we use no post-treatment data for the treated unit. This is additionally useful with missing or misaligned data, which is frequently observed in standard causal inference tasks. This also means that multiple treated units can be inferred at the same time.

One challenge is the assumption that $k^q(\boldsymbol{x}, \boldsymbol{x}') = k_d^q(\boldsymbol{x}, \boldsymbol{x}') \; \forall \; d \in [1, \ldots, m]$. An obvious implication of this assumption is that patterns that are evident in the task are not separately modelled. The Linear Coregionalisation Model models each task as a linear combination of latent functions that are shared across tasks. Consequently, tasks with unique covariance structures that a modeller might want to account for – periodic or linear structure, for example – need to be handled separately. Cheng et al. (2020) imperfectly addresses this by using the spectral mixture kernel (Wilson and Adams, 2013; Parra and Tobar, 2017). Alternatively, Nguyen and Bonilla (2014) flexibly handle task-specific structure by modelling a task as the linear combination of coregionalised components plus an additional task-specific component.

To get around this and account for the unique structure of each task, we slightly modify the Nguyen and Bonilla (2014) Collaborative Multioutput Gaussian process approach to account for task-specific structure. We model each task such that:

$$f_d(\boldsymbol{x}) = \sum_q^Q w_d^q f^q(\boldsymbol{x}) + v_d(\boldsymbol{x}) \,, \tag{5.16}$$

where

$$v_d(\boldsymbol{x}) \sim \mathcal{GP}(\mu_d^v(\boldsymbol{x}), k_d^v(\boldsymbol{x}, \boldsymbol{x'})) \,. \tag{5.17}$$

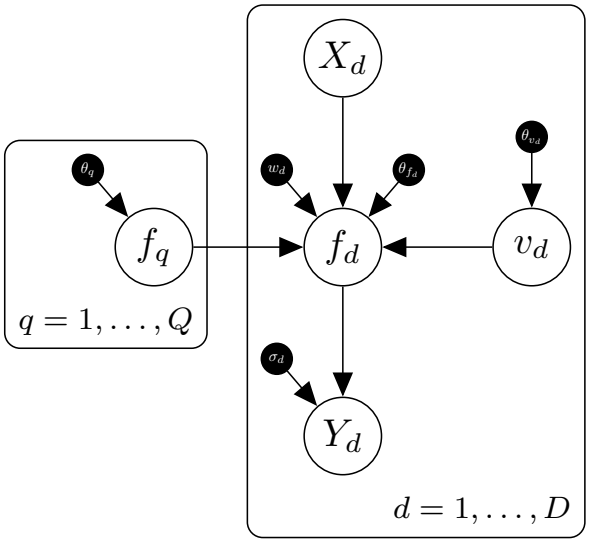


Fig. 5.2: A graphical model of our modified LCM. Each task-wise function f_d is a linear combination of latent functions f^q modified by weights w_d and an optional task-specific function v_d. Each task has its own likelihood variance parameter σ_d.

This formulation is similar to the cokriging estimator (Goovaerts, 1997) but where weights are placed on latent processes, rather than one weight on each of the i inputs. This can be implemented quite elegantly, where the m tasks each receive up to m additional latent functions f^q — one for each task — where we fix $w_{d^q} = 0$ if $d \neq j$ for some task j. That is, the contributing weight of an additive latent function is zero if the latent function is not the one additional latent function added for that task. The resulting $\boldsymbol{B}^{Q+m}$ is positive semi-definite. In this framework, where we had Q latent functions in the linear coregionalisation model, we now have up to $Q + m$ latent functions. The benefit of this framework is that we do not need to separately estimate task-specific and intertask models as in Forrester, Sóbester, and Keane (2007); Yang et al. (2019) and can use the same covariance formulation as in the LCM.

Interpreting cross-task correlations

If each basis kernel represents a distinct coherent covariance pattern, each function can be interpreted as a linear contribution of that kernel's covariance pattern to the task. The advantage in this case is that we can go further than task correlations and represent task-pattern correlations; two tasks may have similar linear patterns but different periodic ones, for example. We can, in this sense, break out the representational similarities in each task, providing a more detailed understanding of how and why some tasks are more or less related to other tasks.

We can then use $\boldsymbol{B}^q$ for $q < Q$ as a pattern-specific similarity matrix. However, while $\boldsymbol{B}^q$ is a valid symmetric positive definite matrix, it is unclear how to interpret its components. Here, we follow the transformation proposed in Durichen et al. (2014) which normalises the lower triangular from the Cholesky decomposition of $\boldsymbol{B}^q$ to constrain each $\theta^q_{d,d'} \in [-1, 1]$. Following Durichen et al. (2014) we can then interpret the resulting cross-task correlation matrix.

Inference

Posterior inference in multitask Gaussian processes can be performed identically as with single task ones, as derived in Rasmussen and Williams (2006). As a result, for posterior predictive inference of the counterfactual outcome y_δ we only need to first optimise the hyperparameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}^q \mid q = 1, \ldots, Q\}$, that is, every vector of hyperparameters parameterising each basis kernel, in addition to $\boldsymbol{\theta}_v$, the hyperparameters of the single-task Gaussian process. We optimise these hyperparameters by minimising the negative marginal log likelihood $-\log p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta})$ via gradient descent as implemented in GPFlow (de G. Matthews et al., 2017). We exploit the separable Kronecker product structure to allow for inference speed ups, by decomposing the inverse of the total

kernel as the Kronecker product of the inverses of each separate kernel.

As with all Gaussian process models, multitask Gaussian process is limited by the complexity of posterior predictive inference. MTGPs have significantly higher cost. Whereas m single GPs take $m \times \mathcal{O}(n^3)$ time, MTGPs take $\mathcal{O}(m^3 n^3)$ time (as the kernel matrix $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ is $mn \times mn$-dimensional.) In the synthetic control setting, usually m is chosen to be relatively small and $m \ll n$, so the effect of the size of the data n dominates the cost. Frequently, in counterfactual inference where *Synth* is applied, n is manageable under this constraint. Indeed, a frequent challenge of real world scenarios of causal inference – with political, economic, medical, sociological data, for example – is small data. This is one motivation to use rich nonparametric methods like Gaussian processes, over parametric methods that make limiting assumptions that are especially influential when modelling small amounts of data. Cheng et al. (2020) have developed a recent approach that modifies the LCM using a sparsity-inducing prior over each $\boldsymbol{\theta}^q$ to allow better scaling to thousands of patients in the medical context. We later discuss alternative approaches to scaling or approximating multitask Gaussian processes.

Unifying with Synth

It should also be noted that the canonical *Synth* approach can be approximated in a multi-task Gaussian process framework. If $Q = 0$ but we introduce $m-1$ individual task functions, one for each untreated task, which are then independently optimised. Then we set $w_d^q = 0$ if unit d is untreated, leaving only positive weights for the treated unit. A final round of optimisation is performed only on pre-treatment data, constraining w_d^q to the simplex if desired. This amounts to *Synth*, but where donor data is instead a linear combination of Gaussian processes that model each untreated task.

5.3.2 Use in causal inference

There are several components of the MTGP framework that makes it naturally useful for counterfactual inference, as in our use-case. Indeed, the motivations below are the key insight in this chapter: many limitations of the synthetic control framework are naturally handled by the MTGP approach.

Fit for the multitask problem: the LCM model makes explicit the assumed latent shared parent as in the *Synth* model. Further, we can interpret the structure of this covariance that is learned by the MTGP model through each $\boldsymbol{B}^q$.

Probabilistic: Second, this model is fully probabilistic. Where a treatement effect is informing a policy decision, this addtional probabilistic information is useful for a decision-maker to estimate the range of interventional outcomes.

Nonparametric: Third, the MTGP's nonparametric nature enables three special characteristics: accounting for flexible functional relationships, using post-interventional data, and handling nonstationarity. Abandoning the parametric paradigm eliminates the need to constrain the donor-treated relationship to a functional form. Further, as the functional form may change over time (or at least its parameter values, which motivates Brodersen et al., 2015) the MTGP form can account for changes in the relationship by incorporating information on how non-treated tasks change in their evolution.

The MTGP's nonparametric formulation also uses post-intervention data that would otherwise be ignored. The post-intervention data – in this case, the observed values of donor units post-intervention – are still used in optimisation to infer the hyperparameters of the kernels. However, this extra information is thrown away in the parametric approach as the donor units are not modelled individually and only the pre-treatment relationship between the donor and treated units matter for optimisation. Using post-intervention data makes inference more expensive, of course, but this is

the cost of learning better latent functions. This is potentially complicated if there is a non-stationarity in a donor unit, which could be addressed by having more latent functions or accounting for non-stationarity using a non-stationary kernel.[1]

One can also account for changes in relationships by using non-stationary kernels (Genton, 2002; Remes, Heinonen, and Kaski, 2017; Gelfand et al., 2004). This allows one to relax the restrictive identifying assumption that the donor-treated relationship changes after the point of intervention. Practically, a modeller would remove unsuitable donors, but this becomes potentially impossible if nonstationarities are complex.

Temporal: Fourth, the Gaussian process approach easily handles temporal relationships without arduous parametric specification. This occurs in the task-specific kernel $k_T(\boldsymbol{x}, \boldsymbol{x}')$ in the MTGP formulation and in each basis kernel $k_T^q(\boldsymbol{x}, \boldsymbol{x}')$ in the LCM formulation.

Missing data: the model can easily handle missing data. Further, by virtue of expressing uncertainty while retaining basic structural assumptions, the model can operate well with small data. Small and missing data is frequently observed in real-world social scientific settings and this flexibility becomes useful.

In aggregate, we argue that the benefits of having probabilistic estimates, flexible functional form, nonstationarity, post-interventional data, task-specific patterns, small and missing data be managed by a single model — all the while retaining donor-treated similarity as in *Synth* — makes the MTGP framework a very natural fit for the temporal causal inference context.

[1]Note that this is a consequence of using our specific modified LCM framework. In the canonical multitask Gaussian process, only data points observed simultaneously are used for optimisation. Given the counterfactual is entirely missing after the intervention timepoint, this makes using post-intervention data infeasible without a variant of our LCM model.

Use of covariate information

We can incorporate covariate information in two ways in the MTGP framework: either as inputs in $\boldsymbol{X}$, or as outcomes themselves in $\boldsymbol{Y}$. Incorporating it as inputs allows for deviations in the input trend of donor units to result in increased counterfactual uncertainty – a naturally desirable quality from an identification point of view. However, if incorporated as inputs, then if covariates do not resolve on the same scale they should either be normalised or modelled with dimension-wise Automatic Relevance Determination kernels (Neal, 1996; Duvenaud, 2014). The latter can make the loss surface highly multimodal during hyperparameter optimisation. The alternative is to model covariates as outputs. In this case the loss surface can also be multimodal, and significantly increase inference time. In our applications we do not consider covariate information, but as discussed it can easily be extended to do so.

5.3.3 Limitations

There are several limitations with this approach that, depending on context, would need to be addressed in an appropriate way.

The first and crucial limitation is the computational expense of this approach. In practical terms, this can make scaling to many data points difficult, requiring special forethought; this is discussed later. More importantly, Multisynth expends computation on all tasks, rather than just the task that has undergone a treatment. This stands in contrast to approaches that only expend computation to accurately model the task of interest, such as *Synth* and other pre-treatment prediction methods. This is due to the multitask nature of Multisynth; in particular, it is important to model post-intervention data for non-intervented tasks well if we are to take advantage of post-intervention patterns. If the modeller has reason to believe post-intervention patterns are not

relevant, in addition to the irrelevance of the coregionalisation kernels and no self-trend, then the modeller may reduce the computational burden by modelling the treated unit as a single output Gaussian process with the other units as inputs.

Secondly, it is likely that, as with Gaussian processes, Multisynth is sensitive to values of hyperparameters. For this reason, a judicious approach can prevent sensitivity: using multiple random restarts of optimisation in order to judge convergence to similar hyperparameters; using sampling-based approaches like Bayesian optimisation to choose hyperparameters; performing cross-validation-based hyperparameter selection and judging convergence on multiple hold-out sets.

5.4 Related work

5.4.1 Advances in flexible synthetic control methods

Since introducing the synthetic control method, the field of econometrics has devised new approaches to address some of its shortcomings (or, more accurately, to enhance the model). Most popularly, Brodersen et al. (2015) rethinks the synthetic control method as a Bayesian structural time series that uses a linear state-space model to capture time-varying dynamics in the relationship between the donor and treated units. This model, applied in the context of online advertising, is the most widely used and is compared in the empirical analyses below. However, the model is restricted to a linear functional form and using exclusively pre-intervention data. Maintaining the linearity of the donor-treated unit relationship, Amjad, Shah, and Shen (2018) devise an approach based on Singular Value Decomposition to easily manage missingness, automatically select informative donors, and be flexible with respect to the amount of covariate information provided. Poulos (2017) appears to be the first instance

of breaking the linearity assumption by modelling the counterfactual time series as the output of a recurrent neural network. However, using this nonlinear model, one loses easy interpretation of task similarity, and one is again restricted to training only on pre-intervention information. Ding and Toulis (2020) then develops an ensemble model that enables any nonlinear model to contribute to counterfactual prediction while also developing a nonparametric test for treatment effect estimation. Modi and Seljak (2019) further relaxes the nonlinearity via a Gaussian Process approach based on standardising the data in Fourier space and relying on the Gaussian process kernel to capture what would be task-specific temporal dynamics in this space; similarly, Cheng et al. (2020) use multitask Gaussian processes with a spectral mixture kernel to capture different unique temporal dynamics. Both Hazlett and Xu (2018) and Ben-Michael, Feller, and Rothstein (2018) address the issue of covariate imbalance due to dimensionality, the former introducing a constructed kernel approach and the latter using an outcome model to debias the estimate *ex post.* Arkhangelsky et al. (2019) develop a full unification of the synthetic control method and difference-in-differences, and proposes an augment synthetic control with time and unit weighting while retaining enough flexibility to incorporate a latent factor model of the relationship between the treated and donor units. Last, Kinn (2018) reviews these developments and finds that across tasks, machine learning methods tend to outperform traditional synthetic control approaches, motivating further use of machine learning in this case.

5.4.2 MTGPs in causal inference

Multitask Gaussian processes have increasingly been used in causal inference-like tasks. Originally introduced in Bonilla, Chai, and Williams (2009), multitask Gaussian processes (MTGPs) augment the standard Gaussian process framework with a kernel

parameterisation that explicitly leverages covariance in outcome space across similar tasks which would otherwise be modelled with individual Gaussian processes. Note that this motivation parallels one of the identifying assumptions in the synthetic control method: that there is shared information across units, in particular the treated and untreated units. Since their introduction, MTGPs have been applied in scenarios where imputation matters. Ghassemi et al. (2015) applied MTGPs to clinical data, motivated by the need to handle sparse, noisy, and missing data; similar Durichen et al. (2014) applied MTGPs to physiological data as a means of extrapolating reliably in situations where biometric signals were jointly measured, as well as providing an interpretable matrix of task similarities for use by decision makers. Cheng et al. (2020) developed a similar approach with a particular focus on hypersparse, large scale medical data. Similarly, Futoma, Hariharan, and Heller (2017) and Urteaga et al. (2019) both develop MTGP approaches to medical prediction, where samples from the MTGP latent function posterior are provided to a deep neural network model; the former predicts sepsis onset and uses a recurrent neural network to capture temporal patterns, while the latter predicts hormonal cycles and uses a dilated convolutional neural network to extract cycle structure.

Finally Alaa and Van Der Schaar (2017) conceives of individual treatment effect (ITE) estimation as a multitask challenge, where separate response manifolds are estimated for factual and counterfactual outcomes. This last work is most similar to our contribution, in that both use multitask Gaussian processes – specifically the Linear Coregionalisation Model – to perform treatment effect estimation. However, this chapter differs in critical ways such that these two pieces are similar in method class only. First, we do not estimate treatment and control response surfaces, instead opting for an arbitrary number of shared and individual latent processes. This is because the synthetic control task is formulated as a prediction task. Second, we do this across a

sequential feature space where the same units are observed repeatedly in that feature space (i.e. the time series setting), rather than single unit observations in different regions of feature space. We do this usually for a single treated observation. Last, we remove the post-intervention information of the treated unit by construction.

5.5 Experiments

We generate synthetic data to compare the representational capacity of the MTGP model to the standard Synthetic Control Method (as implemented in *Synth*). We generate synthetic data of correlated sinusoidal time series with varying parameters, levels of additive noise, and additional patterns – such as nonlinearities, linear components, missingness, and temporal relations. This setting – which is only one representative examples among others – is designed to mimic a range of settings such as biometric signals (e.g. respiration dynamics) or mechanical signals (e.g. motor components), while presenting representational challenges that illustrate the failings of standard synthetic control approaches.

These data characteristics are so common in real-world data so as to suggest that a model that cannot account for them would be of fundamentally limited use. We describe each synthetic data case as evoking a particular benefit of this model. We then apply the model to the state smoking data in Abadie, Diamond, et al. (2010) and show that it tends to outperform Synth and recovers similar importance rankings.

5.5.1 Recovery of interpretable synthetic control

In this experiment, we generate a simple model of four sinusoidal donor units and one treated unit that is a linear combination of the donor units (with some additive noise), with weights being simplex-constrained. In other words, this is a canonical example of

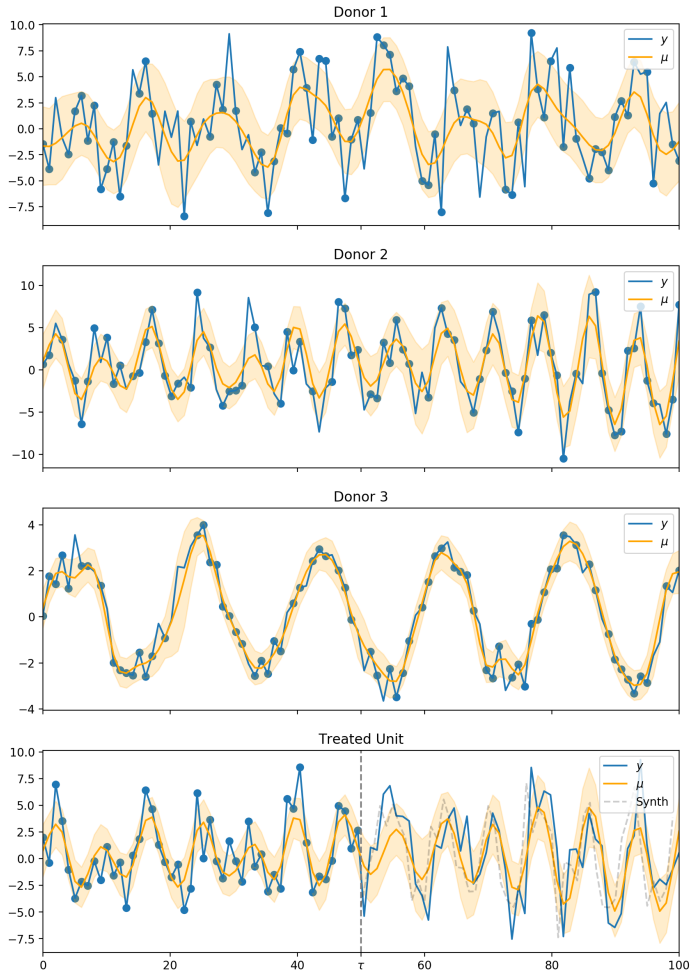


Fig. 5.3: y is the true value for each unit, and μ is the Multisynth predictive posterior mean. Top 3 graphs: 3 synthetic sinusoidal donor units, with different amplitudes, periods, and noise processes, as well as the Multisynth modelled outputs of the units. Bottom: the Multisynth posterior predictive mean μ and $\pm 2\sigma$ credibility interval (orange), and *Synth* synthetic controls (grey, dashed), before and after intervention timepoint $\tau = 50$. Both Multisynth and Synth recover the pattern of the true control.

the *Synth* context. The treated unit has a hypothetical intervention 50% of the way through the sample. Thereafter we observe no counterfactual data (we do not display the intervened curve for simplicity). Each unit is observed randomly at 50% of the possible observation points. In this example, there is no $\boldsymbol{K}^{q+m}$ - all variation is captured through a single squared exponential kernel and its corresponding coregionalisation matrix.

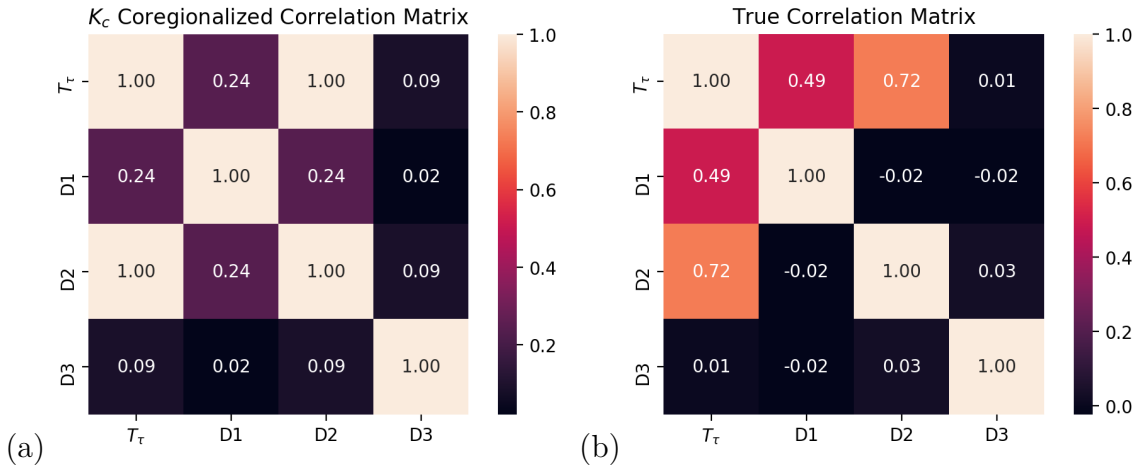


Fig. 5.4: (a) the recovered normalised correlation matrix (b) the true correlation matrix between donors and the treated unit in this experiment. While the normalisation procedure is imperfect, the relative unit-unit relationships are close.

We see in Figure 5.3 that the LCM model with a relatively uninformed structure prior almost fully recovers the true counterfactual. Synth exactly recovers it, as designed. In Figure 5.4, we also see that between the treated and donor units, both models exhibit similar relative similarities – given by the normalised correlation matrix in our model, and by the weights in the *Synth* model.

It should be noted that this is the basic and exact case for which *Synth* is expected to be perfect. In further examples, we see where the modelling assumptions inherent in *Synth* – and not present in our model – cause *Synth* to break down.

5.5.2 Extrapolation

Simplex interpolation is restrictive, and in the traditional synthetic control framework relaxing interpolation trades off interpretability. Here we show that the MTGP framework allows for extrapolation naturally while retaining interpretability.

Using the same donor and treated units as above, we modify the treated unit as a permanent upward shift by the maximum value of the donor units. We can maintain the

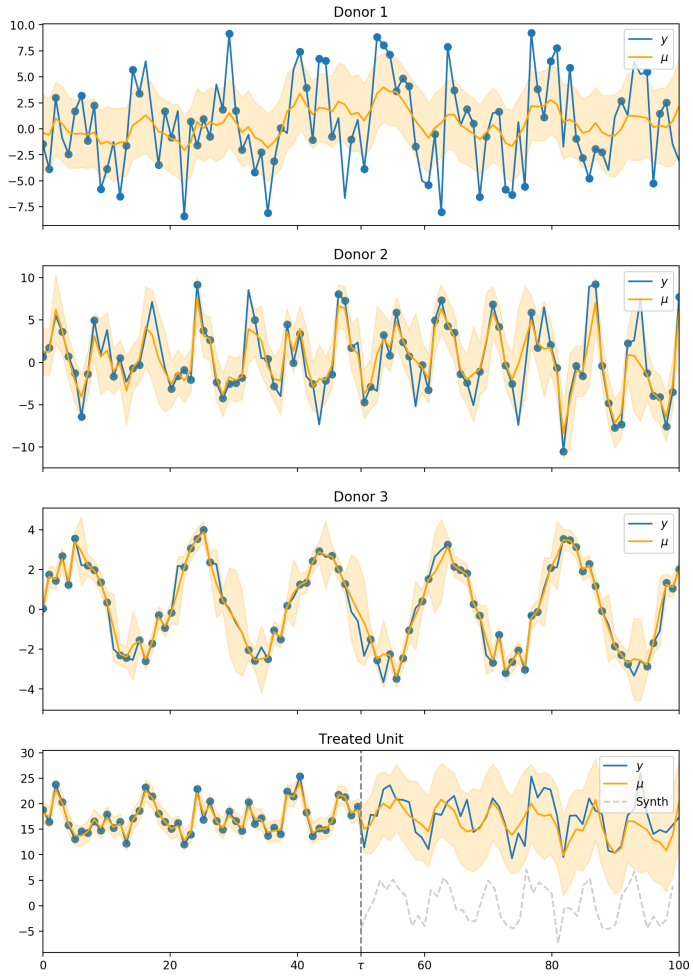


Fig. 5.5: The above experiment, with y being the true unit values and μ the posterior mean of MultiSynth, but where values of the treated unit are shifted upwards by the maxmium value of the donor units at all time points. Simplex-constrained *Synth* cannot recover the counterfactual without unconstraining its parameters and losing interpretability, while Multisynth naturally extrapolates without losing interpretability or requiring adjustment for constant or bias factors.

exact same similarity scores in this scenario, thus losing no interpretability. In Figure 5.5, we see the *Synth* model with no extrapolation fails. In practice, one could add a bias term to *Synth* to account for this shift, but not all extrapolations will be easily handled with a constant bias term. Extrapolation may be a more complex relationship that is a function of time, which requires no change in the modified LCM model.

