Review of Papers applying Reinforcement Learning in Medical Domain

Subhojyoti Mukherjee*
College of Information and Computer Science
University of Massachusetts Amherst
Massachusetts, MA 01003
http://bio-nlp.org/index.php/people

Abstract

1 Introduction

Machine Learning algorithms have come to dominate several applications in our day-to-day life like recommender systems, managing industrial workforce, game-playing, etc. Simultaneously, in the health-care domain as well the machine learning algorithms have found increasing applications in recommending and improving treatment policies for ailing patients. Among the various available approaches within the machine learning framework we specifically discuss about Reinforcement learning (RL) in this report. RL is a sub-field of Machine Learning which has many useful applications in medical domain but simultaneously also faces many challenges as we will discuss in this report.

2 Motivational Examples

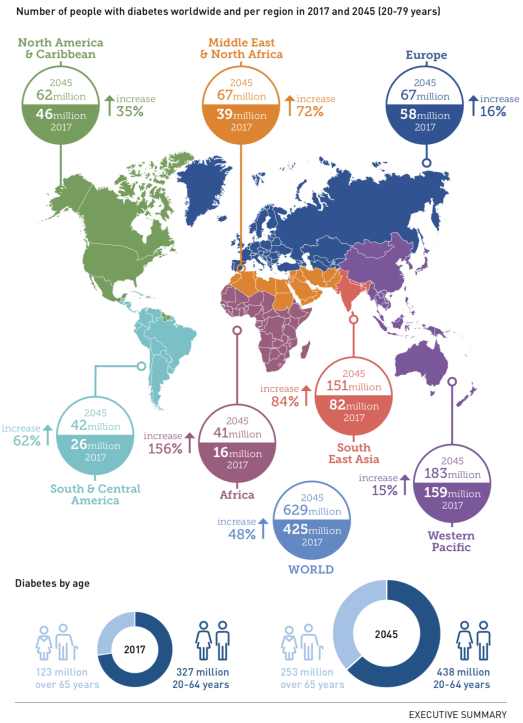
We will illustrate the use of RL in medical scenarios with a few motivational examples. These will be recurring examples throughout the report. We summarize these test-cases below.

2.1 Diabetes

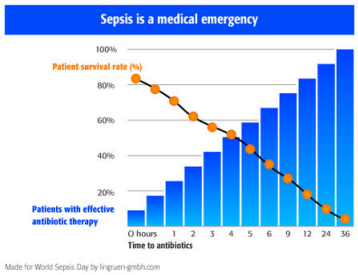
The first motivating example we state is that of treating diabetes. Diabetes is a disease that causes a high blood glucose level in patients. Currently there is no evident cure for this disease (Holt et al., 2011). There are two major sub-types of diabetes mellitus: type-1 and type-2. The treatment for diabetes consists of regulating a patient's blood glucose level to stay within a specific range. In order to keep their blood glucose level in an acceptable range, type-1 diabetic patients must inject insulin several times during a day. The amount of insulin that needs to be injected depends on the amount of carbohydrate in the last meal consumed by the patient and current blood glucose level. This is because, when we eat food, our digestive system breaks the carbohydrates down to glucose. The absorption of glucose in the intestine increases its concentration in the blood stream, which puts the body into a state of hyperglycemia (state of high blood glucose). Glucose, the key source of energy in human body, needs insulin for its routine disposal into cells. In a healthy individual, the pancreas produces insulin, which allows muscle and fat cells to absorb glucose from blood stream. Consequently the blood glucose level decreases back to the normal level. Other mechanisms operate when the blood glucose goes below its normal value – that is, when the body enters a state of hypoglycemia. The global situation of diabetes afflicting people is shown in Figure 1(a).

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

Preprint. Work in progress.



(a) Diabetes Scenario Worldwide



(b) Sepsis severity resulting in death

Figure 1: Severity of Diabetes and Sepsis

2.2 Sepsis

Our second motivating example is sepsis treatment in ICU. Sepsis is a complication of an infection resulting out of an extreme immune system response triggering widespread inflammation throughout the body. Sepsis can range from mild to severe and because it can be potentially life-threatening, it requires sustained and immediate medical attention. Sepsis treatment varies and depends on the cause of the infection that led to sepsis, as well as the severity of symptoms. Because mild sepsis can rapidly progress to severe sepsis and then septic shock, doctors must work quickly to reduce inflammation. Common treatments for sepsis include: 1. administering Antibiotics 2. injecting Intravenous (IV) Fluids and 3. in the extreme cases when blood pressure has fallen dangerously low using Vasopressors. An illustrative figure showing the severity of sepsis leading to death resulting from delay in administering antibiotics is shown in Figure 1(b).

3 Why Reinforcement Learning?

A large number of problems in science and engineering, robotics and game playing, resource management, financial portfolio management, medical treatment design, ad placement, website optimization and packet routing can be modeled as sequential decision-making under uncertainty. Many of these real-world interesting sequential decision-making problems can be formulated as reinforcement learning (RL) problems (see (Bertsekas and Tsitsiklis, 1996), (Sutton and Barto, 1998)). In an RL problem, an agent interacts with a dynamic, stochastic, and unknown environment, with the goal of finding an action-selection strategy or policy that optimizes some long-term performance measure. Every time when the agent interacts with the environment it receives a signal/reward from the environment based on which it modifies its policy. The agent learns to optimize the choice of actions over several time steps which is learned from the sequences of data that it receives from the environment. This is the crux of online sequential learning.

This is in contrast to supervised learning methods that deal with labeled data which are independently and identically distributed (i.i.d.) samples from the considered domain and train some classifier

on the entire training dataset to learn the pattern of this distribution to predict the labels of future samples (test dataset) with the assumption that it is sampled from the same domain. In contrast to this, an RL agent learns from the samples that are collected from the trajectories generated by its sequential interaction with the system. For an RL agent, the trajectory consists of a series of sequential interactions whereby it transitions from one state to another following some dynamics intrinsic to the environment while collecting the reward till some stopping condition is reached. This is known as an episode. Here, for an action a_t taken by the agent at the t-th timestep, the agent transitions from its current state denoted by S_t to state S_{t+1} and observes the reward $R(s_t, a_t)$. An illustrative image depicting the reinforcement learning scenario is shown in Figure 2.

