Mechanical Ventilation Control with Reinforcement Learning

Golnaz Mesbahi
mesbahi@ualberta.ca

Seyed Parsa Darbouy
darbouy@ualberta.ca

1 Ethics Statement

We the undersigned understand that it is plagiarism to copy text from an external source and hand it in as our own work. If we copy from an external source, we must put the full copied section within quotes and then cite the source. A proposal or report with a large amount of copied text (even if properly cited) is not acceptable. It is also plagiarism to copy text from an external source and then change some words, even if a citation is given.
Golnaz Mesbahi
Seyed Parsa Darbouy

2 Group Members

Golnaz Mesbahi and Seyed Parsa Darbouy

3 introduction

Many ongoing open challenges in data science and machine learning are valuable both for scientific purposes and for the benefit of society. Therefore, we have decided to base our project on a vital medical challenge on the Kaggle website named *Ventilator Pressure Prediction*.(Kaggle)
Many patients with respiratory illnesses need assistance to breathe normally. A mechanical ventilator uses pressure to help these patients(Shepard). A nurse is needed to observe the patient's condition and control the ventilator(Jubran, 2012). Due to the global shortage of nurses (Biron and Oliver, 2019), needing a nurse constantly to control a ventilator can be problematic during a pandemic. Therefore, building an AI agent to control the ventilator can be a remarkable help to healthcare systems.
Our goal in this project is to simulate a ventilator connected to sedated patient's lung and predict what pressure the ventilator should be set to.(Kaggle)

4 Related Work

PID controllers

One of the traditional control methods of a mechanical ventilator is to use a PID controller. However, this method needs the supervision of a medical practitioner. A nurse is responsible for continuously monitoring and adjusting the mechanical ventilator with PID controllers. A PID controller aims to correct the output of the ventilator based on the collected errors from previous outputs. (Belgaid, 2022)

Machine Learning Approaches

Using machine learning in health applications is an important research trend, due to the strengths of machine learning, especially in the presence of large-scale data.
During the COVID-19 pandemic, many papers worked on improving ventilator control by using machine learning.(Suo et al., 2021) They showed the advantages of using machine learning techniques in controlling a ventilator. (Belgaid, 2022) Based on this paper, and the corresponding competition on Kaggle, other machine learning, and deep learning methods have been applied to this problem. For example, Belgaid (2022) presents a deep neural network approach to simulate the pressure of a mechanical ventilator. We also plan to follow this trend and design a Reinforcement learning model which can solve this control problem efficiently.

Reinforcement Learning

A team in Google AI Princeton lab used a model-based approach for the mechanical ventilation problem. (Suo et al., 2021). They produced their train data using the People's Ventilator Project (LaChance et al., 2020), an open-source ventilator created by Princeton University. Moreover, they utilized a test lung provided by IngMar Medical called QuickLung Products (IngMar_Medical) to do their mechanical ventilation tasks. At the end, they reach a model that has a mean square error that is 22% lower than the best PID controller.(Google_AI_Princeton_Team)

5 Data

Kaggle has separated the data into train and test data. Predicting the mechanical ventilator's ideal pressure is our goal. The data in Table 1 are taken from the Kaggle website. (Kaggle)
In addition, we discovered that each breath has 80 time steps through data analysis.

6 Methodology

We decided to view this project as a prediction problem, where the goal is to predict the value of pressure the ventilator should pass to the patient's lung in order to help them breathe. This prediction should be made based on the representation we build from the information we have about the current patient's lung and other attributes of the environment provided in the dataset. We employ the idea of general value functions in a reinforcement learning problem setting to build an agent and solve this regression problem.

To provide a brief background, reinforcement learning is a problem setting where an agent aims to learn from its interaction with the environment. At each time step, the agent takes action A_t, receives a reward R_{t+1} from the environment, and goes from state S_t to S_{t+1}. The goal of the agent is to maximize cumulative reward, called "return," with this definition: $G_t = \sum_{k=0} \gamma^k R_{t+1+k}$.γ is called the discount factor and is a value between 0 and 1 that indicates how far-sighted the agent is.(Sutton and Barto, 2018)

Reinforcement learning problems can be divided into prediction and control problems. In a prediction problem, like ours, the goal is to estimate the value of the state. Value functions predict future return. This is the definition of a value function: $V_\pi(s) = E_\pi[G_t \mid S_t = t]$

General value functions (GVFs) make two relaxations to the definition of value functions. (Sherstan, 2020)

1. Instead of the reward, R, we are allowed to choose any signal of interest that can be observed by the agent. We call this prediction target, cumulant C or pseudo reward.
2. Discount factor γ does not need to be constant and can be defined as a function of the current state, current action, and next state.