5.5.3 Use of post-intervention information

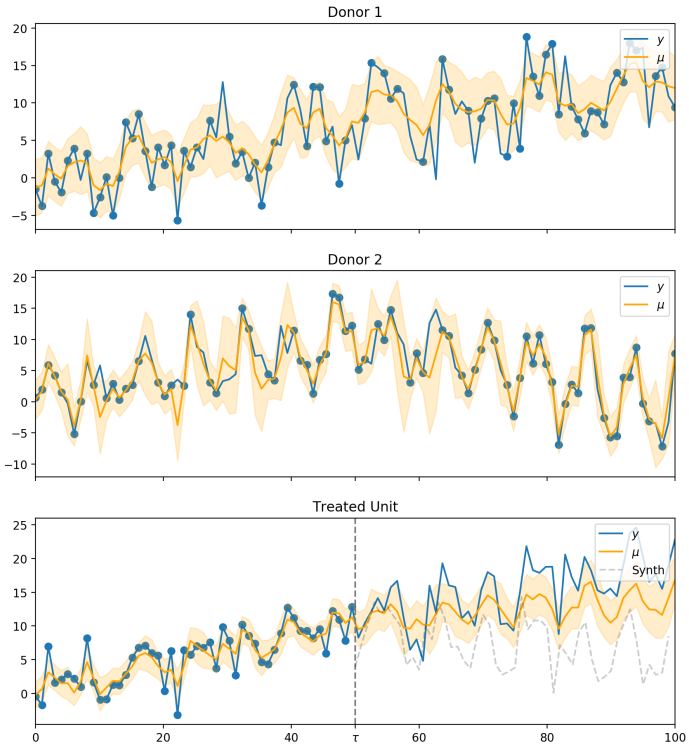


Fig. 5.6: Top 2 graphs: generated noisy sinusoidal donor units with linear trends, but where the linear trend in Donor 2 reverses at $t = \tau$. Multisynth is able to account for this reversal because it observes post-intervention data, and is able to extract relevant predictive information while not spuriously reversing the predictive trend in the treated unit. On the other hand, *Synth* only observes pre-treatment data and is unable to remove the effect of the structural change in the Donor 2 process that we assume does not occur in the treated unit.

Post-intervention information is used for three primary purposes: to better construct the shared latent basis functions, to inform cross-task similarity, and to calibrate post-interventional uncertainty. In a parametric model, the canonical multitask Gaussian process, or the multi-input single-task Gaussian process, post-intervention data is discarded. Indeed, any data where units are not observed simultaneously is disregarded. In *Synth*, post-intervention is discarded at prediction time but used by the individual modeller to select out bad donor units – something that is subjective as well as not

scaleable to high-dimensional donor information.

In Figure 5.6, we show a simple failure mode where failing to account for post-intervention data leads to spurious conclusions. Two noisy sinusoidal donor units and one sinusoidal treated unit are generated. Each unit has a linear trend. The first donor unit – the true sibling – has a co-directional linear trend with the treated unit, and no change in variance. The second donor – the false cousin – has a linear trend that reverses some time after the intervention, and with increasing variance. We see that Multisynth accounts for this change. In 5.7 we see that Multisynth adjusts its correlation matrices accordingly.

In practice, using post-intervention information is not just for better estimation of the nonparametric relationship between donors and treated units. Using post-intervention information also mimics the principled donor inclusion procedure a modeller will undertake in the *Synth* setting. A modeller will look at donors – both before and after the intervention – and judge that there is reasonable similarity to the treated unit, and no major characteristic that disqualifies it as a donor. If so, it will be selected. This process is rather subjective. One approach is to focus solely on prediction and optimise donor selection; for example, Brodersen et al. (2015) uses a full Bayesian treatment with a spike-and-slab prior over donor inclusion to automate selection. Still, without using post-intervention information, selection may be misinformed. In this nonparametric model, post-intervention information will naturally assign greater similarity to more similar units, invoking the use of all post-intervention data.

5.5.4 Accounting for self-trend patterns

In many applications, especially in social science and finance, time series are frequently "detrended": a preprocessing step is applied to data to remove trends that the modeller

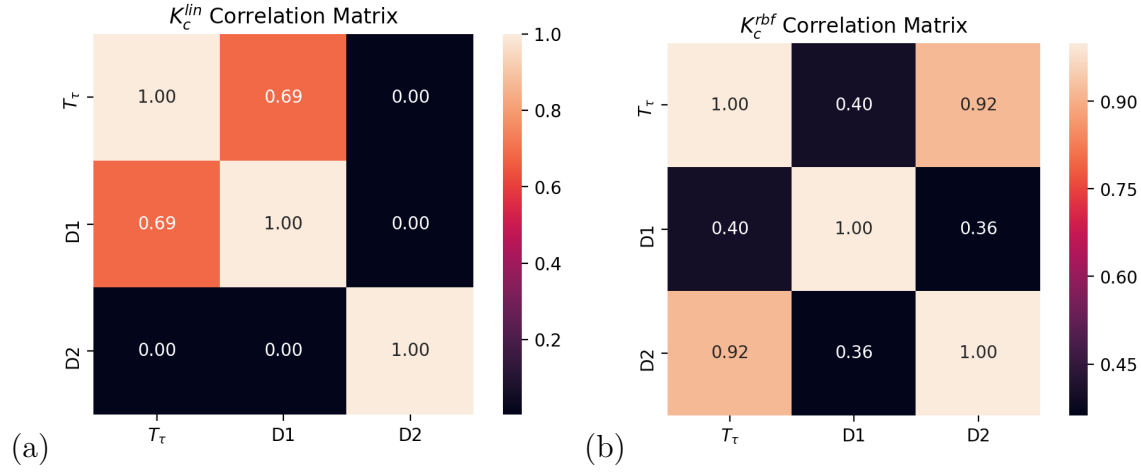


Fig. 5.7: The two normalised correlation matrices for the kernels of the Multisynth model in this experiment: (a) the linear kernel correlation matrix (b) the RBF kernel correlation matrix. Note that the treated unit T_τ and Donor 2 are dissimilar in their linear patterns – expectedly, as D2 experiences a linear trend reversal – but similar with respect to the RBF-kernel hyperparameterised latent function. This is as constructed.

believes would improperly influence the modelling task (Watson, 1986). Often, these are some function of time. Whether or not a modeller accounts for trends can have a significant impact on causal inferences drawn (Kang, 1985).

Beyond the applied implications of trends, in the *Synth* setting there is also the case that accounting for trends leads to better counterfactual prediction. We can disaggregate the treated unit into a specific trend unrecoverable by donor units, as well as patterns that share similarities with donor units. We create two donor units, drawn from Gaussian processes with a periodic and a squared exponential kernel, respectively. Our treated unit has its own quadratic time trend as well as additive periodic and squared exponential. We hope to account for its own trend as well as its distinct shared relationship with each donor.

The motivation for this example is that in real-world settings without a perfectly controlled randomised experiment, there are often confounders between treatment and outcome. For example, let the noisy sinusoidal curves represent biometric data

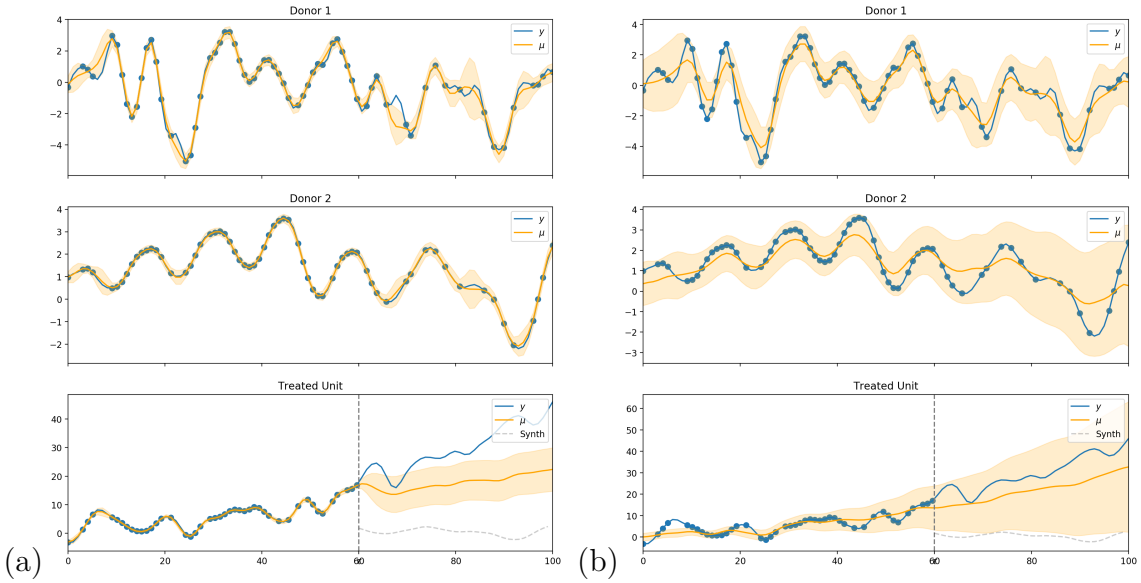


Fig. 5.8: (a) modelling the treated unit using latent functions shared across all units, i.e. no unit has a latent function that belongs solely to itself; (b) the former, but now the treated unit is given its own quadratic kernel to model a supposed exponential trend. Without a self-trend kernel, the true counterfactual is underestimated. Note that *Synth* picks up on no exponential trend in the treated unit because there is no similar information in either of the donor units.

for patients. A patient with increasing and unstable vitals may be the recipient of a preemptive treatment. Accounting for the trends can provide better counterfactual prediction in the presence of confounders without a robust theory of confounders.

5.5.5 Application to real-world data

Showing that any counterfactual estimation method is preferable to another, on real-world data, requires knowing the counterfactual of a real-world unit. Therefore, we cannot validate the usefulness of any model solely by estimating a synthetic control of an intervened unit. Instead, we take only donor units in scenarios in which an intervention is plausible and we model one them as a treatment unit at a hypothetical intervention point. Then, we compare the ability of counterfactual estimation methods to recover the true counterfactual, which is known.

We perform this analysis on the *California* dataset, which is a canonical application of the Synth method, designed to exploit a tobacco control policy instituted by California in 1988 to estimate the effect on smoking. Other states without similar shocks are used as donor units. We run 500 experiments; for each experiment, 10 donor states and 1 treated state are randomly selected, with the treated state having a hypothetical intervention in the year 1985. Multisynth used an RBF and Bias constant kernel for the latent functions, with minimal hyperoptimisation (Adam optimiser with learning rate of 0.005 for 25000 steps).

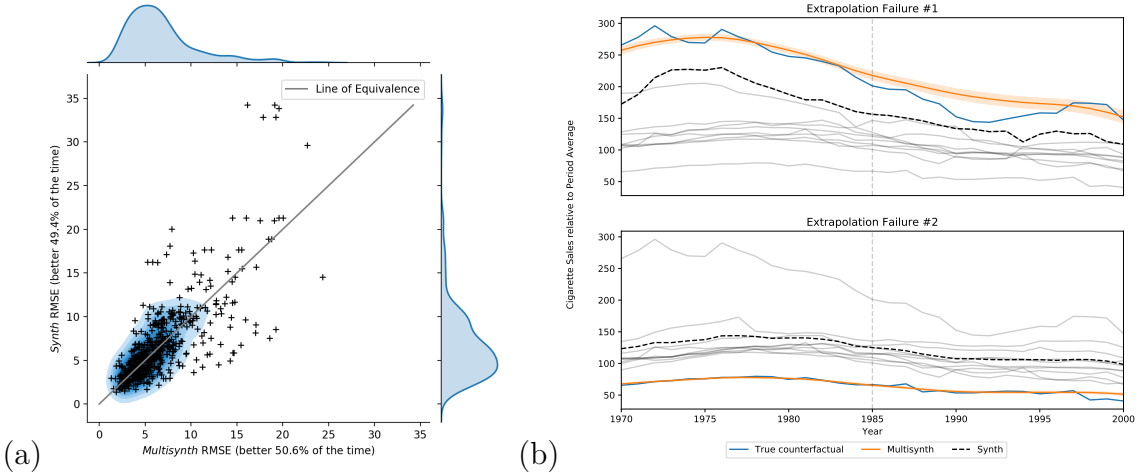


Fig. 5.9: Comparing *Synth* and Multisynth on California smoking data. (a) a distribution of the errors of *Synth* and Multisynth on 500 experiments of random groups of 10 states. In this canonical example, both methods are about equivalent. (b) An example of Synth's failure to extrapolate. Two treated units (blue) cannot be extrapolated to by *Synth* (black dotted), with *Synth* putting almost all its weight on a single donor. Multisynth (orange) does not have this problem. Donor units in grey.

In Figure 5.9(a), we see that Multisynth performs effectively as well as Synth. Equal performance on this canonical dataset is a good measure of Multisynth's reliability to perform as well as *Synth* in the canonical setting. We also highlight a failure mode identified by Multisynth in this dataset: there are some states that are strictly extrapolations that cannot be recovered by *Synth*. In Figure 5.9(b) we show how

Multisynth naturally recovers extrapolations without having to explicitly account for them in the model structure or in a pre-processing step.

5.6 Conclusion

The ability to construct time series counterfactuals for decision-making is increasingly valuable across a wide variety of fields – not just ones with a long relationship with causal inference. Further, we are applying temporal causal inference to a wider variety of problems, and we are collecting more data in this quest. These trends point towards an increasing need for temporal causal inference tools that stand up to the increasing complexity of these applications.

We have proposed the multitask Gaussian process framework as an advancement on our traditional approaches. We have argued that our MTGP model is a natural one for temporal causal inference as it:

- embodies the motivation of the traditional synthetic control approach;
- offers principled probabilistic estimates;
- uses available post-intervention information for model estimation instead of throwing it away;
- performs extrapolation rather than interpolation, with no additional modelling assumptions necessary (indeed, interpolation does not even make sense in this framework);
- provides interpretability of donor-treated and donor-donor similarity in the form of coregionalisation matrices $\boldsymbol{B}^q$, which can be broken down into representational similarity;
- allows for explicitly modelling unique (or not unique) temporal relationships in each unit;

- can be extended to incorporate multiple inputs and nonstationarity;
- and do all of the above while working in a small-data or missing-data context.

The goal of this work was to introduce this model, validate the benefits on synthetic data, and demonstrate an application to real-world examples where the synthetic control method is used. We discussed the limitations of standard approaches and augmented approaches. We argued that the benefits of all of the above properties in a single model are advantageous and frequently match the requirements of real-world deployments.

We demonstrated the ability of the model to model complex counterfactual scenarios more flexibly than the *Synth* model with fundamentally different properties of competing models. We demonstrated that the MTGP framework handles failure modes of *Synth* not due to any special modelling considerations, but by its probabilistic, nonparametric, nonlinear nature.

It should be clarified that our model is not restricted to single-dimension and temporal cases. As noted, the model can be extended to incorporate multi-input cases. In fact, one could use the same input importance discovery method as in Abadie, Diamond, et al. (2010) with this model, or simply use Automatic Relevance Determination kernels (Neal, 1996; Duvenaud, 2014). Further, it is agnostic to what feature space represents. Though the canonical task is temporal, non-temporal treatment effect inference is also being explored (Alaa and Van Der Schaar, 2017).

5.6.1 Future work

We encourage future work to study improvements to computational efficiency, applications of this model, and real-world cases where its interpretability is vital.

Inference in the MTGP framework is potentially impractically expensive leading to $\mathcal{O}((\sum_{d=1}^{m} n_d)^3 \approx m^3 n^3)$ complexity. First, it should be noted that this cost is likely

acceptable in traditional applications of *Synth*. Data where causal inference is normally used (e.g. in the social sciences, or clinical trials) is often much smaller than large and densely sampled data settings (e.g. industrial sensors). m is also usually small and pre-selected by the modeller. Additionally, we generally only need to incur this cost once, as most traditional examples of causal inference are single-shot estimation procedures rather than online. However, the problem is not always unavoidable. To take just one relatively small example, $m = 27$ units (all countries in the European Union, for example) measured at 63 time points each (one for each year since the founding of the EU), we enter a regime where typically approximations are used for Gaussian processes. In this case, a modeller might consider using inducing point methods (Quiñonero et al., 2005; Titsias, 2009), low-rank approximations of $\boldsymbol{K}_q$ (Rasmussen and Williams, 2006), or ensembles of experts via Bayesian committee machines (Tresp, 2000; Liu et al., 2018).

In particular, we think it is important to identify real-world cases where post-intervention information and extrapolation are not just beneficial, but crucial. One such case may be in the biological domain: protein engineering involves (frequently nonlinear) biological processes of different scales. Because of these processes' complexity, post-intervention information can be highly clarifying. In this particular example, our model may be useful for identifying optimal interventions and their timepoints. It is also an application where interventions are exceedingly costly with combinatorially large intervention spaces.

Second, we think there is scope to analyse the basis correlation matrices derived from $\boldsymbol{B}^q$ to interpret representational dynamics in real-world situations. We did not emphasise the value of representational similarity in this work, despite similarity in the *Synth* setting being very desirable for decision-making. For example, we might decompose behaviour of the equities markets in Countries A, B, and C, into short-term and long-term behaviour. Countries A and B may have similar long-run economic

growth dynamics, but country A shares short term financial system dynamics with country C. This might be achieved by looking at similarity for a basis kernel $k_q(\boldsymbol{x}, \boldsymbol{x}') = k_{l1}(\boldsymbol{x}, \boldsymbol{x}') \times k_{l2}(\boldsymbol{x}, \boldsymbol{x}')$ where $k_{ln}(\boldsymbol{x}, \boldsymbol{x}')$ is a linear kernel such that the kernel represents a long-term quadratic relationship, whereas short term dynamics might be represented by the squared exponential kernel.

"I already have my tombstone carved. On it is the law of counterfactuals. It's a simple equation. A counterfactual in terms of a model surgery. That's it. Everything follows from that. If you get that, all the rest comes... I can die in peace, and my student can derive all my knowledge by mathematical means."

— Judea Pearl

6

Copy, paste, infer: twin networks for counterfactual inference

This is based on joint work with Dr. Ciarán Lee and Yura Perov, conducted while on, and after, a research visitation at Babylon Health. CL originated the line of research, and YP provided further insights. I defined the research question, conducted all experiments, and led analysis. Originally published at the 2019 NeurIPS workshop on Causal Machine Learning; in submission to further venues. I thank Jonathen Richens, Kostis Gourgoulias, Anish Dhir, Saurabh Johri, and Jérémie Valée for helpful contributions and support while crafting this work.

6.1 Introduction

Answering counterfactual queries—ones that posit a *what if X had happened* scenario, where—is of interest in general reasoning and applied domains. For example, counterfactual queries can powerfully answer questions of individualised treatment effect analysis (Shalit, Johansson, and Sontag, 2017; Johansson, Shalit, and Sontag, 2016; Schulam and Saria, 2017), medical diagnosis (Richens, Lee, and Johri, 2019), legal responsibility analysis (Chockler and Halpern, 2004; Lagnado, Gerstenberg, and Zultan, 2013), fairness in machine learning models (Kusner, Loftus, et al., 2017; Kilbertus et al., 2017), and help with self-supervised learning and planning (Buesing et al., 2018; Bottou et al., 2013). Because counterfactuals are, by construction, of *observation* and *intervention*, counterfactual analysis offers a powerful framework for asking and answering questions both within and outside of the capability set of typical machine learning approaches.

Counterfactual queries allow one to estimate an outcome if an intervention had been performed in a particular observed context (as defined by some context evidence E). This unique structure of the counterfactual requires a careful consideration of query computation. The standard approach to calculating the probability of such a counterfactual query in the Structural Causal Model framework is described in Pearl (2009) and is known as the *abduction-action-prediction* method. First, the joint posterior over unobserved variables is inferred given the observed evidence E, determining a belief over the context. Second, the desired intervention is applied to the model—forcing a specific set of variables to take fixed values. Lastly, the updated model distributions (i.e. the belief over the context) are used to make predictions in the intervened model.

However, such a computation may be expensive, as the standard approach requires three separate steps, including inferring and storing a posterior over exogenous variables

given the evidence. Balke and Pearl (1994b) introduce a method, *twin networks*, that takes a different approach by changing the structure of the graph to explicitly represent counterfactuals. They argue that the new graph allows us to avoid the three-step computation and take advantages of inference efficiencies in the new graph.

However, no investigation into the specific circumstances in which twin networks are more advantageous than the standard method has been conducted. *A priori* it is not clear that twin networks are preferable to abduction-action-prediction. Twin networks require inference to be performed over a causal model whose graphical structure has up to one and a half times the number of nodes. At the same time, the new twin network may allow for inference efficiencies. If these efficiencies are larger than the cost of the increased graph size, then twin networks would be allow for computational speedups over the abduction-action-prediction setting.

In this work, an empirical analysis of twin networks is undertaken. Assuming causal identification [1], we compare the time cost of inference between both methods across graphs with different characteristics. Our analysis suggests conditions in which twin networks under and over-perform. We show that in certain contexts, twin networks are faster and less memory intensive by orders of magnitude than standard counterfactual inference in the exact inference setting. We speculate and show that the major gains from twin networks come from two sources: efficiencies in compressing the twin network graph, and efficiencies from the inference algorithm exploiting certain motifs in the graph. We then show that in some approximate inference settings, both approaches are effectively equivalent. However, twin networks can be advantageous in the approximate case depending on the evidence set supplied and the counterfactual query. We consider the implications of these findings for scenarios like reinforcement learning and bandits,

[1] to be more precise, for each query we assume the graphical structure is the true structure and we know each functional mechanism in the structure. For more detail, see 6.4.3

and how using twin networks might make counterfactual inference in these scenarios more efficient. Our results indicate that one should choose counterfactual inference methods depending on the structure of one's causal model, as well as the counterfactual question of interest; namely, the later the intervention in the topological ordering, the more efficient twin networks are. However, we hypothesise that advantages are due to the ability of the method of inference to exploit structures in the graph; we believe a full understanding of twin network advantages would come from understanding for which graph types (made into twin networks) different inference algorithms can exploit structure in answering a known counterfactual query. Last, we introduce a prototype probabilistic programming framework to leverage insights from twin networks to make native counterfactual inference more efficient.

6.2 Related work

Counterfactual frameworks—Counterfactual inference has a long history going back to Splawa-Neyman (1923), but has been rigorously mathematically formalised by Balke and Pearl (1994a); Pearl (2009), and independently by Rubin (1974), in the last few decades.

Inference for counterfactuals—The subject of inference techniques for counterfactual analysis is sparsely populated, partly because using counterfactuals is new in inference-based work, and partly because identifying the model or estimand (the quantity of interest, such as the causal effect) has been the focus of research. More recently however, new methods for estimating counterfactual queries of interest have been devised using tools from machine learning (Shalit, Johansson, and Sontag, 2017; Schulam and Saria, 2017; Perov et al., 2019; Johansson, Shalit, and Sontag, 2016; Subbaswamy and Saria, 2018). In addition, there has been much related work in

learning causal structures from observational and experimental data (Dhir and Lee, 2019; Lee and Dhir, 2020; Lee, Hart, et al., 2019; Lee and Spekkens, 2017; Janzing, Mooij, et al., 2012; Mitrovic, Sejdinovic, and Teh, 2018; Hoyer et al., 2009), which is a needed component to counterfactual questions in the Pearl causal model.

Counterfactuals in application—Interestingly, in the last few years, counterfactual inference has been leveraged to solve difficult problems in high profile sectors like medicine (Richens, Lee, and Johri, 2019; Schulam and Saria, 2017; Gilligan-Lee, 2020), computational advertising (Bottou et al., 2013), legal responsibility analysis (Chockler and Halpern, 2004; Lagnado, Gerstenberg, and Zultan, 2013), and fairness in automated decision making (Kusner, Loftus, et al., 2017; Kilbertus et al., 2017). Causal inference has even begun to be employed in quantum information (Allen et al., 2017; Lee, 2019; Lee and Hoban, 2018; Chaves, Majenz, and Gross, 2015; Barnum, Lee, and Selby, 2018; Barrett et al., 2019; Lee and Selby, 2018).

6.3 Structural Causal Models and counterfactual inference

Structural Causal Models, the *do*-operator, and counterfactuals will now formally be defined. See chapter 7 of Pearl's "Causality" (Pearl, 2009) for a more in depth discussion.

Definition 5 (Structural Causal Model) *A structural causal model specifies:*

1. *a set of unobservable, or* exogenous*, variables U, distributed according to $P(U)$,*

2. *a set of observable, or* endogenous*, variables $V = \{v_1, \ldots, v_n\}$,*

3. *a directed acyclic graph G, called the* causal structure *of the model, whose nodes are the variables $U \cup V$,*

4. *a collection of functions $F = \{f_1, \ldots, f_n\}$, where f_i is a mapping from $U \cup V \setminus \{v_i\}$ to v_i. This is symbolically represented as*

$$v_i = f_i(PA_i, u_i), \text{ for } i = 1, \ldots, n\,,$$

where PA_i denotes the parent endogenous nodes of the ith endogenous variable in G.

As the collection of functions F forms a mapping from exogenous variables U to observable variables V, the distribution over exogenous variables induces a distribution over endogenous variables. Given a Structural Causal Model M, let $Y \subseteq V$ be a set of variables of interest, and let $Y(\boldsymbol{u})$ be defined as the values of Y when the vector $\boldsymbol{u}$ of exogenous variable values is passed through the model following a topological ordering of G and the values of nodes V are evaluated.[2] Then the probability that Y takes values $\boldsymbol{y}$ is:

$$P(Y = \boldsymbol{y}) := \sum_{\{\boldsymbol{u} | Y(\boldsymbol{u}) = \boldsymbol{y}\}} P(\boldsymbol{u})\,.$$

In this manner, we can assign uncertainty over observable variables despite the fact that they are generated deterministically. An example structure of a Structural Causal Model is depicted in Fig. 6.1(a), where orange nodes denote endogenous variables and green nodes denote exogenous variables.

