Distribution Agreement

Signature:

______________________________ ______________

Rongmei Lin Date

A Reinforcement Learning Based Decision Support System for Heparin Dosing

By

Rongmei Lin
Master of Science

Computer Science

Shamim Nemati, Ph.D.
Advisor

Eugene Agichtein, Ph.D.
Committee Member

Joyce Ho, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

A Reinforcement Learning Based Decision Support System for Heparin Dosing

By

Rongmei Lin
B.A., South China University of Technology, 2013

Advisor: Shamim Nemati, Ph.D.

An abstract of
A thesis submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science
in Computer Science
2017

Abstract

A Reinforcement Learning Based Decision Support System for Heparin Dosing
By Rongmei Lin

Medication dosing is a comprehensive problem with uncertainties. Every patient has unique condition, meanwhile some drugs have narrow therapeutic windows. Mis-dosing might result in preventable adverse event. Therefore, a robust decision support system would be helpful to clinicians by providing advisable dosing suggestions. Heparin is one of the sensitive drugs. In order to build up the decision support system for heparin patients, we present a clinician in the loop framework with deep reinforcement learning algorithm. There are two main objectives in this thesis, the first one is providing individualized dosing suggestion based on the multi-dimensional features of patients. The second one is evaluating the dosing predicted by our decision support system. We implemented several experiments to achieve these objectives. The data used in the experiments including simulated data, MIMIC-II intensive care unit data and Emory hospital intensive care unit data. There are two important processes with respect to our objectives. In the training process, the decision support system learned from the dosing executed by clinicians and the corresponding response of patients. In the evaluating process, we explored the results from several aspects and focused on the causality between variables and outcomes. The experimental results suggested that given the states of patients, our medication dosing support system is able to provide a reasonable recommendation.

A Reinforcement Learning Based Decision Support System for Heparin Dosing

By

Rongmei Lin
B.A., South China University of Technology, 2013

Advisor: Shamim Nemati, Ph.D.

A thesis submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science
in Computer Science
2017

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In the medical therapeutic process, especially for patients in the intensive care unit (ICU), medication dosing problem is technically complicated. Although there are corresponding treatment protocols for reference based on clinical knowledge and previous experiences, the actual situation is always more diverse. In medical practice, each patient has specific health condition, most of the time clinicians need to determine the dosage by considering many factors of the particular scenario instead of applying protocols mechanically. These decisions are mainly made using heuristic strategies. However, an increasing number of voices from clinicians suggested that treatment decisions in some complex scenario are hard to make by using intuitions solely [1]. On the other hand, misdosing could be resulted from accidentally human errors, such as illegible handwriting, inappropriate method of infusion, improper documentation and similarly named medications. In conclusion, sophisticated situations and tiny mistakes could both cause the misdosing of drugs. Misdosing is a serious issue which will increase the risk of preventable adverse events (PAEs), a little bit of misjudgment or negligence might result in irreparably consequences. Based on an evidence-based estimation, there are approximately 400,000 premature deaths associated with PAEs per year in the United States. The number of severe harm cases are even as many as 10 to 20 times of these fatal harm cases [2].

Heparin, also know as unfractionated heparin (UFH), is one of the examples that might have misdosing issues in practice. Basically, heparin is an effective drug used as anticoagulant (blood thinners) to prevent the formation of blood clots. It is also a method with several adverse effects under circumstance of improper dosing, such as bleeding, renal failure, osteoporosis and Heparin-induced thrombocytopenia(HIT). Patients are usually sensitive to this medication due to its narrow therapeutic window. Over dose of heparin might result in bleeding complication such as decrease in platelet count. Under dose of heparin might lead to clotting complication such as pulmonary embolism (PE) and deep vein thrombosis (DVT). The frequent occurrences of misdoing on heparin and its consequences call for a comprehensive decision support system. In order to provide robust dosing suggestions, the system need to investigate the relationship between heparin dosing and multiple parameters of patients. In recent years, there is growing trend of retrospective clinical data both in breadth (through multi-center initiatives) and depth (using higher resolution data) [3], which provides a valuable opportunity to explore the medical treatment process. From the above, one of the objectives of this thesis is to provide individualized heparin dosing suggestion using a data-driven approach. Once we have the result, a further evaluation focused on the possible effect of suggested dosing is needed to support our method.

1.2 Approaches

1.2.1 Reinforcement learning

Reinforcement learning (RL) is a powerful learning method focus on how to make optimal decisions. The setting of RL problem can be described as: how an agent become proficient in an unknown environment, given only its perceptions and occasional rewards. In general, our goal of this project is to learn from retrospective clinical data and ultimately yield reasonable heparin dosing for each patient, which is well matched with the pattern of RL. So the data-driven approach we will present later is based on RL.

The environment in RL is typically described as Markov decision process (MDP), which is defined by:

- a set of states $s \in S$,
- a set of actions $a \in A$,
- an initial state distribution with density $p_1(s_1)$,
- a stationary transition dynamics distribution with conditional density $p(s_{t+1}|s_t, a_t)$,
- a reward function $r : S \times A \to \mathbb{R}$.

Figure 1.1 shows a simple MDP with three states (green circles), two actions (orange circles), several transition probabilities (black arrows) and two rewards (orange arrows). The "Markov" here means that the next state outcome only depends on the current state and action. This property reflects on the transition distribution for any trajectory in state-action space: $p(s_{t+1}|s_1, a_1, \ldots, s_t, a_t) = p(s_{t+1}|s_t, a_t)$. MDP is based on an assumption that the state is completely observable. However, in most of the real world sequential decision processes, the agent cannot directly observe the underlying

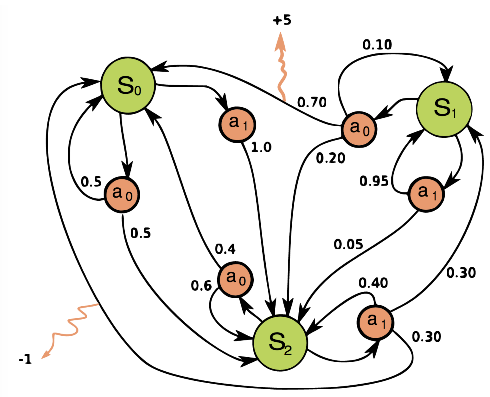

Figure 1.1: Example of MDP (Source: Wikepedia)

state. Partially observable Markov decision process (POMDP), which maintains a probability distribution over possible states, can be used to model internal observations.

Given the transition distribution $p(s_{t+1}|s_t, a_t)$ and the reward r of a environment, the RL agent will be able to calculate the optimal action without any interaction with the environment. Nevertheless, the environments in reality are usually model-free, which means the transition distributions and rewards of the MDP are unknown. In terms of learning patterns, there are several categories listed below. In the experiments part, we will explore different combinations of these methods.

- online learning: execute the policy and learn from the experience of in-complete episodes. (e.g. Temporal-Difference learning)

- offline learning: first collect complete episodes experience under policies, then solve the MDP directly from samples. (e.g. Monte-Carlo learning)

- on-policy learning: learn from the executed policy. (e.g. SARSA)

- off-policy learning: learn from the optimal policy which is independent of executed policy. (e.g. Q-learning)

1.2.2 Value based algorithm

As mentioned in the previous section, the goal of RL agent is to learn from unknown environment described as a MDP and select the optimal policy which maximizes future reward. In order to achieve this goal, given a state in the environment, we need to estimate action-value function with respect to possible action. An important assumption here is that the future reward is discounted by a factor γ, so the future accumulated reward of each time step can be defined as $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$, T in the equation is the termination time of each episode. With the reward, the optimal action-value function can be defined as $Q^*(s,a) = \max_\pi \mathbb{E}[R_t | s_t = s, a_t = a, \pi]$. We can use Bellman equation as an iterative update to estimate the $Q^*(s,a)$:

$$Q_{i+1}(s,a) = \mathbb{E}_{s' \sim S}[r + \gamma \max_{a'} Q_i(s',a') | s, a]$$

The action-value function will converge from $Q_i \to Q^*$ as $i \to \infty$. In practice, we need a function approximator $Q(s,a;\theta)$ to estimate the $Q(s,a)$ function. Such function approximator can be linear functions, or more commonly, non-linear functions like neural networks. We use a neural network with parameters θ in this thesis to approximate action-value function. A neural network can be trained by minimizing a loss function $L_i(\theta_i)$:

$$L_i(\theta_i) = \mathbb{E}_{s \sim S, a \sim A}[(y_i - Q(s,a;\theta_i))^2]$$

where $Q(s,a;\theta_i)$ is estimated value and $y_i = \mathbb{E}_{s' \sim S}[r + \gamma \max_{a'} Q(s',a';\theta_{i-1})]$ is the ground truth value. It is notable that the parameters θ_{i-1} used to calculate ground truth y_i is the old parameters in previous iterations. In order to tune the parameters in back propagation, we need to differentiate the loss function to get the gradient with respect to parameter θ:

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s,s' \sim S, a \sim A}\Big[\Big(r + \gamma \max_{a'} Q_i(s', a'; \theta_{i-1}) - Q(s, a; \theta_i)\Big)\nabla_{\theta_i} Q(s, a; \theta_i)\Big]$$

Finally, we could obtain a good estimator of $Q(s, a)$ function by minimizing the loss function with stochastic gradient decent. The RL agent will select optimal action refer to this action-value function.

1.2.3 Policy based algorithm

Value based algorithm is suitable for high dimensional states and discrete actions environment. However, it is impractical to use it directly to solve problems in continuous action domain. In order to solve problems with continuous actions, such as physical control tasks and medical dosing task in this thesis, we can use policy based algorithms. Policy Gradient algorithm is probably the most common used continuous action algorithm. The basic idea behind this algorithm is to tune the parameter of policy in the direction of the objective function gradient. There are two branches of policy gradient: stochastic policy $\pi_\theta(s, a) = p(a|s; \theta)$ and deterministic policy $a = \mu_\theta(s)$.

Stochastic policy

In terms of the classic stochastic policy gradient, it will consider objective function $J(\theta)$ for state density $\rho^\pi(s)$ as follows:

$$J(\theta) = \mathbb{E}_s\Big[\int_a \pi_\theta(s, a) R(s, a) da\Big]$$

For a stochastic policy $\pi_\theta(s, a)$, the gradient of objective function can be calculated by the policy gradient theorem [4]:

$$\begin{aligned}\nabla_\theta J(\theta) &= \mathbb{E}_s[\int_a \nabla_\theta \pi_\theta(s, a) Q^\pi(s, a) da] \\ &= \mathbb{E}_{s,a}[\nabla_\theta \log \pi_\theta(s, a) Q^\pi(s, a)]\end{aligned}$$

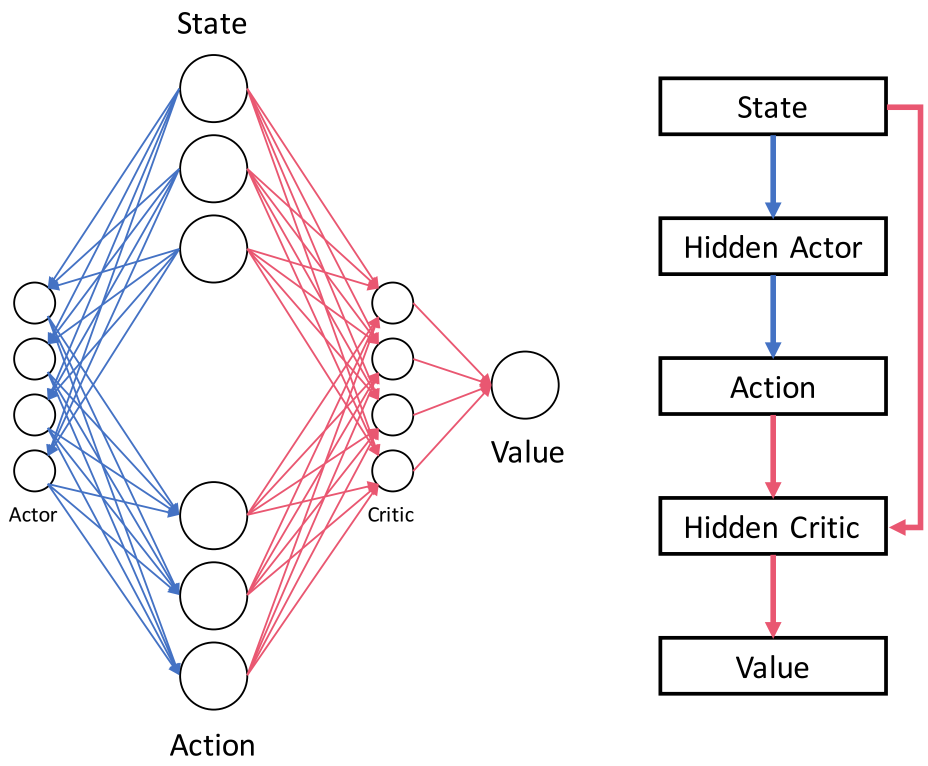

Figure 1.2: Actor-Critic architecture

Based on this fundamental theorem, the actor-critic architecture (as shown in figure 1.2) is widely used to represent the components inside policy gradient. An actor adjusts the parameter θ of stochastic policy $\pi_\theta(s, a)$ by stochastic gradient decent. Instead of the true unknown $Q^\pi(s, a)$, a critic will estimate the action-value function with $Q^\omega(s, a)$ using an appropriate value based algorithm. Therefore, the policy gradient above will lead to the following stochastic actor-critic algorithm:

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s, a) Q^\omega(s, a)$$

where $\pi_\theta(s, a)$ is the actor, $Q^\omega(s, a) \approx Q^\pi(s, a)$ is the critic.

