CHALMERS
UNIVERSITY OF TECHNOLOGY

Medical decision support for treating infectious diseases

Master's thesis in Engineering Mathematics and Computational Science

Hugo Ölund

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020
www.chalmers.se

MASTER'S THESIS 2020

Medical decision support for treating infectious diseases

Hugo Ölund

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Medical decision support for treating infectious diseases
Hugo Ölund

© Hugo Ölund, 2020.

Supervisor: Marina Axelson-Fisk, Department of Mathematical Sciences
Examiner: Johan Jonasson, Department of Mathematical Sciences

Master's Thesis 2020
Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Gothenburg, Sweden 2020

Abstract

With an ever increasing digitalization of health care and records thereof, possibilities of applying data driven methods to analyze and predict healthcare outcomes are becoming more numerous. Markov decision processes (MDPs) are an interesting class of models to explore in this context. Originally born out of the operations research field, they allow us to model sequential decision making under uncertainty. Furthermore they may be extended to partially observable MDPs (POMDPs), in order to model further uncertainty, such as decision making with incomplete information which is often the case in healthcare. In this project, a literature review has been made in order to analyze and discuss mainly POMDPs from the perspective of treating sepsis and COVID-19.
Keywords: Partially observable Markov decision processes, Bayesian reinforcement learning

Contents

1 Introduction

Across all parts of society, there exists a growing interest in the potential of artificial intelligence (AI) to revolutionize key fields of technology. Within the field of healthcare and medical technology, several applications of AI have proven to be successful [27]. For example, a deep learning method based on a convolutional neural network has been proven to be comparable to human physicians in classifying melanomas [10]. However, a common drawback for deep learning methods is their opacity and lack of explainability.

In medical applications, there exists a need for physicians to be able to understand why a certain output is given by AI applications and/or support systems [15]. On a more general level, transparency of algorithms and AI systems that are increasingly being used in society is one of the key elements in building trustable AI as per a recent communication from the European Commision on that same matter[1].

1.1 Background

Sepsis is a condition associated with organ failure due to the immune system overreacting to an infection. It is generally a serious condition. When treating patients infected with sepsis, it is critical to reach a correct diagnosis and begin proper treatment as soon as possible [14]. Unfortunately, in its early stages, sepsis is difficult to differentiate from several other conditions due to its range of possible symptoms. Reaching a correct diagnosis as early as possible requires experienced clinicians and even then, reliability of current diagnostic procedures may falter.

COVID-19 is also a condition where a quick and correct diagnosis is important. The disease is contagious and complications may be severe, wherefore reliable decision support may be useful in this context.

This project is aimed at exploring viable mathematical models for AI-based decision support systems to be used in clinical environments. An important part of assessing the models will be their explainability and transparency, since it is important for physicians to know why and how decision support systems arrives at certain treatment suggestions.

1.2 Aim

The aim of this master thesis project is to explore and to some degree implement models to be used in an explainable decision support system for correctly diagnosing infectious diseases. Specifically, this will be investigated in the context of sepsis and COVID-19.

To this end, Markov decision processes (MDP) will be employed in order to form a robust and explainable mathematical framework in which to model treatment of patients. Recent research has suggested that they may be useful in supporting sequential decision making in sepsis treatment [18]. An MDP is a Markov process which incorporates a set of actions that influence the transitions between states, and a notion of scalar rewards structured to indicate an agent's performance in the modeled environment [1].

[1] https://ec.europa.eu/digital-single-market/en/news/communication-building-trust-human-centric-artificial-intelligence

As models, MDPs excel at describing sequential decision making in stochastic domains. Finding a solution to an MDP generally entails finding a policy that maximizes an agent's acquired cumulative reward. A policy is in general a decision rule which decides which action to perform in each given state, typically with regard to an action's expected utility. From an established optimal policy it is possible to extract a suggested action based on the current state and the expected utility of possible actions. Properly communicating the mathematical reasoning for suggesting certain actions would then lead to physicians being able to make ethically sound decisions based on these suggestions.

For a general model of the course of diseases, MDPs in their basic configuration are not sufficient due to the non-Markovian nature of clinical decision making. If the states are taken to be the patient's health status, any decision taken will naturally be dependent on earlier states and on subsequent decisions. Furthermore, it is impossible to know the complete state of a patient at any time. It is therefore necessary to introduce partially observable Markov decision processes (POMDP), where the state is no longer assumed to be fully knowable [16]. This also introduces the need to estimate the latent state of patients over the course of time with e.g. Bayesian methods. Another complicating matter is that measurements on a patient are rarely done at equally spaced intervals, meaning that the whole problem may be better modeled as a semi-Markov decision process [11]. The field of semi-Markov decision processes and specifically partially observable semi-Markov decision processes has not been widely studied and are outside the scope of this project.

In conclusion, a literature review has been performed to form a basis for models and implementations. Earlier similar implementations have been studied with regard to appropriate approximations and heuristics, available data has been assessed and implementations have been evaluated with regard to both their performance and explainability. The project has been carried out in collaboration with medical and software engineering experts in order to influence the end result towards a usable medical decision support system.

1.3 Limitations

First and foremost, the project is limited to developing and testing mathematical models to be used in the context of medical decision support. Only sepsis and COVID-19 have been included in any reasoning about the viability of models. Due to the COVID-19 pandemic, healthcare data that was supposed to be analyzed in the context of this project was delayed and therefore the focus of this report shifted from practical applications to theoretical considerations. There was also a shift of focus from sepsis to COVID-19 due to a greater availability of data.

1.4 Specification of issue under investigation

The project is aimed to result in possible mathematical models for supporting medical decision making in the context of treating sepsis and COVID-19. This has required a survey of literature pertaining to partially observable Markov decision processes. Since solution methods for POMDPs include e.g. reinforcement learning, there has existed a need for patient data to be analyzed. This has been done to some extent but there was no time for practical models built on this data.

In order to get a grasp of the models viability, some basic version limited in features will need to be implemented. A vital part of an assessed model's performance will be its ability to inform any user of why it recommends certain decisions. This might for example entail highlighting which factor(s) that contributes the most in e.g. the estimated probability of sepsis. A basic toy POMDP has been implemented and analyzed in order to form an understanding of POMDPs in general.

2 Theoretical primers

In this section we introduce a few necessary basic mathematical concepts and results which we will later build upon. The sections are in no particular order, and can be viewed as a reference for the more applied sections.

2.1 Markov chains and the Markov property

A discrete stochastic process is a sequence of random variables $\{X_t : t \in \mathscr{T}\}$ where $\mathscr{T}$ is a finite set of discrete indices. Let the set of values taken by the random variables be denoted $\mathscr{S}$ and let $|\mathscr{S}| = N$. With this broad definition, there exist many possible types of stochastic processes with different properties, see e.g. [12]. However, we will focus on a special subset of stochastic processes called Markov chains.

Definition 2.1.1. A stochastic process is a Markov chain if it satisfies

$$\mathbf{P}(X_t = j | X_1 = i_1, \ldots, X_{t-1} = i_{t-1}) = \mathbf{P}(X_t = x_j | X_{t-1} = i_{t-1})$$

for all $x_i \in \mathscr{S}$ [2].

This property means that only the latest state of the process contains information that affects the process' next state.

Definition 2.1.2. The transition matrix A of a Markov chain is a $|\mathscr{S}| \times |\mathscr{S}|$ matrix with entries

$$a_{ij} = \mathbf{P}(X_t = j | X_{t-1} = i)$$

for $i, j \in \mathscr{S}$ and where

- $a_{ij} \geq 0$ for all i, $j \in \mathscr{S}$ and
- $\sum_j p_{ij} = 1$ for all i.

That is, the element a_{ij} of transition matrix A gives the probability of the process moving from state i to state j. Therefore, all entries of the matrix must be nonnegative and its rows must sum to one.

