Modeling Crowd Work in Open Task Systems

Nicholas Hoernle

Doctor of Philosophy

Artificial Intelligence Applications Institute

School of Informatics

The University of Edinburgh

2022

Abstract

This thesis aims to harness modern machine learning techniques to understand how and why people interact in large and open, collaborative online platforms: *task systems*. The participants who interact with the task systems have a diverse set of goals and reasons for contributing and the data that is logged from their participation is often observational. These two factors present many challenges for researchers who wish to understand the motivations for continued contributions to these projects such as Wikipedia and Stack Overflow. Existing approaches to scientific investigation in such domains often take a “one-size-fits-all” approach where aggregated trends are studied and conclusions are drawn from overview statistics.

In contrast to these approaches, I motivate a three-stage framework for scientific enquiry into the behaviour of participants in task systems. First I propose a modelling step where assumptions and hypotheses from Behavioural Sciences are encoded directly into a model’s structure. I will show that it is important to allow for multiple competing hypotheses in one model. It is due to the diversity of the participants’ goals and motivations that it is important to have a range of hypotheses that may account for different interaction patterns present in the data.

Second, I design deep generative models for harnessing both the power of deep learning and the structured inference of variational methods to infer parameters that fit the structured models from the first step. Such methods allow us to perform maximum likelihood estimation of parameter values while harnessing amortised learning across a dataset. The inference schemes proposed here allow for posterior assignment of interaction data to specific hypotheses, giving insight into the validity of a hypothesis. It also naturally allows for inference over both categorical and continuous latent variables in one model - an aspect that is crucial in modelling data where competing hypotheses that describe the users’ interaction are present.

Finally, in working to understand how and why people interact in such online settings, we are required to understand the model parameters that are associated with the various aspects of their interaction. In many cases, these parameters are given specific meaning by construction of the model, however, I argue that it is still important to evaluate the interpretability of such models and I, therefore, investigate several tests for performing such an evaluation.

My contributions additionally entail designing bespoke models that describe people’s interactions in complex and online domains. I present examples from real-world domains where the data consist of people’s actual interactions with the system.

Acknowledgements

The work that was done in this thesis would not have been possible had it not been for the support that I had.

First and foremost, thank you to my supervisor, Kobi. Your enthusiasm for your work is an inspiration and your ability to articulate and detail complex ideas in writing is masterful. It has been an honour to learn from you and I hope that the exposition to come does your mentorship justice. The time that we spent in Israel was hugely enlightening to me, not only from an academic standpoint, and I thank you greatly for offering me this opportunity.

I thank the Commonwealth Commission Scholarship (CSC) for the financial support throughout my PhD studies. This journey would not have been possible without them. I have benefited hugely from my time in Edinburgh City and for this I am grateful.

Thank you to Cillian and Georgios for the interesting discussions, and sometimes more one-sided rants, over our lunches in the forum. The previous years, during the pandemic restrictions, have been challenging for us all and while I have been disappointed by the support that the university has provided, I am so happy to have found kindred spirits in you two. Postgraduate studies are a fundamentally collaborative endeavour where we should learn from and work with our peers. With empty offices and nobody at work, this is incredibly hard to achieve, however, having the two of you to discuss ideas became a highlight of my day throughout 2021.

To my family Mum, Dad, Ali, Doug and Niki, I thank you for the support throughout this long process. While you might not have always understood my reasons for embarking on this journey, or even for continuing when the process became marred by the pandemic, you have always offered your unconditional enthusiasm and support to me and for that I thank you!

Finally and most importantly, thank you to Theresa for being the main pillar of support for me throughout these studies. You have endured, with me, both the highlights and the low-lights of the process and you have offered your loving support through it all.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Nicholas Hoernle)*

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many online platforms rely on the motivation of their userbase to create the content that their platform delivers (Ipeirotis and Gabrilovich, 2014). Examples of such “task systems” include freely available, online encyclopedias like Wikipedia, Question and Answer (Q&A) sites such as Stack Overflow (SO), citizen science platforms (Simpson et al., 2014) and even mapping applications such as Waze[1], Prusik, and Fatmaps[2]. However, due to the inherent diversity of a platform’s userbase, it can be a challenging mechanism design problem to curate relevant rewards and pathways that ensure the users are motivated and fulfilled in their contributions (Immorlica et al., 2015; Anderson et al., 2013).

A task system broadly consists of an online platform (e.g., SO or Waze) where participants (e.g., users) work to complete tasks (e.g., ask or answer questions on SO or log traffic events on Waze) on the platform (Segal, 2018). Importantly, each of the participants in the task system can have a unique goal. For example, some SO users ask questions with the goal of overcoming a challenge in their work or studies, other users answer questions, sometimes with the goal of testing their knowledge (Hoernle et al., 2020b), and still others, merely try to increase their online developer profile by achieving as many badges and reputation points as possible. A task system must be able to *identify* the major goals and working habits that are present in the user base such that the environment can be designed to support the interaction of its users.

My thesis directly addresses this challenge by presenting a framework for **understanding the behaviour** of participants in task systems. This framework allows for designing and testing models that can be fitted to the interaction data that are collected from these online communities. The models’ fit to the data can be used to compare different hypotheses for people’s behaviour, which stem from behavioural science research, and thereby to validate the archetypal behaviours that are present in the interaction data from a task system’s userbase.

The remainder of this chapter provides an overview of the main content chapters of this thesis. I also introduce the interesting (and sometimes novel) domains that I have explored in my research. Studying and presenting examples of how to use the real-world data that are generated from such domains is a theme throughout my research and it forms an important contribution that I have made.

After the preliminary Background and Related Work Chapters A and 2, I present a case study, in Chapter 3, where we apply the proposed framework to data from SO. We

[1] `https://www.waze.com`

[2] `https://fatmap.com`

show the value of this methodology by identifying and characterising distinct groups of different behaviour. In doing so, we validate a certain hypothesis from behavioural science but we show that this only applies to a fraction of the user base. This chapter corresponds to work that has been published at the 20^{th} IEEE International Conference on Data Mining (ICDM) in 2020 (Hoernle et al., 2020b).

In Chapter 4, I formalise the main inference task in this framework by identifying how constraints can be introduced into a model of user behaviour. Inference over a latent categorical variable is necessary to introduce different and competing hypotheses for how people interact with task systems. Importantly, when there are two or more different modes of interaction in a dataset, a model that aims to describe these data must allow for the possibility of such diversity. This chapter corresponds to work that has been submitted for peer review at the 36^{th} AAAI Conference on Artificial Intelligence 2022.

The final content chapter, Chapter 5, addresses an important aspect of interpretability in the framework. The proposed approach designs a model to implement hypotheses from behavioural science. However, the nature of machine learning is that there are still black-box and uninterpretable aspects to any complex model. Thus it is important to have robust tests that allow for the comparison of the interpretability of competing models. I investigate the design and evaluation of such tests in Chapter 5. I perform this investigation in the context of another real-world domain: This is an immersive exploratory learning environment where students interact together and with the simulation to learn about the causal and temporally delayed effect of their actions on an environment. This chapter corresponds to work that has been published at the 30th International Joint Conference on Artificial Intelligence 2020 (Hoernle et al., 2020a).

1.1 Contributions and Thesis Overview

The contributions of this thesis can be understood in the context of Figure 1.1. Here, I have presented a version of Box's loop that aims to distil the major undertakings when investigating and validating hypotheses from data (Blei, 2014; Box and Hunter, 1962). In particular, I have tied the three undertakings to the specifics that should be performed when attempting to understand how people behave when interacting with task systems.

We first require a modelling phase, where we propose hypotheses about how people might act in certain settings. It is important to define these hypotheses in terms of an

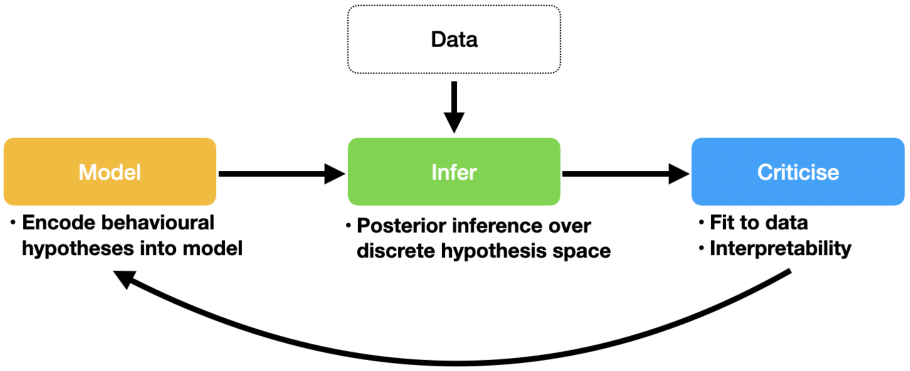

Data
Model
Infer
Criticise
• Encode behavioural hypotheses into model
• Posterior inference over discrete hypothesis space
• Fit to data
• Interpretability

Figure 1.1: Overview of the main contributions of this thesis. Here, Box's loop, showing a standard Data Science investigation pipeline, has been tailored to the task of inferring the behaviour of participants in task systems.

observable quantity that can be measured. For example, in Chapter 3, we use the observed quantity $count(actions)/day$. The assumptions of the hypotheses should be directly encoded into the allowable values for parameters in a model. In Chapter 4, I explore how to perform this encoding when the assumptions of the model become complex and atypical. I thereby allow for complex constraints in the form of a logical program to dictate the parameters' domain.

The inference step is performed separately from the modelling and the encoding of domain assumptions. Here, we aim to fit parameter values to the hypotheses given in step 1 ("Model"). In particular, a challenge arises in how to perform inference over both categorical and continuous latent variables. The continuous latent variables correspond to parameters that encode for the behavioural hypotheses, while the categorical variable allows inference to be performed between competing hypotheses. In Appendix A I provide the necessary preliminaries to understand how inference in these latent variable models is performed. In Chapter 3, I present an example of how this is achieved, based on a case study on SO data. And in Chapter 4, I formalize these inference tasks to allow for the specification of any set of behavioural models.

The final step is to criticise the model (1) on its fit to the data and (2) on its value in understanding how people interact in task systems. Model evaluation is a well-addressed subject in statistics and thus I use standard metrics of the likelihood of held-out data to evaluate the fit of the models to unseen data. However, in Chapter 5, I investigate how to evaluate a model based on its perceived interpretability to people. I assign an interpretability score to different models, and I discuss a trade-off that may be present between how interpretable a model might be and how well it fits the data. In some cases, especially when comparing two models of different complexity, one may achieve better statistical (e.g., predictive) performance while another might be simpler for people to understand. This trade-off is especially important to consider when investigating scientific hypotheses.

The main contributions of this thesis, therefore, touch on all three of the steps highlighted in Figure 1.1. These contributions are also summarised in the three conference papers (two published and one under review) that each corresponds to the three content chapters of this thesis. However, I have worked on several other published works, each connected to the themes that I have explored in this thesis. In Yanovsky et al. (2019, 2021) we explored how the response of users to badge rewards is not homogeneous and we identified the presence of different groups of users in the SO dataset. This work highlighted the need for a more detailed model of user behaviour and it was highly inspirational for Chapter 3 and indeed much of the framework that this thesis espouses.

In Geller et al. (2020), we modeled the interaction of students in a university course discussion forum. Specifically, we identified traits that are associated with confusion and we show that by using the students own use of hashtags, we can identify more accurately, potential threads that display confusion. This work, along with Shillo et al. (2019), where we modelled the creativity of users during an online ideation task, was another example of how to use a machine learning solution to inform novel insights about the behaviour of people when they contribute to a task system.

1.2 The Phantom Steering Effect in Q&A Websites (Chapter 3)

Hull (1932) defined the *goal gradient hypothesis* as the tendency of animals to increase their effort as they near a goal (e.g., rats run faster in a maze when they are closer to a food reward). Kivetz et al. (2006) extended these results to humans, showing that by framing (Kahneman, 2011) a reward in a particular way, customers can be influenced to purchase coffee more frequently. The authors designed two types of loyalty card, one type (the control) with 8 empty spaces and one type with 2 completed spaces and an additional 8 empty spaces. Thus, both groups in the intervention study had to purchase 8 coffees before they received their reward of one free coffee. Their surprising result was that the customers in the second group (having two coffees already completed and thus were relatively closer to their goal – 20% vs 0%) finished their cards faster than the control group. This is an example of how a rewards can motivate behaviour from their target audience.

Past work has shown that virtual badges "steer" people's behaviour toward increasing their overall contributions to online Q&A platforms (Anderson et al., 2013; Li et al., 2012; Yanovsky et al., 2019). That is, users' contribution levels rise as they get closer to the threshold that is required for obtaining the badge, and they experience a sharp decline thereafter, returning to their baseline contribution levels. These works all apply the *goal gradient hypothesis* from behavioural research (Hull, 1932) to online domains and they use evidence from observational data collected from online platforms to demonstrate the validity of this hypothesis in these domains.

In such settings, the mean of a large dataset has been used to study a population-level response to an intervention (e.g., the presence of a badge). However, there is no allowance for competing hypotheses that could also describe how users behave. I show that we can build one model that contains two or more distinct hypotheses for the data. The first is the null hypothesis that states people are not affected by a badge reward: The users under this hypothesis do not change their behaviour around a badge achievement event. A contrasting hypothesis is one that allows for adherence to the goal gradient hypothesis. By allowing for both eventualities in one model, we can perform inference over which of the users change their behaviour in response to a badge and how this change is characterised. *We thereby allow for the possibility of heterogeneity in the users' response to rewards.*

Due to the variety of ways that a user can interact with a platform, it is critical to allow for these different hypotheses when modelling the behaviour of users in task systems. We show that by adopting this more nuanced view of user behaviour, we arrive at more robust conclusions about steering in online domains. Specifically, we show that the users who do change their behaviour appear to experience a much greater effect than was previously identified. Unfortunately, we also show that a vast majority of users on a platform who do achieve these rewards, do not experience the effects of steering. Identifying this fact is important as future studies will be in a position to design more relevant rewards that appeal to a greater portion of users.

1.3 Constraining Deep Generative Networks by Domain Knowledge (Chapter 4)

I present a general means for specifying constraints in a generative model. This formalises the empirical investigation introduced above by allowing any specification of logical constraints for constraining the output domain of a network. Specifically, constraints can describe an expert's prior knowledge about a domain (e.g., the predictions of the goal gradient hypothesis). I explore how to encode these constraints directly and tractably into a generative model.

We assume here that prior knowledge can be specified as a first-order logical formula. This formula places a restriction on the allowable domain for a generative model. Certain choices of observation distribution would place these constraints naturally (e.g., a Poisson distribution implies the random variable being modelled is both discrete and non-negative). However, in this framework, we look to introduce a flexible language for specifying more complex constraints. For example, in Chapter 3 we will see that the predictions of the goal gradient hypothesis need to be encoded into a model's structure. Specifically, under the goal gradient hypothesis, users must experience an increase in the rate of behaviour before the goal is achieved and this should return to a baseline level after the achievement of the goal.

I, therefore, propose a model that (1) accepts any general logical formula over the target observation random variables, and (2) restricts the domain of the model to obey these constraints. This goal of restricting a network's output has been explored in a number of contexts (Manhaeve et al., 2018; Xu et al., 2018; Fischer et al., 2019); however, I show how it can be achieved in conjunction with a generative model. I assume that the logical input can be compiled into a disjunctive normal form (DNF) and I represent the choice of the correct term as a latent categorical variable in the model. This naturally frames the problem as a latent variable problem and allows end-to-end learning that benefits from the domain restrictions. By doing so, we can support disjoint constraints (possibly modelling disjoint modes in the target distribution), and the model learns a posterior assignment of data points to a specific mode.

1.4 Interpretable Models for Understanding Immersive Simulations (Chapter 5)

In designing models that provide useful insights into the behaviour of peoples' interactions in task systems, we have so far assumed that the models are interpretable by construction. For example, the goal gradient hypothesis can be encoded into a model by constraining certain parameters that are used to describe the changes to the users' rate of interaction (as a function of time). The inclusion of a categorical variable that reasons over the competing hypotheses, allows us to consider the parameters associated with each hypothesis in isolation. However, there are scenarios where it might be useful to select a model, not only on its fit to data but also on how interpretable that model is and how useful the insights from the model are. In this chapter, we investigate one such setting — one where we design a new test for measuring interpretable models.

This work used data from an exciting exploratory learning environment called Connected Worlds (Mallavarapu et al., 2019; Hoernle et al., 2018) that is installed at the New York Hall of Science. Here, students in groups of 10 – 20 interact with an environment simulation and attempt to grow plants in different areas of the simulation. The difficulty is that the shared resource (water) needs to be carefully managed to allow life to flourish in all areas of the simulation. We designed time-series models that used the log data from this simulation to attempt to infer periods of time where the students had brought the simulation to a steady-state behaviour.

Specifically, we were interested in evaluating the interpretability of the various models and as such we designed a number of tests that aimed to measure the interpretability of a model. Higher scores on the tests suggest that one model might be more interpretable than another. To this end, we ran an interpretability study that gathered the responses from participants from Amazon Turk and a large undergraduate university in Israel with a total of 240 experiment participants. We scored various models based on their interpretability and we presented an example of how these models can be evaluated on this interpretability measure. Importantly, we also discuss how a trade-off can be made between statistical measures of model fit (held out likelihood) and the interpretability score when comparing models of different complexity.

Chapter 2

Related Work

2.1 Introduction

This chapter relates my work to that of the broader community. First, I deal with the related work of incentive and mechanism design in online spaces. This is presented in Section 2.2 and the work relates to Chapter 3. Second, in Section 2.3, I consider prior work that uses constraints in neural network design. The work presented in this section is related to Chapter 4. Finally, I review the important work that has been done on evaluating the interpretability of black-box models. This review is presented in Section 2.4 and it relates to the work in Chapter 5.

2.2 Virtual Badges and the "Steering Effect"

I begin this section by relating to the general literature on the effect of badges in online communities. I then present, in detail, the specific work of Anderson et al. (2013) which helps to motivate the generative models that we develop in Chapter 3.

2.2.1 The Study of Online Badges

The goal-gradient hypothesis stems from behavioral research where animals were observed to increase their effort as they approach a reward (Hull, 1932; Kivetz et al., 2006). Kivetz et al. (2006) studied the behavior of different populations of people who were working toward various rewards. They concluded that the goal-gradient hypothesis also holds true for people. Subjects who received a loyalty card, which tracked the number of coffees purchased from a local coffee chain, purchased coffee significantly more frequently the closer they were to earning a free cup of coffee. The authors recognized the existence of a group of participants who did not complete their coffee cards for the duration of the study, and did not exhibit a noticeable change in their coffee purchasing habits. They concluded that the loyalty card effect was constrained to the population of participants who handed in their completed loyalty cards in exchange for the free-coffee reward. However, the authors had no means for estimating what fraction of users did not submit their cards and therefore they could not estimate how pervasive this effect might be when evaluated on the population at large.

Anderson et al. (2013) and Mutter and Kundisch (2014) were the first to study the goal-gradient hypothesis in online settings. They studied the *observed* effect of virtual badge rewards on the behavior of participants in large Q&A sites. Both studies found evidence that users increase their rate of work as they approach the badge threshold.

However, they did not address the possibility that some users might achieve the badge as a consequence of their routine interactions on the website rather than being steered by the badge. There is a possibility that people's actions are governed by motivations other than badges. In Chapter 3, we extend these works by allowing for this possibility, such that we can characterize the true changes to users' behavior under the influence of a badge, and distinguish this from the case where users do not noticeably change their interaction behavior.

Other studies have independently confirmed that the presence of online badges increases the probability that a user will act in a manner to achieve the badge, as well as the rate at which the user will perform those actions (Kusmierczyk and Gomez-Rodriguez, 2018; Yanovsky et al., 2019; Bornfeld and Rafaeli, 2017; Ipeirotis and Gabrilovich, 2014). Kusmierczyk and Gomez-Rodriguez (2018) highlight the importance of modeling the "utility heterogeneity" among the users but they study badges which have a threshold of 1 action and do not characterize *how* one might change one's behavior in the presence of the badge incentive. Yanovsky et al. (2019) study the presence of different populations within the SO database by employing a clustering routine. They discovered notably different responses to the badge based on the cluster that a user belongs to. Their study did not acknowledge the possibility that the observed data might be consistent with a hypothesis that some users do not exhibit steering. Anderson et al. (2014) studied the implementation of a badge system in a massive open online course and they provide a prescriptive system for the design of badges such that there is a maximum effect on the population. Zhang et al. (2019) suggest that SO create new badges to encourage users to integrate helpful comments into the accepted answers. They thereby present an example of how system designers might use a badge to encourage a desired behavior from their user base. In contrast to this, we suggest that badges have a limited scope and work should be completed to understand other motivations that the users' have such that better and more effective rewards can be designed to motivate online communities.

2.2.2 A Utility Model for Steering

Most relevant to our work in Chapter 3 is the paper from Anderson et al. (2013) who present a parametric description of a user's utility when the user is steered by badges. The model describes users as having their own preferred distribution from which actions are sampled. As users approach the required threshold for achieving a badge, they *deviate* from their preferred distributions. The deviation from the preferred distribution is controlled by the utility gained by achieving the badge and the cost for deviating from the preferred distribution.

We let A_u^d refer to the distribution over the count of actions that a user u takes on day d. The user's utility is a function of A_u^d and it is the sum of three terms.[1] The first term, $\sum_{b \in B} I_b V_b$, is the non-negative value that a user derives from already-attained badge rewards (where V_b is the assumed value of a badge and I_b is the indicator that the user has attained badge b). The second term, $\theta \mathbb{E}_{A_u^d}[U_{u,d+1}(A_u^{d+1})]$, describes the user's expected future utility, discounted by θ, when acting under the distribution A_u^d. The final term, $g(A_u^d, P_u^d)$, is a cost function that penalises the user for deviating from the preferred distribution P_u^d on that day. The cost g represents the unwillingness of the users to change their behavior, and it is in tension with the users' desire to achieve future badges.

We note that the strictly positive "badge value term" ($\sum_{b \in B} I_b V_b$) and the strictly negative "cost term" ($g(A_u^d, P_u^d)$) could be represented by one "reward" term (allowed to be both positive or negative). However, Anderson et al. (2013) make assumptions about the convexity of the cost term and thus find it useful to make this distinction. It is due to the need for such assumptions that we motivate for modeling the behaviours directly and not an abstract reward that might be hard to quantify.

The utility on day d for user u is then (Anderson et al., 2013):

$$U_{u,d}(A_u^d) = \sum_{b \in B} I_b V_u^b + \theta \mathbb{E}_{A_u^d}[U_{u,d+1}(A_u^{d+1})] - g(A_u^d, P_u^d)$$

It is important to note that the cost term g is non-zero only when users deviate from their preferred distribution P_u^d. As such, this model assumes users deviate only to attain the value from the badge and only if that value outweighs the cost that is paid for deviating. This means that a deviation on the rate of actions which are incentivised by the badge must be an increase before the badge is achieved and cannot be an increase after the badge is achieved (under a standard utility-theoretic assumption that all the utility of the badge is conveyed to the user upon receipt of the badge). We will make these same assumptions in the models presented in Chapter 3.

This utility-based model presents a compelling description of how people respond to badges; however, it was not evaluated or tested by fitting it to specific data from

[1] Our notation differs slightly from that of Anderson et al. (2013). Anderson et al. (2013) uses a parameter $\mathbf{x_a}$ to refer to a user's distribution over the next action. We rather use A_u^d to denote the distribution over the count of actions on a particular day. The two are linked (the distribution over the next action influences the count of actions on a specific day), however, we choose to model directly the data that is available from SO rather than a quantity that we do not observe.

SO. Rather, predictions of the model were compared to aggregated data from SO and we show in Section 3.6 that the aggregated analysis from these count data can lead to incorrect conclusions. The lack of analysis on individual level predictions limits the credibility of the study as well as its practical value — it is difficult to apply the utility-based model to the mechanism design problem of badge placement without a means for determining the appropriate model parameters for a given community of contributors.

In Chapter 3 we address the shortcomings of the utility-based approach by introducing a probabilistic model which allows us to use the vast literature on posterior inference in such models to assist with parameter estimation (Blei, 2014; Rezende and Mohamed, 2015; Kingma and Welling, 2014; Kingma et al., 2016; Ranganath et al., 2014). The probabilistic model has two advantages over this prior work: (1) posterior distributions for latent parameters in the model can be learnt from real-world interaction data and (2) the model's fit to data can be used to test and update scientific hypotheses (for example, in this paper we propose and validate that while some users may steer in a similar way, there exist users who do not experience steering).

2.3 Incorporating Domain Constraints into the Training of Deep Neural Networks

The integration of domain knowledge into the training of neural networks is an emerging area of focus. Many previous studies attempt to translate logical constraints into a numerical loss. The two most relevant works in this line are the DL2 framework by Fischer et al. (2019) and the Semantic Loss approach by Xu et al. (2018). DL2 uses a loss term that trades off data with the domain knowledge. It defines a non-negative loss by interpreting the logical constraints using fuzzy logic and defining a measure that quantifies how far a network's output is from the nearest satisfying solution. Semantic Loss also defines a term that is added to the standard network loss. Their loss function uses weighted model counting (Chavira and Darwiche, 2008) to evaluate the probability that a sample from a network's output satisfies some Boolean constraint formulation. The work in Chapter 4 differs from both of these approaches in that we do not add a loss term to the network's loss function, rather we compile the constraints directly into its output. Furthermore, in contrast to the works above, any network output from the approach in Chapter 4 will satisfy the domain constraints, which is crucial in certain settings; e.g., safety critical domains.

In Chapter 4 we introduce the MultiplexNet approach. It is important to compare the expressiveness of the MultiplexNet constraints to those permitted by Fischer et al. (2019) and Xu et al. (2018). The constraints in MultiplexNet can consist of any quantifier-free linear arithmetic formula over the rationals. Thus, variables can be combined over $+$ and $\geq$, and formulae over $\neg$, $\vee$ and $\wedge$. For example, $(x+y \geq 5) \wedge \neg(z \geq 5)$ but also $(x+y \geq z) \wedge (z > 5 \vee z < 3)$ are well defined formulae and therefore well defined constraints in our framework. The expressiveness is significant — for example, Xu et al. (2018) only allow for Boolean variables over $\{\neg, \wedge, \vee\}$. While Fischer et al. (2019) allow non-Boolean variables to be combined over $\{\geq, \leq\}$ and formulae to be used over $\{\neg, \vee, \wedge\}$, it is not a probabilistic framework, but one that is based on fuzzy logic. Thus, the work in Chapter 4 is probabilistic like the Semantic Loss (Xu et al., 2018), but it is more expressive in that it also allows real-valued variables over summations too.

Hu et al. (2016) introduce "iterative rule knowledge distillation" which uses a student and teacher framework to balance constraint satisfaction on first order logic formulae with predictive accuracy on a classification task. During training, the student is used to form a constrained teacher by projecting its weights onto a subspace that ensures satisfaction of the logic. The student is then trained to imitate the teacher's restricted predictions. Hu et al. (2016) use soft logic (Bach et al., 2017) to encode the logic, thereby allowing gradient estimation; however, the approach is unable to express rules that constrain real-valued outputs. Xsat (Fu and Su, 2016) focuses on the Satisfiability Modulo Theory (SMT) problem, which is concerned with deciding whether a (usually a quantifier-free form) formula in first-order logic is satisfied against a background arithmetic theory; similar to what we consider. They present a means for solving SMT formulae but this is not differentiable. Manhaeve et al. (2018) present a compelling method for integrating logical constraints, in the form of a ProbLog program, into the training of a network. However, the networks are embedded into the logic (represented by a Sentential Decision Diagram (Darwiche, 2011)), as "neural predicates" and thus it is not clear how to handle the real-valued arithmetic constraints that we represent in MultiplexNet.

Chapter 4 also relates to work on program synthesis (Solar-Lezama, 2009; Jha et al., 2010; Feng et al., 2017; Osera, 2019) where the goal is to produce a valid program for a given set of constraints. Here, the output of a program is designed to meet a given specification. These works differ from Chapter 4 as they don't focus on the core problem of aiding training with the constraints and ensuring that the constraints are fully satisfied.

Other recent works have also explored how human expert knowledge can be used to guide a network's training. Ross and Doshi-Velez (2018); Ross et al. (2017) explore how the robustness of an image classifier can be improved by regularizing input gradients towards regions of the image that contain information (as identified by a human expert). They highlight the difficulty in eliciting expert knowledge from people but their technique is similar to the other works presented here in that the knowledge loss is still represented as an additive term to the standard network loss. Takeishi and Kawahara (2020) present an example of how the knowledge of relations of objects can be used to regularise a generative model. Again, the solution involves appending terms to the loss function, but they demonstrate that relational information can aid a learning algorithm. Alternative works have also explored means for constraining the latent variables in a latent variable model (Ganchev et al., 2010; Graça et al., 2007). In contrast to this, we focus on constraining the output space of a generative model, rather than the latent space.

Finally, we mention work on the post-hoc verification of networks. Examples include the works of Katz et al. (2017) and Bunel et al. (2018) who present methods for validating whether a network will operate within the bounds of pre-defined restrictions. Our own work in Chapter 4 focuses on how to guarantee that the networks operate within given constraints, rather than how to validate their output.

2.4 Evaluating the Interpretability of Machine Learning Models

Doshi-Velez and Kim (2017) suggested three tests to evaluate how interpretable a model's representations are to people. Forward Simulation: requires a human evaluator to predict the output of a model for a given input. Binary Forced Choice: requires an evaluator to choose one of two plausible model explanations for a data instance. Counterfactual Simulation: requires an evaluator to identify what must be changed in an explanation to correct it for a given data instance.