The goal is still to maximize future return, but the definition of return is now relaxed based on the two ideas mentioned above. The return is now defined as: $G_t = \sum_{k=0} \left(\prod_{j=1}^{k} \gamma_{t+j} \right) C_{t+1+k}$

GVFs allow the agent to show the elements of representation in the form of predictive questions (Sherstan, 2020). Consider the mechanical ventilation problem. The agent will ask the below question in order to form an awareness of its environment.
Question: If I go through consecutive steps of inhaling and exhaling, how much pressure will I observe from the professional?
Policy: going through time
Cumulant: scaled true pressure
Gamma: 1
(we will discuss the details of the choices of policy, cumulant, and gamma in the next sections.) This means we have changed perspective and are looking at the pressure as another sensory data that the agent aims to gain awareness of in a GVF framework.

Once we specify the details of our GVF framework, we are free to use any usual reinforcement learning algorithm, in our case, temporal difference learning. (more details on the algorithm will be provided in the next sections.)

6.1 Reinforcement Learning

First, we need to define a Markov decision process (MDP) in order to formulate this problem as a reinforcement learning problem. A Markov decision process is defined as a tuple S, A, P , R Where
S is the finite set of states,
A is the set of actions the agent can take.
P is the state transition probability,
R is the reward function.
We will go through each of these in the next sections.

6.1.1 States

We can form a state based on each sample in the dataset. Each sample in the dataset is one step of a time series data that gives information about the observations of the environment. The state space is continuous. Thus, we need to use a discretization

Name	Information	Range of Data	Type
id	identifier of time_steps and breaths, globally-unique	1 - 6.04m	Numerical
breath_id	Identifier of a Breath, globally-unique for a single breath	1 - 126k	Numerical
R	“Lung attribute indicating how restricted the airway is (in cmH2O/L/S)”	5 - 50(cmH2O/L/S)	Numerical
C	“Lung attribute indicating how compliant the lung is (in mL/cmH2O)”	10 - 50(mL/cmH2O)	Numerical
time_step	“The actual timestamp”	0 - 2.94(Second)	Numerical
u_in	“The control input for the inspiratory solenoid valve.”	0 - 100	Numerical
u_out	“The control input for the exploratory solenoid valve”	Either 0 or 1	Categorical
pressure	“The airway pressure measured in the respiratory circuit, measured in cmH2O”	-1.9 - 64.8	Numerical

Table1: Data

technique to form a set of features as states' representations and use them to train our agent. The details about state representation are described in the next sections.

6.1.2 Actions (Policy) and Transition Probability Function

The dataset we have for this problem is a set of time series in nature. Therefore, we can define the policy as: “ going through time “ This means that the next state after each state is its next row of data in the dataset. The algorithm goes through the stream of data in order to train the agent.

6.1.3 Cumulant

We first tried to define a cumulant based on how good the predicted pressure is in comparison with the true pressure. We got the idea from a previous paper (Günther et al., 2016) where they used domain knowledge to define cumulant baed on how good the prediction was. We did not get good results with this choice, and we assume we need more domain knowledge to define such better cumulants. Therefore, we changed our approach, and got some inspiration from another previous research (Behavior et al., 2014) whtere they scale the true values in order to define cumulant. We scaled the true pressure to a value between 0 and 1, and used it as the cumulant. This choice worked for us, and we were able to train the agent based on this idea.

6.1.4 Discount Factor

First, in our experiments, we assumed that the problem is an episodic task. Based on the dataset, each breath consists of 80 data points; therefore, at first, we assumed that each breath is one episode. Later in the training, we realized that if we set the discount factor to 1, meaning that we consider the task is continuing, we get better results. In a general reinforcement learning setting, the discount factor can range from 0 to 1, and it remains constant throughout the training. In the GVF framework, we can think of the 1-discount factor as the probability of task termination, and the value can change over time. In our case, a constant value of 1 gave us reasonable results.