Definition 2.1.3. A homogeneous Markov chain is a Markov chain that satisfies

$$\mathbf{P}(X_t = j | X_{t-1} = i) = \mathbf{P}(X_h = j | X_{h-1} = i)$$

where $i, j \in \mathscr{S}$ and $t \neq h$.

This means that the transition probabilities are constant with respect to time i.e. they are only dependent on states. Inhomogeneous Markov chains may better capture the dynamics of certain processes but they are more complex.

Definition 2.1.4. A distribution $\tau = \{\tau_i : \tau_i \in \mathscr{S}\}$ over state space $\mathscr{S}$ is stationary if

- $\tau_i \geq 0$ for all i, $\sum_i \tau_i = 1$ and
- $\tau = \tau A$, i.e. $\tau_j = \sum_i^n \tau_i a_{ij}$ for all $j \in \mathscr{S}$.

If we let $\mathbf{P}(X_0 = i) = \tau_i$, then $\mathbf{P}(X_t = i) = \tau_i$ for all $i \in \mathscr{S}$ [12].

Existence of a stationary distribution for Markov processes is important. If a process converges to a stationary distribution, we know that it behaves a certain way after a sufficient amount of time has passed.

The key takeaway from this short chapter on Markov chains is the Markov property and the behavior of these chains in the limit, i.e. that under certain assumptions they will behave according to some fixed distribution. We are interested in constructing models based on Markov processes where transition probabilities of the process not only depend on states, but also a set of actions available to a decision maker. For a more rigorous treatment on Markov chains, the interested reader may look to to other sources such as [2] or [12].

2.2 Hidden Markov models

A Markov model is generally suited for sequential data, where we cannot make the assumption that our samples are independent and identically distributed (i.i.d.). For example, the Markov chains detailed above may be used to model a random walk on some set of states. The next state of the chain will then only depend on the current state. We can train a model (i.e. estimate transition probabilities, initial distribution, etc.) from samples of such random walks, and then validate the fitted model with regard to predictive ability. Markov models may be generalized so that the Markov chain is obscured in such a way that we only receive partial observations of the chain's state. We call such a model a hidden Markov model (HMM) and a typical question we then want to answer is: given a sequence of observations, what is the most likely sequence of hidden states?

With $\{X_t : t \in \mathscr{T}\}$ denoting our Markov chain defined as above, we also consider the sequence $\{Y_t : t \in \mathscr{T}\}$ of observations. For the purpose of an example, let the observed process take values in some finite set of observations so that $Y_t \in \mathscr{V}$. We then assign probabilities to each observation via an emission distribution $p_j(Y_t|Y_1,\ldots,Y_{t-1}) = \mathbf{P}(Y_1,\ldots Y_{t-1}, X_t = j)$. From this we can construct a joint density over the observed process and the latent process as

$$\mathbf{P}(X_1,\ldots,X_T,Y_1,\ldots,Y_T|\pi,A,B) = \pi_{X_1} b_{X_1}(Y_1) \prod_{t=2}^{T} a_{X_{t-1},X_t} b_{X_t}(Y_t|Y_1,\ldots,Y_{t-1})$$

where π is the initial state distribution of the latent process, A its transition matrix and B the emission probabilities [2, 4].

2.3 Markov decision processes

In its most general form, the MDP is a very flexible tool for modeling sequential decision making under uncertainty. In this text the decision maker will be referred to as an agent, which in practice is represented by e.g. an algorithm. The MDP extends the Markov chain by adding a set of actions that are used by the agent to exert control on the transition probabilities.

In order to mathematically extract useful results from MDPs, a certain amount of stringency is required in their definition. First, we assume that the MDP transitions between states at discrete, unit intervals, i.e. that transitions occur at decision epochs $t \in \mathscr{D} = \{1, 2, \ldots\}$. It is possible to define both finite and infinite horizon MDPs. A finite horizon MDP would have $\mathscr{D} = \{1, 2, \ldots T\}$ where T is some finite natural number.

The states of an MDP are representations of the information an imagined agent is basing its decisions on. We denote the state of the process at time t with s_t, where $s_t \in \mathscr{S} = \{s_1, s_2, \ldots s_n\}$ is finite. A key aspect of MDPs is the observability of states. For the Markov property to hold, the state of the process must at all times be fully observable to the agent. This means that all information relevant to making an optimal decision at time t must be encapsulated in state s_t.

We define the action space of an MDP as the finite set $\mathscr{A} = \{a_1, a_2, \ldots a_k\}$. It consists of all the actions an agent may take to influence the process' transitions between states. Here we let a_t denote the action taken at time t, i.e. when the process is in state s_t.

As with a general Markov process, the system transitions between states stochastically. Comparably to a Markov chain's transition matrix, there exists a transition function which governs the probability of moving to state $s' \in \mathscr{S}$ from state $s \in \mathscr{S}$ when taking action $a \in \mathscr{A}$. Formally the function is defined as the mapping $T : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \longrightarrow [0, 1]$. We introduce the notion of reward as a function which maps transitions to some scalar value. The reward is to be considered as an indication whether certain transitions are beneficial in the modeled system. The reward can be used to shape how an agent acts in the model, e.g. penalizing some transitions and rewarding others. Similarly to the transition function the reward function is defined as $R : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \longrightarrow \mathscr{R}$.

Definition 2.3.1. We define a discrete time Markov decision process as a tuple $[\mathscr{S}, \mathscr{A}, T, R]$ whose constituent parts $\mathscr{S}, \mathscr{A}, T, R$ are defined as above [17, 26].

To model decision making we introduce policies. A policy is in general some rule for selecting an action depending on which state the process occupies. Typically this would mean that the policy is a function that maps states to actions.

Definition 2.3.2. A deterministic policy is a function $p : \mathscr{S} \longrightarrow \mathscr{A}$ such that for any state s, $p(s) = a$.

Definition 2.3.3. A stochastic policy is a function $p : \mathscr{S} \times \mathscr{A} \longrightarrow [0, 1]$ such that for any state s, $\sum_{a \in \mathscr{A}} p(s, a) = 1$.

Remark. It should be noted that in this report the mappings T, R, p are assumed to be stationary, i.e. independent of the number of steps the process has taken.

In order to construct policies that maximize cumulative rewards, we need means to evaluate the value of each state. The value for each state is based on the amount of reward that may be gained from following a certain policy, starting in a specific state $s \in \mathscr{S}$.

Definition 2.3.4. A stationary value function is a mapping $V : \mathscr{S} \longrightarrow \mathscr{R}$. Furthermore, we define $V^p(s)$ as the value of state s when following policy p at subsequent decision epochs. Given some measure of utility u, $V^p(s) = u\left(r_t^p, r_{t+1}^p, \ldots\right)$ where $r_t^p = R(s,a,s')$, i.e. the reward for transitioning between s and s' when the action is chosen with p at decision epoch t. Since the reward function is stationary, r_t^p is independent of t, and we use t for indexing only.

We note that the value of a certain state is defined by a measure of the utility to be gained from subsequent rewards when starting from that state. The utility function u is a mathematical representation of what subjective value is placed on gaining rewards.

Definition 2.3.5. The optimal solution of an MDP is a policy p^* such that $V^{p^*}(s) \geq V^p(s)$ for all states $s \in \mathscr{S}$ and policies p.

For infinite horizon DPs with stationary transition and reward functions, finite state and action spaces there exist results that guarantees the existence of an optimal policy, see e.g. [17]. The same holds true for finite horizon MDPs, albeit without the need for stationary policies, reward and transition functions.