[2]In Pearl's terminology, $Y(\boldsymbol{u})$ is termed "the solution for Y of the set of equations F." To aid intuition, this amounts to setting $U = \boldsymbol{u}$, evaluating each function $f_i \in F$ to derive v_i in a sequence given by some topological ordering of G, and collecting the resulting values of Y.

Definition 6 (Submodel) *Let M be a Structural Causal Model, X a subset of variables in V, and $\boldsymbol{x}$ a particular realisation of X. A submodel $M_{\boldsymbol{x}}$ of M is the causal model with the same exogenous variables U and endogenous variables V as M, but where the collection of functions F is replaced with $F_{\boldsymbol{x}}$, where*

$$F_{\boldsymbol{x}} = \{f_i \mid v_i \notin X\} \cup \{f_j^{'}(PA_j, u_j) := x_j \mid v_j \in X\,.\}$$

F_x is formed by replacing all f_j corresponding to members of the set X with the new function f'_j. If $f'_j(PA_j, u_j) := x_j$ (a constant function), this is referred to as a **hard intervention** (Eberhardt and Scheines, 2007)).[3] This can also be conceived of graphically as deleting all incoming edges to X on the G associated with M while setting X to take a certain value.

Definition 7 (Effect of action, do-operator) *Let M be a causal model, X a set of variables in V, and $\boldsymbol{x}$ a particular realisation of $\boldsymbol{x}$. The effect of action $do(X = \boldsymbol{x})$ on M is given by the submodel $M_{\boldsymbol{x}}$.*

That is, the do-operator forces variables to take certain values, regardless of the original causal mechanism. Probabilities involving the *do*-operator, such as $P(Y = \boldsymbol{y} \mid \text{do}(X = \boldsymbol{x}))$, correspond to evaluating ordinary probabilities in the submodel $M_{\boldsymbol{x}}$, in this case $P_{M_{\boldsymbol{x}}}(Y = \boldsymbol{y})$.

[3] If $f'_j(PA_j, u_j)$ is not a constant function, it is referred to as a **soft intervention**.

Definition 8 (Situation $\boldsymbol{u}$) *The situation $\boldsymbol{u}$ is a particular realisation (i.e. particular values) of variables U.*

6.3.1 Standard method of counterfactual inference

Definition 9 (Counterfactual) *Let X and Y be two subsets of variables in V, where X are intervention nodes and Y are nodes of interest. The counterfactual sentence "Y would be $\boldsymbol{y}$ (in situation $U = \boldsymbol{u}$), had X been $\boldsymbol{x}$, denoted $Y_{\boldsymbol{x}}(\boldsymbol{u}) = \boldsymbol{y}$, is the equality $Y = \boldsymbol{y}$ when the value of Y in submodel $M_{\boldsymbol{x}}$ is evaluated within a particular context $U = \boldsymbol{u}$.*

As with observable variables in Definition 5, the unobserved distribution $P(U)$ allows one to define the probabilities of counterfactual statements, i.e.

$$P(Y_{\boldsymbol{x}} = \boldsymbol{y}) = \sum_{\{u | Y_{\boldsymbol{x}}(\boldsymbol{u}) = y\}} P(\boldsymbol{u}).$$

Moreover, one can define a distribution over counterfactual statements given seemingly contradictory evidence. For instance, given a set of evidence variables $E \subseteq V$, consider the probability of the following counterfactual query $P(Y_{\boldsymbol{x}} = \boldsymbol{y'} \mid E = \boldsymbol{e})$, which reads "what is probability that Y would have been $\boldsymbol{y'}$, given that E was observed to be $\boldsymbol{e}$, if X had been $\boldsymbol{x}$?"[4] Despite the fact that this query may involve an intervention that contradicts evidence (e.g. some $e_i \in E$ & $e_i \in X$), it is well defined in the Structural Causal Model framework because the intervention defines a new submodel. Indeed, we

[4]Note that by "if X had been $\boldsymbol{x}$", we mean "if X had been intervened upon to take the value $\boldsymbol{x}$ in the situation of $U = \boldsymbol{u}$."

can derive from Pearl (2009) that $P(Y_x = \boldsymbol{y}' \mid E = \boldsymbol{e})$ can be computed as follows:

$$P(Y_{\boldsymbol{x}} = \boldsymbol{y}' \mid E = \boldsymbol{e}) = \sum_{\boldsymbol{u}} P(Y_{\boldsymbol{x}}(\boldsymbol{u}) = \boldsymbol{y}')P(\boldsymbol{u} \mid E = \boldsymbol{e})\,.$$

That is, first update the distribution over exogenous variables under the observed evidence $\boldsymbol{e}$ to get $P(\boldsymbol{u} \mid E = \boldsymbol{e})$, and use this to compute the expected value of the probability that $Y_{\boldsymbol{x}} = \boldsymbol{y}'$.

The ability—given knowledge of the full causal model—to mathematically answer such queries is an incredibly powerful decision making tool. The following theorem provides the general solution to computing such probabilities.

Theorem 1 (Theorem 7.1.7 from Ref. (Pearl, 2009)) *Given a functional causal model M with exogenous distribution $P(U)$ and evidence $\boldsymbol{e}$ the conditional probability $P(Y_{\boldsymbol{x}} \mid E = \boldsymbol{e})$ can be evaluated using the three following steps:*

1. ***Abduction***— *Infer the posterior of the exogenous variables with evidence $\boldsymbol{e}$ to obtain $P(U \mid E = \boldsymbol{e})$*
2. ***Action***— *Apply the action $do(\boldsymbol{x})$ to obtain submodel $M_{\boldsymbol{x}}$*
3. ***Prediction***— *Use the submodel $M_{\boldsymbol{x}}$ with updated exogenous distribution $P(U \mid E = \boldsymbol{e})$ to compute the probability of Y, the consequence of the counterfactual.*

The abduction step, as per Definition 9, infers the situation of the observed outcome. According to the counterfactual, we make a change in the context of this situation. We use the term "situation" to refer to a particular realisation of exogenous variables U. We may not infer the situation precisely, and may instead represent a belief over situations, denoted $P(U \mid E = \boldsymbol{e})$.

Note that the abduction step, in combination with any intervention, is sufficient for defining a counterfactual. This implies that a strict counterfactual need not follow other

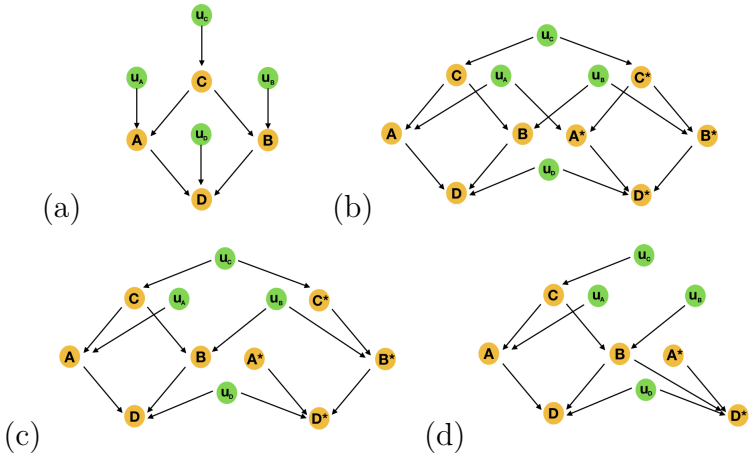


Fig. 6.1: Orange nodes are endogenous variables; green are exogenous variables. (a) an example unmodified SCM; (b) its twin network representation; (c) an intervention in the twin network on a counterfactual node; (d) a node-merged representation of the twin network given the intervention.

assumptions, and so we can ask counterfactual queries that might be unconventional: we can make inferences about $Y_{\boldsymbol{x}}$ without observing Y in the original model M, and we can condition on an additional evidence set in the prediction step (though doing so might require a further inference step; see 6.4.2 for further details).

6.4 Twin networks

A twin network is a Structural Causal Model that explicitly represents counterfactuals via a modification of the graph structure of an original Structural Causal Model. This allows one to avoid the multistep abduction-action-prediction approach to counterfactual inference. Instead, twin networks can perform counterfactual inference in a single step just as one would perform inference on a graphical model. Instead of performing a bespoke inference procedure in the original structure, twin networks modify the graph structure to enable arbitrary inference.

Constructing a twin network given a Structural Causal Model is straightforward,

and is shown in Fig. 6.1. First, one duplicates the Structural Causal Model (here we denote nodes in the duplicated model via superscript $*$); let $V = \{v_1, \ldots, v_n\}$ and $V^* = \{v_1^*, \ldots, v_n^*\}$. Then, for every endogenous node v_i^* in the duplicated, or "counterfactual", model its exogenous parent u_i^* is replaced with the original exogenous parent u_i in the "factual" model such that the original exogenous parent is now a parent of two endogenous nodes, v_i and v_i^*. Thus, the two graphs are linked only by common exogenous parents, but share the same node structure and generating mechanisms. One half of the endogenous nodes represent the "factual" part of the graph and the other half the "counterfactual" part of the graph. Let nodes of interest $Y^* = \{y_1^*, \ldots, y_k^*\} \subseteq V^*$ and counterfactual intervention nodes $X_\delta = \{x_1^*, \ldots, x_h^*\} \subseteq V^*$. Counterfactual inference in the twin networks now reduces to computing queries over counterfactual nodes (i.e. nodes in the duplicated model) given factual evidence (i.e. over endogenous variables in the original model) but counterfactual interventions (i.e. interventions over endogenous variables in the duplicated model): $P(Y^* \mid E = \boldsymbol{e}, \mathrm{do}(X_\delta = \tilde{\boldsymbol{x}}_\delta))$.[5] This computation is simply posterior inference in a graphical model.

The advantage of the twin network approach is that instead of maintaining a posterior over exogenous variables, then intervening on endogenous nodes and predicting, one computes the posterior over nodes of interest directly. This allows one to choose methods of inference to (potentially) more efficiently compute the quantity of interest. The heart of Balke and Pearl (1994b) is the observation that the ability to exploit the graph structure may speed up inference. An additional advantage is that twin networks do not need to explicitly store and manage a posterior over the exogenous nodes, as required in the standard method.

[5]Note that nodes of interest Y^* and interventions X_δ correspond to Y and $\boldsymbol{x}$ in $Y_{\boldsymbol{x}}$ in Definition 9, respectively. Obviously, $E = \boldsymbol{e}$ corresponds to $\boldsymbol{e}$ there.

6.4.1 Node merging

Twin networks as they stand, however, contain redundant information. Following Shpitser and Pearl (2007), twin networks can be compressed if a counterfactual query is known. Given nodes of interest Y^* and intervention nodes X_δ, then all counterfactual nodes that are not in X_δ or descendants of any node in X_δ are distributionally the same as their factual counterparts. As such, so long as $x_i^* \notin Y^*$, and also not in X_δ and not in the descendants of any node in X_δ, we can *merge* x_i^* into its factual counterpart x_i, eliminating x_i^* from the graph and x_i inheriting its children.[6] This process is shown in 6.1 (c)–(d), showing that 2 out of 8 endogenous nodes in the twin network can be removed, along with 3 edges. As illustrated in the following experiments, node merging is an important factor in twin network performance.

6.4.2 Additional benefits of twin networks

Counterfactual conditioning

An additional advantage of twin networks is that they allow one to condition on counterfactual evidence without a second round of inference. We term this "counterfactual conditioning". There is no reason why a counterfactual node x_i^* cannot be treated as "observed" in this approach.

For example, one might observe a patient with a medical variable of interest (health) and evidence of other medical variables (biology). A doctor may wonder what that patient's health state would have been after they had given the patient a

[6] If this $x_i^* \in Y^*$, it would also be distributionally equivalent to its factual counterpart. As a result, one could merge it with its factual counterpart, but this removes the explicit representation of the counterfactual node of interest. So long as the modeller then replaces the node(s) of interest in the original query with its factual counterpart(s), this is valid; otherwise, one needs to explicitly represent the counterfactual node(s) of interest even though distributionally it would be the same.

particular treatment, and the patient had reacted poorly to that treatment. That is, we are conditioning on the counterfactual counterpart of a *biology* variable that is affected by the treatment. This can be answered in a single query $P(\text{health}^* \mid \text{biology}, \text{do}(\text{treatment}^* = \text{drug}), x^*_{\text{reaction}} = \text{adverse})$, where x_{reaction} is e.g. some part of biology. In the standard model, one would have to perform abduction over evidence E, then intervene to create sub-model M_x, then perform a second round of abduction in the new M_x given the posterior from the previous abduction as a prior and new evidence E^*.

Note that one should be careful in choosing counterfactual nodes to condition on. If a node x^*_i is not affected by an intervention and is set such that $x^*_i = e^*_i$ *and* $x_i = e_i$ such that $e^*_i \neq e_i$, this introduces a contradiction and inference will return a joint probability of zero. This contradiction is avoided when using the merged representation of a twin network to define a query and perform inference.

Additionally, one should be aware that if x_i is not observed and is not in X_δ or the descendants of any node in X_δ, then observing $x^*_i = e^*_i$ amounts to observing $x_i = e^*_i$.

Cheaper posterior inference via d-separation

Secondly, twin networks are a useful graphical tool for found d-separated nodes we can ignore in inference. Further, it is a useful tool for finding currently unmeasured nodes that, if measured, d-separate the counterfactual node(s) from other nodes. Given some evidence $E = \boldsymbol{e}$ (which we notationally shorten to E when conditioned upon), nodes of interest Y^*, and a counterfactual intervention set X_δ, the resulting twin network graph G_δ may admit a d-separation between Y^* and U_d, where U_d is the set of exogenous nodes that are d-separated from Y^*. Therefore, $P(Y^* \mid E, U, \text{do}(X_\delta)) = P(Y^* \mid E, U \setminus U_d, \text{do}(X_\delta))$, where $Y^* \perp\!\!\!\perp U_\delta \mid \text{do}(X_\delta), E$ in the graph G_δ. While d-separated nodes can be found in both the twin network and standard methods, to do so in the standard method might require constructing a twin network. For this reason,

we conceive of twin networks as a useful graphical tool for identifying d-separated nodes and using that to our advantage to increase inference efficiency. They are also a useful graphical tool for finding nodes that make inference more efficient if measured. We give an example of such a situation below.

For example, we take the scenario from Buesing et al. (2018). This scenario is of a traditional Partially Observed Markov Decision Process (POMDP), such as found in reinforcement learning, where an agent takes an action A_T, that action affects the state of the world S_{T+1}, and receives an outcome O_{T+1}; the counterfactual query of interest is what an outcome would have been at some time point $T+1$ had some action A_T^* been taken. The node-merged twin network representation of the POMDP graph of this model makes immediately clear that it suffices to maintain a belief over the state value in order to use a node-merged twin network to avoid a potentially costly full-history abduction step. Representing the counterfactual query in a single structure immediately allows one to easily spot nodes that d-separate a node of interest from other nodes – in this case, the state value. This can be seen in Fig. 6.2, as in the node-merged twin network conditioning on the state S_T d-separates the counterfactual node of interest O_{T+1}^* from all variables upstream of S_T (those with time index $\leq T-1$). This insight motivates a different approach in reinforcement learning to what information you keep track of if you intend to perform counterfactual inference at any point. A similar result was noted in Jaques et al. (2019), albeit in the multi-agent case.

Another way to conceive of this result is a temporal tradeoff between computation and storage. An extra calculation at each time step, in this case storing a state belief (i.e. updating $P(S_t \mid u_{s,t}, A_{T-1}, S_{T-1})$ at each step), allows abduction over only n-time-steps, where n is the number of periods since the counterfactual intervention. On the other hand, if one does not store a state belief, a full rollback across all time steps must be computed. This grows in computational cost as t grows.

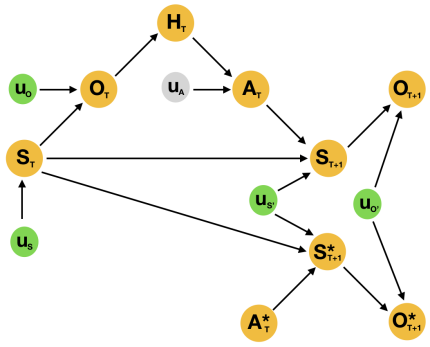


Fig. 6.2: A node-merged twin network POMDP. Orange nodes are observed, green nodes are unobserved latent nodes, and grey nodes are immutable observed nodes. By composing the counterfactual POMDP from (Buesing et al., 2018) as a node-merged twin network, then conditioning on S_T allows one to avoid inference over previous variables as this d-separates O^*_{T+1} from nodes prior to $T-1$.

Intuitively, this result is a sort of "Markov butterfly effect". It is costly and difficult to infer the long-term impact of a change to a context far in the past. However, if a hypothesised intervention happened recently, then it can be made easier to infer its counterfactual effect in-the-moment if one keeps track of the minimum beliefs necessary as shown by a node-merged twin network. In cases such as medical treatment, this can be desirable if one implements a new policy at some point in the process of treating the patient; in game-playing, this may be desirable if an agent incorporates counterfactual information at different steps.

6.4.3 Identification vs. inference in a graphical model

An intuition that may not be readily apparent is that in our application we have already assumed that the graph G and its generating mechanism functions F already *identify* quantities of interest. Identification in causal inference is usually meant in the sense that a causal query can be consistently estimated given observed data. However, when one has knowledge of the causal model, that is the causal structure, the generating functions, and the distribution over exogenous variables, then causal

queries can be consistently computed, as one can sample arbitrary amounts from the interventional distribution. Without identification, an estimate derived from a query $P(Y \mid \text{do}(X = x))$ is not guaranteed to be the same as the true interventional effect of setting $X = \boldsymbol{x}$ on Y. In our twin network, we assume that $P(Y \mid \text{do}(X^* = \boldsymbol{x}^*), E)$ is identified as a result of knowing the true causal graph and its functions, and therefore being able to generate interventional data.

6.5 Experiments: Exact inference

Here we evaluate the empirical effect of randomly generated graphs on the time cost of inference using standard counterfactual inference and using twin networks. We start by randomly generating a Structural Causal Model as defined in Def. 5 to construct a graph G with K vertices V. The structure of G is generated according to the `randomDAG` method in the R package `pcalg` (Kalisch et al., 2012) with a default size of 10 and density of 0.5 unless otherwise specified. This method generates graphs by generating an ordering of endogenous nodes and assigning an edge from X_i to X_j with $p(X_i \to X_j) = \rho$, where ρ is the density parameter and with X_i coming before X_j in the ordering.

Each exogenous variable u_i receives a generating function that samples $u_i \sim \text{Bern}(p_i); p_i \sim U[0, 1]$; each endogenous vertex X_i receives a generating function $x_i = \text{Bern}(f(PA_i, u_i))$ such that $f(PA_i, u_i) = \frac{(\sigma((PA_i \oplus u_i) \cdot \boldsymbol{\theta}_i))}{\sigma(\sum \boldsymbol{\theta}_i)}$. This functional form represents a sigmoid-constrained and normalised linear combination of PA_i, the parent variables of x_i, and each linear coefficient in $\boldsymbol{\theta}_i \sim \text{Beta}(5, 5)$. We then enumerate node values to construct conditional probability tables for each node that replicate the functional relationships mapping parent values to child values. [7]

[7] So long as these conditional probability tables can be created, then the effect of a different exogenous distribution on speed will be determined by the inference algorithm. We use variable elimination in the exact experiments, implying that changing the exogenous distribution will not effect

Given a parameter value of interest we generate 250 DAGs (with $|V| = 10$ by default) as described above holding the parameter value constant. For each parameter value we generated a counterfactual query: an evidence set of factual nodes E (default set size of $\sim 50\%$ of observable factual nodes), a node of interest y^* (the last node in the topological ordering by default), and an intervention node x^*_δ (located halfway in the ordering by default). Given this query $P(y^* \mid E, x^*_\delta)$, we measured the effect of graph size, graph density, size of the evidence set, location of the evidence set, and location of the intervention on the time it takes to infer the answer of the query. We perform the same evaluation for both the standard and the twin network approach. For inference we used the Variable Elimination (Koller and Friedman, 2009) implementation from the `pgmpy` Python package (Ankan and Panda, 2015). The time measured is wall-clock time for the factor elimination steps given an elimination ordering. This deliberately ignores irrelevant architectural implementation details.[8]

To be clear, this experimental set-up replicates perhaps the only setting where $P(U \mid E)$ is small enough to make exact inference tractable. For example, even trinary-valued exogenous nodes on a graph with 30 endogenous nodes would be parameterised by a conditional probability table with over 200 trillion entries. Outside of these restrictions, it becomes effectively impossible to use exact standard counterfactual inference; the ability to efficiently handle potentially intractably large joint distributions is the main proposed benefit of twin networks, as proposed in Balke and Pearl (1994b).

inference speed.

[8]For the standard method, we did not account for the time it takes to sample and estimate the quantity of interest. We did this because it is unclear the number of samples needed for a charitable comparison, and because the cost of the abduction step begins to dwarf the cost of the prediction step. Variable elimination to perform exact abduction is exponential in the treewidth of a graph, while the prediction (sampling) step is only $\mathcal{O}(n|V|)$, where n is the number of samples and $|V|$ is the number of nodes.

6.5.1 Results

As originally speculated in Balke and Pearl (1994b), twin networks become advantageous as the graph grows, as can be seen in Fig. 6.3 (a). However, two caveats should be noted. First, twin networks do not appear to change in relative performance as evidence size, density, or evidence topology change. Second, for the most part, twin networks outperform the standard approach, but not clearly all the time. It appears that especially for small graphs with early interventions, it is not clear that twin networks are preferable at all. However, real-world situations often involve large graphs and late interventions.

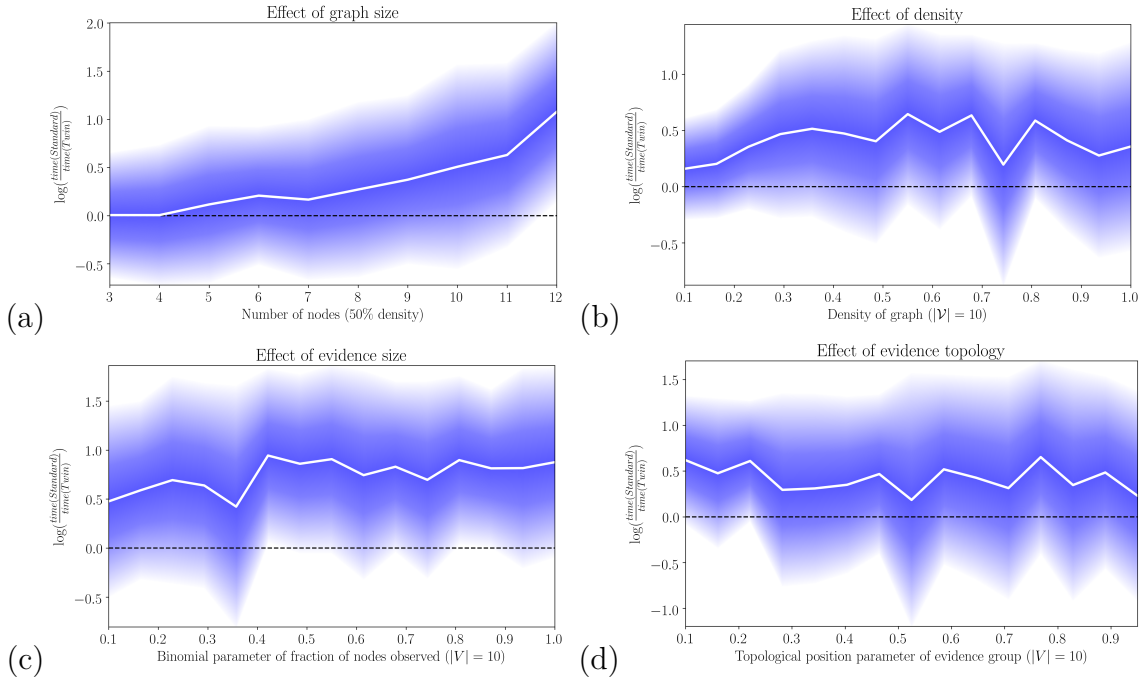
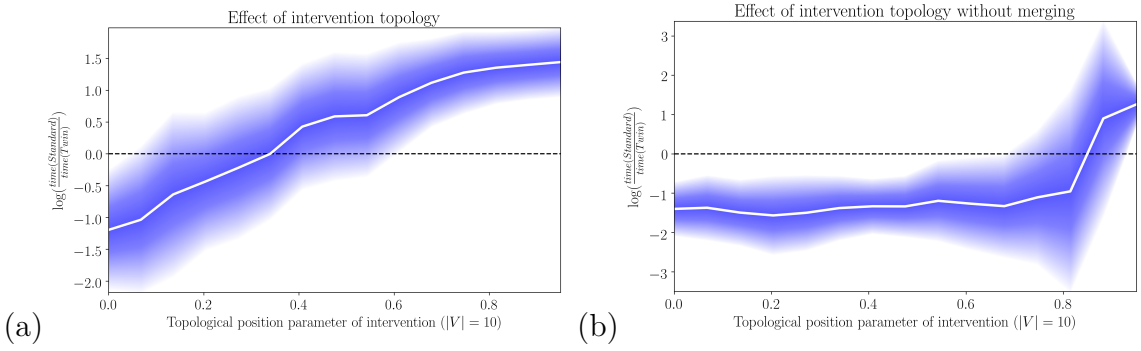


Fig. 6.3: Here we plot the median of the $\log_{10}$ of the ratio of wall times of standard and twin network counterfactual inference. The shaded regions represent $\pm$ 2 standard deviations. The black dashed line denotes parity in the time each method takes; twin networks are superior above the black dashed line, and inferior below.