Deterministic policy

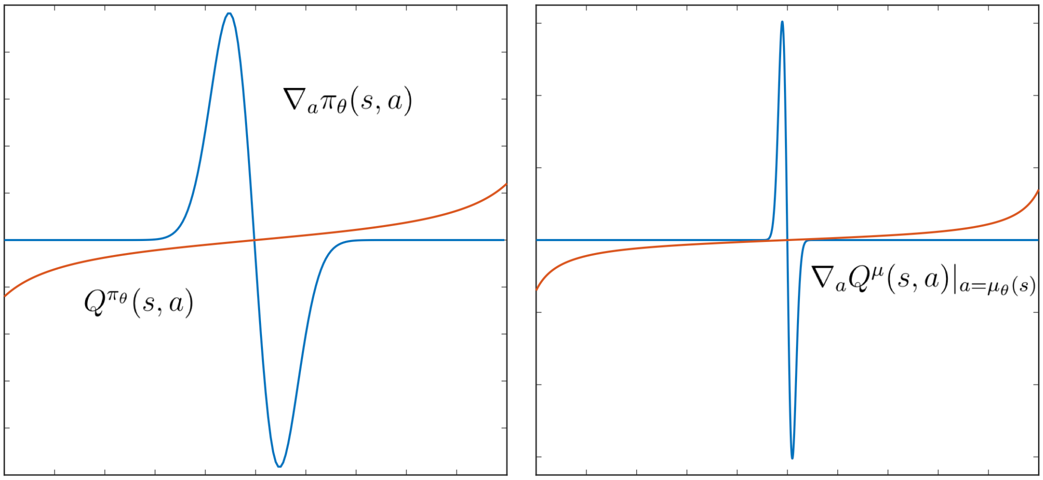

Figure 1.3: Illustration of policy gradient

One of the limitations of the stochastic algorithm is the variance of policy gradient. The policy $\pi_\theta(s,a)$ is typically bump-shaped distribution over actions. When we estimate $\nabla_\theta J(\theta) = \mathbb{E}_s[\int_a \nabla_\theta \pi_\theta(s,a) Q^\pi(s,a)da]$ by integrating product, the gradient of policy $\nabla_a \pi_\theta(s,a)$ will first go up and then go down as shown in the left part of figure 1.3. As a result, the variance of policy gradient estimate will approach infinite as policy approach deterministic. It is natural and intuitive to use the deterministic mean $\nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}$ to calculate the integrating product [5]. For a deterministic policy $a = \mu_\theta(s)$, the policy gradient is:

$$\nabla_\theta J(\theta) = \mathbb{E}_s\Big[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\Big]$$

leading to the following deterministic policy gradient update:

$$\Delta\theta = \nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}$$

which will update policy in the direction that most improves Q. In the view of divide and conquer, deterministic policy gradient simplify this task to two problems: first estimate Q, then move in the gradient of Q.

- Critic: approximate $Q^{\mu}(s, a)$ using differentiable function approximator with parameter.
- Actor: use deterministic policy gradient with respect to critic gradient $\nabla_a Q^{\mu}(s, a)$

Chapter 2

Data preprocessing

2.1 Simulated data

2.1.1 Coagulation model

Blood coagulation process is mediated and controlled by a multitude of distinct coagulation-promoting and anticoagulative factors, ensuring an individual's viability. In order to simulate the coagulation process in humans, Wajima et al., have proposed a comprehensive mathematical model as shown in the figure 2.1 [6]. There are total 51 components in the coagulation model for describing the time courses of coagulation factors by extrinsic and intrinsic pathway activation, as well as by the in vitro (outside body) blood coagulation tests of activated partial thromboplastin time (aPTT). The model was also successfully applied to describe the effects of hemophilias (A and B) on aPTT. With these features, this model could serve as the environment of our task: given a state (patient parameters) and an action (heparin dosing), generate corresponding reward (patient response: aPTT) and following state (patient parameter).

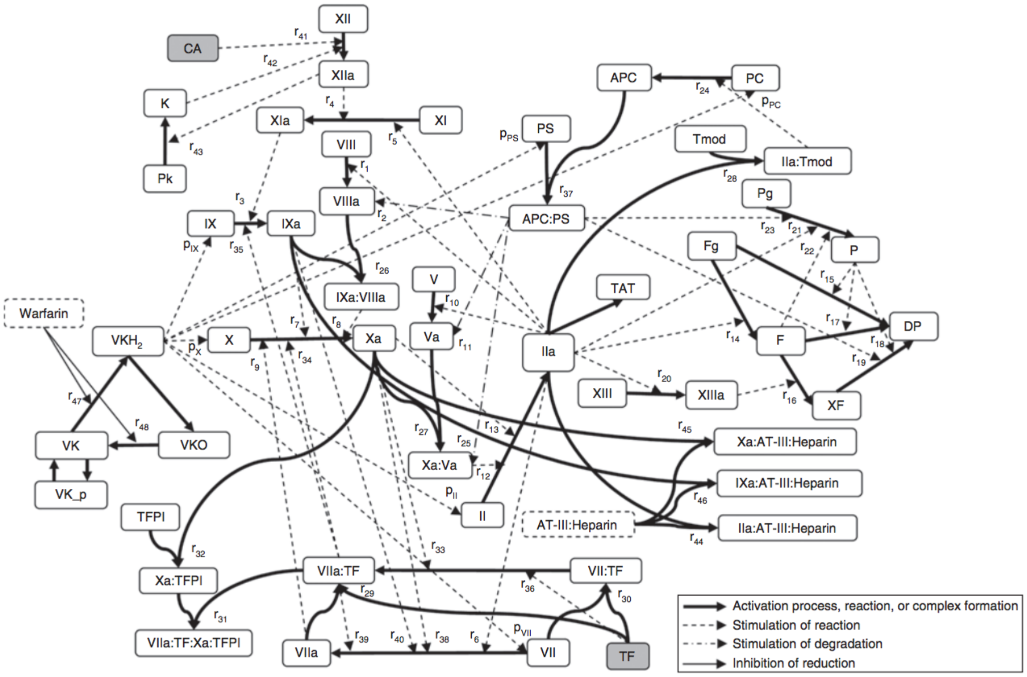

Figure 2.1: Coagulation network model (Source: Wajima et al.,2009)

2.1.2 Simulation process

In this work we build up a emulator based on the coagulation model, which generates all the data needed and is implemented in MATLAB using the MATLAB Ordinary Differential Equation (ODE) solver. Specifically, the simulation has two stages. The in-vivo (inside body) simulation is to run the ODE solver for a time span of 24 hours to simulate the patients physical conditions. Note that the simulation is broken down to 24 1-hour simulations since we would apply different medication dosing for different hour. The beginning state of a certain hour is based on the simulation outcome from the preceding hour. This stage of simulation is sequential. The in-vitro (outside body) simulation happens at each measurement time with a time span of 2 minutes. It runs the ODE solver on the information simulated in the in-vivo simulation and output the aPTT value. Note that this process could be parallelized since these simulations throughout 24 hours are not dependent upon each other. This parallelization would greatly reduce the simulation time.

As a result, the emulator is able to receive state and action and yield next state and reward in the following forms:

- state: including hemophilia factor, hemophilia value, aPTT and latent state inside emulator: APC, activated protein C; AT-III, antithrombin-III; CA, activator for the contact system; DP, degradation product; F, fibrin; Fg, fibrinogen; II, prothrombin; IIa, thrombin; K, kallikrein; P, plasmin; PC, protein C; Pg, plasminogen; Pk, prekallikrein; PS, protein S; TAT, thrombinantithrombin complex; TF, tissue factor; TFPI, tissue factor pathway inhibitor; Tmod, thrombomodulin; VK, vitamin K; VKH2, vitamin K hydroquinone; VKO, vitamin K epoxide; XF, cross-linked fibrin.

- reward: simulated aPTT. According to the therapeutic window of Heparin, the normal (good) aPTT value is approximately within the range of 60-100, and the bad aPTT value is outside this range. We applied a scaling function proposed in [3] to scale the reward to a value between -1 and 1:

$$r_t = \frac{2}{1 + e^{-(aPTT_t - 60)}} - \frac{2}{1 + e^{-(aPTT_t - 100)}} - 1$$

- action: heparin dosage.

2.2 MIMIC-II

2.2.1 MIMIC-II data description

The MIMIC-II (Multiparameter Intelligent Monitoring in Intensive Care-II) database [7] is a publicly available database released by the Laboratory of Computational Physiology at MIT. Such databases are often privately owned, have highly restricted access or require fees. Access to MIMIC-II has been provided to more than 1,000 individuals or institutions and it has already facilitated many valuable results [8]. The MIMIC-II database contains high resolution data of 25,328 intensive care unit stays at the Beth Israel Deaconess Medical Center in Boston, MA and combined with structured clinical data. All data are collected from adult patients admitted between 2001 and 2007. Figure 2.2 illustrated how the data in MIMIC-II database was collected. It is also a standard and general process can be used to collect clinical data.

There are 4470 patients in MIMIC-II database who received a heparin intravenous infusion during their ICU stay. Activated partial thromboplastin time (aPTT) is also used in MIMIC-II to indicate the heparin status of patients. After excluding those patients with heparin infused but without any aPTT records, in the end, we extracted 2598 patients with following laboratory features:

- State: arterial carbon dioxide level (CO_2), heart rate (HR), albumin, diastolic arterial blood pressure (DBP), systolic arterial blood pressure (SBP), bilirubin, creatinine, Glasgow Coma Score (GCS), hematocrit, hemoglobin, International normalized ratio of prothrombin (INR), blood PH, platelet count, prothrombin time, respiration rate, oxygen saturation in arterial blood (SaO_2), Sequential Organ Failure Assessment (SOFA) scores, Oxygen saturation in arterial blood measured by pulse

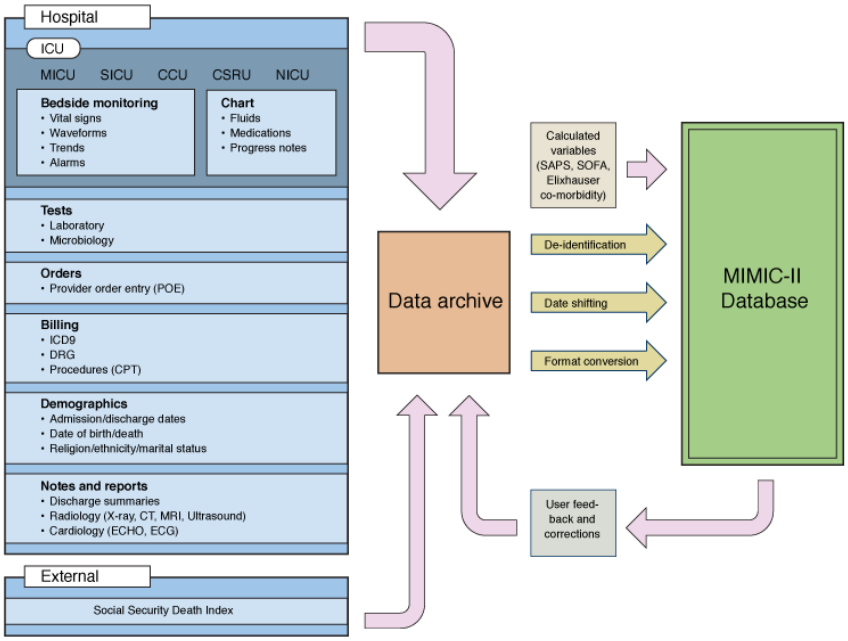

Figure 2.2: MIMIC-II (Source: MIT Critical Data)

oximeter (SpO_2), temperature, troponin, urea, white blood cell count (WBC), lagged aPTT and heparin dose measurements over the three hours prior to the selected time (t-1h : t-3h). Additionally, we also collected the following covariance features: gender, age, weight, different ICU unit, different ethnicity, pulmonary embolism and overall SOFA score.

- Reward: aPTT. The safe region of aPTT used in MIMIC-II is still 60-100. After applying the same scaling function in the previous section, the aPTT is scaled to a value between -1 and 1.

- Action: heparin dosage.

2.2.2 MIMIC-II data preprocessing

After extracting the raw clinical data, there are several preprocessing steps:

1) Find data segments: Patients usually spend a relatively long time in the ICU, The existence of heparin and aPTT records is not contiguous from the start to the end, it might only be a small subset of original data. We need to locate the right data segments inside the whole trajectory of patient history for the later training process. The reasonable time window should at least start from the time when first heparin was detected to the time when last aPTT was measured. Finally, we still need to make sure the initial heparin is no earlier than 6 hours than the first measured aPTT to ensure full effect.

