Interpretable Policy Learning

Alex J. Chan

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Wolfson College August 2020

This thesis is dedicated to my loving family, without whom all of this would not be possible.

Declaration

I, Alexander James Chan of Wolfson College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

All software used in this thesis was written from scratch in Python. Alihan Hüyük did kindly provide me a copy of the code for his algorithm in Huyuk et al. (2020) for reference, though for my own work which builds upon it I re-wrote everything for use within an automatic differentiation framework.

Code for the Transductive Dropout model can be found at:
https://github.com/XanderJC/transductive_dropout

All other code can be found at:
https://github.com/XanderJC/mphil_thesis

This report contains no more than 14667 words.

Alex J. Chan
August 2020

Acknowledgements

First and foremost I would like to thank my supervisor, Mihaela van der Schaar for her constant support and guidance throughout this project and my time during the MPhil.

Many members of the ML-AIM group contributed to thoughtful discussions about the important questions raised in this thesis for which I am very grateful but particular thanks must go to both Ahmed Alaa and Alihan Hüyük whose suggestions and help in understanding have been invaluable.

Finally I'd like to thank Sam Livingstone of the UCL Statistical Science department who first ignited my interest in research, his enthusiasm and advice have pushed me to be in this position today.

Abstract

The medical setting produces a number of significant and unique challenges for machine learning practitioners based on the accessibility of data alongside the need for clear and transparent decision-making to ensure patient safety.

In this work we are primarily concerned with how we can augment clinical decision making through the use of machine learning and analytics. While there are of course many frontiers to be explored we tackle two key challenges: In the first case we consider the problem of accurate and well calibrated prediction under covariate shift, a common issue in healthcare and more generally when learning imitation policies using behavioural cloning. We propose *Transductive Dropout*, leveraging the unlabelled data to regularise uncertainty information over predictions in Bayesian neural networks. By tackling the problem of risk prediction for prostate cancer patients across global populations we demonstrate significant improvement in uncertainty calibration.

In the second we tackle the problem of using imitation learning specifically for the goal of *understanding* clinicians. The aim being to produce transparent and interpretable generative models of their behaviour in order to later support their decision making and catch anomalous actions. We derive a variational Bayesian approach to *Direct Policy Learning* in order to appropriately handle uncertainty in decision making as well as introducing InterPoLe, and algorithm for *Interpretable Policy Learning* that uses evolving soft decision trees to generate personal and interpretable policies that can be easily inspected. We apply these methods to understanding the diagnosis of Alzheimer's disease, as well as treating cystic fibrosis patients in order to better understand the decision making process of doctors, uncovering insights that simply were not possible before.

Table of contents

Nomenclature

Roman Symbols

A	Action space
O	Observation function
R	Reward function
S	State space
T	Transition function
Z	Observation space

Greek Symbols

μ	Mean parameter
Ω	Regulariser function
ϕ	Variational distribution parameter vector
π	Policy function
ψ	Digamma function
ρ	Policy occupancy measure
σ^2	Variance parameter
θ	Model parameter vector
α,β,γ,η	Various parameters

Other Symbols

$\mathscr{D}$	Observed data set
$\mathbb{E}$	Expectation operator
exp	Exponentiation operator
$\mathscr{F}$	Evidence lower bound functional
log	Natural logarithm
∇	Vector Differential operator
$\mathscr{O}$	Big O complexity notation
$p(\cdot)$	Probability density function
$q(\cdot)$	Probability density function of a variational distribution
$\mathbb{R}$	Set of real numbers
$\Delta(\cdot)$	Space of probability distributions over the set
$\mathbb{V}$	Variance operator

Acronyms / Abbreviations

ADNI	Alzheimer’s Disease Neuroimaging Initiative
AUROC	Area Under Receiving Operator Curve
BC	Behavioural Cloning
BNN	Bayesian Neural Network
CF	Cystic Fibrosis
DIPOLE	Direct Policy Learning
ELBO	Evidence Lower Bound
EM	Expectation Maximisation
FEV	Forced Expiratory Volume
GAN	Generative Adverserial Network

HMM	Hidden Markov Model
i.i.d.	Independent and Identically Distributed
IOHMM	Input-Output Hidden Markov Model
IRL	Inverse Reinforcement Learning
LSTM	Long Short Term Memory
MC	Monte Carlo
MCDP	Monte Carlo Dropout
MCI	Mild Cognitive Impairment
MCMC	Markov Chain Monte Carlo
MDP	Markov Decision Process
MLE	Maximum Likelihood Estimator
MRI	Magnetic Resonance Imaging
NCF	Normal Cognitive Function
NN	Neural Network
PSA	Prostate Specific Antigen
RL	Reinforcement Learning
SSL	Semi Supervised Learning
TD	Transductive Dropout
UDA	Unsupervised Domain Adaptation
VBEM	Variational Bayesian Expectation Maximisation
VI	Variational Inference

Chapter 1

Introduction

In the age of big data, settling on a decision can be an overwhelming task for humans; paradoxically the more information made available to someone the more it can complicate the process (Malhotra, 1982), through both being unable to deal with the all aspects of the information as well as potentially nuisance variables masking the signal. In the medical setting the rise of electronic health records has led to an unprecedented amount of data being made available to clinicians. As a result, cases of *information overload* have been noted where missteps have been taken or important factors overlooked (Singh et al., 2013). In a cruel twist it is often the patients most in need that are disproportionately highly affected; patients with chronic and severe diseases are the ones routinely tested and surveyed, producing vast quantities of data that need to be considered.

Machine learning provides an excellent opportunity to alleviate this problem by embracing the increase in data and leveraging the power of modern tools to improve patient outcomes. In this thesis we are interested in how we can best assist medical professionals by providing information and augmenting their decision making process in order to consistently arrive at the correct diagnosis and treatment while overcoming the unique challenges of the healthcare setting.

We believe there are two ways that systems can support medical policies and decisions. The first fits more traditionally within the current machine learning literature and is focused on the accurate prediction of patient risk given all the available information that a human might not be able to appropriately handle. While there are a plethora of modern supervised learning techniques that can be applied to do just that (Litjens et al., 2017), there can be issues applying out-of-the-box algorithms due to the peculiarities of medical data sets and privacy concerns. Consider for example the concrete case of trying to predict the risk of death for a group of patients suffering from prostate cancer in a country where we have no labelled data. This might be due to tight privacy regulations on medical data for example, however we do have access to labelled examples from a different country which we could instead use to train a model. It is well known that modern machine learning methods struggle to generalise well and the populations of each country may differ in their underlying distribution of features (a common problem, known as *covariate shift*) so a model purely trained on the labelled data may perform poorly on the unlabelled data both in terms of accuracy and uncertainty estimation. Our first contribution in this thesis then deals directly with this problem, introducing a new Bayesian neural network scheme that produces better calibrated uncertainty and predictions over covariate shifted data. While originally considered for the aforementioned scenario we note that it is more broadly appropriate for the policy learning setting, since it is also well known that covariate shift arises significantly in behavioural cloning since a lack of state dynamics awareness can cause an agent to drift away from previously seen states (Osa et al., 2018).

While these directly predictive models are undeniably useful for clinicians there can be significant issues when it comes to implementing them in practice. This is down to the fact that we simply *must* be able to explain decisions in order to ensure the safety of patients, and when the stakes are so high there needs be a level of accountability. It is not clear how this really works when it is an algorithm is making or informing the decisions, even a transparent and interpretable one (De Laat, 2018). The second way then we believe machine learning can be used effectively is as a supportive tool that aims to essentially *de-bug* a doctor's decision making process. This involves learning to *understand* the doctor, to be able to describe why they made particular actions and inspect what appear to be their goals and motivations. This requires a generative model of their behaviour but more importantly an *interpretable* one that we can see how they consider the environment to be behaving and how that translates into actions. Armed with that model we can work with clinicians, supporting their decisions by learning appropriate practice and being able to alert them if it appears they are taking unusual steps or might have overlooked something. With this in mind the remaining contributions of this thesis involve two models for learning transparent representations of an agent's decision making process - one with a focus on uncertainty quantification through the use of approximate inference and one with a focus on complete interpretablity by using decision tress as the basis for all policies.

1.1 Contributions

In the course of this work we make a number of contributions that we highlight now:

1. A **review of existing methods for imitation learning**, highlighting particularly where they fall short when it comes to applications in the medical setting.
2. Proposing Transductive Dropout, **a novel method for improving uncertainty calibration under covariate shift**, aimed at producing better risk scores when applying models to new domains and which is appropriate for many behavioural cloning tasks in order to better handle generalisation to new areas of the state-space.
3. Deriving **a new stochastic variational inference scheme** for learning an approximate posterior in direct policy learning. This allows for a decomposition in the uncertainty surrounding an agents actions and lets us capture the natural variation in practice that arises.
4. Proposing InterPoLe, **an algorithm for interpretable policy learning** that uses a **novel soft decision tree architecture** to learn an inherently interpretable description of an agent's policy and decision dynamics.
5. A **demonstration of the potential of all of the proposed methods** using real examples from medical data to understand decision making and improve patient outcomes.

1.2 Outline of the Thesis

Following this introduction, in chapter 2 we briefly cover the general field of sequential decision making an imitation learning. While many of these methods will be inappropriate for the medical setting and even more so for learning an interpretable representation of a policy it is important to understand the background and setting of our work in the current literature.

We will then move swiftly into our contributions; in chapter 3 we introduce Transductive Dropout, a development of Bayesian neural networks for better calibrated uncertainty under covariate shift. We explain the current issues with approximate Bayesian inference for neural network models and show that by appropriately designing a posterior regularisation scheme we can obtain better uncertainty estimates in this setting.

In chapter 4 we introduce Variational DIPOLE. First we explain the model of DIPOLE Huyuk et al. (2020) and how it can be used to capture a transparent description of the decision making process. Then we build upon the method by showing how we can use a variational Bayesian Expectation Maximisation algorithm to learn a full approximate posterior over the model parameters instead of just the maximum likelihood estimate.

Moving on in chapter 5 we introduce InterPoLe, an algorithm for learning a more explicitly interpretable policy in the form of evolving decision trees. We cover the current decision tree architectures before introducing our new gating function for soft trees that allows for a more traditional interpretation of the partitions. We show how this policy over beliefs induces a decision tree policy over observations that transforms at every time-step and how this can be used within a system for supporting clinicians such that they make fewer mistakes.

Having introduced our methods, in chapter 6 we apply all of them to a variety of real medical problems. We demonstrate Transductive Dropout's abilities in accurately quantifying uncertainty for predicting prostate cancer mortality across globally diverse populations. We apply Variational DIPOLE to the task of understanding how clinicians diagnose Alzheimer's disease and finally we use InterPoLe to gain insight into the treatment of patients suffering from cystic fibrosis.

We conclude in chapter 7 with some final thoughts and directions for future work.

Chapter 2

Learning to Make Decisions

In this chapter we review the key concepts and related work required to understand our later contributions and place it within the current literature. We will briefly cover the general ideas of sequential decision making, which is primarily concerned with a fully online setting, optimising a given reward that is provided to the agent. Then we will move to the imitation learning setting that we most concern ourselves with, where we assume no reward is given and instead learning is motivated through demonstrations provided from some *expert* that we wish to match.

2.1 Sequential Decision Making

Elementary concepts and results in this section can be considered quoted from Sutton and Barto (2018) unless otherwise stated.

While sequential decision making can really describe any scenario that involves repeatedly having to make decisions, in machine learning it tends to be synonymous with reinforcement learning (RL), the framework by which an agent learns to act optimally in an environment purely through interaction and feedback in the form of some real valued reward. Formally the environment is considered to be a Markov decision process (MDP), defined by the tuple $\langle S, A, T, R, \gamma \rangle$, where:

- S is the set of states;
- A is the set of actions;

- $T : S \times A \rightarrow \Delta(S)$ is the transition function, where $T(s'|s,a)$ is the probability of transitioning into state s' after taking action a in state s;
- $R : S \rightarrow \mathbb{R}$ is the reward function, where $R(s)$ is the reward for being in state s, and;
- $\gamma \in [0,1]$ is the discount factor.

In this case an agent interacts with the environment by observing a state and taking an action before subsequently receiving a reward and transitioning to a new state, repeating the process while following some policy $\pi : S \rightarrow \Delta(A)$. The goal then is to find the optimal policy that will maximise the expected discounted sum of future rewards:

$$\pi^* = \underset{\pi}{argmax}\{\mathbb{E}_{\pi,T}[\sum_{t=0}^{\infty} \gamma^t R(s_t)]\} \tag{2.1}$$

A full review of RL would be beyond the scope of (and largely irrelevant to) this thesis since the vast majority of recent work has revolved around the *online* setting where agents are free to test out policies by interacting in the environment as much as they wish. Recently though the *offline* setting has received a lot more attention as interest has shifted towards implementing these systems in high impact environments in the real world including healthcare. Often in these places interaction with the environment is costly, or completely inappropriate for an untrained agent (we can't let an ε-greedy agent randomly see what happens if we give a patient a drug that could be harmful)

It is useful to define some important auxiliary functions that will simplify the handling of optimal policies. Let the *value* of state s when following some policy π be given by:

$$V_\pi(s) = \mathbb{E}_{\pi,T}[\sum_{t=0}^{\infty} \gamma^t R(s_t)|s_0 = s], \tag{2.2}$$

the expected sum of total discounted rewards following policy π, assuming a start in s. Similarly we can extend the value to include which action is taken to arrive at the *Q-function* given by:

$$Q_\pi(s,a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t)|s_0 = s, a_0 = a], \tag{2.3}$$

which defines the expected sum of total discounted rewards following policy π, assuming a start in s and taking action a. An important point on the Q-function is that according to the Bellman Optimality Theorem π is an optimal policy if and only if for all s:

$$\pi(s) \in \underset{a \in A}{argmax} Q_\pi(s,a). \tag{2.4}$$

This makes sense - if there is another action for which the Q-function is higher it would surely be better to select that action instead. It also means obtaining the true Q-function is sufficient to solve a given MDP. This has led to a number of very popular off-policy algorithms, including the original Q-Learning (Watkins and Dayan, 1992), that aim to simply learn the Q-function from experience. Modern deep-RL is essentially built on the foundation of function approximators for the Q-function (Mnih et al., 2013; Van Hasselt et al., 2015). These algorithms use the recursive nature of the Bellman equations to minimise the *temporal-difference error* between the Q-values predicted by their networks and the actual reward received at every time step. While they have been shown to be extremely powerful at learning a successful policy across a variety of domains they do still have their limitations. In particular they are known to require a very large amount of experience and interaction with the environment in order to reach human level performance - in the order of many years (Arulkumaran et al., 2017). The question of online exploration is also still not well answered, in order to learn well the agent needs to be able to experience every part of the state space but it needs to balance this with taking actions it knows are reasonable and current schemes that will randomly take actions contribute to the long training times especially in complicated environments. Importantly for our work in healthcare as well there is the required notion of a reward given at each step. It is very hard to craft meaningful and useful rewards based on the state of patients, which we also would not know to be accurate, without resorting to very sparse and not too informative rewards like a mortality indicator. As such "core" RL will not play a huge role in our work although it is very relevant for grounding concepts and when it comes to the *inverse* question.