In follow-up work Lage et al. (2018) propose a model selection process that considers both a model's accuracy and its degree of interpretability, according to one of the above tests. They provide a framework for iteratively optimizing the interpretability of a model with a human-in-the-loop optimization procedure. Their work applied this framework to tests in the lab in which human judgment was used to optimize supervised learning models. Other works that studied interpretability tests for supervised learning settings include Wu et al. (2018); Ribeiro et al. (2016); Choi et al. (2016); Lipton (2016). In Chapter 5, we extend this literature on interpretability by adapting the model selection process to an unsupervised learning setting, that of segmenting a multi-dimensional time series into periods. Moreover, we implement examples of the Forward Simulation and Binary Forced Choice tests suggested by Doshi-Velez and Kim (2017) and apply them to a high dimensional time series setting.

Our work was inspired by Chang et al. (2009) who were the first to show that optimizing machine learning models in unsupervised settings using predictive log-likelihood may not induce models that are interpretable to people. They focused on the use of topic models for finding meaningful structure in documents and they compared the models that are selected to optimize *perplexity* (analogous to held-out log-likelihood) to the models that were selected by the human interpretability tests that they designed. Chang et al. (2009) operationalized two Forward Simulation tests for evaluating the interpretability of a topic model: word intrusion, in which the evaluator is required to identify which of several words does not belong together in one topic represented by the other words; and topic intrusion, in which the evaluator is required to identify which of several topics is not associated with a given document. We extend this work to a multi-dimensional time series domain and we introduce a Binary Forced Choice test to complement the “intrusion” Forward Simulation test.

Chapter 3

The Phantom Steering Effect in Q&A Websites

3.1 Introduction

A well-known finding from behavioural science research is that efforts towards a goal increase with proximity to that goal. This phenomenon, termed the goal-gradient hypothesis, has been demonstrated in a variety of settings, from animal studies in the lab to consumer purchasing behavior (Hull, 1932; Kivetz et al., 2006). More recently, the goal-gradient effect was observed in people's behaviour in online communities that use virtual rewards, such as badges and reputation points, to increase users' contributions to the site (Mutter and Kundisch, 2014; Anderson et al., 2013). In these contexts, the goal-gradient hypothesis has been referred to as "steering" Anderson et al. (2013, 2014). Recent examples of online settings that use badges include communication platforms such as MS teams, ride-sharing platforms such as Lyft and online learning platforms such as Duolingo.

In this chapter, we study the steering phenomenon, in one such community, that of Stack Overflow (SO), where users can acquire badges and obtain reputation points for making different contributions to the platform, such as editing or voting on posts. We identify *who* exhibits steering, who does not, and *how* this steering behaviour can be characterised from observational data. Our surprising result is that a large population (at least 60%) of highly active badge achievers, do not appear to exhibit steering towards those badges.

We present a generative model of steering which models users as having default activity rates that they can deviate from when approaching the requirements for achieving a badge. The model can fit a complex multimodal distribution over the parameters that govern users' activities. This allows it to capture different levels of steering in the population. We apply the model to data collected from thousands of SO users, and investigate the following research questions:

1. Are all badge achievers affected by the steering (or goal-gradient) hypothesis in the same way?

2. If some users do not steer, what portion of the population falls under this category?

3. Does the presence of these users in the dataset change any conclusions that were previously drawn about the phenomenon of steering?

Our results revealed the following insights: First, more than 60% of the users are not steered, in that they exhibit a consistent activity rate in SO that is not affected by the badge. We prove that a "bump" in the activity that is conveyed by prior work arises as an artefact of centring the data on the date of badge achievement (Anderson et al., 2013; Yanovsky et al., 2019). Conceptually, given i.i.d draws from a static, non-negative distribution (e.g., draws from a Poisson distribution with a constant rate parameter), we are interested in studying the mean value of the draw that crosses some predefined threshold. A larger draw is more likely to cross the threshold than any other draw and thus the mean of the draw that crosses the threshold is higher than the mean of the original random variable. It is also evident that a draw of 0 is always possible from a Poisson distribution. However, 0 makes no progress toward crossing the threshold, this fact alone means the mean of the draw that crosses the threshold must be higher than the original distribution's mean. We call this phenomenon *the Phantom Steering Effect* and we formalise this intuition with a discussion and a proof in Appendix B.1. Second, about $5\% - 30\%$ of users are steered, in that they dramatically increase their rate of activity before achieving the badge. It is the effect that this small population of steered users has on aggregate measures that have led to the previous and broader claims of steering (Anderson et al., 2013; Yanovsky et al., 2019; Mutter and Kundisch, 2014). Third, a large portion of these steered users decrease their activity rate beyond what is claimed in prior work (Anderson et al., 2013), reaching close to 0 after the badge has been achieved. Our conclusions are supported by responses to a user survey that included 70 active SO participants, in which only 24% of participants selected badges as a motivating factor for their contribution.

We extend our approach to modelling people's behaviour under another popular incentive mechanism in SO, that of reputation points thresholds. When users cross predefined thresholds, they earn privileges on the site. For example, crossing 200 points results in a reduction of advertisements; $1K$ points denotes users as "established" and gives them the option to see the total count of both up and down votes on a post; and $20K$ points unlocks further editing, deletion and un-deletion privileges. There are other thresholds, all associated with privileges on SO that can be found on the SO webpage.[1] We argue that crossing a threshold and earning the associated privileges, can be viewed in the same light as earning a badge (Immorlica et al., 2015). Thus, in this work, we occasionally refer to the achievement of a reputation threshold and the achievement of a badge synonymously. This investigation applies the same model used for badges to the reputation point threshold and investigates whether the above hypotheses hold in this new setting.

[1] `https://stackoverflow.com/help/privileges/`

Our results revealed that more than 90% of the threshold achievers were not steered by the threshold. For the small minority of users that did change their behaviour, this change mirrored that from the badge study. Moreover, we find an inconsistency between the qualitative, self-reported results from the user survey and the computational results that are presented. Users claimed that the privileges were a motivating factor towards further contribution to SO but our computational results suggest that the effect is limited. As such, we posit that such rewards may still contribute towards an ecosystem that can keep users engaged even if the goal gradient effect is not directly displayed.

Our study has important ramifications for system designers who invest resources into the implementation of badge rewards systems and for researchers who wish to understand the factors that contribute towards users' continued participation in online communities. It provides a sobering perspective on the efficacy of badges and reputation point thresholds as effective incentives, in that for much of the population, the steering effect does not appear to hold. This does not mean that the ecosystem fails to incentivise users. It is possible that rewards that foster a "sense of community" (Immorlica et al., 2015) engage users toward continued contribution. However, our results do suggest that the steering effect (goal gradient hypothesis) holds only in a limited capacity (Anderson et al., 2013; Mutter and Kundisch, 2014).

3.1.1 Contributions

In the work that follows, I was the main contributor to this project. I designed the experiments, implemented the algorithms for the experiment, designed the user study and deployed and collected these data. My co-author, Greg Kehne, helped greatly in providing the mathematical expertise that resulted in our proof in Section 3.6. Of course, this chapter would not have been possible without the guidance from our supervisors Kobi Gal and Ariel Procaccia.

3.2 Modeling User Activities

We model users' activities in SO as a distribution over their action counts. The model aims to incorporate the major aspects of the utility model from Anderson et al. (2013) but it frames the problem such that parameters can be estimated from data and the models can be tested on their fit to unseen user action data to allow for model comparison (Box and Hunter, 1962; Blei, 2014). Moreover, the model allows for different users to experience different levels of steering allowing for a more detailed investigation into the steering phenomenon.

3.2.1 A Generative Model of Steering

Let P_u be a latent parameter that controls the rate of activity for user u; this is the *preferred distribution* of user u. P_u induces a probability distribution over the action counts A_u of user u. Let β denote the deviation of the user's activity from P_u as a result of steering. The observed data for each user, A_u, consists of daily action counts for a predetermined number of weeks before and after achieving the badge. Thus, for D days of interaction, P_u, A_u and β are all vectors of length D.

Figure 3.1 presents four plausible generative models of user behaviour in SO where each model presents an increasingly complex description of how people might respond to badge incentives. White circles denote latent random variables and coloured circles denote observed random variables; solid lines represent conditional dependence between the random variables. Model 0 (Figure 3.1, left) describes a non-steering scenario, in which the observed action counts, A_u, depend only on the user's preferred distribution, P_u. Model 1 (Figure 3.1, center-left) is a steering model in which all users deviate systematically from P_u in a manner that is controlled by β. As the values for β increase (above 0), the users experience an increased activity rate (above their preferred distribution). Similarly, as β decreases (below 0), the users experience a decreased activity rate. Model 1 assumes that all users are steered in the same way. Model 2 (Figure 3.1, center-right) relaxes this assumption by introducing a user-specific Bernoulli parameter $S_u \in \{0,1\}$ dictating whether or not user u adheres to the effect of β. Finally, we introduce Model 3 (Figure 3.1, right) which allows for K different deviations where each deviation, β^k, describes a different response to the badge incentive. For this model, $S_u \in \{0,\dots,K\}$ now represents a Categorical random variable that selects which deviation, β^k, that user u adheres to.

The parameter β, which controls how a user responds to a badge, is a vector of length D (each day relative to the date of badge achievement). Reflecting the intuition that steering positively influences a user before the badge achievement, we constrain β to be strictly positive before *day* 0 — the day when the user achieves the badge. Moreover, for the Models 1 and 2, we constrain β to be strictly negative after this day to reflect the intuition that a user gains no further utility from the badge once it has

	Deviation from P_u	Model 0	Model 1	Model 2	Model 3
β^1	Set to **0**; No deviation.	✓	✓	✓	✓
β^2	Non-negative before badge; Non-positive after badge.		✓	✓	✓
β^3	Non-negative before badge; **0** after badge.				✓

Table 3.1: Table detailing the constraints on the β^k parameters and which models these parameters apply to

been achieved (and thus does not work harder than his preferred distribution P_u). β therefore implicitly includes the trade-off between the cost function g and the badge utility V that is discussed in Section 2.2.2. We relax this second constraint for Model 3 to test the hypothesis that users maintain their base rate of activity well after the achievement of the badge, as is described by Anderson et al. (2013); Yanovsky et al. (2019).

Model 3 includes three possibilities for β^k; $k \in \{1,2,3\}$. β^1 sets the deviation to **0** implying no deviation and capturing the assumptions of Model 0. β^2 uses the same assumptions from the Models 1 and 2 above in that β^2 is strictly positive before the badge is achieved and strictly negative after this day. Finally, β^3 is strictly positive before the badge is achieved but it is set to **0** after this day. These details are summarised in Table 3.1.

3.2.2 Likelihood of Action Counts

In this section, we define the parameters that govern the distribution over users' action counts in SO. We wish to describe a variety of behaviours, including users who contribute sporadically and those who are more consistent. We therefore model action counts using a zero-inflated Poisson distribution. The zero-inflated Poisson distribution has a rate parameter, λ_u^d, and a Bernoulli probability, α_u^d, associated with each user u and each day d of interaction. The Bernoulli parameter, α_u^d, describes the probability that user u is active or not on a given day d. The rate parameter, λ_u^d, describes the expected count of actions that the user will perform under a Poisson distribution, conditioned on the user being active on the platform. Note that a user can be active on the platform without acting (e.g., logs on to the SO website but does not contribute). Conceptually, this would correspond to drawing a 1 from the Bernoulli distribution but a count of 0 actions from the Poisson distribution.

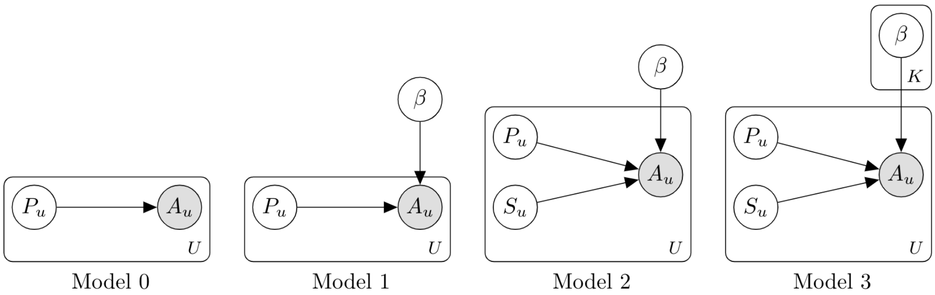


Figure 3.1: Model 0 (baseline model) has no notion of a badge — only a user's preferred distribution induces the distribution over the observed actions. Model 1 allows for a global badge deviation (β) from a user's preferred distribution and it is experienced by all users. Model 2 has a user-specific strength parameter (S_u) that selects whether or not user u adheres to β. Model 3 allows for multiple parameters ($\beta^k, k \in 1, \ldots, K$) that the users might adhere to, in this model (S_u) becomes a switching variable that chooses between the β^k parameters.

The probability that user u performs m actions on day d is presented in (3.1). We refer to the parameters α_u^d and λ_u^d as a user's *rate parameters* for day d.

$$Pr[A_u^d = m] = \begin{cases} (1-\alpha_u^d) + \alpha_u^d Pois(0 \mid \lambda_u^d), & \text{if } m = 0 \\ \alpha_u^d Pois(m \mid \lambda_u^d), & \text{otherwise} \end{cases} \tag{3.1}$$

3.2.3 Deriving the Rate Parameters α_u and λ_u

This section connects the rate parameters, α_u and λ_u, to the generative models of Section 3.2.1. Each of P_u, β^k and S_u includes one component for α_u and one component for λ_u. As such, for D days of interaction, $P_u = (P_{u,\alpha}, P_{u,\lambda})$ comprises two real-valued vectors, each of length D. $P_{u,\alpha}$ is the user's preferred distribution that is associated with α_u and $P_{u,\lambda}$ is the user's preferred distribution associated with the parameter λ_u. Similarly, $\beta^k = (\beta^k_\alpha, \beta^k_\lambda)$ comprises two real-valued vectors of length D that are associated with α_u and λ_u respectively. Finally, S_u is a tuple of two Categorical variables (of order K) that selects among the steering parameters β^k. When there is only one steering parameter, Model 2 is accurately described by Model 3 by setting $K = 2$ and $\beta^1 := \mathbf{0}$. In this special case, the variable S_u becomes a Bernoulli random variable that indicates the presence (or lack thereof) of the steering parameter β^2. As such Model 2 is a simplification of Model 3; similarly, Models 0 and 1 can be seen to simplify Model

2.

Equation (3.2) derives a vector of probability values α_u (one for each day of interaction) as the element-wise sigmoid transformation of a vector that is the addition of the user's preferred distribution $P_{u,\alpha}$ with β_α^j where β_α^j is the steering parameter that is selected by $S_{u,\alpha} = j$. Equation (3.2) also derives a vector of strictly positive rate values λ_u (one for each day of interaction) as the element-wise softplus transformation of the vector $P_{u,\lambda} + S_{u,\lambda}^j \times \beta_\lambda^j$. Below, $\mathbb{1}^j$ refers to the indicator variable that is 1 if $S_u, . = j$ and 0 otherwise.

$$\begin{aligned} \alpha_u &= \sigma\left(P_{u,\alpha} + \sum_{j=1}^{K} \mathbb{1}_{S_{u,\alpha}}^j \times \beta_\alpha^k\right) \\ \lambda_u &= softplus\left(P_{u,\lambda} + \sum_{j=1}^{K} \mathbb{1}_{S_{u,\lambda}}^j \times \beta_\lambda^k\right) \end{aligned} \tag{3.2}$$

The complete generative description for Model 3 is as follows in Algorithm 1. Models 2, 1 and 0 are generated in the same way with the corresponding restrictions on β and S_u and the algorithms are given in Appendix B.4.

Algorithm 1 Generative Pseudocode for Model 3

```
$Z_u \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Sample $Z$ from standard normal prior (see Section 3.3)}
$S_u \sim Categorical(\mathbf{1})$ {Sample $S$ from Categorical Prior (see Section 3.3)}
$P_u = f_\theta(Z_u)$ {Compute $P_u$ as a forward pass of the network $f_\theta$}
Use Eq. (3.2) to compute $\alpha_u$ and $\lambda_u$ from $P_u$ and $S_u$.
Sample $A_u$ from zero-inflated Poisson likelihood Eq. (3.1)
```

In practice, we wish to model the activity of users on SO as they progress through time as accurately as possible. We, therefore, employ a recurrent model, and in our experiments, we used a GRU with a single hidden layer (Chung et al., 2014). This approach uses the product rule of probability to factor the joint distribution over actions through time (recalling that the actions also depend on the users' steering parameters, S_u, and their preferred distributions, P_u). Below the notation $A_{u,<T}$ is used to refer to all actions users u performs before time step T:

$$\begin{aligned} p(A_{u,\leq T} \mid P_u, S_u) = p(A_{u,T} \mid A_{u,<T}, P_u, S_u) \dots \\ p(A_{u,T-n} \mid A_{u,<T-n}, P_u, S_u) \dots p(A_{u,1}, P_u, S_u) \end{aligned} \tag{3.3}$$

3.3 Amortized Variational Inference for Steering

A fully-specified generative model defines a joint distribution over some latent random variables, P_u and S_u, and the observed random variables, A_u. The challenge is to infer the posterior of the latent parameters given the data that was actually observed: $p(P_u, S_u \mid A_u)$. For all but a handful of conjugate models, the posterior is intractable to derive analytically (Neal, 1993; Blei et al., 2003; Hoffman et al., 2013).

Rather, to infer the underlying parameters in the latent space, we use amortized variational inference (Ranganath et al., 2014; Kingma and Welling, 2014; Kingma et al., 2014). Amortized inference uses a neural network to encode a data point into the latent parameters that are associated with its approximate posterior distribution. Moreover, the inference objective allows model comparison such that hypotheses about the data can be tested (e.g., allowing us to validate the inclusion of the steering parameter, S_u).

Variational inference is a popular method for approximating the intractable posterior distribution by introducing a different (and more easily sampled from and evaluated) distribution over the same latent variables: $q(A_u, S_u) = q(A_u)q(S_u)$. By minimizing the KL-divergence between $q(A_u, S_u)$ and the true posterior $p(A_u, S_u \mid A_u)$, one obtains an approximation to the true posterior (Hoffman et al., 2013).

It is important to note that minimizing the KL-divergence between $q(A_u, S_u)$ and $p(A_u, S_u \mid A_u)$ is equivalent to maximizing the variational objective, called the Evidence Lower BOund (see Hoffman et al. (2013) for a derivation and discussion of the ELBO). This *ELBO* derives its namesake from the fact that it lower-bounds the marginal log-likelihood of the data under the assumptions of the model, a fact easily derived in Equations 3.4 and 3.5, where Jensen's inequality is applied in the second line of Equation 3.5 (Bishop, 2006). It is due to this lower bound on the marginal log-likelihood, that it is also common to use the ELBO for model comparison, as is done in Section 3.5.1 (Burda et al., 2015).

$$\begin{aligned}\log p(A_u) &= \log \int \sum_{S_u} p(A_u, P_u, S_u) \partial P_u \\ &= \log \int \sum_{S_u} q(P_u, S_u) \frac{p(A_u, P_u, S_u)}{q(P_u, S_u)} \partial P_u\end{aligned} \tag{3.4}$$

The second line in Equation 3.4 can be recognised as computing the expectation of $\frac{p(A_u, P_u, S_u)}{q(P_u, S_u)}$ with respect to the approximating distributions $q(P_u, S_u) = q(P_u)q(S_u)$.

Moreover, we assume $q(P_u)$ exists in a distributional family where it is possible to compute the pathwise derivative via the reparameterisation trick (Kingma and Welling, 2014). As the steering parameter, S_u, is not continuous, this same reparameterisation cannot be done. It is possible to replace the Categorical variable with a continuous approximation as is done by Maddison et al. (2017) and Jang et al. (2016); or, if the dimensionality of the Categorical variable is small, it can be marginalised out (Kingma et al., 2014). We choose this latter approach leading to the ELBO as defined in Equation 3.5.

$$\begin{aligned}
\log p(A_u) &= \log \mathbb{E}_{q(P_u,S_u)}\left[\frac{p(A_u,P_u,S_u)}{q(P_u,S_u)}\right] \\
&\geq \mathbb{E}_{q(P_u,S_u)}\left[\log\frac{p(A_u,P_u,S_u)}{q(P_u,S_u)}\right] \\
&= \sum_{S_u} \mathbb{E}_{q(P_u)}\left[q(S_u)(\log p(A_u,P_u,S_u) - \log q(P_u) - \log q(S_u))\right] \\
&:= ELBO(A_u)
\end{aligned} \tag{3.5}$$

Throughout this discussion, we have assumed that P_u is directly related to the rate parameters α and λ, made explicit in Equation (3.2). However, we do not implement this quantity directly. Rather, we represent P_u as the output from a transformation of an isotropic Gaussian: $P_u = f_\theta(Z)$ where Z is a standard normal, and f_θ is a parameterised network. This is done partly for convenience and partly as we are not interested in the explicit posterior of P_u. P_u is therefore not a distribution in this construction, however, should the case arise that we do need to explicitly model P_u, we can change the implementation to correctly represent a valid distribution. For example, a normalizing flow (Rezende and Mohamed, 2015) would correctly constrain P_u to be a distribution. Therefore, following standard practice $q(Z)$ is assumed to be an isotropic Gaussian with $\mu_\Phi(A_u)$ and $\sigma^2_\Phi(A_u)$ computed by an inference (encoding) network with parameters Φ. The prior $p(Z)$ is a standard normal Gaussian distribution (again emphasizing that $P_u = f_\theta(Z)$). Similarly, the categorical encoding distribution $q(S_u)$ simply computes the probability that user u belongs to class j, $j \in \{1,\dots,K\}$.[2]

[2] All modeling and inference code can be found at the repository: `https://github.com/NickHoernle/icdm2020`

3.4 Data Domains for Empirical Study

We consider two types of threshold rewards that are present on SO. The first is the threshold badge rewards that are awarded for completing common actions on the website. Completing the required action directly progresses a user toward the threshold for achieving the badge. The second type of threshold reward are the privileges that are awarded for reaching a predefined number of reputation points. These privileges "*control what [users] can do on Stack Overflow [and users] gain more privileges by increasing their reputation.*"[3] The privilege rewards are in contrast to the badge rewards that we study in that the reputation point system requires feedback from other users, in the form of accepts and upvotes, whereas a user can progress toward a threshold badge directly by completing the requisite action (Anderson et al., 2013). We aim to investigate the prevalence of steering in these two settings and to document any structural differences in how people respond to these different reward types.

We consider four common badge types on SO. Table 3.2 details the different badges that we study. We present: **Incentivised Action** – the specific action(s) that the badge is designed to incentivise; **Threshold** – the required number of that actions that should be completed to achieve the badge; and, **# Users** – the number of users in the sample that have achieved the badge. Note that the Electorate badge incentivises one of the same actions (question-votes) as the Civic Duty badge but it has a higher requirement for achievement. We have removed all the users who achieved the Electorate badge from our study of the Civic Duty badge, to remove the confounding effect of the Electorate badge on the users who achieved Civic Duty. The same holds for the Copy Editor badge which incentivises the same action (edits) as the Civic Duty badge. Additional details can be found about these badges, and others, on the SO website.[4] If more than one action is directly incentivised by the badge (e.g., for the Electorate badge), we model the combined activity by summing over the different action types. The interaction data was kindly supplied by SO in an anonymized form and it consists of the action counts per day of users on the website from January 2017 to April 2019.

Figure 3.2 presents the mean number of actions per day averaged across the entire user base for 70 days before and 70 days after the users achieved the badge. We plot only the actions that are directly incentivised by the badge. The steering effect, as described by Anderson et al. (2013) and Mutter and Kundisch (2014), can be seen

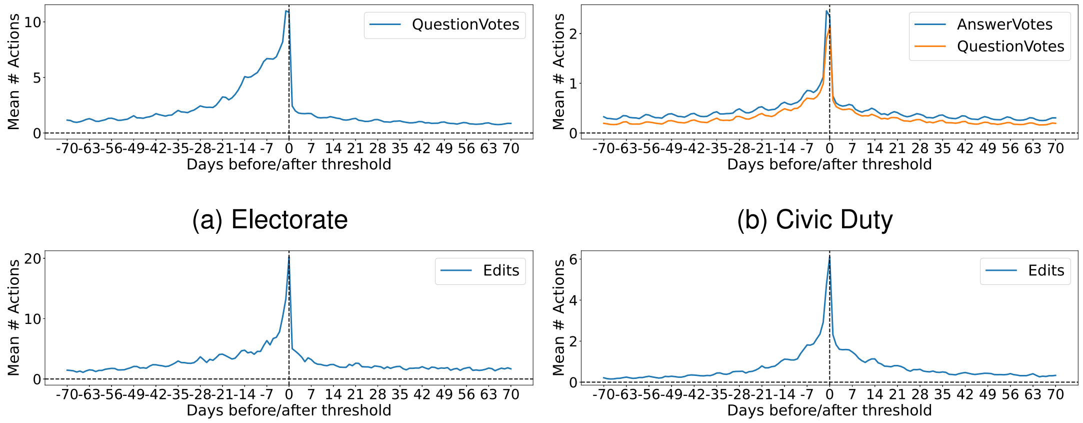
[3] `https://stackoverflow.com/help/privileges`

[4] `https://stackoverflow.com/help/badges`

Badge	Incentivised Action	Threshold	# Users
Electorate	Votes on Questions	600	5,701
Civic Duty	Votes on Questions and Answers	300	20,880
Copy Editor	Edits	500	750
Strunk & White	Edits	80	3,101

Table 3.2: Table detailing the badge rewards under study

by the increase in the rate of actions leading into the badge achievement date. After the badge has been achieved, the rate of activity rapidly drops and returns to a more constant rate of interaction (Anderson et al., 2013). The steering effect is most evident on the interaction data from the Electorate and Copy Editor users (Figure 3.2a) but the same general increase and then decrease can be seen in the trends from the other badges.



(a) Electorate

(b) Civic Duty

(c) Copy Editor

(d) Strunk & White

Figure 3.2: Plot of the mean count of actions per user per day 10 weeks before and 10 weeks after the users achieved the corresponding badges. Notice the different limits on the y-axis for the average number of actions that are performed.

Next, we consider four different reputation point thresholds that unlock different privileges on SO. Users achieve reputation points on SO by completing several different actions and critically by having other users validate their contributions. For example, users achieve reputation points by having their questions and answers upvoted, by having their answers accepted or by having their edits accepted. Table 3.3 details the different thresholds for gaining privileges that we study. We present: **Threshold** – the required number of reputation points that should be achieved to unlock the privilege; **# Users** – the number of users in our dataset that have achieved the privilege;

Threshold	# Users	Unlocked Privileges
1K	$71,795$	Established User: View the vote counts on posts.
2K	$29,161$	Edit Questions and Answers: Edits to posts are applied immediately without being reviewed.
20K	$1,316$	Trusted User: Expanding editing, deletion and un-deletion privileges.
25K	966	Access to Site Analytics: Access to internal and Google site analytics.

Table 3.3: Table detailing the reputation privileges under study

and, **Unlocked Privileges** – a brief description of the privileges that are unlocked on the website. Other reputation thresholds and their associated privileges can be found on the SO website.[5] The reputation data was obtained from the publicly available SO data dump[6] and it was filtered to users who joined SO after $2012/01/01$. We study the interaction data aggregated by week due to the sparsity of the actions through time.

Figure 3.3 shows the mean number of actions per week, averaged across users, for 20 weeks before and 20 weeks after crossing the defined reputation threshold. Differences in the rates of activity can be seen before and after the threshold was achieved; with a higher rate before the threshold and a lower rate after the threshold. Again, this appears to reflect the steering hypothesis — especially for the lower thresholds. Moreover, different behaviours around the different reputation thresholds are evident. The rates of answering are much lower for the lower thresholds than for the higher thresholds.

A further point of interest is evident in Figure 3.3b: The rate of editing from these users decreases to near 0 after the badge has been achieved. This is in comparison to the $1K$ threshold where the change in editing behaviour appears symmetric around the origin and the $20K$ and $25K$ thresholds where this rate is consistently low. A plausible reason for this is that once a user crosses the $2K$ reputation threshold, they no longer receive reputation points for editing posts.[7] This provides evidence that, for some users, the points that they receive for editing do serve to motivate their contributions.

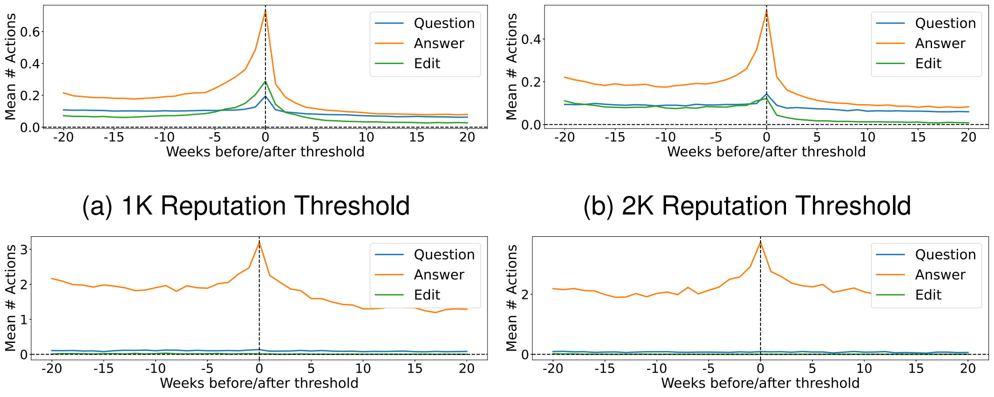
3.5 Empirical Study

We begin by detailing the evaluation criteria for comparing the models, and for selecting the most appropriate model for each domain. Thereafter, we compile the results

[5] https://stackoverflow.com/help/privileges/

[6] https://archive.org/details/stackexchange

[7] https://meta.stackexchange.com/questions/201728



(a) 1K Reputation Threshold

(b) 2K Reputation Threshold

(c) 20K Reputation Threshold

(d) 25K Reputation Threshold

Figure 3.3: Plot of the mean count of actions per user per day 10 weeks before and 10 weeks after the users achieved the corresponding reputation threshold.

from the models, for each of the domains, to investigate the conclusions that we can draw about steering in online settings.