2) Remove outlier: In order to avoid the potential influence of noise data, after first step we will remove the outliers. We use the quantile for the cumulative probabilities 0.995 and 0.005 as upper and lower threshold to determine whether a data should be classified as outliers.

3) Impute missing values: A phenomenon that we often observed in the original data is missing values: NaN. There are three different ways to handle with different missing values. For missing heparin values, if it appears right after an aPTT measurement, it is likely to be the clinicians' decision to stop the medication dosing. In this case the heparin will be set to 0. Otherwise we applied the sample and hold interpolation which might be the most practical form of interpolation in most clinical settings; For missing aPTT values, we will use sample and hold method to create a new version of this measurement; For missing observation values, if all the values of this features are NaN, we will impute them with mean or most common value of the whole data set depending on what type the feature is. Otherwise, if there are values appears in previous hours, we could use the sample and hold method again.

4) Normalize features: The range of values of raw data varies widely, objective functions will not work properly without normalization. For example, if one of the features has a broad range of values (weight: 168 compared with bilirubin: 0.8), the inner products of units in neural network will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final result. Specifically, the method we utilized is zero-mean normalization: firstly calculate the mean and standard deviation of each feature in the training dataset, then perform normalization on training dataset and testing dataset by subtracting by mean and dividing by standard deviation.

2.3 Emory ICU data

2.3.1 Emory ICU data description

The third data we used in experiment is the ICU data collected from Emory University Hospital in Atlanta, GA. One of the most important parts in the clinical data archive process is to protect the privacy of patients. After de-identification, date shifting and format conversion, all clinical data from Emory ICU has been fully transformed in a Health Insurance Portability and Accountability Act (HIPAA) compliant manner. The Emory ICU data consists of more than 30,000 patient admitted between 2013 and 2015. It is also a comprehensive database including demographic information (gender, ethnicity), laboratory testing result (microbiology results, blood counts) and so on. In order to determine the patient's certain medical condition and analyze our experiment results, we also extracted the billing information from IMBills in the format of International Classification of Diseases-9 code (ICD-9 code), which is used to report medical diagnoses and procedures in U.S. health care settings.

There are several approaches of heparin injection in Emory ICU data, sometimes heparin is given in the form of one time shot for other medical purposes. We only extract those heparin be injected intravenously (into a vein) and last for a certain time (no less than 8 hours and no more than 20 days). After filtering out unrelated and incomplete data on the original dataset, we obtained 2310 heparin patients with following comprehensive information:

- State: mean arterial pressure (MAP), heart rate (HR), oxygen saturation (SO_2), systolic arterial blood pressure (SBP), diastolic arterial blood pressure (DBP), respiratory rate, temperature, Glasgow Coma Scale (GCS), partial pressure of oxygen in arterial blood (PaO_2), fraction of inspired oxygen (FiO_2), potential of hydrogen of arterial blood gas (pH of ABG), partial pressure of carbon dioxide of arterial blood gas (pCO_2 of ABG), bicarbonate of arterial blood gas (HCO_3 of ABG), the loss of buffer base to neutralize acid of arterial blood gas (Base Excess of ABG), oxygen saturation in arterial blood arterial blood gas (SaO_2 of ABG), white blood cell count (WBC), hemoglobin, hematocrit, creatinine, bilirubin, direct bilirubin (DBil), platelet count, International normalized ratio of prothrombin (INR), partial thromboplastin time (PTT), aspartate aminotransferase (AST), alkaline phosphatase (ALP), Lactate, Glucose, Potassium, Calcium, blood urea nitrogen (BUN), Phosphorus, Magnesium, Chloride, brain natriuretic peptide (BNP), Troponin, Fibrinogen, C-reactive protein (CRP), Sedimentation Rate, Ammonia, Coagulatory sequential organ failure assessment scores (Coagulatory SOFA), lagged heparin level and heparin dose measurements over the three hours prior to the selected time (t-1h : t-3h). Additionally, we also collected the following covariance features: gender, age, weight, different ICU unit, different ethnicity, surgery record and wound class.

- Reward: heparin level. The reference of coagulation in Emory ICU data is not aPTT anymore. Clinicians in Emory hospital refers to heparin level (typically from 0 to 1) to evaluate the therapeutic effect. There are two standards in Emory heparin protocol, according to different standard, the value of heparin level could be scaled to different rewards.

- Action: heparin dosage.
- IMBills: By matching the ICD-9 medical diagnosis codes with corresponding complications, we managed to break downs the ICD-9 codes into 4 categories: **1)** History of Clotting; **2)** History of Bleeding; **3)** Clotting Care (patient received care for clotting related complications); **4)** Bleeding Care (patient received care for bleeding related complications). With these extra information, we could classify patients to the appropriate standard more accurately and analyze the consequence of clinicians' dosing more comprehensively.

2.3.2 Emory ICU data preprocessing

The preprocessing process of Emory ICU data is almost the same with the procedures implemented on MIMIC-II database. Firstly, find the right data segments; Secondly, remove the abnormal outlier; Thirdly, impute missing values of some features; Finally, normalize all features. The difference between the two real medical dataset is we have more information about patients in the Emory ICU data. Such as surgery start time stamps, surgery stop time stamps and transfusion records. In the data imputation phase, we use these surgery time stamps to interpret missing heparin dosing more accurately instead of naive sample and hold. In the outlier removing phase, we use transfusion information to help determine real data noise. Besides, Emory ICU data also have advantage in time accuracy. Each heparin dosing in the two datasets has corresponding time stamps, we use these time stamps to form the medical trajectories. Some of the time stamps in MIMIC-II dataset are "order time", which means clinicians ordered the heparin at that time but the heparin might be injected after one hour. In the meanwhile, all the time stamps in Emory ICU data are “service time”, which means the exact heparin injection time.

2.3.3 Emory heparin protocol

Besides the ICU data from Emory University Hospital, we have the weight based Emory heparin infusion protocols to help analysis our medication dosing. The protocol we have is a summary of 4 active protocols used in Emory Hospital from 2013 to 2015.

First of all, there are several general guidelines to help with heparin infusion and standard classification:

1) Heparin infusion is 25,000 units/250mL = 100 units/mL.

2) Weight is actual body weight (kg) on admission.

3) Nursing instructed to call provider if:

a. Platelet levels drop >= 50%.

b. Any sign/symptom of bleeding.

c. HIT positive.

4) Recommendations:

a. Low Standard: Age $>$ 75, ischemic stroke, high bleeding risk, AMI, unstable angina.

b. High Standard: DVT, PE, mechanical heart valve replacement, high risk for clot formation.

According to different standard, this protocol is divided into 2 parts which shown in the table 2.1 and table 2.2:

Heparin Level	Actions	Next Heparin Level
<=0.14	Repeat bolus with 40 units/kg And inc infusion by 3 units/kg/hr	6hrs after rate change
0.15-0.29	Inc infusion by 2 units/kg/hr	6hrs after rate change
0.3-0.5	No Change	Next AM
0.51-0.7	Dec infusion by 1 units/kg/hr	6hrs after rate change
0.71-0.89	Dec infusion by 2 units/kg/hr	6hrs after rate change
>=0.9	Stop infusion for 2 hr then resume And dec infusion by 3 units/kg/hr	2hrs after infusion resumed
Initial rate: 15 units/kg/hr; Max initial rate: 10 ml/hr		

Table 2.1: Low standard protocol

Heparin Level	Actions	Next Heparin Level
<=0.14	Repeat bolus with 40 units/kg And inc infusion by 3 units/kg/hr	6hrs after rate change
0.15-0.29	Inc infusion by 2 units/kg/hr	6hrs after rate change
0.3-0.49	Inc infusion by 1 unit/kg/hr	6hrs after rate change
0.5-0.7	No Change	Next AM
0.71-0.79	Dec infusion by 2 units/kg/hr	6hrs after rate change
0.8-0.89	Stop infusion for 1 hr then resume And dec infusion by 2 units/kg/hr	6hrs after infusion resumed
>=0.9	Stop infusion for 2 hr then resume And dec infusion by 3 units/kg/hr	2hrs after infusion resumed
Initial rate: 18 units/kg/hr; Max initial rate: 24 ml/hr		

Table 2.2: High standard protocol

Chapter 3

Experiment procedures

3.1 Preliminary

The framework we present is based on Deep deterministic policy gradient (DDPG) by Lillicrap et al., [9]. DDPG is a novel technique designed for the continuous action domain. This algorithm combines Deterministic Policy Gradient(DPG) [5] and Deep Q-Networks(DQN) [10]. First of all, In order to solve our task in continuous action domain, an action based policy gradient is necessary. The actor-critic model in DPG not only provide such action based network, but also offer a value based Q-learning network to increase the training efficiency. The parameter is updated in each episode by using policy gradient only, with the Q value estimated by critic model, updating is proceeded in each step. In addition, naive DPG using Actor-Critic is prove to be unstable. DDPG is able to learn a optimal action from continuous domain stably and quickly with the two innovations in DQN: Replay buffer and target network. Replay buffer is a technique that store the experiences of the agent during training, and then randomly sample experiences to use for learning. The replay mechanism sought to break up the temporal correlations within different training episodes; Target Network is a technique which create a copy of the actor and critic networks, in the training process, current parameters are different with those used to generate samples. Target network can regularizes the learning algorithm and increases stability. Finally, in order to explore the environment more efficiently, the Ornstein-Uhlenbeck (OU) [11] process is used in agent's exploration phase. It is simply a stochastic process which has mean-reverting properties. With OU process, the agent could reduce the risk of being stuck in a local minimum. Overall, the following Figure 3.1 illustrate the structure of DDPG algorithm in the setting of our medication dosing task.

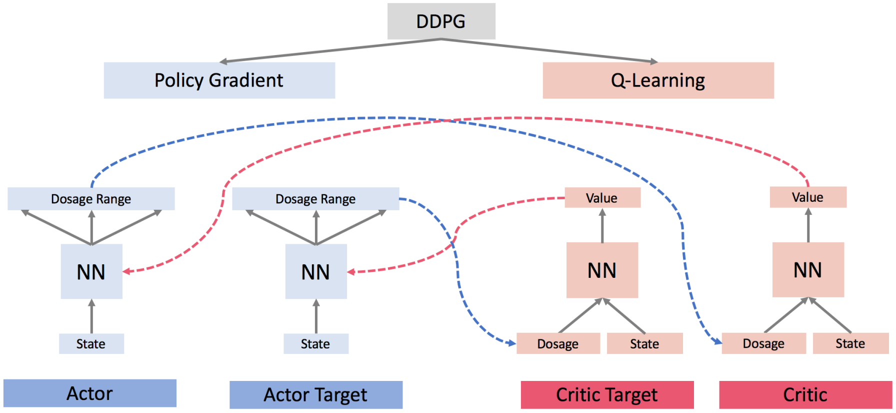

Figure 3.1: Structure of DDPG

3.2 Proposed method

One challenge of reinforcement learning application in medication domain is the problem of exploration. In terms of the simulated patients, RL agent is able to explore environment thoroughly by execute various dosages. The learning process is relatively straight forward. However, when it comes to the real world scenario in ICU, randomly exploration over action domain is not realistic in the training process. The RL algorithm should be used for policy evaluation and not for policy improvement. Instead of generating new episode by interacting with environment, the feasible way to implement RL algorithm is analyzing real episode from retrospective clinical data. The method we presented to handle with real data task is a clinician-in-the-loop framework. Which means in the sequential decision making process, the agent will predict a action according to the current state, but the executed action is determined by clinicians. Meanwhile, the agent will learn from this executed action, state and patient's response. Figure 3.2 illustrate this clinician-in-the-loop framework in terms of continuous action task.