To construct optimal policies we need the specify our utility functions in order to know what we are optimizing. With a so called discount factor $\gamma \in [0,1]$ we may define such a function as the expected discounted total reward from time t

$$u(r_t, r_{t+1}, \ldots) = \mathbf{E}\left[\sum_{t=t'}^{|\mathscr{D}|} \gamma^{t'-t} r_{t'}\right] = \mathbf{E}\left[\sum_{t'=0}^{|\mathscr{D}|-t} \gamma^{t'} r_{t'+t}\right] \tag{1}$$

where $\mathbf{E}$ denotes expected value [17, 16]. The discount factor γ will for $\gamma \in [0,1]$ make the agent prioritize rewards closer in time over those far away. Note that for an infinite horizon MDP, i.e. one with $|\mathscr{D}| = \infty$, we must further limit the discount factor so that $\gamma \in [0,1)$, or else the sum might not converge. On the other hand, $\gamma = 0$ is uninteresting since no reward will have an effect on the agent's perceived utility.

With our state value function defined, we may write it as

$$\begin{aligned} V^p(s) =& \mathbf{E}\left[\sum_{t'=0}^{|\mathscr{D}|-t} \gamma^{t'} r_{t'+t}\right] = \mathbf{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \ldots\right] \\ =& \mathbf{E}\left[r_t + \gamma V^p(s_{t+1})\right] = \sum_{s' \in \mathscr{S}} T(s,a,s')\left(R(s,a,s') + \gamma V^p(s')\right) \end{aligned} \tag{2}$$

which means that the value of state s is defined recursively over subsequent states, starting at any decision epoch t. Denoting the optimal value function as $V^{p^*} = V^*$, one can prove that

$$V^*(s) = \max_{a \in \mathscr{A}} \sum_{s' \in \mathscr{S}} T(s,a,s')\left(R(s,a,s') + \gamma V^*(s')\right)$$

and that V^* is stationary [17, 26]. Given that we know the optimal value function, we may for example apply a greedy policy

$$p^*(s) = \arg\max_{a \in \mathscr{A}} \sum_{s' \in S} T\left(s, a, s'\right)\left(R\left(s, a, s'\right) + \gamma V^*\left(s'\right)\right)$$

in order to select actions. With this policy we greedily select the action with the highest estimated return at each decision epoch.

Remark. Selecting actions greedily may of course not always be ideal, for example in some cases of online reinforcement learning (e.g. temporal difference learning), where an agent learns a representation of the value function via interaction with the environment [26]. Using the greedy policy before the value function estimation has converged to an optimum may result in suboptimal behaviour.

2.4 Partially observable Markov decision processes

The need for full observability of states limits what we may model with an MDP. Introducing uncertainty over which true state the process currently inhabits extends the set of phenomena that may be sufficiently described as MDPs. Instead of supposing full observability of the state, we now consider a process where at each decision epoch, we receive an (incomplete) observation of the agent's state.

Definition 2.4.1. We define a partially observable Markov decision process as a tuple $[\mathscr{S}, \mathscr{A}, T, R, \Omega, O]$ where

- $\mathscr{S}, \mathscr{A}, T, R$ are the components of an MDP,
- Ω is a finite set of possible observations of the underlying state and
- $O : \mathscr{S} \times \mathscr{A} \times \Omega$ is the observation function which gives conditional distributions $O(o|s, a)$ over possible observations given state s [16, 7].

Partial observability means that all information about the state needed for optimal decision making is no longer available. Consider for example an agent trying to find a path through a labyrinth with restricted vision. The partial observability leads to the agent being uncertain of exactly where it is in the labyrinth. Furthermore, if the process is to be Markovian, the agent cannot know where it has been because of the memorylessness property.

In order to construct policies for acting optimally under state uncertainty, the agent must instead estimate what state it is in since we cannot fully know it. Ideally we would like to use all information that could possibly be available, i.e. the observation of the current state together with a history of previous observations and transitions.

Definition 2.4.2. A belief state is a probability distribution $b \in \mathscr{B}$ over the possible states in $\mathscr{S}$ where $\mathscr{B}$ is the set of possible belief states.

We have that $b(s)$ gives the agent's subjective probability that the true state is s. The distribution b forms a sufficient statistic for the agent's history, i.e. it should contain all information that has been available to the agent [16]. It is therefore plausible to construct an MDP over belief states in a partially observable setting, where the optimal policy maps beliefs to actions. Crucially, this belief state is continuous, in contrast to the discrete state space, which is a complicating factor in solving belief MDPs.

2.5 The Bayesian paradigm

At the heart of many contemporary machine learning algorithms lies the field of Bayesian statistics. It is an approach to statistical inference, based on a fundamental result of probability theory called Bayes theorem. Consider two random variables a and b. Letting $\pi(a)$, $\pi(b)$ denote the respective distributions over those variables and $\pi(a,b)$ their joint distribution, we state the chain rule of probability as

$$\pi(a,b) = \pi(b|a)\pi(a) \tag{3}$$

where $\pi(b|a)$ is the conditional distribution of b given a. Noting that, by the definition of a joint probability, $\pi(a,b) = \pi(b,a)$ we have that

$$\pi(b|a)\pi(a) = \pi(a|b)\pi(b) \tag{4}$$

$$\pi(b|a) = \frac{\pi(a|b)\pi(b)}{\pi(a)} \tag{5}$$

which is known as Bayes theorem [4].

Typically when we speak about statistics, we speak about frequentist statistics. Conceptually it is based on the frequencies of events and outcomes in our samples, from which we try to infer knowledge about some unknown set of quantities. There exists another paradigm in mathematical statistics, called Bayesian statistics, which differs conceptually from frequentist statistics.

Bayesian inference contrasts that of frequentist inference in the view of unknown quantities. In frequentist inference we are typically interested in some unknown set of parameters which are taken to characterize some model of a phenomenon occuring in our world. For example, we may assume that human male heights follow a normal distribution with some mean μ and standard deviation σ. We would then collect a sample of heights and estimate the parameters via e.g. maximum likelihood estimation. If the fitted normal distribution is deemed to be close enough to the sample, and the sample is collected in the right way, we have a statistical model which may sufficiently explain what we have observed. The interpretation of this is that we have computed which set of parameter estimates has the highest likelihood of characterizing our assumed model, given the sample. To communicate the uncertainty about our findings, we could for example construct confidence intervals around the estimated parameters.

Now let's denote our set of parameters θ. From the Bayesian viewpoint, we would begin by assuming that θ itself is a random variable. This allows us to place a distribution of our prior belief over this variable, i.e. the prior distribution $\pi(\theta)$. The prior distribution may for example represent expert knowledge of a field or initial measurements. One may also use an uninformative prior, which says as little as possible about our prior belief. Suppose then that we gather a random sample x, and we want to weigh the sample together with our prior belief $\pi(\theta)$. We let $\pi(x|\theta)$ denote the likelihood of our sample given some fixed value of θ and $\pi(x)$ denote that likelihood marginalized on our parameter space. According to Bayes' theorem, we get

$$\pi(\theta|x) = \frac{\pi(x|\theta)\pi(\theta)}{\pi(x)} \tag{6}$$

where we define $\pi(\theta|x)$ as the posterior distribution of our parameters θ.

Now, all our uncertainty about the parameter set in question is encoded in this posterior distribution. Moving forward, we may also collect new data from the same population and use the posterior distribution as a prior. That is, we may take what we already know about the population and update our knowledge using new data. The goal is then to gain a more precise posterior distribution on which to perform inference. For example, we may place intervals on the distribution called credibility intervals, which differ somewhat in their interpretation from confidence intervals. Since we place a credibility interval directly on our posterior distribution, we may say that according to our current knowledge, the parameter lies in that certain interval with some probability.

In practice, Bayesian inference may become quite hard. In the simplest case one may rely on the concept of conjugacy between probability distributions belonging to the exponential family of distributions. This, however, will seldom be possible in realistic applications. A few problems that may occur when not using conjugate priors is e.g. intractable posterior distributions or likelihoods. The finer points of Bayesian inference are outside of the scope of this report and the interested reader is directed to more detailed works such as [4].