3.5.1 Model Comparison

For all models, we report two measures of performance: The evidence lower bound (the ELBO), which is the lower-bound on the marginal log-likelihood of the data under the model assumptions (Kingma and Welling, 2014; Hoffman et al., 2013; Rezende and Mohamed, 2015); and the mean square error (MSE) of the model for reconstructing the original number of actions for each user. To compute the ELBO, we use the importance sampled weighted estimator (with K=10 samples) from Burda et al. (2015), shown to produce a tighter bound on the true negative log likelihood of the model. Parameter estimation is done in Pytorch and Adam is used to maximize the ELBO with an initial learning rate of 0.01 (Kingma and Ba, 2014). The learning rate was decreased with an exponential decay factor. We set the dimensionality of the latent space to $m := 10$.

We first report the results for the badge studies in Table 3.4. All models are trained on 60% of the data, with 20% of the data left for a validation set for model selection and 20% of the data is held out for a test set. Table 3.4 compares the performance of the models on the same test set (the standard deviation is in parentheses). The results from Table 3.4 show that Model 2 outperforms the other models achieving a higher bound on the marginal log-likelihood (ELBO) and a lower mean-squared reconstruction error on unseen data (MSE) on all instances except the Civic Duty (where it is still near-optimal) and on the MSE metric for the Strunk & White badge. Models 1, 2 and 3

Badge	Model 0		Model 1		Model 2		Model 3	
	ELBO	MSE	ELBO	MSE	ELBO	MSE	ELBO	MSE
Electorate	−256.94(6.95)	2881	−254.6(6.19)	2855	−235(7.18)	2629	−239.0(7.59)	2717
Civic Duty	−137.3(5.36)	798	−137.1(4.10)	794	−133.6(4.20)	761	−132.7(7.99)	754
Copy Editor	−392.2(7.96)	10122	−409.1(6.86)	10003	−385.2(9.85)	9951	−408.9(5.28)	10609
Strunk & White	−120.0(5.14)	669	−119.4(4.39)	655.1	−118.7(4.34)	654	−119.1(5.97)	651

Table 3.4: ELBO and MSE on held out data for badge study.

all outperform Model 0, suggesting that the inclusion of the steering parameter β does increase the probability of the activity data. Similarly, Models 2 and 3 outperform Model 1 which suggests that the steering strength parameter, S_u, is a useful way to segment the population of users. However, the additional complexity of Model 3 does not appear to help the model in better describing the data. While the results do suggest that models 2 and 3 do outperform models 0 and 1, the comparison of models 2 to 3 is not robust. The standard errors overlap and thus this interpretation could be due to the experimental setup. It is entirely plausible that the goal gradient effect is not the only factor influencing the behaviour of people and thus further investigate is required to reach stronger conclusions.

Table 3.5 presents the results for the reputation thresholds study. We use the same 60%, 20% and 20% splits for the train, validation and test sets respectively. Due to the large data sizes for the $1K$ and $2K$ thresholds, we limit the data to a maximum of $10K$ users for each of the splits. Similar to the badge study, we report the ELBO and MSE on the held-out test set. We also extend Model 2 to allow for an additional response to the reward that might be present in the data: As a user unlocks a privilege, she might choose to interact more on the website to explore the newly available features (Chou, 2019). Thus this model has an additional steering parameter, β^3, that is restricted to be $\mathbf{0}$ before the threshold is reached and non-negative after the threshold is reached. The other models remain the same as those used in the badge study.

In general, the Models 2 and 3 do outperform Models 0 and 1; however, the differences in their performance is less pronounced than those observed in the badge study. We also note that the standard error suggests the results could be due to chance. While we are led towards the same conclusion as above that the steering parameter S_u plays an important role in segmenting the behaviour of the users, we do note that the simpler models still capture the interaction dynamics well which suggests a more homogeneous set of reactions to the threshold. We choose Model 2 as the simplest model that describes the data for these three thresholds (it is optimal for the $1K$, $2K$ and $25K$

Threshold	Model 0		Model 1		Model 2		Model 3	
	ELBO	MSE	ELBO	MSE	ELBO	MSE	ELBO	MSE
1K	−20.3(1.14)	90	−20.2(1.24)	88	−19.9(1.40)	87	−20.0(1.24)	87
2K	−19.4(1.39)	53	−19.4(1.05)	52	−19.2(1.18)	50	−19.0(1.26)	50
20K	−63.1(4.70)	315	−57.4(2.03)	263	−56.1(2.84)	260	−58.6(3.22)	268
25K	−63.2(1.50)	417	−71.4(1.98)	511	−62.6(1.56)	435	−63.4(2.71)	455

Table 3.5: ELBO and MSE on held out data for reputation study.

thresholds and it is near-optimal for the $2K$). In all cases, it is important to note how well Model 0 performs, suggesting that many users do not, in fact, deviate from their preferred distributions for interaction and the null hypothesis (that steering does not occur) is a broadly good hypothesis for these reputation point threshold domains.

3.5.2 Analysis of Steering

This section studies separately the steering effect that is inferred by Model 2 on the Electorate badge population (Section 3.5.2.1) and the effect inferred by Model 2 on the $1K$ reputation threshold population (Section 3.5.2.2). Although we study in detail only the Electorate badge, in particular, the conclusions that are reached for the other badges are similar and thus we omit them for clarity; replicated plots for these domains can be found in Appendix B.2. Similarly, our focus below is on the $1K$ threshold for the reputation study. There are some subtleties regarding the behaviour of the SO users when they cross the different thresholds; most notably, the user behaviour around the higher thresholds appears to be different to that when they cross the lower thresholds. When discussing these results, we note when the activity around a specific threshold departs from the general trend that we observe. As with the badge study, the replicated plots on the other datasets are available in Appendix B.3.

3.5.2.1 Analysis of Steering Towards Badges

We analyse the inferred parameters from Model 2 on the Electorate dataset to make conclusions about how people steer towards badges. Model 2 allows for four different types of users:

Type 1 (Non-Steerers): Users who do not deviate from their preferred distribution. In this case $S_u = (0,0)$ and there is no effect of β on their activity.

Type 2 (Strong-and-Steady): Users who experience the full adherence to β_λ on their activity parameter λ_u but do not change how often they interact on the platform (e.g., in a given day, they will complete more work but they do not work on more days). In this case $S_u = (0,1)$.

Type 3 (Dropouts): Users who appear to work on more days before the badge has been achieved and on fewer days after the badge has been achieved, thereby experiencing the effect of β_α. They do not appear to change the number of actions that they will perform on a given day. In this case $S_u = (1,0)$.

Type 4 (Strong-Steerers): Users who adhere to the full steering effect described by $\beta = (\beta_\alpha, \beta_\lambda)$, both on how often they act on the platform and on how many actions they are likely to perform on any given day. In this case $S_u = (1,1)$.

Figure 3.4 presents the inferred assignment of users to the four user types (when considering the entire dataset). We can see that the most common assignment type is Non-Steerers making up 63.2% (3602 users) of the user base. The next most common type is the Strong-and-Steady group (19.8%; 1130 users) followed by the Strong-Steerers (13.5%; 772 users) and finally the Dropouts (3.5%; 197 users). A key finding is that the largest group that is inferred in the data does not appear to respond to the badge incentives in a way that has been predicted by previous works (Anderson et al., 2013; Mutter and Kundisch, 2014; Yanovsky et al., 2019). We highlight the fact that this Non-Steerer population is twice as large as the Strong-Steerer and Strong-and-Steady groups together! While these "steering groups" form a smaller population of users, it is the highly engaged interactions from these users that drive the aggregated trends that we notice in Figure 3.2a.

We demonstrate the markedly different behaviour of the users from each group by presenting samples from their interaction data, along with the models' reconstruction of their activity. Figure 3.5 shows 10 random samples from these users who achieved the Electorate badge for each of the 4 user types. The plots show the true count of actions as a function of time alongside the expected number of actions under the assumptions of Model 2. The black vertical line, on day 0, corresponds to the day that the user achieved the Electorate badge. The left-most column of Figure 3.5 presents samples from the Type 1 (Non-Steerer) population. The counts of actions appear to show no change around day 0; these users appear not to change their behaviour in the presence of the badge. This is in stark contrast to all the other columns where there does appear to be a change around day 0. On the right-hand column, we present samples from the Type 4 (Strong-Steerer) population of user. It is important to note the high number of actions (both expected and true) before day 0 when the badge was achieved. After day

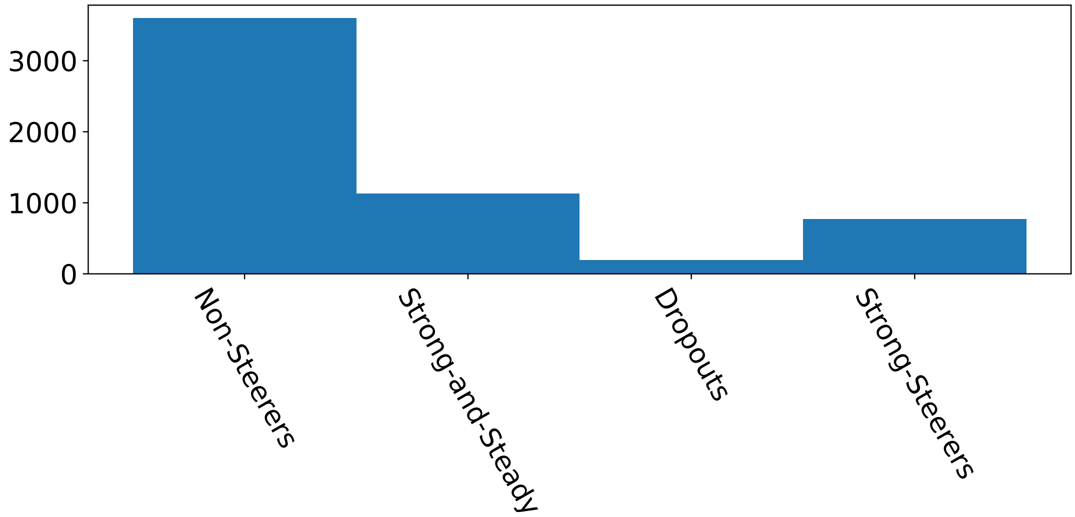


Figure 3.4: Cluster assignments (as inferred by S_u from Model 2) for the users who achieved the Electorate badge.

0, both the true and expected numbers of actions drops dramatically. The centre-left column of Figure 3.5 presents samples from the Type 2 (Strong-and-Steady) population. These users appear to increase the number of actions that they perform on a day leading into the badge achievement. They continue to work even after the badge has been achieved but at a reduced rate. This suggests that they have other reasons than merely the badge, for contributing to SO. Finally, the centre-right column of the plot shows samples from Type 3 (Dropouts). These users appear to hold a steady (and low) rate of interaction leading to the badge achievement followed by a decrease in their rate of activity after the badge is achieved.

Figure 3.6 shows the effect of steering on users, plotting β as a function of time. The magnitude of the values of β indicates direct changes to the probability that the user is active, as well as expected changes in the number of actions on a given day. In accordance with related work, and the predictions of the goal gradient hypothesis, users increase their rate of activity as they approach the day upon which they achieved the badge (Anderson et al., 2013; Bornfeld and Rafaeli, 2017; Mutter and Kundisch, 2014).

A novel insight from our model is that the β_α parameter, affecting both the Strong-Steerer and the Dropout groups, decreases well below 0 after the badge has been achieved. That is, users may decrease their activity well beyond their preferred distribution after they have achieved the badge. This result suggests that some of the users who are steered strongly may stop contributing altogether once the badge has

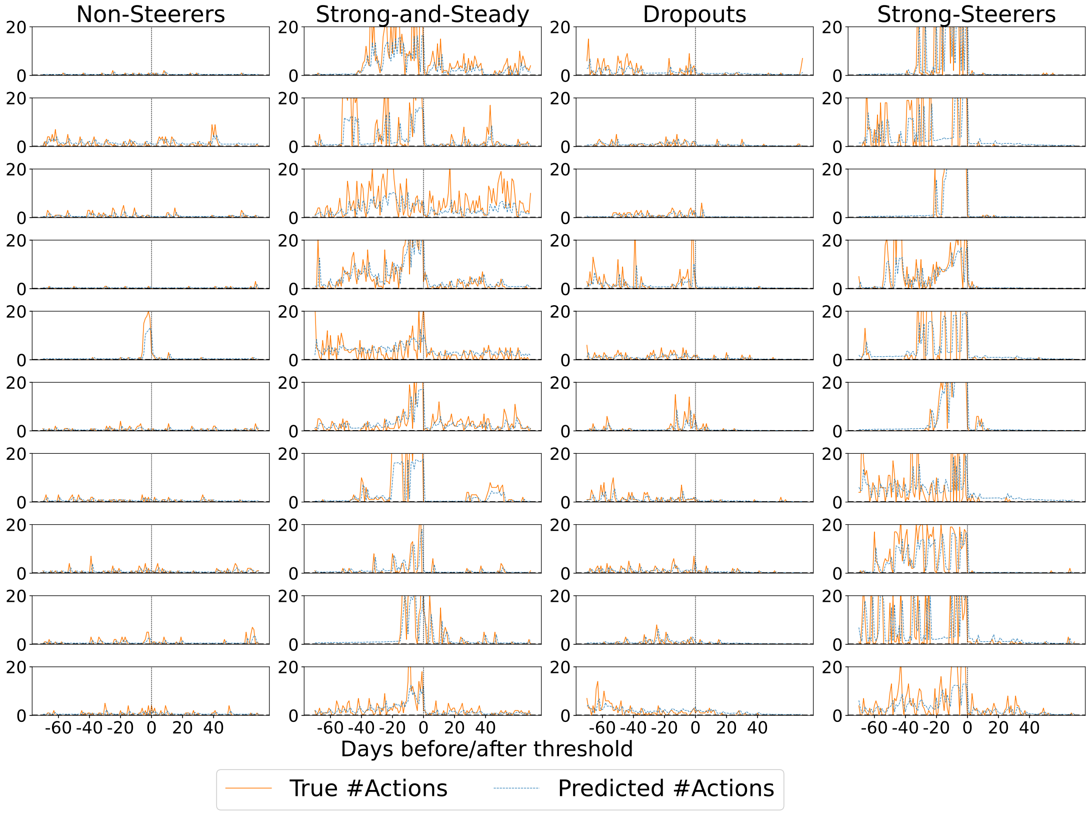


Figure 3.5: 10 samples of users' interaction data, with the corresponding model reconstructions, for each type of user as inferred by Model 2. The left column is the Non-Steerer group who appear to show no change in their behaviour around the badge achievement. Center-left is the Strong-and-Steady group that increase the number of actions they perform in a given day before achieving the badge. These users mainly continue to interact even after the badge has been achieved. Center-right presents samples from the Dropout users who appear to decrease their activity after achieving the badge. The right column presents the Strong-Steerer population who increase their rate of activity strongly before achieving the badge but decrease their activity rate to near 0 after the badge is achieved.

been achieved. This would align with a utility theoretic model of the behaviour where all the utility of the badge is conveyed upon receipt of the badge and thus there is no reason to continue to contribute (Immorlica et al., 2015). This does not hold for all of the users as evidenced by the comparatively large size of the Strong-and-Steady population.

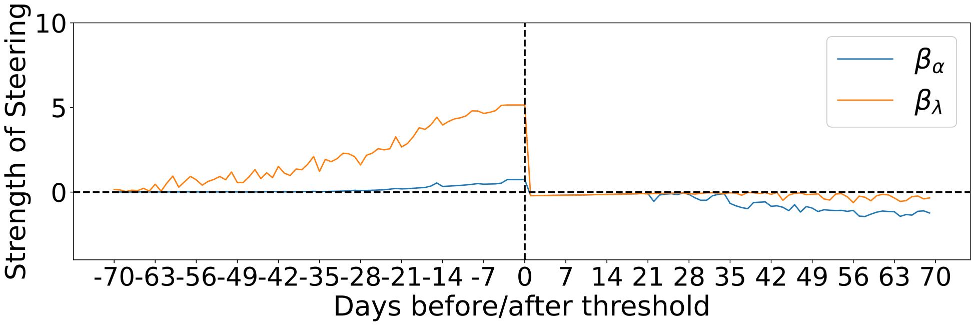


Figure 3.6: Plot of the inferred magnitude of β: the expected deviation from a user's preferred distribution P_u under the assumptions of Model 2.

Figure 3.7 presents the mean number of interactions per user as a function of the number of days until/after the badge is achieved. The four lines correspond to the four groups that are inferred by Model 2. The mean interaction rates of these groups show the vastly different behaviours that are described above. In particular, we make the comparison of this plot to that in Figure 3.2a. We can see that the steering behaviour that is evident in Figure 3.2a is mainly driven by the Strong-Steer and the Strong-and-Steady groups (together accounting for 33.36% of the population). Notice that the mean interaction count from the Strong-Steerer and Dropout groups drops past the other groups to close to 0 after they achieve the badge. Of interest is the Strong-and-Steady group (13.6%) who act exactly as Anderson et al. (2013) describes in that they return to a baseline level of work and continue to interact after the badge has been achieved. The thin dotted line for the Dropout user group is used to indicated that this group consists of less than 5% of the user base.

The Non-Steered population (63.2%) show no change in their interaction rates before or after the receipt of the badge. There is a distinct uptick in the mean number of question-vote actions on the day before and on the day of the badge achievement (Figure 3.7, blue line). It is possible that this “bump” might mistakenly be seen as the response of the users to the badge incentive. In fact, this bump is an artefact of the analysis technique which centres trajectories around a threshold that is crossed by the cumulative sum of the trajectory entries (see Section 3.6 and Appendix B.1 for a

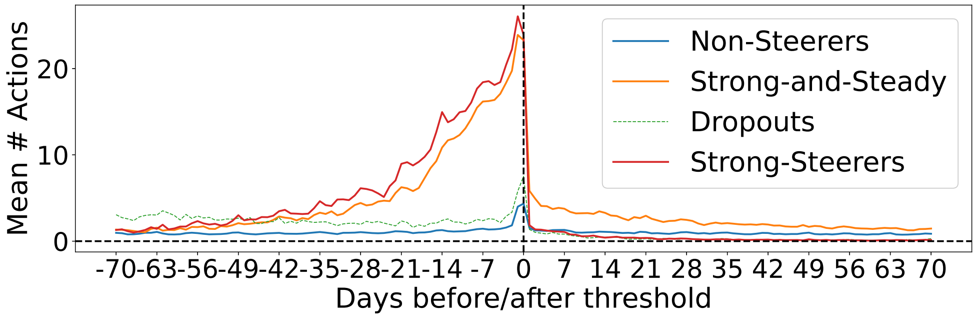


Figure 3.7: Mean number of actions per day for users who are classified by their steering parameters (S_u). The thin dotted line for the Dropout user group indicates that this group consists of less than 5% of the user base.

discussion and proof of this claim).

3.5.2.2 Analysis of Steering Towards Reputation Points

In studying the response of the users to the reputation thresholds, we use the same grouping as that introduced above for the analysis of the Electorate badge. Figure 3.8 shows that the Non-Steerer population is again the dominant group that is inferred in this reputation threshold dataset. These users account for approximately 96.0% (68,941 users) of the user base whereas the Strong-and-Steady, the Dropouts and the Strong-Steerers only account for 1.6% (1146), 0.04% (34) and 2.3% (1674) respectively. The inferred fraction of Non-Steerers for the reputation thresholds is, therefore, greater than what is inferred for the badges thresholds. This holds for all the reputation thresholds and badges that we study in Appendices B.2 and B.3.

Figure 3.9 shows the mean plot of activity for the groups, as inferred by the S_u variable. The Non-Steering group is striking in that it shows the same low activity rates as those observed in Section 3.5.2.1 but for an even larger fraction of the population. The general trend that we observed in Figure 3.3a is driven by the $< 5\%$ of the population who appear to respond to the badge. The Strong-and-Steady group shows the steering effect by a rapid increase in actions into the goal achievement, followed by a return to their base level of interaction. The thin dotted lines in the plot emphasise that each of these groups consist of less than 5% of the users who achieved the $1K$ threshold.

The Strong-Steerer group that was inferred for the $25K$ threshold consisted of 5.3% of the population with the Strong-and-Steady accounting for 4.0% (Figures B.16 and B.17 where the line for the Strong-Steering group is dark to reflect this). The higher portion of steerers for this population could be due to the lack of further thresh-

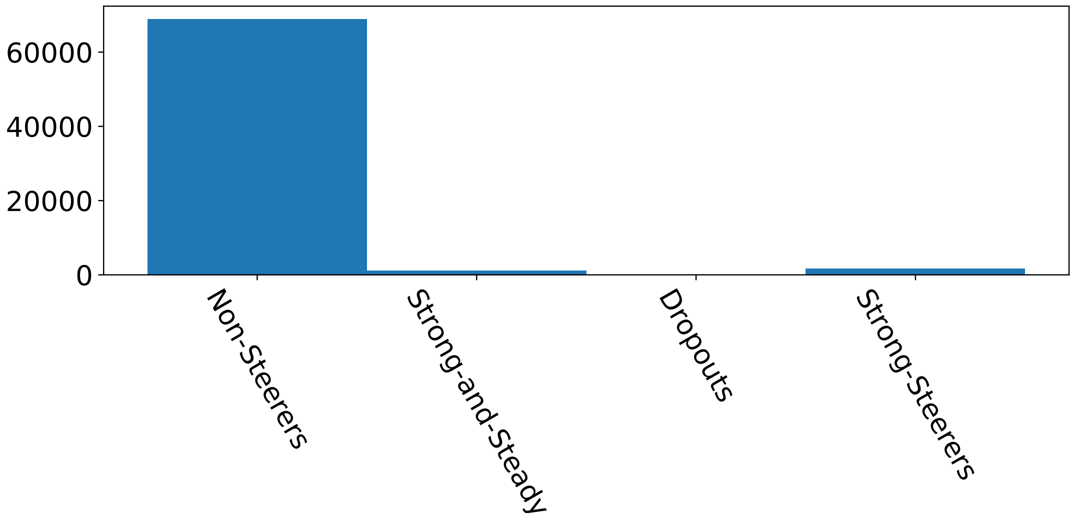


Figure 3.8: Cluster assignments, as inferred by S_u from Model 2, for the users who achieved the 1K reputation point threshold.

olds/privileges but we also note that the sample size for this population is the smallest and thus it could be due to the small sample who achieved this threshold.

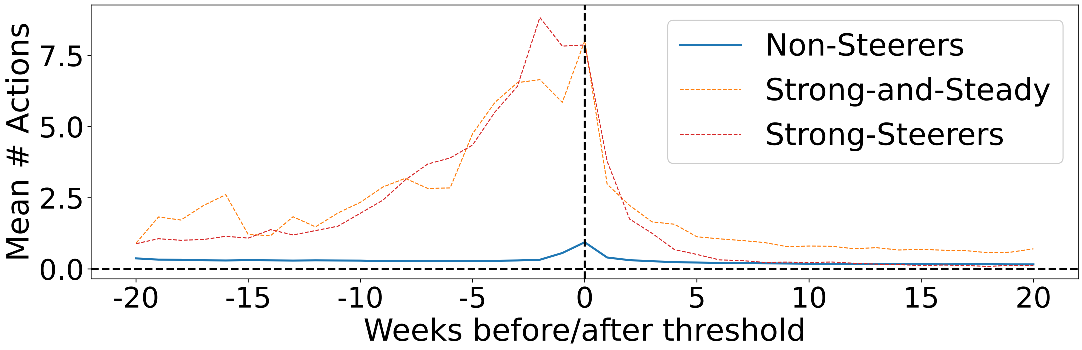


Figure 3.9: Mean number of actions per day for users who are classified by their steering parameters (S_u) for the users who passed the $1K$ reputation threshold.

3.5.3 Limitations of Empirical Study

The empirical study of steering that is presented in this section has several limitations which we list here. We only studying 4 of the threshold badges, 4 reputation thresholds and the study is limited to studying user behaviour on one platform: SO.

There are alternative types of badges that are present on SO. For example, the

Famous Question badge[8] is awarded to a question that gets $10,000$ views. It is not clear how users can "work towards" a "qualitative" badge of this nature and thus our study does not extend to badges of this type.

Secondly, we have focused our study on only 4 out of the total 26 privilege thresholds that SO defines. Our overarching conclusion is that steering is a rare phenomenon in these settings but there may be a threshold where users have exhibited a greater steering effect than what we observed. Our choice of $threshold \in \{1K, 2K, 20K, 25K\}$ was motivated by the fact that the $1K$, $20K$ and $25K$ thresholds are three out of the five "milestone" thresholds on SO. Moreover, the $2K$ threshold provides a very well defined privilege that may have resulted in a behaviour change (as noted in Section 3.4).

A final limitation is that the study was conducted only on SO data. While the goal gradient effect has been documented in many different domains Hull (1932); Kivetz et al. (2006), and steering has even been noticed on other question and answer platforms (Mutter and Kundisch, 2014; Yanovsky et al., 2019; Bornfeld and Rafaeli, 2017), our results are limited to the behaviour of users on the SO platform.

3.6 Proving the Existence of Phantom Steering

The population of non-steerers in Figures 3.7 and 3.9 display a sharp uptick in the mean of their action counts on the day before and on the day of the badge achievement. We prove that such a bump arises as an artefact of centring the data on day 0, and it is therefore expected to arise even in the absence of a steering effect. We show this "phantom steering" bump occurs in the setting of Model 0 (Figure 3.1) where daily action counts are independent draws from some unchanging latent distribution. Our proof (and the intuition arising from it) suggests that a similar bump arises in the presence of steering as well. This bump may have served to inflate previous conclusions about how users change their behaviour when working to achieve badges (Anderson et al., 2013; Yanovsky et al., 2019; Mutter and Kundisch, 2014).

For users acting under Model 0 we present Theorem 3.6.1, which implies that for sufficiently large badge thresholds the expected number of actions on day 0 (the day of badge achievement) is greater than the expected number of actions on any other day.

We introduce this theorem via the following intuitive example: Suppose that the badge threshold N is chosen randomly from some large range $N \in [m, M]$ of possible action counts. Let S_n be the cumulative number of actions from a user up to (and

[8] https://stackoverflow.com/help/badges/28/famous-question

including) day n. As long as the user continues to act on the platform, S_n will eventually traverse the interval $[m, M]$. Moreover, as the count of actions on any day n is a random variable (drawn from the user's preferred distribution), S_n is more likely to cross the threshold N on a day on which the user makes relatively more contributions. This claim holds even when actions are drawn under the no-steering assumptions of Model 0 which assumes that users' action counts on each day are independent draws from their preferred distribution P_u (which is not influenced by steering).

We formalize this intuition in Theorem 3.6.1, the proof of which appears in Appendix B.1. Recall that the random variable A_u^0 describes the number of actions that user u performs on the day that they receive the badge. Denote the number of actions required to achieve the badge by N, and let $A_{u,N}^0$ denote this random variable when the badge threshold is N actions and user u acts according to Model 0.

Theorem 3.6.1. *If P_u is bounded then:*

$$\lim_{N\to\infty} \mathbb{E}[A_{u,N}^0] = \mathbb{E}[P_u] + \frac{Var[P_u]}{\mathbb{E}[P_u]}.$$

This expected bump size holds in the limit as the badge threshold becomes large with respect to the mean of P_u. For fixed P_u the convergence to this limit is exponential in the threshold.

3.7 User Survey

As an additional form of validation for the analytical results that are presented in this paper, we hosted a survey that recruited participants from SO to answer questions relating to their motivations for contributing to the website. A clickable advertisement was placed on SO and willing participants were directed to the survey. We paid each survey participant \$10 in an Amazon gift voucher for completing the survey. In total, we received 86 responses from the community. We rejected 16 of these responses as the account IDs that were associated with these users did not exist or the users had not contributed to SO, making them not part of the target population. This left 70 valid survey responses.

Figure 3.10 summarizes the responses to the question: "What are your reasons for participating in SO?" The majority of users claimed to have personal and/or altruistic reasons for contribution to the website with 87.1% claiming to contribute to increasing their own knowledge (and 68.6% claiming to want to "contribute to the community").

In contrast to this, only 24.2% of the users selected the reason to "achieve badges". 50% of users claimed that they had a goal of increasing their reputation score.

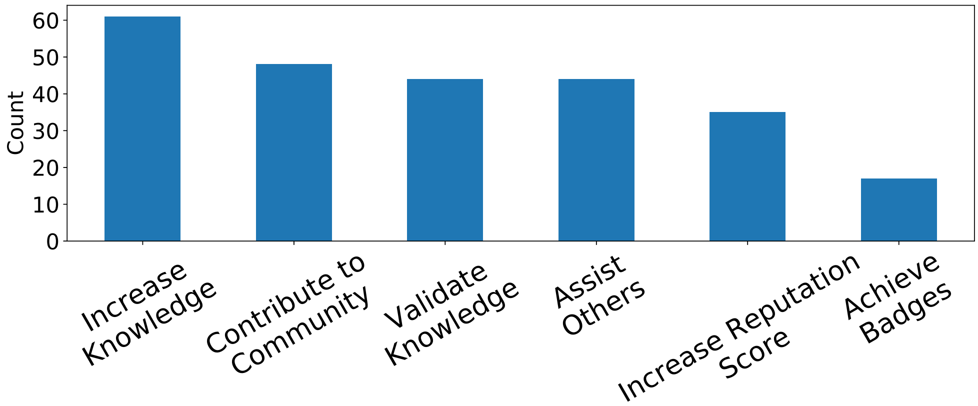


Figure 3.10: Counts of responses to the reasons for contributing to Stack Overflow.

