Winter 12-6-2018

Model-based reinforcement learning: A survey

Fengji Yi
Communication University of China, Beijing, yifengji1020@cuc.edu.cn

Wenlong Fu
Communication University of China, Beijing, fwl2000@163.com

Huan Liang
Communication University of China, Beijing, lianghuan@cuc.edu.cn

Yi, F.J., Fu, W.L., & Zhang, H.L. (2018). Model-based reinforcement learning: A survey. In *Proceedings of The 18th International Conference on Electronic Business* (pp. 421-429). ICEB, Guilin, China, December 2-6.

Model-based reinforcement learning: A survey

(Full Paper)

Fengji Yi, Communication University of China, Beijing, yifengji1020@cuc.edu.cn
Wenlong Fu*, Communication University of China, Beijing, fwl2000@163.com
Huan Liang, Communication University of China, Beijing, lianghuan@cuc.edu.cn

ABSTRACT

Reinforcement learning is an important branch of machine learning and artificial intelligence. Compared with traditional reinforcement learning, model-based reinforcement learning obtains the action of the next state by the model that has been learned, and then optimizes the policy, which greatly improves data efficiency. Based on the present status of research on model-based reinforcement learning at home and abroad, this paper comprehensively reviews the key techniques of model-based reinforcement learning, summarizes the characteristics, advantages and defects of each technology, and analyzes the application of model-based reinforcement learning in games, robotics and brain science.

Keywords: Reinforcement learning, data efficiency, optimizing, dynamic models, value function approximation.

*Corresponding author

INTRODUCTION

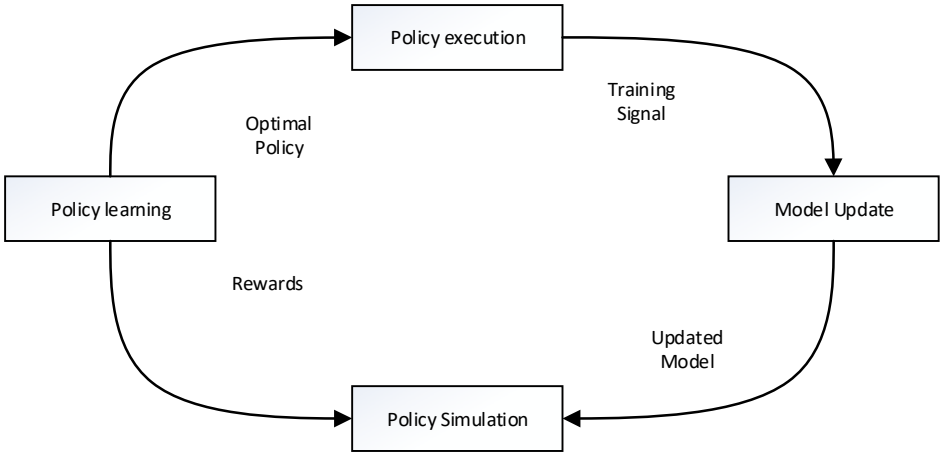
Model-based reinforcement learning refers to the establishment of a model according to the environment, so that the agent knows how the environment shifts the state and the feedback rewards, and then finds the optimal policy based on the model to get the maximum cumulative reward. With the development of deep learning (LeCun *et al.*, 2015) in recent years, model-based reinforcement learning has greatly improved in terms of data efficiency and generalization ability of models. With model-based reinforcement learning in the classic Atari games (Stadie *et al.*, 2015), Alpha Go (Silver & Huang *et al.*, 2016), face recognition (Rao *et al.*, 2017), robotics (Mordatch *et al.*, 2012), medicine (Hamaya *et al.*, 2016; Sharp *et al.*, 2015), brain science (Daw *et al.*, 2005; Niv *et al.*, 2007), automatic driving (Deisenroth *et al.*, 2013), natural language processing (Scheffler *et al.*, 2002) etc., gradually playing an increasingly important role, so it is necessary to a comprehensive review of the new developments in model-based reinforcement learning.

Kaelbling *et al.* (1996) reviewed the work of early reinforcement learning and the core issues from the perspective of computer science, including the compromise of exploration and exploitation, the construction of models to accelerate learning, and so on; Quan *et al.* (Quan *et al.*, 2018) reviewed three main methods of deep reinforcement learning and some frontier directions of deep reinforcement learning; Yuxi Li (2018) summarizes the algorithm and development of reinforcement learning from the basic elements of reinforcement learning; Dongbin *et al.* (2016) reviewed the algorithm of reinforcement learning and the future development from the perspective of computer Go; Wenji & Yang (2017) reviewed the problem of hierarchical reinforcement learning in dimensional disasters. It is worth noting that with the development of deep learning in recent years, model-based reinforcement learning has made great progress in the generalization of the model and data efficiency. Different from the existing review (Kaelbling *et al.*, 1996; Quan *et al.*, 2018; Yuxi Li, 2018; Dongbin *et al.*, 2016; Wenji & Yang, 2017), this paper focuses on the recent progress, advantages and disadvantages of the model-based reinforcement learning field.