We additionally see in Fig. 6.4(a) that the position in the topological order where the intervention occurs has a significant effect on performance of twin networks.

This is because node-merging reduces the size of the graph as possible. When the intervention occurs closer to the end of the topological order, its set of non-descendents grows. In the node-merged twin network with d-separated nodes removed, there are fewer factors that need to be eliminated. We validate this by observing the effect of removing node merging, as shown in Fig. 6.4(b). Without node merging, twin network inference begins to take significant time as in many cases we have to eliminate duplicate factors across a graph that approaches 36 nodes in our example. This is an important result: without node merging, twin network inference is not necessarily better than standard counterfactual inference.

Beyond these two effects, we see no consistent trend for density, evidence size, or evidence topology, as can be seen in Fig. 6.3 (b)-(d). In these cases, twin networks are consistently better, likely because they occur on a graph of size 10. This is expected as neither evidence size or topology change the nature of the counterfactual query.



(a) (b)

Fig. 6.4: The effect of intervention topology on performance, with and without node merging. Twin networks without merging perform well only when the intervention is highly likely to d-separate the node of interest from a large portion of the graph, which here occurs when p is close to 1. The shaded regions represent $\pm$ 2 standard deviations. The black line denotes performance parity.

6.5.2 Explaining twin network performance

Twin networks have two structural advantages over standard counterfactual inference: a (potentially) compressed graph, and inference that is optimised for calculating the posterior over the node(s) of interest. Both of these benefits are derived from the fact that the abduction-action-prediction approach does not consider the location of the intervention and the node of interest.

The potential compression of the graph occurs due to node merging, while the optimisation of inference occurs due to two factors: first, because the node of interest may be d-separated from some nodes in the twin network, and second, because the inference method may exploit the evidence-to-node-of-interest relationships without abduction over all exogenous variables first. Because of these advantages, it is an open question as to what the relative performance is from either method if the query involves calculating the counterfactual joint posterior over all endogenous nodes.

These insights imply that, at least in the exact case, knowing the topology of the intervention, the complexity of exogenous inference, and the dependencies in the graph can guide a modeller towards saving time in inference. This theoretical grounding motivates further exploration of the real-world implications of these insights.

It is possible, however, to improve on the standard method with these insights. By knowing the location, the intervention, and the node(s) of interest, one can find any nodes that are d-separated from the node of interest. These nodes can then be removed from the abduction step. The improvements do not, however, account for any exploitation of the graph structure by the method of inference. We show that in sampling-based and variational approximate inference, this improvement is irrelevant, rendering standard and twin network inference effectively equivalent.

6.6 Experiments: Approximate inference

In almost all real-world cases, the graph is so large or the distributions of random variables so complex that exact inference is infeasible. Instead, one would use a sampling or variational-based approximate inference scheme to estimate variables of interest. However, we find that when using sampling-based and variational approximate inference, the standard and twin network approaches are equivalent. Indeed, this shows that twin networks amount to a version of standard counterfactual inference that leverages full information of the query, and where the abduction and prediction steps are done in parallel. Because they are equivalent, the posteriors and the speed of inference are the same. However, there are some cases in approximate inference where twin networks will be advantageous: when there is a bootstrap bias from resampling from limited samples with the standard method; when the modeller wants to condition on counterfactual evidence in a single step; when there is a potentially large cost of sampling from $P(U \mid \boldsymbol{e})$; last, as previously discussed, in use-cases when a twin network can be used to find nodes that would reduce computational expense if observed. Here we first show the equivalency between the two methods, then we elaborate on these additional benefits.

For each experiment below, we generate one 20-node structural causal model with a 10-node randomly-selected evidence set. We intervene on the node halfway in the topological ordering between the first node and the node of interest. The intervention is chosen so that it causes the outcome variable to take a value that is an outlier compared to its pre-intervention distribution. To do this, we follow Tukey's rule for defining an intervention. That is, we select an intervention that causes the counterfactual outcome variable Y^* to be at least $1.5 \times IQR$ outside of its 25^{th} or 75^{th} percentiles, where IQR is the interquartile range.

We implement the approximate inference-enabled SCM in Pyro (Bingham et al., 2018). In the approximate case, an important implementation consideration occurs due to the fact that endogenous nodes are a deterministic function of its parents. An endogenous node is delta distributed, i.e. $p(X_i) = \delta(g_i(PA_i, u_i))$ where g_i is the generating function for x_i given its endogenous parent values PA_i and its exogenous parent value u_i. In order for most samples not to be rejected, we implement a "smart guide". This smart guide is a proposal distribution which constrains the proposed sample from u_i if x_i is within the evidence set. Because we generate graphs according to additive error, i.e. $x_i = g_i(PA_i, u_i) = f_i(PA_i) + u_i$, then if x_i is observed, then $u_i = x_i - f_i(PA_i)$. This prevents the vast majority of samples being given importance weights of 0 leading to highly inefficient sampling. in this scenario so long as they define the inversion function where $u_i = h(x_i, PA_i)$ if x_i is observed. For an approximation of the above process without writing a smart guide, one can set every $x_i \sim \mathcal{N}(f_i(PA_i, u_i), \sigma^2)$ where σ^2 is small. For further detailed discussion on how to construct a smart guide for SCMs, see Perov et al. (2019).

6.6.1 Posterior equivalence in the vanilla case

Here we discuss the "vanilla" approximate inference case: applying sampling-based or variational approximate inference conventionally to infer the posterior over exogenous variables. This stands in contrast to non-vanilla approximate inference, such as when there is counterfactual evidence, or when an inference procedure has been modified in some way to account for the structure of the graph.

Sampling-based approximate inference

In the case with a query such as $P(Y^* \mid E, \text{do}(X_i^* = x_i^*))$, where evidence $E = \boldsymbol{e}$ is observed only in the factual world, approximate inference by sampling in the two methods results in the same computational cost and outcome. This is because in our Structural Causal Model, all uncertainty is derived from the exogenous variables. Consider importance sampling: we take the same proposal distribution Q over the k exogenous variables in both methods, and the same m samples from Q, $S_Q = \{\boldsymbol{q_m}\}_{i=1}^{M}$. Then, the importance weights $W_Q = \{\boldsymbol{w_i}\}_{i=1}^{M}$ will be the same as the likelihood is derived only from information in the factual network. This similarly occurs with Markov Chain Monte Carlo methods. Note that for the most optimistic comparison we assume cost-free sampling in the prediction step as one could reuse samples from the empirical importance posterior or the final samples in an MCMC chain to represent draws from the posterior $P(U \mid e)$. We show in Figure 6.5 below that for the same query, evidence, intervention, and number of inference samples, the counterfactual distribution is the same.

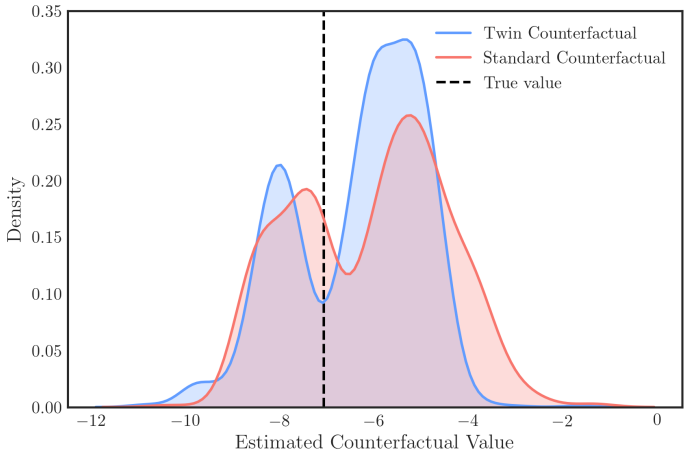


Fig. 6.5: With the same number of samples (1000), and delta-distributed endogenous nodes, the standard and twin counterfactual distributions are approximately equivalent. These distributions are not exactly equivalent as we performed importance sampling de-novo for each method, without using the same samples. The distribution may also be bimodal because evidence is taken on only half the nodes.

However, there are cases where sampling methods will be time-wise faster with a twin network. First, if there is a large cost to the resampling step after having inferred $P(U \mid \boldsymbol{e})$ then twin networks have no additional cost. Second, sampling in twin networks is faster than in the standard case. In the subgraph comprised of $\text{desc}(X_\delta)$ (the set containing the descendants of the intervention node(s)), twin networks propagate samples simultaneously through this subgraph and its counterpart subgraph in the factual side of the graph. Hence inference takes the amount of time the sampling step takes, while in the standard method case one incurs the additional cost of reusing samples for inference in a new mutilated graph. This is potentially advantageous for very large graphs or graphs where sampling is costly. The exact same savings can be achieved with the standard method if one creates a new mutilated graph and passes sample and weights through it simultaneously in parralel. However this situation exactly describes twin networks. As such, one can think of twin networks of as an elegant way to represent standard counterfactual inference in an optimised and parallelised incarnation.

Mean-field variational inference

The same phenomenon occurs in variational inference, where a variational prior $Q(U)$ is placed over exogenous variables U and then optimised. In the standard method, $Q(U)$ is placed over $P(U \mid E)$; in twin networks $Q(U)$ approximates the full query $P(Y^* \mid E, \text{do}(X_i^* = x_i^*))$, but because of the deterministic nature of endogenous nodes, $Q(U)$ is again placed over only exogenous nodes. As a result, the optimisation results in the same nodes and the same evidence to inform the variational lower-bound.

However, one consideration needs to be taken into account when using variational inference. One should be very clear to not resample from the variational distribution $Q(U)$ if it makes the mean-field assumption such that $P(U \mid E) \approx Q(U) = \Pi_i^K Q(u_i)$, as is most often the case in variational inference. Resampling from $Q(U)$ to estimate the

quanity of interest means sampling from each $Q(u_i)$ independently, breaking the very likely scenario that exogenous variables become correlated given evidence E. As a result, one should use samples from the variational distribution. However, one should also be careful to use the final samples, as they reflect the outcome of iterative optimisation.

6.6.2 Resampling-based approximation error

The above equivalences hold as the number of samples taken in the prediction step tends towards infinity. However, if the number of samples taken in the prediction is limited, then standard counterfactual inference will suffer a resampling error if the prediction step resamples from the constructed posterior. We show this in Figure 6.6 below: in an example case, the counterfactual posterior appears more poorly predicted when the number of samples (during inference and prediction) is small, though we note the significant overlap in errors. This error occurs for the same reason that bootstrapping error occurs; naive sampling from a set of samples may not use all samples available. In order to get around this problem, a modeller has to be conscientious to use the full set of importance weights and samples exactly once each, rather than sampling from it, in forming an estimate of the quantity of interest. In implementation, this is an important step which we address natively in a novel probabilistic programming engine introduced in Section 6.7.

6.6.3 Counterfactual conditioning inducing different likelihoods

Another case where the single-step procedure of twin networks is advantageous is the counterfactual conditioning case. As described in Section 6.4.2, counterfactual conditioning allows us to find the posterior $P(Y^* \mid E, \text{do}(X_i^* = x_i^*), E^*)$ in a single inference step. This both saves on inference time (as we do not need to set up two

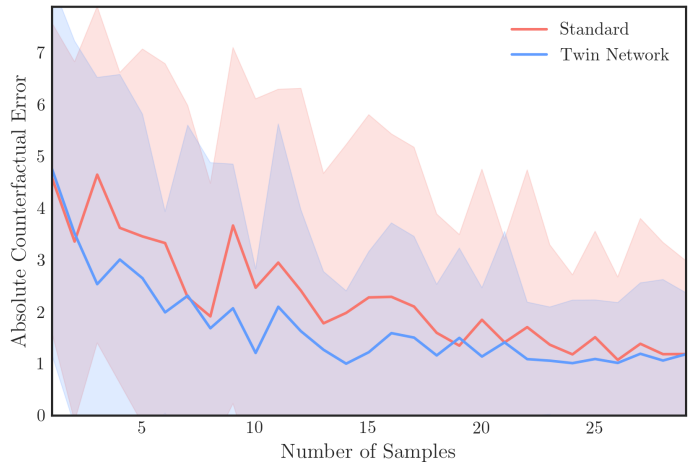


Fig. 6.6: Representative results of absolute errors ($\pm 2\sigma$) from trials comparing resampling errors in the twin and standard cases using approximate inference. Whereas twin networks use all samples, the standard methodology resamples from an empirical posterior, introducing a bootstrapping bias.

separate inference procedures—one for abduction, and then second to update the posterior in face of evidence in the counterfactual world) as well as error, as we do not suffer the above resampling error. We illustrate this in Figure 6.7, where we compare the time and approximation error of the counterfactual conditioning query to the two-step standard case.

6.6.4 Sampling from an expensive posterior

Last, it may be the case that the prediction step in the standard case is expensive because sampling from the approximate posterior is expensive. For example, consider a proposal or variational distribution where a sample is taken and then passed through a large neural network, as in the case of using implicit distributions. This practice is increasingly common because of the ability of implicit distributions to represent complex distributions without the need for complex proposal or variational distributions. With twin networks, in this example, a sample is passed through the neural network

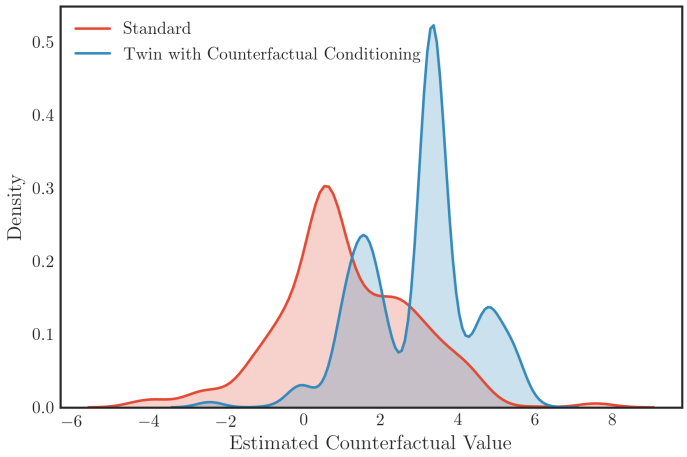


Fig. 6.7: Comparing posteriors with and without counterfactual evidence. The twin network posterior represents the posterior over the node of interest when counterfactual evidence is observed. The standard posterior represents the posterior over the node of interest, but without the second round of inference required to account for counterfactual evidence. The disparity demonstrates that the implicit posterior over the unobserved endogenous differs after one round of inference for the two. The posterior is equivalent once the standard method performs a second round of inference in the submodel.

a single time, rather than twice as in the standard method.

6.7 Probabilistic programming languages for causal reasoning

As machine learning has progressed, we have increasingly sought to build probabilistic programming languages (PPLs) that make it easy to express probabilistic models, integrate non-probabilistic models into probabilistic models, and automate inference within these models. However, PPLs have traditionally ignored causal models, and causal reasoning in general. Pyro is the only major PPL with a native *do*-operator (Bingham et al., 2018), though it can be implemented with some care in Edward (Tran, Hoffman, et al., 2017).

Due to this deficit, we concurrently developed MultiVerse, a prototype PPL for easily

performing associational, interventional, and counterfactual reasoning with approximate inference (Perov et al., 2019). The key insight of MultiVerse is that the inference procedure – rather than the structure of the program – can be modified to achieve the same effect that creating a twin network representation of the program achieves.

MultiVerse is based on several insights learned from twin networks. It encapsulates the conclusion that in sampling-based approximate inference both the standard and twin network methods are equivalent. MultiVerse does this not by explicitly representing a twin network with additional nodes (to avoid the additional cost of storing extra nodes), but in modifying the abduction-action-prediction inference procedure to accomplish the same effect. Unlike the abduction-action-prediction framework (which naively finds the full posterior over all exogenous variables), MultiVerse takes into consideration the nodes of intervention and of interest. Then, MultiVerse performs inference only over relevant variables, which is based on the insight of d-separation from the twin network. Last, MultiVerse propagates the effect of the intervention through its descendants, instead of resampling non-descendants. This is the same outcome as achieved by node merging. Despite no explicit twin network graph, the outcome is the same as in twin network inference.

In MultiVerse variables are nodes just as in PPLs; however, nodes also have an assigned exogenous variable, making the representation analogous to a Structural Causal Model, but without having to explicitly write exogenous nodes. Then, MultiVerse contains a bespoke importance sampling method where each node maintains a history of its samples from the proposal, as well as whether the node has been intervened upon. Last, MultiVerse uses a "smart guide" that modifies the proposal distribution to avoid sampling exogenous nodes when the exogenous value can be fully determined due to its functional relationship with its exogenous child node.

As a result, MultiVerse is, to the best of our knowledge, the first PPL with native

associational, interventional, and counterfactual reasoning. MultiVerse was crafted to leverage several inference optimisations for counterfactual reasoning based on the insights found in twin networks.

6.8 Future work & limitations

This work has been a first empirical foray in examining twin networks, and as a result suffers from limitations. Based on our observations here, we suggest a priority list of future questions that should help to illuminate which settings twin networks are advantageous in pure computational cost.

First, these experiments should be augmented to be comprehensive across types of graphs and the inference methods used with them. We suspect that most of the explanation for twin work efficiency using exact inference is determined by how the inference method leverages efficiencies due to the structure of the graph. Thus, twin network inference depends on graph types and motifs, and whether the inference method can exploit such structure. We considered randomly generated graphs, but other types of graphs may have different properties. Causal graphs with unique structures often show up in practice, such as POMDPs, cell signaling pathways, and risk factor-symptom-disease diagnostic models. For this reason, a comprehensive study of twin network effectiveness should examine advantages depending on graph types with different motifs. In tandem, other inference approaches should be explored depending on the type of graph. We considered only one general exact inference method, but others should be explored if they are deemed to be more advantageous depending on graph type. We believe that a theoretical analysis of the effect of graph structures and inference methods on relative performance should be done to subsume this empirical work.

Additionally, there is space for future research on counterfactual inference given

other types of unknown components of the causal model. This work considered only uncertainty on nodes. On the other hand, for example, twin networks may perform differently when some functions in F are unknown. The function uncertainty could be represented as additional exogenous parent nodes for each node. This turns the problem into one of uncertainty over variables, but with a specific structure of exogenous variables. Alternatively, inference could be performed when the graph G is uncertain. In this setting, twin network inference may be more efficient when the possible graphs have substructures that admit efficiencies given the inference method. When they do not, twin network inference may be less efficient.

6.9 Conclusion

We conducted, to our knowledge, the first empirical study of twin networks for counterfactual inference. In summary, twin networks modify the graph structure rather than the inference procedure in order to perform counterfactual inference. In the exact inference case, we show that twin networks are computationally advantageous as the graph size increases and as the intervention topology allows for more compressed graphical representations through node merging. However, we speculate that any advantage is due to the structure of the graph and how exact inference exploits this structure, and we encourage theoretical analysis that evaluates performance across graph types. We discussed how twin networks reduce the computational load to perform the same counterfactual query in approximate inference under certain conditions. Last, we introduce a prototype probabilistic programming engine that leverages the insights of twin networks in approximate inference.

These insights imply that those wishing to perform counterfactual inference can use the twin network structure to think about potential inference advantages given a

counterfactual query, dependency structure, and intervention and evidence locations. Casting the inference task in a twin network may reduce unnecessary computation and predict counterfactual quantities in parallel.

7

Conclusion

7.1 Advancing causal interpretable systems

This thesis has explored the role interpretability and causality have in machine learning systems for human decision making. The core motivation of this thesis is the belief that interpretability and causality are critical for integrating machine learning-derived information into our most sensitive decision-making processes. Further, the two are important to move the locus of decision-making from humans to computers. Doing so would allow more decisions to be made in a shorter amount of time, and without the bottleneck of human availability.

However, to do so, we face research bottlenecks at each stage of the Ladder of Causation. In associative reasoning, the key challenges are defined by the problems that the majority of modern machine literature attempts to solve: namely, how to have the right data for the problem at hand, how to learn useful representations from the data, how to build provably good estimators on top of them, how to compute these more efficiently, and how to derive action-related insights from them. In interventional reasoning, the key challenges are to generate or discover valid interventional data, model interventional distributions, and deploy machine learning systems that intervene in real-world settings. In counterfactual reasoning, the key challenges are to perform counterfactual inference, incorporate counterfactual feedback, and derive context-critical insights from counterfactual information.[1]

Each piece of research in this thesis constitutes an attempt to tackle at least one bottleneck on each rung of the ladder. In PIGEBaQ (Chapter 3), we address the challenge of deriving interpretable notions of expected change of outputs with respect to inputs, as a tool for deiciosn-makers. In the survey (Chapter 4), we accumulate a new dataset of a sensitive topic and apply a PIGEBaQ-like approach to aggregate and interpret subjective expert perspectives on the automatability of work. In Multisynth (Chapter 5), we introduce a new model for modelling the interventional distribution in an increasingly common problem domain. And in Twin Networks (Chapter 6), we build a library for performing counterfactual inference and evaluate an understudied but promising method for performing counterfactual inference.

[1]This is to say nothing of other very important research questions that enable these tools, such as how to develop optimal interaction interfaces, schemas for acceptable definitions of interpretability and fairness and when to choose them, and how to make large-scale computing easy.

7.1.1 Charting the path forward

This research was conducted concurrently with the emergence of causality and interpretability as major research themes in the machine learning community. To be clear, they have been very important topics for at least a century within the statistics and computational intelligence communities. However, they have been never more relevant and energetically discussed as distinct topics as now. Presumably, this is because of our need to understand the systems we are deploying, and because they also provide insights for how to make system perform better in the real-world. The downside of being part of an early emergence is that it only becomes clear how much further there is to go, and how rapidly the dominant problems change.

Here I lay out a research agenda, set of applications, and a handful of practices that, as I see it, would accelerate our transition to trusted, interpretable, causal, and highly-performant machine systems.

Merging interpretability and causality

In this thesis, interpretability has been explored as a tool for understanding in associative inference. However, this is not the full scope of interpretability. Interpretability is really just a set of characteristics that we can ascribe to any level of inference. For the very simple reasons of confounding and causal misspecification, interpretability is likely best achieved via causal inference. PIGEBaQ breaks this guidance because, as argued, non-causal relations can still be useful; if we are sufficiently aware of when we are making causal or non-causal interpretations, we need not throw the baby out with the bathwater.

Ultimately, however, interpretability tools serve as inputs to decisions, and we want our decisions to be based on models of the world that generalise to the contexts we intend to act in. Further, we usually want our interpretations to be ones we *can* act on

in the first place. That might suggest that the most satisfying and useful explanations are the ones that give us a policy lever we can use.

For these reasons, I foresee a future where interpretations are causal, and models are built with causal tools so as to provide easy interpretations. Several areas of research will expedite this integration, and are active areas of research:

Human-in-the-Loop Interpretation Satisfying explanations are difficult to program, but intuitive for humans to generate. A machine that can capture this implicit human knowledge may more quickly learn how to generate satisfying and correct explanations. There are several current approaches: observational data is augmented with human-generated explanations (Hind et al., 2019); humans modify model components to reflect human explanations (Kim, Glassman, et al., 2015); humans evaluate the explainability of model outputs (Lage et al., 2018). With regards to SCM-derived explanations, one might imagine the SCM as an explanation itself. For example, a doctor-facing system may output a set of possible SCMs for a patient's condition, and the doctor may select the subset that she feels is the better explanation. The system would then iteratively learn from this expressed knowledge.

Effect Bounds under Causal Misspecification Most models assume causal sufficiency, or *no unmeasured confounders.* In reality this is rarely true. As a result, it is important for decision-makers that are relying on estimates of treatment effects to have a lower and upper bound on the effects based on misspecified graphs, confounding, and non-ignorability, as in Balke and Pearl (1997); Geiger, Janzing, and Schölkopf (2014); Shalit, Johansson, and Sontag (2017)

Distributions over Explanatory Models Maintaining distributions over DAGs has been unattractive because the number of possible DAGs is superexponential in

the number of nodes. However, very rarely is the posited DAG correct. The difference between a correct and incorrect DAG can mean catastrophic outcomes, so maintaining distributions over explanatory DAGs becomes very important. This research area would likely benefit from heuristics, graph embeddings, and other non-conventional approaches that make maintaining distributions over models feasible in practice.

Frontiers in causality

In order to merge causal learning with machine learning, we need to (a) learn *causal* representations on which we can make predictions or take actions, (b) incorporate causal insights to improve generalisation and efficiency in machine learning, and (c) endow machines with the ability to make interventions in deployed systems. These three categories broadly summarise the research themes we see in the literature. However, they also sit on a spectrum of difficulty. Most research has focused on learning causal representations. On the other hand, deploying machines that can make and learn from interventions, in physical systems or complicated digital ones, is very complicated. However, this latter task is the ultimate dream: systems which can make interventions in the real world, acting as humans do, and hopefully learning better than they do.

First, in order to incorporate causal reasoning, we must have representations that allow us to learn and act on causal information. By representations, I mean model structures (hyperparameters) and derived information (parameters) that capture causal information.

Causal priors allow us to inform the structure of models so that the model might better use causal information. A model might separately learn from interventional and observational data, learning contextual weights for how valuable interventional data is and why. More effort might be conducted to endow models to ask for data it would most benefit from; an intervention on an uncertain causal relationship is

likely more useful than an additional observational datapoint on a relationship that is well-understood. A secondary model, which sits outside of the task model, may attempt to learn causal relationships while providing feedback for the representations being learned by the task model. Alternatively, a model manager may maintain a distribution over models that each represent different causal assumptions, iteratively prioritising models whose assumptions are most generalisable or validated by interventions (Peters, Bühlmann, and Meinshausen, 2016).