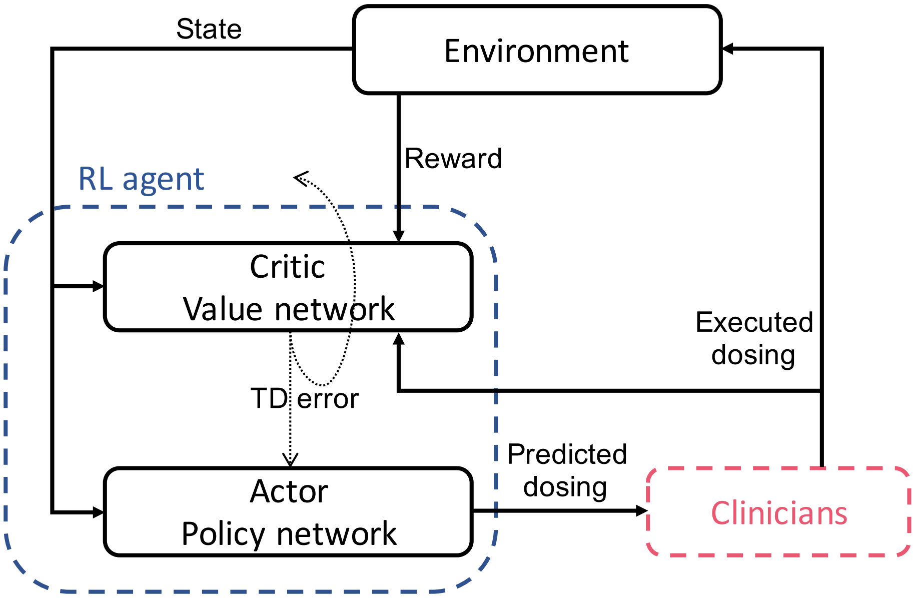

Figure 3.2: Clinician-in-the-loop framework

More specifically, our algorithm based on DDPG for clinician-in-the-loop framework is as follows:

```
Algorithm 1
Require: Clinical data including well defined states, actions and rewards
  Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with
  weights $\theta^Q$ and $\theta^\mu$. Initialize target network $Q'$ and $\mu'$ with same weights
  $\theta^{Q'}$ and $\theta^{\mu'}$. Initialize replay buffer $R$.
for episode = 1, M (all patients in database) do
    Receive initial observation state $s_1$ from patient's initial status
    for t = 1, T (trajectory of each patient) do
        Predict action $a_t^R = \mu(s_t|\theta^\mu)$ according to the current policy
        Execute clinician action $a_t^C$ and observe reward $r_t$ and new state $s_{t+1}$
        Store transition $(s_t, a_t^C, r_t, s_{t+1})$ in $R$
        Sample a random minibatch of N transitions $(s_i, a_i^C, r_i, s_{i+1})$ from R
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta\mu')|\theta^{Q'})$
        Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
        Update the actor policy using the sampled policy gradient:
            $$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$
        Update the target networks:
            $$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
            $$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$
    end for
end for
```

3.3 Experimental details

3.3.1 simulated data experiment

General settings:

Training parameters: reward discount γ=0.99, replay buffer size=10^6, replay buffer batch size=64, soft target network updates τ=0.001, in OU process μ=0, θ=0.15 and σ=0.2.

Actor network structure: the input layer is 3 features from patients; following by 2 fully connected hidden layers with 400 and 300 neurons respectively, each neuron uses ReLU as activation function; the output layer has 1 neurons producing real action value directly, which uses tanh as activation function to scale the output value from -1 to 1. In the learning process, we used Adam optimizer with learning rate 10^{-4} to update the parameters of the network.

Critic network structure: it shares almost the same structure as Actor network. The two differences lie in second layer and output layer. In second hidden layer, Critic network add 1 action neurons from the Actor network. In output layer, the Critic network only has 1 neuron with no activation function to predict Q-value. In the learning process, we used Adam optimizer with learning rate 10^{-3} to update the parameters of the network.

Besides, most of the neural networks in our experiments were build with TensorFlow[12].

Baseline settings:

1) DQN

Our first baseline algorithm is the DQN algorithm which discretize the continuous action domain into several discrete actions. The general steps of DQN are as follows: 1. Set up the simulated coagulation model with hemophilias and obtain the initial state; 2. Feed the neural network with current state to calculate the Q-value of different action; 3. Execute the action with maximum Q-value (or randomly with exploration probability) in the simulated model and obtain next state and reward; 4. Store transition composed by state, action, reward and next state into replay buffer; 5. Sample mini batch transitions from replay buffer to calculate gradient of loss (mean square error between predicted Q-value and true Q-value calculated by Bellman equation); 6. Move the weights of Neural Networks in the direction of the gradient; 7. Repeat this process until the weights are tuned to a satisfied point.

Training parameters: replay buffer size=5000, update batch size=16, Adam optimizer with learning rate=0.001, reward discount γ=0.99, probability of exploration is decreased with iteration and probability of exploitation is increased with iteration.

Neural network structure: 3 neurons in the input layer (represent the aPTT, hemophilia factor and hemophilia value); 64 neurons in the hidden layer(using ReLU as activation function); 9 neurons in the output layer (represent discrete dosages: 0, 30, 100, 200, 300, 500, 1000, 1500, 2000).

2) DPG (minibatch NFQCA)

We implemented the DPG algorithm using minibatch NFQCA (Neural Fitted Q Iteration with Continuous Actions [13]) as the second baseline algorithm. The general steps of minibatch NFQCA are as follows: 1. Initialize Actor and Critic network with small random numbers; 2. Set up the simulated coagulation model with hemophilias and obtain the initial state; 3. Feed the state into Actor and Critic, obtain predicted action from Actor and calculated Q-value from critic; 4. Execute the predicted action from Actor (or randomly with exploration probability) in the simulated model and obtain next state and reward; 5. Calculate the loss between Q-value predicted by critic and Q-value calculated by Bellman equation; 6. Obtain gradient of Critic by minimizing this loss, obtain gradient of Actor by deterministic policy gradient equation; 7. Sum the gradient over minibatch and take the average, update the weights of networks in the direction of the gradient; 8.

Repeat this process until the weights are tuned to a satisfied point.

Training parameters: batch size=16, reward discount γ=0.99, probability of exploration is decreased with iteration and probability of exploitation is increased with iteration.

Actor network structure: the input layer is 3 features from patients; following by 2 fully connected hidden layers with 400 and 300 neurons respectively, each neuron uses ReLU as activation function; the output layer has 1 neurons producing real action value directly, which uses tanh as activation function to scale the output value from -1 to 1. In the learning process, we used Adam optimizer with learning rate 10^{-4} to update the parameters of the network.

Critic network structure: it shares almost the same structure as Actor network. The two differences lie in second layer and output layer. In second hidden layer, Critic network add 1 action neurons from the Actor network. In output layer, the Critic network only has 1 neuron with no activation function to predict Q-value. In the learning process, we used Adam optimizer with learning rate 10^{-3} to update the parameters of the network.

3.3.2 real data experiment

Reward settings:

When learning from the real clinical data, one issue is how to define the sparse and delayed reward. For MIMIC-II database, the important parameter that clinicians will refer to is aPTT. In order to transfer aPTT value to reward ranged from -1 to 1, We applied the following scaling function:

$$r_t = \frac{2}{1+e^{-(aPTT_t-60)}} - \frac{2}{1+e^{-(aPTT_t-100)}} - 1$$

For Emory ICU data, the dosing of heparin is based on the value of heparin level. As shown in table 2.1 and 2.2, same heparin level might corresponds to different actions in the 2 separate heparin protocols of different standard. However, the standard adopted by clinicians is not recorded in the ICU data. In order to get the consistent reward among all patients, we need to determine the standard first. It is important to note that we can not use the standard recommendations in protocol guidelines directly to determine which standard is used by clinicians. The recommendation can only be used as a suggestion, ultimately, it is up to clinicians' own knowledge and judgment to choose the standard. For example, a patient has pulmonary embolism (PE) in the medical history. According to the recommendations this patient should be classified as the high standard patient. However, if the patient just had surgery, the clinician might decide to place him on the low standard protocol to minimize the risk of bleeding. As a conclusion, although we have the information from demographics and complication history from IMBills, we still need to determine the standard in a more precise way. The specific procedures are: 1) for each patients in training set, we extract the weight and heparin level in the data preprocessing phase; 2) use these two features to calculate corresponding high standard dosage and low standard dosage; 3) compare the initial rate between real infused dosage and the two predicted dosage; 4) calculate mean square error based on the difference between real dosage and predicted dosages; 5) choose the most similar standard and use it to calculate reward with following two scaling functions:

$$r\_low_t = \frac{2}{1+e^{-10(HL_t-0.3)}} - \frac{2}{1+e^{-10(HL_t-0.5)}} - 0.5$$

$$r\_high_t = \frac{2}{1+e^{-10(HL_t-0.5)}} - \frac{2}{1+e^{-10(HL_t-0.7)}} - 0.5$$

General settings:

Training parameters: reward discount γ=0.99, replay buffer size=10^6, replay buffer batch size=64, soft target network updates τ=0.001, in OU process μ=0, θ=0.15 and σ=0.2.

Actor network structure: the input layer is composed of clinical states from patients (30 features for MIMIC-II, 49 features for Emory ICU); following by

2 fully connected hidden layers with 400 and 300 neurons respectively, each neuron uses ReLU as activation function; the output layer has 1 neurons producing real action value directly, which uses tanh as activation function to scale the output value from -1 to 1. In the learning process, we used Adam optimizer with learning rate 10^{-4} to update the parameters of the network.

Critic network structure: it shares almost the same structure as Actor network. The two differences lie in second layer and output layer. In second hidden layer, Critic network add 1 action neurons from the Actor network. In output layer, the Critic network only has 1 neuron with no activation function to predict Q-value. In the learning process, we used Adam optimizer with learning rate 10^{-3} to update the parameters of the network.

Chapter 4

Results and evaluation

4.1 Simulated data

4.1.1 Results

As illustrated in chapter 3, firstly, we implemented several RL algorithms on the simulated data. The simulation setting in the coagulation model is heparin patients with hemophilia. Initial states, hemophilia factors and hemophilia values of patients are randomly assigned before the RL training process. In the training process, each simulated patient received dosages for 24 hours and responded to these actions in the format of the aPTT values. In the coagulation model, aPTT value can be tested in a impractical interval time: every hour. The RL agent is able to learn from these interactions efficiently. In the testing process, instead of random exploration, the RL agent will exploit the current policy and choose the optimal dosing in 24 hours given the initial state. The safe and desirable range of aPTT is from 60 to 100. In order to judge the performance of RL agent, we can plot the aPTT and heparin dosing in the whole simulated trajectory to see whether the aPTTs are in the safe region. One of the baseline algorithm is DQN, which is a value based algorithm suitable for discrete action domains. Figure 4.1 shows some results of DQN algorithm in the testing process. As shown in the left, the aPTT exceed the safe region both in the upper bond 100 and lower bond 60, which might result in bleeding instance and clotting instance in the real medical practice. On the right hand side, the dosing generated by DQN agent is fluctuated from 0 to 2000 in the whole trajectory. In conclusion, the performance of DQN is not stable on heparin patients with hemophilia problem.

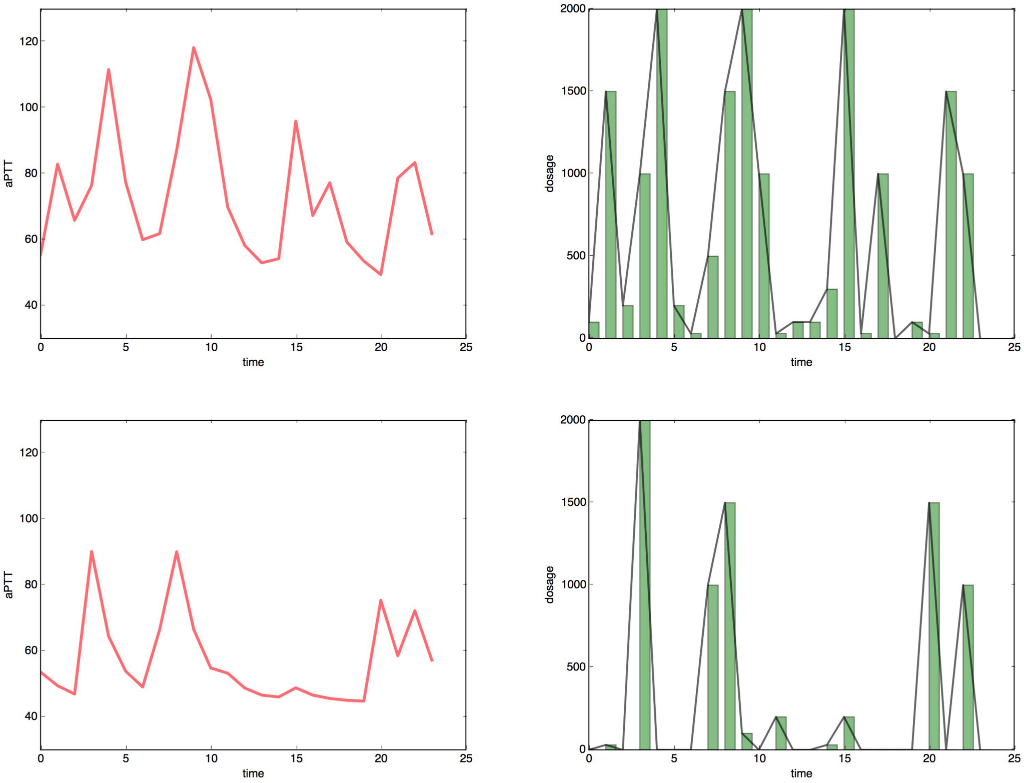

Figure 4.1: DQN dosing

Meanwhile, we also implemented the policy based DDPG algorithm to deal with this problem in continuous action. Some results are shown below.