PROBLEM DEFINITION

Reinforcement Learning (RL) (Sutton & Barto; 1998) refers to learning to behave optimally in a stochastic environment by taking actions and receiving rewards. Markov Decision Process (MDP) are meant to be a straightforward framing of the problem of learning from interaction to achieve a goal, the interaction between the agent and the environment is modeled as the MDP, it is, $[S, A, \varepsilon, P(s'|s,a), R(s,s',a), \gamma]$. Where S is the set of possible states of the agent; A is a set of actions; P is the probability that the agent transfers from the current state s to the next state s′ in the current action a; R is the immediate return that the action a transfers from state s to next state s′, γ is the discount factor, $\gamma \in [0,1]$, indicating the difference between the future reward and the current reward, that is, the proportion of each reward is different. According to the environmental models, reinforcement learning can be divided into model-based reinforcement learning and model-free reinforcement learning.

Model-based reinforcement learning (Ray & Tadepalli, 2010) refers to learning optimal behavior indirectly by learning a model of dynamics by taking actions and observing the outcomes that include the next state and the immediate reward. The specific process is shown in Figure 1:



Source: Polydoros & Nalpantidis, 2017
Figure 1: Flow chart of model-based reinforcement learning

The difference between model-based reinforcement learning and model-free reinforcement learning is that model-based reinforcement learning does not require large number of trial and error experiments to generate optimal actions as model-free reinforcement learning. While model-based reinforcement learning is that the agent gets the optimal action by the transition model, the data efficiency of model-free reinforcement learning is lower because of the large amount of the trial and error experiments. However, model-based reinforcement learning can improve the data efficiency, it can quickly arrive at the near-optimal control with learned models under fairly restricted dynamics. When generalizing to the new environment, agents can rely on the learned models for reasoning. Therefore, the structure of this review is as follows: section 2 introduces the method of approximating value function in model-based reinforcement learning; section 3 discusses the method of dynamics; in section 4, the policy search methods are covered; section 5 presents the application in model-based reinforcement learning; section 6 summarizes this review.

VALUE FUNCTION METHODS

A method that the true value of value function is fitted by a certain function (linear function or nonlinear function, etc.) on a small part of the training sample called function approximation (Xian & Yongchun, 2018). In reinforcement learning, value function approximation is reflected in many algorithms: deep Q learning (Gu *et al.*, 2016) (DQN), double DQN (Van *et al.*, 2016), dueling DQN (Wang *et al.*, 2015), and so on. In model-based reinforcement learning, if the value function is approximated from the perspective of mathematics, it can be divided into two categories, namely the parameterized value function approximation method and the nonparametric value function approximation method.

A. Function approximation method based on parameters

The parameterized value function approximation method means that the value function can be approximated by a set of parameters θ. In general, the value function approximation can be written as $\hat{v}(s,\theta)$. Therefore, when the structure of the approximating value function is determined, then the approximation of the value function is equal to the approximation of the parameter, and the update of the value function is equal to the update of the parameters. For example, approximating the dataset $T = \{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$ with N training samples, firstly the method selects a set of the basis functions, and then sets the form of the function $f(x) = \sum_{i=1}^{n}\theta_i\phi_i(x)$ to obtain the parameters $\theta_1,\theta_2,\ldots,\theta_n$ using the training data set and the optimized method. This parameter-based approach does not depend on the amount of data in the training set, the form of the basis function, the number of parameters, because they are given beforehand.

Initially, researchers used the Cerebellar Model Articulation Controller (CMAC) neural network (Albus, 1975) as a value function approximator. The biggest advantage of such a neural network is local approximation, and it has certain generalization ability. Since the neural network has the function of the cerebellum, it was originally used to solve the joint motion of the robot (Miller, Hewes *et al.*, 1990), and was later applied to the fields of robot control (Kim & Lewis, 2000), pattern recognition (Glanz & Miller., 1988; Herold *et al.*, 1989), and adaptive control (Chen & Chang., 1996; Kraft & Campagna., 1989). Although it converges faster than the BP network, as an online learning, it is still difficult to meet its rapidity requirements (Miller, Glanz & Kraft, 1990). Sutton *et al.* (Kuvayev & Sutton., 1996) used CMACs neural network as a value function approximator compared with the model-free method, and the model-based method has higher stability. With the rapid development of machine learning, Mnih, Veness et al (Mnih, Veness *et al.*, 2015) proposed a general value function approximation method using supervised learning. Not only can it be generalized to state s but also generalized to target g; Kamalapurkar *et al.* (Kamalapurkar *et al.*, 2016) used a neural network-like representation to approximate the optimal value function V* and the optimal policy u*. Such methods rely heavily on the initial model settings, so it is easy to fall into the local optimum.

B. Function approximation based on nonparametric method

The nonparametric value function approximation method does not refer to the function approximation without any parameters, but refers to the function approximation method determined by the samples which the number of parameters and the form of the base are not fixed. Common methods for nonparametric function approximation include nonparametric approximation methods based on kernel functions (Hang Li, 2012) and gaussian process-based methods (Rasmussen & Williams, 2005). In the nonparametric function approximation method, the data set $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ with the training sample N is approximated, and each sample will become a part of the function approximation, so it is called the approximation method determined by the samples. For example, in a kernel-based function approximation method, the final form of the approximation function is $f(x) = \sum_{i=1}^{N} a_i y_i K(x, x_i)$. Where N is the number of samples and $K(x,x_i)$ is a basis function.