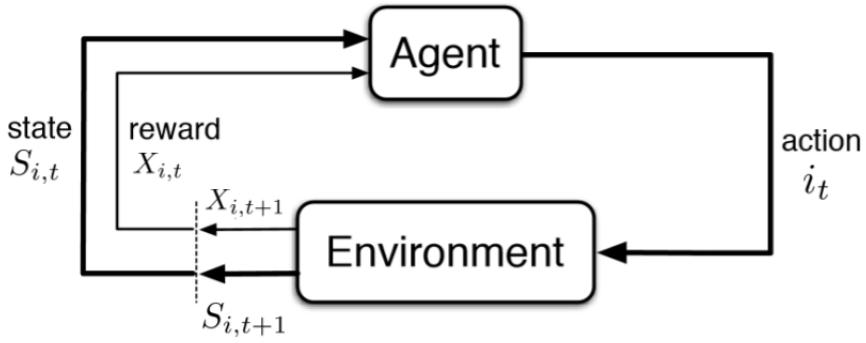


Figure 2: Reinforcement Learning

In the healthcare domain, there exists many scenarios in which the treatment involves taking a series of decisions over a long time. After every medical decision is made, and a treatment administered the condition of the patient changes. Based on this new condition of the patient medical practitioners may alter their evaluation policy to administer a new set of treatments or may continue with the last policy with the same dosage. Notice, that this is quite similar to the general RL framework where the condition of the patient can be defined by the state S_t, the treatment administered can be defined by action a_t and after administering the treatment the new condition that the patient transitions to can be defined by S_{t+1}. We will formalize this setting in Section 5 while we will illustrate several challenges where this simple framework will fail in real world scenarios in Section 6.

As mentioned in Nemati et al. (2016) RL is particularly well-suited for the medication dosing problem given the sequential nature of clinical treatment where multiple treatment decision are performed without immediate knowledge of effectiveness. Indeed, the lack of a one-to-one correspondence between actions and outcomes makes it difficult to assign credit or blame to individual actions along the way to an intermediate or terminal outcome. Moreover, the effect of interventions for a given patient can be non-deterministic, and attempting to predict the effects of a series of treatments over time only causes more uncertainty.

4 Notations, Assumptions and Definitions

We use capitalized calligraphic notations to denote sets while individual elements within the set is denoted by non-capitalized alphabets. The random variables are denoted by capitalized, non-calligraphic alphabets. $\mathcal{A}$ denotes the finite set of actions with individual action indexed by $A_t = a$ such that action taken at time t is a. We assume that the total number of actions is constant throughout the time horizon. We assume that the transition function is stationary that is, it is not changing between episodes.

5 MDP Formulation

A RL setting is usually characterized by a MDP or Markov Decision Process. A MDP is defined by the tuple $\{\mathcal{S}, \mathcal{A}, P, d_R, d_0, \gamma\}$ where each element of the tuple is defined as:-

1. $\mathcal{S}$ is the finite state space such that at each time step t the patient is in state $S_t \in \mathcal{S}$. This state space can be discrete or continuous depending on the modeling assumption of the learner.
2. $\mathcal{A}$ is the action space such that at each time t, the agent takes action $\mathcal{A}_t \in \mathcal{A}$, which causes it to change its state from S_t to S_{t+1}. Again this action space can be discrete or continuous depending on the modeling assumption of the learner.
3. P is the transition function which describes how the state of the environment changes. So, $P(s, a, s') = Pr(S_{t+1} = s' | S_t = s, A_t = a)$
4. d_R denotes the process of reward generation when the state of the agent changes.
5. d_0 denotes the initial state distribution of the agent.
6. The discount factor, γ, determines the relative weight of immediate and long-term rewards.

The goal of the RL agent is to learn a policy, i.e. a mapping $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ from states to actions, that maximizes the expected discounted return G_t

$$J(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} G_t | \pi] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_t | \pi]$$

where R_t is all the accumulated rewards by the agent and T denotes the time horizon.

6 Some Challenges of Medical Domain

Some of the challenges that rises out of the medical domain is to formulate the various aspects of the MDP.

6.1 State Representation

The medical environment is a partially observed environment. At any instant the physician is only exposed to some of the factors influencing the health of the patient. The state space can be discrete or continuous, depending on the disease that is being specified or how the model is defined. The continuous state space suffers from the same problems as in general reinforcement learning. Often the data about the patient history is inadequate or missing and hence cannot be represented effectively by all the features specified. There maybe cases when the patient itself does not comply with the prescribed treatment and so the interaction itself is missing. Often treatments cannot be directly administered to the patient and the policy needs to be learnt from the patient's history of interaction. This results in the situation called off-policy policy evaluation algorithms that learns an effective treatment policy without actually running the treatment itself. Again these off-policy algorithms have high variance and in the continuous state space their performance suffers heavily.

6.2 Reward function formulation

The medical environment suffers from long horizon problem where the learner only receives the feedback at the end of the episode or the feedbacks are very sparse in nature. Often the reward function itself has to be defined based on the disease itself. This can be handled to some extent by the inverse RL (Ng and Russell, 2000) approach where we learn the reward function itself from the patient history. parse rewards and confounding variables in the real-life datasets are another set of challenges that needs to be handled carefully. If not handled with care these may result in the algorithm proposing bizarre policy which will not go well with the clinicians.

6.3 Action formulation

As specified earlier, for a variety of reasons, the state-action interaction history may not exist at all. Handling such situation is a difficult situation. The action space can also be continuous, for example dosing range (Bastani, 2014) which is a difficult scenario to handle. The actions proposed by the algorithm at each state needs to be safe and trustworthy to the physician. Deriving such confidence

interval for action for off-policy algorithms in continuous state space (and possibly continuous action space) is another important challenge.

3. Transition Probability formulation: (Have to write)

7 Discussion on Algorithms

7.1 Off-Policy algorithms

In the off-policy setting, there are two stationary Markov policies, one used to generate the data, called the behavior policy and another one called the target policy whose value function we seek to estimate. The two policies can be completely arbitrary but subject to some constraints. The behavior policy must be soft, that is it must have a non-zero probability of selecting every action in each state. Some algorithms require even weaker constraints on the behavior policy that it can be stationary and non-starving.

Off-policy evaluation is difficult because there is a mismatch of distributions. Since the learner has to estimate the target policy but is only goven samples from the behavior policy. A classical way of handling such situations comes from Rubinstein (1981) by the way of *Importance Sampling*. Several interesting algorithms in the Reinforcement Learning setting have been proposed for off-policy evaluation incorporating Importance Sampling. These Per-Decision Importance Sampling (Precup et al., 2000), Per-Decision Weighted Importance Sampling (Precup et al., 2000), Doubly Robust Importance (Jiang and Li, 2015) Sampling and Weighted Doubly Robust Importance Sampling (Thomas and Brunskill, 2016).