We also asked the users specifically about their voting contributions: "What motivates you to vote on other people's posts?" The responses to this question are summarised in Figure 3.11. Participants could select any combination of three different reasons for voting: badge acquisition ("I wished to achieve one of the voting badges: e.g., Supporter, Critic, Suffrage, Vox Populi, Civic Duty or Electorate"); altruism ("I think it is important to provide feedback about other's work"), or "other". Only 12.9% of participants who engage in voting actions reported badge acquisition is a motivating factor for their work. (Eight of the participants in the study claimed to not engage in voting actions and were not counted.)

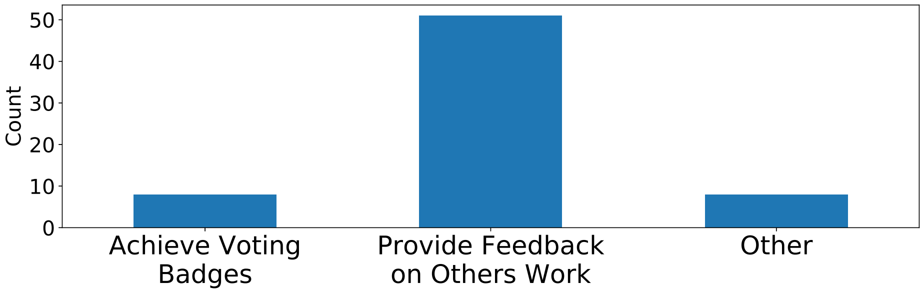


Figure 3.11: Counts of responses to the reasons for voting in Stack Overflow.

Together these results present further evidence to corroborate the model predictions that only a minority of the SO participants are indeed steered by badges.

A surprising result, and one that stands in contradiction to the computational results presented in Section 3.5 is shown in Figure 3.12. Participants were asked if

"Privileges that are associated with a high reputation score motivate [them] to achieve a higher score?" The overwhelming response from the surveyed population was that these privileges did motivate the users, however, our results from Section 3.5 suggest that the steering hypothesis is weaker in this setting than in the badge setting (where the reward is more explicit).

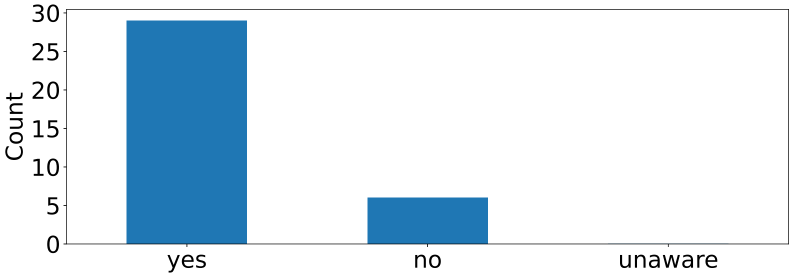


Figure 3.12: Surveyed users' answers to the question: "Do the privileges that are associated with a high reputation score motivate you to achieve a higher score?"

We note the limitation of possible sample bias in the self-reported survey. A clickable advertisement was placed on the SO website and from there users opted-in to completing the survey. It is possible that the users who choose to complete such a survey have a biased perspective toward the rewards on SO. These biases would then show in our results. Moreover, we only had 70 users complete the survey and thus this represents a very small sample from the SO user base.

3.8 Conclusion

We have presented a novel probabilistic model that describes how users interact on the SO platform and in particular how these users respond to badge incentives and to the reputation thresholds that unlock new privileges on the website. We demonstrated how this model can be fit to the data that is provided by SO and we investigated the distribution that is learnt over the latent space that describes the "steering effect".

Our results provide a more informed understanding of how users respond to badges in online communities. First, that some users do exhibit steering supports the claims made by previous work. These users comprise approximately 30% of the users for the badge studies and approximately 5% of the users for the reputation threshold studies. The users in this group significantly increase their levels of activity leading into the day when they achieve the goal. Some of the users, the "Strong-and-Steady" group, continue to interact at a base rate well after achieving the goal. This behaviour is well documented by previous work (Anderson et al., 2013; Mutter and Kundisch, 2014; Yanovsky et al., 2019; Li et al., 2012). However, other users, the “Strong-Steerers” and the “Dropouts”, actually decrease their activity rates, well below any previous level of activity, once the goal has been achieved. It is possible that assigning additional badges, with thresholds beyond those already in place in SO will continue to motivate such users.

Second, we identify the presence of a large population of users, approximately 60% of the population for the badge study, who do not exhibit steering. In the case of the reputation point thresholds, our results suggest that approximately 90% of the population does not exhibit steering. These “Non-Steerer” users do not appear to change the rate of their activity for the period under study (20 weeks for the badge studies and 40 weeks for the reputation points). Rather, they continue to act at the same rate well after the goal has been achieved. This suggests that these users have reasons for performing actions on SO which do not include specific receipt of the badge or privilege reward.

Third, any analysis of user behaviour around a goal must take into account the presence of the phantom steering bump which has not previously been acknowledged in the context of badges. This statistical artefact is model-independent and may lead to inflated conclusions about the effect of badges on users’ behaviour.

Future work will extend the models of Section 3.2 to study the feedback mechanisms on Q&A websites such as SO. While our empirical results suggest a limited effect of the reputation privileges, our survey results suggest that the reputation points and the user-generated feedback that drives this system remains an important factor in motivating further contributions from the community. We believe that this relationship might depend on a tight feedback loop from action to response (vote or accept) and back to action. For example, a user who answers many questions, and receives social validation from many “upvotes” and “accepts” (leading to reputation points), might experience an increased drive to continue interacting on the website. The model that is presented in this paper can form the foundations for this work in that the β parameters (i.e., the generic response to rewards) can be adapted to rather model this point process style of feedback data. It will allow more detailed inference into who is motivated by the current feedback mechanisms and it will provide insight into how the feedback influences the behaviour of the users on such platforms.

Chapter 4

MultiplexNet: Towards Fully Satisfied Logical Constraints in Neural Networks

4.1 Introduction

An emerging theme in the development of deep learning is to provide expressive tools that allow domain experts to encode their prior knowledge into the training of neural networks. For example, in a manufacturing setting, we may wish to encode that an actuator for a robotic arm does not exceed some threshold (e.g., causing the arm to move at a hazardous speed). Another example is a self-driving car, where a controller should be known to operate within a predefined set of constraints (e.g., the car should always stop completely at a stop street). In such *safety critical* domains, machine learning solutions must guarantee to operate within distinct boundaries that are specified by experts (Amodei et al., 2016).

One possible solution is to encode the relevant domain knowledge directly into a network's architecture which may require non-trivial and/or domain-specific engineering (Goodfellow et al., 2016). An alternative approach is to express domain knowledge as logical constraints which can then be used to train neural networks (Xu et al., 2018; Fischer et al., 2019; Allen et al., 2020). These approaches compile the constraints into the loss function of the network, by quantifying the extent to which the output of the network violates the constraints. This is appealing as logical constraints are easy to elicit from people. However, the solution outputted by the network is designed to minimize the loss function — which combines both data and constraints — rather than to guarantee the satisfaction of the domain constraints. Thus, representing constraints in the loss function is not suitable for safety-critical domains where 100% constraint satisfaction is desirable.

Safety-critical settings are not the only application for domain constraints. Another common problem in the training of large networks is that of data inefficiency. Deep models have shown unprecedented performance on a wide variety of tasks but these come at the cost of large data requirements.[1] We propose that, for tasks where domain knowledge exists, we can also use this knowledge to structure a network's training to reduce the data burden that is placed on the learning process.

This paper directly addresses both of these challenges by providing a new way of representing domain constraints directly in the output layer of a network. The proposed approach represents domain knowledge as a logical formula in disjunctive normal form (DNF). It augments the output layer of an existing neural network to include a separate

[1] For instance OpenAI's GPT-3 (Brown et al., 2020) was trained on about 500 billion tokens and ImageNet-21k, used to train the ViT network (Dosovitskiy et al., 2020), consists of 14 million images.

transformation for each term in the DNF formula. We introduce a latent categorical variable that selects the best transformation that optimizes the loss function of the data. In this way, we can represent arbitrarily complex domain constraints in an automated manner, and we are also able to guarantee that the output of the network satisfies the specified constraints.

We show the efficacy of this MultiplexNet approach in three distinct experiments. First, we present a density estimation task on synthetic data. It is a common goal in machine learning to draw samples from a target distribution, and deep generative models have shown to be flexible and powerful tools for solving this problem. We show that by including domain knowledge, a model can learn to approximate an unknown distribution on fewer samples, and the model will (by construction) only produce samples that satisfy the domain constraints. This experiment speaks to both the data efficiency and the guaranteed constraint satisfaction desiderata. Second, we present an experiment on the popular MNIST data set (LeCun et al., 2010) which combines structured data with domain knowledge. We structure the digits in a similar manner to the MNIST experiment from Manhaeve et al. (2018); however, we train the network in an entirely label-free manner (Stewart and Ermon, 2017). In our third experiment, we apply our approach to the well-known image classification task on the CIFAR100 data set (Krizhevsky et al., 2009). Images are clustered according to "super classes" (e.g., both *maple tree* and *oak tree* fall under the super class *tree*). We follow the example of Fischer et al. (2019) and show that by including the knowledge that images within a superclass are related, we can increase the classification accuracy at the superclass level.

The chapter contributes a novel and general way to integrate domain knowledge in the form of a logical specification, into the training of neural networks. We show that domain knowledge may be used to restrict the network's operating domain such that any output is guaranteed to satisfy the constraints; and in certain cases, the domain knowledge can help to train the networks on fewer samples of data.

4.1.1 Contributions

In the work that follows, I was the main contributor to this project. I designed the experiments and implemented the algorithms for the experiment. For project ideation and technical input on the logical constraints, I think my co-author Vaishak Belle and as usual, thank you to Kobi Gal for his overall insights and writing expertise.

4.1.2 Problem Specification

We consider a data set of N i.i.d. samples from a hybrid (some mixture of discrete and/or continuous variables) probability density (Belle et al., 2015). Moreover, we assume that: (1) the data set was generated by some random process $p^*(x)$; and (2) there exists domain or expert knowledge, in the form of a logical formula Φ, about the random process $p^*(x)$ that can express the domain where $p^*(x)$ is feasible (non-zero). Both of these assumptions are summarised in Eq. 4.1. In Eq. 4.1, the notation $x \models \Phi$, denotes that the sample x satisfies the formula Φ (Barrett et al., 2009). For example, if $\Phi := x > 3.5 \wedge y > 0$, and given some sample $(x,y) = (5,2)$, we denote: $(x,y) \models \Phi$.

$$x \sim p^*(x) \implies x \models \Phi \tag{4.1}$$

We aim to approximate $p^*(x)$ with some parametric model $p_\theta(x)$ and to incorporate the domain knowledge Φ into the maximum likelihood estimation of θ, on the available data set.

Given knowledge of the constraints Φ, we are interested in ways of integrating these constraints into the training of a network that approximates $p^*(x)$. We desire an algorithm that does not require novel engineering to solve a reparameterisation of the network and moreover, especially salient for safety-critical domains, any sample x from the model, $x \sim p_\theta(x)$, should imply the constraints are satisfied. This is an especially important aspect to consider when comparing this method to alternative approaches, namely Fischer et al. (2019) and Xu et al. (2018), that do not give this same guarantee.

4.2 Incorporating Domain Constraints into Model Design

We begin by describing how a satisfiability problem can be hardcoded into the output of a network. We then present how any specification of knowledge can be compiled into a form that permits this encoding. An overview of the proposed architecture with a general algorithm that details how to incorporate domain constraints into training a network can be found in Appendix 4.2.3.

4.2.1 Satisfiability as Reparameterisation

Let $\tilde{x}$ denote the unconstrained output of a network. Let g be a network activation that is element-wise non-negative (for example an exponential function, or a ReLU (Nair and Hinton, 2010) or Softplus (Dugas et al., 2001) layer). If the property to be encoded is a simple inequality $\Phi : \forall x\, cx \geq b$, it is sufficient to constrain $\tilde{x}$ to be non-negative by applying g and thereafter applying a linear transformation f such that: $\forall \tilde{x} : cf(g(\tilde{x})) \geq b$. In this case, f can implement the transformation $f(z) = sgn(c)z + \frac{b}{c}$ where sgn is the operator that returns the sign of c. By construction we have:

$$f(g(\tilde{x})) \models \Phi \tag{4.2}$$

It follows that more complex conjunctions of constraints can be encoded by composing transformations of the form presented in Eq. 4.2. We present below a few examples to demonstrate how this can be achieved for a number of common constraints (where $\tilde{x}$ always refers to the unconstrained output of the network):

$$a < x < b \;\rightarrow\; x = -g(-g(\tilde{x}) + k(a,b)) + b \tag{4.3}$$

$$x = c \;\rightarrow\; x = c \tag{4.4}$$

$$x_2 > h(x_1) \;\rightarrow\; x_1 = \tilde{x_1}\;;\; x_2 = h(x_1) + g(\tilde{x_2}) \tag{4.5}$$

In Eq 4.3, we introduce the function $k(a,b)$. This is merely a function to compute the correct offset for a given activation g. In the case of the Softplus function, which is the function used in all of our experiments, $k(a,b) = log(exp(b-a)-1)$.

In Section: Experiments, we implement three varied experiments that demonstrate how complex constraints can be constructed from this basic primitive in Eq. 4.2. Conceptually, appending additional conjunctions to Φ serves to restrict the space that the output can represent. However, in many situations, domain knowledge will consist of complicated formulae that exist well beyond mere conjunctions of inequalities.

While conjunctions serve to restrict the space permitted by the network's output, disjunctions serve to increase the permissible space. For two terms ϕ_1 and ϕ_2 in $\phi_1 \vee \phi_2$ there exist three possibilities: namely, that $x \models \phi_1$ or $x \models \phi_2$ or $(x \models \phi_1) \wedge (x \models \phi_2)$. Given the fact that any unconstrained network output can be transformed to satisfy some term ϕ_k, we propose to introduce multiple transformations of a network's unconstrained output, each to model the different terms ϕ_k. In this sense, the network's output layer can be viewed as a multiplexor in a logical circuit that permits for a branching of logic. If $h_1(\tilde{x})$ represents the transformation of $\tilde{x}$ that satisfies ϕ_1 and $h_2(\tilde{x}) \models \phi_2$ then we know the output must also satisfy $\phi_1 \vee \phi_2$ by choosing either h_1 or h_2. It is this branching technique for dealing with disjunctions that gives rise to the name of the approach: MultiplexNet.

We finally turn to the desideratum of allowing any Boolean formula over linear inequalities as the input for the domain constraints. The suggested approach can represent conjunctions of constraints and disjunctions between these conjunctive terms, which is exactly a DNF representation. Thus, the approach can be used with any transformed version of Φ that is in DNF (Darwiche and Marquis, 2002). We propose to use an off-the-shelf solver, e.g., Z3 (De Moura and Bjørner, 2008), to provide the logical input to the algorithm that is in DNF. We thus assume the domain knowledge Φ is expressed as:

$$\Phi = \phi_1 \vee \phi_2 \vee \ldots \vee \phi_k \tag{4.6}$$

If h_k is the branch of MultiplexNet that ensures the output of the network $x \models \phi_k$ then it follows by construction that $h_k(\tilde{x}) \models \Phi$ for all $k \in [1,\ldots,K]$. For example, consider a network with a single real-valued output $\tilde{x} \in \mathbb{R}$. If the knowledge $\Phi :=$ $(x \geq 2) \vee (x \leq -2)$, we would then have the two terms $h_1(\tilde{x}) = g(\tilde{x}) + 2$ and $h_2(\tilde{x}) =$ $-g(-\tilde{x}) - 2$. Here, g is the network activation that is element-wise non-negative that was referred to in Section: Satisfiability as Reparameterisation. It is clear that both $x_1 = h_1(\tilde{x})$ and $x_2 = h_2(\tilde{x})$ satisfy the formula Φ.

It is worth considering the case where two (or more) constraint terms overlap in the output domain of the network. For example, if the logic is $\Phi := (x \geq 2) \vee (x \leq 3)$ we have the case where both $\phi_1 = (x \geq 2)$ and $\phi_2 = (x \leq 3)$ could be *true* (namely if $x \in [2,3]$). In this case, it still holds that MultiplexNet could choose either h_1 (the transformation corresponding to ϕ_1) or h_2 (the transformation corresponding to ϕ_2). This is important, because, for a formula with K terms, there only needs to be a choice of K options (and not $K^2 - 1$ in the event that combinations of branches were necessary). Moreover, in this example, we can further see that Φ itself could be simplified to $\Phi := x$ (specifying that there is no transformation necessary as there is no constraint). Although we do not implement the compilation of the logic in this manner, we can reasonably expect a compiler to make this form of simplification.

Lemma 4.2.1. *Suppose Φ is a quantifier free first-order formula in DNF over $\{x_1,\ldots,x_J\}$ consisting of terms $\phi_1 \vee \ldots \vee \phi_K$. Since each branch of MultiplexNet (h_k) is constructed to satisfy a specific term (ϕ_k), by construction, the output of MultiplexNet will* *satisfy* Φ*:* $\{\hat{x}_1, \ldots, \hat{x}_J\} \models \Phi$.

4.2.2 MultiplexNet as a Latent Variable Problem

MultiplexNet introduces a latent categorical variable k that selects among the different terms $\phi_k, k \in [1, \ldots, K]$. The model then incorporates a constraint transformation term h_k conditional on the value of the categorical variable.

$$p_\theta(x) = p_\theta(h_k(x)|k)p(k) \tag{4.7}$$

A lower bound on the likelihood of the data can be obtained by introducing a variational approximation to the latent categorical variable k. This standard form of the variational lower bound (ELBO) is presented in Eq. 4.8.

$$\log p_\theta(x) \geq \mathbb{E}_{q(k)}[\log p_\theta(h_k(x)|k) + \log p(k) - \log q(k)] := ELBO(x) \tag{4.8}$$

Gradient-based methods require calculating the derivative of Eq. 4.8. However, as $q(k)$ is a categorical distribution, the standard reparameterisation trick cannot be applied (Kingma and Welling, 2014). One possibility for dealing with this expectation is to use the score function estimator, as in REINFORCE (Williams, 1992); however, while the resulting estimator is unbiased, it has a high variance (Mnih and Gregor, 2014). It is also possible to replace the categorical variable with a continuous approximation as is done by Maddison et al. (2017) and Jang et al. (2016); or, if the dimensionality of the categorical variable is small, it can be marginalised out as in Kingma et al. (2014). In the experiments in Section 4.3, we follow Kingma et al. (2014) and marginalise this variable,[2] leading to the following learning objective:

$$\mathscr{L}(\theta; x) = -\sum_{k=1}^{K} q(k) \left[\log p_\theta(h_k(x)|k) + \log p(k) - \log q(k)\right] \tag{4.9}$$

We show in Section 4.3 that this approach can be applied equally successfully for a generative modelling task (where the goal is density estimation) as for a discriminative task (where the goal is structured classification). This helps to demonstrate the universal applicability of incorporating domain knowledge into the training of networks.

4.2.3 Architecture of MultiplexNet

MultiplexNet accepts as input a data set consisting of samples from some target distribution, $p^*(x)$, and some constraints, Φ that are known about the data set. We assume

[2] Although we note that the alternatives should also be explored.

that the constraints are correct, in that Eq. 4.1 holds for all x. We aim to model the unknown density, p^*, by maximising the likelihood of a parameterised model, $p_\theta(x)$ on the given data set. Moreover, our goal is to incorporate the domain constraints, Φ, into the training of this model.

We first assume that the domain constraints are provided in DNF. This is a reasonable assumption as any logical formula can be compiled to DNF, although there might be an exponential number of terms in worst case scenarios (discussed further in Section 4.4). For each term ϕ_k in the DNF representation of $\Phi = \phi_1 \vee \phi_2 \vee \cdots \vee \phi_K$, we then introduce a transformation, h_k, that ensures any real-valued input is transformed to satisfy that term. Note that we assume efficient and non-redundant compilation of the formula (thus for any two terms ϕ_k and ϕ_j, for $j! = k \implies \phi_k! = \phi_j$. Given a Softplus transformation g, we can suitably restrict the domain of any real-valued variable such that the output satisfies some specification of ϕ_k. For example, consider the constraint, e.g., $\phi_1 : x > y + 2 \wedge x < 5$. The transformation $h_1(x') = -g(-(g(x') + \alpha) + \beta)$ will constrain the real-valued variable x' such that ϕ_1 is satisfied. In this example, y does not need to be constrained. Here $\beta = 5$ and $\alpha = \log(e^{5-(y+2)} - 1)$. Any combination of inequalities can be suitably restricted in this way. Equality constraints, can be handled by simply setting the output to the value that is specified.

MultiplexNet therefore accepts the unconstrained output of a network, $x' \in \mathbb{R}$, and introduces K constraint terms h_k that each guarantee the constrained output $x_k = h(x')$ will satisfy a term, ϕ_k, in the DNF representation of the constraints. The output of the network is then K transformed versions of x' where each output x_k is guaranteed to satisfy Φ. The Categorical selection variable $k \sim q(k \mid x)$ can be marginalised out leading to the following objective:

$$\mathscr{L}(\theta) = \sum_{i=1}^{20} \pi_k \left[\mathscr{L}'(x_k) + \log \pi_k \right] \tag{4.10}$$

In Eq. 4.10, $\mathscr{L}'$ refers to the observation likelihood that would be used in the absence of any constraint. x_k is the k^{th} constrained term of the unconstrained output of the network: $x_k = h_k(x')$. This architecture is represented pictorially in Fig. 4.1.

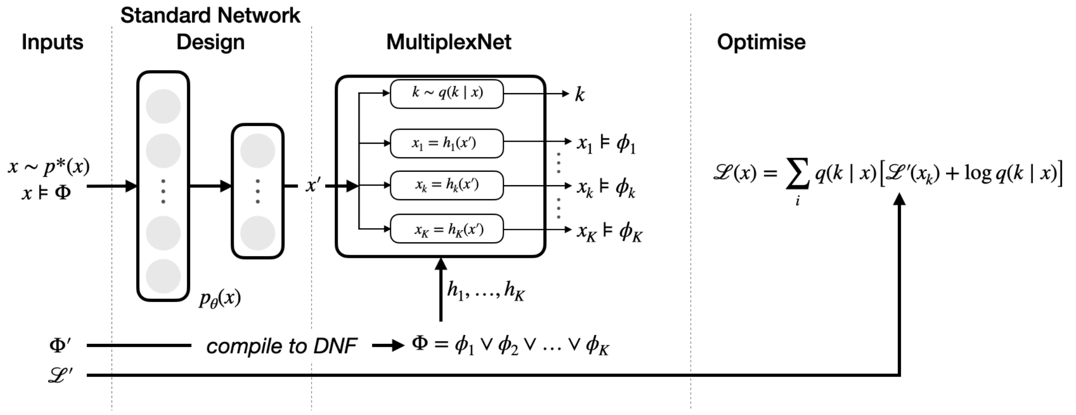


Figure 4.1: Architecture of the MultiplexNet. We show how to append this framework to an existing learning scheme. The unconstrained output of the network x', along with the constrain transformation terms $h_1, \ldots, h_K$ are used to create K constrained output terms $x_1, \ldots, x_K$. The latent Categorical variable k is used to select which term is active for a given input. In this paper, we marginalise the Categorical variable leading to the specified loss function.

4.3 Experiments

We apply MultiplexNet to three separate experimental domains. The first domain demonstrates a density estimation task on synthetic data when the number of available data samples is limited. We show how the value of the domain constraints improves the training when the number of data samples decreases; this demonstrates the power of adding domain knowledge into the training pipeline. The second domain applies MultiplexNet to labelling MNIST images in an unsupervised manner by exploiting a structured problem and data set. We use a similar experimental setup to the MNIST experiment from DeepProbLog (Manhaeve et al., 2018); however, we present a natural integration with a generative model that is not possible with DeepProbLog. The third experiment uses hierarchical domain knowledge to facilitate an image classification task taken from Fischer et al. (2019) and Xu et al. (2018). We show how the use of this knowledge can help to improve classification accuracy at the superclass level.

4.3.1 Synthetic Data

In this illustrative experiment, we consider a target data set that consists of the six rectangular modes that are shown in Figure 4.2. The samples from the true target density

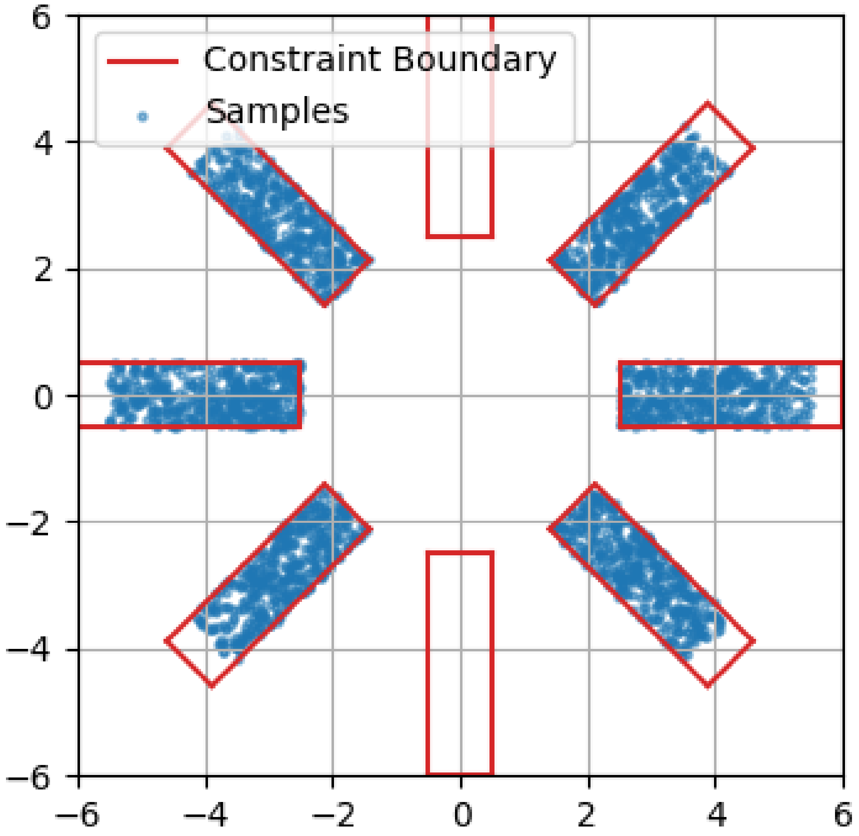


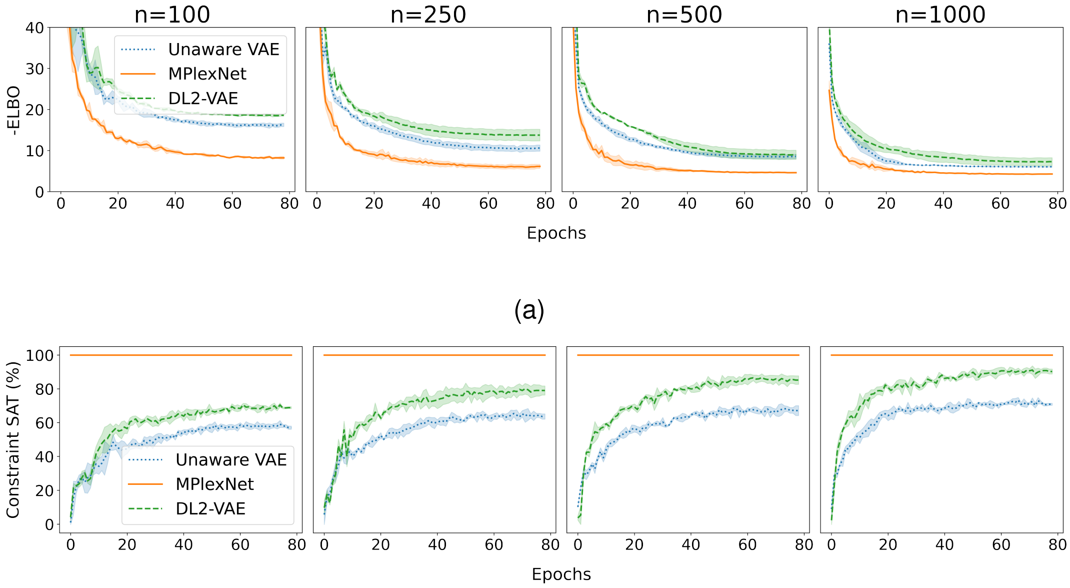
Figure 4.2: Simulated data from an unknown density. We assume that we know some constraints about the domain; these are represented by the red boxes. We aim to represent the unknown density, subject to the knowledge that the constraints must be satisfied.

are shown, along with 8 rectangular boxes in red. The rectangular boxes represent the domain constraints for this experiment. Here, we show that an expert might know where the data can exist but that the domain knowledge does not capture all of the details of the target density. Thus, the network is still tasked with learning the intricacies of the data that the domain constraints fail to address (e.g., not all of the area within the constraints contains data). However, we desire that the knowledge leads the network towards a better solution, and also to achieve this on fewer data samples from the true distribution.

This experiment represents a density estimation task and thus we use a likelihood-based generative model to represent the unknown target density, using both data samples and domain knowledge. We use a variational autoencoder (VAE) which optimizes a lower bound to the marginal log-likelihood of the data. However, a different generative model, for example, a normalizing flow (Papamakarios et al., 2019) or a GAN (Goodfellow et al., 2014), could as easily be used in this framework. We optimize Eq. 4.9 where, for this experiment, the likelihood term $\log p_\theta(\cdot \mid k)$ is replaced by the standard VAE loss. Additional experimental details, as well as the full loss function, can be found in Appendix C.1.1.