In model-based reinforcement learning, Jong & Stone (Jong & Stone, 2002) defined an approximation model from the samples, the goal is to approximate each transition probability distribution function P and the expected reward function R using the sample of empirical data D applying the value iteration to model-based reinforcement learning; Duan & Xu (Duan & Xu, 2007) used a network integrated with Fuzzy Inference Systems (Jouffe, 1998) (FIS) and neural network as a value function approximator on the soccer robot, the approximator can progress the mapping from state space to action space, which can effectively guarantee the stability and convergence of learning. Since the non-parametric value function approximation method is determined by the samples, the calculation speed is difficult to guarantee when the number of samples is very large.

TRANSITION MODELS

Model-based reinforcement learning is that agents learn a model from the dynamics and has strong demands on the dynamic model, that is, they can effectively and accurately predict future dynamic conditions. Depending on the model, model-based reinforcement learning can be divided into two types to fit the dynamic environment. They are reinforcement learning based on deterministic models and stochastic models.

A. Deterministic case

The deterministic model means that the agent has already known the transition model structure, and then makes action decisions, such as optimal control and trajectory optimization. Since the environment is known and the goal is to maximize the cumulative reward, then this type of problem becomes the environment giving the agent an initial state s_1, and then the agent making a series of action decisions directly from the environment:

$$a_1, \ldots, a_T = \underset{a_1, \ldots, a_T}{\operatorname{argmax}} \sum_{t=1}^{T} r(s_t, a_t) s.t. s_{t+1} = f(s_t, a_t)$$

.

In deterministic case, Erez *et al.* (Erez *et al.*, 2012) proposed a method combining off-line trajectory optimization and online model predictive control (Garcia *et al.*, 1989) (MPC), which can generate robust controllers for complex periods with unilaterally constrained domains; Mordatch *Et al.* (Mordatch *et al.*, 2012) proposed a physical feature-based model that allows for efficient consideration of dynamic environment aspects in the internal loop of the optimization process; Liu *et al.* (Liu *et al.*, 2014) proposed a motion model, SteadyFlow, to represent adjacent video frames movements to achieve stability. The advantage of using a deterministic dynamic model in model-based reinforcement learning is that the deterministic dynamics can greatly improve the data efficiency and the disadvantage is that this model does not consider the potential changes of the dynamics.

B. Stochastic case

In a stochastic case, in general, we can fit the dynamics with models with Gaussian processes (Engel *et al.*,2005) (GP), neural network models (Fragkiadaki *et al.*, 2015; Nagabandi *et al.*, 2017), or other models such as mixed Gaussian models (Cai *et al.*, 2016).

When a model with a GP(Rasmussen & Williams, 2005) is used to fit an unknown dynamics, based on the known input (x_t,u_t), the agent establishes a Gaussian distribution $p(x'_t|x_t, u_t)$ based on the learning data, and obtains the output as xt+1. McAllister *et al.*(McAllister, van& Rasmussen, 2016) used the GP as the environment learning model p(x'|x, u) to maximize $\sum_t log\, gp(x'_t|x_t, u_t)$; Xuesong *et al.* (Xuesong *et al.*, 2009) transformed the reinforcement learning into a binary classification problem in the continuous state space. Then, based on the classification ability of GP model, the reinforcement learning policy is obtained. Engel *et al.* (Engel *et al.*, 2005) has proposed a GP time difference learning framework, and further considers the randomness and action selection of state transition; Ko *et al.* (Ko *et al.*, 2007) proposed an example application of GP model in reinforcement learning. The advantage of this model is that it can be used effectively and efficiently, and the disadvantage is that such a model is slow to calculate for a system with many data.

When using neural network model to fit the unknown dynamics, the input of the neural network is the matrix of states and actions (x_t, u_t), the output is x'_t, and its loss function is Euclidean distance of the actual state x_t and the output x'_t, Fragkiadaki *et*

al. (Fragkiadaki *et al.*, 2015) proposed a model of a neural network similar to the recurrent neural network's (Graves, 2013) encoder-recurrent-decoder to predict human walking; Nagabandi *et al.*(Nagabandi *et al.*, 2017) parameterized the learned dynamics into a medium-sized deep neural network. The input is the action-state pair, and the output is the difference of the joint states, and the model is combined with the model predictive control (MPC) to achieve a high degree of sample complexity, so that the stable and effective gait can be obtained in various complex motion tasks. The advantage of this type of model is that the model is very expressive and good at using lots of data; but the disadvantage is that the performance is bad when the amount of data is not large enough.

Mixed Gaussian models are very common models in other models to fit unknown dynamics, and such models are used more in robots. The input to train the model is the combination of the current state-action pair and the next combination (x, u, x'), and then obtains the gaussian distribution $p(x_t, u_t, x'_t)$ and takes the conditional distribution to obtain $p(x'_t|x_t, u_t)$. Ha & Schmidhuber (Ha & Schmidhuber, 2018) proposed a model inspired by cognitive systems, which is divided into three parts: the variational self-encoding (Kingma & Welling., 2013) part, the mixed-density (Bishop, 1994) recurrent network (Graves, 2013) part, and the control part. The mixed density recurrent network partially outputs gaussian mixture, which is used to predict the distribution density of the next observation.