Fitted Q iteration (FQI) is a batch RL algorithm whose main fea- ture lies in the way that it handles the experience (Ernst et al., 2005). Unlike incremental algorithms like Watkin's Q-learning (Watkins and Dayan, 1992), FQI uses the complete set of transitions each time that updates the estimation of the optimal Q-function. Although this process involves more computation, it allows to extract more information from the stored experience. Conse- quently, FQI is more data-efficient than other RL algorithms. This feature makes FQI a very suitable algorithm in many application domains. In certain scenarios it is quite expensive to conduct an experiment with respect to both money and time. For, example administering a dose to a patient and waiting to observe its effect. Thus, reducing the quantity of data required by the algorithm can be crucial.

7.2 Value Function based methods

7.3 Policy Gradient Methods

7.4 Using Linear and Non-Linear function approximation

8 Related Papers

We create a comprehensive list of papers which uses Reinforcement Learning in clinical applications by scraping through PubMed and Google scholar. This process is elaborately shown in Figure 3.

Some of the survey papers which give a broad description of the state-of-the-art approaches for RL in biological data are Mahmud et al. (2018), Kappor et al. (2018). Among theses Mahmud et al. (2018) focuses in non-linear function approximation using Deep Learning techniques in RL for biological data. They review papers from bio-imaging, medical-imaging, human-machine-interfaces, etc which uses RL as their learning mechanism of incorporating feedback and using function approximation using deep learning architectures as function approximators.

We review some of the relevant papers related to sepsis, ICU patients, Lung Cancer, Epilepsy, Heparin Dosing treatment.

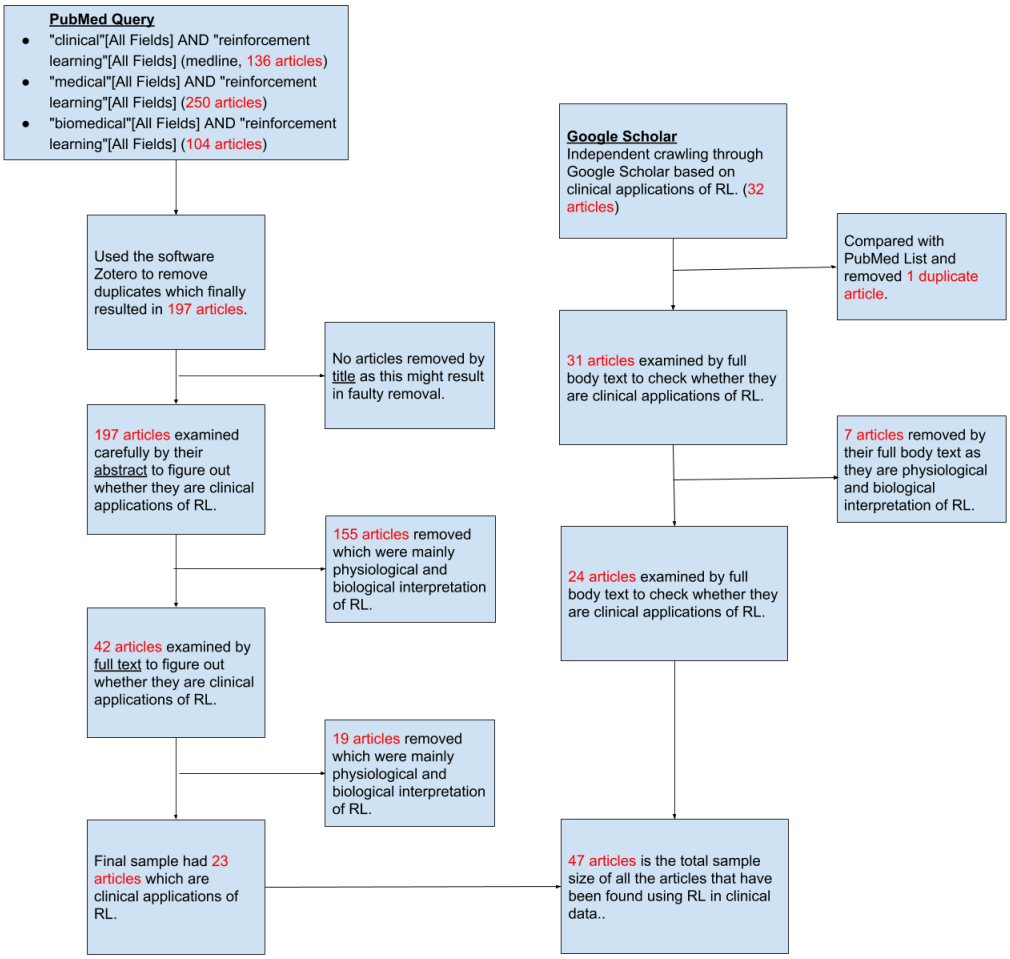


Figure 3: Scraping paper from PubMed and Google Scholar

Table 1: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Raghu et al. (2017)	Sepsis	Dueling Double-Deep Q Network (Off-policy algorithm)	**1)** Continuous States **2)** Discrete Actions	**1)** Deep RL models with continuous-state spaces, improving on earlier work with discretized models. **2)** Identify treatment policies that could improve patient outcomes **3)** Investigate the learned policies for clinical interpretability	**1)** Q-values are frequently overestimated in practice, leading to incorrect predictions and poor policies. So, uses Double-Deep Q Network, where the target Q values are determined using actions found through a feed-forward pass on the main network, as opposed to being determined directly from the target network. **2)** For finding optimal treatments, they separate the influence on Q-values of a) a patient's underlying state being good (e.g. near discharge), and b) the correct action being taken at that timestep. So, uses a Dueling Q Network, where the action-value function for a given (s, a) pair, Q(s, a), is split into separate value and advantage streams. The value stream represents the quality of the current state, and the advantage represents the quality of the chosen action. Training such a model can be slow as reward signals are sparse and only available on terminal timesteps. They use Prioritized Experience Replay to accelerate learning by sampling a transition from the training set with probability proportional to the previous error observed.	Their policies learned that vasopressors may not be a good first response to sepsis and maybe harmful in some populations.	**1)** The reward assignment in this model is quite sparse, with rewards/penalties only being issued at terminal states. There is scope for improvement here; one idea could be to use a clinically informed reward function based on patient blood counts to help learn better policies. **2)** Another approach could be to use inverse RL techniques to derive a suitable reward function based on the actions of experts (the physicians).