Having a flexible way of adapting models to incorporate causal information seems useful. I do not think it is unlikely that we would have, for example, models whose final structure partly or fully resembles a Structural Causal Model. A Graph Neural Network is an ontologically close example. Given some causal regularisation parameter during learning, a GNN may learn an object-relational structure that mirrors the real SCM. If the SCM is an imprecise model for the real causal phenomena – for example, measured variables do not usefully correspond to nodes – a causal GNN would represent a mid-point for flexibly learning pseudo-causal structure. Additionally valuable research would search for low-dimensional graph embeddings which address the fact that SCM-space is super-exponential in the number of nodes, while also maintaining conditional independences in graph latent space.

This raises the question of which causal concepts are useful regularisers or priors on the parameters of the model. The graph structure is one prior; the Independence of Causal Mechanisms capturing the asymmetry of cause and effect is another; robustness to known interventions is a further. Beliefs of conditional independencies may be useful priors for weights. Causal regularisation may be useful for learning pseudo-causal representations; for example, Bengio et al. (2017) propose a scheme for learning latent encodings that maximise interventional independence. Additionally, we could maintain beliefs over causal regime; use a mixture of experts that act in regions of feature-space

depending on which causal regime we detect due to nonstationarity.

Second, as mentioned in Chapter 2, we can incorporate the fact that causal information can allow for better generalisation, transfer, learning efficiency, predictive performance, and seemingly-impossible queries.

Last, we can put the two together by endowing our deployed systems with the ability to actively intervene. Useful causal information only comes from causal assumptions of interventional distributions, or data that identifies a causal effect – namely, interventions, or approximations thereof. In order to benefit from interventions, machine systems need to determine which interventions to make, be able to make interventions, and learn from them. In practice, this might look like like an enclosed vertical farm that continuously experiments with optimal configurations of lighting, irrigation, and airflow, and soil metabolic profile; a high-throughput chemical lab that can generate and test a new compound to learn more about the mechanistic behaviour of a different compound; an automated at-home physician that can randomly prescribe non-harmful treatments (like walking n kilometres or sleeping h hours) to identify the healthiest lifestyle for a patient. The ability to make interventions – or create interventional data, by introducing instruments, for example – is the last step towards closing the loop on causal reasoning in the real world.

In practice, many desirable interventions tend to be impractical, costly, or immoral. Luckily, there are two positive trends. First, many real-world applications of machine learning today admit interventions. Indeed, many already do; A/B testing, for example, is widely used in online service optimisation. Other use cases involve simulation of environments where interventions can increasingly be simulated (e.g. for robotic grasping and manipulation, or by post-hoc examining algorithms for fairness), multi-context transfer across environments can be done (e.g. training self-driving cars trained in an Arizona summer and a Northern Michigan winter), and experiments can be

carried out in the real world (e.g. Amazon Go stores changing their product layouts, or randomising chatbot responses based on perceived emotional state of the chatter). Second, we are actively beginning to shape our environment to make it easier for machine learning tools to interact with the real world. Once we have reached a critical mass of use, it seems natural that it will be easier to relinquish interventional ability to algorithms. At the extreme, your doctor is an algorithm (De Fauw et al., 2018) (or an algorithm helps your doctor (McKinney et al., 2020)) and perpetually in your pocket; the Nobel Prize in Chemistry is won by high throughput labs with artificially intelligent experiment designers (Stokes et al., 2020); parts of monetary and fiscal policy are entirely automated, conducting hundreds of small experiments each year, almost in secret, in order to learn how to best stabilise and grow an economy (Chakraborty and Joseph, 2017). Each of these has deployed components that would seem exotic a decade ago.

In order to reach this dream, we must trust the systems we are creating. While there are many mechanisms for trust (Brundage et al., 2020), most of the energy has been spent on first understanding what machine learning systems are doing. I surmise this is because we humans seek explanations in order to understand how a system works so we can predict future action better. The shortcomings of this intuition should be evident: it would imply we need to ensure our explanations do indeed robustly predict future action. I propose that causal explanations are the basic framework for doing so. However, we have a long path from the wild west of interpretability of today to distributionally-robust causal explanations. In the meantime, any effort to advance interpretability should help us to fill in the map, identifying promising slipstreams and avoiding treacherous jungles or dead-end cliffs.

Major future applications of causality and interpretability

If research is to be guided, it should be guided both by theory and by application. Not only are we increasingly seeing series of interpretable, causal machine learning in application, but several attractive and ambitious domains are defined by these advances. Here, I speculate on four of the most impactful and quickly growing applications.

Personalised Healthcare Personalised healthcare is an almost ideal embodiment of a counterfactual: given patient i's observed measurements $\{x_{i1}, x_{i2}, \ldots, x_{im}\}$, what would have been the patient's outcome y_i had I intervened with a drug at time t? The forward looking case ("*what* will *be the outcome?*") embodies an interventional query. By the nature of healthcare, confounders are significant and we can only measure a small amount of available biological processes at any one time. As a result, for healthcare to be truly personalised, we need to understand the interventional distributions for each patient, and we need to learn from counterfactual feedback. Each patient might have their own causal structure, which may be related to the causal structures of similar patients, which might share their fundamentals with known global models of biology.

Computational social science Causal inference is critical for assertion-making in social science because environmental variables are much more difficult to control, and confounders are much more numerous. Further, phenomena are likely difficult to characterise, requiring representational approximations. As a result, modelling complex social systems likely requires tools that can flexibly learn representations, integrate local causal information into those representations, and identify desirable interventions (either for learning's sake, or to produce desired outcomes). For example, building a full *psychohistory*-like simulation of the global economy would be valuable to test the effect of shocks, such as pandemics, environmental catastrophe, housing

market collapses, or technology shocks. To do so would require representing the global economy at the right scale (national? municipal?), incorporate causal knowledge or a priori micro axioms (from RCTs and natural experiments, for example), and output interpretations relevant for policy to advise desirable interventions. Current national economic models are already based on structural equation models (Brayton et al., 1996), and calibrated simulations are increasingly used for economic networks (Aymanns et al., 2018; Elliott, Golub, and Jackson, 2014).

Automating the scientific laboratory Accelerating the rate of scientific discovery is potentially one of the most welfare enhancing technologies we could build. Long-term economic growth, which generally translates to distributed wellbeing, is hypothesised to be significantly dependent on the rate of advancement of the technical frontier. However, advancing science is costly, both monetarily and time-wise. Viewing science as a combination of hypothesis-forming, tool-building, experimentation, and creative interpretation, we can bring these together by uniting a causal intelligence with the hardware tools to run experiments. If this view of science is correct – that it is a long game of causal discovery where interpretable insights lead to new hypotheses – then it does not seem in principle impossible to automate. Indeed, advances in automated scientific laboratories (Stokes et al., 2020; Zymergen, 2018) and model explanations for natural sciences (Roscher et al., 2020) are already creating the capacity to do so.

Generative models of physics and biology Humans are likely to be involved in creative scientific exploration for a long time, and so we can advance the rate of discovery by building better exploration tools. Some areas of science have neatly constrained substructures that admit a causal generative model. A particularly good example is biology. The Central Dogma describes the neatly-structured flow of information

from genome to proteome, which uses the metabolome to generate a phenome. If we can account for this structure, we could build a simulation (generative model) of biology to play with, without needing an organism in front of us. Similarly, we are increasingly good at producing machine learning-calibrated simulations of physics, which enable us to perform generative design or explore latent spaces previously unseen. For more complex systems, such as climate systems, imposing a causal structure prior can address dimensionality (Chalupka et al., 2016; Runge et al., 2019), allowing us to simulate longer-term forecasts under the presence of hypothesised interventions (such as geoengineering).

7.1.2 A last glimpse of the vision

Recent progress in machine learning has allowed us to dream of extraordinary futures of symbiotic human-machine intelligence. It would be nothing short of magical to live a life where personalised healthcare is a solved and banal topic; policymakers in a meeting can interact in real-time with information that incorporates the latest causal knowledge to simulate different policy outcomes; "dark laboratories" operate 24/7 without a single human, experimenting with and discovering unknown frontiers of nature; and a high school student can simulate climate dynamics or generate unseen organisms on their laptop for homework. Indeed, as we saw, some of these are already able to be automated. These shifts are uniquely enabled by systems that can ask and search for the answer to *why* and *what if* questions, and return insights that humans can understand and use.

However, if we are to reach such futures, we must be able to trust the tools we are using. Trust requires knowing that our tools are adequate at picking up robust, causal relationships. It also requires knowing we can interpret our systems and their outcomes (at least until humans are removed from decision-making entirely). These will also help

us tremendously in ensuring that the future we build is not just a exciting one, but one that is prosperous for all: human, environment, machine, and beyond.

Bibliography

Abadie, Alberto, Alexis Diamond, et al. (2010). "Synthetic Control Methods for Comparative Case Studies: Estimating the Effect of California's Tobacco Control Program". In: *Journal of the American Statistical Association* 105.490, pp. 493–505. ISSN: 0162-1459. DOI: 10.1198/jasa.2009.ap08746.

Abadie, Alberto and Javier Gardeazabal (2003). "The economic costs of conflict: A case study of the Basque country". In: *American Economic Review* 93.1, pp. 113–132. ISSN: 00028282. DOI: 10.1257/000282803321455188.

Acemoglu, Daron and Pascual Restrepo (2018). "Artificial Intelligence, Automation, and Work". In: *The Economics of Artificial Intelligence: An Agenda*. University of Chicago Press, pp. 197–236.

Alaa, Ahmed M. and Mihaela Van Der Schaar (2017). "Bayesian inference of individualized treatment effects using multi-task Gaussian processes". In: *Advances in Neural Information Processing Systems*. Vol. 2017-Decem, pp. 3425–3433.

Allen, John-Mark A et al. (2017). "Quantum common causes and quantum causal models". In: *Physical Review X* 7.3, p. 031021.

Álvarez, Mauricio A, Lorenzo Rosasco, and Neil D Lawrence (2012). "Kernels for Vector-Valued Functions: A Review". In: *Foundations and Trends in Machine Learning* 4.3, pp. 195–266.

Amjad, Muhammad, Devavrat Shah, and Dennis Shen (2018). "Robust synthetic control". In: *Journal of Machine Learning Research* 19, pp. 1–51. ISSN: 15337928.

Amos, Storkey (2008). "When Training and Test Sets Are Different: Characterizing Learning Transfer". In: *Dataset Shift in Machine Learning*, pp. 2–28. ISBN: 9780262170055. DOI: 10.7551/mitpress/9780262170055.003.0001.

Ankan, Ankur and Abinash Panda (2015). "pgmpy: Probabilistic graphical models using python". In: *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*. Citeseer.

Arkhangelsky, D et al. (2019). *Synthetic Difference in Differences*. Tech. rep. National Bureau of Economics Research.

Arntz, Melanie, Terry Gregory, and Ulrich Zierahn (2016). "The Risk of Automation for Jobs in OECD Countries: A Comparative Analysis". In: *OECD Social, Employment and Migration Working Papers* 2.189, pp. 47–54. ISSN: 14777029. DOI: 10.1787/5jlz9h56dvq7-en.

Athey, Susan and Guido W Imbens (2017). "The State of Applied Econometrics: Causality and Policy Evaluation". In: *Journal of Economic Perspectives* 31, pp. 3–32. DOI: 10.1257/jep.31.2.3.

Autodesk (2017). *What Is Generative Design*.

Autor, David H (2013). "The "Task Approach" to Labor Markets: an Overview". In: *Journal for Labour Market Research* 46.3, pp. 185–199. ISSN: 1614-3485. DOI: 10.1007/s12651-013-0128-z. arXiv: arXiv:1011.1669v3.

Aymanns, Christoph et al. (2018). "Models of Financial Stability and Their Application in Stress Tests". In: *Handbook of Computational Economics* 4, pp. 329–391. ISSN: 15740021. DOI: 10.1016/bs.hescom.2018.04.001.

Babibker, Housam and Randy Goebel (2017). "Using KL-divergence to focus Deep Visual Explanation". In: *NIPS 2017 Symposium on Interpretable Machine Learning*.

Baehrens, David et al. (2010). "How to Explain Individual Classification Decisions". In: *Journal of Machine Learning Research* 11, pp. 1803–1831. ISSN: 1532-4435.

Bahadori, Mohammad Taha et al. (2017). "Causal Regularization". In: arXiv: 1702.02604 [cs.LG].

Bakhshi, Hasan et al. (2017). *The Future of Skills Employment in 2030*.

Balke, Alexander and Judea Pearl (1994a). "Counterfactual probabilities: Computational methods, bounds and applications". In: *Uncertainty Proceedings 1994*. Elsevier, pp. 46–54.

— (1994b). "Probabilistic evaluation of counterfactual queries". In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

— (1997). "Bounds on treatment effects from studies with imperfect compliance". In: *Journal of the American Statistical Association* 92.439, pp. 1171–1176.

Bareinboim, Elias, Andrew Forney, and Judea Pearl (2015). "Bandits with unobserved confounders: A causal approach". In: *Advances in Neural Information Processing Systems*. Vol. 2015, pp. 1342–1350.

Barnum, Howard, Ciarán M Lee, and John H Selby (2018). "Oracles and query lower bounds in generalised probabilistic theories". In: *Foundations of physics* 48.8, pp. 954–981.

Barratt, Shane (2017). "InterpNET: Neural Introspection for Interpretable Deep Learning". In: arXiv: 1710.09511.

Barrett, Jonathan et al. (2019). "The computational landscape of general physical theories". In: *npj Quantum Information* 5.1, pp. 1–10.

Bengio, Emmanuel et al. (2017). "Independently Controllable Features". In: arXiv: 1703.07718.

Ben-Michael, Eli, Avi Feller, and Jesse Rothstein (2018). "The augmented synthetic control method". In: arXiv: 1811.04170.

Besserve, Michel et al. (2018). "Group invariance principles for causal generative models". In: *International Conference on Artificial Intelligence and Statistics*, pp. 557–565.

Bickel, Steffen, Michael Brückner, and Tobias Scheffer (2009). "Discriminative Learning Under Covariate Shift". In: *Journal of Machine Learning Research* 10, pp. 2137–2155. ISSN: 15324435. DOI: 10.1063/1.4771869.

Bingham, Eli et al. (2018). "Pyro: Deep Universal Probabilistic Programming". In: *Journal of Machine Learning Research*.

Bloebaum, Patrick et al. (2018). "Cause-Effect Inference by Comparing Regression Errors". In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*.

Bollen, Kenneth A. (2005). "Structural Equation Models". In: *Encyclopedia of Biostatistics*. Chichester, UK: John Wiley & Sons, Ltd. DOI: 10.1002/0470011815.b2a13089.

Bonilla, Edwin V., Kian Ming A. Chai, and Christopher K.I. Williams (2009). "Multi-task Gaussian Process prediction". In: *Advances in Neural Information Processing Systems*. ISBN: 160560352X.

Bottou, Léon et al. (2013). "Counterfactual reasoning and learning systems: The example of computational advertising". In: *The Journal of Machine Learning Research* 14.1, pp. 3207–3260.

Brayton, Flint et al. (1996). "A Guide to FRB/US : A Macroeconomic Model of the United States". In: *Finance and Economics Discussion Series* 1996.42, pp. 1–45. ISSN: 19362854. DOI: 10.17016/feds.1996.42.

Brodersen, Kay H et al. (2015). "Inferring Causal Impact Using Bayesian Structural Time-Series Models". In: *The Annals of Applied Statistics* 9.1, pp. 247–274. DOI: 10.1214/14-AOAS788.

Brundage, Miles et al. (2020). "Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims". In: arXiv: 2004.07213.

Buesing, Lars et al. (2018). "Woulda, coulda, shoulda: Counterfactually-guided policy search". In: *arXiv preprint arXiv:1811.06272*.

Bureau of Labor Statistics (2017). *Occupational Employment Statistics*.

Carvalho, Diogo V, Eduardo M Pereira, and Jaime S Cardoso (2019). *Machine learning interpretability: A survey on methods and metrics*.

Chakraborty, Chiranjit and Andreas Joseph (2017). *Machine Learning at Central Banks*. Tech. rep. DOI: 10.2139/ssrn.3031796.

Chalupka, Krzysztof et al. (2016). "Unsupervised Discovery of El Niño Using Causal Feature Learning on Microlevel Climate Data". In: *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*.

Chaves, Rafael, Christian Majenz, and David Gross (2015). "Information–theoretic implications of quantum causal structures". In: *Nature communications* 6.1, pp. 1–8.

Cheng, Li-Fang et al. (2020). "Sparse Multi-Output Gaussian Processes for Medical Time Series Prediction". In: *BMC Medical Informatics and Decision Making* 20.

Chockler, Hana and Joseph Y Halpern (2004). "Responsibility and blame: A structural-model approach". In: *Journal of Artificial Intelligence Research* 22, pp. 93–115.

Choi, Edward et al. (2016). "RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism". In: *Advances in Neural Information Processing Systems*.

Chu, Wei and Zoubin Ghahramani (2005). "Gaussian processes for ordinal regression". In: *Journal of Machine Learning Research* 6.Jul, pp. 1019–1041.

De Fauw, Jeffrey et al. (2018). "Clinically applicable deep learning for diagnosis and referral in retinal disease". In: *Nature Medicine* 24.9, pp. 1342–1350. ISSN: 1546170X. DOI: 10.1038/s41591-018-0107-6.

De G. Matthews, Alexander G. et al. (2017). "GPflow: A Gaussian Process Library using TensorFlow". In: *Journal of Machine Learning Research* 18.40, pp. 1–6.

Dhir, Anish and Ciarán M. Lee (2019). "Integrating overlapping datasets using bivariate causal discovery". In: arXiv: 1910.11356 [stat.ML].

Ding, Yi and Panos Toulis (2020). "Dynamical Systems Theory for Causal Inference with Application to Synthetic Control Methods". In: arXiv: 1808.08778v3.

D.O.E., U.S. (2017). "College Scorecard Data". In: *College Scorecard Data*.

Doshi-Velez, Finale and Been Kim (2017). "Towards A Rigorous Science of Interpretable Machine Learning". In: arXiv: 1702.08608 [stat.ML].

Doudchenko, Nikolay and Guido W Imbens (2016). *Balancing, Regression, Difference-In-Differences and Synthetic Control Methods: A Synthesis*. Tech. rep. National Bureau of Economic Research.

Du, Mengnan, Ninghao Liu, and Xia Hu (2019). "Techniques for interpretable machine learning". In: *Communications of the ACM* 63.1, pp. 68–77.

Durichen, Robert et al. (2014). "Multi-task Gaussian process models for biomedical applications". In: *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE, pp. 492–495. ISBN: 978-1-4799-2131-7. DOI: 10.1109/BHI.2014.6864410.

Duvenaud, David (2012). *Bayesian Quadrature: Model-based Approximate Integration*. Tech. rep. University of Cambridge.

— (2014). "Automatic model construction with Gaussian processes". PhD thesis. University of Cambridge.

Eberhardt, Frederick and Richard Scheines (2007). "Interventions and Causal Inference". In: *Philosophy of Science* 74.5, pp. 981–995. ISSN: 00318248, 1539767X.

Elliott, Matthew, Benjamin Golub, and Matthew O Jackson (2014). "Financial networks and contagion". In: *American Economic Review* 104.10, pp. 3115–53.

Engelbrecht, A. P., I. Cloete, and J. M. Zurada (1995). "Determining the significance of input parameters using sensitivity analysis". In: *International Workshop on Artificial Neural Networks*. Springer, Berlin, Heidelberg, pp. 382–388. DOI: 10.1007/3-540-59497-3_199.

Fisher, Ronald Aylmer (1926). "The arrangement of field experiments". In: *Journal of the Ministry of Agriculture* 33, pp. 503–513.

Forrester, Alexander I.J, András Sóbester, and Andy J Keane (2007). "Multi-fidelity optimization via surrogate modelling". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463.2088, pp. 3251–3269. ISSN: 14712946. DOI: 10.1098/rspa.2007.1900.

Frey, Carl Benedikt and Michael A Osborne (2017). "The future of employment: how susceptible are jobs to computerisation?" In: *Technological Forecasting and Social Change* 114, pp. 254–280.

Futoma, Joseph, Sanjay Hariharan, and Katherine Heller (2017). "Learning to detect sepsis with a multitask Gaussian process RNN classifier". In: *Proceedings of the 34th International Conference on Machine Learning*. Vol. 3. International Machine Learning Society (IMLS), pp. 1914–1922. ISBN: 9781510855144.

Geiger, Philipp, Dominik Janzing, and Bernhard Schölkopf (2014). "Estimating Causal Effects by Bounding Confounding." In: *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 240–249.

Gelfand, Alan E et al. (2004). "Nonstationary multivariate process modeling through spatially varying coregionalization". In: *Test* 13.2, pp. 263–312.

Genton, Marc G (2002). "Classes of kernels for machine learning: a statistics perspective". In: *Journal of Machine Learning Research* 2.2, pp. 299–312. ISSN: 1533-7928.

Ghassemi, Marzyeh et al. (2015). "A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data". In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. Vol. 1, pp. 446–453. ISBN: 9781577356998.

Ghavamzadeh, Mohammad, Yaakov Engel, and Michal Valko (2016). "Bayesian Policy Gradient and Actor-Critic Algorithms". In: *Journal of Machine Learning Research* 17.66, pp. 1–53.

Gilligan-Lee, Ciarán (2020). "Causing trouble". In: *New Scientist* 246.3279, pp. 32–35.

Gilpin, Leilani H et al. (2018). "Explaining explanations: An overview of interpretability of machine learning". In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, pp. 80–89.

Goodman, Bryce and Seth Flaxman (2017). "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"". In: *AI Magazine* 38.3. DOI: 10.1609/aimag.v38i3.2741.

Goovaerts, Pierre (1997). *Geostatistics for natural resources evaluation*. Oxford University Press on Demand.

Goudet, Olivier et al. (2017). "Causal Generative Neural Networks". In: DOI: 10.7910/DVN/3757KX. arXiv: 1711.08936.

Grace, Katja et al. (2018). "When will AI exceed human performance? Evidence from AI experts". In: *Journal of Artificial Intelligence Research* 62, pp. 729–754.

Graham, Logan et al. (2020). *Artificial intelligence in hiring: Assessing impacts on equality*. Tech. rep. Institute for the Future of Work.

Grewal, Dhruv, Anne L Roggeveen, and Jens Nordfaelt (2017). "The Future of Retailing". In: *Journal of Retailing* 93.1, pp. 1–6. ISSN: 00224359. DOI: 10.1016/j.jretai.2016.12.008.

Guidotti, Riccardo et al. (2018). "A Survey Of Methods For Explaining Black Box Models". In: arXiv: 1802.01933.

Gunter, Tom et al. (2014). "Sampling for Inference in Probabilistic Models with Fast Bayesian Quadrature". In: *Advances in Neural Information Processing Systems*.

Guo, Ruocheng et al. (2018). "A Survey of Learning Causality with Data: Problems and Methods". In: arXiv: 1809.09337.

Hara, Satoshi and Kohei Hayashi (2016). "Making Tree Ensembles Interpretable". In: *ICML Workshop on Human Interpretability in Machine Learning*.

Hart, Jordan (2017). *Keras Ordinal Categorical Crossentropy*. https://github.com/JHart96/keras_ordinal_categorical_crossentropy.

Hazlett, Chad and Yiqing Xu (2018). "Trajectory Balancing: A General Reweighting Approach to Causal Inference With Time-Series Cross-Sectional Data". In: *SSRN Electronic Journal*. DOI: 10.2139/ssrn.3214231.

Hechtlinger, Yotam (2016). "Interpretation of Prediction Models Using the Input Gradient". In: arXiv: 1611.07634 [stat.ML].

Hill, Steven M. et al. (2019). "Causal learning via manifold regularization". In: *Journal of Machine Learning Research* 20, pp. 1–14. ISSN: 15337928. DOI: 10.17863/CAM.44718.

Hind, Michael et al. (2019). "TED: Teaching AI to explain its decisions". In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 123–129.

Hoyer, Patrik O et al. (2009). "Nonlinear causal discovery with additive noise models". In: *Advances in Neural Information Processing Systems*, pp. 689–696.

Janzing, Dominik, Lenon Minorics, and Patrick Bloebaum (2020). "Feature relevance quantification in explainable AI: A causal problem". In: ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. Online: PMLR, pp. 2907–2916.

Janzing, Dominik, Joris Mooij, et al. (2012). "Information-geometric approach to inferring causal directions". In: *Artificial Intelligence* 182, pp. 1–31.

Jaques, Natasha et al. (2019). "Social influence as intrinsic motivation for multi-agent deep reinforcement learning". In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, pp. 3040–3049.

Johansson, Fredrik, Uri Shalit, and David Sontag (2016). "Learning representations for counterfactual inference". In: *Proceedings of the 33rd International Conference on Machine Learning*, pp. 3020–3029.

Julious, Steven A. and Mark A. Mullee (1994). "Confounding and Simpson's paradox". In: *BMJ* 309.6967, p. 1480. ISSN: 14685833. DOI: 10.1136/bmj.309.6967.1480.

Kalisch, Markus et al. (2012). "Causal Inference Using Graphical Models with the R Package pcalg". In: *Journal of Statistical Software* 47.11, pp. 1–26.

Kallus, Nathan and Angela Zhou (2018). "Confounding-robust policy improvement". In: *Advances in Neural Information Processing Systems*. Vol. 2018-Decem, pp. 9269–9279.