C. Model bias

The key challenge for model-based reinforcement learning is model bias, especially when the training is just starting and the data set is small, the learned model is inaccurate and the data efficiency is not high. In response to this problem, Deisenroth & Rasmussen (Deisenroth & Rasmussen, 2011) proposed an algorithm called probabilistic inference for learning control (PILCO), which has a big improvement in learning speed and has performed well in many tasks such as the swinging of the car (Hesse *et al.*, 2018) and the training of the robot arm (Deisenroth, Rasmussen&Fox, 2011). Nevertheless, the algorithm does not have the guarantee of global optimality, and the computational complexity increases exponentially with the dimension of data, which is difficult to apply to high-dimensional systems. Therefore, the researchers proposed some improved algorithms for the PILCO algorithm, using the bayesian depth dynamic model instead of the previous Gaussian model to solve the problem of time correlation without previously considering the model uncertainty between continuous state transitions (Gal *et al.*, 2016); Rowan McAllister *et al.* have used guided exploration (Thrun, 1992) to solve the problem that the PILCO algorithm did not use any possible exploration before (McAllister, van& Rasmussen, 2016) and also modified and extended partial observed MDP (POMDP) combined with PILCO algorithm, and has solved the problem of the influence of noise in the actual state (McAllister & Rasmussen, 2016); Higuera *et al.* (Higuera *et al.*, 2018) have proved that using neural network controller can improve the data efficiency of (Gal *et al.*, 2016).

POLICY METHODS

Trust region policy optimization (Schulman *et al.*, 2015) methods and depth deterministic policy gradient methods (Lillicrap *et al.*, 2015) are typical model-free RL methods to optimize policies. Model-free reinforcement learning methods have many advantages, such as they do not need to model the external dynamics, when the external dynamics is complex, it is the only random method available in the policy search algorithms. The solution to the stochastic policy search problem in reinforcement learning is to use the model to search policies. With the model, using model and model-based optimization can result in higher reward data and take advantages of demonstration to learn.

A. Dynamic programing

Dynamic programing (Sutton & Barto., 1998) (DP) is mainly to solve the problem of how to calculate the optimal policies after having a model that can perfectly simulate the MDP. The core idea of DP is to use the value function as a basis to guide the process of policy search.

In order to get the following policy v_k, the same operation is performed for each state s: the value of the current state s is updated to a new value, and the new value is followed by an old value and an instantaneous expectation reward, summed along all possible state transition probabilities. In this algorithm, each iteration of the operation updates the value of all states in reverse, resulting in a new value. The process is called policy estimation.

The reason why we want to calculate the value function under a policy is because we want to evaluate the policy. Assuming that the value function v_π of a certain policy is known, the method of choosing a better policy to proceed is called policy promotion.

When a policy becomes a better one by policy promotion, then a new policy is generated after policy iteration. Further optimized into a better policy after policy promotion, so that the update sequence shown in Figure 2 can be obtained.

$$\pi_0 \xrightarrow{policy\ estimation} v_{\pi_0} \xrightarrow{policy\ improvement} \pi_1 \xrightarrow{policy\ estimation} v_{\pi_1} \xrightarrow{policy\ improvement} \cdots \xrightarrow{policy\ estimation} \pi_* \xrightarrow{policy\ improvement} v_{\pi_*}$$

Figure 2: Using policy estimation and policy improvement to get a better policy

One disadvantage of policy iteration is that it includes a policy estimation every time during the iteration process, and the policy estimation itself is the iteration after iteration. By the end of the policy estimation, v_k can be guaranteed to converge to v_π accurately, but the calculation speed is very slow. The value iterative algorithm is to improve the policy after evaluating the policy of all the policies in an epoch, and obtain a greater convergence speed.

Song *et al.* (Song *et al.*, 2015) used DP in optimized hybrid energy storage systems to obtain optimal configurations for hybrid energy storage systems for electric city buses including batteries and supercapacitors; Askew (Askew, 1974) applied DP to limit water resource systems, achieving good performance with the possibility of failure; Wall & Fenech (Wall & Fenech, 1965) used DP algorithms to optimize minimum unit power costs in fuel management optimization for nuclear power plants. Although the DP algorithm is a classic algorithm in the optimization algorithm, in practical applications, as the data increasing, it will encounter dimensional disasters and there is no unified standard model for use.

B. Guided policy search

Since direct policy search can be effectively extended to high-dimensional systems, the effectiveness of this approach is greatly reduced for complex policies with hundreds of parameters. Because these methods require a large number of samples and often fall into poor local optimum. So, Sergey Levine proposed the Guided Policy Search (Levine & Koltun, 2013) (GPS) algorithm for this problem. The algorithm performs well for deterministic and linear dynamics, but it is very challenging for nonlinear dynamic systems and complex tasks. Nevertheless, the algorithm has a good starting for the following work.