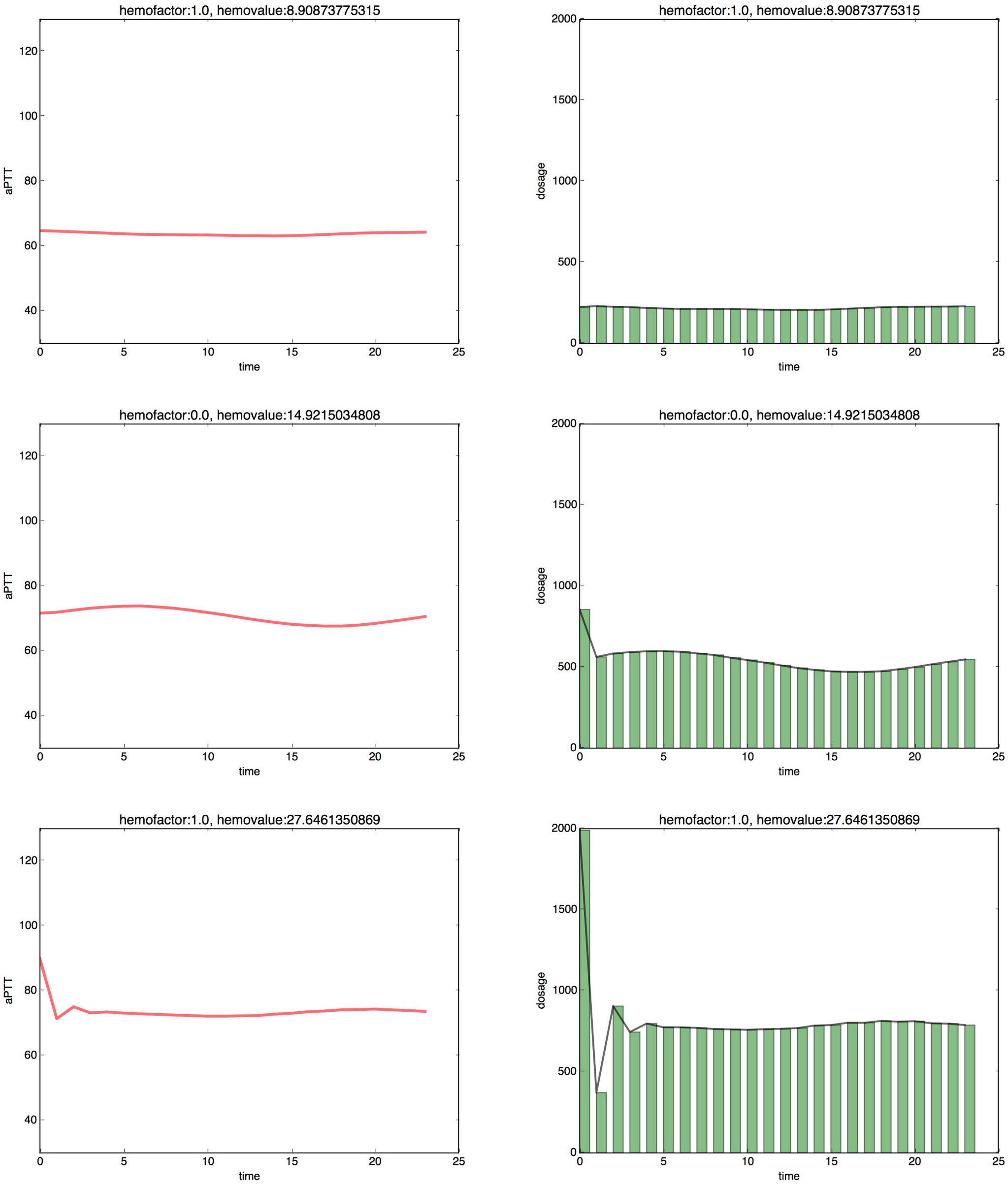

Figure 4.2: DDPG dosing

Figure 4.2 reveals that the DDPG agent is capable to manage various situations including different hemophilia factor and hemophilia value. The RL agent is able to figure out the optimal dosage from the start and maintain stable in the whole process. It is pretty difficult to locate a satisfied aPTT in the DQN results,by contrast, the curves of aPTT in figure 4.2 are much more smooth and always stay in the safe range.

4.1.2 Evaluation

The dosing results are just independent unit extracted from the whole dataset after all, the figures above are used for visualization. In order to evaluate the performance of different RL algorithms on heparin problems, we need to analyze the results more comprehensively. In the simulated experiment, evaluation is pretty straight forward. The simulated environment is able to respond to our RL agent. RL agent could explore the environment and learned the optimal action, in the end, we could evaluate the performance based on the averaged reward of each episode.

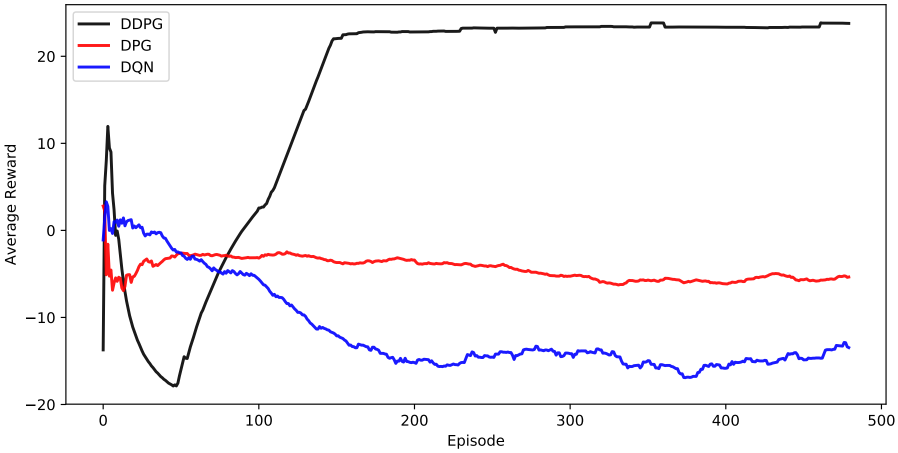

Figure 4.3: Averaged Reward of different algorithm

The setting time of our experiment is 24 hours per patient and the aPTT is tested every 1 hour. So the reward of each episode is ranged -24 to +24 (lowest to highest). As illustrated in figure 4.3, the DDPG algorithm is able to converge and achieve +23 at approximately 130 episodes. On the other hand, the DQN algorithm is still stuck in the reward less than -10. The DPG algorithm is slightly better than the DQN and its reward is likely to remain steady around -5. These two algorithms might need more episodes to have an increased trend on reward.

In addition, we realized that testing aPTT in every hour is not realistic in the real medical procedures. So we modified the testing time of the simulated model in the following ways: every 1 hour, every 4 hours, every 6 hours, every 8 hours and every 12 hours. In these ways, the aPTT measurement will be tested 24 times, 6 times, 4 times, 3 times and 2 times in the 24 hours treatment. The corresponding highest accumulated rewards in one episode will be: 24, 6, 4, 3 and 2. We normalized all the accumulated rewards to the range of [-1, 1] and plot them together in the figure 4.4.

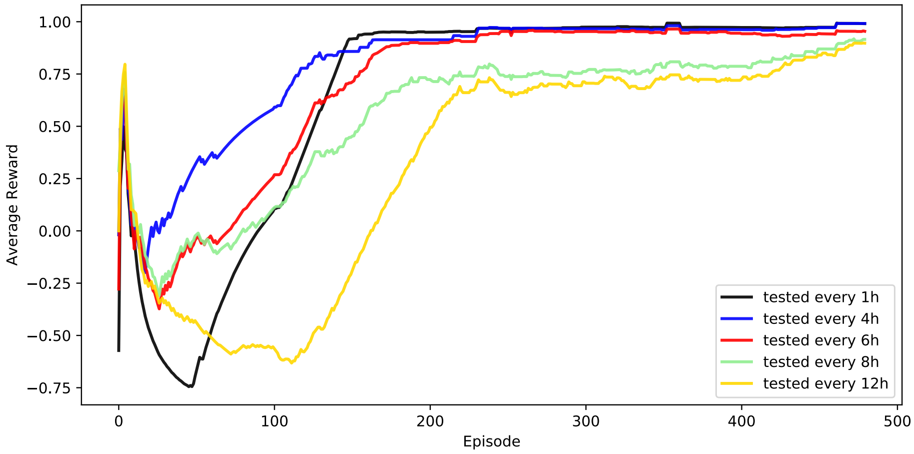

Figure 4.4: Averaged Reward of different time interval

As can be seen from the figure 4.4, after the exploration and learning phases, there have been steady increases in the 5 different accumulated rewards. Especially the curves of every 4 hours and every 6 hours, which almost peaked at the same value as the original every 1 hour method and even converged faster. Although this is just a simulated result, it could provide us an intuition that the aPTT value is relatively stable if heparin dosing is in the right range. It is feasible to measure the aPTT in a longer time interval.

4.2 MIMIC-II

4.2.1 Results

After testing the mechanism of various RL algorithms on the coagulation model, we implemented our proposed framework on the real clinical data. The first real dataset we used is MIMIC-II, in which patients have 29 laboratory test features and 7 covariance features (we used 30 features in the model). There are 2598 patients in the dataset, we used 80% of patients as training data and 20 % of patients as testing data. Compared with simulated model, the training process is actually analyzing real episodes instead of generating new episodes. Our method can be used for policy evaluation and not for policy optimization. Therefore, after the training process, our RL agent will learn from previous experiences and tend to predict a reasonable heparin dosing. Similarly, the reward measurement used in the MIMIC-II is also the aPTT. In order to have a better visualization on the results, we plot the recommended dosing predicted by RL agent, real dosing determined by clinicians and corresponding aPTT within the whole trajectory of a patient in one figure.

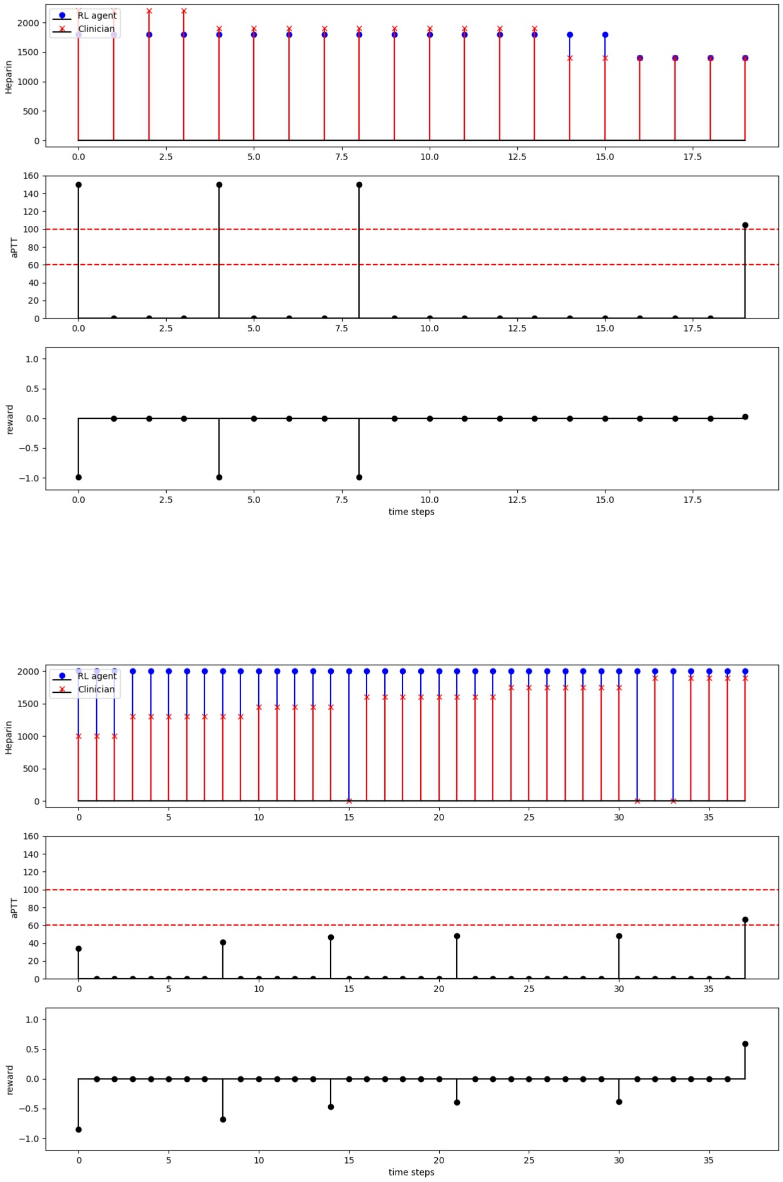

Figure 4.5: MIMIC-II dosing

There are two parts in figure 4.5 representing two different situations in the MIMIC-II dataset. In each part, the red bar in heparin sub figure is the clinicians' dosing and the blue bar is the RL agent's dosing. We also plot the safe range from 60 tp 100 in the aPTT sub figure. Specifically, the upper part shows the situation when aPTTs were too high in the whole trajectory, in which case the patient might got overdosed. By contrast, the RL agent tends to choose less dosing than clinicians. The lower part shows the situation when aPTTs were too low in the treatment process, in which case the patient might not received enough amount of heparin. As can be seen from the sub figure of heparin, the RL agent choose higher dosing from the start.

4.2.2 Evaluation

The heparin dosing results shown in the previous section are only independent cases. In order to evaluate the results more comprehensively, we need to find the relationship between our suggested dosing and actual rewards. As shown in figure 4.6, we divided all the patients into 5 classes based on the averaged distance between suggestion and real dosing.