2.2 Imitation Learning

In the imitation learning framework, unlike in RL, the learning signal does not come through some reward received from the environment but rather from a collected dataset of *expert* demonstrations that show what the "correct" way to do a task is. We maintain the assumption though that correctness is based on the demonstrator acting according to the optimal policy π^* in the environment and that having learnt from a demonstrator our goal will still be to maximise expected reward at some test time.

2.2.1 Behavioural Cloning

In the simplest case of imitation learning we arrive at behavioural cloning (BC) (Bain and Sammut, 1995). In this paradigm the environment is considered fully observed and an algorithm is given a training data set comprising the states visited by an agent and their corresponding actions. Based on this a purely discriminatory policy is learnt that regresses actions directly on states, borrowing any appropriate model from the supervised learning literature. This has its advantages and despite its simplicity has been shown to be quite effective in a number of domains including flying drones (Giusti et al., 2015) and driving cars (Bojarski et al., 2016). The primary reason for applying BC is that it requires no further interaction with the environment before being immediately able to imitate the demonstrator.

The key issue with BC is that it lacks awareness of state dynamics - thus a greedy imitator picking the most likely action every time can quite soon drift away from states that they have seen in the logged data due to accumulating error. The further from previously seen states the agent gets, the worse its performance will tend to be as well as modern supervised learning methods often fail to generalise well outside of their training data (Ovadia et al., 2019).

Additionally in real-world problems it is often unreasonable to assume full observability and Markovianity and so we may be concerned that we are not considering the full history up until the current time step where an action is selected. This can be remedied slightly through the use of recurrent models although this generally loses all pretence at interpretability, it is impractical to gain insight from the latent states of an LSTM (Hochreiter and Schmidhuber, 1997).

2.2.2 Inverse Reinforcement Learning

Introduced by Ng et al. (2000), inverse reinforcement learning (IRL) offers an alternative approach for imitation learning. The principal concept is simple - given an MDP without the reward function (MDP\R) but with some demonstration trajectories ($\mathscr{D}$), determine the reward R that the demonstrator appears to optimise. By itself this doesn't elicit an imitator policy although having obtained an estimated reward function this can be reached through running any given (forward) RL algorithm (e.g. Q-learning).

While this approach appears a sensible way to understand the behaviour and motivations of a demonstrator it is important to note that the IRL task is technically ill-posed. That is to say that for any given MDP\R there will be an infinite number of reward functions for which $\mathscr{D}$ is an optimal demonstration including in the simplest case a constant reward everywhere. Ng et al. solves this heuristically by introducing the max-margin approach, aiming to learn a reward function that gives high reward to the demonstrator while returning as low as possible reward to all other policies. By assuming that the reward function is a linear combination of state features: $R = \theta^T \mathbf{f}$, they show that they're searching for a parameter vector θ that induces a policy that matches *feature expectations* (the expected amount of times a feature is seen should be the same under the induced policy as in the demonstrations), requiring the use of an oracle MDP solver.

Ziebart et al. (2008) build on this and introduce an alternative to the max-margin approach by way of the maximum entropy principle which exponentially prefers rewards that grant higher returns to expert trajectories than those that don't. While the method for breaking ambiguities is different, this retains the limitations of a linear reward and essentially the need for the environment to be solvable in reasonable time.

Bayesian IRL

An altogether different approach through the use of Bayesian inference in order to reason about the posterior distribution of the reward given some seen demonstrations. Having set a prior, Ramachandran and Amir (2007) defines the likelihood of an action at a state as a Boltzmann distribution given by an inverse temperature and the respective Q-values of each action. This yields an intractable posterior distribution leading to a Markov chain Monte Carlo algorithm using a random grid-walk to sample from the posterior. This maintains the use of a linear reward and for each sample the likelihood needs to be calculated - meaning the MDP needs to be solved with the sampled reward in order to obtain the Q-values, rendering this impractical in large environments.

While there have been several extensions that consider maximum a posteriori inference and multiple reward functions (Choi and Kim, 2011b, 2012), the requirement to solve an inner loop MDP significantly hinders a Bayesian approach to IRL. This is true even when the linearity of reward is relaxed and Gaussian processes used for inference (Levine et al., 2011). Recently Brown and Niekum (2019) has looked to solve this by introducing an alternative formulation of the likelihood, one based on human recorded pairwise preferences over demonstrations that significantly reduces the complexity of likelihood calculation but does necessitate that we have these preferences available.

The Bayesian approach offers a principled way to deal with all possible reward functions and also grants uncertainty information over agents preferences. That being said it can be hard to produce meaningful insights from a nebulous reward even when uncertainty is attached. In chapter 4 we show how a Bayesian approach to directly learning a behavioural policy allows for much cleaner reasoning over uncertainty.

2.2.3 Adversarial Imitation Learning

While inferring a reward function is in of itself an important goal it has been noted that when the goal is ultimately to obtain an imitator policy the IRL part can be done implicitly. This has been popularised through the use of *generative adversarial imitation learning* (Ho and Ermon, 2016) which learns a policy through occupancy measure matching, and variants (Li et al., 2017) have been shown to uncover some level of interpretability. Briefly the idea is to see the task of imitation leaning as the composition RL ∘ IRL. They show that in the maximum-entropy setting inferring some reward constrained by some regularising function Ω implicitly seeks an optimal policy that minimises some divergence between the occupancy measures of the induced policy and demonstrator policy. Where the divergence used depends on Ω and the occupancy measure is given by:

$$\rho_\pi(s,a) = \pi(a|s)\sum_{t=0}^{\infty}\gamma^t P(s_t = s|\pi), \tag{2.5}$$

and uniquely defines the policy. With an appropriate choice of Ω then they produce an objective that minimises the Jensen-Shannon divergence (Gray, 2011) between occupancy measures, thus minimising a true metric between the learnt and demonstrator policies allowing for potentially exact policy matching that would not be possible under linear apprenticeship methods. In order to solve this optimisation problem, Ho and Ermon draws a connection to generative adversarial networks (GANs) (Goodfellow et al., 2014) by introducing a discriminator network whose job is to distinguish trajectories generated between the policy network and the demonstrator based on their state occupancy measures. The two networks can then be trained similarly to GANs to find a saddle point of this mini-max objective.

While this can very effective, this family of algorithms require significant interaction with the environment in order to calculate occupancy measures of the learnt policy, making them unsuitable in the medical setting which is almost always offline. Additionally it inherits a lot of the training instability of GANs and the assumption that the discriminator network reaches optimality which is unlikely to be entirely true.

2.2.4 Direct Policy Learning

All the methods so far explicitly aim to produce a policy that excels at the given task at hand. As such little attention has been paid to interpretable parameterisations, with most forms of π being some form of neural network. This leads to very poor understanding of why the agent is acting in the way they are. To address this issue Huyuk et al. (2020) introduce a method for *direct policy learning*, that is one that learns a direct map from previous history into a predicted action without relying on rewards as an intermediary. We distinguish this from BC in that there is a relaxation of the Markovianity of the observations and an incorporation of a hidden latent state to induce time dependency. It should be quickly noted that this has also been considered in the area of robotic control where there have been a few model-based approaches in a fully observable setting, with Ude et al. (2004) learning a kinematic model of the robot dynamics as well as Van Den Berg et al. (2010) and Englert et al. (2013) learning autoregressive exogeneous model, but that these are inappropriate for the healthcare setting given the setup.

The method of Huyuk et al. (which we shall expand on formally in chapter 4) involves specifically modelling the decision dynamics of the demonstrator. They assume that the demonstrator thinks about the environment as an Input-Output Hidden Markov Model (IOHMM) (Bengio and Frasconi, 1995) and that they accumulate all the information from the past history of observations in a trajectory in the form of a belief distribution over the hidden states of the

IOHMM. They then parameterise a policy from the belief directly into actions by way of "mean-vectors" on the belief simplex that represent all the possible actions, the likelihood of each being the relative distance from the belief at each time step.

This method has a number of key benefits that existing methods in imitation learning do not. Firstly it operates in an entirely offline manor, secondly its parameterisation is interpretable, and thirdly it copes with partially observed environments. Crucially this method, and the ones introduced in chapters 4 and 5, specifically capture the *cognitive dynamics* of the decision maker and not that of the environment. This is important for our objective to learn a descriptive model of an agent's observed behaviour, to understand how they are effectively behaving under a certain (interpretable) parameterisation. This makes us not concerned with the true environment (which will be too complicated for the decision maker themselves to comprehend).

Chapter 3

Transductive Dropout: Calibrating Uncertainty under Covariate Shift

The content of this chapter has already been published at the International Conference on Machine Learning 2020 in Chan et al. (2020). However, as all of the work was conducted during the course of the MPhil and has not been submitted in any capacity for assessment as part of this or any other degree it is appropriate for inclusion here.

In this chapter we offer a solution to the general problem of calibrating uncertainty under covariate shift in a supervised setting through Bayesian neural networks (Neal, 2012) - a highly significant problem in the medical setting. These models aim to solve the uncertainty quantification problem by learning neural networks via Bayesian inference and encapsulate the prediction uncertainty in the posterior predictive distribution, which is typically intractable and has to be approximated (Blundell et al., 2015; Graves, 2011). While existing approximation methods are able to produce reliable uncertainty estimates over in-distribution data, it has been shown that they tend to be over-confident under covariate shift (Ovadia et al., 2019). Thus we propose Transductive Dropout, a method leveraging information from the unlabelled target data to find a better approximation to the posterior. We make the following observation: a point being in the target data is an indication that the model should output higher uncertainty because the target distribution is not well-represented by training data due to covariate shift. Therefore, we use whether the data come from training or target set as a "pseudo-label" of model confidence. This naturally leads to a posterior regularisation term which we incorporate into the variational approximation objective.

As established in Chapter 2, simple behavioural cloning suffers significantly from this exact issue, happening when the agent drifts away from the area of the state space seen from the demonstrator. A particularly serious case occurs when this happens without anyone noticing - if it is clear to everyone that this is new ground and we don't know what is going on this is useful information that can inform a decision. On the other hand if from the point of view of the model everything looks fine and it confidently predicts an *incorrect* action this could lead to a lot of unfortunate behaviour. Thus our method has great application to the policy learning setting, as correctly calibrated uncertainty is crucial here in knowing when it will be appropriate to trust the model predictions.

3.1 Overview of Related Methods

Utilising unlabelled data to improve uncertainty estimate under covariate shift is a previously less explored area in the literature. Here we highlight some of the key methods in the surrounding fields to contextualise our work.

Bayesian Uncertainty Estimate for Neural Networks Bayesian methodology has been applied to quantify the predictive uncertainty of neural networks leading to a large family of methods known as Bayesian Neural Networks (BNNs). A BNN learns a posterior distribution over parameters that encapsulates the model uncertainty. Due the complexity of deep neural networks, the exact posterior is usually intractable. Hence, much of the research in BNN literature is devoted to finding better approximate inference algorithms for the posterior. Popular approximate Bayesian approaches include dropout-based variational inference (Gal and Ghahramani, 2016; Kingma et al., 2015) and Stochastic Variational Bayesian Inference (Blundell et al., 2015; Graves, 2011; Louizos and Welling, 2017). These methods are known to achieve reliable uncertainty estimate in i.i.d scenario. However, recent research has cast doubt about the validity of these uncertainty estimates under covariate shift (Ovadia et al., 2019). Moreover, the above methods do not make use of any unlabelled data for training or inference.

Semi-Supervised Learning Semi-supervised learning (SSL) covers the broad field of learning from both labelled and unlabelled data (Zhu and Goldberg, 2009). It's generally separated into two with most of the work covering *inductive* SSL which aims to use the unlabelled data to learn a general mapping from the features to the outcome. Many recent works encourage the model to generalise better by using a *regularisation* term computed on the unlabelled data (Berthelot et al., 2019). This includes *entropy minimisation* which encourages the model to produce confident predictions on unlabelled data (Grandvalet and Bengio, 2005; Jean et al., 2018; Lee, 2013) and *consistency regularisation* which ensures the predictions for slightly perturbed data stay similar (Sajjadi et al., 2016). The other split covers *transductive* SSL where the aim is to make predictions over only the unlabelled points given with no need to generalise further. As we will show later, the proposed Transductive Dropout fits more into this framework, using the unlabelled data as a regulariser in order to induce a better variational approximation to the intractable posterior distribution.

However, our work is significantly different from traditional SSL in several ways. First, we note that most existing works in SSL focus entirely on using unlabelled data to improve predictive performance (e.g. accuracy), but much less thoughts have been given to improving the uncertainty estimate for those predictions, which is the focus of this paper. Furthermore, our work explicitly addresses the issue of covariate shift between source and target data whereas traditional SSL often assumes that they are i.i.d. In addition, most of the recent work in SSL considers problems like image classification and natural language processing where the methods can leverage the complicated dependencies in the features - we don't consider this a focus and develop a method that works appropriately for tabular data as well.

Unsupervised Domain Adaptation Unsupervised domain adaptation (UDA) is the task of training models to achieve better performance on a target domain, with access to only unlabelled data in the target domain and labelled data from a (different) source domain. (Kouw and Loog, 2019) contains a detailed review of popular UDA methods. As with SSL, existing works on UDA centre around improving predictive performance rather than producing well-calibrated uncertainty estimates. Our work contributes to the UDA literature by proposing a method to improve the uncertainty estimates on the predictions made in the target domain.

Transfer Learning In the setting of transfer learning (Torrey and Shavlik, 2010) the task does involve a change in distribution over features but typically also involves some amount of labels on the target set (known as one-shot or few-shot learning). This has led to a lot of work that uses the training set to learn a useful prior for a second model that can be trained on the labelled data in the target set (Karbalayghareh et al., 2018; Raina et al., 2006). Given the complete lack of labels in our target data set this is inapplicable for our problem.

3.2 Notation and Problem Setup

Let $\mathbf{x} \in \mathbb{R}^d$ be a d-dimensional feature vector, and $y \in \mathscr{Y}$ be the prediction target; where $\mathscr{Y} = \mathbb{R}$ for regression targets, and $\mathscr{Y} = \{1, \ldots, K\}$ for K-class classification targets. We are presented with *two* sources of training data: a labelled data set $\mathscr{D}_L$, and an unlabelled data set $\mathscr{D}_U$. The labelled data set comprises a collection of n feature-label pairs, i.e., $\mathscr{D}_L = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, whereas the unlabelled set comprises a collection of m feature instances $\mathscr{D}_U = \{\mathbf{x}_j\}_{j=1}^m$.