3 Markov decision processes in practice

There exist many phenomena which may be modeled as MDPs, either completely observable (we shall refer to them as COMDPs henceforth) or partially observable. Interestingly they capture sequences of events and actions that vary in scope and time horizon. The applications of MDPs in the literature also vary broadly. We will touch on a few general examples, and then focus on applications in the medical context.

3.1 General examples of MDPs

In a 2015 paper, using deep neural networks, Mnih et al. constructed a reinforcement learning agent that was able to play a set of video games with human-level aptitude [19]. At the lowest level, the gameplay is modeled as an MDP for which an optimal policy is constructed via reinforcement learning. Reinforcement learning is a subset of methods for solving MDPs which rely on interaction with their environment, and will be treated more in depth later on in this work. The agent was trained on frames from various Atari 2600 games. Actually, small sequences of frames were used in order for the agent to be able to infer e.g. the velocities of on-screen objects. Were it not for this detail, the problem would not have been possible to model as a COMDP since the agent would not be able to infer velocities etc. from only one frame. However, one may model it as a POMDP and treat it similarly, see e.g. [13].

3.2 MDP examples in a medical setting

An MDP has been used in modelling the course of treating patients with intracranial aneurysms. The article in question focused on modelling patients with an asymptomatic intracranial aneurysm, a condition where a blood vessel in the brain is deformed and bulges outward [24]. In the asymptomatic case the condition is generally not noticed without medical imaging of some sort, meaning that the patient does not notice it unless it progresses adversely. If the vessel bursts, it may have catastrophic consequences for the patient. On the other hand, treatment carries risk of severe complications.

The authors fit a transition model according to existing knowledge and then solve it. This generates an optimal policy which can be used as a support in decisionmaking. It is important to note that the policies obtained are only as good as the model itself, meaning that the model must be reasonable if the answers obtained are to have any meaning. Another interesting factor in this article is that the authors incorporate a way to take patients' risk aversion into account. Since the treatment carries risk, a relatively mild aneurysm with lower risk of progressing adversely may be left untreated with regard to a patient's life expectancy and the quality of years therein.

3.3 Solving MDPs

When considering COMDPs either over a finite or a infinite horizon, solutions generally exist [17]. For illustrative purposes we will walk through one of the fundamental solution methods for both COMDPs and POMDPs, value iteration.

3.3.1 Dynamic programming

In certain complex optimization or programming problems we can leverage recursive structures that enable us to break the problem down into smaller subproblems. A problem whose optimal solution may be constructed from its subproblems, who in turn overlap (i.e. their solutions are possibly used several times) may be confronted with a method called dynamic programming. The defining characteristic for dynamic programming is that each subproblem is computed once and that those solutions are used to solve the larger problem.

3.3.2 Value iteration

Value iteration is a fundamental solution technique for MDPs which builds upon iteratively computing the value function for all states and actions. Starting arbitrarily at some initialization V_0 over the state space, we compute the value function at iteration n as

$$V_n(s) = \max_{a \in \mathscr{A}} \sum_{s' \in \mathscr{S}} T(s,a,s') \left(R(s,a,s') + \gamma V_{n-1}(s') \right) \tag{7}$$

for all $s \in \mathscr{S}$. We continue to iterate until some condition has been met, e.g. $\|V_n - V_{n-1}\| < \varepsilon$ using an appropriate norm and some $\varepsilon > 0$.

The sequence of value functions computed as in eq. (7) is guaranteed to converge to the optimal value function V^* as n tends to infinity [26, 17]. Since all components of the MDP are known, we may easily compute the value function for each iteration. However, each iteration requires $\mathscr{O}\left(|\mathscr{S}||\mathscr{A}|^2\right)$ operations, and there is no bound on how many iterations are required for convergence [26]. In the infinite-horizon case, the number of iterations required for convergence is dictated by the discount factor γ, where a bigger γ means a longer horizon is considered.

In constructing this sequence of value functions we may leverage the problem's recursive structure and utilize dynamic programming to simplify computations. Since we compute the value for each state in every iteration, saving them and utilizing the pre-computed values in later iterations will make the problem easier to solve. A sketch of a value iteration algorithm for COMDPs can be seen in algorithm 1.

There exist of course many other methods of solving COMDPs which utilize other problem structures, approximations, etc. However, a full discussion of these lies outside the scope of this report. We mainly consider value iteration since this kind of point iteration is central also in basic POMDP solution methods.

Algorithm 1 Value iteration for COMDPs

```
Choose arbitrary initialization V_0(s) ∀s ∈ 𝒮
Let n = 0
while convergence criterion not met do
  for all s ∈ 𝒮 do
    Let n = n + 1
    V_n(s) = max_{a∈𝒜} ∑_{s'∈𝒮} T(s,a,s') (R(s,a,s') + γV_{n-1}(s'))
    compute convergence criterion
  end for
end while
```

3.4 Solving POMDPs

Moving on to the partially observable setting presents us with additional problems. We can no longer completely rely on our knowledge of the model since we are not presented with the true state. Instead we rely on keeping a belief state as an estimation of the most likely state given previous states and the current observation. This means that we now estimate the expected future returns given what state the agent believes it is occupying.

3.4.1 Updating belief

As previously defined, a belief state b is a probability distribution over the state space $\mathscr{S}$, meaning that $b(s)$ gives the estimated probability of occupying state s. After every transition between states, we use the current belief state b and corresponding observation o and action a to update the belief state with our new knowledge using Bayes theorem [16]. For some new updated belief b' over a new state s' we then have that

$$
\begin{aligned}
b'(s') =&\mathbf{P}(s'|o,a,b) = \frac{\mathbf{P}(s',o,a,b)}{\mathbf{P}(o,a,b)} \\
=&\frac{\mathbf{P}(o|s',a,b)\mathbf{P}(s'|a,b)\mathbf{P}(a,b)}{\mathbf{P}(o|a,b)\mathbf{P}(a,b)} \\
=&\frac{\mathbf{P}(o|s',a,b)\mathbf{P}(s'|a,b)}{\mathbf{P}(o|a,b)} \\
=&\frac{O(s',a,o)\sum_{s\in\mathscr{S}}\mathbf{P}(s'|a,b,s)\mathbf{P}(s|a,b)}{\mathbf{P}(o|a,b)} \\
=&\frac{O(s',a,o)\sum_{s\in\mathscr{S}}T(s',a,s)b(s)}{\mathbf{P}(o|a,b)}
\end{aligned}
\tag{8}
$$

with O,T defined as above. We consider $\mathbf{P}(o|a,b)$ as a factor that makes b' a proper distribution since it is independent of s'. We call eq. (8) the state estimation function with $\mathrm{SE} : \mathscr{B} \times \mathscr{A} \times \Omega \longrightarrow \mathscr{B}$.

3.4.2 Belief MDPs

In order to solve a POMDP, we want to find a policy capable of mapping belief states to actions which maximizes the cumulative reward. Taking the belief state as a sufficient statistic for estimating the true state means that all information we can hope to have is contained in our belief. We may then construct an MDP on our belief space, resulting in a so-called "belief MDPs". With the added caveat that the belief space is continuous, in order to encode the partial observability in a belief MDP we need a few new definitions.

Since the state is not accessible, we take the continuous belief space $\mathscr{B}$ as a replacement to the state space $\mathscr{S}$. We also cannot compute any transition probabilities between states so we instead define a new state transition function $\tau : \mathscr{B} \times \mathscr{A} \times \mathscr{B} \longrightarrow [0,1]$ as

$$\tau(b,a,b') = \mathbf{P}(b'|a,b) = \sum_{o\in\Omega} \mathbf{P}(b'|b,a,o)\mathbf{P}(o|a,b)$$

where

$$\mathbf{P}(b'|b,a,o) = \begin{cases} 1 & \text{if } \mathrm{SE}(b,a,o) = b' \\ 0 & \text{otherwise.} \end{cases}$$

Lastly we define a new reward function

$$\rho(b,a) = \sum_{s\in\mathscr{S}} b(s)R(s,a)$$

which now maps a belief and an action to a reward.