Subsequently, some improved algorithms of GPS algorithm are proposed, including the cGPS (Levine & Abbeel, 2014) of gaussian controller under unknown dynamic conditions. The algorithm solves the learning problem of gaussian controller under effective unknown dynamic conditions by introducing iterative fitting local dynamic model and learning time-varying linear gaussian controller. A time-varying linear model with iterative modification is proposed to learn a group of trajectories, and then these trajectories are unified into a single control policy to increase the generalization of the dual GPS (Levine, Wagener & Abbeel, 2015). This algorithm can make the robot learn abundant interactive operation skills. The improved GPS algorithm (Levine, Finn, Darrell & Abbeel, 2015) is transformed into supervised learning, and then formalizes as an example of Bregman ADMM (Wang & Banerjee, 2013), the algorithm achieves local optimum and can be used to train the joint system of perceptual system and control system end-to-end. It is proposed to train a complex, high-dimensional policy with approximate mirror descent GPS (Montgomery & Levine, 2016) by alternating between reinforcement learning and supervised learning in the trajectory center, and guiding the connection between the policy search method and the mirror descent method. Sergey Levine proposes a new GPS algorithm which combines the random policy optimization based on path integral with GPS to train the high-dimensional nonlinear neural network policy for vision-based robots and improve their operation skills (Chebotar *et al.*, 2016) and so on.

APPLICATION

An important reason why model-based reinforcement learning is becoming more and more popular is that it is a theoretical tool for studying the behavior of agents. Undoubtedly, it is used by many researchers to build many applications, including robotics, games, and brain science. Model-based reinforcement learning is playing an increasingly important role.

A. Games

The DeepMind team applied model-based reinforcement learning to the Go robot, AlphaGo, the first artificial intelligence defeating the human professional Go players. Subsequently, a new version of AlphaGo Zero (Silver, Schrittwieser *et al.*, 2017) has been introduced, which alternates the optimization policy of policy iteration and Monte Carlo tree search (Chaslot, 2010) in reinforcement learning. This version is different from the old one: AlphaGo Zero completely abandons human knowledge and has defeated the old version of AlphaGo after three days of training from scratch.

The DeepMind team also achieved remarkable achievements in the deep reinforcement learning algorithm based on Atari video games. In 2013, using the improved TD-gammon's, the model-free reinforcement learning algorithm (Mnih, Antonoglou *et al.*, 2013) similar to Q-learning, it relied on continuous trial and error learning to finally become the game master AI system that defeated human professional players. The AI system playing the Atari game, Breakout, is beyond the human level. Later publishing a paper (Mnih, Veness *et al.*, 2015) using the reinforcement learning system to learn how to play 49 Atari games, the system can achieve human level performance in most games, but there is almost no progress in Montezuma's revenge game.

B. Robot

Model-based reinforcement learning has a wide range of applications in the field of robot, such as the application of robotic arms, unmanned aerial vehicle (UAV), and automatic driving. Researchers use a model-based reinforcement learning algorithm and apply it to the robotic arm to accurately grasp the object and place it in a specific location (Deisenroth, Rasmussen & Fox, 2011; Levine, Wagener & Abbeel, 2015), and can also push objects and deal with objects that can't be seen during training (Finn & Levine, 2016). In the field of UAVs, researchers have used the idea that birds can locate objects that heat around them, and developed a glider that can constantly locate and update the location of objects that heat around them (Reddy *et al.*, 2018). The simulation helicopter also trains the task of avoiding obstacles in (Guo, X 2017; Khan & Hebert,

2018). In the field of automatic driving, researchers successfully transfer the data from virtual environment to real environment (Pan *et al.*, 2017; Tan *et al.*, 2018), and can adapt to the sudden change of transfer probability in 3D navigation tasks (Corneil *et al.*, 2018).

C. Brain science

Brain science and cognitive neuroscience have always been the source of inspiration of reinforcement learning, and the source often brings revolutionary success to the reinforcement learning algorithms. Representatives of this direction, such as the DeepMind team's series of papers on memory (Wang *et al.*, 2018; Kirkpatrick *et al.*, 2016; Stachenfeld, Botvinick & Gershman, 2017). Dopamine is a well-known reward signal and is often considered an analogy of the reward used in reinforcement learning algorithms. Patients with Parkinson's disease leading to loss of dopamine are impaired in learning rewards, and many studies have indicated the important role of dopamine in model-free learning (Starkweather & Babayan *et al.*, 2017; Sadacca *et al.*, 2016). However, recent studies have shown that model-based reinforcement learning may also involve dopamine regulation (Smittenaar *et al.*, 2013; Deserno *et al.*, 2015), which increases the likelihood that model-based reinforcement learning may lead to learning disabilities in Parkinson's disease. Sharp *et al.* (Sharp *et al.*, 2015) evaluated that patients with Parkinson's were tested for dopamine replacement therapy and learning in a healthy control group. Surprisingly, disease or medication has no effect on model-free reinforcement learning. In contrast, patients who underwent drug testing showed significant obstacles in model-based reinforcement learning that were corrected by dopaminergic drug therapy. In addition, model-based reinforcement learning is positively correlated with individual measures of working memory performance, and the results suggest that certain learning disabilities in Parkinson's disease may be related to the inability to pursue rewards based on a full representation of the environment.