We assume that $\mathscr{D}_L = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ consists of i.i.d samples of features and labels drawn from the distribution

$$(\mathbf{x}_i, y_i) \sim p(\mathbf{x}) \times p(y|\mathbf{x}),\ \forall i \in \{1, \ldots, n\},$$

where both $p(\mathbf{x})$ and $p(y|\mathbf{x})$ are unknown, and could only be accessed empirically through $\mathscr{D}_L$. We will refer to $p(\mathbf{x})$ as the *feature distribution* — feature instances in the unlabelled data set are assumed to be drawn from a shifted feature distribution as follows:

$$\mathbf{x}_j \sim p'(\mathbf{x}),\ \forall j \in \{1, \ldots, m\},$$

where $p'(\mathbf{x}) \neq p(\mathbf{x})$, whereas the unobserved labels in the data set $\mathscr{D}_U$, i.e., the blue dots in Figure 3.2 corresponding to $\{y_j\}_{j=1}^m$, are generated from the same conditional distribution $y_j \sim p(y|\mathbf{x}_j)$. Note that even though the feature distributions $p'(\mathbf{x})$ and $p(\mathbf{x})$ differ, the conditional $p(y|\mathbf{x})$ is invariant across the two data sets. This situation is commonly known as *covariate shift* (Shimodaira, 2000). We denote the entirety of observed data $\mathscr{D} = \{\mathscr{D}_L \cup \mathscr{D}_U\}$.

3.2.1 Learning from (and for) unlabelled data

Our key objective is to use the (source) labelled data set $\mathscr{D}_L$ to train a model that would be applied to the (target) unlabelled data set $\mathscr{D}_U$. However, since the feature distributions in $\mathscr{D}_L$ and $\mathscr{D}_U$ mismatch, we cannot expect a model trained on $\mathscr{D}_L$ to perfectly generalise to $\mathscr{D}_U$. Thus, we aim at training the model to *learn* which prediction instances can be *confidently* transferred from $\mathscr{D}_L$ to $\mathscr{D}_U$, and which cannot be confidently generalised across the two distributions. To this end, we train the model to score its uncertainty on predictions issued for all feature instances in $\mathscr{D} = \{\mathscr{D}_L \cup \mathscr{D}_U\}$.

Taking a Bayesian approach to uncertainty estimation, for a model with parameter θ and a test point $\mathbf{x}^* \sim p'(\mathbf{x})$, the Bayesian posterior distribution over y^* is

$$\underbrace{p(y^*|\mathbf{x}^*,\mathscr{D})}_{\textbf{Total uncertainty}} = \int \underbrace{p(y^*|\mathbf{x}^*,\theta)}_{\substack{\textbf{Data}\\ \textbf{uncertainty}}} \underbrace{p(\theta|\mathscr{D})}_{\substack{\textbf{Model}\\ \textbf{uncertainty}}} \, d\theta. \tag{3.1}$$

The posterior decomposition in (3.1) comprises two types of uncertainty (Malinin and Gales, 2018): *data uncertainty*, also referred to as aleatoric uncertainty, is the variance of the true conditional distribution $p(y|\mathbf{x})$, reflecting the inherent ambiguity or noise in the true labels y (Gal et al., 2017). The second type of uncertainty, *model uncertainty*, pertains to the model's epistemic uncertainty created by the lack of training examples in the vicinity of the test feature $\mathbf{x}^*$. Since the conditional $p(y|\mathbf{x})$ is invariant across the source and target distributions, it is the model uncertainty that we focus on.

3.2.2 Standard approximate Bayesian falls short

A true Bayesian model (with appropriate priors) would completely capture model uncertainty in $\mathscr{D}_U$ by simply training the model on $\mathscr{D}_L$ in a supervised fashion, while completely ignoring the unlabelled data in $\mathscr{D}_U$ (Sugiyama and Storkey, 2007). However, exact Bayesian inference in neural networks is generally intractable (and computationally expensive), hence existing practical solutions to Bayesian modelling rely on approximate inference schemes, for example based on Monte Carlo dropout (MCDP) (Gal and Ghahramani, 2016).

While approximate inference via MCDP — with appropriate hyper-parameter tuning — provides reliable uncertainty estimates for in-distribution data (i.e., feature instances in $\mathscr{D}_L$), it has been shown in Ovadia et al. (2019) that these methods lead to miscalibrated estimates of uncertainty for out-of-distribution data. In the next Section, we develop an approximate Bayesian scheme that makes use of the unlabelled data in $\mathscr{D}_U$ to provide more accurate uncertainty estimates on the predictions made for features instances drawn from the shifted distribution $p'(\mathbf{x})$.

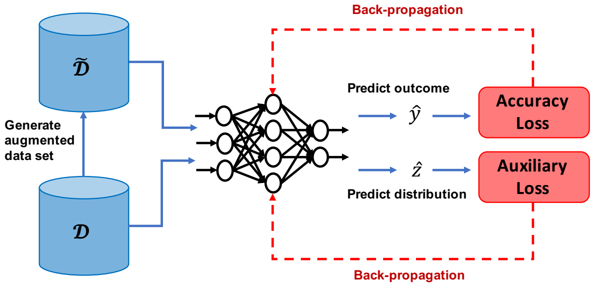

Fig. 3.1 **High-level depiction of our approach.** We first generate our augmented data set with pseudo-labels before feeding forward to make predictions and then back-propagating both errors through the network.

3.3 Transductive Regularisation

How can we use our knowledge of the unlabelled data in $\mathscr{D}_U$ to improve the uncertainty estimates on predictions made for the target distribution $p'(\mathbf{x})$? In this Section, we develop an approximate Bayesian method tailored to this task. Here, we regard a neural network (NN) as a distribution $p(y|\mathbf{x},\theta)$ that assigns a probability to each possible output y.

3.3.1 Variational inference with posterior regularisation

In a Bayesian framework, we specify a prior distribution $p(\theta)$ on the NN parameters, and obtain the posterior $p(\theta|\mathscr{D})$ via Bayes rule. In practice, the posteriors $p(\theta|\mathscr{D})$ and $p(y|\mathbf{x},\mathscr{D})$ in (3.1) are both intractable. To address this issue, we use variational inference, whereby we use a surrogate distribution $q_\phi(\theta)$ parameterised by ϕ to approximate $p(\theta|\mathscr{D})$. The parameter ϕ is obtained by minimising the KL-divergence between p and q as follows (Graves, 2011):

$$\phi^* = \arg\min_{\phi} \mathrm{KL}\big[q_\phi(\theta)||p(\theta|\mathscr{D})\big]. \tag{3.2}$$

In practice the KL divergence is not minimised directly, rather it is achieved my maximising the Evidence Lower BOund (ELBO), which can be written as:

$$\mathscr{F}(\mathscr{D},\phi) = \mathbb{E}_{q_\phi}\big[\log p(\mathscr{D}|\theta)\big] - \mathrm{KL}\big[q_\phi(\theta)\,||\,p(\theta)\big], \tag{3.3}$$

being seen as the balance of two terms. The objective being to maximise the log-likelihood under the surrogate distribution (first term) while regularising the approximation to not be too far from the prior (second term). Variational inference also leads to an approximate posterior predictive distribution $q_\phi(y|\mathbf{x},\mathscr{D})$, obtained by replacing $p(\theta|\mathscr{D})$ in (3.1) with its variational counterpart $q_\phi(\theta)$. Note that the unlabelled data in $\mathscr{D}_U$ is ancillary to the optimisation problem in (3.2), since mere evidence maximisation would render $p(\theta|\mathscr{D}_L)$ as the only relevant conditional for finding the variational parameter ϕ. Hence, the vanilla variational Bayes is insufficient in our setup as it cannot capitalise on our knowledge of the unlabelled data in $\mathscr{D}_U$.

To incorporate the unlabelled data in $\mathscr{D}_U$ into our inference machine, we resort to *posterior regularisation* (Zhu et al., 2014). That is, instead of computing the variational posterior that best matches the true posterior in KL distance, we add a regulariser Ω to the objective in (3.2), i.e.,

$$\phi^* = \arg\min_{\phi} \mathrm{KL}\big[q_\phi(\theta)\,||\,p(\theta|\mathscr{D})\big] + \Omega(q_\phi(\theta|\mathscr{D})), \tag{3.4}$$

in order to explicitly influence the learned variational posterior so that it produces the desired uncertainty profiles, i.e., posterior variance, over the target feature distribution $p'(\mathbf{x})$.

What do our sought-after uncertainty profiles look like? In order to design the regulariser Ω, we first need to specify the influences it needs to exert on the learned variational posterior q_ϕ. Let $\mathbb{E}[q]$ and $\mathbb{V}[q]$ denote the mean and variance of a given distribution q, respectively. A "good" variational posterior is one that matches the true posterior $p(\theta\,|\,\mathscr{D})$, and induces the following uncertainty profile: for any pair of features $\mathbf{x},\mathbf{x}' \sim p'(\mathbf{x})$ drawn from the target distribution, the variational posterior satisfies the following condition:

$$\mathbb{V}[q_\phi(y|\mathbf{x},\mathscr{D})] \geq \mathbb{V}[q_\phi(y|\mathbf{x}',\mathscr{D})] \Leftrightarrow p(\mathbf{x}') \geq p(\mathbf{x}). \tag{3.5}$$

That is, the variance of the variational posterior, which quantifies the model's uncertainty, should be smaller for target test points that are close (in distribution) to the labelled data in $\mathscr{D}_L$, and vice versa. The key idea behind our posterior regularisation approach is that the augmentation of labelled and unlabelled data serve as "pseudo-labels" of model confidence — by regarding the condition in (3.5) as an auxiliary classification task wherein q_ϕ predicts whether a feature $\mathbf{x}$ is drawn from the source or target distributions, we can "train" q_ϕ to make this binary prediction via its variance. Building on this insight, the rest of this Section builds a regulariser Ω that enables q_ϕ to discriminate source and target features.

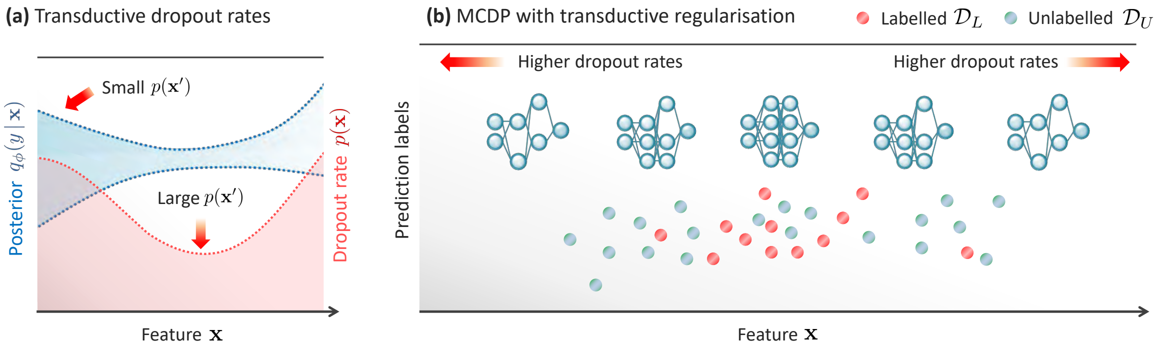

Fig. 3.2 **Pictorial depiction of transductive dropout inference.** (a) Here, we depict an exemplary one-dimensional feature space, along with the corresponding variational posterior $q_\phi(y|\mathbf{x})$ and feature-dependent dropout rate $p(\mathbf{x})$. Transductive dropout inference operates by adapting the dropout rate so that it induces larger posterior variance for regions with dense concentration of unlabelled data, but low density for labelled data (small $p(\mathbf{x}')$ for some $\mathbf{x}' \sim p'(\mathbf{x})$). (b) This panel shows an exemplary realisation of labelled and unlabelled data sets for the same example in panel (a). Red dots are fully observed, whereas for blue ones, we only observe the locations but not the outputs on the y-axis. The typical behaviour of the transductive dropout is to increase the dropout rates in regions where unlabelled data are denser than labelled data, creating more variability in the Monte carlo samples of the network outputs. Here, exemplary instances of test-time dropout applied to the network architecture for different values of the feature $\mathbf{x}$ are depicted.

3.3.2 Posterior regularisation via transductive dropout

As discussed above, we seek a variational posterior that best fits the labelled data in $\mathscr{D}_L$, and discriminates source and target data. Before proceeding, we first define an augmented data set $\widetilde{\mathscr{D}} = \{(\mathbf{x}_k, y_k, z_k)\}_{k=1}^{n+m}$, where

$$(\mathbf{x}_k, y_k, z_k) = \begin{cases} (\mathbf{x}_k, y_k, 0), & \forall\, (\mathbf{x}_k, y_k) \in \mathscr{D}_L, \\ (\mathbf{x}_{k-n+1}, *, 1), & \forall\, \mathbf{x}_{k-n+1} \in \mathscr{D}_U, \end{cases}$$

where $*$ corresponds to a missing value for the label y. In addition, we define the monotonic function $g : \mathbb{R}^+ \rightarrow [0,1]$ as a map from positive real values to the unit interval. Given the variational distribution q_ϕ, our prediction of whether the feature $\mathbf{x}$ comes from the source or target distributions is

$$\hat{z}_\phi(\mathbf{x}) \triangleq g\Big(\mathbb{V}[q_\phi(y|\mathbf{x}, \mathscr{D})]\Big), \tag{3.6}$$

which follows directly from the condition in (3.5). Given (3.6), we define the regulariser Ω in (3.4) as the cross-entropy loss between predicted and true auxiliary variables, $\hat{z}$ and z, i.e.,

$$\Omega(q_\phi(\theta \,|\, \widetilde{\mathscr{D}})) = \sum_{k=1}^{n} \log\left(1 - \hat{z}_\phi(\mathbf{x}_k)\right) + \sum_{k=n+1}^{n+m} \log\left(\hat{z}_\phi(\mathbf{x}_k)\right). \tag{3.7}$$

Thus, our variational posterior is obtained by plugging the regulariser $\Omega(q_\phi(\theta \,|\, \widetilde{\mathscr{D}}))$ in (3.4) and solving for ϕ, with the optional inclusion of a hyperparamter λ to control the level of regularisation. The exact choice of g can as well be controlled although from our experiments it made little difference, and we settled on $g(x) = 1 - \frac{1}{1+x}$. We note that this regularisation scheme addresses the issue of over-confident predictions on the target set without taking the naive approach of just increasing the variance everywhere — it is balanced by the location of the source data set that will lower the variance in our appropriately confident locations. Since the regulariser above solves the *transductive* learning problem of classifying source and target data in a way that resembles semi-supervised learning (Rohrbach et al., 2013), we call Ω a transductive regulariser. In what follows, we propose a practical way to implement transductive regularisation within the MCDP approximate inference framework.

Transductive Dropout. We extend the MCDP approximate inference scheme in Gal and Ghahramani (2016) by applying our posterior regularisation penalty, and allowing the dropout rates to vary per data point, dependent on the feature values. By enabling the dropout rates to be a function of $\mathbf{x}$, we provide more degrees-of-freedom to flexibly craft the posterior variance $\mathbb{V}[q_\phi]$ so that it accurately discriminates source and target data points.