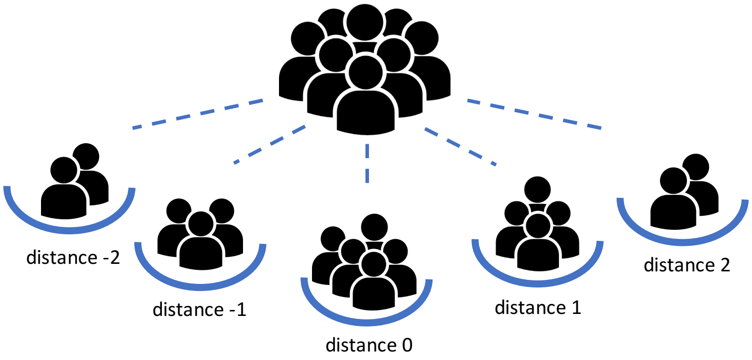

Figure 4.6: Patients classification

In MIMIC-II dataset, $Distance = \mathbb{E}_t[Recommendations - Clinicians]$. More specifically, distance -2 is defined as $(-\infty, -1200]$, distance -1 is defined as $(-1200, -400]$, distance 0 is defined as $(-400, 400]$, distance 1 is defined as $(400, 1200]$, distance 2 is defined as $(1200, \infty)$. After the classification, we evaluated the results on several measurement. The first one is the real reward scaled from aPTT value. For each patient, we calculate the averaged reward among the treatment. Based on these data, we obtain the mean and standard deviation of rewards in each class and plot them together.

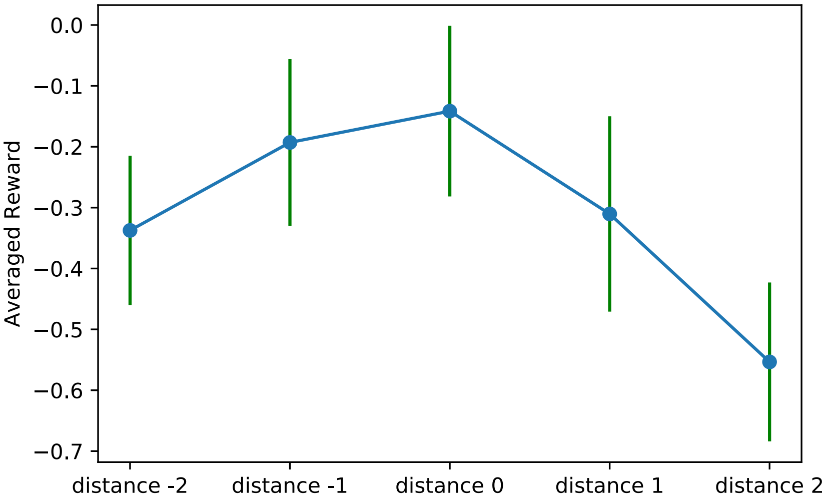

Figure 4.7: Reward of each class (MIMIC-II)

According to figure 4.7, the averaged reward of distance 0 class is greater than the others. In addition, with the increase of absolute distance between recommendation dosing and clinicians dosing, there is a slight decline in the averaged reward. These phenomenons suggested that our RL agent is providing reasonable and useful recommendations in the MIMIC-II dataset. The closer from the recommendation, the higher reward will achieved. However, the relationship between distance and reward is not well supported. some might argue that those patients in distance 0 class are just healthier than other patients by coincidence or other factors. In order to prove the causality between distance and reward, we performed a multiple linear regression analysis to determine whether the distance is a significant variable and adjust other confounding variables. Multiple linear regression analysis is an extension of simple linear regression analysis, used to assess the association between two or more independent variables and a single continuous dependent variable. The multiple linear regression equation is as follows:

$$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + ... + b_pX_p$$

where $\hat{Y}$ here is the predicted value of reward, X_1 through X_p are distinct independent variables. The variables we used in this regression model are: absolute distance, gender, age, weight, PE and overall SOFA score. After standardizing these variables into zero mean and unit variance, we used the Statsmodels [14] and scikit-learn [15] package to get the ordinary least squares (OLS) regression results shown in table 4.1.

OLS Regression Results

	coef	std err	t	$P > \|t\|$	[95% Conf. Interval]
const	-0.3503	0.012	-29.524	0.000	[-0.374 , -0.327]
distance	-0.1235	0.012	-10.360	0.000	[-0.147 , -0.100]
gender	-0.0108	0.012	-0.885	0.376	[-0.035 , 0.013]
age	0.0018	0.013	0.142	0.887	[-0.023 , 0.026]
weight	-0.0166	0.013	-1.303	0.193	[-0.042 , 0.008]
PE	0.0093	0.012	0.775	0.438	[-0.014 , 0.033]
SOFA	-0.0090	0.012	-0.748	0.454	[-0.015 , -0.003]

Table 4.1: Regression on reward (MIMIC-II)

Hypothesis Test and P-values

Except the coefficient of each variable, there is an important feature called P-value in the regression summary. In general, Statsmodels calculates 95% confidence intervals for the model coefficients. Hypothesis testing is a method closely related to confidence intervals. The conventional hypothesis test are as follows:

- null hypothesis: There is no relationship between variable X_p and predicted value $\hat{Y}$ (and thus b_p equals zero)
- alternative hypothesis: There is a relationship between variable X_p and predicted value $\hat{Y}$ (and thus b_p is not equal to zero)

The testing process is to check whether the data supports rejecting the null hypothesis or not. Intuitively, we reject the null (and thus believe the alternative) if the 95% confidence interval does not include zero. Conversely, the p-value represents the probability that the coefficient is actually zero: If the 95% confidence interval includes zero, the p-value for that coefficient will be greater than 0.05. If the 95% confidence interval does not include zero, the p-value will be less than 0.05. Therefore, comparing the p-value with 0.05 is one method to determine whether the variable is a significant one.

Coefficients interpretation

It is noteworthy that the p-value of the distance variable is smaller than 0.05 in the regression summary. Thus we could reject the null hypothesis and interpret the relationship between distance and reward as follows: with the increase of absolute value of distance between recommendation dosing and clinicians dosing, the change on reward would be negative. Choosing the heparin dosing close to the recommendation would lead to higher reward. The other variables seem like fail to reject the hypothesis when the confidence intervals is 95%.

Meanwhile, we also analyze the relationship between dosing distance and the pulmonary embolism (PE) instance of patients. PE is a sudden blockage in a lung artery. The cause of this disease is usually a blood clot in the leg. In our heparin case, a patient might got the PE symptom due to under dosed of anticoagulant. Although it is widely accepted that PE is an often preventable cause of death, the incidence and case-fatality rates of acute PE are uncertain [16]. It would be helpful to explore the probability of PE instance. In the MIMIC-II dataset, PE record is either 0 (negative) or 1 (positive). There are only 229 PE instances in 2598 patients, so the averaged PE value in each class is approaching 0. We calculated the averaged PE value among the same 5 distance classes. The higher the PE value is, the more PE instances in this class. As shown in figure 4.8, the averaged value of PE in distance 2 class achieved the highest value among 5 classes. Whereas the PE value in distance 0 class is the lowest one.

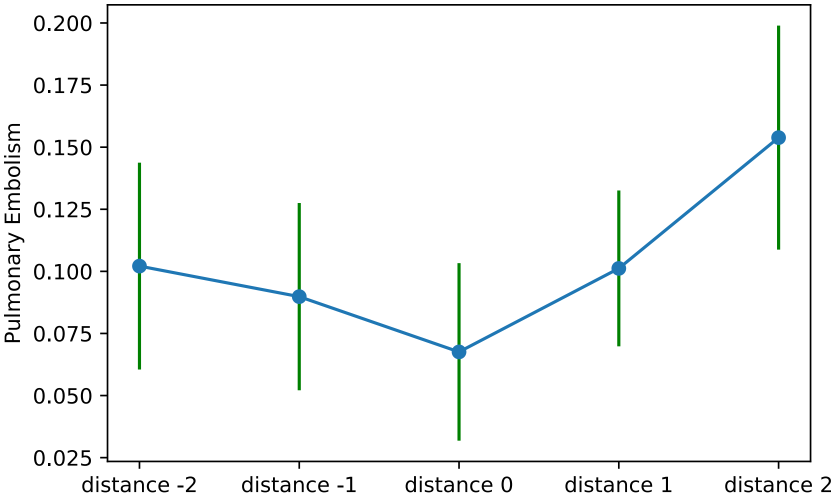

Figure 4.8: PE instance of each class (MIMIC-II)

$Distance = \mathbb{E}[Recommendations - Clinicians]$, thus the curve of PE values in figure 4.8 suggested that the probability of PE instance will increase if distance is getting larger, which means a patient received less heparin dosing than recommendations. In order to support this causal relationship, we extracted the same covariance variables and the distance to predict the PE instance. Since the PE label is a binary value, we used logistic regression model to analyze the significance of variables.

Logit Regression Results

	coef	std err	z	$P > \|z\|$	[95% Conf. Interval]
const	2.4008	0.073	32.893	0.000	[2.258 , 2.544]
distance	0.1396	0.067	2.075	0.038	[0.008 , 0.272]
gender	0.0729	0.071	1.028	0.304	[-0.066 , 0.212]
age	0.2204	0.070	3.138	0.002	[0.083 , 0.358]
weight	0.1807	0.062	2.911	0.004	[0.059 , 0.302]
SOFA	0.1438	0.074	1.938	0.053	[-0.002 , 0.289]

Table 4.2: Regression on PE (MIMIC-II)

From the table 4.2 we could find three significant variables. The first one is the distance between recommendation and actual dosing. The positive coefficient indicates that the probability of PE complication will increase with the distance. Therefore, the causality between distance and PE is supported by the coefficient and p-value. The other two variables are age and weight. These two variables both have positive coefficient. Elder people are more likely to get blood clot. It is also known that one of the risk factors of getting PE is being overweight [17]. Thus the probability of PE might increase with the weight and the age, which means these coefficients are reasonable.

4.3 Emory ICU data

4.3.1 Results

Besides the MIMIC-II dataset, we implemented our method on the Emory ICU data. There are 2310 patients records qualified to the experiment requirements. More specifically , each patient has 47 laboratory test features and 9 covariance features. Compared with MIMIC-II dataset, there are more features and the time stamps are more accurate. One notable difference compared with previous data is the referenced medical measurement in the Emory ICU data is not aPTT anymore. Heparin level is used for monitoring the therapeutic anticoagulation effects of heparin. Generally, the therapeutic range of these unfractionated heparin levels is 0.3 to 0.7 units/mL of anti-Xa activity. Inside the Emory ICU unit, the range is divided into 0.3 to 0.5 and 0.5 to 0.7, there are two standards inside the Emory heparin protocol to determine the final therapeutic range of heparin level.

We used 80% of patients as training data and 20% of patients as testing data. After the training phase, our RL agent is able to predict a reasonable heparin dosing given medical features of patients. In addition, we have the weight based heparin protocol to calculate the protocol dosing given the heparin level and weight of patients. To visualize the result more clearly, we plot the clinicians dosing (green stem with circle), protocol dosing (red stem with cross) and recommendation dosing (blue stem with cross) together in the sub figure of dosage to see the comparison. In the sub figure of heparin level, we plot the high standard therapeutic range (0.5-0.7) and the low standard therapeutic range (0.3-0.5) according to actual situation. Some results are shown in following figures:

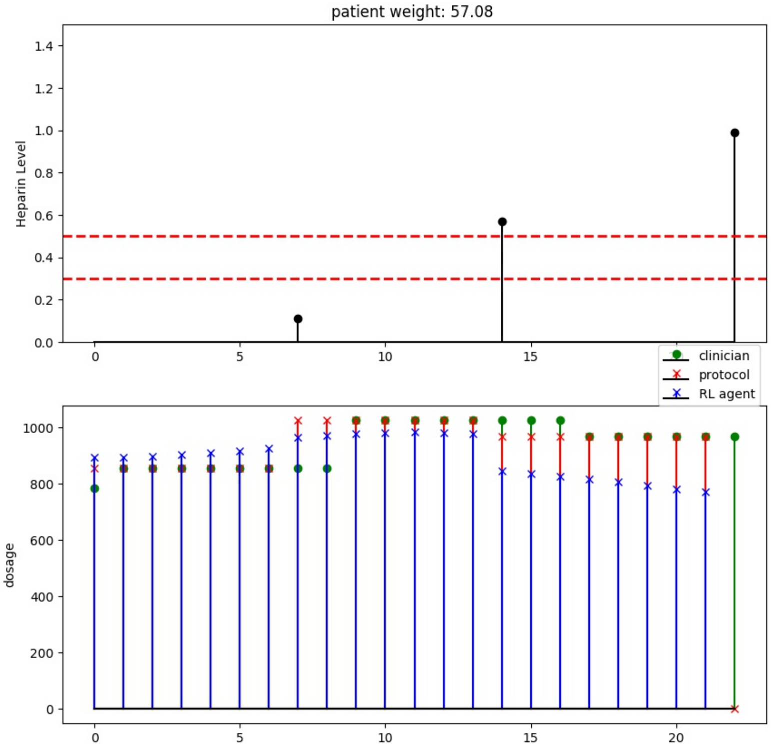

Figure 4.9: Emory low standard dosing (1)

Figure 4.9 shows the sequential dosing of of a low standard patient. The trend and values of protocol dosing are almost the same with the real dosing executed by clinicians. Compared with real dosing, the protocol responded promptly to heparin level. The protocol dosing will kept steady and only changed with respect to the presence of heparin level. The trend of recommendation dosing is slightly different. It increased gradually then decreased right after the second heparin level. The RL agent start to decrease the dosing before the third heparin level which is too high.