SUMMARY

Model-based reinforcement learning performs better in data efficiency than model-free reinforcement learning; but suffers from significant bias, since complex unknown dynamics cannot always be modeled accurately enough to produce effective policies. Model-free methods have the advantage of handling arbitrary dynamical systems with minimal bias, but tend to be substantially less sample-efficient (Chebotar *et al.*, 2017). It can be seen that although the research and application of model-based reinforcement learning has entered a new height compared to model-free reinforcement learning, it is extremely important for robotics, games and brain science, but relatively, these areas are not mature enough, there are still many problems to be solved, and the active participation and collaboration of researchers is needed.

ACKNOWLEDGEMENT

This work was supported by Beijing Municipal Commission of Science and Technology (Grant No. Z171100000117018), and Communication University of China (Grant Nos. CUC18A001, CUC18A003-3, 3132014XNG1454，3132015XNG1531, 3132017XNG1724).

REFERENCES

[1] Albus, J. S. (1975). A new approach to manipulator control: The cerebellar model articulation controller (CMAC). Journal of Dynamic Systems, Measurement, and Control, 97(3), 220-227.
[2] Askew, A. J. (1974). Chance-constrained dynamic programing and the optimization of water resource systems. Water Resources Research, 10(6), 1099-1106.
[3] Bishop, C. M. (1994). Mixture density networks (p. 7). Technical Report NCRG/4288, Aston University, Birmingham, UK.
[4] Cai, C., Chigansky, P., & Kleptsyna, M. (2016). Mixed Gaussian processes: a filtering approach. The Annals of Probability, 44(4), 3032-3075.
[5] Chaslot, G. (2010). Monte-carlo tree search.
[6] Chebotar, Y., Hausman, K., Zhang, M., Sukhatme, G., Schaal, S., & Levine, S. (2017). Combining model-based and model-free updates for trajectory-centric reinforcement learning.
[7] Chebotar, Y., Kalakrishnan, M., Yahya, A., Li, A., Schaal, S., & Levine, S. (2016). Path integral guided policy search. 3381-3388.
[8] Chen, F. C., & Chang, C. H. (1996). Practical stability issues in CMAC neural network control systems. IEEE Transactions on control systems technology, 4(1), 86-91.
[9] Corneil, D., Gerstner, W., & Brea, J. (2018). Efficient model-based deep reinforcement learning with variational state tabulation.
[10] Daw, N. D., Niv, Y., & Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. Nature neuroscience, 8(12), 1704.
[11] Deisenroth, M. P., Fox, D., & Rasmussen, C. E. (2013). Gaussian processes for data-efficient learning in robotics and control. IEEE Transactions on Pattern Analysis & Machine Intelligence, (1), 1.
[12] Deisenroth, M. P., Rasmussen, C. E., & Fox, D. (2011). Learning to control a low-cost manipulator using data-efficient reinforcement learning.
[13] Deisenroth, M., & Rasmussen, C. E. (2011). PILCO: A model-based and data-efficient approach to policy search. In Proceedings of the 28th International Conference on machine learning (ICML-11) (pp. 465-472).