Let p be the dropout rate of the underlying NN model. We parameterise p to be dependent on the feature value $\mathbf{x}$ as follows. Let $v_\beta(.)$ be a neural network with a sigmoid output layer and parameters β, i.e., $v_\beta : \mathbb{R}^d \to [0,1]$ maps feature values to dropout rates so that $p = v_\beta(\mathbf{x})$. This equates approximately to a surrogate distribution over the weights:

$$q_\phi(\mathbf{w}) = \prod_{i=1}^{N} (1 - v_\beta(\mathbf{x}))^{\frac{w_i}{m_i}} \, v_\beta(\mathbf{x})^{\frac{m_i - w_i}{m_i}} \tag{3.8}$$

for $w_i \in \{m_i, 0\}$, 0 otherwise, where $\mathbf{w} = \{w_i\}_i$ is the set of weights for the NN modelling the conditional distribution $q_\phi(y|\mathbf{x}, \mathscr{D})$. Bear in mind this is not exactly the case but is reflective of the approximation and we use a concrete relaxation of the Bernoulli distribution to allow for the reparameterisation trick to get derivatives as detailed in Gal et al. (2017). This leaves an optimisation objective (of the form in (3.4)) over the variational parameters $\phi = \{\beta, \mathbf{m}\}$. Using the equivalence between KL minimisation and squared loss minimisation under dropout regularisation, we can write the objective function in (3.4) as

$$R(\phi) = \sum_{\mathbf{x}_i \in \mathscr{D}_L} \|\mathbb{E}[q_\phi(y_i \,|\, \mathbf{x}_i)]\|_2^2 + \Omega(q_\phi(\theta \,|\, \widetilde{\mathscr{D}})), \tag{3.9}$$

with the possibility of adding an ℓ_2 regulariser $\|\phi\|_2^2$ as well. As we can see, this objective incorporates both labelled and unlabelled data: the data set $\mathscr{D}_L$ contributes to the first term, which is concerned with fitting the observed labels drawn from the source distribution, whereas the second term, which depends on the entire augmented data set $\widetilde{\mathscr{D}}$, makes sure that the induced variational posterior is aware of the mismatch between source and target feature distributions. We can see that this scheme, as depicted in figure 3.1, acts in a similar way to (3.3), primarily optimising the likelihood of the data under the approximation while constrained by a reqlariser on the form of the distribution, only now the regulariser induces more specific behaviour and makes use of $\mathscr{D}_U$.

The regulariser in (3.9) can be computed in backpropagation using sample estimates of the posterior variance as follows. Let $\tilde{\phi}$ be the current estimate of the variational parameters at a given iteration of the gradient descent procedure. To evaluate the model loss and gradients, we use the MCDP forward pass to sample M outputs $\{\hat{y}_k^1, \ldots, \hat{y}_k^M\}$ for every $\mathbf{x}_k$ in $\widetilde{\mathscr{D}}$, and compute a Monte Carlo sample estimate of the transductive regularisation term as follows:

$$\widehat{\Omega}(q_{\tilde{\phi}}(\theta \,|\, \widetilde{\mathscr{D}})) = g\left(\frac{1}{M} \sum_{m=1}^{M} (\hat{y}_k^m - \bar{y}_k)^2\right). \tag{3.10}$$

Computations of the estimator in (3.10) only involve the forward pass, and evaluating its gradients is straightforward.

Key insights Figure 3.2 provides a pictorial depiction of our transductive dropout inference procedure applied to an exemplary, one-dimensional feature space. A key insight is that transductive dropout inference learns to adapt the dropout rate so that it induces larger posterior uncertainty for regions with dense concentration of unlabelled data, but low density for labelled data.

Chapter 4

An Approximate Bayesian Approach to Direct Policy Learning

In chapter 2 we discuss that Huyuk et al. (2020) introduce their algorithm DIPOLE for direct policy learning by modelling the agent's decision dynamics as an IOHMM and their policy in terms of distances to "mean-vectors" on the belief simplex. This produces a transparent representation of how the agent arrives at their actions and allows us to inspect important aspects that we simply could not using for example deep behavioural cloning methods. What is conspicuously absent however is a handling of the associated uncertainty in the decision dynamics as only the maximum likelihood estimator of the model parameters is learnt. For the purpose of understanding the agent it is important for us to be able to capture when we are unsure about their actions due to both not seeing enough example data but also when there is natural variation in how they act.

To that end we now derive and establish a stochastic variational inference scheme for learning the posterior distribution over both the model parameters and latent hidden states.

4.1 Preliminaries

Let us first formally introduce the decision dynamics model that we will use to represent how the agent both interacts with, and understands, the environment, which was established in Huyuk et al. (2020). We consider such a decision-making environment with partial observability, where decisions are made over discrete time steps. At each time step $t \in \mathbb{N}_+$, the decision-maker takes an action a_t chosen from the finite action space A and observes an outcome z_t from the finite observation space Z. We are interested in inferring the policy π_b of the decision-maker (i.e. the *behaviour policy*) given an observational data set $\mathscr{D} = \{(a_1^{(i)}, z_1^{(i)}, \ldots, a_{\tau^{(i)}}^{(i)}, z_{\tau^{(i)}}^{(i)})\}_{i=1}^n$ of n-many *demonstrations* from the decision-maker, where $a_t^{(i)}$ is the action taken, and $z_t^{(i)}$ is the observation made, at step t during the ith demonstration, with $\tau^{(i)}$ the (max) time horizon of the given demonstration.[1]

Denote by $h_t = (a_1, z_1, \ldots, a_{t-1}, z_{t-1})$ the observed history at the beginning of time step t, with $h_1 = \emptyset$. Let $H_t = (A \times Z)^{t-1}$ indicate the set of all possible histories at the beginning of time step t, with $H_1 = \{\emptyset\}$. Finally, let $H = \cup_{t=1}^{\infty} H_t$ denote the set of all possible histories. Then, a proper policy acting in the decision-making environment that is described would be a mapping $\pi : H \to \Delta(A)$ from observed histories to action distributions, where $\pi(a|h)$ is the probability of taking action a having observed history h.

The space of H becomes exponentially complicated over time and so we require some method to simplify a handling of all past possible histories. To do so we assume that the decision-maker acts with respect to their belief over some underlying (unobserved) state of the environment and that crucially they aggregate all available information into this belief. This gives rise to an Input-Output Hidden Markov Model (IOHMM) (Bengio and Frasconi, 1995) over the belief dynamics of the decision-maker.

Formally, an IOHMM is identified by the tuple (S, A, Z, b_1, T, O), where:

- S is the finite set of (unobservable) states;
- A is the previously defined set of actions;
- Z is the previously defined set of observations;
- $b_1 \in \Delta(S)$ is the initial state distribution, where $b_1(s)$ denotes the probability of state s being the initial state;
- $T : S \times A \to \Delta(S)$ is the transition function, where $T(s'|s, a)$ is the probability of transitioning into state s' after taking action a in state s; and
- $O : A \times S \to \Delta(Z)$ is the observation function, where $O(z|a, s')$ is the probability of observing z after taking action a and transitioning into state s.

[1] For brevity, we will omit indices (i) unless explicitly required.

Among the elements of this tuple, the spaces S, A, and Z are known, while the parameters b_1, T, and O are unknown. The belief $b_t \in \Delta(S)$ at the beginning of each time step t can be defined such that $b_t(s) = \mathbb{P}(s_t = s|h_t)$ is the probability of state s being the current state given the observed history h_t so far. Note that the initial state distribution b_1 also doubles as the initial belief for each trajectory/demonstration. Given action a_t and observation z_t, the subsequent belief b_{t+1} can easily be expressed in terms of the current belief b_t:

$$b_{t+1}(s') \propto \sum_{s \in S} b_t(s) T(s'|s, a_t) O(z_t|a_t, s') \; . \tag{4.1}$$

4.1.1 Parameterising Policies

Having introduced beliefs, as well as a map from histories into beliefs, policies can now be reasonably defined as mappings $\pi : \Delta(S) \to \Delta(A)$ from beliefs to action distributions, where $\pi(a|b)$ is the probability of taking action a when the current belief is b. We parameterise policies in terms of $|A|$-many "mean" vectors, each corresponding to an action in A, and living (like the belief b) on the $|S|$-dimensional simplex. Which action is taken then is defined by the belief's relative *distances* from the actions' mean vectors, formalised through the radial basis function kernel (Park and Sandberg, 1991) such that:

$$\pi(a|b) = \frac{e^{-\eta \|b - \mu_a\|^2}}{\sum_{a' \in A} e^{-\eta \|b - \mu_{a'}\|^2}} \; , \tag{4.2}$$

where $\eta \geq 0$ is the inverse temperature, $\|\cdot\|$ the ℓ_2-norm, and $\mu_a \in \mathbb{R}^{|S|}$ the mean vector corresponding to $a \in A$.

Intuitively, these mean vectors are interpreted in terms of the decision boundaries (and decision regions) that they induce over the belief space $\Delta(S)$. Given a belief, the action whose corresponding mean is the closest one to that belief is more likely to be taken than any other action. Hence, the beliefs that are closest to the mean of a particular action form a *decision region* where that action is the most likely one to be taken, and similarly the lines that are equidistant to the means of two actions form the *decision boundary* between those two actions.

The inverse temperature η controls how "smooth" the decision boundaries between the regions are (i.e. how smooth the transitions are between regions). Larger ηs induce more deterministic policies (where behavior changes more abruptly between decision boundaries), whereas smaller ηs induce more stochastic policies. In the extremes, $\eta = 0$ describes the case where actions are taken uniformly at random regardless of the given belief, and $\eta = \infty$ recovers the case of actions being taken deterministically per decision regions.

4.2 Learning the Approximate Posterior

With the model defined, we come to our contribution, where our aim is now to reason about the unknown quantities given our observational dataset. Specifically, in order to coherently deal with uncertainty appropriately we would like to uncover the posterior distribution of the model parameters, as well as the belief over the underlying states denoted by $\mathbf{s}$, given our data. The task is then summarised as:

$$\begin{aligned} Given &: \mathscr{D}, S, A, Z \\ Determine &: p(b_1, T, O, \eta, \{\mu_a\}_{a\in A}, \mathbf{s}|\mathscr{D}) \ . \end{aligned}$$

For simplicity we shall denote the collection of all the unknown parameters by $\theta = (b_1, T, O, \eta, \{\mu_a\}_{a\in A})$. Given the complication of the model a simple application of Bayes rule yields a completely intractable posterior and so we shall have to resort to variational inference methods for learning a principled approximation to the distribution.

Central to Bayesian learning is the quantity $\log p(\mathscr{D})$, the log marginal evidence of the observed data, which in our model is similarly intractable to evaluate. However by introducing an auxiliary distribution over θ and $\mathbf{s}, q(\theta, \mathbf{s})$ we can lower bound it using Jensen's inequality:

$$\begin{aligned} \log p(\mathscr{D}) &= \log \int\int p(\mathscr{D}, \theta, \mathbf{s}) d\theta d\mathbf{s} && (4.3) \\ &= \log \int\int q(\theta, \mathbf{s}) \frac{p(\mathscr{D}, \theta, \mathbf{s})}{q(\theta, \mathbf{s})} d\theta d\mathbf{s} && (4.4) \\ &\geq \int\int q(\theta, \mathbf{s}) \log \frac{p(\mathscr{D}, \theta, \mathbf{s})}{q(\theta, \mathbf{s})} d\theta d\mathbf{s}. && (4.5) \end{aligned}$$

We assume a factorisation $q(\theta, \mathbf{s}) = q(\theta)q(\mathbf{s})$, leading to:

$$\begin{aligned}
\log p(\mathscr{D}) &\geq \int\int q(\theta)q(\mathbf{s})\log\frac{p(\mathscr{D},\mathbf{s}|\theta)p(\theta)}{q(\theta)q(\mathbf{s})}d\theta d\mathbf{s} && (4.6)\\
&= \int\int q(\theta)q(\mathbf{s})\Big[\log\frac{p(\theta)}{q(\theta)} + \log\frac{p(\mathscr{D},\mathbf{s}|\theta)}{q(\mathbf{s})}\Big]d\theta d\mathbf{s} && (4.7)\\
&= \int\int q(\theta)q(\mathbf{s})\Big[\log\frac{p(\theta)}{q(\theta)}\Big]d\theta d\mathbf{s} + \int\int q(\theta)q(\mathbf{s})\Big[\log\frac{p(\mathscr{D},\mathbf{s}|\theta)}{q(\mathbf{s})}\Big]d\theta d\mathbf{s} && (4.8)\\
&= \int q(\theta)\Big[\log\frac{p(\theta)}{q(\theta)} + \int q(\mathbf{s})\log\frac{p(\mathscr{D},\mathbf{s}|\theta)}{q(\mathbf{s})}d\mathbf{s}\Big]d\theta && (4.9)\\
&= \mathscr{F}(q(\theta),q(\mathbf{s})) && (4.10)
\end{aligned}$$

where $\mathscr{F}(q(\theta),q(\mathbf{s}))$ is the ELBO (as similarly defined in the Chapter 3). It can be seen that maximising with respect to $q(\theta)$ and $q(\mathbf{s})$ results in minimising the KL divergence between the surrogate distributions and the true posterior, $KL\big[q(\theta,\mathbf{s})||p(\theta,\mathbf{s}|\mathscr{D})\big]$.

Now the question becomes what would be an appropriate form for $q(\mathbf{s},\theta)$ to take? For $q(\mathbf{s})$ it is simple to use a categorical distribution over the possible states, while in order to approximate the posterior distribution, whose components have restricted domains we employ a mean-field factorisation given by:

$$\begin{aligned}
b_1(\cdot) &\sim \text{Dirichlet}(\{\alpha_s^{b_1}\}_{s\in S}),\\
T(\cdot|s,a) &\sim \text{Dirichlet}(\{\alpha_{s,a,s'}^{T}\}_{s'\in S}),\\
O(\cdot|s,a) &\sim \text{Dirichlet}(\{\alpha_{s,a,z}^{O}\}_{z\in Z}),\\
\eta &\sim \text{Gamma}(\alpha^\eta,\beta^\eta),\\
\mu_a &\sim \mathscr{N}(\bar{\mu}_a,\sigma_a).
\end{aligned}$$

We denote this joint distribution $q_\phi(\theta)$, taking ϕ to be the collection of all the given parameters.

Given the factorisation between latent variables $\mathbf{s}$ and model parameters θ we can make use of a variational Bayesian expectation maximisation algorithm for iteratively updating each of the distributions separately. Variational Bayesian methods have previously been applied to HMM inference (effectively learning b_1, T, and O) in Beal (2003). We extend those methods to direct policy learning (by jointly learning η and μ_as as well). While the E-step remains broadly similar, for a joint solution we have to depart significantly from those methods in the M-step as the policy breaks the conjugate-exponential properties of traditional HMMs.