6.2 State Representation

The state space is continous, and we need a way to form state representations based on the raw dataset. We choose tile coding as the method to construct features. For each state s, we need to form a feature vector $x(s)$ that represents the state. The idea of tile coding is that when we have a continous state space, we fit a set of tilings with offsets from each other to the state space. Then we index the tiles of the tilings, and for any point in the state space, we look for the active tiles. (The tiles that contain that point). This forms a binary feature vector with a particular number of active features.(Sutton and Barto, 2018) Figure 1 shows how a tile coder works in general. We ought to specify some hyperparameters of the tile coder. In our case, the hash table size is 2048, the number of tiles is 16, and the number of tilings is 16 as well. Figure 2 shows the idea of our tile coder.

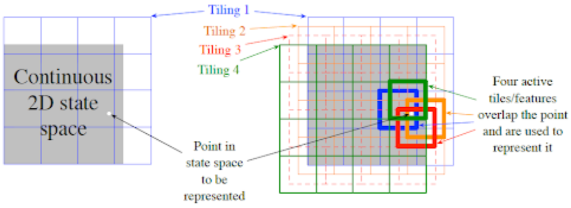
Tiling 1
Tiling 2
Tiling 3
Tiling 4
Continuous 2D state space
Point in state space to be represented
Four active tiles/features overlap the point and are used to represent it

Figure 1: How a Tile Coder Works

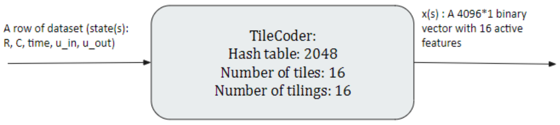
A row of dataset (state(s): R, C, time, u_in, u_out)
TileCoder:
Hash table: 2048
Number of tiles: 16
Number of tilings: 16
x(s) : A 4096*1 binary vector with 16 active features

Figure 2: the Tile Coder for the Ventilation Control Problem

6.3 Algorithm Formulation

After setting up the attributes of the general value function, we can use any RL prediction algorithm to solve the problem. (Sherstan, 2020) One of the powerful prediction algorithms is Temporal Difference Learning(TD) (Sutton and Barto, 2018). TD performs bootstrapping and is a suitable algorithm to handle the non-stationarity of the environment. We choose Semi Gradient TD learning with linear function approximation as our prediction algorithm. The pseudocode for the algorithm is shown in Algorithm 1.

Algorithm 1 Semi Gradient TD(0) with Linear Function Approximation

```
initialize $w$
for $t = 1, 2, \dots$ do
    calculate $C$
    $\delta \leftarrow C + \gamma w^T x(t+1) - w^T x(t)$
    $w \leftarrow w + \alpha \delta x(t)$
    if $t\%1500 == 0$ then
        $\alpha \leftarrow \alpha \times 0.9$
    end if
end for
```

6.4 Training, Testing, and Analysis

We go through a stream of data with 16000 steps to train the model based on the algorithm. The agent learns the weight w through the training process. In order to evaluate the model, we need a performance measure. We choose mean absolute error as the performance measure of the model. We also test the trained model on unseen data to evaluate offline performance.

6.5 Step Size

When we use tile coding, the value of the step size should be divided by the number of tilings. The reason is that we want to make updates to the weight vector proportional to the unit value of the feature vector. The initial value of step size at the beginning of the training is $\frac{1}{number of tilings}$ which is equal to $\frac{1}{16}$. In order to help the agent learn for a longer period of time, we multiply the value of the step size by 0.9 every 1500 steps. This avoids divergence and allows the agent to take larger steps at the start of learning when it is not close to the ideal values and smaller steps as it learns over time.

7 Results

We trained the agent on 16000 time steps and analyzed the online learning curve. The learning curve (mean absolute error over time) is shown in figure 3, and can give us valuable information about the convergence of the agent. The figure shows that the agent is learning since the error is decreasing over time. We also tested the model offline on unseen data and figure 7 displays the mean absolute error of our model. We examined different values for each hyperparameter to find the best configuration of the agent.