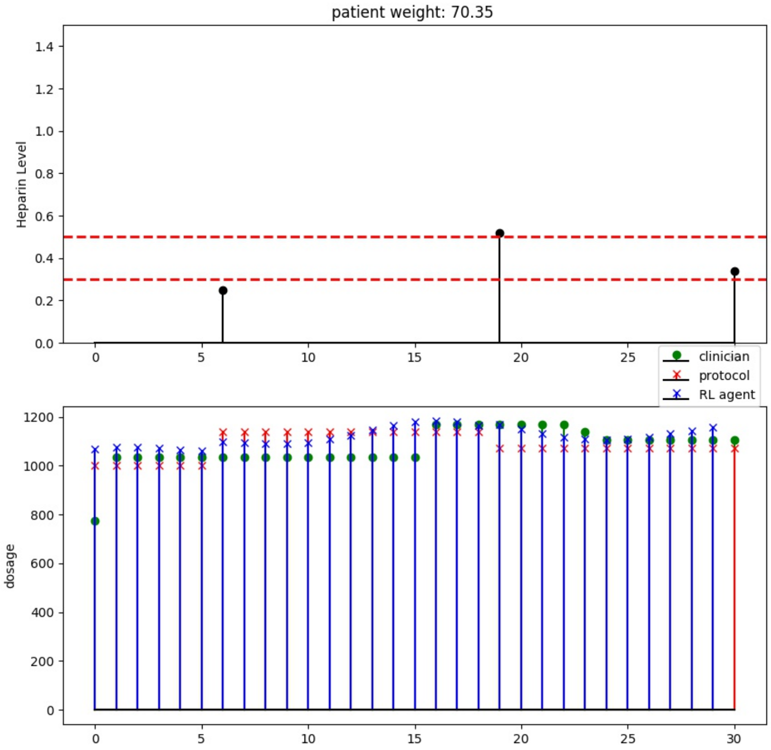

Figure 4.10: Emory low standard dosing (2)

Figure 4.10 is another example of low standard patients. Compared with previous patient, the three heparin level tested here are within or very close to the therapeutic range. The behaviors of RL agent are very similar with clinicians in this case. From these low standard examples we also found that the RL agent tends to choose a higher initial rate of heparin for low standard patients.

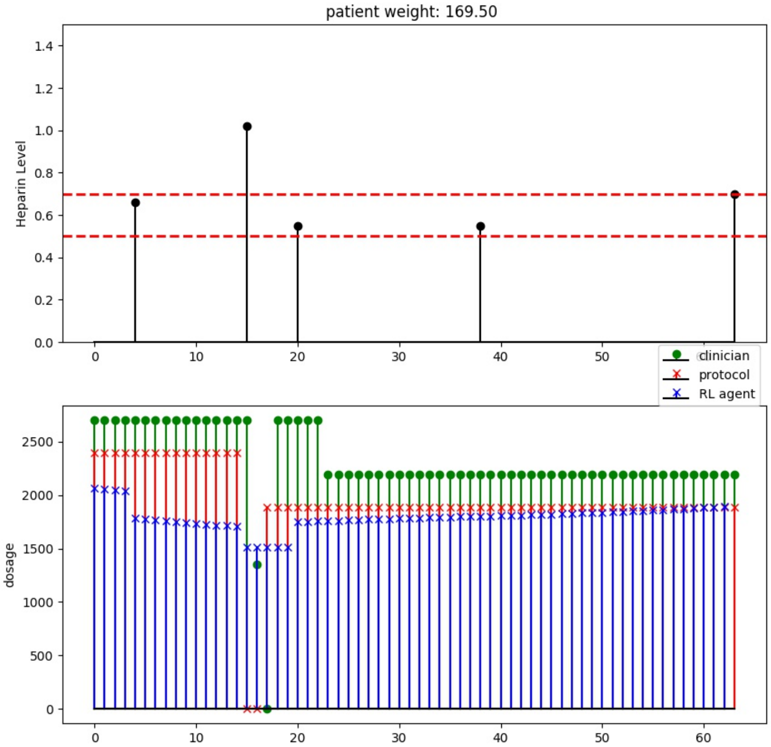

Figure 4.11: Emory high standard dosing (1)

Figure 4.11 shows the dosing summary of a high standard patient. The overall trends of the three dosing are similar. Clinicians and protocol will choose to stop the heparin infusion for 1 or 2 hours after the second heparin level which is too high. However the RL agent will just decreased the doing. It seems that the RL agent haven't learned the turn off strategy.

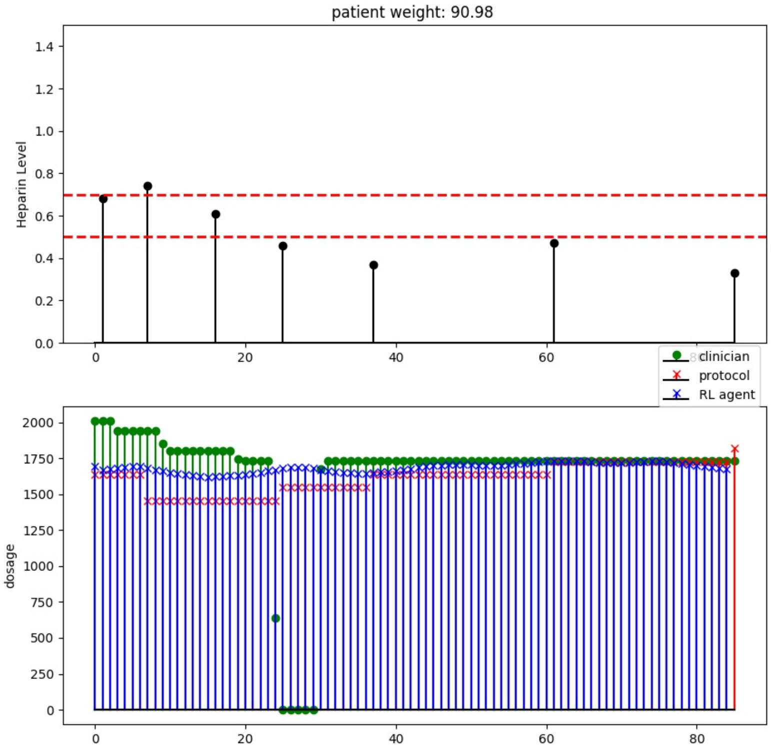

Figure 4.12: Emory high standard dosing (2)

Figure 4.12 is another example of high standard patients. This patient stayed in the ICU for 80 hours. Clinicians change the heparin dosing for several times then keep steady at a specific level. The RL agent managed to find that level of dosing from the start. Besides, we could also found that the RL agent tends to choose a lower initial rate of heparin for high standard patients.

4.3.2 Evaluation

Evaluation on averaged reward

We implemented the same evaluation method used in MIMIC-II. Firstly, classifying patients into 5 bins. $Distance = \mathbb{E}_t[Recommendations - Clinicians]$. We modified the distance range according to the distribution of patients. In the Emory ICU data, distance -2 is defined as $(-\infty, -750]$, distance -1 is defined as $(-750, -250]$, distance 0 is defined as $(-250, 250]$, distance 1 is defined as $(250, 750]$, distance 2 is defined as $(750, \infty)$. After the classification, we evaluated the results on several aspects. The first one is the real reward scaled from heparin level. For each patient, we calculate the averaged reward among the treatment. Based on these data, we obtain the mean and standard deviation of rewards in each class and plot them together.

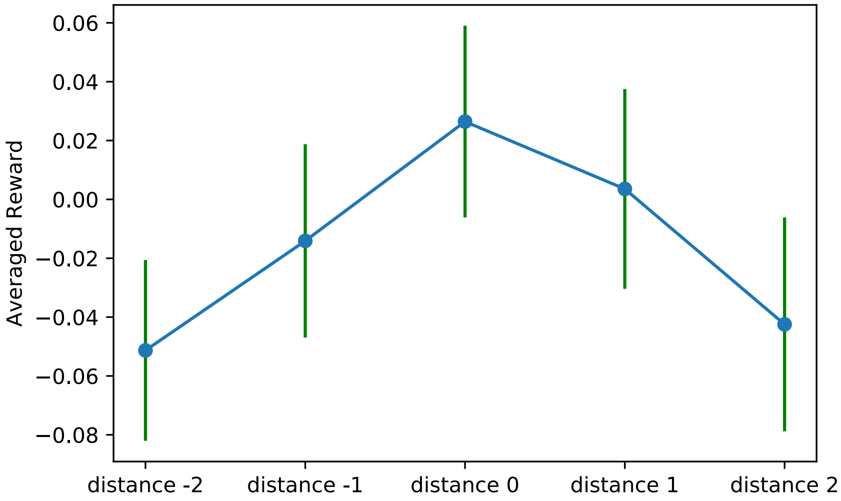

Figure 4.13: Reward of each class (Emory data)

It can be seen from figure 4.13 that the distance 0 class achieved the highest reward. The reward will decrease with the increase of absolute distance. In order to determine the causality between distance and reward, the multiple linear regression is used again. We extract the history of clotting complication and bleeding complication from IMBills of patients, combined with absolute distance, weight, age and coagulation SOFA score to predict the reward.

OLS Regression Results

	coef	std err	t	$P > \|t\|$	[95% Conf. Interval]
const	0.0032	0.004	0.847	0.397	[-0.004 , 0.011]
distance	-0.0198	0.006	-3.397	0.001	[-0.030 , -0.008]
hi_clot	0.0074	0.006	1.217	0.224	[-0.005 , 0.019]
hi_blood	0.0048	0.006	0.788	0.431	[-0.007 , 0.017]
weight	0.0004	0.006	0.060	0.952	[-0.011 , 0.012]
age	0.0059	0.006	1.024	0.306	[-0.005 , 0.017]
SOFA	-0.0029	0.006	-0.517	0.605	[-0.014 , 0.008]

Table 4.3: Regression on reward (Emory data)

From the regression summary in table 4.3, we found that only the distance variable has significant p-value. Thus we reject the null hypothesis of distance variable. In addition, the distance is negatively associated with rewards according to the coefficient. In other words, the closer a dosing compared with the recommendation, the higher reward it will achieve.

Evaluation on clotting complication

The second evaluation of Emory ICU data is focused on the clotting complications. From the IMBills assigned in the heparin treatment process, we extract the time stamp of complication related with blood clot including pulmonary embolism (PE) and Deep vein thrombosis (DVT). In the evaluation part, we set a “clot” value for each patient (1 for patients with clotting complication, 0 for patients without clotting complication). So the higher value of “clot” represents higher probability to get clotting complications.

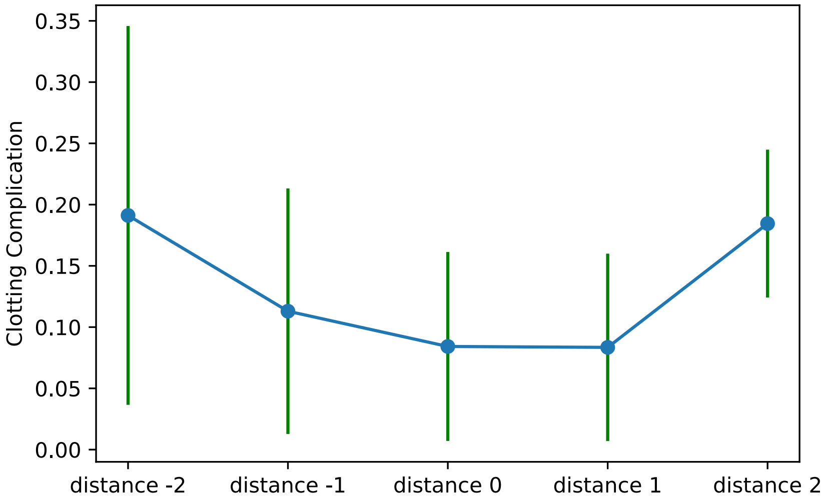

Figure 4.14: Clot complication of each class (Emory data)

As shown in figure 4.14, the probability of getting clotting complication is relatively high in both distance 2 class and distance -2 class. For distance 2 class, 81.2% patients are treated as low standard by clinicians, which means these patients are more likely to get bleeding complication. Therefore, we assume that the high probability is due to the under dose of heparin. For distance -2 class, 80.9% patients are treated as high standard by clinicians, which means these patients are more likely to get clotting complication. So we assume that the high probability might due to other factors before the heparin treatment. In order to support our assumption, we use the covariance features and distance to predict the clotting complication by using logistic regression.