4.2.1 The Variational Bayesian E-Step

This is the more familiar of the two steps, where we hold $q_\phi(\theta)$ constant and update $q(\mathbf{s})$ in order to increase the ELBO. Taking the variational derivative of $\mathscr{F}(q(\theta), q(\mathbf{s}))$ with respect to $q(\mathbf{s}))$ and setting to zero we have that:

$$\begin{aligned}
\log q(\mathbf{s}) &= \mathbb{E}_{q(\theta)}[\log p(\mathscr{D}, \mathbf{s}|\theta)] - C && (4.11)\\
&= \mathbb{E}_{q(\theta)}\Big[\sum_{i=1}^{n}\Big(\log b_1(s_1) + \sum_{t=1}^{\tau}\log \pi_\gamma(a_t|b_t) + \sum_{t=1}^{\tau}\log O(z_t|s_t, a_t) + \\
&\qquad \sum_{t=1}^{\tau-1}\log T(s_{t+1}|s_t, a_t)\Big)\Big] - C && (4.12)\\
&= \sum_{i=1}^{n}\Big(\mathbb{E}_{q(b_1)}\big[\log b_1(s_1)\big] + \sum_{t=1}^{\tau}\mathbb{E}_{q(O)}\big[\log O(z_t|s_t, a_t)\big] + \\
&\qquad \sum_{t=1}^{\tau-1}\mathbb{E}_{q(T)}\big[\log T(s_{t+1}|s_t, a_t)\big]\Big) - C && (4.13)
\end{aligned}$$

which should look familiar in that it is the same objective that is solved by the traditional forward-backward algorithm for Hidden Markov Models (Rabiner and Juang, 1986). There is a small difference though in that the parameters are now taken to be the expected value of the log of the parameters, which is now what we need to work with. They can be calculated as:

$$\hat{b}_1(s) = \exp\Big[\psi(\alpha_s^{b_1}) - \psi\Big(\sum_{i=1}^{|S|}\alpha_{s_i}^{b_1}\Big)\Big], \tag{4.14}$$

$$\hat{T}(s'|s, a) = \exp\Big[\psi(\alpha_{s,a,s'}^{T}) - \psi\Big(\sum_{i=1}^{|S|}\alpha_{s,a,s_i}^{T}\Big)\Big], \tag{4.15}$$

$$\hat{O}(z|s, a) = \exp\Big[\psi(\alpha_{s,a,z}^{O}) - \psi\Big(\sum_{i=1}^{|Z|}\alpha_{s,a,z_i}^{Z}\Big)\Big], \tag{4.16}$$

where ψ is the digamma function (Beal, 2003). These result in sub-normalised probabilities that will change the normalisation constant in the forward messages but otherwise will not have an effect on the posterior . Thus the E-step consists simply of evaluating the expected value of the log of the belief dynamics parameters before running the usual forward-backward algorithm to get state marginals.

4.2.2 The Variational Bayesian M-Step

Handling the maximisation step is slightly more complicated, in that there is no closed-form solution to be applied. Beal (2003) leveraged the conjugate-exponential properties of the Dirichlet distribution and likelihood to derive simple update rules for the variational parameters. In our model, the introduction of the policy completely breaks this conjugacy and renders such a solution impossible. Thus we will have to resort to a stochastic variational inference optimisation procedure based on a Monte Carlo approximation of the ELBO. We can re-write the ELBO as:

$$\mathscr{F}(q(\theta), q(\mathbf{s})) = \mathbb{E}_{q(\theta)}\Big[\mathbb{E}_{q(\mathbf{s})}\big[p(\mathscr{D}|\mathbf{s},\theta)\big]\Big] - KL(q(\theta)||p(\theta)) - KL(q(\mathbf{s})||p(\mathbf{s})). \tag{4.17}$$

Holding $q(\mathbf{s})$ constant in the M-step we're interested in optimising:

$$\mathscr{F}(q(\theta)) = \underbrace{\mathbb{E}_{q(\theta)}\Big[\sum_{\mathbf{s}} p(\mathscr{D}|\mathbf{s},\theta)q(\mathbf{s})\Big]}_{\text{Monte Carlo estimate}} - \underbrace{KL(q(\theta)||p(\theta))}_{\text{Analytically tractable}}. \tag{4.18}$$

Instead of the traditional route of fully optimising this quantity at every step and saturating the bound, we will satisfy ourselves with simply taking a single gradient step in the *right* direction and wait for convergence. This requires gradients of (4.18) with respect to ϕ - fortunately given our choice of $q_\phi(\theta)$, the second (KL) term is analytically tractable. The first (expected log likelihood) term is not however, but can be easily approximated through Monte Carlo sampling. This, alongside the reparameterisation trick for pathwise gradients (Kingma and Welling, 2013), allows us to get low-variance estimators for the gradient of the ELBO with respect to ϕ. We should note that the standard reparameterisation trick applies to neither Dirichlet nor Gamma distributions, although more recent work on implicit pathwise gradient estimators (Figurnov et al., 2018) has now made them possible.

4.3 Disentangling Uncertainty in Actions

The combination of a parameterised policy that is inherently stochastic alongside a probabilistic treatment of the model parameters allows for a useful factorisation of the uncertainty over the predictive distribution that was not previously possible, which we obtain by integrating out the parameters:

$$\underbrace{\pi(a|b)}_{\textbf{Total}} = \int \underbrace{\pi(a|b,\mu,\eta)}_{\textbf{Data}} \underbrace{q(\mu,\eta)}_{\textbf{Policy}} \underbrace{q(b|T,O,b_1)q(T,O,b_1)}_{\textbf{Dynamics}} d\theta, \tag{4.19}$$

where $q(b|T,O,b_1)$ is the predictive distribution over the belief given the past history of the trajectory and the dynamics (note as a repeated application of equation 4.1 it is a deterministic map to a distribution over states given the dynamics parameters). The data uncertainty reveals where there is intrinsic uncertainty and variation in practice in the observed data, while the policy and dynamics uncertainty tell us where we may have uncertainty in the decision-makers belief system due to a lack of information. This second type of uncertainty is equally important, both to drive potential future data acquisition but also to point out if natural inequalities in both actions and transitions do arise in the data.

4.4 On the Implementation

The training procedure is then summarised in algorithm 1. The only real potential difficulties arise in the evaluation of the gradient, because of the belief system the dynamics have a compounding effect as you progress through a trajectory. This means we have to differentiate through all the beliefs leading to computational complexity that scales at $\mathcal{O}(\tau^2)$. Fortunately though this can be handled simply by modern automatic differentiation packages, which in many cases also automatically reparamteterise samples to allow for easy gradients of ϕ.

```
Result: Parameters $\phi$ of variational distribution
Input: $\mathscr{D}, S, A, Z$ ;
Initialise $\phi$;
Set learning rate $\lambda$;
while not converged do
    Calculate expected values of $\hat{\theta}$ ;
    Forward-backward algorithm for $q(\mathbf{s})$ ;          ▷ VBE-step
    Sample $\theta$;
    Evaluate $\nabla_\phi \mathscr{F}(q(\theta))$;
    $\phi \leftarrow \phi + \lambda \nabla_\phi \mathscr{F}(q(\theta))$ ;          ▷ VBM-step
end
Return: $\phi$
```

Algorithm 1: Variational DIPOLE

Chapter 5

InterPoLe: Soft Decision Trees for Building Interpretable Policies

Having examined how we can appropriately handle uncertainty in policy learning, we now turn to the question of how we can produce policies that are the most interpretable and useful to the medical community, remembering our aim is to *understand* the general decision making process of the doctors. Certainly neural networks are a hard sell and in Variational DIPOLE policies are still defined in terms of slightly abstract “distances” in the belief space.

With this in mind we propose InterPoLe, an algorithm for *interpretable policy learning* where given the actions and observations of an agent we learn an interpretable representation of their empirical decision making process. Specifically we use a novel soft decision tree architecture to parameterise their policy from an internal representation (or *belief*) into actions in a comprehensible, hierarchical structure of simple binary questions. These internal decision maker dynamics model a policy from beliefs into actions - modelling how confidence in the underlying state of the environment translates into behaviour. This makes intuitive sense, a doctor does not treat a patient *exactly because* their temperature is high, rather having seen their temperature is high they are confident the patient is ill and so they then treat them. However it is important to also consider how this translates into reacting to observations and we show that using the learnt dynamics we can at every time step induce a new decision tree over observations. We see this as a system to help and augment the decision making process of clinicians as it is important to feedback information to doctors about what it is they look like they’re doing - this is summarised in figure 5.1.

Table 5.1 Summary of the key features of related work.

Work	Dynamics	Observability	Direct	Black-box
Choi and Kim (2011b)	Known	Fully	No	No
Ramachandran and Amir (2007)	Known	Fully	No	No
Ziebart et al. (2008)	Known	Fully	No	No
Choi and Kim (2011a)	Known	Partially	No	No
Makino and Takeuchi (2012)	Unknown	Partially	No	No
Ho and Ermon (2016)	Online	Partially	Yes	Yes
Li et al. (2017)	Online	Partially	Yes	Yes
Englert et al. (2013)	Unknown	Fully	Yes	No
Ude et al. (2004)	Unknown	Fully	Yes	No
Van Den Berg et al. (2010)	Unknown	Fully	Yes	No
(Ours)	Unknown	Partially	Yes	No

Why are decision trees better? In the medical community, guidelines are almost exclusively given in the form of decision trees (Chou et al., 2007; Qaseem et al., 2012), as clinicians agree they are the best way to simply break down steps and guidance in order to limit confusion. Interestingly though while they are set on expert advice, they are often left vague such that there is room for individual medical professionals to use their own judgement, leading to substantial variability in clinical practice (O'Sullivan et al., 2018). An additional use for our method, other than attempting to understand the decision making process of individual doctors, would allow for quantifying exactly how they appear to implement these guidelines. This information could be used when reviewing and updating advice, particularly in seeing how closely it appears they are being followed.

Being in the medical setting, we are reminded that this places several key restrictions on the types of methods that are applicable: (1) It must be **offline**, there is no capacity to allow an agent (especially an untrained one) to interact with the environment and real patients in order to collect data; (2) It must work in a **partially-observable environment**, as clearly we are nowhere close to a full understanding of this setting and many aspects of patients true health are unavailable most of the time; and (3) it must directly parameterise an **interpretable** policy, we require that humans can follow and understand the policy, being able to explain the actions in order to compare with their own decision making process. Table 5.1 then summarises the position of our algorithm within the current literature (and methods mentioned in chapted 2) with respect to these desiderata. We note that we are the first to directly work in a model-free/offline, partially observed setting with a focus on interpretability.

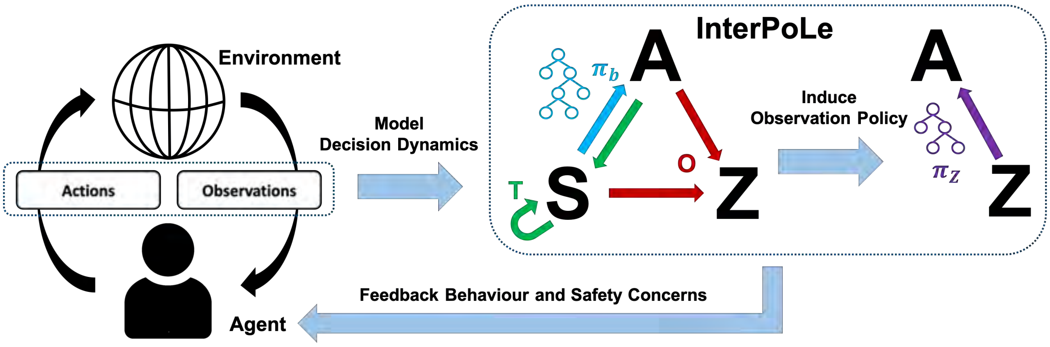

Fig. 5.1 **Our method in practice.** Using only action/observation pair trajectories we learn the transition (T) and observation (O) functions of the IOHMM belief model as well as a decision tree policy (π_b) from the state space (S) to the action space (A). Once learnt, these dynamics induce a new decision tree policy (π_Z) from the observation space (Z), that adaptively changes over the course of each trajectory. It is then important to share with the original agent how we believe they are acting so they can properly align their intentions and actions as well as catch potential oversights.

5.1 Belief-policy learning directly

We adopt the decision dynamics model of chapter 4, that is to say we still assume that the agent considers an IOHMM model of the environment. However we will no longer assume the same parameterisation of the policy π, instead we model the policy of the agent to be a soft decision tree $\pi_\gamma : \Delta(S) \mapsto \Delta(A)$; we will leave the exact details of the architecture for later, for the moment it is sufficient to assume π_γ is a function that takes in a belief over states and outputs a probability distribution over actions while being differentiable w.r.t. its parameters γ. Given this parameterisation of the decision dynamics we now discuss how they can be jointly optimised, unlike in chapter 4 we don't seek a posterior distribution over the parameters, simply their maximum likelihood estimate. The task amounts to learning b_1, T, O, and γ once we are given $\mathscr{D}, S, A$, and Z. We can do this by making use of a standard (non variational Bayesian) EM algorithm.

Notationally, let $\theta = (b_1, T, O, \gamma)$ be the collection of all objects we would like to estimate, we need to also consider the set of true but unobserved state trajectories, $\mathbf{s}$, such that we can define the complete likelihood of parameters θ given both $\mathscr{D}$ and $\mathbf{s}$, given by[1]:

$$\begin{aligned}\log L(\theta;\mathscr{D},\mathbf{s}) &= \log\mathbb{P}(\mathscr{D},\mathbf{s}|\theta)\\ &= \sum_{i=1}^{n}\Big[\log b_1(s_1)+\sum_{t=1}^{\tau}\log\pi_\gamma(a_t|b_t)+\sum_{t=1}^{\tau}\log O(z_t|s_t,a_t)+\sum_{t=1}^{\tau-1}\log T(s_{t+1}|s_t,a_t)\Big]. \end{aligned}\tag{5.1}$$

Without access to $\mathbf{s}$ this is of course impossible to evaluate or optimise and so we introduce an auxiliary objective, given an estimate of the parameters $\hat{\theta}'$ we define the *expected* log-likelihood as:

$$Q(\theta,\hat{\theta}') = \mathbb{E}_{\mathbf{s}|\mathscr{D},\hat{\theta}'}\log L(\theta;\mathscr{D},\mathbf{s}) \tag{5.2}$$

$$= \sum_{\mathbf{s}}\log L(\theta;\mathscr{D},\mathbf{s})\mathbb{P}(\mathbf{s}|\mathscr{D},\hat{\theta}'). \tag{5.3}$$

The optimisation procedure is summarised in algorithm 2 and proceeds as follows: first given a set of parameters we calculate and fix the posterior distributions over states, $\mathbb{P}(\mathbf{s}|\mathscr{D},\hat{\theta}')$, using the forward-backward algorithm; second, given this posterior, we calculate the gradient of the expected log-likelihood $\nabla_\theta Q(\theta,\hat{\theta}')$ and find a new value of θ such that it increases, for example by taking a single step of gradient ascent. The calculation of the gradient is easily handled by automatic differentiation packages, as beliefs are calculated successively in a feed forward manner they can be back-propagated through.

```
Result: Maximum likelihood estimator for $\theta$
Input: $\mathscr{D},S,A,Z$ ;
Initialise $\hat{\theta}$;
Set learning rate $\lambda$;
while not converged do
    Calculate $\mathbb{P}(\mathbf{s}|\mathscr{D},\hat{\theta})$;                ▷ E-step
    Evaluate $\nabla_\theta Q(\theta,\hat{\theta})$;
    $\hat{\theta} \leftarrow \hat{\theta} + \lambda\nabla_\theta Q(\theta,\hat{\theta})$ ;      ▷ M-step
end
Return: $\hat{\theta}$
```

Algorithm 2: InterPoLe

[1] Again we drop the trajectory index i for brevity unless explicitely required

5.2 Soft Decision Trees for Unstructured Data

We now discuss the decision tree architecture that we shall use in the algorithm. Decision trees are unparalleled in the machine learning landscape for their interpretability and clarity in how predictions are made, their hierarchical structure of simple binary questions makes it easy to follow the decision making process. By comparison, neural networks and kernel methods rely on complex abstractions that while achieving state-of-the-art performance are difficult, if not impossible, to comprehend. Unfortunately, classical decision trees are fundamentally discrete objects, they are traversed deterministically until you arrive at a leaf node that returns a single class, making them incompatible with gradient based optimisation methods.