Table 2: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Weng et al. (2017)	Sepsis	Off-policy evaluation using policy iteration for π^*, π^r from real trajectories.	**1)** Discrete State Space, with patient conditions being noted at regular intervals. **2)** Discrete Actions with continuous glucose level being categorized into 11 bins.	**1** They hypothesize that the patient states, glycemic values, and patient outcomes can be modeled by a Markov decision process (MDP) whose parameters and optimal policies can be learned from data. **2)** They develop a decision support for glycemic control to target specific ranges of serum glucose that optimizes outcomes and is personalized for each patient depending on their specific circumstances.	**1)** To learn the patient state representation they use two types of feature representations: raw interpretable clinical features and the feature representation generated by a sparse autoencoder. After they generate the state representation, they categorize the patients into 500 clusters by using k-means clustering algorithm. **2** Policy Iteration algorithm is used to learn π^* which is the behavior policy. The estimation policy π^r is evaluated based on real trajectories where they limited the action space of each state to only the one with the highest probability in the transition matrix instead of exploring all possible actions. **3)** π^r and real mortality rate were used to obtain the estimated *mortality–expected return function*, which reveals the relationship between expected return and the estimated 90-day mortality rate. This function was used to compute and compare the estimated mortality rate of real and optimal glycemic trajectories obtain by π^r and π^*	**1)** If clinicians chosen dosages can actually achieve the target glucose levels chosen by the policy then it may reduce the mortality rate of septic patients. **2)** Their mortality–expected return function shows that using raw feature representation or learned feature representation using autoencoder may yield a good result, that is both are close to mortality rate calculated from the real data. latent representation. Both are close to the mortality rate calculated from the real data (31.17%).	**1)** State space is discrete, which is an issue. **2)** The off policy evaluation needs to be better.

Table 3: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Gottesman et al. (2018) **This paper is very impt. as it is like a review paper detailing the challenges**	Sepsis	Comapre Per-Decision Importance Sampling (PDIS), Weighted Per-decision Importance sampling (WPDIS), Doubly-Robust (DR), and Weighted Doubly-Robust (WDR).	**1)** Discrete State Space, with patient conditions being noted at regular intervals. **2)** Discretized treatment IV fluids and vasopressors each into 5 bins, the first representing no treatment (zero dosage), and the rest representing quartiles of the actions prescribed by physicians. Hence total 25 actions. **3** Reward is zero till the last action.	**1)** Data needs to be processed correctly otherwise the susceptibility of AI algorithms to learn harmful policies due to artifacts in the data increases. **2)** The algorithm learns to recognize patients who need additional care *but* lack of options in actions makes the algorithm choose intubation which is not recommended. **3)** They observed the learned policies recommend minimal treatment for patients with very high SOFA (Sequential Organ Failure Assessment) score. This recommendation is faulty but algorithms predict this because the mortality rate for this subpopulation is high and hence the policy have not learnt what to do.	**1)** The weighted methods (WPDIS, WDR) trade increased bias for reduced variance, while the per decision methods reduce variance by computing the weights in a way that does not penalize the probability of a current action based on future ones. **2)** Doubly robust methods (DR, WDR) leverage an approximate model of the reward function to reduce variance. **3)** All of the policies have relatively close median values and large variances, making it hard to draw definitive conclusions. The model-based WDR estimator uses a model to reduce variance, but also inherits the optimistic bias of the model. The model-free WPDIS estimator also suffers from large variances. **4)** To the patients belonging to a lower risk group, the WPDIS method suffers from a selection bias. It predicts a no-treatment policy to these group as they have lower mortality rate.	**1)** State representation need to account for any variables that might confound estimates of outcomes under the policy. **2)** It's impossible to account for the entire history of the patient and determine/avoid such confounding variables. Instead, domain knowledge by an expert/clinical researcher must be applied to take care of this. This is especially a difficult problem to solve in sequential setting.	**1)** If outcomes are sparse then performance suffers. **2)** High variance in the performance of Importance sampling algorithms as some actions which are never tested has close to zero probability. **3)** Sufficient confidence on the action by the policies cannot be guaranteed.

Table 4: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Raghu et al. (2018)	Sepsis	Per Horizon Weighted Importance sampling (PHWIS), and Per Horizon Weighted Doubly-Robust (PHWDR).	**1)** Continuous State Space (Toy domain) **2)** Discrete Action Space (Toy Domain)	**1)** This work evaluates the sensitivity of off-policy evaluation to calibration errors in the learned behaviour policy. They show how powerful parametric models such as neural networks can result in highly uncalibrated behaviour policy models on a real-world medical dataset	**1)** They use PHWIS and PHWDR instead of step-wise IS and DR to reduce variance. **2)** To split the horizon for estimation and behavior policy, two methods are considered, random and intervention splitting. Random splitting randomly chooses half the trajectories for each policies, while intervention splitting splits patients who have been treated with vasopressors (or not). **3)** To compare between π_e and π_b the use Mean square estimation.	**1)** Uncalibrated behaviour policy models can result in highly inaccurate OPE in a simple, controlled navigation domain. **2)** In a real-world sepsis management domain, powerful parametric models such as deep neural networks produce highly uncalibrated probability estimates. Neural networks can produce overconfident and incorrect probability estimates of actions. **3)** A simple, non-parametric, k-nearest neighbours model is shown to be better calibrated than all the other parametric models in their medical domain, and using this as a behaviour policy model results in superior OPE.	**1)** The proposed procedure can be used in other situations where the behaviour policy is unknown, and could improve the quality of OPE estimates.