Remark. The original reward function $R(s,a,s')$ maps rewards according to not only from which state an action is taken, but also which state the action results in. Using a variant such as $R(s,a)$ rewards actions only with regard to which state they were taken in.

Definition 3.4.1. A belief MDP is a tuple $[\mathscr{B}, \mathscr{A}, \tau, \rho]$ with components defined as above [16].

We now have a nicer formulation of what it would mean to solve a POMDP, and we turn to extending value iteration to POMDPs.

3.4.3 Value iteration

The actions available and their value to an agent are typically closely related to the time horizon under consideration. We will consider finite horizon solutions since we can get an arbitrarily close approximation of the optimal infinite horizon value function using a long enough horizon [16].

A way to represent policies in POMDPs is to draw policy trees. We take the root node as an action, with arcs to its children representing different possible observations. Then each child node is another action with possible resulting observations. The depth of the tree then depends on the time horizon considered, and we call it a t-step policy tree. Denoting a policy tree p, we want to compute the expected value of executing it. In the trivial case, i.e. $t = 1$ we have that p represents a single action and its value when taken in some state s is $V^p(s) = R(s, a(p))$. We let $a(p)$ denote the action corresponding to the root node of policy tree p.

Similarly to the COMDP value function in eq. (2), we now turn to the expected future rewards. For a t-step policy tree p defined as above, we have

$$\begin{aligned} V^p(s) =& R(s,a(p)) + \gamma \mathbf{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \dots\right] \\ =& R(s,a(p)) + \gamma \sum_{s' \in \mathscr{S}} T(s,a(p),s') \sum_{o_i \in \Omega} O(o_i, s', a(p)) V^{o_i(p)}(s') \end{aligned} \tag{9}$$

where γ is a discount factor defined as above and $o_i(p)$ denotes the $t-1$ policy subtree of p corresponding to observation o_i. We can now begin to recognise a recursive structure also in these computations.

Given that we define policy trees as above, we then have a definition of the value function for states in POMDPs. Since the agent should not have access to the state, we define the value for a belief $b \in \mathscr{B}$ as

$$V^p(b) = \sum_{s \in \mathscr{S}} b(s) V^p(s). \tag{10}$$

If we write

$$\mathbf{V}^p = (V^p(s_1), \dots, V^p(s_n))$$

we have that

$$V^p(b) = b \cdot \mathbf{V}^p$$

since b is defined on $\mathscr{S}$. We then have that each value function given by some policy tree p depends linearly on belief b. The same is then true for the optimal value function

$$V^*(b) = \max_{p \in \mathscr{P}} b \cdot \mathbf{V}^p$$

where $\mathscr{P}$ is the set of possible policy trees. We note here that V^* is not only piecewise linear (PWL) and non-negative, it is also convex [16].

Theorem 3.4.1. *The optimal value function defined as above is convex.*

We can leverage the optimal value function's convexity by projecting each linear segment, corresponding to a value function and its respective policy tree, to the belief space. In this way we may divide the belief space into polyhedral regions corresponding to where certain value functions dominate all others. So an agent's optimal action in some region of belief space is given by the root node of the policy tree whose corresponding value function dominates in that region.

For any $\tilde{\mathscr{V}} \subseteq \mathscr{P}$ we may construct a unique minimal subset $\mathscr{V} \subseteq \tilde{\mathscr{V}}$ such that

$$V^*(b) = \max_{p \in \tilde{\mathscr{V}}} b \cdot \mathbf{V}^p = \max_{p' \in \mathscr{V}} b \cdot \mathbf{V}^{p'},$$

i.e. the maximal PWL value function generated by $\tilde{\mathscr{V}}$ is the same as the one generated by $\mathscr{V}$. A value function generated by the minimal unique set of policy trees is called parsimonious, and any tree belonging to this set is called useful.

A value function that is dominated by other value functions over the whole belief space contribute nothing to the solution of a POMDP. For a vector of state values $\mathbf{V}^p$, we then define a belief region where it dominates all other value functions as

$$\mathscr{R}(\mathbf{V}^p, \mathscr{V}) = \left\{ b \in \mathscr{B}, p \in \mathscr{V} | b \cdot \mathbf{V}^p > b \cdot \tilde{\mathbf{V}}^{p'} \quad \forall p' \neq p \right\}.$$

Finding points in this region, or concluding that it is empty, requires solving a linear program [16]. By using the information from those solutions, we may prune a set of policy trees so that those whose value functions are everywhere dominated can be discarded.

In order to compute the value function of a POMDP using value iteration, we want to compute a parsimonious representation of V_n from the previous iteration's result V_{n-1}, which is also parsimonious. A simple algorithm to achieve this is called *exhaustive enumeration*. Consider a set $\mathscr{V}$ of policy trees and vectors $\mathbf{V}^p$ for each $p \in \mathscr{V}$. Recall that each t-step policy tree has a number of subtrees equal to $|\Omega|$, i.e. the number of possible observations. Since we assume that we have the results of the previous iteration available, we take only the useful policy trees of the previous iteration as candidates for the current. If we let $\mathscr{V}_{t-1}$ be the set of useful $t-1$ step policy trees we will have a superset of $\mathscr{V}_t$, which we can prune down to only useful policy trees.

This outlined algorithm is one of the simplest possible algorithms for the task. There exist several examples of exact POMDP solving algorithms with better asymptotic complexity, see e.g. [16, 6, 5].

3.5 Reinforcement learning

In the context of COMDPs, reinforcement learning (RL) introduces methods that do not explicitly rely on complete knowledge of a complete MDP model. Instead we rely on sampling the MDP model in order to "explore" the model's dynamics, which means that interaction with the model is a key element in building policies. There exists extensive literature on solving COMDPs with RL, see e.g. [7, 26, 22]. We now turn to using RL in the partially observable case.

3.5.1 Bayes adaptive POMDPs

Requiring full knowledge of a model is in general a very harsh assumption when dealing with real world problems. A framework called the Bayes-adaptive POMDP (BAPOMDP) has been proposed as a way of bridging the gap between the extensive model knowledge needed in exact POMDP solution algorithms and the more loosely formulated assumptions of RL [20]. Basically we want to extend the POMDP model introduced in definition 2.4.1 so that we may attempt to solve POMDPs where the transition and observation probabilities are unknown *a priori*. In order to incorporate this into our models, we use Dirichlet distributions.

Definition 3.5.1. A Dirichlet distribution is a discrete multivariate probability distribution over a vector of K random variables μ_k restricted to $[0,1]$ with the added constraint that $\sum_{k=1}^{K} \mu_k = 1$. Given counts $(\phi_1, \dots, \phi_K)$ of events observed over n trials, we have that

$$\mathrm{Dir}(\mu_1, \dots, \mu_K | \phi_1, \dots, \phi_K) = \frac{\Gamma(\sum_k \phi_k)}{\prod_k \Gamma(\phi_k)} \prod_{k=1}^{K} \mu_k^{\phi_k - 1}. \tag{11}$$

The Dirichlet distribution gives the probability that the true underlying distribution of a random variable given discrete observations $(\phi_1, \dots, \phi_K)$ is $(\mu_1, \dots, \mu_K)$ [4].