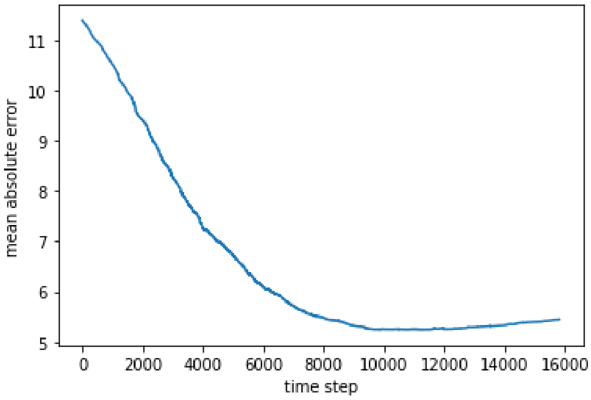

Figure 3: Mean Absolute Error Over Time (online)

7.1 Discount Factor

We tried different values for gamma, ranging from 0.8 to 1, and only when we set gamma to 1, we got a reasonable result. This shows that the problem is a continuing task, not an episodic task, contrary to what we hypothesized at the beginning.

7.2 Sensitivity to the Step Size

We found out that the step size in some cases might result in the divergence of the agent if not dealt with carefully. The reason is that our problem setting demands the agent to learn continually. Therefore, as the agent learns, it should take smaller steps to avoid divergence and learn for a longer period of time.

7.3 Sensitvity to the Tile Coder

We also found out that the tile coding we designed, although working, can be improved. Figure 4 and Figure 5 compares the true pressure and the estimated pressure by the model in 800 steps in online and offline settings.

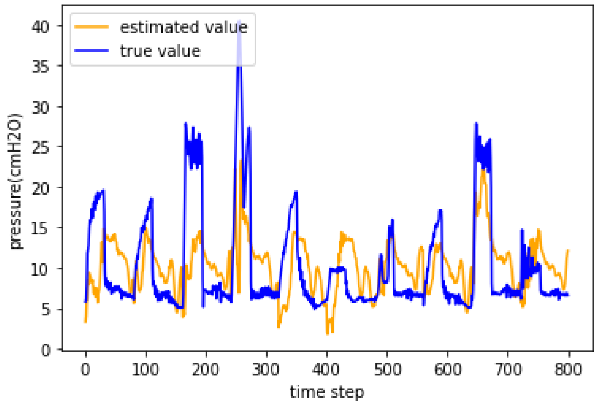

Figure 4: true pressure and estimated pressure(online)

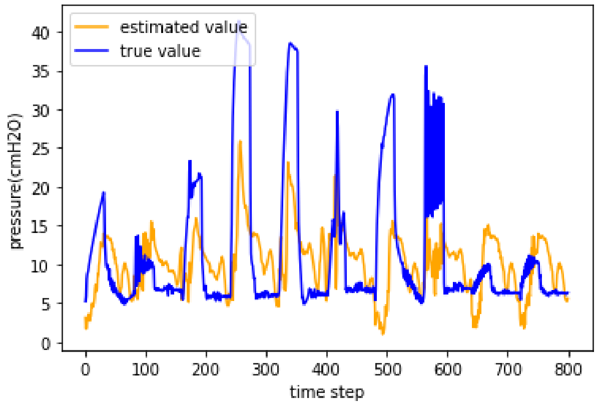

Figure 5: true pressure and estimated pressure(offline)

Mechanical ventilation is a critical task, and the estimated pressure needs to be as accurate as possible. However, this figure shows there is still room for improvement. We discussed with Dr. Adam White why the estimated pressure has captured the alternating shape of the true pressure, (which means the agent is learning), but it can still be improved. (which means we can change some parameters to make the agent learn even better). He mentioned that the way we are doing the tile coding is resulting in some unwanted generalizations over states. We give the tile coder all of the features at once, and try to get the binary representation in one round. However, one better way to do the tile coding is to pass features as groups of two, get the binary representation for each of them, and then concatenate the representations to get one representation for each state.

About other choices of the design of the tile coder, it is worth noting that larger values for the number of tiles and the number of tilings will result in better generalization and discrimination. We choose values of 16 and 16 for these two hyperparameters for our case. We also should specify the size of the hash table for the tile coder, which in our case is 2048. The size of the hash table indicates the dimension of the binary feature vector and weight vector.