In general a decision tree is a hierarchical tree structure comprising of individual nodes. For some input, $x \in \mathbb{R}^d$, each non-leaf node n will apply some decision rule, or gating function, $g_n(x)$ that determines which path out of its children to return (Loh, 2011). Most of the more recent work revolves around how we can design an interesting gating function, which is the area our architecture focuses on also.

In most classical and typical hard decision trees, $g_n(x)$ inspects a single feature of x to divide the space. Extending to more than one feature, multivariate linear trees take a linear combination of features to discriminate (John, 1996), and can be simply extended to non-linear combinations as well (Guo and Gelfand, 1992). In soft decision trees $g_n(x)$ becomes probabilistic. Beginning with the hierarchical mixture of experts, (Jordan and Jacobs, 1994) introduce using generalised linear models as the gating function that outputs a probability for weighting the nodes children. Most commonly, logistic regression is used, where $g_n(x)$ outputs a linear combination of the features passed through a sigmoid function. This idea is extended in (Irsoy et al., 2012) where they use that gating function but also learn where it is appropriate to split nodes and grow the tree. Most recently they have been used as well to build on a neural network model but learn an interpretable representation (Frosst and Hinton, 2017). Our model differs from previous work significantly in the parameterisation of $g_n(x)$.

Formally let $\pi_\gamma : \mathbb{R}^d \mapsto \Delta(\{1,\ldots,k\})$ be a soft decision tree of depth L and parameters γ, comprising of layers $l = \{1,\ldots,L\}$ each containing 2^{l-1} nodes. Let $n_{i,j}$ denote the j^{th} node of the i^{th} layer, as well as $n^l_{i,j}$ and $n^r_{i,j}$ denote the node's left and right children respectively should they exist. Each non-leaf node is connected to two children in the subsequent layer which have them as a unique parent. A forward pass through a node consists of calculating a split probability $p = g_n(x) \in [0,1]$ and returning a weighted sum of the values returned by their children, thus a forward pass of the tree starts at the root node and returns a weighted sum of the leaf node values which we consider a categorical distribution parameterised by a vector of length k passed through a softmax function. Ultimately then the predictive distribution of the tree can be decomposed into a mixture of categoricals where the mixing proportions are calculated based on the input. In this work we introduce a new parameterisation of the split probability, given by:

$$g_n(x) = \prod_i^d \frac{1}{1+\exp(-\alpha_i(x_i - \eta_i))}, \tag{5.4}$$

where for each dimension of the input, $i \in \{1,\dots,d\}$ there is an associated real valued steepness parameter α_i and location parameter η_i. We can understand the relationship between this architecture and a classical decision tree by observing that at each node every dimension of the input is passed through a soft step function, the location of which is set by the η parameter, while the steepness and direction is set by α, then taking the product at the end acts as a soft AND gate (all dimensions must be close to one for p to be also). Crucially, this allows us to recover the rules of classical decision trees as the α parameters goes to $\pm\infty$, and further means a trained soft tree can be *hardened*, approximated well by an equivalent classical decision tree (replace with a hard step function with the location given by η and the direction given by the sign of α).

5.3 Evolving Policies in the Observation Space

While a description of an agent's policy in terms of beliefs is useful for understanding why an agent makes its decisions, the nature of the hidden states produces an abstraction such that it is non-obvious how interaction with the environment affects the decision making process. Thus we note that the learnt dynamics induce at every time point a possible different policy over observations. This can be seen considering that beliefs are continually updated at each time step given by:

$$b_{t+1}(s') \propto \sum_{s\in S} b_t(s)T(s'|s,a_t)O(z_t|s,a_t), \tag{5.5}$$

which given the learnt values of T and O defines a function $f : Z \times \Delta(S) \times A \mapsto \Delta(S)$ which takes in an observation as well as a belief state and action taken and returns the new belief state. We denote it as $f(z;b_t,a_t)$ which allows us to define the generalised inverse $f^{-1}(b_{t+1};b_t,a_t) = \{z : f(z;b_t,a_t) = b_{t+1}\}$. Intuitively this inverse function answers the question as to if an agent is

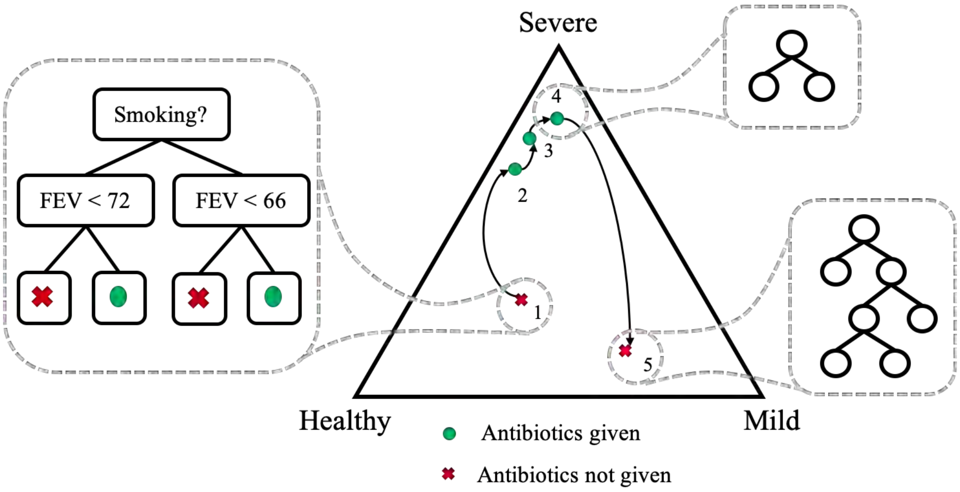

Fig. 5.2 **Example trajectory.** An example trajectory from the cystic fibrosis data we introduce later. Starting in the beginning, initial observations suggest a severe case and treatment is given over a few time steps, then the observations improve and the belief moves towards the mild state where treatment is not deemed necessary. Note that each time step is a result of a unique history and induces a different observation decision tree every time.

in a state and takes an action, what observation from the environment would cause its belief to update in a specific way. We can then define the observation policy at some belief and previous action as: $\pi_{Z;b_t,a_t}(z_t) = \pi_\gamma(f(z_t;b_t,a_t))$.

While every type of policy over the belief space induces some observation policy, passing f and f^{-1} through the policy can be tricky. The decision tree structure allows us to handle this easily as the policy can be broken down into successive simple rules. We will assume to be dealing with a hardened tree since inverting a probabilistic traversal of the tree adds practical though not theoretic complication. Given the decision tree structure of the policy over beliefs, an equivalent policy over observations can be obtained by sequentially going through every condition of every node, which will amount to whether the next belief b_{t+1} is in some decision region D, and replacing the condition with: $z_t \in D_Z = \{f^{-1}(b_{t+1};b_t,a_t) : b_{t+1} \in D\}$. We thus arrive at a sequential set of rules, defined over the observation space that dictate which action should be taken. Figure 5.2 shows an example of how this appears in practice.

5.4 Feature Importance in Error Detection

We have shown that InterPoLe can be used to obtain from past demonstrations an interpretable representation of the decision maker's empirical decision dynamics. Moving forward then the task becomes slightly different; with this description in hand and assuming the decision maker aims to continue to follow this policy, can we detect when they deviate, and moreover where specifically the deviation comes from?

The probabilistic interpretation of the soft decision tree allows for a formalism through which we can identify *unexpected* actions and inform when it is appropriate to alert users to potential mistakes. We assume that θ is a sufficient statistic for $\mathscr{D}$, in that once we've trained the model we can make no further use of $\mathscr{D}$, and that the agent *should* be acting according to the policy. It is then simple to set a threshold ε and wait for the agent to make a new action a_t at some belief b_t, if $\pi_\gamma(a_t|b_t) < \varepsilon$ we can send an alert that this action appears sufficiently unlikely that we are concerned. The more interesting question is whether we can see how a potential mistake might have been made. We make the further assumption that the agent's new action a_t is arrived at by following the policy tree π_γ, but allowing for the fact that a mistake may have happened at one of the nodes and was incorrectly followed. We wish to identify potential nodes of interest.

Let $\pi_{n_{i,j}}$ denote the sub-tree consisting of node $n_{i,j}$ and all of its descendants, this can be thought of as returning the probability of actions given you start at a particular place in the tree. Fix the belief state b_t, according to the policy let the optimal action $a' = \underset{a}{argmax}\{\pi_\gamma(a|b_t)\}$. We can assume $a' \neq a_t$, otherwise the action would not be of further interest.

Define:

$$r_a(n_{i,j}) = (p-q)\log\left(\frac{p(1-q)}{q(1-p)}\right),$$

where $p = \pi_{n^r_{i,j}}(a|b_t)$ and $q = \pi_{n^l_{i,j}}(a|b_t)$, to be the relative importance of node $n_{i,j}$ for making action a in belief state b_t. We can think about this as a simplification of the action space to $A' = \{a, A\backslash a\}$ and taking the symmetric KL divergence between the two predictive distributions of the action, i.e. it is equivalently defined:

$$r_a(n_{i,j}) = KL\big(\pi_{n^r_{i,j}}(a|b_t)||\pi_{n^l_{i,j}}(a|b_t)\big) + KL\big(\pi_{n^l_{i,j}}(a|b_t)||\pi_{n^r_{i,j}}(a|b_t)\big).$$

Note that if $r_a(n_{i,j}) = 0$ there is no difference in the prediction of the two children and as such that node has no relevance for the decision as either path has an equivalent probability of picking action a. Further the larger the value of $r_a(n_{i,j})$ the greater the difference in probability of picking a by taking the different path, making the node important in arriving at action a.

With notation defined, let us stop to consider what properties interesting nodes would have. Firstly the values of both $r_{a'}(n_{i,j})$ and $r_{a_t}(n_{i,j})$ should be large - we are searching for a node that splits the path with a' on one side and a_t on the other. This will also ensure that either $\pi_{n^r_{i,j}}(a_t|b_t)$ or $\pi_{n^l_{i,j}}(a_t|b_t)$ is large, which is necessary to explain how the agent arrived at a_t. Thus with these metrics we can order the nodes by their importance in what would have been the expected action, $r_{a'}(n_{i,j})$, before filtering those for which there is not a clear path to a_t. This will provide our most likely points of departure from the tree, allowing us to inspect the nodes and where in the feature space they define their partitions.

Chapter 6

Insights on Medical Data: Understanding Clinical Decision Making

In this chapter we take the methods introduced in chapters 3, 4, and 5 and we apply them to real medical case studies to demonstrate their applicability and benefits in supporting clinical decision making.

6.1 Predicting Prostate Cancer Mortality with Transductive Dropout

Similarly to chapter 3, the content of this first section of the chapter has already been published in Chan et al. (2020)

Background Prostate cancer is the third most common cancer in men, with half a million new cases each year around the world (Quinn and Babb, 2002). It is far more common among the elderly with around 75% of cases occur in men aged over 65 years. Therefore, prostate cancer is expected to bring increasing healthcare burden to countries with ageing population (Hsing et al., 2000). The latest clinical guideline for prostate cancer treatment recommends watchful waiting or non-invasive treatment for early-stage patients who have *low mortality rate* (Heidenreich et al., 2011). Surgery (Radical Prostatectomy) is recommended instead for high-risk patients whose health condition deteriorates rapidly. The patient's survival outlook therefore plays an important role in the treatment decisions. Hence, improved accuracy and uncertainty quantification for mortality prediction will help clinicians to design effective treatment plans and improve patients' life expectancy.

Dataset We consider the problem of predicting and estimating the uncertainty of the mortality rate for patients with prostate cancer. Our training data consists of 240,486 patients enrolled in the American SEER program (SEER, 2019), while for our target data we consider a group of 10,086 patients enrolled in the British Prostate Cancer UK program (UK, 2019). For both sets of patients we have identical covariate data with information concerning the age, PSA, and Gleason scores as well as what clinical stage they're at and which, if any, treatment they are receiving. Note that while we have the same features for both sets this is an area where we expect a level of covariate shift given the different programs and the transition from American to British patients. Indeed we do see this, without giving a full break down of the summary statistics, patients in the Prostate Cancer UK are in general older with higher Gleason scores though not as far along in the clinical stages.

Benchmarks We compare our method against competitive methods from the probabilistic deep learning literature based on their prevalence and applicability. While we consider this work quite different to semi-supervised learning, which do not usually consider improving uncertainty estimates, we also include MixMatch as a benchmark (Berthelot et al., 2019). The methods we consider are:

1. *MLP* - Standard feed forward neural network to benchmark accuracy.

2. *Dropout* - Monte Carlo dropout with rate 0.5 (Gal and Ghahramani, 2016; Srivastava et al., 2014)

3. *Concrete Dropout* - Dropout with the rate treated as an additional variational parameter and is optimised with respect to the ELBO (Gal et al., 2017).

4. *Ensemble* - Ensemble of feed forward MLPs (Lakshminarayanan et al., 2017) with $K = 10$ the number of models in the ensemble.

5. *MixMatch* - We implement a version of the MixMatch algorithm (Berthelot et al., 2019) where we perform one round of label guessing and mixup and without sharpening. As the base predictive model we use a MC Dropout network.

Table 6.1 Area under the ROC curve for two tasks, first correctly predicting the mortality rate of patients in the test set and secondly predicting whether for a given patient the model will make an error. We also report the average confidence interval (CI) length over test predictions, the average standard deviation at miss-classified points (MSD), and the increased number of patients receiving treatment (INPT) using the associated uncertainty in the model and a risk level of 15%.