We vary the size of the training data set with $N \in \{100, 250, 500, 1000\}$ as the four experimental conditions. We compare the lower bound to the marginal log-likelihood under three conditions: the MultiplexNet approach, as well as two baselines. The first baseline (Unaware VAE) is a vanilla VAE that is unaware of the domain constraints. This scenario represents the standard-setting where domain knowledge is simply ignored in the training of a deep generative network. The second baseline (DL2-VAE) represents a method that appends a loss term to the standard VAE loss. It is important to note that this approach, from DL2 (Fischer et al., 2019), does not guarantee that the constraints are satisfied (clearly seen in Figure 4.3b). A possible alternative baseline would be rejection sampling whereby any produced sample from the VAE output could be accepted / rejected based on the validation of the sample via the logic. This approach could be used to compare as a baseline for the data likelihood metric; however, one of the goals of this work is to *use the logic to inform a more efficient traning regime* (i.e., we are able to back-propagate through the logic transformation terms h_k). Thus, we chose not to implement this as a baseline as it could not be used to achieve this result (mainly seen in the discussion on sample efficiency below).

Figure 4.3 presents the results where we run the experiment on the specified range of training data set sizes. The top plot shows the variational loss as a function of the number of epochs. For all sizes of training data, the MultiplexNet loss on a test set can be seen to outperform the baselines. By including domain knowledge, we can reach a better result, and on fewer samples of data, than by not including the constraints. More important than the likelihood of held-out data is that the samples from the models' posterior should conform with the constraints. Figure 4.3b shows that the baselines struggle to learn the structure of the constraints. While the MultiplexNet solution is unsurprising, the constraints are followed by construction, the comparison to the baselines is stark. We also present samples from both the prior and the posterior for all of these models in Appendix C.1.1. In all of these, MultiplexNet learns to approximate the unknown density within the predefined boundaries of the provided constraints.



(a)

(b)

Figure 4.3: Results from the synthetic data experiment (a) Negative lower bound to the held-out likelihood of data (-ELBO). The MultiplexNet approach learns to represent the data with a higher likelihood, and faster than the baselines. (b) % of reconstruction samples from the VAE that obey the domain constraints. The MultiplexNet approach, by construction, can only generate samples within the specified constraints.

4.3.2 MNIST - Label-free Structured Learning

We demonstrate how a structured data set, in combination with the relevant domain knowledge, can be used to make novel inferences in that domain. Here, we use a similar experiment to that from Kingma et al. (2014) where we model the MNIST digit data set in an unsupervised manner. Moreover, we take inspiration from Manhaeve et al. (2018) for constructing a structured data set were the images represent the terms in a summation (e.g., $image(2) + image(3) = 5$). However, we add to the complexity of the task by (1) using no labels for any of the images;[3] and, (2) considering a generative task.

Kingma et al. (2014) propose a generative model that reasons about the cluster assignment of a data point (a single image). In particular, in their popular "Model 2," they describe a generative model for an image x such that the probability of the image pixel values are conditioned on a latent variable (z) and a class label (y): $p_\theta(x \mid z, y)p(z \mid y)p(y)$. We can interpret this model using the MultiplexNet framework where the cluster assignment label $y = k$ implies that the image x was generated from cluster k. Given a reconstruction loss for image x, conditioned on class label y ($\mathscr{L}(x, y)$), the domain knowledge in this setting is: $\Phi := \bigvee_{k=1}^{10} \mathscr{L}(x, y) \wedge (y = k)$. We can successfully model the clustering of the data using this setup but there is no means for determining which label corresponds to which cluster assignment.

We therefore propose to augment the data set such that each input is a quintuple of four images (x_1, x_2, x_3, x_4) in the form $label(x_1) + label(x_2) = (label(x_3), label(x_4))$. Here, the inputs $label(x_1)$ and $label(x_2)$ can be any integer from 0 to 9 and the result $(label(x_3), label(x_4))$ is a two digit number from (00) to (18). While we do not know explicitly any of the cluster labels, we do know that the data conform to this standard. Thus for all i, j, k where $k = i + j$, the domain knowledge is of the form:

$$\Phi := \bigvee_{i,j,k} \left[(y_1 = i) \wedge (y_2 = j) \wedge (y_3 = \mathbb{1}_{k>9}) \wedge (y_4 = k \bmod 10) \bigwedge_{n=1}^{4} \mathscr{L}(x_n, y_n) \right] \quad (4.11)$$

In this setting, the categorical variable in the MultiplexNet chooses among the 100 combinations that satisfy $label(x_1) + label(x_2) = (label(x_3), label(x_4))$. This experiment has similarities to DeepProbLog (Manhaeve et al., 2018) as the primitive $\mathscr{L}(x, y)$ is repeated for each digit. In this sense, it is similar to the "neural predicate" used by

[3] In the MNIST experiment from Manhaeve et al. (2018), the authors use the result of the summation as labels for the algorithm. We have no such analogy in this experiment and thus cannot use their DeepProbLog implementation as a baseline.

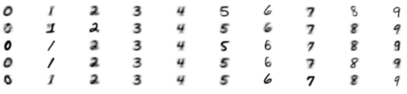


Figure 4.4: Reconstructed/Decoded samples from the prior, z, of the trained model where each column conditions on a different value for y. It can be seen that the model has learnt to represent all of the digits $[0-9]$ with the correct class label, even though no labels were supplied to the training process.

Manhaeve et al. (2018), and the MultiplexNet output layer implements what would be the logical program from DeepProbLog. However, it is not clear how to implement this label-free, generative task within the DeepProbLog framework.

In Figure 4.4, we present samples from the prior, conditioned on the different class labels. The model can learn a class-conditional representation for the data, *given no labels for the images*. This is in contrast to a vanilla model (from Kingma et al. (2014)) which does not use the structure of the data set to make inferences about the class labels. We present these baseline samples as well as the experimental details and additional notes in Appendix A. Empirically, the results from this experiment were sensitive to the network's initialisation and thus we report the accuracy of the top 5 runs. We selected the runs based on the loss (the ELBO) on a validation set (i.e., the labels were still not used in selecting the run). The accuracy of the inferred labels on held-out data is 97.5 ± 0.3.

4.3.3 Hierarchical Domain Knowledge on CIFAR100

The final experiment demonstrates how to encode hierarchical domain knowledge into the output layer of a network. The CIFAR100 (Krizhevsky et al., 2009) data set consists of 100 classes of images where the 100 classes are in turn broken into 20 superclasses (SC). We wish to encode the belief that images from the same SC are semantically related. Following the encoding in Fischer et al. (2019), we consider constraints that specify that groups of classes should together be very likely or very unlikely. For example, suppose that the SC label is *trees* and the class label is *maple*. Our domain knowledge should state that the *trees* group must be very likely even if there is uncertainty in the specific label *maple*. Intuitively, it is egregious to misclassify this example as a *tractor* but it would be acceptable to make the mistake of *oak*. This can be implemented by training a network to predict first the SC for an unknown image and thereafter the class label, conditioned on the value for the SC.

We chose rather to implement this same knowledge using the MultiplexNet framework. Let $x_k \in SC_i$ denote the output of a network that predicts the k^{th} class label within the i^{th} SC. Let $\alpha \in [0,1]$ denote the minimum requirement for a SC prediction (e.g., if $\alpha = 0.95$, we require that a SC be predicted with probability 0.95 or more). The domain knowledge is:

$$\Phi := \bigvee_{i=1}^{20} \left[\bigwedge_{k \in SC_i} \Big(x_k > \log(\frac{\alpha}{1-\alpha}) + \log \sum_{j \notin SC_i} \exp\{x_j\} \Big) \right] \tag{4.12}$$

Eq. 4.12 states that for all labels within an SC group, the unnormalised logits of the network should be greater than the normalised sum of the other labels belonging to the other SCs with a margin of $\log(\frac{\alpha}{1-\alpha})$. We explain Eq. 4.12 further and present other experimental details in Appendix C.1.3. This constraint places a semantic grouping on the data as the network is forced into a low entropy prediction at the superclass level.

We compare the performance of MultiplexNet to three baselines and report the prediction accuracy on the fine class label as well that on the superclass label. We use a Wide ResNet 28-10 (Zagoruyko and Komodakis, 2016) model in all of the experimental conditions. The first two baselines (Vanilla) only use the Wide ResNet model and are trained to predict the fine class and the superclass labels respectively. The second baseline (Hierarchical) is trained to predict the superclass label and thereafter the fine class label, conditioned on the value for the superclass label. This represents the bespoke engineering solution to this hierarchical problem. The final baseline (DL2) implements the same logical specification that is used for MultiplexNet but uses the DL2 framework to append to the standard cross-entropy loss function.

Table 4.1 presents the results for this experiment. Firstly, it is important to note the difficulty of this task. The Vanilla ResNet that predicts only the super-class labels for the images underperforms the baseline that is tasked with predicting the true class label. Moreover, while the hierarchical baseline does outperform the vanilla models on the task of super-class prediction, this comes at a cost to the true class accuracy. The MultiplexNet approach provides a slight improvement at the SC classification accuracy and importantly, the domain constraints are always met. Surprisingly, the DL2 baseline improves upon the class accuracy but it has a limited impact on the superclass accuracy and on the constraint satisfaction.

Table 4.1: Accuracy on class label prediction and super-class label prediction, and constraint satisfaction on CIFAR100 data set

Model	Class Accuracy	Super-class Accuracy	Constraint Satisfaction
Vanilla ResNet	$75.0 \pm (0.1)$	$84.0 \pm (0.2)$	$83.8 \pm (0.1)$
Vanilla ResNet (SC only)	NA	$83.2 \pm (0.2)$	NA
Hierarchical Model	$71.2 \pm (0.2)$	$84.7 \pm (0.1)$	$\mathbf{100.0} \pm (\mathbf{0.0})$
DL2	$\mathbf{75.3} \pm (\mathbf{0.1})$	$84.3 \pm (0.1)$	$85.8 \pm (0.2)$
MultiplexNet	$74.4 \pm (0.2)$	$\mathbf{85.4} \pm (\mathbf{0.3})$	$\mathbf{100.0} \pm (\mathbf{0.0})$

4.4 Limitations and Discussion

The limitations of the suggested approach relate to the technical specification of the domain knowledge and to the practical implementation of this knowledge. We discuss first these two aspects and then we discuss a potential negative societal impact.

First, we require that experts be able to express precisely, in the form of a logical formula, the constraints that are valid for their domain. This may not always be possible. For example, given an image classification task, we may wish to describe our knowledge about the content of the images. Consider an example where images contain pictures of *dogs* and *fish* and that we wish to express the knowledge that dogs must have four legs and fish must be in water. It is not clear how these conceptual constraints would then be mapped to a pixel level for actual specification. Moreover, it is entirely plausible to have images of dogs that do not include their legs or images of fish where the fish is out of the water. The logical statement itself is brittle in these instances and would serve to hinder the training, rather than to help it. This example serves to present the inherent difficulty that is present when expressing robust domain knowledge in the form of logical formulae.

The second major limitation of this approach deals with the DNF requirement on the input formula. We require that knowledge be expressed in this form such that the “or” condition is controlled by the latent categorical variable of MultiplexNet. It is well known that certain formulae have worst-case representations in DNF that are exponential in the number of variables. This is undesirable in that the network would have to learn to choose among the exponentially many terms.

One of the overarching motivations for this work is to constrain networks for safety-critical domains. While constrained operation might be desired on many accounts, there may exist edge cases where an autonomously acting agent should act in an undesirable manner to avoid an even more undesirable outcome (a thought experiment of this spirit is the well known Trolley Problem (Hammond and Belle, 2021)). By guaranteeing that the operating conditions of a system be restricted to some range, our approach does encounter vulnerability with respect to edge, and unforeseen, cases. However, to counter this point, we argue it is still necessary for experts to define the boundaries over the operation domain of a system in order to explicitly test and design for known worst-case scenario settings.

4.5 Conclusion

This work studied how logical knowledge in an expressive language could be used to constrain the output of a network. It provides a new and general way to encode domain knowledge as logical constraints directly in the output layer of a network. Compared to alternative approaches, we go beyond propositional logic by allowing for arithmetic operators in our constraints. We can guarantee that the network output is 100% compliant with the domain constraints, which the alternative approaches, which append a "constraint loss," are unable to match. Thus our approach is especially relevant for safety-critical settings in which the network must guarantee to operate within predefined constraints. In a series of experiments, we demonstrated that our approach leads to better results in terms of data efficiency (the amount of training data that is required for good performance), reducing the data burden that is placed on the training process. In the future, we are excited about exploring the prospects for using this framework on downstream tasks, such as robustness to adversarial attacks.

Chapter 5

Interpretable Models for Understanding Immersive Simulations

5.1 Introduction

This chapter investigates methods for evaluating the interpretability of models of time series data arising from people's interactions in immersive simulations such as those used for teaching in healthcare, disaster response and science education (Alinier et al., 2014; Amir and Gal, 2013). In such simulations, people's interactions engender a rich array of emergent outcomes and yield diverse opportunities for learning (Smørdal et al., 2012). In the immersive simulation used in this study, Connected Worlds[1] (CW), students interact with an ecological simulation to learn about the causal effects of their actions on environments over time (Mallavarapu et al., 2019).

Rich causal relationships, simultaneous participation from students and the changing dynamics of immersive simulations can make it difficult for people to determine how their interactions with the simulation caused the changes they observe in the simulated world. Machine learning methods can be used to summarize the effects of participants' actions over various time periods. For such methods to be effective, though, they must meet the challenge of identifying a model that is both "true to the data" and understandable to the target audience interested in uncovering the causal relationships.

This chapter applies the general framework from Doshi-Velez and Kim (2017) and demonstrates an application of how to design tests that evaluate models in an immersive simulation setting. To this end, we show how to: determine that machine learning model, from a set of candidates, that people understand best (Caruana et al., 2015). It compares the selection of a model according to a criterion that optimizes for maximum statistical information with one that optimizes for interpretability. The ability to identify the model that is best (or among the top choices) for interpretability is essential to a system's capability to explain its conclusions (Rosenfeld and Richardson, 2019).

Our approach to addressing the interpretability problem comprises the following: (1) select a set of machine learning models for segmenting time series data; in the domain we investigated, the segmentation is of students' interactions with CW into coherent periods of time; (2) design tests for computing the interpretability score of a model for a given input; (3) empirically evaluate the models with respect to their interpretability score in a user study.

To infer the boundaries of stable periods in the data of CW dynamics, we use a family of hidden Markov models (HMMs). These HMMs are augmented with an

[1] Installed at the New York Hall of Science (NYSCI): `https://nysci.org/home/exhibits/connected-worlds/`

additional “sticky” hyperparameter which biases the transition dynamics of the latent state-space (Fox et al., 2008). The input to each HMM is a multidimensional time series representing the response of the CW system to actions performed by students in the simulation. The output of the HMM is a segmentation of the time series into a set of periods, which are contiguous lengths of time during which the system dynamics form a stable linear process.

We implemented two tests of interpretability for CW models: the Forward Simulation and Binary Forced Choice (Doshi-Velez and Kim, 2017). These tests each determine the extent to which the learnt representations are interpretable to people, albeit in different ways. They both use a visualization of the inferred periods that shows experimental subjects snapshots of the CW system’s state from the selected periods that the HMM inferred.

The results showed that the interpretability of the different models varied according to the value(s) of HMM parameters. In particular, the HMM that optimized statistical information criteria did not optimize interpretability quality. In addition, a fully Bayesian approach, which does not require hyperparameter tuning, offered a good balance between interpretability and performance on the theoretical statistical tests. We argue that the Bayesian approach could be suitable for situations in which it is not possible to engage people in determining interpretability or doing so would be unethical or impractical.

This chapter makes three contributions. First, it provides an end-to-end paradigm for the design and evaluation of the interpretability of models for unsupervised learning in time series domains. Second, it defines new interpretability tests for unsupervised time-series settings and applies them to real-world data. Third, in identifying the Bayesian solution, it provides an attractive alternative to model selection when human subject experimentation is not possible. Finally, we note that the results of this investigation have been deployed in a classroom study to assist teachers in explaining systems thinking to students who participated in the CW simulation study.

5.1.1 Contributions

In the work that follows, I was the main contributor to this project. I designed the experiments and implemented the algorithms for the experiments. Moreover, I designed the user studies and collected these data from an online webpage that I designed, deployed and managed. Thank you to Barbara Gross, Andee Ruben and Kobi Gal for their overall insights, expertise in the experimental design and of course in the polishing of the writing.

5.2 The Connected Worlds Domain

Connected Worlds (CW), a multi-person ecology simulation (installed at NYSCI), aims to teach students about complex systems and systems thinking. Its immersive environment comprises four biomes (Desert, Grasslands, Jungle & Wetlands) connected by a central water flow fed by a waterfall. Students plant trees which flourish or die, animals arrive or depart, and rain clouds form and rain feeds the waterfall.

Students control the direction of water flows in the simulation by moving foam logs to direct water among the biomes. Water enters the simulation through rainfall events, which are not under student control. Figure 5.1 gives a snapshot of the system state, we refer to this snapshot as the session-view. This session-view is a system-generated representation of the water flows and it directly reflects the logged water flows and levels in the simulation. The output of a CW session is a time series recording the levels of water in the different biomes for 8 minutes at a 1Hz frequency. The ability to model the effects of student actions on the environment was limited by two factors: The time series was the only source of information about students' interactions, and it was not possible to access the CW simulation except at NYSCI.

The CW simulation is complex on several dimensions as a large number of students simultaneously execute actions that change the state of the simulated environment. Each participant has a different view of what transpired, depending on the actions s/he took and the environmental changes that resulted. Students' activities are recorded as a movie (see Figure 5.1) that can be shown to students and teachers. This movie can inform discussions about the causal effects of the students' actions on simulation outcomes, but it obscures temporal dependencies in their interactions. This limitation motivated the use of ML algorithms to better support students' understanding of the effects of their actions on the simulation's progression.

5.3 Interpretability Tests for CW

Let D be a time series that records the levels of water in the different biomes. Let M be a model that takes as input a time series D and outputs a segmentation of D into

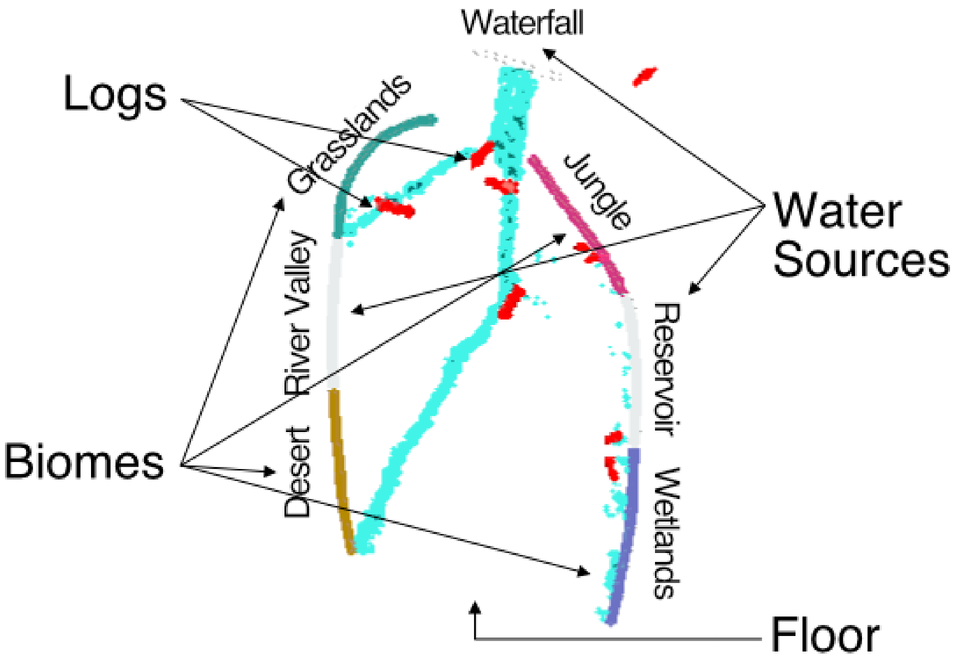


Figure 5.1: CW session-view. Biomes are labelled on the perimeter and logs appear as thick red lines. Water (blue stream in the middle of the image) enters via the waterfall and in this image it mainly flows toward the Grasslands and the Desert.

periods. Each period aims to provide a coherent description of the water flow for a length of time.

Importantly, a single period is insufficient for modelling the effects of students' interactions with CW, because students' sustained actions have complex effects on the system dynamics over time. For example, when students choose to direct water to the Desert and Plains and plant trees in the Desert, the system dynamics are entirely different from the case when water is directed towards the Jungle and the Desert, and the Plains are left to dry. We must therefore allow for multiple periods. Each period describes a length of time where water flowed to a sufficiently stable target. From the above example, one period can describe water that mainly flows to the Plains and to the Desert; students then move logs to re-route water flow to the Jungle, thus starting a new period.

We use an interpretability score IS to measure the interpretability of a model M applied to D. The interpretability score is computed via an average across test instances, $T(M,D,i)$, which each take as input a model M, a time series D and a selected point in time i from the time series. Each test instance returns True if an evaluator successfully

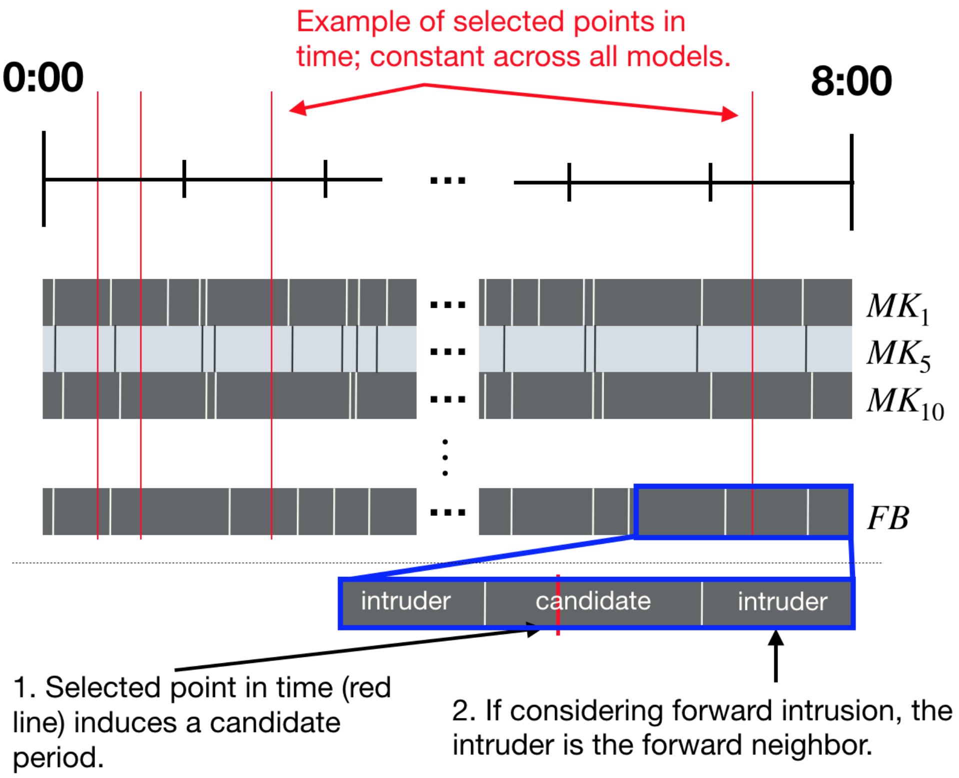

Example of selected points in time; constant across all models.
0:00
8:00
...
MK_1
MK_5
MK_{10}
FB
intruder
candidate
intruder
1. Selected point in time (red line) induces a candidate period.
2. If considering forward intrusion, the intruder is the forward neighbor.

Figure 5.2: The time series is represented as a horizontal line from minute 0 to 8; red vertical lines denote sampled time points in the time series; each model is shown as a grey rectangle; models segment time series into periods delimited by white vertical lines. The forward or backward neighbour of the candidate period is selected as an intruder.

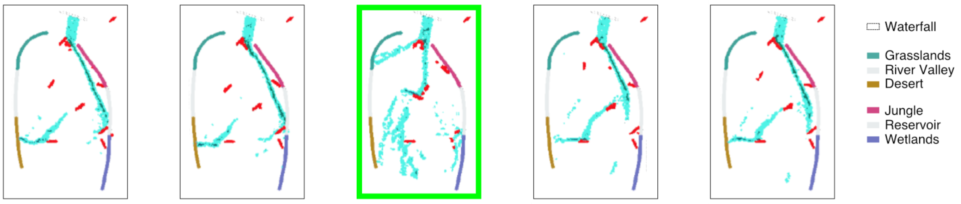


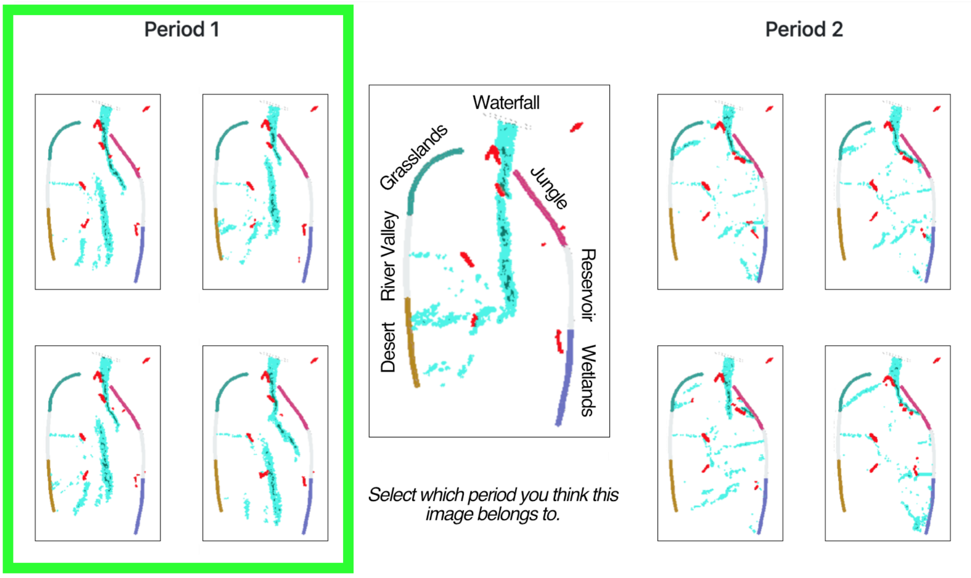
Figure 5.3: Screenshot of the Forward Simulation test interface. Here 4 of the images show water flowing towards the Desert. An intruder image, the highlighted one, comes from a different period and shows water flowing to both the Desert and the Grasslands.

completes a required objective.

We adapted the Forward Simulation and Binary Forced Choice tests (Doshi-Velez and Kim, 2017) to the CW domain using the notion of *candidate* and *intrusion* periods. We say that period p is *active* for model M at time i if M infers the period p to describe a contiguous length of time in the time series, and p includes the time i. Figure 5.2 shows how the tests select candidate and intrusion periods. First, a time point (red vertical line) is used to select a candidate period where the candidate period is the active period from model M at i (the active period for a model intersects with the red line). Then, the intrusion period is selected as a direct neighbour to a candidate. Each test is operationalized via a *visualization* which presents any period as a set of images extracted from the session view.

Figure 5.3 shows an example of the Forward Simulation test on a real data instance. As shown by this figure, the test sampled 4 session-view images from the candidate period of model M at time i, and a single session-view image sampled from the intrusion period. The images were presented in random order. In Figure 5.3, the image that is outlined in green is the intrusion image that corresponds to the intrusion period. A test evaluator was required to identify which image was the intrusion image.

Figure 5.4 presents an example of the Binary Forced Choice test. The test displays an unknown session-view image from a candidate period (centre of the screen) and additional images from two competing periods that contain this image ("Period 1" or "Period 2"). Each of the two competing periods is visualized as four images sampled from the candidate or the intruder period. The unknown image is sampled in time close to the boundary of when the candidate period transitions into the intruder period. In Figure 5.4, Period 1, highlighted in green, is the period that correctly explains the unknown image (i.e., the images in "Period 1" and the "unknown image" are all sampled from the candidate period). A test evaluator is required to choose between the two



Select which period you think this image belongs to.

Figure 5.4: Screenshot of the Binary Forced Choice user interface. An unknown centre image needs to be associated with either “Period 1” or “Period 2”. In this case, streams of water flowing to both the Grasslands and to the Jungle capture the dynamics in Period 2. Period 1 has a small amount of water reaching the Desert which is consistent with the unknown image.

possible periods.

Hypothetically, the intruder period can be chosen arbitrarily, as in Chang et al. (2009). However, intrusion periods that are further away in time from the candidate period would be easier to detect due to the non-stationary evolution of the system. We made a design decision to choose the period that is immediately adjacent to the candidate period, either forward or backwards in time. This makes it harder to distinguish between candidate and intrusion period but provides a rigorous test for the specific choice of the boundary between the two periods.

Given data set D and model M, the interpretability score IS of a model is equal to the average success of the test instances for model M over multiple points $\{i\}$ in a time series D. The set of time points $\{i\}$ were uniformly sampled from the time series with the additional constraint that each minute of interaction had at least one sample. For every model we test, we hold constant the selected times $\{i\}$ in the time series (as shown in Figure 5.2). In this way, we control for different areas in the time series being more or less difficult to segment into coherent periods.

5.4 Modeling Students' Activities in CW

In this section, we describe the design of general models for segmenting students' activities into periods of time and thereafter present the specific classes of models that are used in our interpretability tests.

5.4.1 Segmenting Time Series Data into Periods

Hoernle et al. (2018) used an HMM to model the system responses to students' activities in CW in which the latent states of the HMM corresponded to periods. Transitions between different states equate to the system changing between different periods, while self transitions mean the system persists within the same period. The authors did not address the question of how to choose the number of states. To this end, we augment the HMM with a hierarchical Dirichlet process which places this non-parametric prior over the state space, following the approach detailed by Teh et al. (2005) and Fox et al. (2008).

The "Sticky-HMM" approach introduced by Fox et al. (2008) includes a hyperparameter, κ, that biases the model to persist in a state, given that it has already adopted that state. Applied to CW, the greater the value for κ, the more the model will try to persist in any given state. The increase in the length of periods corresponds to a decrease in the number of latent states. The opposite is true for lower values of κ where there is a lower bias to persist within a given state and consequently, there are more periods that are inferred. For a detailed description of the model, including the Gibbs sampling inference scheme that is used to infer the model parameters, refer to Fox et al. (2008) and Fox (2009).

5.4.2 Model Classes

We introduce three classes of model that segment time into periods that can be used to explain the water flows:

1. MK_X: sticky HMM with fixed κ. We use the basic structure of the sticky HMM described by Fox et al. (2008) with set values for κ to produce 10 unique models, spanning a wide range of possible settings[2].

[2] $\kappa \in \{1, 5, 10, 50, 100, 150, 200, 300, 500, 700\}$.

2. *FB*: fully Bayesian sticky HMM with Gamma prior on κ. This approach places a weakly informative, conjugate Gamma prior on the hyperparameter that expresses high uncertainty over the κ values[3].

3. *Rand*: Random baseline. The random baseline generates periods of random length drawn from a Poisson distribution with the mean set to be the mean of all other periods induced by the parametric models. The random periods are defined to include the selected time points ($\{i\}$ from Section 5.3).

We refer to FB as the fully Bayesian model to indicate the fact that none of the parameters of interest are specified and consequently posterior inference is over all of the parameters in the model (including κ). This is in contrast to the MK_X models where we explicitly set the value for the sticky parameter κ.

For models in class 1 and 2, we use the Gibbs sampler, described by Fox et al. (2008), to perform inference over the parameters in the model, this includes inference over the state sequence and thus the period segmentation of the model. The observation distribution was chosen to be a mixture of two multivariate Gaussians with conjugate Normal-inverse-Wishart priors. This mixture model addresses the noise in the CW water flow, such as "splashes", which prior work has identified as a challenge in this domain (Hoernle et al., 2018).

5.5 Model Selection for Interpretability

The goal of model selection is to optimize a metric such that a specific parameter setting can be chosen as the best model for use during inference. We compare how the models from section 5.4 perform on both statistical tests and on the human interpretability tests outlined in section 5.3.

5.5.1 Selection using Statistical Information

When human interpretability testing is infeasible, one could choose to optimize some proxy to interpretability (Doshi-Velez and Kim, 2017; Lage et al., 2018). For example, Chang et al. (2009) compared the proxy of held-out log-likelihood to the human interpretability score that was a result of two tests that were run on Amazon Mechanical Turk (Mturk).

[3] The *(shape,rate)* parameters were chosen to be $(1, \frac{1}{4})$; empirical results were invariant to a range of these values.

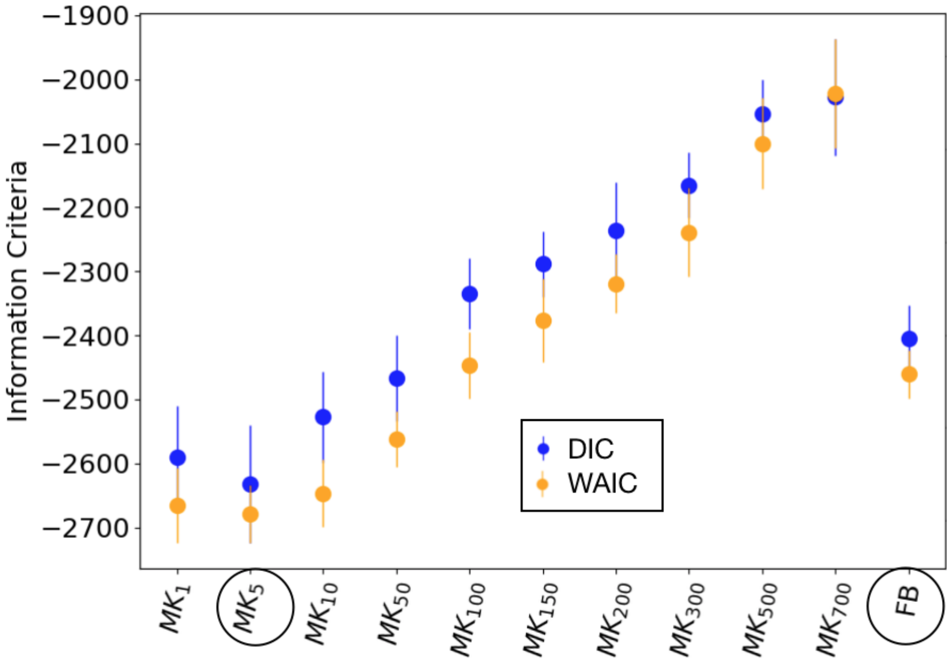


Figure 5.5: DIC and WAIC as a function of the model (lower is better). The MK_5 model is optimal, the FB approach is in 5th place.

Ideally, the model parameters would be optimized on held-out data using predictive log-likelihood as the objective (Chang et al., 2009). However, the difficulty of collecting controlled sessions of student interaction in CW meant we had few data instances available (see limitation discussion in the next section). To address this challenge we use statistical information criteria as a theoretical approximation to the predictive accuracy of a model (Gelman et al., 2013).

Figure 5.5 shows the two information criteria (the Deviance Information Criteria, DIC, and the Watanabe-Akaike Information Criteria, WAIC (Gelman et al., 2013)) plotted as a function of the model (the random model has no notion of information criteria and so was not compared here). The data set comprised of both of the log files of students' interactions (8 minutes each). The optimal model for both DIC and WAIC is the MK_5 model but we note that MK_1, MK_5 and MK_{10} all perform close to this optimal setting. Notice that the fully Bayesian model (FB) is not optimal but it is in the top 5 models for both criteria.

5.5.2 Selection using Interpretability Test

This section describes the choice of model according to interpretability quality, as measured by the interpretability tests. The set of models used in this study includes the 12 CW models described in Section 5.4. IRB was obtained for the study.

We recruited participants from two cohorts: undergraduate engineering students in a large public university and Mturk workers (with a total of 240 people who participated in the experimentation). For a given time series D in CW, we randomly sampled a set of 12-time points, which remained constant across all model conditions. Each time point was used to generate a candidate and two intrusion periods (both forward and backward in time, see Figure 5.2), making for $2 \times 12 \times 12 = 288$ tests per time series. We divided participants into two cohorts, one for Forward Simulation, and one for Binary Forced Choice tests. Both cohorts varied the models used to generate their respective tests. Each participant performed 20 tests, with no more than 2 tests generated from any given model, to ensure a representative range of models. After making their choice, participants received brief visual feedback on whether or not their selection was in agreement with the model's choice.

All participants received a detailed tutorial about CW and the study, as well as a pre-study comprehension quiz[4]. Mturk workers were paid a base rate of \$0.25 for participating and a bonus structure of \$0.1 for each correct response.

We first describe results in terms of accuracy (the percent of correctly labelled test instances). The top-performing model was MK_{200} with an accuracy of 83% on the Forward Simulation test and MK_{100} with an accuracy of 82% on the Binary Forced Choice test. The random baseline model performed consistently poorly with an average accuracy of 53% on both tests. The fully Bayesian model achieved an accuracy of 72% and 70% respectively on the two tests.

To control for ordering effects, chosen time periods, data instance used, and effects of individual participants, we applied an L2 regularized logistic regression for predicting the user-specific success on the experiment trial, shown in Figure 5.6. The y-axis presents the improvement in log-odds that a model has on the expected response accuracy (higher is better). As shown by this figure, the Forward Simulation shows a high variance with no clear maximum. In contrast, the Binary Forced Choice test has a clear maximum in the region of MK_{100} and MK_{150}.

From Figures 5.5 and 5.6 we can infer the following four conclusions. First, all

[4] Tutorial pdf slides are available at `https://www.dropbox.com/s/pu2nxk2k0g81ql6/forijcai.pdf`

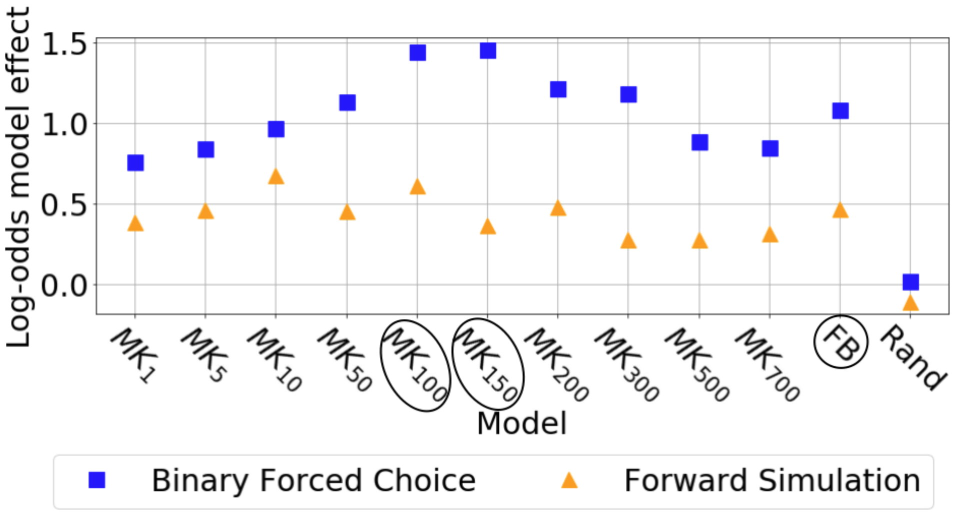


Figure 5.6: Effect of each model on the log-odds of a test evaluator selecting the correct response (controlling for the test evaluator, the experiment trial, log file and ordering effects).

of the models ($MK_1,\dots,MK_{700},FB$) outperform the random baseline: participants are more likely to select the correct response from any of these models. This result suggests that periods of stable dynamics exist in the data and that it is possible to construct models, which describe these dynamics, that are interpretable to people.

Second, the Binary Forced Choice test is a preferable measure for interpretability to the Forward Simulation test. Figure 5.6 shows that the Binary Forced Choice test exhibits a clear peak (around MK_{100} and MK_{150}) where interpretability of the model is maximized. These models also maximized the raw accuracy of the Binary Forced Choice test.

On the other hand, the Forward Simulation test has a greater variance across models and across data instances. Two possible causes for this higher variance are: (1) there is more room for error in the Forward Simulation test (5 choices vs. 2 choices in Binary Forced Choice); (2) sampling a single image to represent a period (as in Forward Simulation) presents less information to the user than sampling 4 images (as in Binary Forced Choice).

Third, the best κ settings vary for different tests and information criteria. Model interpretability grows steadily as the value of κ increases, with MK_{100} and MK_{150} being the optimal models, and then proceeds to decrease steadily. These models are not consistent with the model MK_5 that optimized the information criteria. Note that higher κ values are “sticky” - they bias the model towards longer periods, which condense too many activities to make sense to people. On the other end of the spectrum, lower κ values allow for more (shorter) periods that may capture noise in the system. The κ value for models MK_{100} and MK_{150} represent a "sweet spot" in-between these two extremes.

Finally, the fully Bayesian model (FB) performs consistently well on both information criteria and interpretability tests. It is interesting to note that while this model does not find the optimal setting (from neither the statistical information criteria nor from the human interpretability task) it does perform well across all tests, tasks and instances, and is fully automated (no human evaluation is required to choose an optimal parameter setting).

We conclude this section by mentioning the limitation that the user study was based on a small number $(n = 2)$ of instances. This was due to the difficulty in obtaining controlled sessions of student behaviour in CW. Despite this issue, the differences between the models in Figure 5.6 are statistically significant, having been evaluated across 12 different time points for each instance and with hundreds of evaluators.

5.6 Conclusion

With the growing prevalence of immersive simulations, the need arises for AI systems that help people gain insight into the ways participants' activities affect the simulation outcomes. We have studied an environmental simulation intended to teach students about the causal effects of their actions. Our results show that algorithms can segment time-series log data into periods that are meaningful for people. Selecting hyperparameters in these models is a challenge, especially when trying to optimize the representations they produce for their interpretability. We have described ways to select these hyperparameters using two tests that are grounded in the literature. We showed that the fully Bayesian method is a promising technique for implementing a model when people cannot directly assess and evaluate the models. Our results are important for any unsupervised machine learning task for which interpretability is an important criterion because in such cases the model selection problem will be encountered. The work forms part of a broader project where the goal is to generate relevant summaries of the CW dynamics such that teachers can effectively engage their students in discussions about their own experiences with the simulation.

In future work, we plan to explore alternative ways to measure interpretability quality in time series domains, including the design of a counterfactual simulation test (Doshi-Velez and Kim, 2017), and the application of our approach to additional domains.

Chapter 6

Conclusion

6.1 Introduction

I have presented a framework that uses a Data Science life-cycle to investigate how and why people contribute to task systems. Building on the work of Segal (2018), I define a task system to consist of an online platform where participants complete actions that contribute to the shared knowledge repository that constitutes the system. I have shown how to encode hypotheses from Behavioural Sciences into the definition of a model and how to perform posterior inference over the assignment of data to these different hypotheses. By including more than one hypothesis in a single model, we allow for heterogeneity in a data set; this is critically important when modelling people's behaviour in task systems where individuals have different goals. The framework will help researchers to gain deeper insights into which behavioural hypotheses apply to which people.

Figure 1.1 presented a pictorial view of the main tasks in the framework. Step 1 requires a model specification. Chapter 3 includes an example of how to encode a behavioural hypothesis, the "Goal-Gradient Hypothesis" (Hull, 1932; Kivetz et al., 2006), into the output layer of a flexible statistical model. The predictions that are made by a hypothesis must be of a measurable quantity and thus they can be directly encoded into a model's output layer. It is also important that we allow for complex behaviours that are independent of the predictions of the behavioural hypotheses. For example, in the Stack Overflow analysis, we noticed a cyclical pattern to people's work whereby some people work more on weekends and some work more on week days (Yanovsky et al., 2019). The proposed approach thus uses the flexibility of deep generative models when encoding the behavioural hypotheses into a statistical model to capture these complexities.

Chapter 4 explores how a very general set of constraints and restrictions, such as those defined by a logical program, can be used to train and constrain deep generative models. In this way, logical programs could be written to encode the predictions associated with certain hypotheses and these would then be used in the inference process. More generally, in this chapter, I showed that by constraining networks with prior knowledge, we can efficiently train complex generative models on fewer data and can often arrive at better solutions than had we not used these constraints.

In Chapter 5, I finally explored how to design tests to evaluate the interpretability of complex models. The real-world data from task systems, stemming from people's interactions with the systems, is often complex and aggregated data might represent an amalgamation of the goals and wants of a diverse set of people. Thus we do not merely seek the model that makes the best predictions but we aim to understand the behaviours of these people who interact with the task system. In this chapter, I showed how to design interpretability tests that aim to evaluate how interpretable a certain model is. I further explored a trade-off that exists between the interpretability of a model and its statistical fit to data. It might be the case that a sub-optimal predictive model is chosen as it provides more insight into the complex behaviours of people in task systems.

This brief chapter proceeds by detailing the contributions that are made in the thesis. It then touches on the limitations of the proposed approach and finally lists some exciting avenues for future work.

6.2 Summary of Contributions

The contributions that are made in this thesis are as follows:

- Modification of a standard Data Science lifecycle (Box's Loop in Figure 1.1) to the task of understanding people's behaviour in task systems.
- Integration of domain constraints into the training of a neural network such that behavioural hypotheses can be explored and compared. This involved integrating both continuous and categorical latent variables into one model.
- Application of the proposed framework and inference scheme to an example real-world domain: That of data from Stack Overflow where we showed that the "Goal-Gradient Hypothesis" does indeed hold but for only a limited set of people.
- Design and integration of tests to evaluate the interpretability of a proposed model. The interpretability of a model is critical when performing inference with the goal of understanding people's behaviour in task systems.
- Application of interpretability tests to an interesting and complex real-world exploratory learning environment.

6.3 Limitations

The framework that is proposed in this thesis has a few limitations that I detail here. First and foremost, we are proposing to *model* the interaction of human behaviour from observational data. The fact that observational data are used results in causal conclusions being difficult to draw. This is a limitation of the framework as we would ideally like to know the causal relations that lead to the observed behaviour of the participants in task systems. In Section 6.4, I propose that this can be dealt with in future extensions of this work. Moreover, any model is a simplification of reality. By using insights from behavioural sciences, we can build more informed models and may therefore arrive at more interesting conclusions but we will not fully capture the complex data that arise from people's activities with task systems.

A second limitation is a requirement that the behavioural hypotheses make predictions over the observable action space of the participants. For example, the goal-gradient hypothesis from Chapter 3 predicts that a users' rate of activity will increase as a goal is approached. Activity in the context of Stack Overflow was defined as the number of actions per day (where an action can be to: *ask* a question, *answer* a question, *vote* on a post etc). This is not always possible or the data might not be available for the hypothesis that is under investigation. For example, if a certain hypothesis predicts a change of a person's emotional state, or if it predicts a change to an action that is not seen on the task system, then we are unable to model these cases. Moreover, privacy restrictions make access difficult to these data for researchers and thus it is a challenge in the future to gain access to the sensitive interaction data of users without violating privacy rights.

6.4 Future Work

The work explored in this thesis contains many avenues for possible future work. I detail two of these possibilities that I find most exciting.

First, I have shown that users can be efficiently segmented by their behaviour. I did not show that this segmentation persists through time and that people's activity patterns repeat through time. For example, if a user adheres to the goal-gradient hypothesis for one badge or threshold, it is unknown whether that makes the same user more likely to do this again. Excitingly, Kivetz et al. (2006) showed this to be true for customer's purchasing behaviour at coffee shops, however, it is unknown whether these effects will hold in an online domain. Intervention tests could be run to see whether a group of participants who have adhered to a hypothesis once, do it again when offered a similar reward. Research in this direction will help to answer the causal questions of how environment design changes people's behaviour.

Second, I have shown that we can use the data from task systems passively to infer what people do and how they interact. However, the very nature of the *partitioning* of a user base into different groups suggests that we should be able to tailor an environment to the needs of specific types of users. Indeed, Segal (2018) did exactly this by designing a reinforcement learning agent that sends different motivational messages to users based on the inferred needs of the user. I propose to extend the framework in this thesis such that tailored rewards can be designed for different types of users. The posterior assignment of a user's activity to a behavioural archetype will then allow the tailoring of certain rewards for these users. In this setting, the categorical latent space that we use for inference over hypotheses is reminiscent of the inference over a discrete action space that a reinforcement learning agent performs. As such, we can extend the insights from this thesis to provide interpretable and tailored incentivising agents that can help and motivate users when they perform work on a task system.

Bibliography

Alinier, G., Harwood, C., Harwood, P., Montague, S., Huish, E., Ruparelia, K., and Antuofermo, M. (2014). Immersive clinical simulation in undergraduate health care interprofessional education: Knowledge and perceptions. *Clinical Simulation in Nursing*, 10(4):e205–e216.

Allen, C., Balažević, I., and Hospedales, T. (2020). A probabilistic framework for discriminative and neuro-symbolic semi-supervised learning. *arXiv preprint arXiv:2006.05896*.

Amir, O. and Gal, Y. K. (2013). Plan recognition and visualization in exploratory learning environments. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3(3):16.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Anderson, A., Huttenlocher, D., Kleinberg, J., and Leskovec, J. (2013). Steering user behavior with badges. In *Proceedings of the 22nd international conference on World Wide Web*, pages 95–106. ACM.

Anderson, A., Huttenlocher, D., Kleinberg, J., and Leskovec, J. (2014). Engaging with massive online courses. In *Proceedings of the 23rd international conference on World wide web*, pages 687–698. ACM.

Anderson, J. R. and Peterson, C. (1987). A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019.

Bach, S. H., Broecheler, M., Huang, B., and Getoor, L. (2017). Hinge-loss markov random fields and probabilistic soft logic. *J. Mach. Learn. Res.*, 18(1):3846–3912.

Barrett, C., Sebastiani, R., Seshia, S. A., Tinelli, C., Biere, A., Heule, M., van Maaren, H., and Walsh, T. (2009). Handbook of satisfiability. *Satisfiability modulo theories*, 185:825–885.

Belle, V., Passerini, A., and Van den Broeck, G. (2015). Probabilistic inference in hybrid domains by weighted model integration. In *Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2015, pages 2770–2776.

Betancourt, M. (2017). A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blei, D. M. (2014). Build, compute, critique, repeat: Data analysis with latent variable models. *Annual Review of Statistics and Its Application*, 1:203–232.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022.

Blitzstein, J. K. and Hwang, J. (2019). *Introduction to probability*. Chapman and Hall/CRC.

Bornfeld, B. and Rafaeli, S. (2017). Gamifying with badges: A big data natural experiment on stack exchange. *First Monday*, 22(6).

Box, G. E. and Hunter, W. G. (1962). A useful method for model-building. *Technometrics*, 4(3):301–318.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. (2013). Streaming variational bayes. *arXiv preprint arXiv:1307.6769*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Bunel, R., Turkaslan, I., Torr, P. H. S., Kohli, P., and Mudigonda, P. K. (2018). A unified view of piecewise linear neural network verification. *Advances in Neural Information Processing Systems*, pages 4795–4804.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

Chavira, M. and Darwiche, A. (2008). On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799.

Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., and Stewart, W. (2016). Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512.

Chou, Y.-k. (2019). *Actionable gamification: Beyond points, badges, and leaderboards*. Packt Publishing Ltd.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Darwiche, A. (2011). Sdd: A new canonical representation of propositional knowledge bases. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Darwiche, A. and Marquis, P. (2002). A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264.

De Moura, L. and Bjørner, N. (2008). Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. In *eprint arXiv:1702.08608*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., and Garcia, R. (2001). Incorporating second-order functional knowledge for better option pricing. *Advances in neural information processing systems*, pages 472–478.

Feng, Y., Martins, R., Wang, Y., Dillig, I., and Reps, T. W. (2017). Component-based synthesis for complex apis. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, page 599–612.

Fischer, M., Balunovic, M., Drachsler-Cohen, D., Gehr, T., Zhang, C., and Vechev, M. (2019). DL2: Training and querying neural networks with logic. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1931–1941.

Fox, E. B. (2009). *Bayesian nonparametric learning of complex dynamical phenomena*. PhD thesis, Massachusetts Institute of Technology.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2008). An hdp-hmm for systems with state persistence. In *Proceedings of the 25th international conference on Machine learning*, pages 312–319. ACM.

Fu, Z. and Su, Z. (2016). XSat: A Fast Floating-Point Satisfiability Solver. In *Proceedings of the 28th International Conference on Computer Aided Verification, Part II*, pages 187–209. Springer.

Ganchev, K., Graça, J., Gillenwater, J., and Taskar, B. (2010). Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 11:2001–2049.

Geller, S. A., Hoernle, N., Gal, K., Segal, A., Zhang, A. X., Karger, D., Facciotti, M. T., and Igo, M. (2020). # confused and beyond: detecting confusion in course forums using students' hashtags. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 589–594.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. Chapman and Hall/CRC.

Girolami, M. and Calderhead, B. (2011). Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214.

Glynn, P. W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Graça, J. V., Ganchev, K., and Taskar, B. (2007). Expectation maximization and posterior constraints.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Hammond, L. and Belle, V. (2021). Learning tractable probabilistic models for moral responsibility and blame. *Data Mining and Knowledge Discovery*, 35(2):621–659.

Hinton, G. E. and Van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13.

Hoernle, N., Gal, K., Grosz, B., Lyons, L., Ren, A., and Rubin, A. (2020a). Interpretable models for understanding immersive simulations. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2319–2325. International Joint Conferences on Artificial Intelligence Organization. Main track.

Hoernle, N., Gal, K., Grosz, B., Protopapas, P., and Rubin, A. (2018). Modeling the effects of students' interactions with immersive simulations using markov switching systems. In *Educational Data Mining (EDM), Buffalo, USA, July 2018.*

Hoernle, N., Kehne, G., Procaccia, A. D., and Kobi, G. (2020b). The phantom steering effect in q&a websites. *20th IEEE International Conference on Data Mining*.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Hoffman, M. D. and Gelman, A. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623.

Hu, Z., Ma, X., Liu, Z., Hovy, E., and Xing, E. (2016). Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420.

Hull, C. L. (1932). The goal-gradient hypothesis and maze learning. *Psychological Review*, 39(1):25.

Immorlica, N., Stoddard, G., and Syrgkanis, V. (2015). Social status and badge design. In *Proceedings of the 24th international conference on World Wide Web*, pages 473–483.

Ipeirotis, P. G. and Gabrilovich, E. (2014). Quizz: targeted crowdsourcing with a billion (potential) users. In *Proceedings of the 23rd international conference on World Wide Web*, pages 143–154. ACM.

Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jha, S., Gulwani, S., Seshia, S. A., and Tiwari, A. (2010). Oracle-guided component-based program synthesis. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1*, page 215–224.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kahneman, D. (2011). *Thinking, fast and slow*. Macmillan.

Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. (2017). Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations (ICLR)*.

Kivetz, R., Urminsky, O., and Zheng, Y. (2006). The goal-gradient hypothesis resurrected: Purchase acceleration, illusionary goal progress, and customer retention. *Journal of Marketing Research*, 43(1):39–58.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Kusmierczyk, T. and Gomez-Rodriguez, M. (2018). On the causal effect of badges. In *Proceedings of the 2018 World Wide Web Conference*, pages 659–668. International World Wide Web Conferences Steering Committee.

Lage, I., Ross, A., Gershman, S. J., Kim, B., and Doshi-Velez, F. (2018). Human-in-the-loop interpretability prior. In *Advances in Neural Information Processing Systems*, pages 10159–10168.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Li, Z., Huang, K.-W., and Cavusoglu, H. (2012). Quantifying the impact of badges on user engagement in online q&a communities. In *International Conference on Information Systems*.

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

MacKay, D. J. (1997). Ensemble learning for hidden markov models. Technical report, Citeseer.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2017). The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations (ICLR)*.

Mallavarapu, A., Lyons, L., Uzzo, S., Thompson, W., Levy-Cohen, R., and Slattery, B. (2019). Connect-to-connected worlds: Piloting a mobile, data-driven reflection tool for an open-ended simulation at a museum. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 7. ACM.

Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and De Raedt, L. (2018). Deepproblog: Neural probabilistic logic programming. *Advances in Neural Information Processing Systems*, 31:3749–3759.

Mnih, A. and Gregor, K. (2014). Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799. PMLR.

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60.

Mutter, T. and Kundisch, D. (2014). Behavioral mechanisms prompted by badges: The goal-gradient hypothesis. In *International Conference on Information Systems*.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814.

Neal, R. M. (1993). *Probabilistic inference using Markov chain Monte Carlo methods*. Department of Computer Science, University of Toronto Toronto, ON, Canada.

Neal, R. M. and Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer.

Ortega, J. M. and Rheinboldt, W. C. (2000). *Iterative solution of nonlinear equations in several variables*. SIAM.

Osera, P.-M. (2019). Constraint-based type-directed program synthesis. In *Proceedings of the 4th ACM SIGPLAN International Workshop on Type-Driven Development*, page 64–76.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2019). Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822.

Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Rosenfeld, A. and Richardson, A. (2019). Explainability in human-agent systems. *Auton. Agents Multi Agent Syst.*, 33(6):673–705.

Ross, A. and Doshi-Velez, F. (2018). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ross, A. S., Hughes, M. C., and Doshi-Velez, F. (2017). Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, pages 2662–2670.

Segal, A. (2018). *Intelligent Intervention Design in Large Scale Task Systems*. PhD thesis, Ben-Gurion, Israel.

Shillo, R., Hoernle, N., and Gal, K. (2019). Detecting creativity in an open ended geometry environment. *International Educational Data Mining Society*.

Simpson, R., Page, K. R., and De Roure, D. (2014). Zooniverse: observing the world's largest citizen science platform. In *Proceedings of the 23rd international conference on World Wide Web*, pages 1049–1054. ACM.

Smørdal, O., Slotta, J., Moher, T., Lui, M., and Jornet, A. (2012). Hybrid spaces for science learning: New demands and opportunities for research. In *International Conference of the Learning Sciences. Sydney, Australia.*

Solar-Lezama, A. (2009). The sketching approach to program synthesis. In *Proceedings of the 7th Asian Symposium on Programming Languages and Systems*, page 4–13.

Stewart, R. and Ermon, S. (2017). Label-free supervision of neural networks with physics and domain knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Takeishi, N. and Kawahara, Y. (2020). Knowledge-based regularization in generative modeling. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, pages 2390–2396.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2005). Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, D., and Sohl-Dickstein, J. (2017). Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *arXiv preprint arXiv:1703.07370*.

Turner, R. E. and Sahani, M. (2011). Two problems with variational expectation maximisation for time-series models.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Wu, M., Hughes, M. C., Parbhoo, S., Zazzi, M., Roth, V., and Doshi-Velez, F. (2018). Beyond sparsity: Tree regularization of deep models for interpretability. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Xu, J., Zhang, Z., Friedman, T., Liang, Y., and Van den Broeck, G. (2018). A semantic loss function for deep learning with symbolic knowledge. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5502–5511.

Yanovsky, S., Hoernle, N., Lev, O., and Gal, K. (2019). One size does not fit all: Badge behavior in q&a sites. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, pages 113–120. ACM.

Yanovsky, S., Hoernle, N., Lev, O., and Gal, K. (2021). One size does not fit all: A study of badge behavior in stack overflow. *Journal of the Association for Information Science and Technology*, 72(3):331–345.

Zagoruyko, S. and Komodakis, N. (2016). Wide Residual Networks. In *British Machine Vision Conference*.

Zhang, H., Wang, S., Chen, T.-H., and Hassan, A. E. (2019). Reading answers on stack overflow: Not enough! *IEEE Transactions on Software Engineering*.

Appendix A

Background and Preliminaries

A.1 Introduction

This chapter presents some important preliminaries that I build upon throughout the thesis. I introduce the theory of variational inference and I provide an overview of how this is extended to the black-box and amortised inference that is used throughout my work.

A theme throughout my thesis is the use of structured inference algorithms in latent variable models. Specifically, we endow certain latent parameters with meaning in the model (e.g., a cluster assignment variable that assigns a data point to one of the competing hypotheses, or the variables that control a user's response to the goal-gradient hypothesis). The corresponding posterior inference algorithms in these models are therefore of importance and choosing the most applicable and scalable inference algorithms is an important decision for a researcher in this field. Furthermore, in the context of Figure 1.1, the inference algorithms let us analyse data under the modelling assumptions where the behavioural hypotheses are encoded. Inference in these models, therefore, uncovers the hidden structure that best explains our observations.

In Appendix A.2, I present some challenges that exist for any algorithm that aims to perform inference in a latent variable model. These are broad computational intractability problems and any posterior inference algorithm (including the variational inference that we employ) makes assumptions and simplifications to overcome these hurdles. I then present a simplified perspective of variational inference in Appendix A.3 and Appendix A.4. This allows us to understand the amortised inference approach in Appendix A.5.

A.2 Challenges for Inference Algorithms in Latent Variable Models

Latent variable models are primarily concerned with Bayes Theorem, presented in Equation (A.1). To compute the quantity of interest $p(\theta \mid x)$ (also know as the *posterior* distribution), we require the following elements:

1. $p(x \mid \theta)$ is also called the *likelihood* and it can often be represented as a graphical model. Here, the observed data x are controlled by the latent variable(s) θ, hence the naming convention for these models as latent variable models.

2. $p(\theta)$ is referred to as the *prior* as it contains a scientist's beliefs about the pa-

rameters θ *before* any data have been observed.

3. $p(x)$ is also called the *data likelihood*, the *evidence*, the *marginal likelihood* or even the “normalising constant”. Beware of using “normalising constant” in the context of latent variable models as this quantity, while constant with respect to θ, is the source of intractability when it comes to inference. I find the term “normalising constant” misrepresents the complexity that is hidden in this simple expression.

$$p(\theta \mid x) = \frac{p(x \mid \theta)p(\theta)}{p(x)} \tag{A.1}$$

I refer the reader to Sections 10.1 and 10.2 of Bishop (2006) for further details about Equation (A.1).

Solving for $p(\theta \mid x)$ requires designing a “generative model” which is the numerator in Equation (A.1). It then involves computing or approximating the entire expression to solve for $p(\theta \mid x)$. By construction, it is a safe assumption that the numerator is tractable. However, from the law of total probability,[1] the denominator can be seen to include an integration term:

$$p(x) = \int p(x \mid \theta)p(\theta)\partial\theta \tag{A.2}$$

While the integrand is now merely the numerator in Equation (A.1), we perform this integration over the often high dimensional latent variable space of θ. In certain limited cases, the integration problem in Equation (A.2) can be solved analytically. This is only possible when the likelihood term is in an exponential family and there exists a *conjugate prior* such that the posterior is in the same family as the likelihood. While, models of this type can lead to some possibilities (including Gaussian-Gaussian, Poisson-Gamma, Bernoulli-Beta and Multinomial-Dirichlet models) it does not allow direct inference in a large class of other useful models (including Bayesian mixture models, hidden Markov models, linear dynamic systems, matrix factorisation, Dirichlet process mixtures, mixed-membership models). It is thus a very limiting restriction to place on the modelling step to restrict ourselves to conjugate models.

We, therefore, seek a general inference framework that can handle many types of likelihood specifications with a range of different choices for the prior. Options include Monte Carlo approximation, which is a computationally expensive means of

[1] See Blitzstein and Hwang (2019) for a detailed review of the necessary probability theory.

performing the integration in Equation (A.2). Specifically, Markov Chain Monte Carlo (MCMC) methods such as Hamiltonian Monte Carlo (HMC) (Girolami and Calderhead, 2011; Betancourt, 2017) and the later No U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014) have been shown to efficiently draw uncorrelated posterior samples and these methods have excellent convergence guarantees. However, while we know for certain that the samples drawn from an MCMC algorithm will converge, in the limit, to the posterior, we do not have strict guarantees on the number of samples (and thus the time) that will be required to achieve this goal. Often a solution can be arbitrarily bad and it is very hard to diagnose when this is the case, and how bad the solution is. Certain MCMC algorithms (such as HMC and NUTS) do provide good diagnostic tools which help to identify when the samples have not converged to the true posterior.

Other methods for *approximate posterior inference* include Laplace's method and variational inference. The rest of this chapter is devoted to explaining variational inference.

A.3 Variational Inference

Variational inference primarily turns the integration problem of solving Equation (A.2), into an optimisation problem. This was an incredibly exciting development as it opened the field of posterior inference to the possibility of using techniques from the vast optimisation literature (Robbins and Monro, 1951; Dempster et al., 1977; Moon, 1996; Ortega and Rheinboldt, 2000; Broderick et al., 2013)

To achieve this, field pioneers (including Anderson and Peterson (1987), Jordan et al. (1999), Hinton and Van Camp (1993), Neal and Hinton (1998), MacKay (1997), Blei et al. (2003) and Hoffman et al. (2013)) defined a new problem that introduces a family of distributions, parameterised by a set of new variables that allow for optimisation. The new family often introduces independencies that are not present in the original model, thus the solution becomes an approximation to the original problem.

Variational inference thus introduces $q(\theta;\eta)$ that aims to approximate the posterior distribution $p(\theta \mid x)$. We aim to make $q(\theta;\eta)$ "look" as similar as possible to $p(\theta \mid x)$ by minimizing some measure of distance between these two distributions. See Figure A.1 where I present a pictorial representation of this. We propose a family of distributions, over the same latent variables θ and parameterised by some free parameters: $q(\theta;\eta)$. The true posterior might be intractable and thus would live outside of

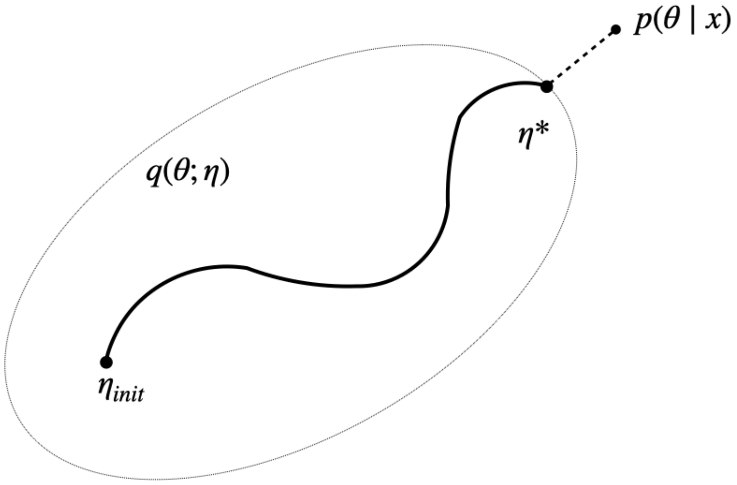


Figure A.1: Pictorial representation of variational inference. The posterior is an intractable object, living outside of a tractable family of distributions that are parameterised by η. We optimise the parameters such that the solution $q(\theta;\eta^*)$ is as "close" as possible to the true posterior.

the family that is proposed. Our goal is to optimise the parameters η (starting from an initialisation point, η_{init}) to find the best parameters η^* where the variational approximation $q(\theta;\eta^*)$ is closest to the true posterior .

The concept of "distance" between two distributions is another point of much research and even more debate. For example, consider Figure A.2 where $q_1(x)$ and $q_2(x)$ have the same variance but very different means. In contrast to this, $q_3(x)$ and $q_4(x)$ have the same means but their variances differ. Is it the case that $q_1(x)$ and $q_2(x)$ are more similar to each other or is $q_3(x)$ more similar to $q_4(x)$? This is an unsolved problem and it requires defining a measure that can be used to compare the similarity between distributions. The choice of measure that is used for comparison necessarily changes the answer to this question.

Many measures exist to compare distributions (e.g., Wasserstein metric (Gretton et al., 2012) and Jensen–Shannon divergence). However, one of the more popular measures is the Kullback-Leibler (KL) divergence. Note that it is called a "divergence" and not a "distance": an important technical difference as the KL divergence is not symmetric. Thus, it matters if we take the KL divergence from q_1 to q_2 (denoted $KL(q_1||q_2)$) or the reverse KL divergence from q_2 to q_1 (denoted $KL(q_2||q_1)$). In standard variational inference, we adopt the reverse KL divergence between the posterior and the approximating distribution; we thus minimise Equation (A.3). For reasons be-

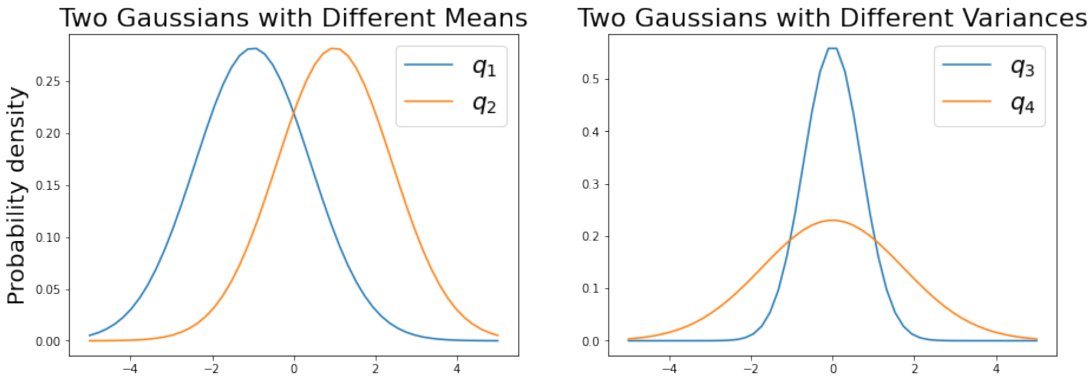


Figure A.2: Example showing two cases with different Gaussian distributions. It is not immediately clear if q_1 is more different from q_2 than q_3 is to q_4. We need a well defined notion of "different" to solve this issue.

yond the scope of the present discussion, this results in a "mode-seeking" property of variational inference.[2] This property is well explained and further explored by Turner and Sahani (2011).

$$\begin{aligned} q^* &= \arg\min_{\eta} KL(q(\theta;\eta)||p(\theta \mid x)) \\ &= \arg\min_{\eta} \int q(\theta;\eta) \log \frac{q(\theta;\eta)}{p(\theta \mid x)} \partial\theta \end{aligned} \tag{A.3}$$

By substituting the solution of Bayes' formula for the posterior we can simplify Equation (A.3).

$$\begin{aligned} q^* &= \arg\min_{\eta} \int q(\theta;\eta) \log \frac{q(\theta;\eta)p(x)}{p(x \mid \theta)p(\theta)} \partial\theta \\ &= \arg\min_{\eta} \left(\int q(\theta;\eta) \log p(x) \partial\theta + \int q(\theta;\eta) \log \frac{q(\theta;\eta)}{p(x \mid \theta)p(\theta)} \partial\theta \right) \\ &= \arg\min_{\eta} \left(\log p(x) - \int q(\theta;\eta) \log \frac{p(x \mid \theta)p(\theta)}{q(\theta;\eta)} \partial\theta \right) \end{aligned} \tag{A.4}$$

The solution in Equation (A.4) contains one term, $\log p(x)$, that does not depend on the distribution q (parameterised by η) and thus it is not involved in the optimisation.

[2] Suppose that we wish to approximate a multimodal distribution with a single Gaussian. The mode seeking behaviour of the reverse KL will result in the single Gaussian finding one of the modes and placing as much mass as possible where that target mode exists. It will entirely "miss" the other modes in the original distribution. In contrast to this, the forward KL divergence would display behaviour that seeks to include all of the modes of the original distribution, at the expense of placing mass where it might not exist in the original distribution. See Bishop (2006) for graphical representations and a further discussion.

The second term, $-\int q(\theta;\eta)\log\frac{p(x|\theta)p(\theta)}{q(\theta;\eta)}\partial\theta$, is called the Evidence Lower BOund (ELBO) and it is the objective for optimisation in variational inference. We will use this same "variational objective" many times throughout the thesis and thus this result is of great importance for us:

$$ELBO(\eta) := \int q(\theta;\eta)\log\frac{p(x \mid \theta)p(\theta)}{q(\theta;\eta)}\partial\theta \tag{A.5}$$

Since in Equation (A.4), we aim to minimise the -ELBO, in general, in variational inference, we aim to maximize Equation (A.5). This naming convention for the "Evidence Lower Bound" can be further understood by studying the form of the KL divergence:

$$KL(q(\theta;\eta)||p(\theta \mid x)) = \log p(x) - ELBO(\eta) \tag{A.6}$$

Concretely, the KL-divergence is non-negative by definition, and $logp(x)$ is called the "log-evidence". Therefore, it holds that $ELBO(\eta)$, forms a lower bound on the log-evidence: $\log p(x)$. This is shown in Equation (A.7).

$$\begin{aligned}\log p(x) &= KL(q(\theta;\eta)||p(\theta \mid x)) + ELBO(\eta)\\ \log p(x) &\geq ELBO(\eta)\end{aligned} \tag{A.7}$$

Therefore variational inference can be understood in both of the following contexts:

1. Minimise the KL-divergence between the approximating function $q(\theta;\eta)$ and the posterior $p(\theta \mid x)$ (as is pictorially represented in Figure A.1).

2. Maximize the ELBO: the lower bound to the log-evidence $logp(x)$ (as it shown in Equation (A.7)). It is this second reason that maximising the ELBO can sometimes be referred to as a maximum likelihood technique.

In practice the solution to a variational inference problem can be found efficiently and is often very good, however, a major criticism of this approach is that very few tools exist to evaluate the quality of an optimised result. If a poor approximation family is chosen, the result will be poor; moreover, the problem is often highly non-convex and thus local optima can result in poor solutions, and most worryingly, it is often very hard to evaluate just how poor these solutions are. On a technical level, these are all problems with the variational inference approach that, to date, are open research questions.

A.4 Black Box Variational Inference

Given the variational objective (the ELBO) in Equation (A.5), we can additionally study its components:

$$\begin{aligned} ELBO(\eta) &= \int q(\theta;\eta)\log\frac{p(x\mid\theta)p(\theta)}{q(\theta;\eta)}\partial\theta \\ &= \int q(\theta;\eta)\log p(x,\theta) - q(\theta;\eta)\log q(\theta;\eta)\partial\theta \end{aligned} \tag{A.8}$$

Equation (A.8) is exactly the definition of the expected value of $\log p(x,\theta) - \log q(\theta;\eta)$ under the distribution $q(\theta;\eta)$. As such, we aim to optimize:

$$ELBO(\eta) = \mathbb{E}_{q(\theta;\eta)}\left[\log p(x,\theta) - \log q(\theta;\eta)\right] \tag{A.9}$$

Equation (A.9) gives the target for optimisation but we have still not covered *how* to perform this optimisation. To perform gradient-based optimisation, we require the gradient of the ELBO.

$$\nabla_\eta ELBO(\eta) = \nabla_\eta \mathbb{E}_{q(\theta;\eta)}\left[\log p(x,\theta) - \log q(\theta;\eta)\right] \tag{A.10}$$

The analytical computation of the expectation term remains a problem (traditionally solved via lengthy derivations for problem-specific solutions e.g., see Blei et al. (2003) for an example of a popular mixed-membership model), but by evaluating the gradient first, we can overcome many of these difficulties. The derivation of the computation is given below, where we let $f(\theta,\eta) = \log p(x,\theta) - \log q(\theta;\eta)$. I have use the differentiation product rule in Equation (A.12), and I have used the log-derivative identity[3] in line Equation (A.13).

$$\nabla_\eta ELBO(\eta) = \int \nabla_\eta\left[f(\theta,\eta)q(\theta;\eta)\right]\partial\theta \tag{A.11}$$

$$= \int q(\theta;\eta)\nabla_\eta f(\theta,\eta) + \nabla_\eta q(\theta;\eta)f(\theta,\eta)\partial\theta \tag{A.12}$$

$$= \int q(\theta;\eta)\nabla_\eta f(\theta,\eta) + q(\theta;\eta)\nabla_\eta \log q(\theta;\eta)f(\theta,\eta)\partial\theta \tag{A.13}$$

$$= \mathbb{E}_{q(\theta;\eta)}\left[\nabla_\eta f(\theta,\eta) + \nabla_\eta \log q(\theta;\eta)f(\theta,\eta)\right] \tag{A.14}$$

[3] This is a simple application of the differentiation chain rule: $\nabla_x \log f(x) = \frac{\nabla_x f(x)}{f(x)}$. However, when the function f, is a likelihood function, $\nabla_x \log f(x)$ becomes a very useful expression and is therefore called the "score function". This significance of the score function is further discussed in: https://blog.shakirm.com/2015/11/machine-learning-trick-of-the-day-5-log-derivative-trick/

Equation (A.14) is a very general means for obtaining the derivative of the ELBO which permits gradient-based optimisation. Moreover, we are able to *estimate* the expectation in Equation (A.14) via Monte Carlo samples but to do this we are required to draw samples from $q(\theta;\eta)$.

There is a choice of two classes of algorithm that allow us to approximate the expectation in Equation (A.14). First, we could use the identity that the expectation of the score function is 0 ($\mathbb{E}_{q(\theta;\eta)}[\nabla_\eta \log q(\theta;\eta)] = 0$). This leads to an unbiased estimator that can be approximated via Monte Carlo samples and is used heavily in reinforcement learning under the name: "likelihood ratio" (Glynn, 1990) or REINFORCE (Williams, 1992; Ranganath et al., 2014; Mnih and Gregor, 2014). While this approach is outside of the present scope, there is much promising research in this direction (Tucker et al., 2017); however, a note is that the resulting estimator has a high variance and thus many of the practical approaches to solving this problem attempt to reduce the variance of the estimator.

The alternative approach, and the one that we use throughout this thesis, is called the "pathwise estimator". This approach requires that the distribution $q(\theta;\eta)$ can be re-written as a deterministic and differentiable transformation of a parameter-free noise source (denoted $s(\varepsilon)$). For example, suppose $q(\theta;\eta)$ is a Gaussian with parameters $\eta = (\mu,\sigma)$. $\theta \sim q(\theta;\eta)$ can then be constructed by a sample from a standard Gaussian ($\varepsilon \sim N(0,1)$) and transformed via a differentiable and deterministic mapping: $\theta = \mu + \varepsilon\sigma$.

Finally, assuming $\log p(x,\theta)$ and $\log q(\theta;\eta)$ are differentiable with respect to the latent variable θ, we are able to "reparameterise" the expectation in Equation (A.14). We then arrive at the pathwise estimator (where this reparameterisation is also known as the "reparameterisation trick" (Kingma and Welling, 2014; Rezende et al., 2014; Kingma et al., 2014, 2016)). We use the function $\theta = g(\varepsilon,\eta)$ to refer to the deterministic transformation of the parameter-free noise source $\varepsilon \sim s(\varepsilon)$ using parameters η to produce a sample from the distribution $q(\theta;\eta)$.

$$\begin{aligned}
\nabla_\eta ELBO(\eta) &= \mathbb{E}_{q(\theta;\eta)}[\nabla_\eta f(\theta,\eta) + \nabla_\eta \log q(\theta;\eta) f(\theta,\eta)] && \text{(A.15)}\\
&= \mathbb{E}_{s(\varepsilon)}[\nabla_\eta f(g(\varepsilon,\eta),\eta) + \nabla_\eta \log s(\varepsilon) f(g(\varepsilon,\eta),\eta)] && \text{(A.16)}\\
&= \mathbb{E}_{s(\varepsilon)}[\nabla_\eta f(g(\varepsilon,\eta),\eta)] && \text{(A.17)}\\
&= \mathbb{E}_{s(\varepsilon)}\left[\nabla_\theta\big(\log p(x,\theta) - \log q(\theta;\eta)\big)\nabla_\eta g(\varepsilon,\eta)\right] && \text{(A.18)}
\end{aligned}$$

Equation (A.17) follows as the noise distribution $s(\varepsilon)$ does not depend on the pa-

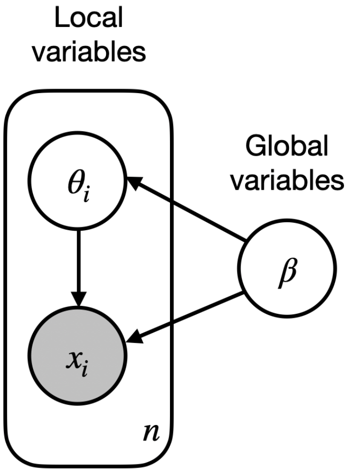


Figure A.3: General graphical model for a large class of generative models commonly seen in the literature. We show the clear split of local and global latent variables.

rameters η and thus the gradient with respect to η is 0. Equation (A.18) follows from another application of the chain rule. We thus arrive at a specification for computing the gradient of the ELBO and we only require that:

1. The distribution $q(\theta;\eta)$ can be re-written as a sample from a noise source $\varepsilon \sim s(\varepsilon)$ and transformed via a deterministic mapping: $\theta = g(\varepsilon, \eta)$.

2. The generative model $\log p(x,\theta)$ is differentiable with respect to the latent variable θ. This allows us to compute $\nabla_\theta \log p(x,\theta)$.

3. The approximating distribution $\log q(\theta;\eta)$ is differentiable with respect to the latent variable θ. This allows us to compute $\nabla_\theta \log q(\theta;\eta)$.

A.5 Amortised Variational Inference

We have now arrived at a general specification for optimizing the ELBO for a flexible range of models p and approximating distributions q. The final piece of the puzzle is to improve learning by introducing amortisation across data.

Figure A.3 presents a pictorial representation of a typical generative model (covering a large range of models that are often encountered in practice including the list of models presented in Appendix A.2). The figure shows that each data point x_i has a

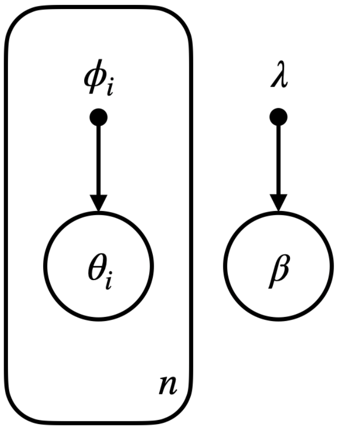


Figure A.4: Graphical model for the mean field approximation of the generative model in Figure A.3

corresponding *local* latent variable θ_i. The *global* latent variables β are shared among all local latent variables and all data points. Traditionally, variational inference loops through an entire dataset updating each data point's local latent variable, thereafter the global latent variable is updated conditioned on the values for the local latent variable (this results in an algorithm that is highly related to expectation maximisation (Neal and Hinton, 1998)). Hoffman et al. (2013) was the first to propose a *stochastic* variational inference alternative where the local latent variables are updated in mini-batches and the global variables are updated after each mini-batch. In practice, their algorithm scaled to data sets orders of magnitude larger than the original algorithm. Amortised inference, in turn, learns a new function that takes as input a data point x_i and outputs the parameters associated with the variational approximation for the local latent variables for that data point. Thus, the problem is now to learn the mapping from data point x_i to the variational approximation for θ_i.

Figure A.4 presents the mean-field variational approximation for the general family of generative models in Figure A.3. Each latent variable has its own variational parameter with the local latent variables θ_i each having their own ϕ_i variational parameter. Similarly, the global variables β have their variational parameters λ. Thus, in this case the variational parameters $\eta = (\lambda, \phi_1, \ldots, \phi_n)$.

We can finally return to the definition of the ELBO from Equation (A.9). Adopting the structure from Figure A.3, the ELBO can be expanded by the implied independencies from the variational approximation in Figure A.4.

$$ELBO(\lambda, \phi_{1,\dots,n}) = \mathbb{E}_q\left[\log p(x, \theta, \beta)\right] - \mathbb{E}_q\left[\log q(\beta; \lambda) + \sum_{i=1}^{n} q(\theta_i; \phi_i)\right] \tag{A.19}$$

Equation (A.19) presents the target for optimisation that is used by Hoffman et al. (2013). Here, we optimise, individually, each parameter ϕ_i based on the data point x_i and the global parameters λ. Not only is this inefficient in that for every new data point we must individually optimise its local parameters, but it also becomes problematic for large data sets (where streaming data may be required) and storing and indexing each of the local parameters ϕ_i is required. Amortised inference rather introduces an *inference network* that maps a data point x_i to its local latent parameters ϕ_i. This change is shown in Equation (A.20) in blue.

$$\begin{aligned} ELBO(\lambda, \phi_{1,\dots,n}) = \mathbb{E}_q\left[\log p(x, \theta, \beta)\right] \\ - \mathbb{E}_q\left[\log q(\beta; \lambda) + \sum_{i=1}^{n} q(\theta_i; \phi_i = h_\Phi(x_i))\right] \end{aligned} \tag{A.20}$$

Here, we now have an inference network h_Φ that accepts a data point x_i and outputs the variational parameters ϕ_i that are associated with that data point. The network that implements q can now be optimised jointly with the generative model that implements p. This setup in Equation (A.20) has been commonly used in the literature and is referred to as a “Variational Autoencoder” (Kingma and Welling, 2014).

Appendix B

Chapter 1 Appendices

B.1 Omitted Proofs

Here we present the proof of Theorem 3.6.1. Let X be a nonnegative, bounded, and integer-valued random variable. Let $\{X_m\}_{m\in\mathbb{N}}$ be independent random variables which are distributed identically to X. We will be concerned with the partial sums $S_n = \sum_{m=1}^{n} X_m$. Let Y_N denote the random variable which is the copy X^m that brings S_n across the threshold N; that is, for which $S_{m-1} < N$ and $S_m \geq N$.

Theorem B.1.1. *If X is nonnegative, integer-valued, and bounded then*

$$\lim_{N\to\infty} \mathbb{E}[Y_N] = \mathbb{E}[X] + \frac{Var[X]}{\mathbb{E}[X]}$$

More generally, we also consider the case when the X are drawn from distributions $X^1,\ldots,X^\tau,\ldots,X^T$ repeatedly in turn. Then the partial sums are $S_n = \sum_{m=1}^{n} X^{m \mod T}$, where all copies of X^τ are independent. Let ξ_τ denote the event that Y_N is drawn from distribution X^τ, and let $Z = \sum_{\tau=1}^{D} X^\tau$. For this setting we have the following theorem:

Theorem B.1.2. *If each of the distributions X^τ is finite, nonzero, nonnegative, and integer valued then*

$$\lim_{N\to\infty} \mathbb{E}[Y_N] = \frac{\sum_\tau \mathbb{E}[(X^\tau)^2]}{\mathbb{E}[Z]}.$$

Theorem B.1.1 follows directly from Theorem B.1.2 by taking the X^τ to be identically distributed. It therefore suffices to prove Theorem B.1.2.

We begin by showing that the likelihood of the sequence $\{S_n\}$ visiting any given number N is asymptotically uniform. Let $p_m := \mathbb{E}\left[|\{n\in\mathbb{N} : S_n = m\}|\right]$, $g := \gcd(range(X))$ and observe that if $X > 0$ then $p_m = \Pr[m \in \{S_n\}]$. Also, if $m \notin g\mathbb{N}$ then clearly $p_m = 0$. For the p_m for which $m \in g\mathbb{N}$, we have the following lemma:

Lemma B.1.3. *If X is nonzero, nonnegative, and bounded then*

$$\lim_{n\to\infty} p_{gn} = \frac{g}{\mu}$$

Proof. First, it suffices to assume that $g = 1$. This is because the integer-valued random variable $X' := X/g$ has mean μ/g and $\gcd(range(X')) = 1$, and proving the claim for X' implies the claim for X. It also suffices to assume that $X > 0$. This is because the sequence $\{S_n\}_{n\in\mathbb{N}}$ remains at a specific value m only so long as the independent draws are $X_n = 0$, after which it leaves m forever. The expected number of steps that $\{S_n\}$ lingers at m for is exactly $\frac{1}{1-\alpha}$, where $\alpha = \Pr[X = 0]$. Since $\mu > 0$ by assumption, we may prove the claim for $X'' := X|X > 0$. Then $\mu = \frac{\mu''}{1-\alpha}$ and

$$p_m = \mathbb{E}\left[|\{n\in\mathbb{N} : S_n = m\}|\right] = \frac{1}{1-\alpha} p''_m$$

Thus proving the claim for X'', proves the claim for X. Therefore, we may assume without loss of generality that $X > 0$ and that $\gcd(range(X)) = 1$.

Let $M := \max\{range(X)\}$ be the maximum value that X obtains. Then the p_m obey the recurrence

$$p_m = \sum_{j=1}^{M} p_{m-j} \Pr[X = j] \tag{B.1}$$

with the initial conditions $p_0 = 1$ and $p_m = 0$ for all $m < 0$. Because X is bounded by M, we may break $\mathbb{N}$ up into "epochs" $\{1,\ldots,M\},\{M+1,\ldots,2M\},\ldots$, and then define $q_r^k := p_{kM+r}$ with $q^0 := (0,\ldots,0,1)^T$. For any $m = kM + r$ we can then iteratively expand the p_{m-j} terms in Equation B.1 for which $m - j \geq kM$ until the expression for each p_m depends only on the previous epoch, which gives an alternative recurrence of the form

$$p_{kM+r} = \sum_{s=1}^{M} \alpha_s^r \, p_{(k-1)M+s} \tag{B.2}$$

where $r,s \in [M]$ (and the initial conditions are the values of p_s for $s \in [M]$). Note that these α_s^r do not depend on k. The recurrences in Equations B.1 and B.2 give p_m as a convex combination of previous values, and so we may rewrite Equation B.2 as $q^k = A^k q^0$, where $A := \{\alpha_s^r\}_{r,s\in[M]}$ is a right stochastic square matrix. Furthermore it follows from the assumption $g = 1$ that A is primitive. Therefore the Perron-Frobenius Theorem implies that A^k converges exponentially quickly to a matrix of the form $\vec{1}\vec{u}^T$, where $\vec{1}$ and $\vec{u}^T$ are the unique right and left eigenvectors of A corresponding to the eigenvalue $\lambda = 1$. This in turn implies that $q^k = A^k q^0$ converges to some uniform vector $(\gamma,\ldots,\gamma)$, and therefore that $\lim_{m\to\infty} p_m = \gamma$.

Finally we argue that $\gamma = 1/\mu$. We can show this by considering $C(N,J) := \mathbb{E}\left[|\{S_n\} \cap [N,J)|\right]$, the mean number of times that $\{S_n\}$ intersects some interval $[N,J)$. Since the p_m converge, for fixed J we may use linearity of expectation to choose N large enough to guarantee that $C(N,J) \in J\gamma \pm \varepsilon$ for any given $\varepsilon > 0$. On the other hand, by considering the $\{S_n\}$ as "restarting" when they reach the epoch preceding N, we may use the central limit theorem to argue that $C(N,J) \in \frac{J}{\mu} \pm O(J^{2/3})$. Taking the limit as J becomes large yields $\gamma = 1/\mu$. ■ ■

B.2 Additional Plots from Badge Study

B.2.1 Civic Duty Badge

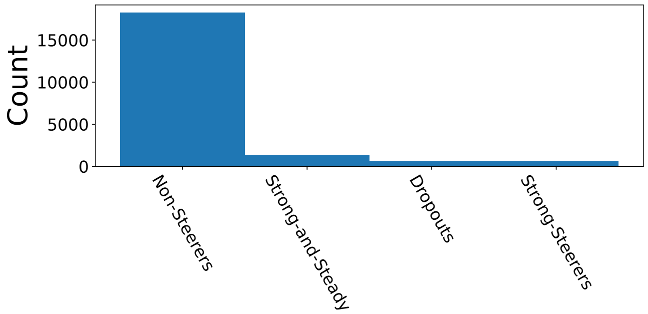


Figure B.1: Cluster assignments (as inferred by S_u from Model 2) for the users who achieved the Civic Duty badge.

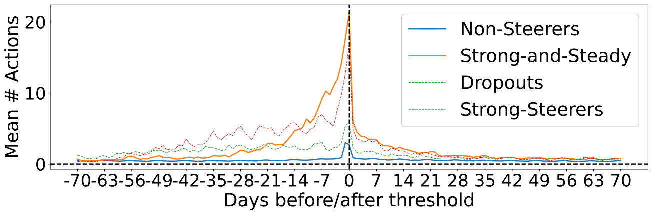


Figure B.2: Mean number of actions per day for users who are classified by their steering parameters (S_u) for the users who achieved the Civic Duty badge.

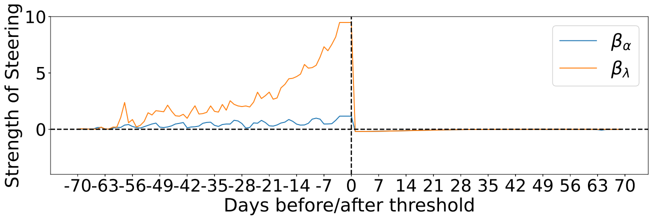


Figure B.3: Plot of the inferred magnitude of β: the expected deviation from a user's preferred distribution P_u under the assumptions of Model 2 for the users who achieved the Civic Duty badge.

B.2.2 Copy Editor Badge

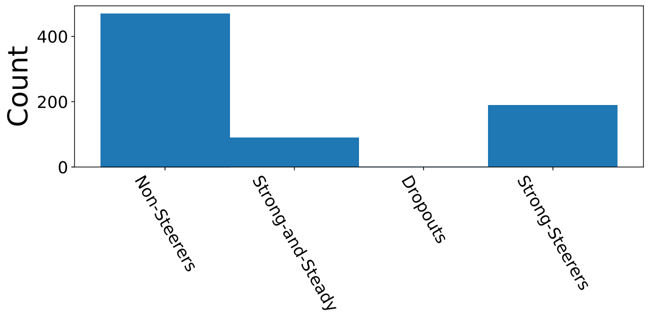


Figure B.4: Cluster assignments (as inferred by S_u from Model 2) for the users who achieved the Copy Editor badge.

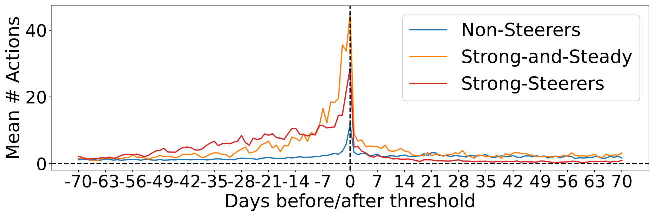


Figure B.5: Mean number of actions per day for users who are classified by their steering parameters (S_u) for the users who achieved the Copy Editor badge.

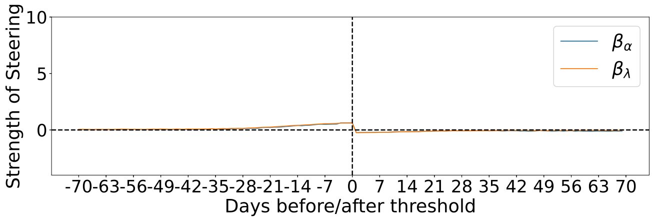


Figure B.6: Plot of the inferred magnitude of β: the expected deviation from a user's preferred distribution P_u under the assumptions of Model 2 for the users who achieved the Copy Editor badge.

B.2.3 Strunk & White Badge

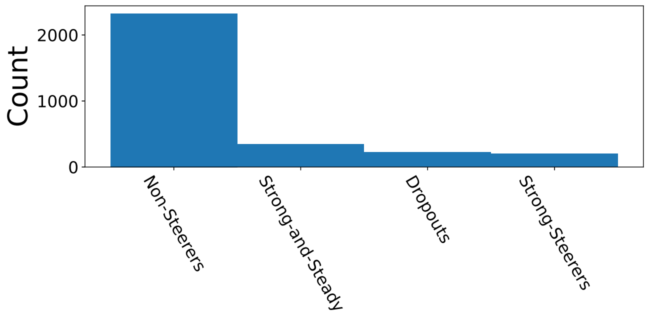


Figure B.7: Cluster assignments (as inferred by S_u from Model 2) for the users who achieved the Strunk & White badge.

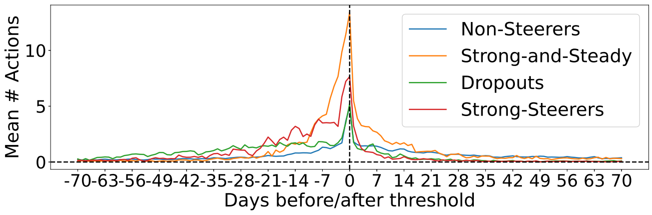


Figure B.8: Mean number of actions per day for users who are classified by their steering parameters (S_u) for the users who achieved the Strunk & White badge.

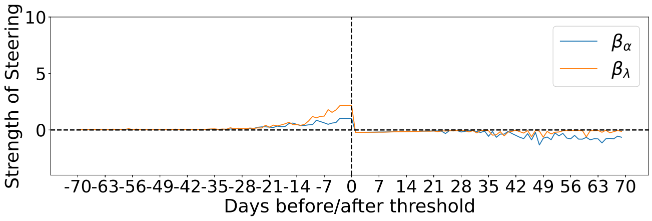


Figure B.9: Plot of the inferred magnitude of β: the expected deviation from a user's preferred distribution P_u under the assumptions of Model 2 for the users who achieved the Strunk & White badge.

B.3 Additional Plots from Reputation Threshold Study

B.3.1 Reputation Threshold = $2K$

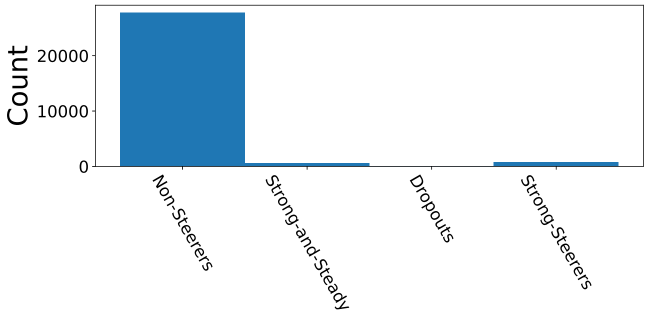


Figure B.10: Cluster assignments (as inferred by S_u from Model 2) for the users who passed the $2K$ reputation threshold.

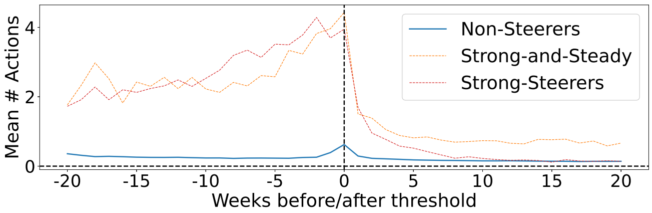


Figure B.11: Mean number of actions per day for users who are classified by their steering parameters (S_u) for the users who passed the $2K$ reputation threshold.

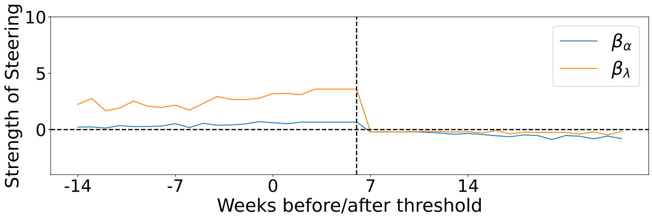


Figure B.12: Plot of the inferred magnitude of β: the expected deviation from a user's preferred distribution P_u under the assumptions of Model 2 for the users who passed the $2K$ reputation threshold.

B.3.2 Reputation Threshold = $20K$

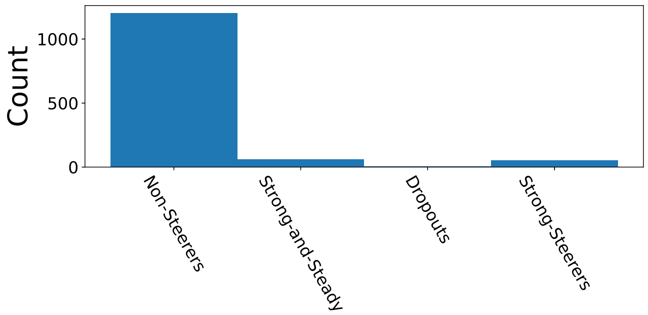


Figure B.13: Cluster assignments (as inferred by S_u from Model 2) for the users who passed the $20K$ reputation threshold.

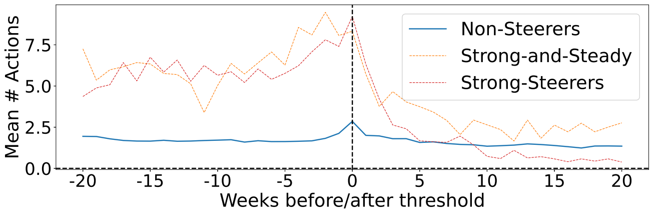


Figure B.14: Mean number of actions per day for users who are classified by their steering parameters (S_u) for the users who passed the $20K$ reputation threshold.

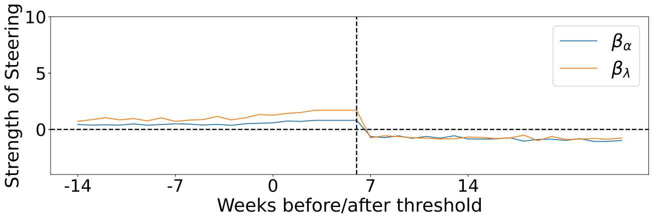


Figure B.15: Plot of the inferred magnitude of β: the expected deviation from a user's preferred distribution P_u under the assumptions of Model 2 for the users who passed the $20K$ reputation threshold.

B.3.3 Reputation Threshold = $25K$

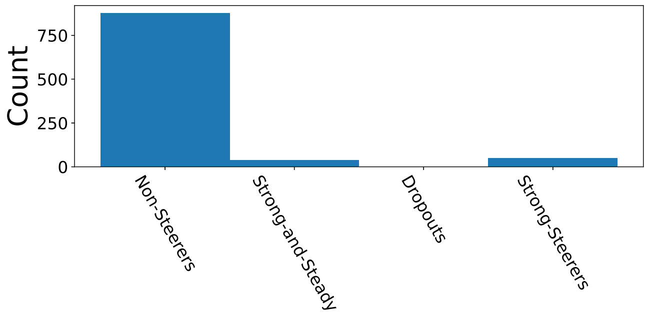


Figure B.16: Cluster assignments (as inferred by S_u from Model 2) for the users who passed the $25K$ reputation threshold.

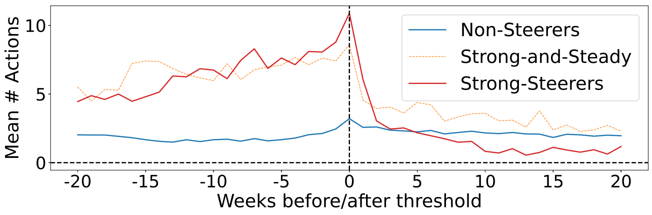


Figure B.17: Mean number of actions per day for users who are classified by their steering parameters (S_u) for the users who passed the $25K$ reputation threshold.

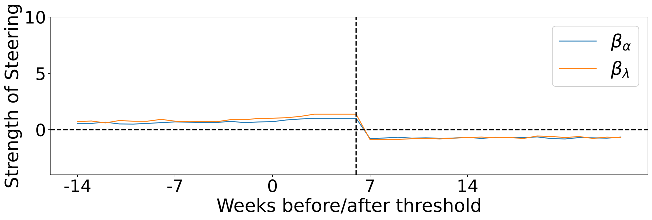


Figure B.18: Plot of the inferred magnitude of β: the expected deviation from a user's preferred distribution P_u under the assumptions of Model 2 for the users who passed the $25K$ reputation threshold.

B.4 Algorithms for Models 0, 1 and 2

Algorithm 2 Generative Pseudocode for Model 0

```
$Z_u \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Sample $Z$ from standard normal prior (see Section 3.3)}
$S_u = 0$ {Define $S_u$ to be 0 as no user steers}
$P_u = f_\theta(Z_u)$ {Compute $P_u$ as a forward pass of the network $f_\theta$}
Use Eq. (3.2) to compute $\alpha_u$ and $\lambda_u$ from $P_u$ and $S_u$.
Sample $A_u$ from zero-inflated Poisson likelihood Eq. (3.1)
```

Algorithm 3 Generative Pseudocode for Model 1

```
$Z_u \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Sample $Z$ from standard normal prior (see Section 3.3)}
$S_u = 1$ {Define $S_u$ to be 1 as all users steer}
$P_u = f_\theta(Z_u)$ {Compute $P_u$ as a forward pass of the network $f_\theta$}
Use Eq. (3.2) to compute $\alpha_u$ and $\lambda_u$ from $P_u$ and $S_u$.
Sample $A_u$ from zero-inflated Poisson likelihood Eq. (3.1)
```

Algorithm 4 Generative Pseudocode for Model 2

```
$Z_u \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ {Sample $Z$ from standard normal prior (see Section 3.3)}
$S_u \sim Bernoulli(\frac{1}{2})$ {Sample $S$ from Bernoulli Prior as a user can either steer or not (see Section 3.3)}
$P_u = f_\theta(Z_u)$ {Compute $P_u$ as a forward pass of the network $f_\theta$}
Use Eq. (3.2) to compute $\alpha_u$ and $\lambda_u$ from $P_u$ and $S_u$.
Sample $A_u$ from zero-inflated Poisson likelihood Eq. (3.1)
```

Appendix C

Chapter 2 Appendices

C.1 Appendix – Additional Experimental Details

All code and data for repeating the experiments can be found at the repository at: `https://github.com/NickHoernle/semantic_loss`. The image experiments (on MNIST and CIFAR100) were run on Nvidia GeForce RTX 2080 Ti devices and the synthetic data experiment was run on a 2015 MacBook Pro with processor specs: 2.2 GHz Quad-Core Intel Core i7. The MNIST data set is available under the *Creative Commons Attribution-Share Alike 3.0 license*; and CIFAR100 is available under the *Creative Commons Attribution-Share Alike 4.0 license*. The DL2 framework (available under an *MIT License*), used in the baselines, is available from `https://github.com/eth-sri/dl2`.

In all experiments, the data were split into a train, validation and test set where the test set was held constant across the experimental conditions (e.g., in the CIFAR100 experiment, the same test set was used to compare MultiplexNet vs the vanilla models vs the DL2 model). In cases where model selection was performed (early stopping on CIFAR100 and selection of the best runs from the MNIST experiment, we used the validation set to choose the best runs and/or models). In these cases the validation set was extracted from the training data set prior to training (with 10% of the data used for validation). The standard test sets, given by MNIST and CIFAR100 were used for those experiments and an additional test set was generated for the synthetic experiment.

C.1.1 Synthetic Data

Deriving the Loss

We first present the full derivation of the loss function that was used for this experiment. We used a variational autoencoder (VAE) with a standard isotropic Gaussian prior. The standard VAE loss is presented in Eq. C.1. In the below formulation, x_i is a datapoint, L is a minibatch size, and z_i is a sample from the approximate posterior q.

$$\mathscr{L}(\theta) = -\sum_{i=1}^{L} \log p(x_i \mid z_i) + \log p(z_i) - \log q(z_i \mid x_i) \tag{C.1}$$

We use an isotropic Gaussian likelihood for $\log p(x_i \mid z_i)$ and an isotropic Gaussian for the posterior. Standard derivations (see Kingma and Welling (2014) for more details) allow the loss to be expressed as in Eq.C.2. In this equation, f_θ is the decoder model and it predicts the mean of the likelihood term. A tunable parameter σ controls the precision of the reconstructions – this parameter was held constant for all experimental conditions. The posterior distribution is a Gaussian with parameters σ_q^2 and μ_q that are output from the encoding network $q_\theta(x_i)$.

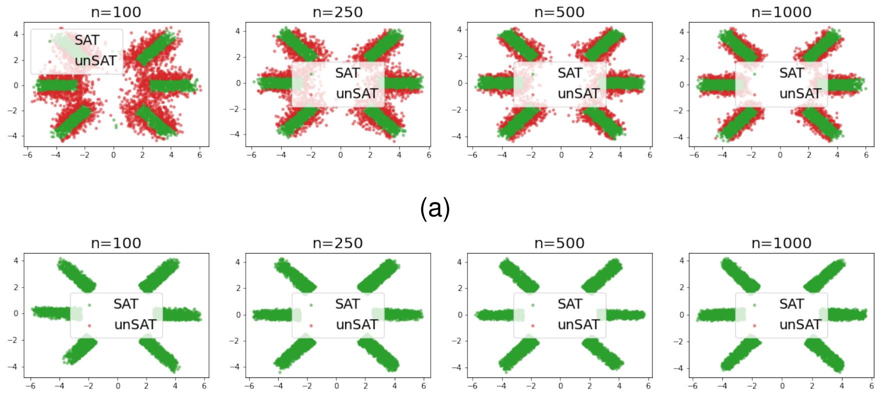
$$\mathscr{L}(\theta) = -\sum_{i=1}^{L} \mathcal{N}(x_i; f_\theta(z_i), \sigma^2) + 0.5 * (1 + \log(\sigma_q^2) - \mu_q^2 - \sigma_q^2) \tag{C.2}$$

Finally, we present how the MultiplexNet loss uses $\mathscr{L}(\theta)$ in the transformation of the output layer of the network. MultiplexNet takes as input the unconstrained network output $f_\theta(z_i)$ and it outputs the transformed (constrained) terms h_k (for K terms in the DNF constraint formulation) and the probability of each term π_k. Let $\mathscr{L}_k(\theta)$ denote the same loss $\mathscr{L}(\theta)$ from Eq. C.2 but with the raw output of the network $f_\theta(z_i)$ constrained by the constraint transformation h_k (i.e., the likelihood term becomes: $\mathcal{N}(x_i; h_k(f_\theta(z_i)), \sigma^2)$) The final loss then is presented in Eq. C.3.

$$MPlexNet(\theta) = \sum_{k=1}^{K} \pi_k \left(\mathscr{L}_{h_k}(\theta) + \log \pi_{h_k} \right) \tag{C.3}$$

Samples from the Posterior

The following plots show samples from the posterior of the two models – these are the attempts of the VAE to reconstruct the input data. It can clearly be seen that while MultiplexNet strictly adheres to the constraints, the baseline VAE approach fails to capture the constraint boundaries.



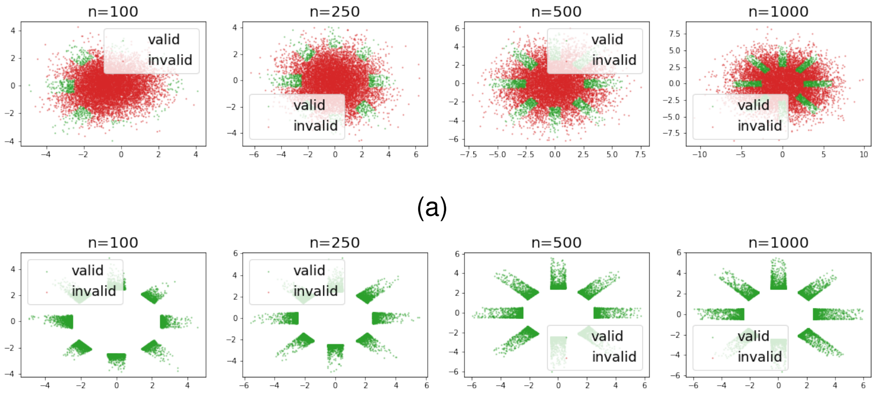
(a)

(b)

Figure C.1: (a) Samples from the vanilla VAE posterior for different sizes of training data sets (b) Samples from the MultiplexNet VAE posterior for different sizes of training data sets.

Samples from the Prior

Samples from the prior show how well a generative model has learnt the data manifold that it attempts to represent. We show these to demonstrate that in this case, the vanilla VAE fails to capture many of the complexities in the data distribution. To sample from the prior for MultiplexNet, we randomly sample from the latent Categorical variable from MultiplexNet. Hence, the two vertical modes (that contain no data in reality) have samples here. This can easily be solved by introducing a trainable prior parameter over the Categorical variable as well – an easy extension that we do not implement in this work.



(a)

(b)

Figure C.2: (a) Samples from the vanilla VAE prior for different sizes of training data sets (b) Samples from the MultiplexNet VAE prior for different sizes of training data sets.

Network Architecture

The default network used in these experiments was a feed-forward network with a single hidden layer for both the decoder and the encoder models. The dimensionality of the latent random variable was 15 and the hidden layer contained 50 units. ReLU activations were used unless otherwise stated.

C.1.2 MNIST - Label-free Structured Learning

Deriving the Loss

We follow the specification from Kingma et al. (2014) where the likelihood of a single image, x, conditioned on a cluster assignment label y, is shown in Eq. C.4. Again, z is a latent parameter, again assumed to follow an isotropic Gaussian distribution.

$$\log p(x,y) \geq \mathbb{E}_{q(z|x)}\left[\log p_\theta(x \mid y,z) + \log p(z) - \log q(z \mid x)\right] \tag{C.4}$$

We refer to the right hand side of Eq. C.4 as $-V(x,y)$. Eq. C.4 assumes knowledge of the label y, but this is unknown for our domain. However, we can implement the knowledge from Eq. 4.11 (in the main text) that specifies all 100 possibilities for the image inference task. Below we assume the data is of the form $image_i + image_j = (image_{k_1}, image_{k_2})$ where $label_i \in [0,\dots,9]$, $label_j \in [0,\dots,9]$, $label_{k_1} \in [0,1]$ and $label_{k_2} \in [0,\dots,9]$. Finally, we let π_h refer to the MultiplexNet Categorical selection variable that chooses which of the 100 possible terms for $(i, j, k_1, k2)$ are present. Following the MultiplexNet framework, the loss is then presented in Eq C.5:

$$\mathscr{L}(\theta) = \sum_{i,j,k} \pi_h \left[V(x_1, y_1 = i) + V(x_2, y_2 = j) + V(x_3, y_3 = k_1) + V(x_4, y_4 = k_2) + \log \pi_h\right] \tag{C.5}$$

Samples from the Vanilla VAE prior from Kingma et al. (2014)

We repeat the experiment from Section 4.3.2 with a vanilla VAE model ("Model 2" from Kingma et al. (2014)). Here we simply show that the model can capture the label clustering of the data but that it cannot, unsurprisingly, infer the class labels correctly from the data without considering the fact that the data set has been structured:

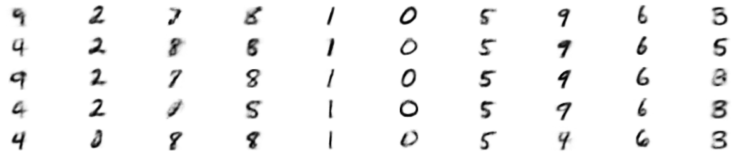


Figure C.3: Reconstructed/Decoded samples from the prior, z, of "Model 2" from Kingma et al. (2014). The clustering of the data is clear but the model is unable to infer the correct class labels without considering the structured data set and domain knowledge.

Network Architecture

The default network used in these experiments was a feed-forward network with two hidden layers for both the decoder and the encoder models. The first hidden layer contained 250 units and the second 100 units. The dimensionality of the latent random variable was 50. ReLU activations were used unless otherwise stated.

C.1.3 Hierarchical Domain Knowledge on CIFAR100

Deriving the Loss

Following the encoding in Fischer et al. (2019), we consider constraints which specify that groups of classes should together be very likely or very unlikely. For example, suppose that the SC label is *trees* and the class label is *maple*. Our domain knowledge should state that the *trees* group must be very likely even if there is uncertainty in the specific label *maple*. Fischer et al. (2019) use the following logic to encode this belief (for the 20 super classes that are present in CIFAR100):

$$(p_{people} < \varepsilon \vee p_{people} > 1-\varepsilon) \wedge \cdots \wedge (p_{trees} < \varepsilon \vee p_{trees} > 1-\varepsilon) \tag{C.6}$$

This encoding is exactly the same as that presented in Eq C.7. However, we rewrite this encoding in DNF such that it is compatible with MultiplexNet.

$$(p_{people} > 1-\varepsilon \wedge p_{trees} < \varepsilon \wedge \ldots) \vee (p_{people} < \varepsilon \wedge p_{trees} > 1-\varepsilon \wedge \ldots) \vee \ldots \tag{C.7}$$

A simplification on the above, as the classes and thus the super classes lie on a simplex, is the specification that $(p_{people} > 1-\varepsilon)$ necessarily implies that $(p_{trees} < \varepsilon \wedge \ldots)$ holds too. Thus the logic can again be simplified to:

$$(p_{people} > 1-\varepsilon) \vee (p_{trees} > 1-\varepsilon) \vee (p_{fish} > 1-\varepsilon) \vee \ldots \tag{C.8}$$

Again, as the probability values here lie on a simplex, we can represent a single constraint as in Eq C.9. Here, Z is the normalizing constant that ensures the final output values are a valid probability distribution over the class labels (computed with a softmax layer in practice). $Z = e^{baby} + e^{boy} + \cdots + e^{cattle} + e^{tractor}$ for all 100 class labels in CIFAR100.

$$(p_{people} > 1-\varepsilon) \implies \frac{e^{baby}}{Z} + \frac{e^{boy}}{Z} + \frac{e^{girl}}{Z} + \frac{e^{man}}{Z} + \frac{e^{woman}}{Z} > 1-\varepsilon \tag{C.9}$$

Finally, we can simplify the right hand side of Eq C.9 to obtain the following specification (for the *people* super class, but the other SCs all follow via symmetry):

$$e^{baby} + e^{boy} + e^{girl} + e^{man} + e^{woman} > \frac{1-\varepsilon}{\varepsilon}\left[e^{beaver} + e^{couch} + \cdots + e^{streetcar}\right] \tag{C.10}$$

Note that the right hand side of Eq C.10 contains the classes for all the other super classes but not including *people*. It thus contains 95 labels in this example.

Studying each of the terms in $j \in [baby, boy, girl, man, woman]$ separately, and noting that e^y is strictly positive, we obtain Eq C.11. We use y_j to denote a class label in the target super class (in this case $people$) and y_i to refer to all other labels in all other super classes (SC).

$$e^{y_j} > \frac{1-\varepsilon}{\varepsilon}\left[\sum_{i \notin SC_{people}} e^{y_i}\right] \tag{C.11}$$

As we are interested in constraining the unnormalized output of the network (y_j) in MultiplexNet, it is clear that we can take the logarithm of both sides of Eq C.11 to obtain the final objective for one class label. Together with Eq C.8, we then obtain the final logical constraint in the main text in Eq 4.12.

$$y_j > \log(\frac{1-\varepsilon}{\varepsilon}) + \log \sum_{i \notin SC_{people}} e^{y_i} \tag{C.12}$$

This implementation can then be directly encoded into the MultiplexNet loss as usual (where y_k refer to the constrained output of the network for each of the 20 super classes, π_k is the MultiplexNet probability for selecting logic term k, and CE is the standard cross entropy loss that is used in image classification.

$$\mathscr{L}(\theta) = \sum_{i=1}^{20} \pi_k \left[CE(y_k) + \log \pi_k\right] \tag{C.13}$$

Network Architecture

We use a Wide ResNet 28-10 (Zagoruyko and Komodakis, 2016) in all of the experimental conditions for this CIFAR100 experiment. We build on top of the pytorch implementation of the Wide ResNet that is available here: `https://github.com/xternalz/WideResNet-pytorch`. This implementation is available under an MIT License.