The extension to the original POMDP formulation we make is to allow uncertainty over the transition and observation functions T, O. Taking the expectation of a μ_k we have that $\mathrm{E}(\mu_k) = \frac{\phi_k}{\sum_{k=1}^{K} \phi_k}$. We then keep track of transitions and observations, i.e. we let $\phi_{sas'}$ represent the number of times the process transitioned from state s to s' given action a, and similarly let $\psi_{as'o}$ be the number of times we observe o in state s' following action a. This gives us a possibility to let a Dirichlet distribution represent our uncertainty of T and O. Taking the expectation of eq. (11) we get that the expected probability of transition (s, a, s') is given by

$$T_\phi(s, a, s') = \frac{\phi_{sas'}}{\sum_{s' \in \mathscr{S}} \phi_{sas'}}$$

and similarly we get

$$O_\psi(s', a, o) = \frac{\psi_{s'ao}}{\sum_{o \in \Omega} \phi_{s'ao}}.$$

In order to fully represent this uncertainty we need to somehow incorporate the counts ϕ, ψ into the POMDP model. Letting

$$\mathscr{T} = \left\{ \phi \in \mathbb{N}^{|\mathscr{S}|^2 |\mathscr{A}|} \middle| \sum_{s' \in \mathscr{S}} \phi_{sas'} > 0 \quad \forall (s,a) \in \mathscr{A} \right\}$$

and

$$\mathscr{O} = \left\{ \psi \in \mathbb{N}^{|\mathscr{S}||\mathscr{A}||\Omega|} \middle| \sum_{o \in \Omega} \psi_{sao} > 0 \quad \forall (s,a) \in \mathscr{A} \right\}$$

represent the count spaces, we define a new state space $\mathscr{S}' = \mathscr{S} \times \mathscr{T} \times \mathscr{O}$ for the BAPOMDP. Since the transition and observation counts are now part of the state space, we need to take them into account in the transition and observation functions. Define $\delta_{sas'}$ as a vector that is everywhere zero except at the index corresponding to transition (s,a,s') where it is set to one. Then we can define an increment of the transition counts as $\phi' = \phi + \delta_{sas'}$. Similarly we define $\delta_{s'ao}$ for the observation counts so that $\psi' = \psi + \delta_{s'ao}$.

Given the counts ϕ and ψ, we have distributions for the current possible models. To properly reflect this we redefine the transition and observation functions so that the counts are taken into account. Let

$$T'((s,\phi,\psi),a,(s',\phi',\psi')) = \begin{cases} T_\phi(s,a,s')O_\psi(s',a,o) & \text{for } \phi' = \phi + \delta_{sas'}, \psi' = \psi + \delta_{s'ao} \\ 0 & \text{otherwise} \end{cases}$$

and correspondingly

$$O'((s,\phi,\psi),a,(s',\phi',\psi'),o) = \begin{cases} 1 & \text{for } \phi' = \phi + \delta_{sas'}, \psi' = \psi + \delta_{s'ao} \\ 0 & \text{otherwise.} \end{cases}$$

In this construction, the received observation is determined by the transition function and the observation function is deterministic.

Definition 3.5.2. We define a Bayes adaptive POMDP as a tuple $[\mathscr{S}',\mathscr{A},T',R,\Omega,O']$ where

- $\mathscr{A}$, R and Ω are defined as in a POMDP and
- $\mathscr{S}' = \mathscr{S} \times \mathscr{T} \times \mathscr{O}$,
- $T'((s,\phi,\psi),a,(s',\phi',\psi')) = \begin{cases} T_\phi(s,a,s')O_\psi(s',a,o) & \text{for } \phi' = \phi + \delta_{sas'}, \psi' = \psi + \delta_{s'ao} \\ 0 & \text{otherwise} \end{cases}$
  and
- $O'((s,\phi,\psi),a,(s',\phi',\psi'),o) = \begin{cases} 1 & \text{for } \phi' = \phi + \delta_{sas'}, \psi' = \psi + \delta_{s'ao} \\ 0 & \text{otherwise.} \end{cases}$

Similarly to the POMDP model, the agent is required to keep a belief state in order to represent the information collected about the model. With the redefined state space $\mathscr{S}'$ it is now also necessary to include the transition and observation counts, so that the belief is a distribution over states and counts. This means that in solving the BAPOMDP we take into account not only the uncertainty over which state the agent is in, but also our uncertainty over the true model dynamics.

The BAPOMDP model can be regarded as a POMDP with a countably infinite state space and hence the optimal value function and belief updates can be defined as in the regular POMDP. However, maintaining an exact belief space will rapidly grow intractable if an infinite number of states have non-zero probability of occuring. There exist approximations that can be leveraged, such that the infinite model itself may be approximated by finite ones and belief can be updated by Monte Carlo sampling [20].

Furthermore, in practices such as value iteration, where we perform a global search over the belief space in order to construct a value function, it may be prudent to visit so-called online methods. The key difference between online and offline planning is that in the online case interaction between the agent and the environment is the point of focus. Here the estimates of the model and state values are updated after each decision epoch where the agent has transitioned according to some policy [20, 26].

3.5.2 Infinite POMDPs

It has been shown that HMMs can be extended to infinite state spaces [3, 23]. The infinite HMM can in turn be extended with rewards and actions, thus forming the infinite POMDP [8]. In theory, this gives us the possibility of building a POMDP model with a countably infinite state space. The key difference from BAPOMDPs is that we need not make any assumption on the size of the state space, this will be inferred by the model itself. Thus, by leveraging Bayesian nonparametric methods, we may hope to find more efficient representations of large POMDPs.

A full account of the theory behind the infinite HMM and POMDP is outside the scope for this report. They build upon taking a building a hierarchical model of all possible unbounded transition and observation models and incorporating actions and rewards. Exact belief tracking becomes intractable over infinite possible models, so approximations in the form of Markov chain Monte Carlo sampling schemes are needed. Planning (i.e. evaluating state and action values) is done via a tree search in the space of estimated models. The infinite POMDP has been shown to learn and enact useful policies in relatively complex environments [9].

4 Interpreting a solved POMDP

In order to illustrate some of the theoretical concepts introduced in this report, we will solve and comment on a toy problem. We will use R, for which an excellent package[2] is available via CRAN. Included in the package comes a toy problem called the tiger problem which we will solve. In order to speed up computations, we will use an algorithm called incremental pruning. This algorithm has been shown to be significantly more efficient with regard to complexity than exhaustive enumeration while still producing exact solutions [5].

4.1 The tiger problem

In this problem an agent is faced with two doors. Behind one of them, a tiger lies in wait and behind the other a small treasure resides. The objective is to choose the correct door, i.e. the one hiding a treasure, based on incomplete information. In each decision epoch, the agent has three actions available: open the left door, open the right door or listen for the tiger. Choosing to listen will not necessarily give the correct observation, it may produce a false negative.

Formally we then have a state space $\mathscr{S} = \{\text{tiger left}, \text{tiger right}\}$ indicating behind which door the tiger is situated. The action space

$$\mathscr{A} = \{\text{open left}, \text{open right}, \text{listen}\}$$

[2] https://CRAN.R-project.org/package=pomdp

gives the possible actions and observation space $\Omega = \{\text{tiger left}, \text{tiger right}\}$ gives observations corresponding to where the agent perceives the tiger to be. The reward function is defined as

$$R(s,a) = \begin{cases} 10 & \text{if e.g. } a = \text{"open left" when } s = \text{"tiger right"}, \\ -100 & \text{if e.g. } a = \text{"open left" when } s = \text{"tiger left"}, \\ -1 & \text{if } a = \text{"listen"} \end{cases} \tag{12}$$

meaning that finding the tiger is severely punished in comparison to finding a treasure. The listen action carries a small penalty which should encourage the agent not to overuse it. Even in this small toy example, there are many factors that contribute to finding an optimal course of action, which illustrates the overall complexity of solving POMDPs exactly.

It remains to define a transition function and an observation function. Regardless of the current state, the listen action will not cause the state to change. Opening a door resets the problem by causing the tiger to either move to the other door or stay put with probability $1/2$. The observation probabilities are defined in a similar fashion. Listening gives a correct observation with probability $3/4$, i.e.

$$O(s' = \text{"tiger left"}, a = \text{listen}, o = \text{"tiger left"}) = 0.75.$$

After opening any door and resetting the problem, the agent receives the correct observation with probability $1/2$ so that it has no preferential information at the start of the problem [16].

We solve the problem in the infinite horizon case with two different discount factors, $\gamma_1 = 0.95$ and $\gamma_2 = 0.15$. We can at once see that a larger discount factor, i.e. one that technically discounts future rewards less, lets the agent adopt a more cautious approach. Comparing fig. 1 and fig. 2, we see that for $\gamma = 0.95$ a larger region of the belief space is dominated by value functions corresponding to the agent listening. The agent is encouraged to gather information until it is relatively certain of where the tiger resides. The reason for this is that long term rewards are not heavily discounted in the constructed value function, meaning that it is not very detrimental for the agent's expected reward to wait.

Conversely, having a small discount factor makes the agent more willing to take risks. It will open a door even though it is less certain of it hiding the treasure rather than the tiger, as can be seen in fig. 2. It is obvious then that the smaller discount factor makes the agent prioritize short term rewards.

Here it is appropriate to also comment on the belief space of this problem. We have two possible states and thus each belief state is a vector containing two probabilities, e.g. $(0.5, 0.5)$ if the agent estimates both states to be equiprobable. We can see that with two possible states, the belief space can be represented by a 1-simplex. This pattern repeats in problems with higher dimensional state spaces; a three-state problem's belief space can be represented by a triangle and so forth. Kaelbling et al. notes an interesting property when comparing the value function to its support in the belief space. Belief states with low entropy (i.e. belief states where the estimated probability of "being surprised" is lower) are valued higher since the agent has a higher expectation of acting usefully [16].

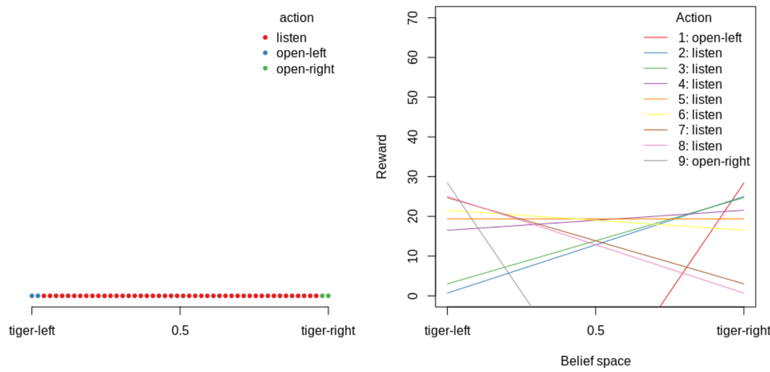

Figure 1: Left: The support regions of the dominating value function are colored corresponding to the optimal action to be taken from there with regard to an infinite horizon. Right: The dominating value functions corresponding to useful policy trees. Discount factor $\gamma = 0.95$.

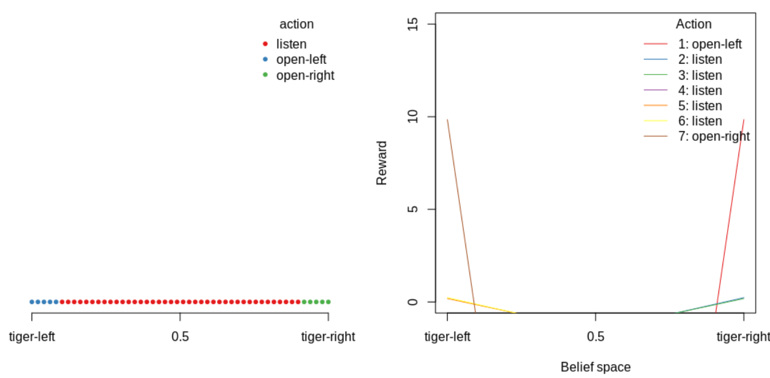

Figure 2: Left: The support regions of the dominating value function are colored corresponding to the optimal action to be taken from there with regard to an infinite horizon. Right: The dominating value functions corresponding to useful policy trees. Discount factor $\gamma = 0.15$.

5 Problem description

From the outset this report has been aimed at exploring Markov decision processes in a medical setting and specifically partially observable Markov processes in relation to diagnosing infectious diseases. Conceptually, a POMDP could very well be a suitable model in trying to represent situations faced by physicians in medicine. Typically we would be faced with observations emitted from a patient, in the form of measurements such as e.g. blood pressure and breathing frequency. Another possibility is that we want to take into account a patient's medical history and other relevant factors.

It is reasonable to assume that the complete state of a patient is not completely known, however. It could be prudent to place some uncertainty on what it is that actually gives rise to any observed measurements. The setting itself, i.e. observing and diagnosing patients under uncertainty could, from a mathematical perspective, typically involve analyzing longitudinal data with assumed latent variables. A good candidate for a model would in this case be a hidden Markov model or some variation thereof. What a POMDP allows us to do is to incorporate actions and rewards into an HMM. Firstly, the actions allow us to assume that some agent acting in the modelled environment will attempt to steer the environment's transitions in some direction. Secondly, constructing some reward structure in the environment allows us to place costs and rewards on various transitions, meaning that some sequences of transitions are preferred over others.

In this project, two distinct infections have been approached from the perspective of using MDPs in order to aid physicians with decision making. We will now turn to discussing them from the perspective of modelling with POMDPs.

5.1 Sepsis

According to a recent publication, sepsis "...should be defined as life-threatening organ dysfunction caused by a dysregulated host response to infection" [21]. It is a serious condition and urgent care is crucial. However, in early stages of the condition it is hard to identify since patients do not necessarily exhibit serious symptoms initially.

Ideally we would like to ascertain whether a patient has a risk of developing severe sepsis as soon as possible. On the other hand, it is not possible to catch all cases. Testing takes time and resources and in some cases it may suffice to send a patient home under monitoring by e.g. family members. Several frameworks exist which are aimed at helping classification of patients with a greater risk of developing severe sepsis. For example they take into account various measurable parameters in a patient, in order to estimate the seriousness of their condition [25].

These frameworks are not perfect, and the complex nature of the condition further exacerbates the difficulty of efficient classification. However, if we can sufficiently model the course of patients afflicted with sepsis using POMDPs, the solutions may give us insight into how these uncertainties can be navigated. An interesting avenue of exploration would be fitting a model to existing data where distinct actions are viable. Adding rewards is then a matter of setting some subjective preference to different transitions or end-states. In a sense we get a way to put a price on different outcomes. As an example, we can juxtapose the benefits of treating all suspected cases of sepsis with the adverse effects of unnecessary treatment using the reward functions.

5.2 COVID-19

Due to the global pandemic of spring 2020, part of this project has been focused on investigating medical decision support also in the context of SARS-COV2. The infection caused by SARS-COV2, COVID-19, carries a risk for severe complications, especially for certain patient groups. From a mathematical perspective it can be said to be quite similar to sepsis, since symptoms are generally quite vague before the infection takes a serious turn. It does however carry with it its own set of complicating factors, e.g. the risk of the virus being transmitted to new hosts which is not a problem in the sepsis case.

From this perspective it should be relevant to explore POMDPs also in this context. Much like with sepsis, we can explore different treatment strategies from a mathematical perspective while taking into account our uncertainty.

5.2.1 Electronic health record analysis

As part of this project an opportunity was given to study actual health records related to patients with suspected or confirmed COVID-19 in western Sweden. Unfortunately, this work was started quite late in the process and thus no practical application of POMDP modelling on this data has been made. The dataset is in many ways promising in this regard and should be explored further.

A promising factor is the generality of electronic health records. It is conceivable that, if we receive data in such a form and manage to build a meaningful POMDP from it, similar applications could be constructed with other medical conditions in mind. There exists also a possibility of applying methods such as Bayes adaptive POMDPs or infinite POMDPs where we can learn to plan and act in environments where all model parameters are not necessarily known [9, 20].