Method	Test Perf.	Error Pred.	CI Width	MSD	INPT
MLP	0.720 ± 0.012	N/A	N/A	N/A	N/A
MC Dropout	0.729 ± 0.016	0.730 ± 0.016	0.093	0.025	8
Concrete Dropout	0.791 ± 0.012	0.794 ± 0.012	0.151	0.066	76
Ensemble	0.761 ± 0.014	0.782 ± 0.014	0.037	0.018	8
MixMatch	0.728 ± 0.016	0.726 ± 0.016	0.082	0.021	0
LL	0.723 ± 0.014	0.696 ± 0.014	0.073	0.028	22
TDNR	0.836 ± 0.010	0.808 ± 0.011	0.197	0.068	18
Transductive Dropout	0.861 ± 0.009	0.857 ± 0.009	0.130	0.110	189

6. *Last Layer Approximations (LL)* - Approximate inference for only the parameters of the last layer of the network (Riquelme et al., 2018), using Dropout.

7. *Transductive Dropout - No Regularisation (TDNR)* - We implement transductive dropout as described above but without the addition of our variance regulariser to show that the gains are not just down to the ability to adapt the dropout rate to the input.

For all of the neural networks we consider the same architecture of two fully connected hidden layers of 128 units each and tanh activation function. The initial weights are randomly drawn from N(0, 0.1) and all networks are trained using Adam (Kingma and Ba, 2015). Hyperparameter optimisation remains an open problem under covariate shift - we used a validation set consisting of 10% of the labelled data selected, not entirely randomly, but based on propensity score matching in order to obtain a set more reflective of the target data. With this, hyperparemeters were selected for all model through grid search.

Evaluation metrics We consider five evaluation metrics for a comprehensive understanding of the model performance. First, we consider the prediction accuracy as measured by AUROC shown as “TEST PERF.” in table 6.1 (Bewick et al., 2004). Second, we consider the standard deviation of the posterior predictive distribution as a (unnormalised) predictor for whether or not the model will make an error on a given input. The corresponding AUROC score (“ERROR PRED”) measures the agreement between model uncertainty and the chance to predict wrongly, and hence reflects whether the model is well-calibrated. Third, we present the average width of the 95% predictive interval as a measure of general model confidence on unlabelled data (“CI WIDTH”). Next, we show the standard deviation of the predictive distribution on misclassified data (“MISCLASSIFIED SD”). Finally, we show the increased number of patients receiving treatment (INPT) using the associated uncertainty in the model and a risk level of 15%. All quantities related to the posterior distribution are estimated by MC sampling.

Main results First, we note that transductive dropout yields an improvement in the AUROC on the mortality prediction against the other benchmarks, demonstrating that our improved uncertainty calibration does not come at the cost of mean accuracy. Our focus though is on the calibration of our uncertainty estimates. While ultimately it is impossible to properly test how close uncertainty predictions are to what would be the *true* uncertainty, we test by using the posterior predictive variance to classify whether or not the model will make a mistake. The intuition here is that if the model is appropriately uncertain the variance will be high when a mistake is likely and low when not, thus a high performance on using variance as a predictor for when the model will make a mistake should demonstrate appropriate uncertainty estimates. Here we see that transductive dropout significantly outperforms the other benchmarks, suggesting that in general the high variance predictions are indeed associated with those that are more likely to be wrong. We additionally focus on these predictions that each method gets wrong and look at the average standard deviation at each of these points. Here transductive dropout shows on average it’s much less confident about its incorrect predictions than the the other benchmarks, which is the preferred behaviour. It is important to note that this is not at the expense of confidence over all predictions as we show that both concrete dropout and TDNR both have on average larger confidence intervals than transductive dropout.

Impact on patients Given our motivations we also ground the performance of our method in how it could be used in real world decision making on the treatments offered to patients. There are many reasons treatment options may not be offered to patients including cost and potential side effects, as such there will usually be an associated risk level which a patient must be above in order to receive treatment. It’s thus very damaging to patients for a model to confidently predict them to be low risk when they are indeed not. In Table 6.1 we set a 15%

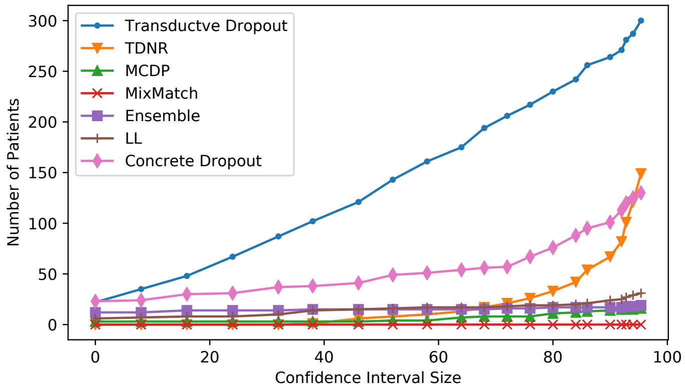

Fig. 6.1 **Improved patient outcome** We show how many more patients, for a risk threshold of 50%, correctly receive treatment as the size of the confidence interval on the prediction of risk changes.

threshold, and show how many more patient would receive treatment if we consider coverage of the 95% confidence interval on the patients risk, with the assumption that these cases can be handed off to a human expert who will correctly classify them. We see that transductive dropout results in a large increase in previously patients misclassified as low risk receiving treatment and we develop the impact on treatment options further in Figure 6.1. Here we set a treatment risk threshold at 50% and show how the size of any predicted confidence intervals over a patients risk impacts the increased number of patients correctly receiving treatment. Naturally for all methods as the confidence interval grows the number of now correctly treated patients increases but transductive dropout consistently outperforms the other benchmarks as it is less often confidently incorrect in its risk prediction.

How does the covariate-shift affect uncertainty? Of interest is to consider how the covariate shift has actually impacted our models performance. To that end we consider the feature distribution of those points misclassified by the model to see how it compares to both source and target sets. One of the most important factors affecting both the treatment and survival of prostate cancer patient is the age at diagnosis (Bechis et al., 2011). Studies have shown that older patients tend to have worse survival outlook and are more likely to receive surgery (Hall et al., 2005). In our source data, the average age at diagnosis is 66 years old moving up to 70 in the target set. Comparing to the distribution over ages for incorrectly predicted cases, where the average is 74, we see that it is for the patients who are considerably older than those usually seen in the training data that the model is less sure about. We see a similar story in their PSA scores (measurements of *prostate specific antigen* in the patients blood). PSA score is known to be a highly sensitive indicator for the risk level and severity of prostate cancer, and it is widely adopted in cancer screening and monitoring (Grimm et al., 2012). Again we see an increase in the average from 14.8 to 18.4 from source to target set but for those that are incorrectly classified the mean is much higher at 28.6. The percentage of patients receiving surgery in the incorrectly classified group is twice that of those correctly classified, suggesting that our models are least confident in areas which we might think are the most at risk given domain knowledge - the more elderly with high levels of PSA. The model struggles with them (is much less confident) though as they are values which don't have high density in the training data, demonstrating that blind application of a model to a covariate shifted data set may easily yield surprisingly incorrect predictions. Fortunately transductive dropout tends to return high uncertainty over its predictions on this covariate shifted data such that the practitioner can suitably inform any decisions to be taken as a result of these predictions.

6.2 Diagnosing Alzheimer's Disease with Variational Direct Policy Learning

Dataset We use data form the Alzheimer's Disease Neuroimaging Initiative (ADNI). Containing information taken every six months of 1737 patients who are suspected to be suffering from dementia, the dataset includes a variety of relevant coginitive tests and MRI scan results as well as biomarker information (Marinescu et al., 2018). At each visit every patient is diagnosed with one of normal cognitive function (NCF), mild cognitive impairment (MCI), or dementia. Cleaning the data by removing patients with missing scores and those whose follow-up visits occur significantly after the six-month normal period leaves us with 1626 patients with full information and trajectories of about three to four visits each.

Method We're interested in learning how the doctor behaves as they diagnose a given patient. The state space then consists of the three diagnoses, namely NCF, MCI, and dementia (which are recorded in order for us to match later but which we don't have access to for training). For the action space, we consider simply either ordering an MRI or not. Cognitive measurements are always taken regardless of whether an MRI is ordered or not and they can be categorised into one of three groups: "normal function" (for scores of 0), "questionable impairment" (for scores between 0.5 and 2.5), and "very mild to severe dementia" (for scores between 3.0 and 18.0) (O'Bryant et al., 2008). MRI outcomes are also categorised into one of four groups: "average", "above average", and "below average" hippocampus volume, as well as "not ordered". In total

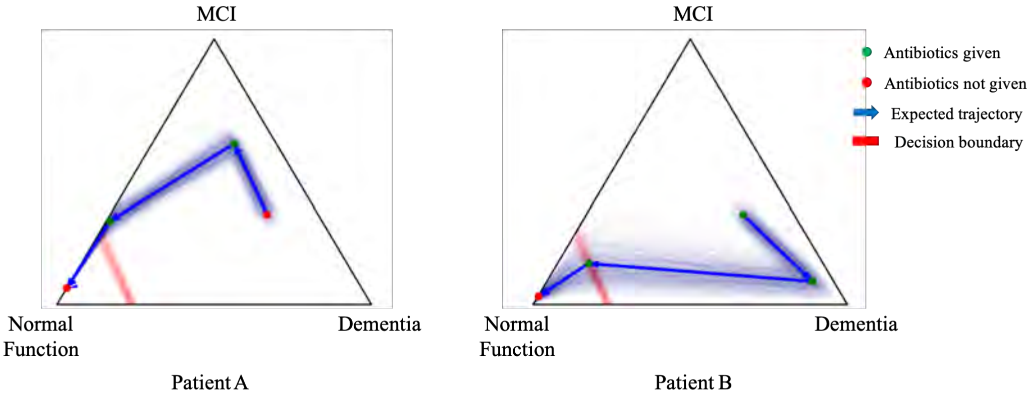

Fig. 6.2 Example patient trajectories. Uncertainty is represented as plots of 1000 Monte Carlo samples of the parameters indicating the posterior predictive density.

then we have $|A| = 12$ consisting of various cognitive measurement categories and the MRI outcomes.

In this setup we are only considering two actions so we fix $\eta = 1$ and simply allow the distance between mean-vectors to determine the stochasticity of the learnt policy. We then learn $\phi \backslash \{\alpha^{\eta}, \beta^{\eta}\}$ as in algorithm 1 though using Adam as our gradient update scheme (Kingma and Ba, 2015).

Example Patients In figure 6.2 we plot the belief trajectories of two example patients that have different uncertainty associated with them. Patient A follows a reasonably typical path, while they are not scanned initially on the first follow up they are: these results appear positive and so the belief moves reasonably towards the NCF. The doctor then performs another scan to be sure which again confirms that there are no issues and the doctor can confidently diagnose that there doesn't appear to be any issues.

The path of Patient B however is much more unusual. They receive an MRI scan immediately which indicates there may be issues and so the doctor's belief moves considerably towards dementia, this in itself is not an unusual thing to happen. What is unusual is that the second MRI actual reveals a completely normal scan which causes a very large change in belief towards NCF, which is then confirmed by a third scan. This is a very unusual case where the doctors sees a large change in belief from dementia toward NCF - usually there would not be conflicting evidence from consecutive scans and so we actually see a lot more uncertainty surrounding this transition reflected in figure 6.2, where the sampled trajectory paths actually vary quite considerably for Patient B, unlike for Patient A.

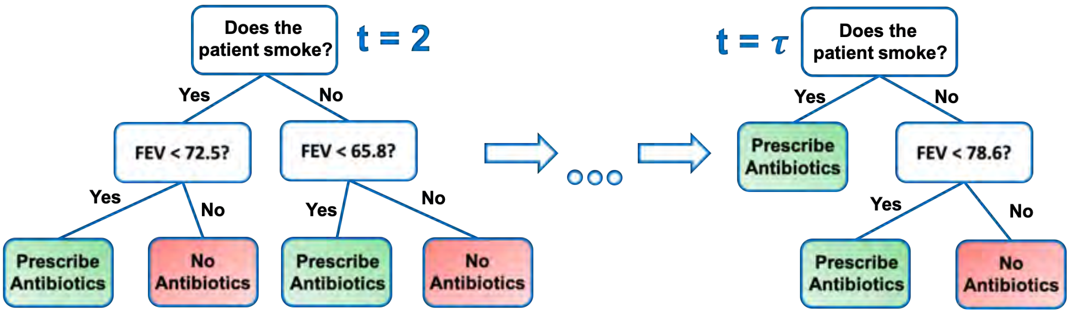

Fig. 6.3 **Observation policies.** An example of how the belief policy induces a different observation policy at different time steps based on how the beliefs have evolved. At the end of the trajectory the doctor is more concerned about their health and so the space of observations that would lead them to prescribe treatment is much larger. Note the start at $t = 2$ as it is the first action taken after an observation.

6.3 Understanding Treatment of Cystic Fibrosis with InterPoLe

Dataset We now explore modelling the decision making process of doctors treating patients with Cystic Fibrosis (CF) making use of data from the UK Cystic Fibrosis Registry, which is sponsored and hosted by the UK Cystic Fibrosis Trust. This consists of information on 10,995 patients during annual check-ups between 2008 and 2015 with covariates including demographics, genetic mutations, lung function scores, bacterial infections, and therapeutic interventions.

Setup Our goal is to model the decision making process of the doctors prescribing antibiotics to the patients. Bacterial lung infections are a common and serious complication for cystic fibrosis patients (Lyczak et al., 2002), though they can often be treated effectively by antibiotics it is important that they are spotted early. We split the state space into four, representing the underlying condition of the patient and consider a single action to be the decision to prescribe antibiotic or not on the part of the doctor. For the observations at each time point we consider the continuous valued forced expiratory volume (FEV1% Predicted) score of the patient as well as the binary indicator of the smoking status of the patient. As such we consider the observation function for every state-action pair to be parameterised by a set of means and variances for a Gaussian distribution over the FEV score and a probability for a Bernoulli distribution over the smoking status. InterPoLe is run on these trajectories until convergence using gradient ascent with step size 1×10^{-4}. An example of how the observation tree evolves is shown in figure 6.3.

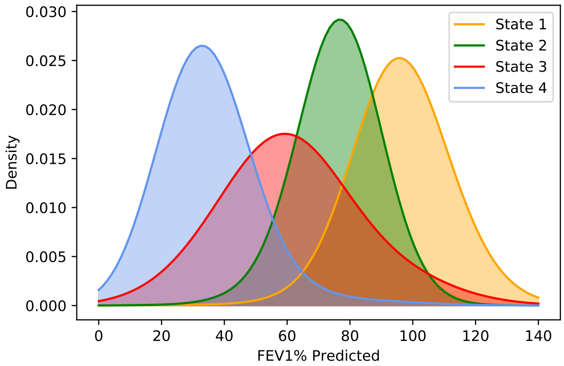

Fig. 6.4 FEV1% distributions over learnt states.

State	Average FEV1%	Infection Rate
1	96.2	0.643
2	76.2	0.805
3	61.7	0.827
4	35.2	0.903

Table 6.2 Average FEV1%'s and infection rates over learnt states.