7.4 Lung Attributes

For the sake of simplifying our model, we tried to combine C and R into one feature and train the model, but it did not result in an improvement. In addition, we tried to train the model without one of these features, and again it did not work. This showed us that lung attributes are important features in this prediction problem and cannot be neglected. Also, other features in the dataset were independent of each other, and therefore, there was no room for reducing the dimensions of the dataset before feeding it to the tile coder.

8 Discussion and Future Work

8.1 What do the results mean?

The mean absolute error curve has a decreasing trend over 16000 steps of training. Therefore, this learning curve demonstrates that the agent is learning. We also test the trained model offline on unseen data, and the calculated MAE is shown in figure 7 for different settings.

8.2 What does the results tell us about the problem?

The overall results show that the algorithm we proposed in this project is a good match for this regression problem for the following reasons.

First, the model has a better performance than traditional PID controllers in most lung attributes. This shows encouraging outcomes that the model can be used in practice in a medical setting.

Secondly, RL framework allows us to design an agent that can learn online through its interaction with the environment. It is an advantage for the prediction model to be able to adapt itself and learn online as it interacts with the world and experiences new conditions, e.g new patients with different conditions and lung attributes.

Finally, there has been recent research on the explainability of general value functions. GVFs can provide a self-explanation that can help explain their decisions to human designers (Kearney et al., 2022). This is an advantage, and in some cases a necessity for a model of this critical application to be explainable.

8.3 Comparison with Previous Results

Based on the article written in google blog(the Google AI Princeton Team that worked on this dataset and proposed a model), The performance measure they used to report their results is Mean Absolute Error (MAE). (Google_AI_Princeton_Team) They trained a model and evaluated its performance against the best traditional PID controller. They divided their test data into groups according to characteristics of the lungs. We repeated their action on our test data in order to make it easier to compare our findings with their results.
Figure 6 shows the results from previous research on this problem. Moreover, figure 7 illustrates MAE of our model.
Comparing Figure 6 and 7 demonstrates that our model performs better than traditional PID when C=20 and C=50. But when C=10 PID controller has better performance. Our model outperforms traditional PID controller by an average of 13%. Although we need to exactly reproduce previous models and perform statistical tests to prove this claim for sure, this is promising preliminary results.

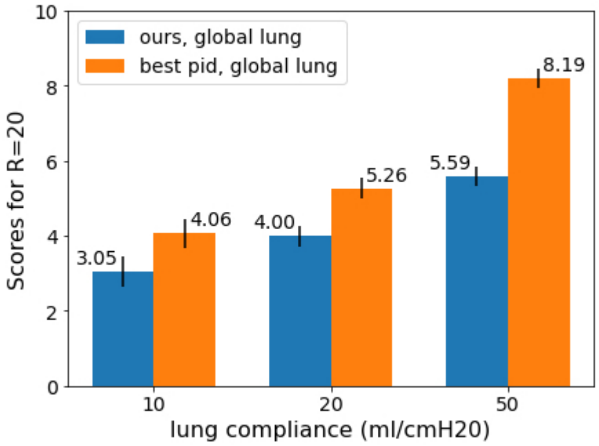

Figure 6: Results from previous research on this problem.

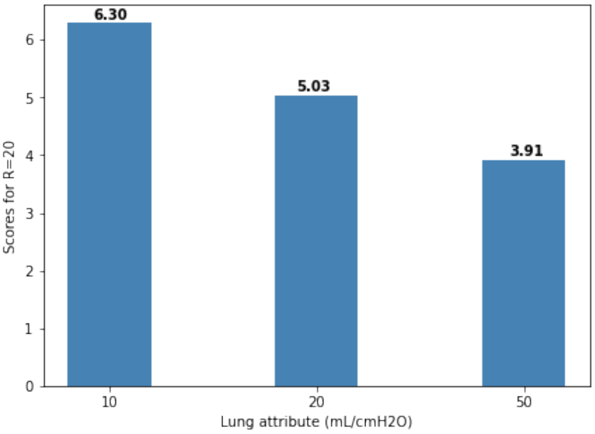

Figure 7: MAE of our model.