Kang, Heejoon (1985). "The Effects of Detrending in Granger Causality Tests". In: *Journal of Business & Economic Statistics* 3.4, p. 344. ISSN: 07350015. DOI: 10.2307/1391720.

Kansky, Ken et al. (2017). "Schema networks: zero-shot transfer with a generative causal model of intuitive physics". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1809–1818.

Kilbertus, Niki et al. (2017). "Avoiding discrimination through causal reasoning". In: *Advances in Neural Information Processing Systems*, pp. 656–666.

Kim, Been, Elena Glassman, et al. (2015). *iBCM: Interactive Bayesian case model empowering humans via intuitive interaction*.

Kim, Been, Julie Shah, and Finale Doshi-Velez (2015). "Mind the Gap: A Generative Approach to Interpretable Feature Selection and Extraction". In: *Advances in Neural Information Processing Systems*.

Kim, Hyun-Chul and Zoubin Ghahramani (2012). "Bayesian classifier combination". In: *Artificial Intelligence and Statistics*, pp. 619–627.

Kinn, Daniel (2018). "Synthetic Control Methods and Big Data". In: arXiv: 1803.00096.

Kocaoglu, Murat et al. (2018). "Causalgan: Learning causal implicit generative models with adversarial training". In: *6th International Conference on Learning Representations*. ISBN: 1600216161.

Koh, Pang Wei and Percy Liang (2017). "Understanding Black-box Predictions via Influence Functions". In: *Proceedings of the 34th International Conference on Machine Learning*, pp. 1885–1894.

Koller, Daphne and Nir Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press. ISBN: 0262013193, 9780262013192.

Kumar, D., A. Wong, and G. W. Taylor (2017). "Explaining the Unexplained: A CLass-Enhanced Attentive Response (CLEAR) Approach to Understanding Deep Neural Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1686–1694. DOI: 10.1109/CVPRW.2017.215.

Kusner, Matt J, Joshua Loftus, et al. (2017). "Counterfactual fairness". In: *Advances in Neural Information Processing Systems*, pp. 4066–4076.

Kusner, Matt J., Chris Russell, et al. (2018). "Causal Interventions for Fairness". In: arXiv: 1806.02380.

Lage, Isaac et al. (2018). "Human-in-the-loop interpretability prior". In: *Advances in Neural Information Processing Systems*, pp. 10159–10168.

Lagnado, David A, Tobias Gerstenberg, and Ro'i Zultan (2013). "Causal responsibility and counterfactuals". In: *Cognitive science* 37.6, pp. 1036–1073.

Lake, Brenden M et al. (2017). "Building machines that learn and think like people". In: *Behavioral and Brain Sciences* 40.2017, pp. 1–101. ISSN: 14691825. DOI: 10.1017/S0140525X16001837.

Lakkaraju, Himabindu, Stephen H. Bach, and Jure Leskovec (2016). "Interpretable Decision Sets". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. New York, New York, USA: ACM Press, pp. 1675–1684. ISBN: 9781450342322. DOI: 10.1145/2939672.2939874.

Lee, Ciarán M (2019). "Device-independent certification of non-classical joint measurements via causal models". In: *npj Quantum Information* 5.1, pp. 1–5.

Lee, Ciaran M and Anish Dhir (2020). "System and method for generating a graphical model". In: *US Patent 10706104*.

Lee, Ciarán M., Christopher Hart, et al. (2019). "Leveraging directed causal discovery to detect latent common causes". In: arXiv: 1910.10174 [stat.ML].

Lee, Ciarán M and Matty J Hoban (2018). "Towards device-independent information processing on general quantum networks". In: *Physical review letters* 120.2, p. 020504.

Lee, Ciarán M and John H Selby (2018). "A no-go theorem for theories that decohere to quantum mechanics". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2214, p. 20170732.

Lee, Ciarán M and Robert W Spekkens (2017). "Causal inference via algebraic geometry: feasibility tests for functional causal structures with two binary observed variables". In: *Journal of Causal Inference* 5.2.

Lemeire, Jan and Erik Dirkx (2006). "Causal Models as Minimal Descriptions of Multivariate Systems". In: pp. 1–16.

Lipton, Zachary C. (2018). "The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery." In: *Queue* 16.3, 31–57. ISSN: 1542-7730. DOI: 10.1145/3236386.3241340.

Liu, Haitao et al. (2018). "Generalized robust Bayesian committee machine for large-scale Gaussian process regression". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 7, pp. 4898–4910. ISBN: 9781510867963.

Loftus, Joshua R et al. (2018). "Causal Reasoning for Algorithmic Fairness". In: arXiv: 1805.05859.

Louizos, Christos et al. (2016). "The variational fair autoencoder". In: *4th International Conference on Learning Representations*.

Lucas, Robert E. (1976). "Econometric policy evaluation: A critique". In: *Carnegie-Rochester Confer. Series on Public Policy* 1.C, pp. 19–46. ISSN: 01672231. DOI: 10.1016/S0167-2231(76)80003-6.

Lundberg, Scott M and Su-In Lee (2017). "A unified approach to interpreting model predictions". In: *Advances in Neural Information Processing Systems*, pp. 4765–4774.

Madras, David et al. (2019). "Fairness through causal awareness: Learning causal latent-variable models for biased data". In: *FAT* 2019 - Proceedings of the 2019 Conference on*

Fairness, Accountability, and Transparency. New York, New York, USA: Association for Computing Machinery, Inc, pp. 349–358. ISBN: 9781450361255. DOI: 10.1145/3287560.3287564.

Manyika, James et al. (2017a). "A Future that Works: Automation, Employment, and Productivity". In: *McKinsey Global Institute*.

Manyika, James et al. (2017b). "Harnessing Automation for a Future that Works". In: *McKinsey Global Institute*.

Matthews, Alexander G. de G. et al. (2017). ""GPflow: A Gaussian process library using TensorFlow"". In: *Journal of Machine Learning Research* 18.40, pp. 1–6.

McKinney, Scott Mayer et al. (2020). "International evaluation of an AI system for breast cancer screening". In: *Nature* 577.7788, pp. 89–94. ISSN: 14764687. DOI: 10.1038/s41586-019-1799-6.

Mitchell, Tom and Erik Brynjolfsson (2017). "Track how technology is transforming work". In: *Nature* 544.7650, pp. 290–292.

Mitrovic, Jovana, Dino Sejdinovic, and Yee Whye Teh (2018). "Causal Inference via Kernel Deviance Measures". In: *arXiv preprint arXiv:1804.04622*.

Modi, Chirag and Uros̲ Seljak (2019). "Generative Learning of Counterfactual for Synthetic Control Applications in Econometrics". In: arXiv: 1910.07178v1.

Moraffah, Raha et al. (2020). "Causal Interpretability for Machine Learning – Problems, Methods and Evaluation". In: arXiv: 2003.03934.

Moravec, Hans (1990). *Mind Children: the Future of Robot and Human Intelligence*. Harvard University Press, 214 p. ISBN: 9780674576186. DOI: cblibrary-fut.

Mothilal, Ramaravind K., Amit Sharma, and Chenhao Tan (2020). "Explaining machine learning classifiers through diverse counterfactual explanations". In: *FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617. ISBN: 9781450369367. DOI: 10.1145/3351095.3372850.

Murdoch, W. James et al. (2019). "Definitions, methods, and applications in interpretable machine learning". In: *Proceedings of the National Academy of Sciences of the United States of America* 116.44, pp. 22071–22080. ISSN: 10916490. DOI: 10.1073/pnas.1900654116.

National Academies of Sciences, Engineering and Medicine (2017). Washington, D.C.: National Academies Press. ISBN: 978-0-309-45402-5. DOI: 10.17226/24649.

National Center for O*NET Development (Jan. 19, 2018). *O*NET OnLine*.

Neal, Radford M (1996). "Bayesian Learning for Neural Networks". PhD thesis. University of Toronto. ISBN: 9783642124648. DOI: 10.1007/978-1-4612-0745-0.

Ng, Andrew Y and Michael I Jordan (2002). "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes". In: *Advances in Neural Information Processing Systems*, pp. 841–848.

Nguyen, Trung V. and Edwin V. Bonilla (2014). "Collaborative multi-output gaussian processes". In: *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*. ISBN: 9780974903910.

O'Hagan, Anthony (1991). "Bayes-Hermite quadrature". In: *Journal of Statistical Planning and Inference*.

Osborne, Michael (2010). "Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature". PhD thesis. University of Oxford.

Osborne, Michael A, Roman Garnett, Stephen J Roberts, et al. (2012). "Bayesian Quadrature for Ratios". In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics* 22, pp. 832–840. ISSN: 15337928.

Osborne, Michael, Roman Garnett, Zoubin Ghahramani, et al. (2012). "Active learning of model evidence using Bayesian quadrature". In: *Advances in Neural Information Processing Systems*, pp. 46–54.

Parbhoo, Sonali, Mario Wieser, and Volker Roth (2018). "Causal Deep Information Bottleneck". In: arXiv: 1807.02326.

Parra, Gabriel and Felipe Tobar (2017). "Spectral mixture kernels for multi-output Gaussian processes". In: *Advances in Neural Information Processing Systems*. Vol. 2017, pp. 6682–6691.

Pearl, Judea (2009). *Causality (2nd edition)*. Cambridge University Press.

Pearl, Judea and Dana Mackenzie (2018). *The Book of Why: the New Science of Cause and Effect*. Basic Books.

Pedregosa-Izquierdo, Fabian (2015). "Feature extraction and supervised learning on fMRI : from practice to theory". Theses. Université Pierre et Marie Curie - Paris VI.

Perov, Yura et al. (2019). "Multiverse: Causal reasoning using importance sampling in probabilistic programming". In: *arXiv preprint arXiv:1910.08091*.

Peters, Jonas, Peter Bühlmann, and Nicolai Meinshausen (2016). "Causal inference by using invariant prediction: identification and confidence intervals". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 5.78, pp. 947–1012.

Peters, Jonas, Dominik Janzing, and Bernhard Schölkopf (2017). *Elements of Causal Inference*. MIT Press. ISBN: 9780262037310.

Pfingsten, Tobias (2006). "Bayesian Active Learning for Sensitivity Analysis". In: *LNAI* 4212, pp. 354–365.

Poulos, Jason (2017). "RNN-based counterfactual prediction". In: arXiv: 1712.03553.

Quiñonero, Joaquin et al. (2005). *A Unifying View of Sparse Approximate Gaussian Process Regression*. Tech. rep., pp. 1939–1959.

Rasmussen, Carl Edward and Zoubin Ghahramani (2003). "Bayesian Monte Carlo". In: *Advances in Neural Information Processing Systems* 15.1, pp. 489–496. ISSN: 10495258. DOI: 10.1016/j.ress.2010.04.014.

Rasmussen, Carl Edward and Christopher Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Remes, Sami, Markus Heinonen, and Samuel Kaski (2017). "Non-stationary spectral kernels". In: *Advances in Neural Information Processing Systems*. Vol. 2017, pp. 4643–4652.

Reserve, United States Federal (2018). "Consumer Credit Series Reference". In:

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016a). "Model-Agnostic Interpretability of Machine Learning". In: arXiv: 1606.05386 [stat.ML].

— (2016b). "Why Should I Trust You?" In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. New York, New York, USA: ACM Press, pp. 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778.

Richens, Jonathan G, Ciarán M Lee, and Saurabh Johri (2019). "Counterfactual diagnosis". In: *arXiv preprint arXiv:1910.06772*.

Rojas-Carulla, Mateo et al. (2018). "Invariant models for causal transfer learning". In: *Journal of Machine Learning Research* 19. ISSN: 15337928.

Roscher, Ribana et al. (2020). "Explainable Machine Learning for Scientific Insights and Discoveries". In: *IEEE Access* 8, pp. 42200–42216. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2976199.

Rubin, Donald B. (1974). "Estimating causal effects of treatments in randomized and nonrandomized studies". In: *Journal of Educational Psychology*. ISSN: 00220663. DOI: 10.1037/h0037350.

Runge, Jakob et al. (2019). "Inferring causation from time series in Earth system sciences". In: *Nature Communications* 10.1, pp. 1–13. ISSN: 20411723. DOI: 10.1038/s41467-019-10105-3.

Santoro, Adam et al. (2016). "Meta-Learning with Memory-Augmented Neural Networks". In: *Proceedings of the 33rd International Conference on Machine Learning*. Vol. 4, pp. 2740–2751. ISBN: 9781510829008.

Schölkopf, Bernhard (2019). "Causality for Machine Learning". In: arXiv: 1911.10500 [cs.LG].

Schölkopf, Bernhard et al. (2012). "On Causal and Anticausal Learning". In: *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, p. 8.

Schulam, Peter and Suchi Saria (2017). "Reliable decision support using counterfactual models". In: *Advances in Neural Information Processing Systems*, pp. 1697–1708.

Schwab, Patrick and Walter Karlen (2019). "CXPlain: Causal Explanations for Model Interpretation under Uncertainty". In: *Advances in Neural Information Processing Systems*.

Shalit, Uri, Fredrik D Johansson, and David Sontag (2017). "Estimating individual treatment effect: generalization bounds and algorithms". In: *Proceedings of the 34th International Conference on Machine Learning*. PMLR, pp. 3076–3085.

Shpitser, Ilya and Judea Pearl (2007). "What counterfactuals can be tested". In: *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*. AUAI Press, pp. 352–359.

Silva, Ricardo (2016). "Observational-interventional priors for dose-response learning". In: *Advances in Neural Information Processing Systems*, pp. 1569–1577. ISSN: 10495258.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2014). "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations*.

Simpson, Edward H (1951). "The interpretation of interaction in contingency tables". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 13.2, pp. 238–241.

Simpson, Edwin et al. (2013). "Dynamic Bayesian combination of multiple imperfect classifiers". In: *Decision Making and Imperfection*. Springer, pp. 1–35.

Smith, Aaron (2016). *Public Predictions for the Future of Workforce Automation: Full Report*. Tech. rep. Pew Research Center.

Solak, Ercan et al. (2003). "Derivative observations in Gaussian process models of dynamic systems". In: *Advances in Neural Information Processing Systems*, pp. 1057–1064.

Spirtes, Peter., Clark N. Glymour, and Richard. Scheines (2000). *Causation, prediction, and search.* MIT Press, p. 543. ISBN: 9780262194402.

Splawa-Neyman, Jerzy (1923). "On the Application of Probability Theory to Agricultural Experiments. Essay on Principles. Section 9." In: *Roczniki Nauk Rolniczych* 10, pp. 1–51. DOI: 10.2307/2245382.

Stokes, Jonathan M. et al. (2020). "A Deep Learning Approach to Antibiotic Discovery". In: *Cell* 180.4, 688–702.e13. ISSN: 10974172. DOI: 10.1016/j.cell.2020.01.021.

Subbaswamy, Adarsh and Suchi Saria (2018). "Counterfactual Normalization: Proactively Addressing Dataset Shift Using Causal Mechanisms". In: *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence*.

Titsias, Michalis K (2009). "Variational learning of inducing variables in sparse Gaussian processes". In: *Journal of Machine Learning Research*. Vol. 5, pp. 567–574.

Toro, Gabriel R. et al. (2010). "Approaches for the efficient probabilistic calculation of surge hazard". In: *Ocean Engineering*.

Tran, Dustin, Matthew D. Hoffman, et al. (2017). "Deep probabilistic programming". In: *International Conference on Learning Representations*.

Tran, Dustin, Francisco J. R. Ruiz, et al. (2016). "Model Criticism for Bayesian Causal Inference". In: arXiv: 1610.09037 [stat.ME].

Tresp, Volker (2000). "A Bayesian committee machine". In: *Neural Computation* 12.11, pp. 2719–2741. ISSN: 08997667. DOI: 10.1162/089976600300014908.

Urteaga, Iñigo et al. (2019). "Multi-Task Gaussian Processes and Dilated Convolutional Networks for Reconstruction of Reproductive Hormonal Dynamics". In: *Proceedings of the 4th Machine Learning for Healthcare Conference*.

Vellido, Alfredo, J D Martin-Guerroro, and P Lisboa (2012). "Making machine learning models interpretable". In: *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. April, pp. 163–172. ISBN: 9782874190490.

Viviano, Davide and Jelena Bradic (2019). *Synthetic learner: model-free inference on treatments over time*. Tech. rep. arXiv: 1904.01490v1.

Watson, Mark W. (1986). "Univariate detrending methods with stochastic trends". In: *Journal of Monetary Economics* 18.1, pp. 49–75. ISSN: 03043932. DOI: 10.1016/0304-3932(86)90054-1.

Wilson, Andrew Gordon and Ryan Prescott Adams (2013). "Gaussian process kernels for pattern discovery and extrapolation". In: *Proceedings of the 30th International Conference on Machine Learning*.

Yang, Xiu et al. (2019). "Physics-informed CoKriging: A Gaussian-process-regression-based multifidelity method for data-model convergence". In: *Journal of Computational Physics* 395, pp. 410–431. ISSN: 10902716. DOI: 10.1016/j.jcp.2019.06.041.

Zech, John R. et al. (2018). "Confounding variables can degrade generalization performance of radiological deep learning models". In: arXiv: 1807.00431.

Zhang, Junzhe and Elias Bareinboim (2017). "Transfer Learning in Multi-Armed Bandit : A Causal Approach". In: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, pp. 1778–1780. ISSN: 10450823.

Zymergen (2018). "The promise of molecular oncology".

Appendices

A PIGEBaQ

A.1 Posterior Integral Equations

Kernel

The exponentiated quadratic kernel is:

$$\kappa(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}')^T\Lambda^{-1}(\boldsymbol{x}-\boldsymbol{x}')}{2}\right) \tag{1}$$

The derivatives are

$$\frac{\partial}{\partial \boldsymbol{x}}\kappa(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2\Lambda^{-1}(\boldsymbol{x}-\boldsymbol{x}')\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}')^T\Lambda^{-1}(\boldsymbol{x}-\boldsymbol{x}')}{2}\right) \tag{2}$$

$$\frac{\partial^2}{\partial \boldsymbol{x}\partial \boldsymbol{x}'}\kappa(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2\Lambda^{-1}(\mathbb{I} - (\boldsymbol{x}-\boldsymbol{x})'(\boldsymbol{x}-\boldsymbol{x}')^T\Lambda^{-1})\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')^T\Lambda^{-1}(\boldsymbol{x}-\boldsymbol{x}')\right) \tag{3}$$

Posterior Mean of the Integral

The expected mean of the integral is:

$$\mathbb{E}[\boldsymbol{I}|f(\boldsymbol{x}_s)] = \sum_{i=1}^{N} \boldsymbol{z}^T K^{-1} f(\boldsymbol{x}_i), \tag{4}$$

where:

$$\boldsymbol{z}_n = \sum_{m=1}^{M} \pi_m \int K_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}_n) p(\boldsymbol{x}|\theta_m) \mathrm{d}\boldsymbol{x} \tag{5}$$

Let us consider a single component:

$$\boldsymbol{z}_{nm} = \int K_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{x}_n) p(\boldsymbol{x}|\theta_m) \mathrm{d}\boldsymbol{x} \tag{6}$$

$$= -\sigma^2\Lambda^{-1}\int (\boldsymbol{x}-\boldsymbol{x}'_n)\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{x}_n)^T\Lambda^{-1}(\boldsymbol{x}-\boldsymbol{x}_n)}{2}\right)\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \Sigma_m)\mathrm{d}\boldsymbol{x} \tag{7}$$

$$= -\sigma^2\Lambda^{-1}(2\pi)^{D/2}|\Lambda|^{1/2}\int (\boldsymbol{x}-\boldsymbol{x}_n)\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_n, \Lambda)\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \Sigma_m)\mathrm{d}\boldsymbol{x} \tag{8}$$

Using standard properties of products and integrals of multivariate Gaussians we arrive at:

$$\boldsymbol{z}_{nm} = -\frac{\sigma^2|\Lambda|^{1/2}}{|\Lambda+\Sigma_m|^{\frac{1}{2}}}(\Lambda+\Sigma_m)^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu}_m)\exp\left(-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu}_m)^T(\Lambda+\Sigma_m)^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu}_m)\right) \tag{9}$$

and consequently:

$$\boldsymbol{z}_n = \sum_{m=1}^{M}\frac{\pi_m\sigma^2|\Lambda|^{1/2}}{|\Lambda+\Sigma_m|^{\frac{1}{2}}}(\Lambda+\Sigma_m)^{-1}(\boldsymbol{\mu}_m-\boldsymbol{x}_n-)\exp\left(-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu}_m)^T(\Lambda+\Sigma_m)^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu}_m)\right). \tag{10}$$

Posterior Variance of the Integral

The posterior variance of the integral is given by:

$$\mathrm{Var}[\boldsymbol{I}|f(\boldsymbol{x}_s)] = \int\int K_{\boldsymbol{x},\boldsymbol{x}'}(\boldsymbol{x},\boldsymbol{x}')p(\boldsymbol{x})p(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' - \boldsymbol{z}^T K^{-1}\boldsymbol{z}, \tag{11}$$

with $K_{\boldsymbol{x},\boldsymbol{x}'}(\boldsymbol{x},\boldsymbol{x}')$ the second derivative of the kernel function. The second can term can be computed using z, the first expression is given by:

$$\left[\int\int K_{\boldsymbol{x},\boldsymbol{x}'}(\boldsymbol{x},\boldsymbol{x}')p(\boldsymbol{x})p(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}'\right]_m \tag{12}$$

$$= \sigma^2\Lambda^{-1}(2\pi)^{D/2}|\Lambda|^{1/2}\int\int(I-(\boldsymbol{x}-\boldsymbol{x}')(\boldsymbol{x}-\boldsymbol{x}')^T)\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}',\Lambda)\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m,\Sigma_m)\mathcal{N}(\boldsymbol{x}'|\boldsymbol{\mu}_m,\Sigma_m)\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \tag{13}$$

Exploiting the properties of the Gaussian distribution again and after some calculation we arrive at:

$$\int\int K_{\boldsymbol{x},\boldsymbol{x}'}(\boldsymbol{x},\boldsymbol{x}')p(\boldsymbol{x})p(\boldsymbol{x}')\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' \tag{14}$$

$$= \sum_{m=1}^{M} \pi_m \frac{\sigma^2|\Lambda|^{\frac{1}{2}}}{|\Lambda+2\Sigma_m|^{\frac{1}{2}}}(\Lambda^{-1}+(\Lambda+\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}+(\Lambda+2\Sigma_m)^{-1}\Sigma_m\Lambda^{-1} \tag{15}$$

$$+\Lambda^{-1}\Sigma_m(\Lambda+2\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}+2\Lambda^{-1}\boldsymbol{\mu}_m\boldsymbol{\mu}_m^T\Lambda^{-1} \tag{16}$$

$$+(\Lambda+\Sigma_m)^{-1}\Sigma_m(\Lambda+2\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}\Sigma_m(\Lambda+\Sigma)^{-1} \tag{17}$$

$$+(\Lambda+\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}\Sigma_m(\Lambda+2\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}\Sigma_m(\Lambda+\Sigma_m)^{-1} \tag{18}$$

$$-(\Lambda_m+\Sigma_m)^{-1}\boldsymbol{\mu}_m\boldsymbol{\mu}_m^T\Lambda^{-1} \tag{19}$$

$$-(\Lambda+\Sigma_m)^{-1}\Lambda\Lambda^{-1}\Sigma_m(\Lambda+2\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}\Lambda \tag{20}$$

$$-(\Lambda+\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}\Sigma_m(\Lambda+2\Sigma_m)^{-1}\Sigma_m\Lambda^{-1} \tag{21}$$

$$-(\Lambda+\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}\boldsymbol{\mu}_m\boldsymbol{\mu}_m^T\Lambda^{-1} \tag{22}$$

$$-\Lambda^{-1}\boldsymbol{\mu}_m\boldsymbol{\mu}_m^T(\Lambda+\Sigma_m)^{-1} \tag{23}$$

$$-\Sigma_m(\Lambda+2\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}\Lambda(\Lambda+\Sigma_m)^{-1}\Lambda^{-1}\Lambda \tag{24}$$

$$-\Lambda^{-1}\Sigma_m(\Lambda+2\Sigma_m)^{-1}\Sigma_m\Lambda^{-1}\Sigma_m(\Lambda+\Sigma_m)^{-1} \tag{25}$$

$$-\Lambda^{-1}\boldsymbol{\mu}_m\boldsymbol{\mu}_m^T\Lambda^{-1}\Sigma_m(\Lambda+\Sigma_m)^{-1}) \tag{26}$$

A.2 Volatility

Let $\boldsymbol{x}^*$ be a prospective point with feature vector of dimensions $1\times d$ and $\boldsymbol{X}$ a matrix of dimension $n\times d$, where n is the number of observed points and d is the dimensionality of the feature vector. $\boldsymbol{y_X}$ is the target vector and has dimensions $n\times 1$.

Imagine taking our attention weighted derivative GP and marginalising out $\boldsymbol{x}^*$, this would give us a non-Gaussian distribution:

$$p(\boldsymbol{y'}) = \int_{\mathcal{X}} d\boldsymbol{x}^*\; p(\frac{\partial y}{\partial \boldsymbol{x}^*} \mid \boldsymbol{X}, y, \boldsymbol{x}^*)p\left(\boldsymbol{x}^*\right) = \int_{\mathcal{X}} d\boldsymbol{x}^*\; p(\boldsymbol{y'} \mid \boldsymbol{X}, y, \boldsymbol{x}^*)p\left(\boldsymbol{x}^*\right) \tag{27}$$

Where:

$$p(\boldsymbol{y'} \mid \boldsymbol{X}, y, \boldsymbol{x}^*) = \mathcal{N}(\boldsymbol{y'} \mid \boldsymbol{\mu_x}, \boldsymbol{\Sigma_x}) \tag{28}$$

$$\boldsymbol{\mu_x} = \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)}\boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})}\boldsymbol{y_X} \tag{29}$$

$$\boldsymbol{\Sigma_x} = \boldsymbol{K}_{\boldsymbol{y'}(\boldsymbol{x}^*),\boldsymbol{y'}(\boldsymbol{x}^*)} - \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)}\boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)} \tag{30}$$

Here $p\left(\boldsymbol{x}^*\right)$ is taken to be a Gaussian Mixture Model:

$$p\left(\boldsymbol{x}^*\right) = \sum_p \pi_p\, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_p}, \boldsymbol{\Sigma_p}) \tag{31}$$

where π_p is the mixture coefficients and $\sum_p \pi_p = 1$, $\boldsymbol{\mu_p}$ and $\boldsymbol{\Sigma_p}$ are mixture means and covariances.

Defining the kernel notation:

$$\boldsymbol{K}_{A(\mathcal{C}),B(\mathcal{D})} = cov(A_{\mathcal{C}}, B_{\mathcal{D}}), \tag{32}$$

this reads: the covariance between functions A and B with input points to A being $\mathcal{C}$ and input points to B being $\mathcal{D}$.

A.3 Approximation of marginalised distribution

Approximating this distribution with a Gaussian distribution $q(\boldsymbol{y'}) = \mathcal{N}(\boldsymbol{y'} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ we can minimise the $\min KL = \min KL\left(p(\boldsymbol{y'}) \mid\mid q(\boldsymbol{y'})\right)$, where:

$$\boldsymbol{\mu} = \mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}] \tag{33}$$
$$\boldsymbol{\Sigma} = \mathbb{V}ar_{p(\boldsymbol{y'})}[\boldsymbol{y'}] = \mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}\boldsymbol{y'}^T] - \mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}]\mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}]^T \tag{34}$$

Mean term μ

Expanding and rearranging the mean $\boldsymbol{\mu}$:

$$\boldsymbol{\mu} = \int_{\mathcal{Y}'} d\boldsymbol{y'}\ \boldsymbol{y'} \int_{\mathcal{X}} d\boldsymbol{x}^*\ p(\boldsymbol{y'} \mid \boldsymbol{X}, y, \boldsymbol{x}^*) p\left(\boldsymbol{x}^*\right) \tag{35}$$
$$= \int_{\mathcal{X}} d\boldsymbol{x}^*\ p\left(\boldsymbol{x}^*\right) \int_{\mathcal{Y}'} d\boldsymbol{y'}\ \boldsymbol{y'}\ p(\boldsymbol{y'} \mid \boldsymbol{X}, y, \boldsymbol{x}^*) \tag{36}$$
$$= \mathbb{E}_{p(\boldsymbol{x}^*)}\left[\mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}]\right] \tag{37}$$

Substituting equation 29 into the above:

$$\boldsymbol{\mu} = \mathbb{E}_{p(\boldsymbol{x}^*)}\left[\boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)} \boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})} \boldsymbol{y}_{\boldsymbol{X}}\right] \tag{38}$$
$$= \int_{\mathcal{X}} d\boldsymbol{x}^*\ p\left(\boldsymbol{x}^*\right) \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)} \boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})} \boldsymbol{y}_{\boldsymbol{X}} \tag{39}$$
$$= \left\{\int_{\mathcal{X}} d\boldsymbol{x}^*\ p\left(\boldsymbol{x}^*\right) \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)}\right\} \boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})} \boldsymbol{y}_{\boldsymbol{X}} \tag{40}$$

Taking the bracketed section:

$$\int_{\mathcal{X}} d\boldsymbol{x}^* \; p\left(\boldsymbol{x}^*\right) \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y}'(\boldsymbol{x}^*)} = \int_{\mathcal{X}} d\boldsymbol{x}^* \; p\left(\boldsymbol{x}^*\right) \begin{bmatrix} \frac{\partial k(\boldsymbol{x}_1,\boldsymbol{x}_a)}{\partial \boldsymbol{x}_a} \\ \vdots \\ \frac{\partial k(\boldsymbol{x}_n,\boldsymbol{x}_a)}{\partial \boldsymbol{x}_a} \end{bmatrix}_{\boldsymbol{x}_a=\boldsymbol{x}^*} \tag{41}$$

$$= \begin{bmatrix} \int_{\mathcal{X}} d\boldsymbol{x}^* \; p\left(\boldsymbol{x}^*\right) \left.\frac{\partial k(\boldsymbol{x}_1,\boldsymbol{x}_a)}{\partial \boldsymbol{x}_a}\right|_{\boldsymbol{x}_a=\boldsymbol{x}^*} \\ \vdots \\ \int_{\mathcal{X}} d\boldsymbol{x}^* \; p\left(\boldsymbol{x}^*\right) \left.\frac{\partial k(\boldsymbol{x}_n,\boldsymbol{x}_a)}{\partial \boldsymbol{x}_a}\right|_{\boldsymbol{x}_a=\boldsymbol{x}^*} \end{bmatrix} \tag{42}$$

Taking one element of this equation and substituting the RBF kernel:

$$\int_{\mathcal{X}} d\boldsymbol{x}^* \; p\left(\boldsymbol{x}^*\right) \left.\frac{\partial k(\boldsymbol{x}_l,\boldsymbol{x}_a)}{\partial \boldsymbol{x}_a}\right|_{\boldsymbol{x}_a=\boldsymbol{x}^*} \tag{43}$$

$$= \int_{\mathcal{X}} d\boldsymbol{x}^* \; \sum_p \pi_p \, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_p}, \boldsymbol{\Sigma_p}) k(\boldsymbol{x}_l, \boldsymbol{x}^*) \; (\boldsymbol{x}_l - \boldsymbol{x}^*)\boldsymbol{\Lambda}^{-1} \tag{44}$$

$$= \sum_p \pi_p \alpha_{\boldsymbol{\Lambda}} \int_{\mathcal{X}} d\boldsymbol{x}^* \, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_p}, \boldsymbol{\Sigma_p}) \, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{x}_l, \boldsymbol{\Lambda}) \; (\boldsymbol{x}_l - \boldsymbol{x}^*)\boldsymbol{\Lambda}^{-1} \tag{45}$$

Combining the two multivariate Gaussian distributions:

$$\mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_p}, \boldsymbol{\Sigma_p}) \, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{x}_l, \boldsymbol{\Lambda}) = k(l,p) \, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu}_{m1}, \boldsymbol{\Sigma}_{m1}) \tag{46}$$

where:

$$k(l,p) = \mathcal{N}(\boldsymbol{\mu_p} \mid \boldsymbol{x}_l, \boldsymbol{\Sigma_p} + \boldsymbol{\Lambda}) \tag{47}$$

$$\boldsymbol{\mu}_{m1} = (\boldsymbol{I} + \boldsymbol{\Sigma}_p\boldsymbol{\Lambda}^{-1})^{-1}\boldsymbol{\Sigma}_p(\boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1}\boldsymbol{\mu_p} + \boldsymbol{\Lambda}^{-1}\boldsymbol{x}_l) \tag{48}$$

$$= (\boldsymbol{I} + \boldsymbol{\Sigma}_p\boldsymbol{\Lambda}^{-1})^{-1}(\boldsymbol{\mu_p} + \boldsymbol{\Sigma}_p\boldsymbol{\Lambda}^{-1}\boldsymbol{x}_l) \tag{49}$$

$$\boldsymbol{\Sigma}_{m1} = (\boldsymbol{\Sigma}_{\boldsymbol{p}}^{-1} + \boldsymbol{\Lambda}^{-1})^{-1} \tag{50}$$

$$= (\boldsymbol{I} + \boldsymbol{\Sigma}_p\boldsymbol{\Lambda}^{-1})^{-1}\boldsymbol{\Sigma}_p \tag{51}$$

Substituting back in and calculating:

$$\int_{\mathcal{X}} d\boldsymbol{x}^* \; p\left(\boldsymbol{x}^*\right) \left.\frac{\partial k(\boldsymbol{x}_l,\boldsymbol{x}_a)}{\partial \boldsymbol{x}_a}\right|_{\boldsymbol{x}_a=\boldsymbol{x}^*} \tag{52}$$

$$= \sum_p \pi_p \alpha_{\boldsymbol{\Lambda}} k(l,p) \int_{\mathcal{X}} d\boldsymbol{x}^* \, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu}_{m1}, \boldsymbol{\Sigma}_{m1}) \; (\boldsymbol{x}_l - \boldsymbol{x}^*)\boldsymbol{\Lambda}^{-1} \tag{53}$$

$$= \sum_p \pi_p \alpha_{\boldsymbol{\Lambda}} k(l,p) \; (\boldsymbol{x}_l - \boldsymbol{\mu}_{m1})\boldsymbol{\Lambda}^{-1} \tag{54}$$

Variance term Σ

Expanding the variance $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}\boldsymbol{y'}^T] - \mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}]\mathbb{E}_{p(\boldsymbol{y'})}[\boldsymbol{y'}]^T \tag{55}$$

$$= \int_{\mathcal{Y}'} d\boldsymbol{y'}\ \boldsymbol{y'}\boldsymbol{y'}^T \int_{\mathcal{X}} d\boldsymbol{x}^*\ p(\boldsymbol{y'} \mid \boldsymbol{X}, y, \boldsymbol{x}^*) p(\boldsymbol{x}^*) - \boldsymbol{\mu}\boldsymbol{\mu}^T \tag{56}$$

$$= \int_{\mathcal{X}} d\boldsymbol{x}^*\ p(\boldsymbol{x}^*) \int_{\mathcal{Y}'} d\boldsymbol{y'}\ \boldsymbol{y'}\boldsymbol{y'}^T\ p(\boldsymbol{y'} \mid \boldsymbol{X}, y, \boldsymbol{x}^*) - \boldsymbol{\mu}\boldsymbol{\mu}^T \tag{57}$$

$$= \mathbb{E}_{p(\boldsymbol{x}^*)}\left[\mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}\boldsymbol{y'}^T]\right] - \boldsymbol{\mu}\boldsymbol{\mu}^T \tag{58}$$

$$= \mathbb{E}_{p(\boldsymbol{x}^*)}[\mathbb{V}ar_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*))}[\boldsymbol{y'}] \tag{59}$$

$$+ \mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}]\mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}]^T] \tag{60}$$

$$- \mathbb{E}_{p(\boldsymbol{x}^*)}\left[\mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}]\right]\mathbb{E}_{p(\boldsymbol{x}^*)}\left[\mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}]\right]^T \tag{61}$$

Expanding the first term of $\boldsymbol{\Sigma}$:

$$\mathbb{V}ar_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*))}[\boldsymbol{y'}] + \mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}]\mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}]^T \tag{62}$$

$$= \boldsymbol{K}_{\boldsymbol{y'}(\boldsymbol{x}^*),\boldsymbol{y'}(\boldsymbol{x}^*)} - \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)}\boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)} \tag{63}$$

$$+ \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)}\boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})}\boldsymbol{y}\boldsymbol{y}^{\boldsymbol{T}}\boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)} \tag{64}$$

$$= \boldsymbol{K}_{\boldsymbol{y'}(\boldsymbol{x}^*),\boldsymbol{y'}(\boldsymbol{x}^*)} \tag{65}$$

$$+ \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)}\left\{\boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})}\boldsymbol{y}_{\boldsymbol{X}}\boldsymbol{y}^T_{\boldsymbol{X}}\boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})} - \boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})}\right\}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)} \tag{66}$$

$$= \boldsymbol{K}_{\boldsymbol{y'}(\boldsymbol{x}^*),\boldsymbol{y'}(\boldsymbol{x}^*)} + \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)}\boldsymbol{A}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)} \tag{67}$$

where:

$$\boldsymbol{A} = \boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})}\boldsymbol{y}_{\boldsymbol{X}}\boldsymbol{y}^T_{\boldsymbol{X}}\boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})} - \boldsymbol{K}^{-1}_{\boldsymbol{y}(\boldsymbol{X}),y(\boldsymbol{X})} \tag{68}$$

$$\boldsymbol{\Sigma} = \mathbb{E}_{p(\boldsymbol{x}^*)}\left[\boldsymbol{K}_{\boldsymbol{y'}(\boldsymbol{x}^*),\boldsymbol{y'}(\boldsymbol{x}^*)} + \boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)}\boldsymbol{A}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y'}(\boldsymbol{x}^*)}\right] \tag{69}$$

$$- \mathbb{E}_{p(\boldsymbol{x}^*)}\left[\mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}]\right]\mathbb{E}_{p(\boldsymbol{x}^*)}\left[\mathbb{E}_{p(\boldsymbol{y'}|\boldsymbol{X},y,\boldsymbol{x}^*)}[\boldsymbol{y'}]\right]^T \tag{70}$$

The first term in equation 69 can be calculated straight away:

$$\mathbb{E}_{p(\boldsymbol{x}^*)}[\boldsymbol{K}_{\boldsymbol{y}'(\boldsymbol{x}^*),\boldsymbol{y}'(\boldsymbol{x}^*)}] = \mathbb{E}_{p(\boldsymbol{x}^*)}\left[\left.\frac{\partial^2 k(\boldsymbol{x}_a,\boldsymbol{x}_b)}{\partial \boldsymbol{x}_a \partial \boldsymbol{x}_b}\right|_{\boldsymbol{x}_a=\boldsymbol{x}_b=\boldsymbol{x}^*}\right] \tag{71}$$

$$= \mathbb{E}_{p(\boldsymbol{x}^*)}\left[\sigma_f \boldsymbol{\Lambda}^{-1}\right] \tag{72}$$

$$= \sigma_f \boldsymbol{\Lambda}^{-1} \tag{73}$$

The second term in equation 69 needs to be teased apart:

$$\left\{\mathbb{E}_{p(\boldsymbol{x}^*)}\left[\boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y}'(\boldsymbol{x}^*)}\boldsymbol{A}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y}'(\boldsymbol{x}^*)}\right]\right\}_{i,j} = \mathbb{E}_{p(\boldsymbol{x}^*)}\left[\boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y}'_i(\boldsymbol{x}^*)}\boldsymbol{A}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y}'_j(\boldsymbol{x}^*)}\right] \tag{74}$$

where $\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y}'_i(\boldsymbol{x}^*)}$ takes the derivative only on the ith feature and is therefore a column vector. Using the useful result:

$$\mathbb{E}[\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y}] = \mathrm{Tr}[\boldsymbol{A}\mathbb{E}[\boldsymbol{y}\boldsymbol{x}^T]] \tag{75}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are stochastic vectors with dimensions $n \times 1$ and $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, the element-by-element result of equation 74 can be calculated.

Equation 74 becomes:

$$\mathbb{E}_{p(\boldsymbol{x}^*)}\left[\boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y}'_i(\boldsymbol{x}^*)}\boldsymbol{A}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y}'_j(\boldsymbol{x}^*)}\right] \tag{76}$$

$$= \mathrm{Tr}\left[\boldsymbol{A}\mathbb{E}_{p(\boldsymbol{x}^*)}[\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y}'_j(\boldsymbol{x}^*)}\boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y}'_i(\boldsymbol{x}^*)}]\right] \tag{77}$$

Taking out the expectation term:

$$\mathbb{E}_{p(\boldsymbol{x}^*)}\left[\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y}'_j(\boldsymbol{x}^*)}\boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y}'_i(\boldsymbol{x}^*)}\right] \tag{78}$$

$$= \int d\boldsymbol{x}^* p(\boldsymbol{x}^*) \left[\begin{matrix} \frac{\partial k(\boldsymbol{x}_1,\boldsymbol{x}_a)}{\partial x_{a,j}} \\ \vdots \\ \frac{\partial k(\boldsymbol{x}_n,\boldsymbol{x}_a)}{\partial x_{a,j}} \end{matrix}\right]_{\boldsymbol{x}_a=\boldsymbol{x}^*} \left[\begin{matrix} \frac{\partial k(\boldsymbol{x}_1,\boldsymbol{x}_a)}{\partial x_{a,i}} \\ \vdots \\ \frac{\partial k(\boldsymbol{x}_n,\boldsymbol{x}_a)}{\partial x_{a,i}} \end{matrix}\right]^T_{\boldsymbol{x}_a=\boldsymbol{x}^*} \tag{79}$$

$$= \int d\boldsymbol{x}^* p(\boldsymbol{x}^*) \left[\begin{matrix} \frac{\partial k(\boldsymbol{x}_1,\boldsymbol{x}_a)}{\partial x_{a,j}}\frac{\partial k(\boldsymbol{x}_1,\boldsymbol{x}_a)}{\partial x_{a,i}} & \cdots & \frac{\partial k(\boldsymbol{x}_1,\boldsymbol{x}_a)}{\partial x_{a,j}}\frac{\partial k(\boldsymbol{x}_n,\boldsymbol{x}_a)}{\partial x_{a,i}} \\ \vdots & \ddots & \vdots \\ \frac{\partial k(\boldsymbol{x}_n,\boldsymbol{x}_a)}{\partial x_{a,j}}\frac{\partial k(\boldsymbol{x}_1,\boldsymbol{x}_a)}{\partial x_{a,i}} & \cdots & \frac{\partial k(\boldsymbol{x}_n,\boldsymbol{x}_a)}{\partial x_{a,j}}\frac{\partial k(\boldsymbol{x}_n,\boldsymbol{x}_a)}{\partial x_{a,i}} \end{matrix}\right]_{\boldsymbol{x}_a=\boldsymbol{x}^*} \tag{80}$$

Looking at one element of this $n \times n$ matrix:

$$\int d\boldsymbol{x}^* \left.\frac{\partial k(\boldsymbol{x}_l, \boldsymbol{x}_a)}{\partial x_{a,j}}\right|_{\boldsymbol{x}_a=\boldsymbol{x}^*} \left.\frac{\partial k(\boldsymbol{x}_m, \boldsymbol{x}_a)}{\partial x_{a,i}}\right|_{\boldsymbol{x}_a=\boldsymbol{x}^*} p\left(\boldsymbol{x}^*\right) \tag{81}$$

$$= \int d\boldsymbol{x}^* k(\boldsymbol{x}_l, \boldsymbol{x}^*)\ k(\boldsymbol{x}_m, \boldsymbol{x}^*)(\boldsymbol{x}_l - \boldsymbol{x}^*)\boldsymbol{\Lambda}_j^{-1}(\boldsymbol{x}_m - \boldsymbol{x}^*)\boldsymbol{\Lambda}_i^{-1} p\left(\boldsymbol{x}^*\right) \tag{82}$$

$$= \int d\boldsymbol{x}^* k(\boldsymbol{x}_l, \boldsymbol{x}^*)\ k(\boldsymbol{x}_m, \boldsymbol{x}^*)(\boldsymbol{\Lambda}_j^{-1})^T(\boldsymbol{x}_l - \boldsymbol{x}^*)^T(\boldsymbol{x}_m - \boldsymbol{x}^*)\boldsymbol{\Lambda}_i^{-1} p\left(\boldsymbol{x}^*\right) \tag{83}$$

Since $k(\boldsymbol{x}_l, \boldsymbol{x}^*)$, $k(\boldsymbol{x}_m, \boldsymbol{x}^*)$ and $p\left(\boldsymbol{x}^*\right)$ are all (scaled) normal distributions dependent on $\boldsymbol{x}^*$, they can be combined into one normal distribution dependent on $\boldsymbol{x}^*$ and two other normal distribution not dependent on $\boldsymbol{x}^*$.

Substituting the forms for all three distributions:

$$k(\boldsymbol{x}_l, \boldsymbol{x}^*)\ k(\boldsymbol{x}_m, \boldsymbol{x}^*)\ p\left(\boldsymbol{x}^*\right) \tag{84}$$

$$= k(\boldsymbol{x}^*, \boldsymbol{x}_l)\ k(\boldsymbol{x}^*, \boldsymbol{x}_m)\ p\left(\boldsymbol{x}^*\right) \tag{85}$$

$$= \alpha_{\boldsymbol{\Lambda}}\, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{x}_l, \boldsymbol{\Lambda}) \alpha_{\boldsymbol{\Lambda}}\, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{x}_m, \boldsymbol{\Lambda}) \sum_p \pi_p\, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_p}, \boldsymbol{\Sigma_p}) \tag{86}$$

$$= \sum_p \pi_p\, \alpha_{\boldsymbol{\Lambda}}^2\, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{x}_l, \boldsymbol{\Lambda})\, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{x}_m, \boldsymbol{\Lambda})\, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_p}, \boldsymbol{\Sigma_p}) \tag{87}$$

$$= \sum_p \pi_p\, \alpha_{\boldsymbol{\Lambda}}^2 c_1(l, m)\, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_{c_1}}, \boldsymbol{\Sigma_{c_1}})\, \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_p}, \boldsymbol{\Sigma_p}) \tag{88}$$

$$= \sum_p \pi_p\, \alpha_{\boldsymbol{\Lambda}}^2 c_1(l, m) c_2(l, m, p) \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_{c_2}}, \boldsymbol{\Sigma_{c_2}}) \tag{89}$$

where c_1 terms are:

$$c_1(l, m) = \mathcal{N}(\boldsymbol{x}_l \mid \boldsymbol{x}_m, 2\boldsymbol{\Lambda}) \tag{90}$$

$$\boldsymbol{\mu_{c_1}} = \tfrac{1}{2}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}^{-1}\boldsymbol{x}_l + \boldsymbol{\Lambda}^{-1}\boldsymbol{x}_m) \tag{91}$$

$$= \tfrac{1}{2}(\boldsymbol{x}_l + \boldsymbol{x}_m) \tag{92}$$

$$\boldsymbol{\Sigma_{c_1}} = (\boldsymbol{\Lambda}^{-1} + \boldsymbol{\Lambda}^{-1})^{-1} \tag{93}$$

$$= \tfrac{1}{2}\boldsymbol{\Lambda} \tag{94}$$

where c_2 terms are:

$$c_2(n,m,p) = \mathcal{N}\left(\tfrac{1}{2}(\boldsymbol{x}_n + \boldsymbol{x}_m) \mid \boldsymbol{\mu_p}, \tfrac{1}{2}\boldsymbol{\Lambda} + \boldsymbol{\Sigma_p}\right) \tag{95}$$

$$\boldsymbol{\mu_{c_2}} = \boldsymbol{\Lambda}(2\boldsymbol{\Sigma_p} + \boldsymbol{\Lambda})^{-1}\boldsymbol{\Sigma_p}\left((\tfrac{1}{2}\boldsymbol{\Lambda})^{-1}\tfrac{1}{2}(\boldsymbol{x}_n + \boldsymbol{x}_m) + \boldsymbol{\Sigma_p}^{-1}\boldsymbol{\mu_p}\right) \tag{96}$$

$$= \boldsymbol{\Lambda}(2\boldsymbol{\Sigma_p} + \boldsymbol{\Lambda})^{-1}\left(\boldsymbol{\Sigma_p}\boldsymbol{\Lambda}^{-1}(\boldsymbol{x}_n + \boldsymbol{x}_m) + \boldsymbol{\mu_p}\right) \tag{97}$$

$$\boldsymbol{\Sigma_{c_2}} = ((\tfrac{1}{2}\boldsymbol{\Lambda})^{-1} + \boldsymbol{\Sigma_p}^{-1})^{-1} \tag{98}$$

$$= (2\boldsymbol{\Sigma_p}\boldsymbol{\Lambda}^{-1} + \boldsymbol{I})^{-1}\boldsymbol{\Sigma_p} \tag{99}$$

$$= \boldsymbol{\Lambda}(2\boldsymbol{\Sigma_p} + \boldsymbol{\Lambda})^{-1}\boldsymbol{\Sigma_p} \tag{100}$$

Substituting equation 89 into 83:

$$\int d\boldsymbol{x}^* \left.\frac{\partial k(\boldsymbol{x}_l, \boldsymbol{x}_a)}{\partial x_{a,j}}\right|_{\boldsymbol{x}_a = \boldsymbol{x}^*} \left.\frac{\partial k(\boldsymbol{x}_m, \boldsymbol{x}_a)}{\partial x_{a,i}}\right|_{\boldsymbol{x}_a = \boldsymbol{x}^*} p\left(\boldsymbol{x}^*\right) \tag{101}$$

$$= \int d\boldsymbol{x}^* \sum_p \pi_p\, \alpha_{\boldsymbol{\Lambda}}^2 c_1(l,m)c_2(l,m,p) \tag{102}$$

$$\mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_{c_2}}, \boldsymbol{\Sigma_{c_2}})(\boldsymbol{\Lambda}_j^{-1})^T(\boldsymbol{x}_l - \boldsymbol{x}^*)^T(\boldsymbol{x}_m - \boldsymbol{x}^*)\boldsymbol{\Lambda}_i^{-1}$$

$$= \sum_p \pi_p\, \alpha_{\boldsymbol{\Lambda}}^2 c_1(l,m)c_2(l,m,p) \tag{103}$$

$$(\boldsymbol{\Lambda}_j^{-1})^T\left[\int d\boldsymbol{x}^* \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_{c_2}}, \boldsymbol{\Sigma_{c_2}})(\boldsymbol{x}_l - \boldsymbol{x}^*)^T(\boldsymbol{x}_m - \boldsymbol{x}^*)\right]\boldsymbol{\Lambda}_i^{-1}$$

$$= \sum_p \pi_p\, \alpha_{\boldsymbol{\Lambda}}^2 c_1(l,m)c_2(l,m,p) \tag{104}$$

$$(\boldsymbol{\Lambda}_j^{-1})^T\left[\int d\boldsymbol{x}^* \mathcal{N}(\boldsymbol{x}^* \mid \boldsymbol{\mu_{c_2}}, \boldsymbol{\Sigma_{c_2}})(\boldsymbol{x}_l^T\boldsymbol{x}_m - \boldsymbol{x}_l^T\boldsymbol{x}^* - \boldsymbol{x}^{*T}\boldsymbol{x}_m + \boldsymbol{x}^{*T}\boldsymbol{x}^*)\right]\boldsymbol{\Lambda}_i^{-1}$$

Calculating the integral within the square brackets:

$$\int d\boldsymbol{x}^* \left.\frac{\partial k(\boldsymbol{x}_l, \boldsymbol{x}_a)}{\partial x_{a,j}}\right|_{\boldsymbol{x}_a = \boldsymbol{x}^*} \left.\frac{\partial k(\boldsymbol{x}_m, \boldsymbol{x}_a)}{\partial x_{a,i}}\right|_{\boldsymbol{x}_a = \boldsymbol{x}^*} p\left(\boldsymbol{x}^*\right) \tag{105}$$

$$= \sum_p \pi_p\, \alpha_{\boldsymbol{\Lambda}}^2 c_1(l,m)c_2(l,m,p) \tag{106}$$

$$(\boldsymbol{\Lambda}_j^{-1})^T\left[\boldsymbol{x}_l^T\boldsymbol{x}_m - \boldsymbol{x}_l^T\boldsymbol{\mu_{c_2}} - \boldsymbol{\mu_{c_2}}^T\boldsymbol{x}_m + \boldsymbol{\Sigma_{c_2}} + \boldsymbol{\mu_{c_2}}^T\boldsymbol{\mu_{c_2}}\right]\boldsymbol{\Lambda}_i^{-1}$$

Since the terms $\boldsymbol{x}_l$, $\boldsymbol{x}_m$, $\boldsymbol{\mu_{c_2}}$, $\boldsymbol{\Sigma_{c_2}}$, $\pi_p\, \alpha_{\boldsymbol{\Lambda}}^2 c_1(l,m)c_2(l,m,p)$ are all independent of i,j they can be calculated independently to produce a 4-D tensor for all l,m with dimensions $n \times n \times d \times d$. The terms $(\boldsymbol{\Lambda}_i^{-1})^T$ and $\boldsymbol{\Lambda}_j^{-1}$ for all i,j can be broadcast multiplied (along the i,j dimensions) with this 4-D tensor to produce another 4-D tensor with similar dimensions.