Table 5: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Prasad et al. (2017)	ICU patient	Fitted Q-Iteration wither Extra Trees and Neural Network as function approximators.	**1)** Continuous State Space **2)** Discrete Action Space **3)** They do not consider this as a POMDP	**1)** This work develops a decision support tool to alert clinicians when a patient is ready for weaning (taken off mechanical ventilation). **2)** It uses available patient information in the ICU setting and proposes the off-policy Fitted Q-Iteration (FQI) algorithm with different regressors for optimal treatment.	**1)** Simple Q-Learning using 3 layers of hidden layer fails to learn propoerly. **2)** They use FQI (with batch mode learning) with Regressor as Extra Trees for Function approximation and this performs well. **3)** Neural FQI with 3 hidden layers for function approximation also performs well in this dataset. Neural FQI achieves a four-fold gain in performance as compared to FQI with extra trees.	**1)** They show that the algorithm is capable of extracting meaningful indicators in recommending extubation time and sedation levels, on average outperforming clinical practice in terms of regulation of vitals and reintubations for patients.	**1)** Policies must show some invariance to reward shaping. The current methods display considerable sensitivity to the relative weighting of various components of the feedback received after each transition. A more principled approach to the design of the reward function, for example by applying techniques in inverse reinforcement learning (Ng and Russell, 2000), can help tackle this sensitivity. **2)** Effective communication of the best action, expected reward, and the associated uncertainty, calls for a probabilistic approach to estimation of the Q-function, which can perhaps be addressed by pairing regressors such as Gaussian processes with Fitted Q-iteration. **3)** Increase the sophistication of the state space by handling long term effects more explicitly using second-order statistics of vitals **4)** Modeling the system as a partially observable MDP, in which observations map to some underlying state space. **5)** Extending the discrete action space to continuous action space so that continuous dosages of specific drug types and settings such as ventilator modes can be taken into account.

Table 6: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Padmanabhan et al. (2014)	Anesthesia of ICU patient with respiratory disease symdromes	Modified Watkin's Q-learning (on-policy).	**1)** Discrete State Space **2)** Discrete Action Space	**1)** This work develop a RL-based closed- loop anesthesia controller using the bispectral index (BIS) as a control variable while concurrently accounting for mean arterial pressure (MAP). **2)** This work uses these two parameters to control propofol infusion rates to regulate the BIS and MAP within a desired range.	**1)** The states of the system should be observable for decision making. **2)** The states of the system are based on the measurable parameters BIS and MAP. **3)** The error is measured based on a weighted combination of the error of the BIS(error) and MAP(error). This reduces the computational complexity of the RL algorithm and consequently the controller processing time**4)** Finally Q-Learning is used to learn the sequence of infusion rates that results in a minimum BIS(error) and MAP(error).	**1)** In this paper, a reinforcement learning-based approach for the simultaneous control of sedation and hemodynamic parameter management is proposed using the regulation of the anesthetic drug propofol. **2)** Simulation results using 30 patient models with varying pharmacokinetic and pharmacodynamic parameters show that the proposed RL control strategy is promising in designing closed-loop controllers for ICU sedation to regulate sedation and hemodynamic pa- rameters simultaneously. **3)** The simulations show that the RL-based, closed-loop control is robust to system uncertainties.	**1)** Discrete State and Action Space is a drawback **2)** Too less number of patients in the experiment, so doubtful conclusions can be drawn.

Table 7: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Zhao et al. (2011)	Treating Non-Small Cell Lung Cancer (NSCLC)	Q-learning with SVR used for function approximation (on-policy).	**1)** Discrete State Space **2)** Discrete Action Space	**1)** This work presents an adaptive reinforcement learning approach to discover optimal individualized treatment regimens for patients with advanced NSCLC. **2)** Q-learning is used to learn an optimal regimen from patient data generated from the clinical reinforcement trial.	**1)**The proposed clinical reinforcement trial for NSCLC involves a randomization of patients among the different therapies in first and second-line treatments, as well as randomization of second-line initiation time. This design enables estimation of optimal individualized treatment regimes. **2)** Next, reinforcement learning is used to analyze the resulting data. They use Q-Learning with a modified SVR (Vapnik et al., 1996) to fit nonlinear Q-functions for each of the two decision times (before first line and before second line). This is required to handle the complex fact of heterogeneity in treatment across individuals as well as right-censored survival data. **3)** In addition, a second, confirmatory trial with a phase III structure is conducted after the first trial to validate the optimal individualized therapy.	**1)** They believe that Q-functions in clinical applications will be too complex for para-metric regression and that semi-parametric and non-parametric regression approaches, such as -SVR-C, is needed.	**1)** Future work includes giving a confidence set for the resulting treatment regimens and associated Q-functions **2)** How to determine an appropriate sample size for a clinical reinforcement trial to reliably obtain treatment regimen that is very close to the true optimal regimen.

Table 8: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Escandell-Montero et al. (2014)	Anemia treatment in Hemodialysis patients	Fitted Q-Iteration algorithm with Extremely Randomized trees.	**1)** Discrete State Space **2)** Discrete Action Space	**1)** The methodology proposed in this work uses the algorithm fitted Q iteration to learn a policy of ESA administration from a set of medical records. The features employed to define the MDP model are extracted in part from the laboratory tests and in part from a clustering procedure of the patient's main attributes. In order to test the methodology, a series of experiments has been conducted using a computational model that simulates the response of the patients. The performance has been assessed against the algorithm Q-learning and a standard protocol of dose adjustment.	**1)** The Gaussian RBF network with fixed bases is employed to approximate the Q-function. This requires the definition of the number of Gaussian functions, their centers and standard deviations. This process typically requires trial and error experimentation with various configurations.	**1)** In this paper, a reinforcement learning-based approach for the simultaneous control of sedation and hemodynamic parameter management is proposed using the regulation of the anesthetic drug propofol. **2)** Simulation results using 30 patient models with varying pharmacokinetic and pharmacodynamic parameters show that the proposed RL control strategy is promising in designing closed-loop controllers for ICU sedation to regulate sedation and hemodynamic pa- rameters simultaneously. **3)** The simulations show that the RL-based, closed-loop control is robust to system uncertainties.	**1)** Discrete State and Action Space is a drawback **2)** Too less number of patients in the experiment, so doubtful conclusions can be drawn.

Table 9: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Guez et al. (2008)	Epilepsy	Fitted-Q Iteration (on-policy).	**1)** Discrete State Space **2)** Discrete Action Space	**1)** This paper examines the problem of applying reinforcement learning technology to optimize control strategies for deep-brain electrical stimulation in the treatment of epilepsy. **2)** In this case, acquiring large amounts of patient data is extremely expensive and invasive. Therefore they use of batch reinforcement learning techniques to learn from in vitro studies of stimulation.	**1)** Informally, the learning problem can be formulated as follows: at every moment in time, given some information about what happened to the signal previously (our state), we need to decide which stimulation action we should choose (if any) so as to minimize seizures now and in the future. **2)** The fitted Q iteration algorithm requires a supervised regression algorithm to learn the Q-functions. In this paper they use Extremely Randomized trees.	**1)** Their results show that by using reinforcement learning, they are able to reduce the incidence of seizures by 25%, compared to the current best stimulation strategies in the neuroscience literature (and 60% compared to when there is no stimulation).	**1)** Discrete State and Action Space is a drawback **2)** Some of the important questions and future directions noted by them are mentioned here:- How should we quantify performance of adaptive strategies? How we can learn from very little training data? Can we design "safe" exploration policies, with formal guarantees on worse-case performance? How can we re-use data, or learned policies, between different patients?