[14] Deserno, L., Huys, Q. J., Boehme, R., Buchert, R., Heinze, H. J., & Grace, A. A., *et al.* (2015). Ventral striatal dopamine reflects behavioral and neural signatures of model-based control during sequential decision making. Proc Natl Acad Sci U S A, 112(5), 1595-600.
[15] Dongbin Zhao, Kun Shao, Yuanheng Zhu, Dong Li, Yaran Chen, Haitao Wang, … & Chenghong Wang. (2016). Deep reinforcement review: Also on the development of computer Go. Control theory and application, 33(6), 701-717. (in Chinese)
[16] Duan, Y., Liu, Q., & Xu, X. (2007). Application of reinforcement learning in robot soccer. Engineering Applications of Artificial Intelligence, 20(7), 936-950.
[17] Engel, Y., Mannor, S., & Meir, R. (2005, August). Reinforcement learning with Gaussian processes. In Proceedings of the 22nd international conference on Machine learning (pp. 201-208). ACM.
[18] Erez, T., Tassa, Y., & Todorov, E. (2012). Infinite-horizon model predictive control for periodic tasks with contacts. Robotics: Science and systems VII, 73.
[19] Finn, C., & Levine, S. (2016). Deep visual foresight for planning robot motion. 2786-2793.
[20] Fragkiadaki, K., Levine, S., Felsen, P., & Malik, J. (2015). Recurrent network models for human dynamics. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4346-4354).
[21] Gal, Y., McAllister, R., & Rasmussen, C. E. (2016, April). Improving PILCO with Bayesian neural network dynamics models. In Data-Efficient Machine Learning workshop, ICML.
[22] Garcia, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control: theory and practice—a survey. Automatica, 25(3), 335-348.
[23] Glanz, F. H., & Miller, W. T. (1988, February). Shape recognition using a CMAC based learning system. In Intelligent Robots and Computer Vision VI (Vol. 848, pp. 294-299). International Society for Optics and Photonics.
[24] Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.
[25] Gu, S., Lillicrap, T., Sutskever, I., & Levine, S. (2016, June). Continuous deep q-learning with model-based acceleration. In International Conference on Machine Learning (pp. 2829-2838).
[26] Guo, X., Guo, X., Guo, X., Guo, X., & Guo, X. (2017). Trajectory generation using reinforcement learning for autonomous helicopter with adaptive dynamic movement primitive. Proceedings of the Institution of Mechanical Engineers Part I Journal of Systems & Control Engineering,231(5), 095965181668442.
[27] Ha, D., & Schmidhuber, J. (2018). World models.
[28] Hamaya, M., Matsubara, T., Noda, T., Teramae, T., & Morimoto, J. (2016, May). Learning assistive strategies from a few user-robot interactions: Model-based reinforcement learning approach. In Robotics and Automation (ICRA), 2016 IEEE International Conference on (pp. 3346-3351). IEEE.
[29] Hang Li. (2012). Statistical learning method. (in Chinese)
[30] Herold, D. J., Miller, W. T., Kraft, L. G., & Glanz, F. H. (1989, March). Pattern recognition using a CMAC based learning system. In Automated Inspection and High-Speed Vision Architectures II (Vol. 1004, pp. 84-91). International Society for Optics and Photonics.
[31] Hesse, M., Timmermann, J., Hüllermeier, E., & Trächtler, A. (2018). A Reinforcement Learning Strategy for the Swing-Up of the Double Pendulum on a Cart. Procedia Manufacturing, 24, 15-20.
[32] Higuera, J. C. G., Meger, D., & Dudek, G. (2018). Synthesizing Neural Network Controllers with Probabilistic Model based Reinforcement Learning. arXiv preprint arXiv:1803.02291.
[33] Jong, N. K., & Stone, P. (2007, May). Model-based function approximation in reinforcement learning. In Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems (p. 95). ACM.
[34] Jouffe, L. (1998). Fuzzy inference system learning by reinforcement methods. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 28(3), 338-355.
[35] Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. Journal of artificial intelligence research, 4, 237-285.
[36] Kamalapurkar, R., Walters, P., & Dixon, W. E. (2016). Model-based reinforcement learning for approximate optimal regulation. Automatica, 64, 94-104.
[37] Khan, A., & Hebert, M. (2018). Learning safe recovery trajectories with deep neural networks for unmanned aerial vehicles. IEEE Aerospace Conference (pp.1-9). IEEE.
[38] Kim, Y. H., & Lewis, F. L. (2000). Optimal design of CMAC neural-network controller for robot manipulators. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 30(1), 22-31.
[39] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.
[40] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., & Rusu, A. A., *et al.* (2016). Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci U S A, 114(13), 3521-3526.
[41] Ko, J., Klein, D. J., Fox, D., & Haehnel, D. (2007, April). Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In Robotics and Automation, 2007 IEEE International Conference on (pp. 742-747). IEEE.

[42] Kraft, L. G., & Campagna, D. P. (1989, June). A comparison of CMAC neural network and traditional adaptive control systems. In American Control Conference, 1989 (pp. 884-891). IEEE.
[43] Kuvayev, L., & Sutton, R. S. (1996). Model-based reinforcement learning with an approximate, learned model. In in Proceedings of the Ninth Yale Workshop on Adaptive and Learning Systems.
[44] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436.
[45] Levine, S., & Abbeel, P. (2014). Learning neural network policies with guided policy search under unknown dynamics. Advances in Neural Information Processing Systems, 1071-1079.
[46] Levine, S., & Koltun, V. (2013). Guided Policy Search. International Conference on Machine Learning (pp.1-9).
[47] Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2015). End-to-end training of deep visuomotor policies. Journal of Machine Learning Research,17(1), 1334-1373.
[48] Levine, S., Wagener, N., & Abbeel, P. (2015). Learning contact-rich manipulation skills with guided policy search. IEEE International Conference on Robotics and Automation (Vol.2015, pp.156-163). IEEE.
[49] Li, Y. (2017). Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274.
[50] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.
[51] Liu, S., Yuan, L., Tan, P., & Sun, J. (2014). Steadyflow: Spatially smooth optical flow for video stabilization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4209-4216).
[52] McAllister, R., & Rasmussen, C. E. (2016). Data-efficient reinforcement learning in continuous-state POMDPs. arXiv preprint arXiv:1602.02523.
[53] McAllister, R., van der Wilk, M., & Rasmussen, C. E. Data-Efficient Policy Search using PILCO and Directed-Exploration.
[54] Miller, W. T., Glanz, F. H., & Kraft, L. G. (1990). Cmas: An associative neural network alternative to backpropagation. Proceedings of the IEEE, 78(10), 1561-1567.
[55] Miller, W. T., Hewes, R. P., Glanz, F. H., & Kraft, L. G. (1990). Real-time dynamic control of an industrial manipulator using a neural network-based learning controller. IEEE Transactions on Robotics and Automation, 6(1), 1-9.
[56] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., & Wierstra, D., *et al.* (2013). Playing atari with deep reinforcement learning. Computer Science.
[57] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., & Bellemare, M. G., *et al.* (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529.
[58] Montgomery, W. H., & Levine, S. (2016). Guided policy search via approximate mirror descent. In Advances in Neural Information Processing Systems (pp. 4008-4016).
[59] Mordatch, I., Todorov, E., & Popović, Z. (2012). Discovery of complex behaviors through contact-invariant optimization. ACM Transactions on Graphics (TOG), 31(4), 43.
[60] Nagabandi, A., Kahn, G., Fearing, R. S., & Levine, S. (2017). Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. arXiv preprint arXiv:1708.02596.
[61] Niv, Y., Daw, N. D., Joel, D., & Dayan, P. (2007). Tonic dopamine: opportunity costs and the control of response vigor. Psychopharmacology, 191(3), 507-520.
[62] Pan, X., You, Y., Wang, Z., & Lu, C. (2017). Virtual to real reinforcement learning for autonomous driving.
[63] Polydoros, A. S., & Nalpantidis, L. (2017). Survey of model-based reinforcement learning: applications on robotics. Journal of Intelligent & Robotic Systems, 86(2), 1-21.
[64] Quan Liu, Jianwei Zhai, Zongchang Zhang, Qian Zhou, Peng Zhang & Jin Xu.(2018). Deep reinforcement review. Journal of Computer, 41(1), 1-27. (in Chinese)
[65] Rao, Y., Lu, J., & Zhou, J. (2017, October). Attention-aware deep reinforcement learning for video face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3931-3940).
[66] Rasmussen, C. E., & Williams, C. K. I. (2005). Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press.
[67] Ray, S., & Tadepalli, P. (2010). Model-based reinforcement learning. Encyclopedia of Machine Learning, 690-693.
[68] Reddy, G., Wong-Ng, J., Celani, A., Sejnowski, T. J., & Vergassola, M. (2018). Glider soaring via reinforcement learning in the field. Nature, 1.
[69] Sadacca, B. F., Jones, J. L., & Schoenbaum, G. (2016). Midbrain dopamine neurons compute inferred and cached value prediction errors in a common framework. eLife,5,(2016-03-03), 5.
[70] Scheffler, K., & Young, S. (2002, March). Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In Proceedings of the second international conference on Human Language Technology Research (pp. 12-19). Morgan Kaufmann Publishers Inc..
[71] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In International Conference on Machine Learning (pp. 1889-1897).