8.4 If We Had More Time

If we had more time, we would improve our tile coding methods to get more accurate results. we could also reproduce some previously proposed models for this dataset, and perform some statistical tests, in particular, a t-test to compare two models, to analyze the statistical significance of our proposed model. Moreover, we could examine other reinforcement learning algorithms such as TD lambda, to see how performance changes or gets better. Lastly, we believe that having more domain knowledge of the problem can lead to better choices of cumulant, and other attributes of the RL algorithm, and can lead to better results. For instance, in Günther et al. (2016) article in which they use a similar algorithm for another application, they use domain knowledge to define a discrete pseudo reward which indicates how good the prediction is. However, we did not have access to such knowledge and built our model based on simpler choices of cumulant. Also, there are some open source resources such as People's Ventilator Project, a ventilator simulator, that can give us valuable information about the problem setting. Adding such a professional medical point of view to the problem might also result in better performance of the model which can be done as future work.

9 Ethical Implications

This project is ethically crucial in the sense that it provides the settings for an equipment that is in close contact with public health. While its inefficiencies can cause much damage, a robust and reliable design of the controller can ease the automation and help the healthcare workers and help patients heal easier.

9.1 Representation Bias

The dataset contains C and R as lung attributes. However, we are not sure if C and R are enough for representing all the necessary information about differences in patients' lungs.

9.2 Measurement Bias

When a medical practitioner sets the pressure of the ventilator, they not only pay attention to the features mentioned in the dataset, but they also check the overall health condition and health signals of the patient. However, it would be expensive to contain all the health data for each patient in the dataset. Therefore, we believe the current features in the dataset might oversimplify the construct (patient's condition).

10 Work Distribution and Source Code

We had several meetings with each other where we discussed the details of implementation and did the implementation together. Here is the link to the implementation: (Mesbahi and Darbouy)

11 Acknowledgements

We would like to thank Dr. Alona Fyshe and Fernando Hernandez Garcia for their insightful comments. We also would like to thank Dr. Adam White for his guidance on the Tile Coder.

References

Adaptive Behavior, Joseph Modayil, Adam White, and Richard S Sutton. 2014. Multi-timescale nexting in a reinforcement learning robot. 22:146–160.

Abdelghani Belgaid. 2022. Deep sequence modeling for pressure controlled mechanical ventilation. *medRxiv*, page 2022.03.02.22271790.

Alain Biron and Catherine Oliver. 2019. Global shortage of nurses the mcgill nursing collaborative for education and innovation in patient-and family-centered care.

Google_AI_Princeton_Team. How ventilators work and why they are so important in saving people with coronavirus — coronavirus — the guardian.

Johannes Günther, Patrick M. Pilarski, Gerhard Helfrich, Hao Shen, and Klaus Diepold. 2016. Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning. *Mechatronics*, 34:1–11.

IngMar_Medical. Quicklung products.

Amal Jubran. 2012. Nurses and ventilators. *Critical Care*, 16:115.

Kaggle. Google brain - ventilator pressure prediction. `https://www.kaggle.com/competitions/ventilator-pressure-prediction/overview`.

Alex Kearney, Johannes Günther, and Patrick M. Pilarski. 2022. Prediction, knowledge, and explainability: Examining the use of general value functions in machine knowledge. *Frontiers in Artificial Intelligence*, 5:826724.

Julienne LaChance, Tom J. Zajdel, Manuel Schottdorf, Jonny L. Saunders, Sophie Dvali, Chase Marshall, Lorenzo Seirup, Daniel A. Notterman, and Daniel J. Cohen. 2020. Pvp1–the people's ventilator project:

A fully open, low-cost, pressure-controlled ventilator. *medRxiv*, page 2020.10.02.20206037.

Mesbahi and Darbouy. Mechanical ventilation control with reinforcement learning.

Bob Shepard. How a ventilator works. and why you don't want to need one. https://www.uab.edu/news/health/item/11430-how-a-ventilator-works-and-why-you-don-t-want-to-need-one#:~:text=The%20machine%20uses%20positive%20pressure,cannot%20breathe%20on%20their%20own.

Craig Sherstan. 2020. Representation and general value functions.

Daniel Suo, Cyril Zhang, Paula Gradu, Udaya Ghai, Xinyi Chen, Edgar Minasyan, Naman Agarwal, Karan Singh, Julienne LaChance, Tom Zajdel, Manuel Schottdorf, Daniel Cohen, and Elad Hazan. 2021. Machine learning for mechanical ventilation control. *medRxiv*, page 2021.02.26.21252524.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*, second edition. The MIT Press.