Table 10: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Nemati et al. (2016)	Heparin Dosing	Discriminative Hidden Markov Model (DHMM) for state estimation. Within the fitted Q-learning framework the Q-function is represented by a neural network. (off-policy)	**1)** Discrete State Space **2)** Discrete Action Space	**1)** This work tries to infer an optimal dosing strategy that accounts for both the activated partial thromboplastin time (aPTT) level, and evolving patient physiological condition. 2)To accomplish this inference, they train a RL model (using DHMM and Neural FQI) using the time series of several common clinical measurements within the patient's electronic medical record (EMR).	**1)** The objective of the RL medication dosing agent is to learn a dosing policy that maximizes the overall fraction of time a given patient stays within his/her therapeutic aPTT range. **2)** Since the actual physiological state of the patient is at best only partially observed, the agent has to infer both the state of the patient and an optimal policy from sample trajectories of its interaction with the environment. **3)** When optimizing over a large patient cohort, a stochastic optimization approach—using mini-batches with a few iterations per batch and a momentum term—yielded improved generalization performance with significant speed up. **4)** Hyper-parameters of the DHMM and the neural network representing the policy (such the number of layers and nodes) were tuned using Bayesian Optimization.	**1)** The RL agent's recommendation starts slightly above the population mean for heparin and then converges to the population mean, which is likely to bring patients within their therapeutic range more quickly. **2)** They further tested this hypothesis, and found that patients whose administered heparin trajectory most closely followed the RL agent's policy could on average expect a positive reward after just a few adjustment and stay within range.	**1)** Whether the suboptimal heparin dosing we observed were from intentional actions on the part of the clinician, mistakes, or simply due to a lack of adherence to hospital guidelines are beyond our ability to investigate with the dataset at hand. This points at one of the major challenges of retrospective analysis of clinical big data; the rational for treatment decisions are often unknown, and some features which may be important for understanding outcomes may be missing, most likely not at random.

Table 11: Review of papers

Paper	Disease	Algo type	MDP info	Contributions	Approach	Conclusions/ Observations	Limitations & Future Works
Ernst et al. (2006)	HIV infected patient	fitted Q iteration	**1)** Discrete State Space **2)** Discrete Action Space	**1)** This work computes optimal structured treatment interruption strategies for HIV infected patients. They show that reinforcement learning may be useful to extract such strategies directly from clinical data, without the need of an accurate mathematical model of HIV infection dynamics.	**1)** They use batch-mode supervised learning Extra-Trees algorithm (Geurts et al., 2006). This algorithm builds a model in the form of the average prediction of an ensemble of regressions trees obtained by randomization.	**1)** Trial-and error approaches were chosen for setting the hyper-parameters. But this is a risky approach and cannot be used on real patients. There is a need to rely on medical expertise in order to state properly the optimal control problem. **2)** Also some specific tools should be built to help in this task. **3)** Based on a sufficient amount of simulated data, they found that reinforcement learning was indeed able to derive STI therapies which appear as excellent when used to "treat" simulated patients.	**1)** One of their limitation was that they did not consider partial observability. In their example they assumed that all the state variables were directly observable. **2)** They also did not account for corrupted measurements. Collected clinical data are not necessarily thorough and accurate. **3)** Furthermore, the patients may not necessarily comply with the prescribed treatment. This may lead to uncertainties and measurement corruption which may significantly degrade the quality of the results obtained. One solution to mitigate the adverse effects of corrupted measurements would be to design some preprocessing algorithms able to filter out highly corrupted data.

9 Some Toy Domains

In this section we build a gadget world or a toy domain for the Reinforcement Learning (RL) setup for the medical domain. A gadget problem is a simple environment which captures some of the complexities of the real-world domain which we are trying to model. We can test various RL algorithms in this gadget worlds before transitioning to the real-world higher complexity environments. The hypothesis behind creating such gadget worlds is that if an RL algorithms performs poorly in this small gadget world, it will surely perform poorly in real-world domains.

We introduce the 10×10 gridworld Figure 4(a) which has the following features:-

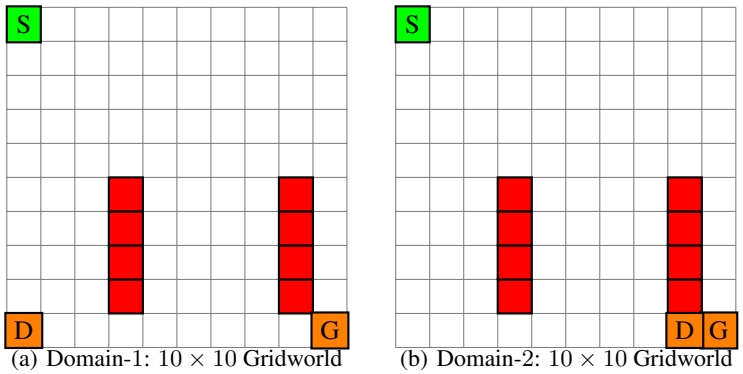
1. The starting state is shown as S in green color.
2. At any grid, only four discrete actions are possible, left, right, bottom, top.
3. The obstacles are shown in red colors. When an agent hits the obstacles, it stays in the state before attempting to transition.
4. The only difference between Domain 4(a) and Domain 4(b) is the position of the death state $S_t = D$.
5. The terminal states are shown in orange. D represents the state "death" with a negative reward of $R(S_t = D, A_t = a) = -60$ while G represents the state "get well" with a time-varying reward of $R(S_t = G, A_t = a) = 60 - t, 1 \leq t \leq 60$. All the other transitions result in a reward of 0.
6. Note, that the time t is part of the representation of state as rewards are changing with time t.
7. The states are featurized by the function $\Phi : S \rightarrow R^{r+c}$, where r is the number of rows and c is the number of columns in the gridworld. So, $\Phi(s)$ is a function that maps states to vectors of features. We define $\Phi(s)$ such that for the state $s_{i,j}$, where i is the row-index and j is the column index in the grid, then $\Phi(s_{i,j})$ is the vector v such that,

$$\begin{aligned} v_k &= 1, \text{ if k = i+j} \\ &= 0 \text{ otherwise,} \end{aligned}$$

   and $k = 1, \ldots, (r + c)$ is the index of the vector v.

Next, we illustrate why these features where included in these gadget worlds and link up with our discussion on the complexities of the medical domain.

1. The state space is discrete and and the action space is also finite and discrete. We wanted to keep the gadget worlds simple.
2. Domain 4(b) is slightly more difficult than Doamin 4(a) as the path to "get well" state $S_t = G$ is more restricted in the former.
3. There is only substantial reward (positive/negative) at the end of the long episodes when the agent reaches the states either $S_t = G$ or $S_t = D$. This handles the long horizon problem.
4. Rewards are also adversarial as they are changing with time. Because of this, if the agent reaches the goal state at $t = 60$, it receives a reward of 0. Moreover, the rewards are diminishing with time indicating, that the agent has to reach the terminal "get well" state quickly.
5. The partially observed environment is captured in how we are featurizing the states. Note that $\Phi(s_{2,3})$ will have the same embedding as $\Phi(s_{3,2})$. This follows from the idea that when feature representation of states are *not* rich enough it result in a partially observed environment.
6. The obstacles, (marked in red) forces the q-value function approximator not to generalize too well. These makes the simple environment slightly more difficult to be generalized well enough.



(a) Domain-1: 10×10 Gridworld (b) Domain-2: 10×10 Gridworld

Figure 4: Description of two Toy gridworld domain

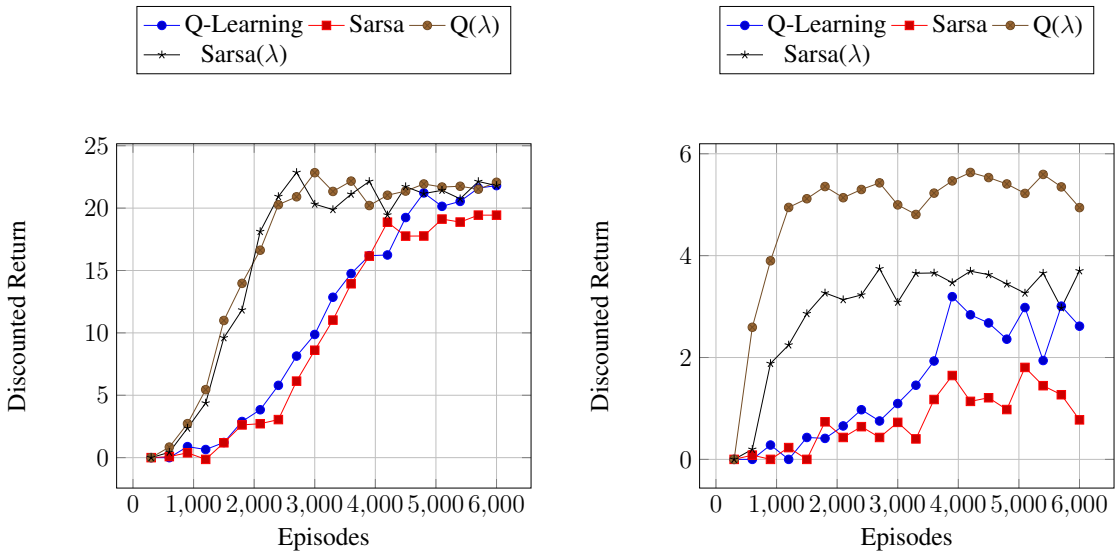
10 Experiments

10.1 Toy Gridworld Domain

In this section, we run Q-learning and Sarsa with linear function approximation in the two gridworld domain shown in Figure 4(a) and Figure 4(b). The results of the experiments are shown in Figure 5(a) and Figure 5(b) for the domain 1 and 2 respectively. All the algorithms were averaged over 50 independent trials and each trial consisted of 6000 episodes.

Experiment 1 (Domain 1): In this experiment we use linear function approximation for both Q-Learning and Sarsa to handle this partially observed environment. From Figure 5(a) we see that Sarsa performs better than Q-Learning in this Domain and stabilizes before Q-Learning.

Experiment 2 (Domain 2): In this experiment again we use linear function approximation for both Q-Learning and Sarsa to handle this partially observed environment. From Figure 5(b) we see that Sarsa performs worse than Q-Learning in this Domain. Infact both the algorithms does not stabilize in this experiment. This results from the fact the the entry to the state $S_t = G$ is restricted and both the algorithms spend considerable amount of time in fruitless exploration.



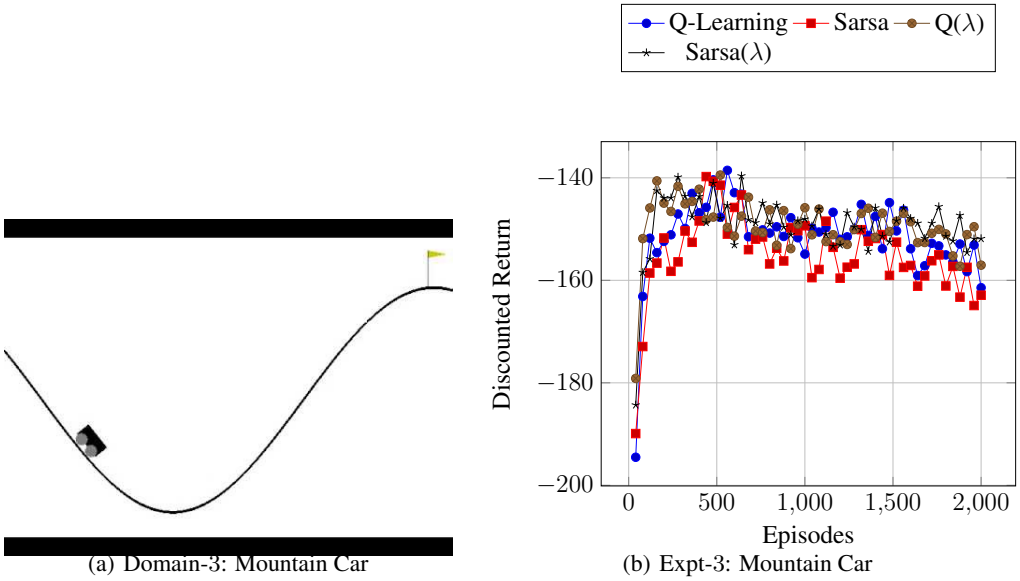
(a) Expt-1: 10×10 Gridworld (Domain 1) (b) Expt-2: 10×10 Gridworld (Domain 2)

Figure 5: A comparison of the performance of various algorithms.