Recalling equation 77:

$$\begin{aligned}&\mathbb{E}_{p(\boldsymbol{x}^*)}\left[\boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y}'_i(\boldsymbol{x}^*)}\boldsymbol{A}\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y}'_j(\boldsymbol{x}^*)}\right]\\&\qquad=\mathrm{Tr}\left[\boldsymbol{A}\mathbb{E}_{p(\boldsymbol{x}^*)}[\boldsymbol{K}_{y(\boldsymbol{X}),\boldsymbol{y}'_j(\boldsymbol{x}^*)}\boldsymbol{K}^T_{y(\boldsymbol{X}),\boldsymbol{y}'_i(\boldsymbol{x}^*)}]\right]\end{aligned}$$

The 2-D matrix $\boldsymbol{A}$ can be broadcast multiplied with the 4-D tensor created using equation 106 along the l, m dimensions. Taking the trace over the l, m dimensions of the resultant 4-D tensor reduces to a 2-D tensor with dimensions $d \times d$.

B Automation Survey

B.1 Expert Survey Description

Table 1: 70 occupations with tasks labeled to construct the training set.

O*NET-SOC Code	Title
11-1011.00	Chief Executives
11-3071.01	Transportation Managers
11-9033.00	Education Administrators, Postsecondary
11-9199.01	Regulatory Affairs Managers
13-1022.00	Wholesale And Retail Buyers, Except Farm Products
13-1075.00	Labor Relations Specialists
13-2053.00	Insurance Underwriters
15-1134.00	Web Developers
15-1143.01	Telecommunications Engineering Specialists
17-1011.00	Architects, Except Landscape And Naval
17-3022.00	Civil Engineering Technicians
21-1011.00	Substance Abuse And Behavioral Disorder Counselors
21-1023.00	Mental Health And Substance Abuse Social Workers
21-1093.00	Social And Human Service Assistants
23-1011.00	Lawyers
25-1011.00	Business Teachers, Postsecondary
25-1071.00	Health Specialties Teachers, Postsecondary
25-1194.00	Vocational Education Teachers, Postsecondary
25-2032.00	Career/Technical Education Teachers, Secondary School
25-2053.00	Special Education Teachers, Middle School
25-9041.00	Teacher Assistants
27-1011.00	Art Directors
27-1026.00	Merchandise Displayers And Window Trimmers
27-2011.00	Actors
27-2022.00	Coaches And Scouts
27-2042.01	Singers
29-1063.00	Internists, General
29-1199.01	Acupuncturists
29-2032.00	Diagnostic Medical Sonographers
29-2052.00	Pharmacy Technicians
29-9011.00	Occupational Health And Safety Specialists
31-9091.00	Dental Assistants
33-1021.01	Municipal Fire Fighting And Prevention Supervisors

33-3012.00	Correctional Officers And Jailers
33-9091.00	Crossing Guards
35-1011.00	Chefs And Head Cooks
35-2012.00	Cooks, Institution And Cafeteria
35-3011.00	Bartenders
35-9011.00	Dining Room And Cafeteria Attendants And Bartender Helpers
35-9021.00	Dishwashers
39-9011.00	Childcare Workers
41-2022.00	Parts Salespersons
41-4012.00	Sales Representatives, Wholesale And Manufacturing, Except Technical And Scientific Products
41-9021.00	Real Estate Brokers
43-3021.01	Statement Clerks
43-4121.00	Library Assistants, Clerical
43-4141.00	New Accounts Clerks
43-4181.00	Reservation And Transportation Ticket Agents And Travel Clerks
43-5021.00	Couriers And Messengers
45-2093.00	Farmworkers, Farm, Ranch, And Aquacultural Animals
47-1011.00	First-Line Supervisors Of Construction Trades And Extraction Workers
47-2021.00	Brickmasons And Blockmasons
47-2051.00	Cement Masons And Concrete Finishers
47-2181.00	Roofers
49-2022.00	Telecommunications Equipment Installers And Repairers, Except Line Installers
49-3021.00	Automotive Body And Related Repairers
49-9052.00	Telecommunications Line Installers And Repairers
51-1011.00	First-Line Supervisors Of Production And Operating Workers
51-2022.00	Electrical And Electronic Equipment Assemblers
51-4021.00	Extruding And Drawing Machine Setters, Operators, And Tenders, Metal And Plastic
51-4072.00	Molding, Coremaking, And Casting Machine Setters, Operators, And Tenders, Metal And Plastic
51-4121.06	Welders, Cutters, And Welder Fitters
51-6021.00	Pressers, Textile, Garment, And Related Materials
51-6031.00	Sewing Machine Operators
51-9111.00	Packaging And Filling Machine Operators And Tenders
51-9198.00	Helpers–Production Workers
53-1021.00	First-Line Supervisors Of Helpers, Laborers, And Material Movers, Hand
53-1031.00	First-Line Supervisors Of Transportation And Material-Moving Machine And Vehicle Operators

53-3022.00	Bus Drivers, School Or Special Client
53-7062.00	Laborers And Freight, Stock, And Material Movers, Hand

B.2 Expert Survey Responses

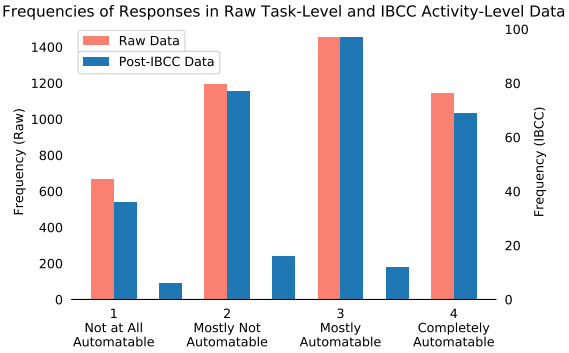


Fig. 1: Distribution of expert task-level responses, and the IBCC combined activity labels.

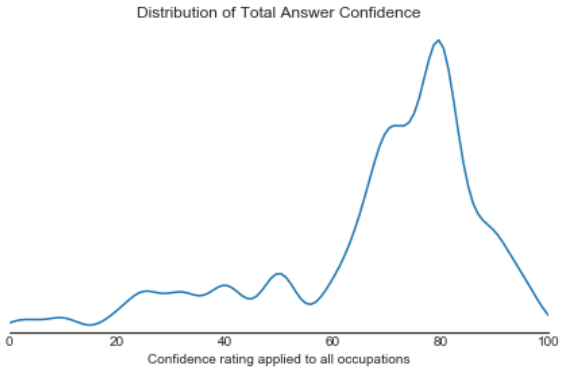


Fig. 2: The distribution of respondents confidences they assigned to their answers (in total). ($\mu = 67.9$, $\sigma = 20.7$)

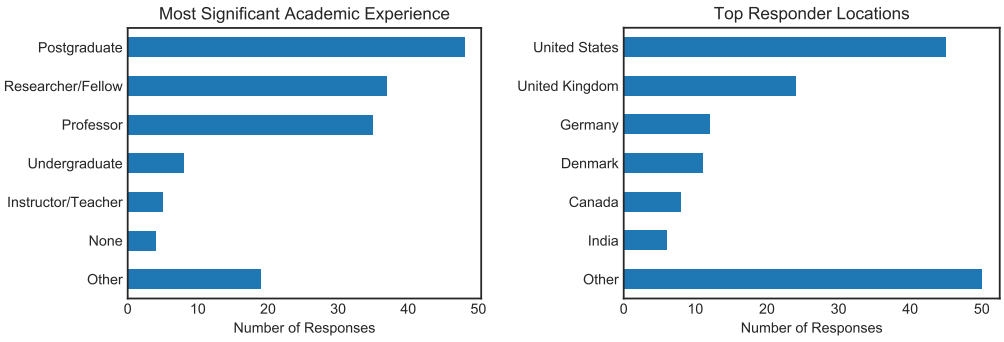


Fig. 3: Expert survey response statistics. Responses by: (left:) academic experience. (right:) geographic location.

B.3 Inferred Work Activity Automatability

Table 2: The 25 most and least automatable work activities.

Activity	Automatability Score (std)
Examine physical characteristics of gemstones or precious metals.	4.00 (0.86)
Adjust fabrics or other materials during garment production.	4.00 (0.67)
Sew materials.	4.00 (0.91)
Assemble garments or textile products.	4.00 (0.68)
Sew clothing or other articles.	4.00 (0.72)
Repair textiles or apparel.	4.00 (0.71)
Attach decorative or functional accessories to products.	3.96 (0.62)
Operate sewing equipment.	3.95 (0.68)
Design templates or patterns.	3.95 (0.68)
Prepare fabrics or materials for processing or production.	3.95 (0.67)
Evaluate log quality.	3.93 (0.71)
Cut fabrics.	3.93 (0.64)
Estimate costs of products, services, or materials.	3.92 (0.67)
Store records or related materials.	3.92 (0.66)
Position patterns on equipment, materials, or workpieces.	3.87 (0.62)
Shape metal workpieces with hammers or other small hand tools.	3.85 (0.65)
Measure physical characteristics of forestry or agricultural products.	3.84 (0.65)
Maneuver workpieces in equipment during production.	3.83 (0.62)
Operate office equipment.	3.83 (0.62)
Route mail to correct destinations.	3.82 (0.64)
Select production input materials.	3.81 (0.61)
Polish materials, workpieces, or finished products.	3.81 (0.64)
Design jewelry or decorative objects.	3.80 (0.78)
Record shipping information.	3.80 (0.63)
Confer with customers or designers to determine order specifications.	3.80 (0.63)
Teach humanities courses at the college level.	1.06 (0.65)
Teach online courses.	1.07 (0.63)
Teach social science courses at the college level.	1.15 (0.61)
Coordinate training activities.	1.16 (0.69)
Conduct scientific research of organizational behavior or processes.	1.19 (0.70)
Choreograph dances.	1.19 (0.86)
Entertain public with comedic or dramatic performances.	1.21 (0.68)
Design video game features or details.	1.22 (0.79)
Advise others on educational matters.	1.32 (0.70)
Evaluate training programs, instructors, or materials.	1.33 (0.65)

Draft legislation or regulations.	1.35 (0.63)
Support the professional development of others.	1.36 (0.74)
Counsel clients on mental health or personal achievement.	1.38 (0.70)
Design psychological or educational treatment procedures or programs.	1.40 (0.65)
Guide class discussions.	1.40 (0.59)
Conduct research on social issues.	1.41 (0.71)
Lead classes or community events.	1.42 (0.66)
Counsel clients or patients regarding personal issues.	1.42 (0.61)
Display student work.	1.43 (0.58)
Develop methods of social or economic research.	1.43 (0.69)
Manage organizational or program finances.	1.44 (0.68)
Evaluate scholarly materials.	1.44 (0.66)
Evaluate effectiveness of educational programs.	1.44 (0.58)
Develop promotional strategies for religious organizations.	1.44 (0.77)
Stay informed about current developments in field of specialization.	1.44 (0.59)

Table 3: Average automatability scores of each of the nine high level work activity groups.

Activity Group	**Automatability Score (std)**
Performing Physical and Manual Work Activities	2.96 (0.45)
Identify and Evaluating Job-Relevant Information	2.88 (0.48)
Administering	2.79 (0.55)
Performing Complex and Technical Activities	2.70 (0.52)
Information and Data Processing	2.58 (0.56)
Communicating and Interacting	2.58 (0.47)
Looking for and Receiving Job-Related Information	2.52 (0.48)
Reasoning and Decision Making	2.44 (0.50)
Coordinating, Developing, Managing, and Advising	2.29 (0.49)

Table 4: Automatability scores of each of the 22 major occupation groups.

Major Occupation Group	Employment Weighted Automatability Score (std)
Production	3.40 (0.19)
Office and Administrative Support	3.30 (0.18)
Farming, Fishing, and Forestry	3.16 (0.28)
Sales and Related	3.16 (0.20)
Transportation and Material Moving	3.12 (0.17)
Building and Grounds Cleaning and Maintenance	2.87 (0.10)
Healthcare Support	2.79 (0.15)
Healthcare Practitioners and Technical	2.75 (0.14)
Construction and Extraction	2.74 (0.15)
Food Preparation and Serving Related	2.66 (0.10)
Architecture and Engineering	2.66 (0.23)
Installation, Maintenance, and Repair	2.65 (0.19)
Business and Financial Operations	2.60 (0.31)
Personal Care and Service	2.59 (0.23)
Protective Service	2.53 (0.13)
Arts, Design, Entertainment, Sports, and Media	2.44 (0.35)
Computer and Mathematical	2.42 (0.18)
Life, Physical, and Social Science	2.37 (0.31)
Management	2.17 (0.13)
Legal	1.98 (0.58)
Community and Social Service	1.83 (0.16)
Education, Training, and Library	1.72 (0.21)

Table 5: The 25 work activities where our model disagrees positively and negatively with the ground truth label.

Activity	Ground Truth	Predicted	Disagreement
Connect electrical components or equipment.	1.0	2.69	1.69
Travel to work sites to perform installation, repair or maintenance work.	1.0	2.61	1.61
Clean food service areas.	1.0	2.53	1.53
Locate suspicious objects or vehicles.	1.0	2.52	1.52
Collect dirty dishes or other tableware.	1.0	2.49	1.49
Update knowledge about emerging industry or technology trends.	1.0	2.48	1.48

Arrange tables or dining areas.	1.0	2.47	1.47
Search individuals for illegal or dangerous items.	1.0	2.43	1.43
Collaborate with others to resolve information technology issues.	1.0	2.39	1.39
Operate vehicles or material-moving equipment.	2.0	3.31	1.31
Communicate with customers to resolve complaints or ensure satisfaction.	1.0	2.31	1.31
Exchange information with colleagues.	2.0	3.30	1.30
Direct operational or production activities.	2.0	3.16	1.16
Evaluate employee performance.	1.0	2.15	1.15
Collaborate with others to determine design specifications or details.	1.0	2.14	1.14
Examine animals to detect illness, injury or other problems.	2.0	3.07	1.07
Meet with individuals involved in legal processes to provide information and clarify issues.	1.0	2.04	1.04
Direct material handling or moving activities.	2.0	3.03	1.03
Advise customers on the use of products or services.	2.0	3.02	1.02
Test materials, solutions, or samples.	2.0	3.00	1.00
Monitor loading processes to ensure they are performed properly.	2.0	3.00	1.00
Clean medical equipment.	2.0	2.98	0.98
Assist practitioners to perform medical procedures.	2.0	2.98	0.98
Hire personnel.	1.0	1.98	0.98
Conduct employee training programs.	1.0	1.95	0.95
Maintain student records.	3.5	1.55	−1.95
Count prison inmates or personnel.	4.0	2.32	−1.68
Estimate supplies, ingredients, or staff requirements for food preparation activities.	4.0	2.38	−1.62
Advise others on career or personal development.	3.0	1.45	−1.55
Administer tests to assess educational needs or progress.	3.0	1.55	−1.45
Process customer bills or payments.	4.0	2.55	−1.45

Measure equipment outputs.	4.0	2.63	−1.37
Implement security measures for computer or information systems.	4.0	2.65	−1.35
Conduct research to gain information about products or processes.	4.0	2.73	−1.27
Record patient medical histories.	4.0	2.77	−1.23
Analyze test or performance data to assess equipment operation.	4.0	2.77	−1.23
Position construction forms or molds.	4.0	2.80	−1.20
Refer clients to community or social service programs.	3.0	1.82	−1.18
Maintain client records.	3.0	1.82	−1.18
Prepare reports detailing student activities or performance.	3.0	1.83	−1.17
Create graphical representations of structures or landscapes.	4.0	2.84	−1.16
Plan work operations.	4.0	2.85	−1.15
Create electronic data backup to prevent loss of information.	4.0	2.86	−1.14
Measure materials or objects for installation or assembly.	4.0	2.86	−1.14
Maintain inventory of medical supplies or equipment.	4.0	2.88	−1.12
Manage control system activities in organizations.	3.0	1.92	−1.08
Balance receipts.	4.0	2.92	−1.08
Refer customers to appropriate personnel.	4.0	2.94	−1.06
Care for animals.	4.0	2.94	−1.06
Maintain inventories of materials, equipment, or products.	4.0	2.98	−1.02

B.4 Sensitivity Analysis

Table 6: The 25 most automatability-increasing and decreasing features across the activity space.

Feature	Average Gradient (std)
Telecommunications	0.16 (0.03)
Clerical	0.14 (0.03)
Wrist-Finger Speed	0.13 (0.02)
Number Facility	0.11 (0.02)
Mathematics	0.09 (0.02)
Depth Perception	0.08 (0.01)
Mathematical Reasoning	0.08 (0.02)
Economics and Accounting	0.07 (0.02)
Response Orientation	0.07 (0.02)
Building and Construction	0.07 (0.04)
Control Precision	0.07 (0.02)
Arm-Hand Steadiness	0.06 (0.02)
Equipment Selection	0.06 (0.02)
Finger Dexterity	0.06 (0.01)
Perceptual Speed	0.06 (0.01)
Visual Color Discrimination	0.06 (0.01)
Static Strength	0.05 (0.01)
Sales and Marketing	0.05 (0.06)
Far Vision	0.04 (0.01)
Spatial Orientation	0.04 (0.02)
Flexibility of Closure	0.04 (0.01)
Night Vision	0.04 (0.02)
Manual Dexterity	0.03 (0.01)
Multilimb Coordination	0.03 (0.03)
Production and Processing	0.03 (0.02)
Installation	−0.18 (0.08)
Programming	−0.14 (0.04)
Technology Design	−0.14 (0.03)
Fine Arts	−0.11 (0.05)
Gross Body Equilibrium	−0.10 (0.07)
Dynamic Flexibility	−0.10 (0.03)
Speed of Limb Movement	−0.10 (0.02)
Psychology	−0.10 (0.02)
Personnel and Human Resources	−0.09 (0.02)
Sociology and Anthropology	−0.09 (0.03)

History and Archeology	−0.09 (0.03)
Science	−0.09 (0.04)
Food Production	−0.08 (0.07)
Management of Personnel Resources	−0.07 (0.02)
Glare Sensitivity	−0.07 (0.03)
Troubleshooting	−0.07 (0.02)
Gross Body Coordination	−0.06 (0.03)
Coordination	−0.06 (0.01)
Learning Strategies	−0.06 (0.02)
Law and Government	−0.06 (0.02)
Negotiation	−0.06 (0.01)
Management of Financial Resources	−0.06 (0.02)
Social Perceptiveness	−0.06 (0.01)
Chemistry	−0.06 (0.02)
Explosive Strength	−0.06 (0.04)

Interpretation: On average, an increase of an activity's *Clerical* score by one point (1 to 5 scale), tends to to *increase* its automatability by 0.14. All else equal, if the 10 most automatability-increasing features increase by 1, automatability tends to increase by one class.

B.5 Average Gradients by Activity Group

Table 7: The 5 most positive and 5 most negative "local" feature gradients by activity group.

Activity Group	Feature	Average Gradient
Administering	Clerical	0.14
	Depth Perception	0.11
	Number Facility	0.10
	Control Precision	0.10
	Mathematical Reasoning	0.09
	English Language	−0.09
	Instructing	−0.09
	Management of Financial Resources	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Communicating and Interacting	Clerical	0.14
	Depth Perception	0.11
	Number Facility	0.11
	Control Precision	0.10

	Mathematical Reasoning	0.09
	Management of Financial Resources	−0.09
	Instructing	−0.09
	English Language	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Coordinating, Developing, Managing and Advising	Clerical	0.14
	Number Facility	0.10
	Control Precision	0.10
	Depth Perception	0.10
	Mathematical Reasoning	0.09
	English Language	−0.09
	Management of Financial Resources	−0.09
	Instructing	−0.09
	Management of Personnel Resources	−0.11
	Learning Strategies	−0.12
Identify and Evaluating Job-Relevant Information	Clerical	0.14
	Number Facility	0.10
	Depth Perception	0.09
	Mathematical Reasoning	0.09
	Mathematics (skill)	0.08
	Installation	−0.09
	Customer and Personal Service	−0.09
	Mechanical	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.11
Information and Data Processing	Clerical	0.14
	Depth Perception	0.10
	Number Facility	0.10
	Control Precision	0.10
	Mathematical Reasoning	0.09
	English Language	−0.09
	Instructing	−0.09
	Management of Financial Resources	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Looking for and Receiving Job-Related Information	Clerical	0.14
	Number Facility	0.10
	Depth Perception	0.10
	Control Precision	0.10

	Mathematical Reasoning	0.09
	English Language	−0.08
	Management of Financial Resources	−0.08
	Instructing	−0.00
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Performing Complex and Technical Activities	Clerical	0.14
	Depth Perception	0.11
	Number Facility	0.10
	Control Precision	0.10
	Mathematical Reasoning	0.09
	Management of Financial Resources	−0.09
	Instructing	−0.09
	English Language	−0.09
	Management of Personnel Resources	−0.10
	Learning Strategies	−0.12
Performing Physical and Manual Work Activities	Clerical	0.14
	Number Facility	0.10
	Mathematical Reasoning	0.09
	Depth Perception	0.09
	Mathematics (skill)	0.09
	Mechanical	−0.09
	Installation	−0.09
	Customer and Personal Service	−0.09
	Management of Personnel Resources	−0.09
	Learning Strategies	−0.11
Reasoning and Decision Making	Clerical	0.14
	Number Facility	0.10
	Depth Perception	0.10
	Control Precision	0.10
	Mathematical Reasoning	0.02
	Coordination	−0.02
	Instructing	−0.02
	Management of Financial Resources	−0.09
	Management of Personnel Resources	−0.11
	Learning Strategies	−0.12

B.6 Visualising the Effect of Features on Automatability

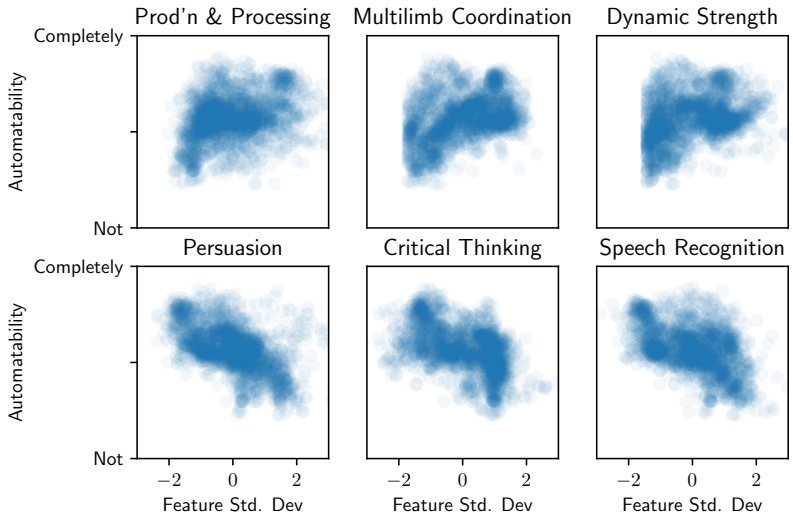


Fig. 4: A sample of 6 features with clusters activities that have feature scores greater than $\sim 2\sigma$ but very different inferred automatability scores. Note that the relationship between feature score and inferred automatability is relatively clear for each feature, yet they still have clusters of outliers at either end. This contradiction suggests useful explanatory theories could be made by studying the contradictions.