6 Discussion

In this project Markov decision processes have been explored from a perspective of using them in the context of medical decision support. Much like in general Markov modelling, we may capture sequential amd longitudinal phenomena but we can also extend the models to modelling the agency of some decision maker. If we further extend MDPs to be partially observable we have a much broader class of models, in similar fashion to hidden Markov models. Further extensions to the POMDP model exist where we are not required to specify all parameters of a model, instead learning it through a Bayesian treatment.

There exist examples of MDPs and POMDPs being applied to real medical data. The typical setting is some known environment where reasonable Markov models may be fitted from available data, and then be extended with actions and rewards. Solving such MDPs may provide insight into how we may navigate uncertainty over risks we can put a price on. Given that we come close enough to reality we may also e.g.

evaluate and simulate strategies that are in place.

The possibility of solving POMDPs exactly is interesting from the perspective of transparency. Given a complete model with values associated with certain states and transitions defined through reward structures, we may extract an optimal course of action in a mathematical sense. Solving POMDPs exactly generally does not scale well with model size however. On the other hand we want to build as rich models as possible, meaning that approximative methods most likely need to come into play. Approximations exist both for e.g. estimating value functions and estimating the most likely model itself when it is only partly known.

Adopting a Bayesian mindset when tackling POMDPs allows us to quantify our uncertainty, both over belief state and what the true model is. With this one should expect the same difficulties, but also benefits, that come with Bayesian inference in general. Results may vary with different sampling schemes, e.g. performing belief updates in BAPOMDPs [20]. On the other hand we will have access to clear decision rules given the inherent uncertainty.

6.1 Exact solutions and their implications

Finding exact solutions to POMDPs requires a nice, finite model. These models may be fitted from data, much like a standard Markov chain. However, we are limited to using a model that we can fully specify *a priori*. Furthermore, solving POMDPs exactly is hard from a computational perspective. Even under simple circumstances with a finite horizon, solving a POMDP exactly is a PSPACE-complete problem, and therefore harder than solving NP-complete problems [6].

Given, however, that we can extract the necessary components of a POMDP (states, actions, transition probabilities, observation probabilities) from data we get transparent models. As we have seen, we may also extract clear decision rules and evaluations of different beliefs from the solutions. A slight drawback is that if we move from problems with more states than three, we lose the geometric interpretability of the belief space and value functions. This should of course not be seen as a major hindrance in any attempt at modelling with POMDPs. Let's use a simple example as an illustration.

Example 6.1. We have a dataset containing patient records from which we have inferred a space of aggregated states

$$\mathscr{S} = \{\text{"no COVID-19", "mild COVID-19", "severe COVID-19"}\}$$

and a similar observation space

$$\Omega = \{\text{"no symptoms", "mild symptoms", "severe symptoms"}\}.$$

We deduce that the possible actions are given by

$$\mathscr{A} = \{\text{"test", "do not treat", "treat"}\}$$

where "test" means testing for what the patient's true condition is, "do not treat" could signify sending the patient home and "treat" means hospitalizing the patient. We will then add transition and observation probabilities. Let's say for example that both the

”mild COVID-19” and ”no COVID-19” states have a probability of giving observation ”mild symptoms”. We then have an interesting model where we can add penalties for e.g. choosing to test a patient that does not have COVID-19, thus capturing some of the difficulties that may arise in treating infectious diseases.

Solving the model in example 6.1 would result in us being able to extract a value function and a policy for whichever time horizon we choose. There is of course the question then if the model itself is sufficient for giving us any new insight into the problem. One could imagine a richer model, but it would come at the cost of increased computational complexity.

6.2 Reinforcement learning as a way forward

In this context, introducing reinforcement learning means that we can allow agents to have uncertainty over the transition and observation dynamics of the model. In the process of solving the problem, the agent learns a representation of the model. Consider for example investigating the tiger problem without having access to the transition and observation probabilities. The reinforcement learning frameworks introduced in this report are aimed at navigating that kind of difficulties. They also offer a possibility of more efficient representations of large POMDPs at the cost of introducing uncertainty. This is not necessarily a drawback, the level of uncertainty over different models may of itself be an interesting factor to explore.

The BAPOMDP and infinite POMDP frameworks introduced in this report may be an interesting area to investigate further. It is clear that research has been conducted into ways of representing and planning in POMDPs under model uncertainty. This approach may provide an alternative which does not rely on models whose parameters are completely known beforehand.

6.3 Conclusion

Markov decision processes and their extensions, partially observable Markov decision processes, may be interesting models to explore in the context of medical decision support systems. They seem able to capture longitudinal events in an interesting way where we can explore and evaluate different strategies. It would be crucial to also further explore how to build meaningful POMDP models from data with the help of medical experts. Given this we may construct rich, explainable models of a wide range of scenarios.

References

[1] Oguzhan Alagoz, Heather Hsu, Andrew J Schaefer, and Mark S Roberts. Markov decision processes: a tool for sequential decision making under uncertainty. *Medical Decision Making*, 30(4):474–483, 2010.

[2] Marina Axelson-Fisk. *Comparative Gene Finding: Models, Algorithms and Implementation*, volume 20. Springer, 2015.

[3] Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. The infinite hidden markov model. In *Advances in neural information processing systems*, pages 577–584, 2002.

[4] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[5] Anthony R Cassandra, Michael L Littman, and Nevin Lianwen Zhang. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. *arXiv preprint arXiv:1302.1525*, 2013.

[6] Anthony Rocco Cassandra. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Brown University, 1998.

[7] Christos Dimitrakakis and Ronald Ortner. Decision making under uncertainty and reinforcement learning, 2018.

[8] Finale Doshi-Velez. The infinite partially observable markov decision process. In *Advances in neural information processing systems*, pages 477–485, 2009.

[9] Finale Doshi-Velez, David Pfau, Frank Wood, and Nicholas Roy. Bayesian nonparametric methods for partially-observable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):394–407, 2013.

[10] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

[11] Franciszek Grabski. 15 - semi-markov decision processes. In Franciszek Grabski, editor, *Semi-Markov Processes: Applications in System Reliability and Maintenance*, pages 229 – 244. Elsevier, 2015.

[12] Geoffrey Grimmett, Geoffrey R Grimmett, David Stirzaker, et al. *Probability and random processes*. Oxford university press, 2001.

[13] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*, 2015.

[14] Tove Helldin, Anna-Karin Pernestig, and Diana Tilevik. Towards a clinical support system for the early diagnosis of sepsis. In *International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management*, pages 23–35. Springer, 2017.

[15] Andreas Holzinger, Chris Biemann, Constantinos S Pattichis, and Douglas B Kell. What do we need to build explainable ai systems for the medical domain? *arXiv preprint arXiv:1712.09923*, 2017.

[16] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[17] Andrey Kolobov. Planning with markov decision processes: An ai perspective. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–210, 2012.

[18] Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11):1716–1720, 2018.

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[20] Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-adaptive pomdps. In *Advances in neural information processing systems*, pages 1225–1232, 2008.

[21] Mervyn Singer, Clifford S Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R Bernard, Jean-Daniel Chiche, Craig M Coopersmith, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8):801–810, 2016.

[22] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[23] Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392, 2005.

[24] Vera Tilson and David A Tilson. Use of a markov decision process model for treatment selection in an asymptomatic disease with consideration of risk sensitivity. *Socio-Economic Planning Sciences*, 47(3):172–182, 2013.

[25] J-L Vincent, Rui Moreno, Jukka Takala, Sheila Willatts, Arnaldo De Mendonça, Hajo Bruining, CK Reinhart, PeterM Suter, and Lambertius G Thijs. The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure, 1996.

[26] Marco Wiering and Martijn Van Otterlo. Reinforcement learning. *Adaptation, learning, and optimization*, 12:3, 2012.

[27] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. Artificial intelligence in healthcare. *Nature biomedical engineering*, 2(10):719, 2018.