10.2 Classic Domain

The mountain car was first described in Andrew Moore's Thesis (**?**) and was latter properly defined in Singh and Sutton (1996). The task consist of driving a car resting in a valley up the mountain. The main challenge of this task is that the car by itself cannot drive up the mountain and it has to swing back and forth to gather the sufficient momentum to reach the top of the mountain (see Figure 6(a)). Nonetheless, this simple environment consist of several challenges that afflicts the medical domain. It's a continuous state space problem, hence function approximation has to be used which makes it a partially observed MDP. Moreover, the car can only accumulate a positive reward of $+50$ when it reaches the top or suffers a negative reward of -1 the time while it swings back and forth. So this models the long horizon problem. The action space is discrete in this toy domain.

In Figure 6(b) we show how Q(λ) and Sarsa(λ) along with Fourier basis can be used to solve this problem.



(a) Domain-3: Mountain Car

(b) Expt-3: Mountain Car

Figure 6: A comparison of the performance of various algorithms.

11 Conclusions and Future Works

In this report we reviewed some of the papers applying reinforcement learning techniques to medical domains. We discussed how RL algorithms are extremely important in modeling medical test-cases. We also describes general settings for the RL algorithm and some pf the challenges the this sequential tasks faces in real-life medical test-cases. Then we discussed in detail several important papers which have proposed some of the seminal RL algorithms in medical settings. Finally we came up with some gadget problems which are easy to handle and yet has sufficient complexities to handle many important and intriguing features of the real-life medical domain. We also showed that both Q-Learning and Sarsa with linear approximation fails to perform well in these domains. Future work includes proposing new algorithm that might perform better in these environments or to test planning algorithms in these domains.

References

Bastani, M. (2014). *Model-free intelligent diabetes management using machine learning*. PhD thesis, University of Alberta.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). Neuro-dynamic programming (optimization and neural computation series, 3). *Athena Scientific*, 7:15–23.

Ernst, D., Geurts, P., and Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556.

Ernst, D., Stan, G.-B., Goncalves, J., and Wehenkel, L. (2006). Clinical data based optimal sti strategies for hiv: a reinforcement learning approach. In *Decision and Control, 2006 45th IEEE Conference on*, pages 667–672. IEEE.

Escandell-Montero, P., Chermisi, M., Martínez-Martínez, J. M., Gómez-Sanchís, J., Barbieri, C., Soria-Olivas, E., Mari, F., Vila-Francés, J., Stopper, A., Gatti, E., and Martín-Guerrero, J. D. (2014). Optimization of anemia treatment in hemodialysis patients via reinforcement learning. *Artificial Intelligence in Medicine*, 62(1):47–60.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

Gottesman, O., Johansson, F. D., Meier, J., Dent, J., Lee, D., Srinivasan, S., Zhang, L., Ding, Y., Wihl, D., Peng, X., Yao, J., Lage, I., Mosch, C., Lehman, L. H., Komorowski, M., Faisal, A., Celi, L. A., Sontag, D., and Doshi-Velez, F. (2018). Evaluating reinforcement learning algorithms in observational health settings. *CoRR*, abs/1805.12298.

Guez, A., Vincent, R. D., Avoli, M., and Pineau, J. (2008). Adaptive treatment of epilepsy via batch-mode reinforcement learning. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 1671–1678.

Holt, R., Cockram, C., Flyvbjerg, A., and Goldstein, B. (2011). *Textbook of Diabetes*. Wiley.

Jiang, N. and Li, L. (2015). Doubly robust off-policy evaluation for reinforcement learning. *CoRR*, abs/1511.03722.

Kappor, R., Walters, S. P., and Al-Aswad, L. A. (2018). The current state of artificial intelligence in ophthalmology. *Survey of ophthalmology*.

Mahmud, M., Kaiser, M. S., Hussain, A., and Vassanelli, S. (2018). Applications of deep learning and reinforcement learning to biological data. *IEEE transactions on neural networks and learning systems*, 29(6):2063–2079.

Nemati, S., Ghassemi, M. M., and Clifford, G. D. (2016). Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2016, Orlando, FL, USA, August 16-20, 2016*, pages 2978–2981.

Ng, A. Y. and Russell, S. J. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 663–670.

Padmanabhan, R., Meskin, N., and Haddad, W. M. (2014). Closed-loop control of anesthesia and mean arterial pressure using reinforcement learning. In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL 2014, Orlando, FL, USA, December 9-12, 2014*, pages 1–8.

Prasad, N., Cheng, L., Chivers, C., Draugelis, M., and Engelhardt, B. E. (2017). A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *CoRR*, abs/1704.06300.

Precup, D., Sutton, R. S., and Singh, S. P. (2000). Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 759–766.

Raghu, A., Gottesman, O., Liu, Y., Komorowski, M., Faisal, A., Doshi-Velez, F., and Brunskill, E. (2018). Behaviour policy estimation in off-policy policy evaluation: Calibration matters. *CoRR*, abs/1807.01066.

Raghu, A., Komorowski, M., Celi, L. A., Szolovits, P., and Ghassemi, M. (2017). Continuous state-space models for optimal sepsis treatment - a deep reinforcement learning approach. *CoRR*, abs/1705.08422.

Rubinstein, R. Y. (1981). *Simulation and the Monte Carlo method*. Wiley series in probability and mathematical statistics. Wiley.

Singh, S. P. and Sutton, R. S. (1996). Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22(1-3):123–158.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT press.

Thomas, P. S. and Brunskill, E. (2016). Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2139–2148.

Vapnik, V., Golowich, S. E., and Smola, A. J. (1996). Support vector method for function approximation, regression estimation and signal processing. In *Advances in Neural Information Processing Systems 9, NIPS, Denver, CO, USA, December 2-5, 1996*, pages 281–287.

Watkins, C. J. C. H. and Dayan, P. (1992). Technical note q-learning. *Machine Learning*, 8:279–292.

Weng, W., Gao, M., He, Z., Yan, S., and Szolovits, P. (2017). Representation and reinforcement learning for personalized glycemic control in septic patients. *CoRR*, abs/1712.00654.

Zhao, Y., Zeng, D., Socinski, M. A., and Kosorok, M. R. (2011). Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer. *Biometrics*, 67(4):1422–1433.