Logit Regression Results

	coef	std err	z	$P > \|z\|$	[95% Conf. Interval]
const	-2.3724	0.076	-31.179	0.000	[-2.522 , -2.223]
distance	0.0146	0.004	3.784	0.001	[0.007 , 0.022]
hi_clot	0.1156	0.064	1.795	0.073	[-0.011 , 0.242]
weight	0.0740	0.069	1.077	0.282	[-0.061 , 0.209]
age	-0.1416	0.073	-1.937	0.053	[-0.285 , 0.002]
SOFA	-0.1059	0.083	-1.275	0.202	[-0.269 , 0.057]

Table 4.4: Regression on clot complication (Emory data)

Table 4.4 illustrate the result of logistic regression. Distance is the only significant variable with p-value smaller than 0.05. The positive coefficient just matched with our assumption. With the increase of distance between recommendation and clinicians dosing, the patient might not received enough heparin dosing. As a consequence, the probability of clotting complication will be higher.

Evaluation on bleeding complication

The third evaluation is related with bleeding complications. From the IMBills assigned in the heparin treatment process, we extract the time stamp of complication related with bleeding instance. In this evaluation part, we set a “bleed” value for each patient (1 for patients with bleeding complication, 0 for patients without bleeding complication) and calculate the averaged value for each distance class. So the higher value of “bleed” represents higher probability to get bleeding complications. There are only 113 patients with bleeding complication in the total 2310 patients. It is shown in figure 4.15 that only the distance -2 class got a relatively high probability on bleeding complication. As mentioned in previous evaluation, there are 80.9% patients in distance -2 class are treated as high standard by clinicians. We assume that either the bleeding instance is caused by over dose of heparin, or it is caused by other confounding factors. In order to find out the causality, we implemented the logistic regression to predict bleeding complications.

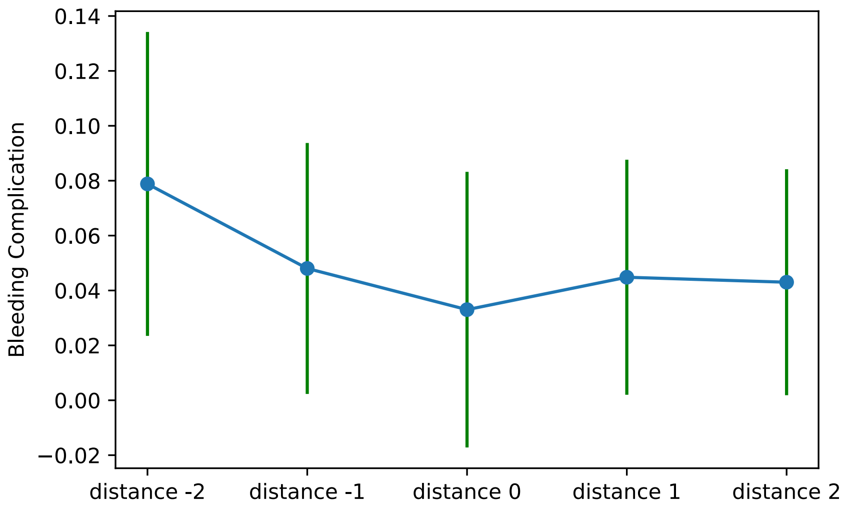

Figure 4.15: Bleed complication of each class (Emory data)

Logit Regression Results

	coef	std err	z	$P > \|z\|$	[95% Conf. Interval]
const	-3.0050	0.099	-30.220	0.000	[-3.200 , -2.810]
distance	-0.0282	0.004	-7.198	0.000	[-0.036 , -0.021]
hi_bleed	-0.1086	0.105	-1.029	0.303	[-0.315 , 0.098]
weight	-0.0112	0.101	-0.111	0.912	[-0.210 , 0.187]
age	0.0027	0.098	0.028	0.978	[-0.190 , 0.196]
SOFA	0.2492	0.074	3.353	0.001	[0.104 , 0.395]

Table 4.5: Regression on bleed complication (Emory data)

The table 4.5 shows that there are two significant variables associated with bleeding complication. The first variable is distance. Since the coefficient is negative, a decrease of distance is associated with an increase of bleeding probability. This relationship support our assumption that if a heparin dosing is much higher than recommendation, it might probably result in bleeding instance. The second variable is the coagulation SOFA scores. SOFA score is used to track a patient's status during the stay in ICU. Generally speaking, high score indicates high mortality rate. The SOFA score in Emory ICU data is based on coagulation system, which is directly related to Platelets count. For example, SOFA score = 0 when Platelets count $\geq 150 \times 10^3/\mu l$. SOFA score = 5 when Platelets count $\leq 20 \times 10^3/\mu l$. The coagulation SOFA score should be positive associated with bleeding complications based on medical knowledge. It can be seen from the summary the coefficient of SOFA score is positive.

4.3.3 Classification task

In terms of the decision support system, there are several different forms. Except providing the exact suggesting dosing at each time step, one feasible format is casting the problem as a classification task. Specifically, whenever the clinician would like to set a heparin dosing, the decision support system should predict the possible outcomes. In this task, the classification labels can be defined as sub-therapeutic (low heparin level), therapeutic and supra-therapeutic (high heparin level).

General settings

Since there are two standards in the Emory ICU data, we defined four classes as follows: class 0 ranged from (0, 0.3], class 1 ranged from (0.3, 0.5], class 2 ranged from (0.5, 0.7] and class 3 ranged from (0.7, 1]. We split the data into 80% and 20% for training and testing. For each patient, whenever there is a heparin level tested, we extracted the heparin level hl_t as labels. Meanwhile, we combined the state of patient at previous hour $state_{t-1}$ and heparin dosing infused at previous hour $dosing_{t-1}$ to form the corresponding features. In the end, the dimension of features is 50. We chose the support vector machine (SVM) as the classifier. In order to deal with the problem of uneven data distribution, we implemented a "balanced" technique which uses the values of labels to automatically adjust weights inversely proportional to class frequencies in the input data. Figure 4.16 is the confusion matrix of this classification task. It can be seen from the matrix that the accuracy is not satisfied. The situations of class 0 and class 2 are slightly better than other classes. The performance of this baseline setting is as follows:

Training accuracy: 0.473361

Testing accuracy: 0.433255

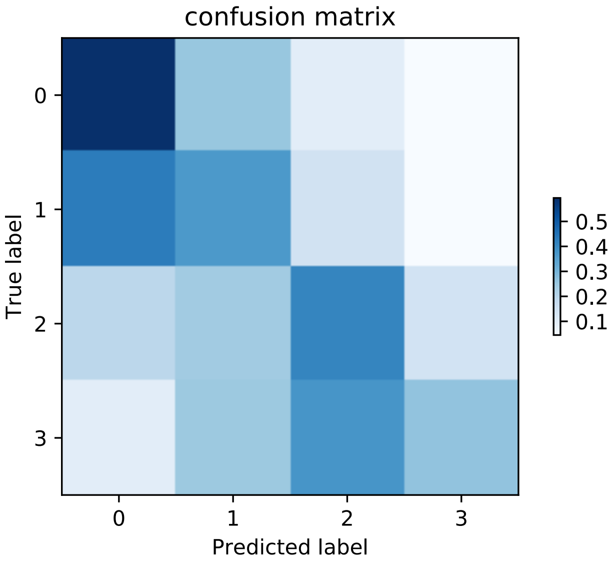

Figure 4.16: Confusion matrix (1)

Adding information

In order to improve the classification accuracy, we add the information of distance between our suggested dosing from RL agent and the clinicians dosing into the features. Now the dimension of features is 51. As shown in figure 4.17, although the accuracy is still relatively low, the confusion matrix is less "confusing" than the previous one. At least the predicted label with highest probability for each class is the true label. Adding the distance variable is providing useful information. The performance of this new setting is as follows:

Training accuracy: 0.475410

Testing accuracy: 0.453162

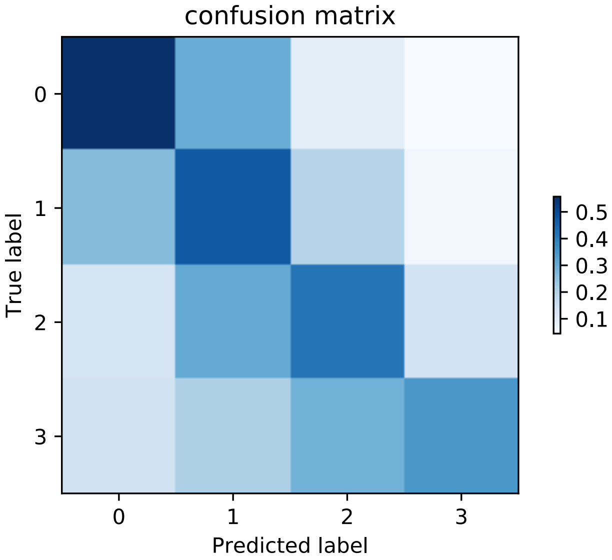

Figure 4.17: Confusion matrix (2)

Chapter 5

Conclusion

In this thesis, the main goal is two fold: one is to provide individualized medication dosing suggestion for patients by implementing a deep reinforcement learning algorithm, the other one is to evaluate the average effect of the suggested dosing after adjusting confounding factors. The experiments performed on simulated data has shown that given well structured features, predicting optimal heparin dosing with respect to various medical conditions is possible. In the real data experiments, we set a clinicians-in-the-loop framework based on the RL algorithm. Rather than executing the action predicted by RL agent, the clinicians will determine the final actions in the loop. In the learning process, instead of predicting the optimal dosing, the RL agent aims to evaluate the clinician's dosing and learn from these experiences. The results from the two real ICU data indicates that our decision support system is able to provide reasonable heparin dosing according to multidimensional features of patients and sparse reward. In conclusion, we assume that the dosing recommendation provided by our system is in the therapeutic range, In order to support our hypothesis, we performed several evaluations related with causality on the results. The multiple linear regression and logistics regression analysis revealed that the distance between recommended dosing and clinicians' dosing is significant associated with outcomes, which means the outcomes such as rewards and complications are related with how close the actual dosing is compared with the recommended dosing. Overall, the evaluations strengthen our hypothesis that the recommended dosing is a safe choice which ranged within the therapeutic window.

An issue that was not addressed in the thesis was the limited learning ability of our RL agent. The limitation was shown in the experiment performed on the Emory ICU data. Whenever the heparin level exceeded above 0.8 (high standard patient) or 0.9 (low standard patient), both the clinicians and protocols will choose to stop infusion completely to decrease the risk of bleeding instance for 1 or 2 hours. However, the RL agent will just decrease the dosing. Although we are not sure whether this kind of behavior will certainly lead to bleeding complications in the real medical treatment, the "stop" behavior is not mastered by the RL agent. This phenomenon suggested that there are limitations on the learning ability of RL agent. One possible reason caused this issue might be the design of reward. In our experiments, reward are calculated from the heparin level at next hour, which could be delayed and sparse. It would be interesting to design a new reward to keep the performance and fix this issue.

A limitation of this study is the quality of data. Unlike the simulated coagulation model, the real medical data is not perfect. There are a lot of missing values in the features. Imputing these missing values simply with mean value or most frequent item might oversimplify the complexity of medical features and lose a significant proportion of the variance. As we can see from the last classification experiment, the training accuracy is pretty low. Reconstructing the features to its original state might greatly enhanced the performance. Future research could also focus on the improvements of medical data.

Bibliography

[1] Andrew J Schaefer, Matthew D Bailey, Steven M Shechter, and Mark S Roberts. Modeling medical treatment using markov decision processes. In *Operations research and health care*, pages 593–612. Springer, 2005.

[2] John T James. A new, evidence-based estimate of patient harms associated with hospital care. *Journal of patient safety*, 9(3):122–128, 2013.

[3] Shamim Nemati, Mohammad M Ghassemi, and Gari D Clifford. Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pages 2978–2981. IEEE, 2016.

[4] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[5] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 387–395, 2014.

[6] T Wajima, GK Isbister, and SB Duffull. A comprehensive model for the humoral coagulation network in humans. *Clinical Pharmacology & Therapeutics*, 86(3):290–298, 2009.

[7] Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

[8] Romain Pirracchio, Maya L Petersen, Marco Carone, Matthieu Resche Rigon, Sylvie Chevret, and Mark J van der Laan. Mortality prediction in intensive care units with the super icu learner algorithm (sicula): a population-based study. *The Lancet Respiratory Medicine*, 3(1):42–52, 2015.

[9] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[11] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[12] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz

Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[13] Roland Hafner and Martin Riedmiller. Reinforcement learning in feedback control. *Machine learning*, 84(1):137–169, 2011.

[14] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[16] Frederick A Anderson, H Brownell Wheeler, Robert J Goldberg, David W Hosmer, Nilima A Patwardhan, Borko Jovanovic, Ann Forcier, and James E Dalen. A population-based perspective of the hospital incidence and case-fatality rates of deep vein thrombosis and pulmonary embolism: the worcester dvt study. *Archives of internal medicine*, 151(5):933–938, 1991.

[17] Samuel Z Goldhaber, Daniel D Savage, Robert J Garrison, William P Castelli, William B Kannel, Patricia M McNamara, Gherardo Gherardi, and Manning Feinleib. Risk factors for pulmonary embolism: the

framingham study. *The American journal of medicine*, 74(6):1023–1028, 1983.