[72] Sharp, M. E., Foerde, K., Daw, N. D., & Shohamy, D. (2015). Dopamine selectively remediates 'model-based' reward learning: a computational approach. Brain, 139(2), 355-364.
[73] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), 484.
[74] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., & Guez, A., *et al*. (2017). Mastering the game of go without human knowledge. Nature, 550(7676), 354-359.
[75] Smittenaar, P., Fitzgerald, T. H. B., Romei, V., Wright, N. D., & Dolan, R. J. (2013). Disruption of dorsolateral prefrontal cortex decreases model-based in favor of model-free control in humans. Neuron, 80(4), 914-919.
[76] Song, Z., Hofmann, H., Li, J., Han, X., & Ouyang, M. (2015). Optimization for a hybrid energy storage system in electric vehicles using dynamic programing approach. Applied Energy, 139, 151-162.
[77] Stachenfeld, K. L., Botvinick, M. M., & Gershman, S. J. (2017). The hippocampus as a predictive map. Nature Neuroscience, 20(11), 1643.
[78] Stadie, B. C., Levine, S., & Abbeel, P. (2015). Incentivizing exploration in reinforcement learning with deep predictive models. arXiv preprint arXiv: 1507.00814.
[79] Starkweather, C. K., Babayan, B. M., Uchida, N., & Gershman, S. J. (2017). Dopamine reward prediction errors reflect hidden-state inference across time. Nature Neuroscience, 20(4), 581-589.
[80] Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction. MIT press.
[81] Tan, B., Xu, N., & Kong, B. (2018). Autonomous driving in reality with reinforcement learning and image translation.
[82] Thrun, S. B. (1992). Efficient exploration in reinforcement learning.
[83] Van Hasselt, H., Guez, A., & Silver, D. (2016, February). Deep Reinforcement Learning with Double Q-Learning. In AAAI(Vol. 2, p. 5).
[84] Wall, I., & Fenech, H. (1965). The application of dynamic programing to fuel management optimization. Nuclear Science & Engineering, 22.
[85] Wang, H., & Banerjee, A. (2013). Bregman alternating direction method of multipliers. Mathematics, 2816-2824.
[86] Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., & Leibo, J. Z., *et al*. (2018). Prefrontal cortex as a meta-reinforcement learning system. Nature Neuroscience, 21(6).
[87] Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., & De Freitas, N. (2015). Dueling network architectures for deep reinforcement learning. arXiv preprint arXiv:1511.06581.
[88] Wenji Zhou, & Yang Yu. (2017). Summary of stratified reinforcement learning. Journal of Intelligent Systems, 12(5), 590-594. (in Chinese)
[89] Xian Guo, Yongchun Fang. (2018). In-depth introduction to the principle of reinforcement learning[M]. Electronic Industry Press. (in Chinese)
[90] Xuesong Wang, Yiyang Zhang, & Yuhu Cheng. (2009). Continuous space reinforcement learning based on Gaussian process classifier. Electronic Journal, 37(6), 1153-1158. (in Chinese)