Uncovering CF progression stages We see in this case that our model learns a representation of the state space that is closely tied to current guidelines on CF progression stages. The FEV1 biomarker is considered the main measure of illness severity in CF patients (Sanders et al., 2010), where the lower the score the more severe the illness, and is used to guide clinical and therapeutic decisions (Braun and Merlo, 2011). In figure 6.4, we plot density estimates for the FEV1 given the learned states and table 6.2 shows the accompanying mean value for those trajectories believed to be in that state alongside the true bacterial infection rate. The mean FEV1 values of the different states broadly align with cutoff values used in clinical guidelines on referring patients for lung-transplants (Braun and Merlo, 2011), allowing us to match them with current understandings of CF progression stages.

Highlighting impactful treatments and raising questions towards patient safety The nature of the decision dynamics allows us to further examine the effect of the actions taken, it is possible to see the resultant change in belief at every time step and so can pick out which actions at specific times have a higher impact on the change in belief. Unsurprisingly perhaps the resultant change in belief after prescribing antibiotics is on average 10% greater than the change when not, since we might expect the intervention to have an impact and perhaps decrease the severity of the illness when we next make some observation. On top of this though we can pick out the timesteps in each trajectory where we see surprisingly large changes in belief; in one example patient we see the belief over their state is very unsure over the course of four timesteps before they are finally prescribed antibiotics. This then results in a significant change in belief to state one (low severity) and we see an accompanying increase in their FEV1. Interestingly, InterPoLe would have predicted the patient be prescribed antibiotics earlier, at time step two. Indeed the decision not to prescribe then is predicted to have a probability of only about 20%, potentially flagging the action for further inspection. Thus it makes it possible to ask the question: "Our model suggests that under similar circumstances, normally you would prescribe a patient antibiotics if their FEV1 was below 65, yet this patient was on 62 and you did not; was this an intentional decision?". This kind of scrutiny is important to be able to ask to ensure that that patients remain safe and are treated correctly. We make it clear though that there should not be an assumption of wrongdoing - our model can learn the general process but there will often be uncaptured nuance to individual treatments, the important thing is to be able to feedback to doctors that this is what is expected and they should be able to explain why it is not necessarily appropriate.

Capturing personalisation of treatment Medicine is increasingly moving away from a one-size-fits-all approach to treatment (Graham, 2016) and by splitting the data along demographics and running InterPoLe separately, we can model apparent changes in the way different people are treated. We examined how male and female patients were treated differently considering that female patients generally face worse outcomes than male ones, especially with infections (Harness-Brumley et al., 2014) and find this reflected in the FEV1 cutoffs for antibiotic prescriptions at the initial timestep and belief. While it seemed smoking overruled the effect of gender where in both cases 78.8 is the cutoff, for non-smokers men would seemingly only be prescribed antibiotics if their FEV1 was below 73.3 while women would receive them below 76.3, suggesting a consequently higher willingness to prescribe to female patients given their increased risk.

Chapter 7

Conclusions

In this thesis we introduced three methods for providing more insight into policy learning for agents in offline sequential decision making, with an emphasis on the unique challenges faced by the healthcare setting.

Starting in chapter 3 we introduced Transductive Dropout, a method for using unlabelled data to calibrate the variance of Bayesian neural networks by introducing the auxiliary task of using the posterior predicted variance to discriminate between source and target distributions. We showed that this amounts to performing posterior regularisation in approximate Bayesian inference and results in more useful uncertainty predictions. We examined an instantiation of this framework within MCDP, transductive dropout, and in chapter 6 we demonstrate its applicability in the real task of predicting prostate cancer mortality, where it outperforms the tested benchmarks and demonstrates a higher level of appropriate uncertainty calibration. This can be usefully applied in the behavioural cloning setting where covariate shift is a serious issue due to the compounding error in action selection.

Then in chapter 4 we address the issue of uncertainty quantification in DIPOLE by extending the method of Huyuk et al. (2020) to learn a full approximate posterior over the parameters instead of just the MLE. This is done using a variational Bayesian EM algorithm that makes use of pathwise implicit derivatives to obtain gradients of the ELBO. In chapter 6 we demonstrate how this method can be used for understanding the diagnosis of Alzheimer's disease and explore a couple of example patient trajectories.

In chapter 5 we introduced a novel algorithm, InterPoLe, for learning interpretable representations of an agent's decision dynamics in an offline, partially observed environment by learning a noovel soft decision tree policy over beliefs and showing how this results in an evolving observation policy over time. In chapter 6 we demonstrated its applicability on real medical data and the modelling of prescribing antibiotics to patients with cystic fibrosis, allowing us to talk quantitatively about underlying themes, and uncovering known trends, of treatment.

We note that in chapter 6, we don't compare our methods of Variational DIPOLE or InterPoLe with other known baselines. The reason for this is simple - in the offline, partially observed setting with no rewards there simply aren't comparable methods for gaining insight into the decision making process. The only methods that can be reasonably applied are behavioural cloning options (like a neural network or standard decision tree) that just don't provide the kinds of insights that we were looking for. It is our hope that this, and our continued work in the area, will inspire more methods in the area that focus on an *understanding* of an agent and not just focused on beating the state-of-the-art on toy game problems.

References

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38.

Bain, M. and Sammut, C. (1995). A framework for behavioural cloning. In *Machine Intelligence 15*, pages 103–129.

Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. PhD thesis, UCL (University College London).

Bechis, S. K., Carroll, P. R., and Cooperberg, M. R. (2011). Impact of age at diagnosis on prostate cancer treatment and survival. *Journal of Clinical Oncology*, 29(2):235.

Bengio, Y. and Frasconi, P. (1995). An input output hmm architecture. In *Advances in neural information processing systems*, pages 427–434.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060.

Bewick, V., Cheek, L., and Ball, J. (2004). Statistics review 13: receiver operating characteristic curves. *Critical care*, 8(6):508.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Braun, A. T. and Merlo, C. A. (2011). Cystic fibrosis lung transplantation. *Current opinion in pulmonary medicine*, 17(6):467–472.

Brown, D. S. and Niekum, S. (2019). Deep bayesian reward learning from preferences. *arXiv preprint arXiv:1912.04472*.

Chan, A. J., Alaa, A. M., Qian, Z., and van der Schaar, M. (2020). Unlabelled data improves bayesian uncertainty calibration under covariate shift. In *Proceedings of the thirty-seventh International Conference on Machine Learning (ICML)*.

Choi, J. and Kim, K.-E. (2011a). Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research*, 12(Mar):691–730.

Choi, J. and Kim, K.-E. (2011b). Map inference for bayesian inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1989–1997.

Choi, J. and Kim, K.-E. (2012). Nonparametric bayesian inverse reinforcement learning for multiple reward functions. In *Advances in Neural Information Processing Systems*, pages 305–313.

Chou, R., Qaseem, A., Snow, V., Casey, D., Cross, J. T., Shekelle, P., and Owens, D. K. (2007). Diagnosis and treatment of low back pain: a joint clinical practice guideline from the american college of physicians and the american pain society. *Annals of internal medicine*, 147(7):478–491.

De Laat, P. B. (2018). Algorithmic decision-making based on machine learning from big data: Can transparency restore accountability? *Philosophy & Technology*, 31(4):525–541.

Englert, P., Paraschos, A., Deisenroth, M. P., and Peters, J. (2013). Probabilistic model-based imitation learning. *Adaptive Behavior*, 21(5):388–403.

Figurnov, M., Mohamed, S., and Mnih, A. (2018). Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pages 441–452.

Frosst, N. and Hinton, G. (2017). Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059.

Gal, Y., Hron, J., and Kendall, A. (2017). Concrete dropout. In *Advances in Neural Information Processing Systems*, pages 3581–3590.

Giusti, A., Guzzi, J., Cireşan, D. C., He, F.-L., Rodríguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., et al. (2015). A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667.

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.

Graham, E. (2016). Improving outcomes through personalised medicine. *NHS England. England.*

Grandvalet, Y. and Bengio, Y. (2005). Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536.

Graves, A. (2011). Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356.

Gray, R. M. (2011). *Entropy and information theory*. Springer Science & Business Media.

Grimm, P., Billiet, I., Bostwick, D., Dicker, A. P., Frank, S., Immerzeel, J., Keyes, M., Kupelian, P., Lee, W. R., Machtens, S., et al. (2012). Comparative analysis of prostate-specific antigen free survival outcomes for patients with low, intermediate and high risk prostate cancer treatment by radical therapy. results from the prostate cancer results study group. *BJU international*, 109:22–29.

Guo, H. and Gelfand, S. B. (1992). Classification trees with neural network feature extraction. *IEEE Transactions on Neural Networks*, 3(6):923–933.

Hall, W., Jani, A., Ryu, J., Narayan, S., and Vijayakumar, S. (2005). The impact of age and comorbidity on survival outcomes and treatment patterns in prostate cancer. *Prostate Cancer and Prostatic Diseases*, 8(1):22–30.

Harness-Brumley, C. L., Elliott, A. C., Rosenbluth, D. B., Raghavan, D., and Jain, R. (2014). Gender differences in outcomes of patients with cystic fibrosis. *Journal of women's health*, 23(12):1012–1020.

Heidenreich, A., Bellmunt, J., Bolla, M., Joniau, S., Mason, M., Matveev, V., Mottet, N., Schmid, H.-P., van der Kwast, T., Wiegel, T., et al. (2011). Eau guidelines on prostate cancer. part 1: screening, diagnosis, and treatment of clinically localised disease. *European urology*, 59(1):61–71.

Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hsing, A. W., Tsao, L., and Devesa, S. S. (2000). International trends and patterns of prostate cancer incidence and mortality. *International journal of cancer*, 85(1):60–67.

Huyuk, A., Jarrett, D., and van der Schaar, M. (2020). Explaining by imitating: Understanding decisions by direct policy learning. *Preprint*.

Irsoy, O., Yıldız, O. T., and Alpaydın, E. (2012). Soft decision trees. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1819–1822. IEEE.

Jean, N., Xie, S. M., and Ermon, S. (2018). Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. In *Advances in Neural Information Processing Systems*, pages 5322–5333.

John, G. H. (1996). Robust linear discriminant trees. In *Learning From Data*, pages 375–385. Springer.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214.

Karbalayghareh, A., Qian, X., and Dougherty, E. R. (2018). Optimal bayesian transfer learning. *IEEE Transactions on Signal Processing*, 66(14):3724–3739.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *ICLR*.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. In *Advances in neural information processing systems*, pages 2575–2583.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kouw, W. M. and Loog, M. (2019). A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413.

Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2.

Levine, S., Popovic, Z., and Koltun, V. (2011). Nonlinear inverse reinforcement learning with gaussian processes. In *Advances in Neural Information Processing Systems*, pages 19–27.

Li, Y., Song, J., and Ermon, S. (2017). Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*, pages 3812–3822.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23.

Louizos, C. and Welling, M. (2017). Multiplicative normalizing flows for variational bayesian neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2218–2227. JMLR. org.

Lyczak, J. B., Cannon, C. L., and Pier, G. B. (2002). Lung infections associated with cystic fibrosis. *Clinical microbiology reviews*, 15(2):194–222.

Makino, T. and Takeuchi, J. (2012). Apprenticeship learning for model parameters of partially observable environments. *arXiv preprint arXiv:1206.6484*.

Malhotra, N. K. (1982). Information load and consumer decision making. *Journal of consumer research*, 8(4):419–430.

Malinin, A. and Gales, M. (2018). Predictive uncertainty estimation via prior networks. In *Advances in Neural Information Processing Systems*, pages 7047–7058.

Marinescu, R. V., Oxtoby, N. P., Young, A. L., Bron, E. E., Toga, A. W., Winer, M. W., et al. (2018). Tadpole challenge: prediction of longitudinal evolution in Alzheimer's disease. *arXiv preprint arXiv:1805.03909*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

Ng, A. Y., Russell, S. J., et al. (2000). Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2.

O'Bryant, S. E., Waring, S. C., Cullum, C. M., Hall, J., Lacritz, L., Massman, P. J., Lupo, P. J., Reisch, J. S., and Doody, R. (2008). Staging dementia using Clinical Dementia Rating Scale Sum of Boxes scores: a Texas Alzheimer's research consortium study. *Arch. of Neurology*, 65(8):1091–1095.

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., and Peters, J. (2018). An algorithmic perspective on imitation learning. *arXiv preprint arXiv:1811.06711*.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*.

O'Sullivan, J. W., Heneghan, C., Perera, R., Oke, J., Aronson, J. K., Shine, B., and Goldacre, B. (2018). Variation in diagnostic test requests and outcomes: a preliminary metric for openpathology. net. *Scientific reports*, 8(1):1–6.

Park, J. and Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257.

Qaseem, A., Fihn, S. D., Dallas, P., Williams, S., Owens, D. K., and Shekelle, P. (2012). Management of stable ischemic heart disease: Summary of a clinical practice guideline from the american college of physicians/american college of cardiology foundation/american heart association/american association for thoracic surgery/preventive cardiovascular nurses association/society of thoracic surgeons. *Annals of Internal Medicine*, 157(10):735–743.

Quinn, M. and Babb, P. (2002). Patterns and trends in prostate cancer incidence, survival, prevalence and mortality. part i: international comparisons. *BJU international*, 90(2):162–173.

Rabiner, L. and Juang, B. (1986). An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16.

Raina, R., Ng, A. Y., and Koller, D. (2006). Constructing informative priors using transfer learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 713–720. ACM.

Ramachandran, D. and Amir, E. (2007). Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591.

Riquelme, C., Tucker, G., and Snoek, J. (2018). Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*.

Rohrbach, M., Ebert, S., and Schiele, B. (2013). Transfer learning in a transductive setting. In *Advances in neural information processing systems*, pages 46–54.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. (2016). Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171.

Sanders, D. B., Hoffman, L. R., Emerson, J., Gibson, R. L., Rosenfeld, M., Redding, G. J., and Goss, C. H. (2010). Return of fev1 after pulmonary exacerbation in children with cystic fibrosis. *Pediatric pulmonology*, 45(2):127–134.

SEER (2019). Surveillance, epidemiology, and end results (seer) program.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Singh, H., Spitzmueller, C., Petersen, N. J., Sawhney, M. K., and Sittig, D. F. (2013). Information overload and missed test results in electronic health record–based settings. *JAMA internal medicine*, 173(8):702–704.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Sugiyama, M. and Storkey, A. J. (2007). Mixture regression for covariate shift. In *Advances in Neural Information Processing Systems*, pages 1337–1344.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global.

Ude, A., Atkeson, C. G., and Riley, M. (2004). Programming full-body movements for humanoid robots by observation. *Robotics and autonomous systems*, 47(2-3):93–108.

UK, P. C. (2019). Prostate cancer uk.

Van Den Berg, J., Miller, S., Duckworth, D., Hu, H., Wan, A., Fu, X.-Y., Goldberg, K., and Abbeel, P. (2010). Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In *2010 IEEE International Conference on Robotics and Automation*, pages 2074–2081. IEEE.

Van Hasselt, H., Guez, A., and Silver, D. (2015). Deep reinforcement learning with double q-learning. *arXiv preprint arXiv:1509.06461*.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.

Zhu, J., Chen, N., and Xing, E. P. (2014). Bayesian inference with posterior regularization and applications to infinite latent svms. *The Journal of Machine Learning Research*, 15(1):1799–1847.

Zhu, X. and Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA.