

DYNAMICAL SYMMETRY MODELING BASED OFFLINE REINFORCEMENT LEARNING METHOD AND APPARATUS

FIELD

[0001] The present application relates to the field of machine learning, and in particular to a dynamical symmetry modeling based offline reinforcement learning method and apparatus.

BACKGROUND

[0002] At present, most of reinforcement learning algorithms may learn a good policy only after going through a large number of attempts and errors in a simulation environment, such as in the field of games and the field of robotics. However, in real-world scenarios, such as autonomous vehicles and control of complex industrial system, there is no perfect simulation environment, but only a batch of pre-collected environment interaction data, namely offline data.

[0003] In offline reinforcement learning, since an intelligent agent cannot interact with environment, the difficulty of learning is greatly increased, and a series of problems are derived. Traditional offline reinforcement learning modes include soft actor-critic (SAC) algorithm, conservative Q-learning (CQL) algorithm, and best-action imitation learning (BAIL) algorithm.

[0004] The SAC algorithm introduces an idea of entropy into a traditional actor-critic framework, and allows the training process of intelligent agent to be more stable and achieves a better outcome performance by soft update. Although the SAC algorithm may be directly transferred to the scenario of offline reinforcement learning, with a poor adaptability and fatal problems such as Q function value divergence often occurs in experiments, the SAC algorithm is not suitable for direct application.

[0005] The CQL algorithm applies additional limits to the Q function, strictly controls the updates of Q function outside a data set, and avoids the problem of overestimation of Q function by limiting the Q function. However, the Q function corresponding to the CQL algorithm is often pessimistic and the obtained policy is too conservative, which is far from the optimal policy. Besides, the CQL algorithm is sensitive to hyperparameters, and requires fine tuning for hyperparameters to achieve a better effect.

[0006] The BAIL algorithm utilizes an idea of imitation learning, completely abandoned the actor-critic framework commonly used in reinforcement learning, does not include a Q function, and fundamentally avoids the error problem of Q function. In addition, the BAIL algorithm screens out better data points from the upper envelope of all data points in the training data set, and utilizes the better data points for imitation learning to obtain a better policy.

[0007] Although the BAIL algorithm is not limited by the Q function, the manner of calculating the upper envelope requires to estimate a future expected return of each point in a data set, such estimation has a certain error, and the error becomes greater especially in a scenario of infinite time sequence.

[0008] In the related art, a problem of extracting effective information from offline data and assisting reinforcement learning to obtain a policy for maximizing long-term reward is important to be solved urgently in the field of machine learning.

SUMMARY

[0009] The present application provides a dynamical symmetry modeling based offline reinforcement learning method and apparatus, which overcomes the problem that in related arts effective information cannot be extracted from offline data and assist reinforcement learning to obtain a policy for maximizing long-term reward, and achieves the optimization of behavior policy.

[0010] The present application provides a dynamical symmetry modeling based offline reinforcement learning method, applied to policy optimization, including:

obtaining an offline data set;

processing an original state-action pair in the offline data set by an encoder and a bidirectional dynamical model to obtain a low-dimensional state-action pair and a current time inversion state variation; and

inputting the low-dimensional state-action pair to a value network, and obtaining a value of a state-action value function corresponding to the original state-action pair,

where the current time inversion state variation includes a forward time inversion state variation and a reverse time inversion state variation.

[0011] In an embodiment, the processing the original state-action pair in the offline data set by the encoder and the bidirectional dynamical model to obtain the low-dimensional state-action pair and the current time inversion state variation includes:

processing the original state-action pair by the encoder to obtain the low dimensional state-action pair; and

processing a low-dimensional state-action pair at a current moment and a low-dimensional state-action pair at a next moment respectively by the bidirectional dynamical model, and obtaining the forward time inversion state variation and the reverse time inversion state variation through a preset loss function.

[0012] In an embodiment, the preset loss function includes a dynamical loss function, a low-dimensional dynamical loss function and a reconstruction loss function,

where formulas of the dynamical loss function, the low-dimensional dynamical loss function and the reconstruction loss function are as follows:

$$L_{dl} = \|\Delta X(s, s') - D_{\varphi}(\dot{Z}_{\theta}(X))\|_2^2;$$

$$L_{ldl} = \|E_{\varphi}(\Delta X(s, s')) - \dot{Z}_{\theta}(X)\|_2^2;$$

$$L_{rl} = \|X(s, a) - D_{\varphi}(Z(X))\|_2^2,$$

where s is a state in the offline data set, a is an action in the state of s in the offline data set, $\Delta X(s, s')$ is a state variation at the next moment, s' is a state at the next moment in the offline data set, θ and φ are hyper-parameters, $\dot{Z}_{\theta}(X)$ is the forward time inversion state variation obtained by processing of the bidirectional dynamic model, $D_{\varphi}(\dot{Z}_{\theta}(X))$ is a state-action variation at the next moment obtained by

decoding the forward time inversion state variation, $E_{\varphi}(\Delta X(s, s'))$ is the forward time inversion state variation obtained by directly encoding of the encoder, $X(s, a)$ is the

original state-action pair in the offline data set and $D_{\varphi}(Z(X))$ is the original state-action pair obtained by decoding the low-dimensional state-action pair.

[0013] In an embodiment, the inputting the low-dimensional state-action pair to the value network, and obtaining the value of the state-action value function corresponding to the original state-action pair includes:

- calculating a low-dimensional policy state variation at the next moment based on a low-dimensional state and a low-dimensional policy output action in the low-dimensional state-action pair;

- calculating the forward time inversion state variation based on the low-dimensional state and a low-dimensional representation of an original state at the next moment in the low-dimensional state-action pair; and

- obtaining the value of the state-action value function by taking a difference value between the low-dimensional policy state variation at the next moment and the forward time inversion state variation as a preset constraint item.

[0014] In an embodiment, the method includes:

- obtaining a behavior policy corresponding to an original state in the original state-action pair by taking the value of the state-action value function and the difference value between a low-dimensional policy action and the low-dimensional action in the low-dimensional state-action pair as a constraint item; or,

- obtaining a behavior policy corresponding to an original state in the original state-action pair by taking the value of the state-action value function, and the difference value between the forward time inversion state variation obtained based on the low-dimensional state-action pair and the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair as a constraint item; or,

- obtaining a behavior policy corresponding to an original state in the original state-action pair by taking the value of the state-action value function, the difference value between the forward time inversion state variation obtained based on the low-dimensional state-action pair and the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair, and the difference value between the low-dimensional policy action and the low-dimensional action in the low-dimensional state-action pair as a constraint item; or,

- obtaining a behavior policy corresponding to an original state in the original state-action pair by taking the value of the state-action value function, the difference value between the forward time inversion state variation obtained based on the low-dimensional state-action pair and the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair, the difference value between the low-dimensional policy action and the low-dimensional action in the low-dimensional state-action pair, and the difference value between the low-dimensional policy action and an original action in the original state-action pair as a constraint item.

[0015] In an embodiment, the bidirectional dynamical model is obtained by training

a time inversion loss function.

[0016] The present application further provides a dynamical symmetry modeling based offline reinforcement learning apparatus, applied to policy optimization, including:

- a data obtaining module, used for obtaining an offline data set;

- a data processing module, used for processing an original state-action pair in the offline data set by an encoder and a bidirectional dynamical model to obtain a low-dimensional state-action pair and a current time inversion state variation; and

- a value obtaining module, used for inputting the low-dimensional state-action pair to a value network, and obtaining a value of a state-action value function corresponding to the original state-action pair,

where the current time inversion state variation includes a forward time inversion state variation and a reverse time inversion state variation.

[0017] The present application further provides an electronic device, including:

- a processor;

- a memory; and

- a computer program stored on the memory and executable on the processor,

where the computer program, when executed by the processor, causes the electronic device to perform any one of the above-mentioned steps of the dynamical symmetry modeling based offline reinforcement learning method.

[0018] The present application further provides a non-transitory computer-readable storage medium storing computer programs, when executed by a processor, cause the processor to perform any one of the above-mentioned steps of the dynamical symmetry modeling based offline reinforcement learning method.

[0019] The present application further provides a computer program product including computer programs, when executed by a processor, cause the processor to perform any one of the above-mentioned steps of the dynamical symmetry modeling based offline reinforcement learning method.

[0020] The dynamical symmetry modeling based offline reinforcement learning method provided by the present application, processing the original state-action pair in the offline data set by an encoder and a bidirectional dynamical model, and obtaining the low-dimensional representation of the original state-action pair, the forward time inversion state variation and the reverse time inversion state variation, and then inputting the low-dimensional representation of the original state-action pair in the value network, the value of the state-action value function corresponding to the original state-action pair can be obtained. The method employs the bidirectional dynamical model to process the offline data, and extracts effective information from the offline data, namely the forward time inversion state variation and reverse time inversion state variation, where the effective information has extremely high generalizability and interpretability. The value of the state-action value function obtained by taking the effective information as an input of the value network can effectively assist a reinforcement learning to obtain a policy that maximizes a long-term reward and realize an optimization of the behavior policy.

BRIEF DESCRIPTION OF DRAWINGS

[0021] In order to illustrate the solutions of the embodiments according to the present application, the accompanying drawings used in the description of the embodiments are briefly introduced below. It should be noted that the drawings in the following description are only some embodiments of the present application. For those of ordinary skill in the art, other drawings can be obtained according to these drawings without creative effort.

FIG. 1 is a schematic flowchart of a dynamical symmetry modeling based offline reinforcement learning method according to an embodiment of the present application; FIG. 2 is a schematic diagram of a prediction for a bidirectional dynamical model according to an embodiment of the present application;

FIG. 3 is a schematic diagram of obtaining a value of a state-action value function according to an embodiment of the present application;

FIG. 4 is a schematic diagram of an overall process of a dynamical symmetry modeling based offline reinforcement learning method according to an embodiment of the present application;

FIG. 5 is a schematic structural diagram of a dynamical symmetry modeling based offline reinforcement learning apparatus according to an embodiment of the present application; and

FIG. 6 is a schematic structural diagram of an electronic device according to an embodiment of the present application.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0022] The solutions of the embodiments of the present application are clearly and completely described below in combination with the accompanying drawings of the embodiments of the present application. The embodiments described are a part of the embodiments of the present application, rather than all the embodiments. Based on the embodiments of the present application, all other embodiments obtained by those of ordinary skill in the art without creative labor are within the scope of the protection of the present application.

[0023] FIG. 1 is a schematic flowchart of a dynamical symmetry modeling based offline reinforcement learning method according to an embodiment of the present application. As shown in FIG. 1, the dynamical symmetry modeling based offline reinforcement learning method includes the following steps.

[0024] S101, obtaining an offline data set.

[0025] In an embodiment, the offline data set is consisted of original state-action pairs at different moments. The original state-action pair includes an original state and an original action, and there is a one-to-one correspondence between the original state and the original action. In addition, the offline data set may further include a reward corresponding to the original state-action pair.

[0026] It should be noted that the offline data set may be simulation data obtained by a simulation or interaction data based on a real environment, and the offline data set may include both type of the simulation data and the interaction data.

[0027] It should be further noted that all data in the offline data set may be directly obtained from an existing large database, such as Robomimic database and D4RL

database. The Robomimic database is an offline database related to a robotic arm. The D4RL database collects large data sets, including records of intelligent agents in interactive environments, such as interaction records of autonomous driving Carla, AntMaze and Mujoco. The D4RL databases also classifies the data sets in simple and complex manners, and the types are also very rich.

[0028] Taking autonomous driving as an example, when the offline data set is originates from data records of autonomous driving, vehicle driving environment information is the original state in the offline data set, and vehicle operation behavior information is the original action in the offline data set. Of course, the offline data set may also include reward information corresponding to the vehicle driving environment information and the vehicle operation behavior information.

[0029] S102, processing the original state-action pair in the offline data set by an encoder and a bidirectional dynamical model to obtain a low-dimensional state-action pair and a current time inversion state variation, where the current time inversion state variation includes a forward time inversion state variation and a reverse time inversion state variation.

[0030] The encoder may be a convolution-based encoding network, used for compressing the original state-action pair in the offline data set and relevant data obtained by extending the original state-action pair, which compresses high-dimensional original data into low-dimensional data. The relevant data obtained by extending the original state-action pair includes a state-action variation at a next moment, namely variations of the original state and original action at the next moment.

[0031] It should be noted that, different from a general deep learning prediction model, the bidirectional dynamical model is obtained by approximating a total differential equation of the current system. The total differential equation describes the equation of how the state changes with time in a current dynamical system. Therefore, the bidirectional dynamical model focuses on a state variation in a time dimension. The general deep learning prediction model fits a data transition probability based on a data distribution, and does not pay attention on a characteristic of how data changes in the time dimension. The bidirectional dynamical model is a deep learning network model with a preset number of network layers. The preset number can be set based on an actual need, and no specific setting is made herein.

[0032] In the embodiment, a reliability of the bidirectional dynamical model for determining a state may be used for increasing a generalizability and a robustness of an algorithm. The bidirectional dynamical model is also used for enhancing of reliability data to a certain extent and expanding data coverage of original data, which enhances an effect of a reinforcement learning algorithm.

[0033] In an embodiment, the bidirectional dynamical model is able to perform a bidirectional prediction, that is, the bidirectional dynamical model can not only predict a variation at a next moment state relative to a current moment state based on an action made at a current moment state, but also can predict a variation from the current moment state to the next moment state based on an action corresponding to the current moment state made at the next moment state, as such, a prediction of forward state variation and reverse state variation on a time axis is achieved.

[0034] It should be noted that, comparing with the original state-action pair of an original dimension, the forward time inversion state variation obtained by processing of the bidirectional dynamical model has more generalizability and interpretability.

[0035] The generalizability refers to that the low-dimensional data obtained by processing of the bidirectional dynamical model may be applied to a variety of scenarios, and accurate predictions may be made based on the low-dimensional data. For example, in the embodiment, the accurate prediction of the forward time inversion state variation and a value of a state-action value function corresponding to the low-dimensional data. The interpretability refers to that each parameter in the low-dimensional data obtained by processing of the bidirectional dynamical model can be interpreted.

[0036] It should be noted that, based on the step S101 of obtaining the offline data set, the original state-action pairs in the offline data set are extracted and processed by the encoder and the bidirectional dynamical model, to obtain the low-dimensional state-action pairs, the forward time inversion state variations and the reverse time inversion state variations. The low-dimensional state-action pair is a low-dimensional representation of the original state-action pair, the forward time inversion state variation is a state variation of a corresponding action made in the state at the current moment, and the reverse time inversion state variation is a state variation of the action corresponding to the current moment made in the next moment state.

[0037] It should be noted that the forward time inversion state variation and the reverse time inversion state variation are low-dimensional representations.

[0038] It should also be noted that processing the original state-action pair by the encoder and the bidirectional dynamical model also includes processing the data obtained by extending the original state-action pair, where the data obtained by extending the original state-action pair may be a state-action variation at the next moment.

[0039] S103, inputting the low-dimensional state-action pair to a value network to obtain a value of a state-action value function corresponding to the original state-action pair.

[0040] The value network in this step refers to a state action value network, and is used for obtaining a value of performing a certain action in a certain state, namely obtaining the value of the state-action value function corresponding to each state-action pair.

[0041] It should be noted that, based on the low-dimensional state-action pair, the forward time inversion state variation and the reverse time inversion state variation obtained in step S102, the inputting the low-dimensional state-action pair to the value network can obtain the value of the state-action value function corresponding to the original state-action pair.

[0042] It should be noted that the forward time inversion state variation and the reverse time inversion state variation obtained in step S103 are used for setting a corresponding policy constraint item to obtain an optimized behavior policy.

[0043] Taking autonomous driving as an example, after obtaining the value of the state-action value function corresponding to the vehicle driving environment

information and vehicle operation behavior information, a policy network may be optimized based on the value of the state-action value function to obtain a behavior policy with a maximized reward, where the behavior policy refers to a vehicle operation behavior. The autonomous driving of the vehicle is more efficiently and accurately controlled based on the optimized behavior policy.

[0044] In the embodiment, by processing the original state-action pair in the offline data set by an encoder and a bidirectional dynamical model, and obtaining the low-dimensional representation of the original state-action pair, the forward time inversion state variation and the reverse time inversion state variation, and then inputting preset input data including the low-dimensional representation of the original state-action pair and/or the time inversion state variation to the value network to obtain the value of the state-action value function corresponding to the original state-action pair, this method employs the bidirectional dynamical model to process the offline data, and extracts effective information from the offline data, namely the forward time inversion state variation and reverse time inversion state variation, where the effective information has high generalizability and interpretability. As such the value of the state-action value function obtained by taking the effective information as an input of the value network may effectively assist a reinforcement learning to obtain a policy with maximized long-term reward and realize an optimization for the behavior policy.

[0045] Based on the above embodiment, the processing the original state-action pair in the offline data set by the encoder and the bidirectional dynamical model to obtain the low-dimensional state-action pair, the forward time inversion state variation and the reverse time inversion state variation includes: processing the original state-action pair by the encoder to obtain the low-dimensional state-action pair; processing the low-dimensional state-action pair and the low-dimensional state at the next moment respectively by the bidirectional dynamical model, and obtaining a corresponding forward time inversion state variation and a corresponding reverse time inversion state variation through a preset loss function.

[0046] It should be noted that the processing the original state-action pair in the offline data set by the encoder and the bidirectional dynamical model refers to that, compressing the original state-action pair by the encoder to obtain the low-dimensional representation of the original state-action pair, which includes the low-dimensional representation of the original state-action pair at different moments, such as, the current moment and the next moment.

[0047] After obtaining the low-dimensional state-action pair, the forward time inversion state variation may be predicted by processing the low-dimensional state-action pair at the current moment using the bidirectional dynamical model; and the reverse time inversion state variation may be obtained by processing the low-dimensional state at the next moment using the bidirectional dynamical model.

[0048] The processing the low-dimensional state-action pair and the low-dimensional state at the next moment by taking the bidirectional dynamical model may be achieved through the preset loss function.

[0049] The formula of the preset loss function is as follows:

$$L_{total} = \lambda_1 L_{ldl} + \lambda_2 L_{dl} + \lambda_3 L_{rl},$$

where L_{ldl} is a dynamical loss function, L_{dl} is a low-dimensional dynamical loss function, L_{rl} is a reconstruction loss function, λ_1 , λ_2 and λ_3 are weight coefficients corresponding to the aforementioned three loss functions respectively.

[0050] It should be noted that, the preset loss function includes the dynamical loss function, the low-dimensional dynamical loss function and the reconstruction loss function.

[0051] Where the dynamical loss function is used for ensuring that the forward time inversion state variation can be decoded back to an original state-action variation at the next moment. The formula of the dynamical loss function is as follows:

$$L_{dl} = \|\Delta X(s, s') - D_\varphi(\dot{Z}_\theta(X))\|_2^2,$$

where s is a state in the offline data set, a is an action in the state of s in the offline data set, $\Delta X(s, s')$ is a state variation at the next moment, s' is the state at the next moment in the offline data set, θ and φ are hyper-parameters, $\dot{Z}_\theta(X)$ is the forward time inversion state variation obtained by processing of the bidirectional dynamic model, $D_\varphi(\dot{Z}_\theta(X))$ is the state-action variation at the next moment obtained by decoding the forward time inversion state variation.

[0052] The low-dimensional dynamical loss function is used for ensuring the accuracy of the forward time inversion state variation $\dot{U}_\theta(X)$ obtained by prediction of the bidirectional dynamical model. The formula of the low-dimensional dynamical loss function is as follows:

$$L_{ldl} = \|E_\varphi(\Delta X(s, s')) - \dot{Z}_\theta(X)\|_2^2,$$

where $E_\varphi(\Delta X(s, s'))$ is the forward time inversion state variation obtained by directly encoding of the encoder. The low-dimensional dynamical loss function takes the forward time inversion state variation $E_\varphi(\Delta X(s, s'))$ obtained by directly encoding of the encoder as a positive label, the closer $\dot{Z}_\theta(X)$ is to $E_\varphi(\Delta X(s, s'))$, the higher accuracy of predicted $\dot{Z}_\theta(X)$ is.

[0053] The reconstruction loss function is used for ensuring that the original state-action pair obtained by decoding the low-dimensional state-action pair can be reconstructed back to the original state-action pair in the offline data set. The formula of the reconstruction loss function is as follows:

$$L_{rl} = \|X(s, a) - D_\varphi(Z(X))\|_2^2,$$

where $X(s, a)$ is the original state-action pair in the offline data set, $D_\varphi(Z(X))$ is the original state-action pair obtained by decoding the low-dimensional state-action pair.

[0054] The preset loss function can be expanded as:

$$L_{total} = \lambda_1 \|E_\varphi(\Delta X(s, a)) - \dot{Z}_\theta(X)\|_2^2 + \lambda_2 \|\Delta X(s, a) - D_\varphi(\dot{Z}_\theta(X))\|_2^2 +$$

$$\lambda_3 \|X(s, a) - D_\varphi(Z(X))\|_2^2.$$

[0055] FIG. 2 is a schematic diagram of a prediction for a bidirectional dynamical model according to an embodiment of the present application and further explains the process of step S102 in FIG. 1. As shown in FIG. 2, firstly, based on the offline data set, adding the state s' at the next moment to the original state-action pair $X(s, a)$ extracted from the offline data set, namely $X(s, a, s')$ is extracted from the offline data set, and based on the original state-action pairs at different moments, subtracting the original state s at the current moment and the original state s' at the next moment to obtain the state variation at the next moment.

[0056] The low-dimensional state-action pair $Z(X)$, namely $Z(s, a)$, is obtained by processing the original state-action pair by the encoding layer (i.e. the encoder mentioned above), and $Z(s', a)$ is obtained by processing the original state s' at the next moment and the original action a at the current moment through the encoder.

[0057] Secondly, the low-dimensional state-action pair $Z(s, a)$ is processed by the bidirectional dynamical model, that is, inputting the low-dimensional state-action pair $Z(s, a)$ to the bidirectional dynamical model, and the forward time inversion state variation $z_\theta(\dot{s})$ may be output by the bidirectional dynamical model.

[0058] In a same manner, the low-dimensional state at the next moment and the low-dimensional representation of the action at the current moment are processed by the bidirectional dynamical model, that is, $Z(s', a)$ in FIG. 2 is inputted to the bidirectional dynamical model, and the reverse time inversion state variation $-z_\beta(\dot{s})$ may be output by the bidirectional dynamical model.

[0059] In an embodiment, when processing the low-dimensional state at the current moment, the low-dimensional state at the next moment and the low-dimensional action at the current moment by the bidirectional dynamical model, through the dynamical loss function and the low-dimensional dynamical loss function, a predicted corresponding forward time inversion state variation and a predicted corresponding reverse time inversion state variation can be obtained.

[0060] In an embodiment, when processing the low-dimensional state-action pair and the low-dimensional state at the next moment by the bidirectional dynamical model, the dynamical loss function and low-dimensional loss function are involved and as such the predicted forward time inversion state variation and predicted reverse time inversion state variation have extremely high generalizability and interpretability.

[0061] Finally, the encoding layer and a decoding layer in FIG.2 are optimized and updated by the preset loss functions including the dynamical loss function, the low-dimensional loss function and the reconstruction loss function, so as to improve a prediction accuracy of a result from the bidirectional dynamical model.

[0062] FIG. 2 further demonstrates that, a time inversion loss function is used in the learning process of the bidirectional dynamical model, where an expression of the time inversion loss function is as follows:

$$\begin{aligned} & \lambda_1 \|\nabla_s E_\varphi(s, a) \cdot \Delta s - z_\theta(\dot{s})\|_2^2 + \\ & \lambda_2 \|\nabla_s E_\varphi(s', a) \cdot (-\Delta s) - z_\beta(\dot{s})\|_2^2 + \end{aligned}$$

$$\begin{aligned} & \lambda_3 \|\Delta s - D_\psi(z_\theta(\dot{s}))\|_2^2 + \lambda_4 \|(-\Delta s) - D_\psi(z_\beta(\dot{s}))\|_2^2 + \\ & \lambda_5 \|(s, a, s') - D_\psi(z(s, a, s'))\|_2^2 + \\ & \lambda_6 (\|\theta\|_1 + \|\beta\|_1 + \|\varphi\|_1 + \|\psi\|_1), \end{aligned}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ and λ_5 are weights corresponding to the loss functions respectively, λ_6 is a regularized weight of a neural network parameter, θ is a forward dynamical model parameter, β is a reverse dynamical model parameter, φ is an encoder parameter, ψ is a decoder parameter, $\nabla_s E_\varphi(s, a)$ is a variation of a current low-dimensional state in the time dimension solved by the total differential equation, Δs is a variation between the state at the next moment and the state at the current moment, $z_\theta(\dot{s})$ is the forward time inversion state variation, $\nabla_s E_\varphi(s', a)$ is a variation of the next low-dimensional state in the time dimension obtained by the total differential equation, $z_\beta(\dot{s})$ is the reverse time inversion state variation, $D_\psi(z_\theta(\dot{s}))$ is decoding the forward low-dimensional state variation to original dimension forward state variation, $D_\psi(z_\beta(\dot{s}))$ is decoding the reverse low-dimensional state variation to original dimension reverse state variation, $D_\psi(z(s, a, s'))$ is decoding the current state, the low-dimensional action and the state at next moment to the current state, the action and the state at next moment in the original dimension, $\|\theta\|_1$ is L1 regularization of the network parameter of the forward dynamical model, $\|\beta\|_1$ is L1 regularization of the network parameter of the reverse dynamical model, $\|\varphi\|_1$ is L1 regularization of the network parameter of the encoder and $\|\psi\|_1$ is L1 regularization of the network parameter of the decoder.

[0063] Based on the above embodiment, the low-dimensional state-action pair and the low-dimensional state-action pair at the next moment are processed by the bidirectional dynamical model, and the corresponding forward time inversion state variation and the corresponding reverse time inversion state variation are obtained by the preset loss function. The obtained forward time inversion state variation and the obtained reverse time inversion state variation have extremely high generalizability and interpretability, and the value of the state-action value function obtained by taking the obtained forward time inversion state variation and the obtained reverse time inversion state variation as input can effectively assist the reinforcement learning to obtain the policy with the maximized long-term reward and realize the optimization of the behavior policy.

[0064] Based on the above embodiment, the inputting the preset input data to the value network, and obtaining the value of the state-action value function corresponding to the original state-action pair includes: based on the low-dimensional state and a low-dimensional policy output action in the low-dimensional state-action pair, calculating a low-dimensional policy state variation at the next moment; based on the low-dimensional state in the low-dimensional state-action pair and a corresponding low-

dimensional action, calculating the forward time inversion state variation; obtaining the value of the state-action value function by taking a difference between the low-dimensional policy state variation at the next moment and the forward time inversion state variation as the preset constraint item.

[0065] It should be noted that the preset input data is input into the value network to obtain the value of the state-action value function corresponding to the original state-action pair. In an embodiment, the low-dimensional policy state variation at the next moment is calculated based on the low-dimensional state and a low-dimensional policy output action in the low-dimensional state-action pair. The low-dimensional policy output action is the low-dimensional representation of the policy output action.

[0066] It should be noted that the policy output action refers to an output action of the policy network when the original state is used as the input of the policy network. The low-dimensional policy output action is obtained by compressing the policy output action by the encoder.

[0067] The formula for obtaining the low-dimensional policy state variation at the next moment is as follows:

$$\dot{z}_{s_t, a_t^\pi} = z_{s_{t+1}}^\pi - z_{s_t},$$

where $z_{s_{t+1}}^\pi$ is the low-dimensional representation of the state at the next moment transitioned by performing the action selected by the policy in state s_{t+1} , and z_{s_t} is the low-dimensional representation of the original state s_t , namely the low-dimensional state in the low-dimensional state-action pair.

[0068] Based on the low-dimensional state in the low-dimensional state-action pair and the low-dimensional representation of the original state at the next moment, the forward time inversion state variation is calculated.

[0069] The formula for obtaining the forward time inversion state variation is as follows:

$$\dot{z}_{s_t, a_t} = z_{s_{t+1}} - z_{s_t},$$

where $z_{s_{t+1}}$ is the low-dimensional representation of the original state at the next moment s_{t+1} , and z_{s_t} is the low-dimensional representation of the original state s_t .

[0070] On the basis of obtaining the low-dimensional policy state variation and the forward time inversion state variation, the difference value between the low-dimensional policy state variation and the forward time inversion state variation is used as the preset constraint item to obtain the value of the state-action value function. The preset constraint item should be noted as a penalty item, so as to ensure that an obtained state-action value correspond to an actual value the original state and the original action in the original state-action pair.

[0071] An optimization process of the original value network can be represented by the following formula:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha [R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)].$$

[0072] In this embodiment, taking the above-mentioned preset constraint item as a constraint of the value network, the optimization process of the original value network

can be represented by the following formula:

$$Q \leftarrow \underset{Q}{\operatorname{argmin}} \mathbb{E}_{(s,a,s') \sim D} [r(s,a) + \gamma \hat{Q}(z_{s'}, a'_\pi) - Q(z_s, z_a)],$$

where $r(s,a)$ is an instant reward of performing the action a in the state s , $z_{s'}$ is the low-dimensional representation of the state at the next moment, a'_π is the low-dimensional representation of the action generated by the policy at the next moment, z_s is the low-dimensional representation of the state at the current moment, z_a is the low-dimensional representation of the action at the current moment, γ is the hyper-parameter.

[0073] In this embodiment, by taking the difference value between the low-dimensional policy state variation at the next moment and the forward time inversion state variation as the constraint item, the value network is optimized and updated, as such the value of the state-action value function output by the value network is more accurate, and the reinforcement learning is effectively assisted to obtain the policy that maximizes the long-term reward and realize the optimization of the behavior policy.

[0074] FIG. 3 is a schematic diagram of an acquisition of a value of a state-action value function according to an embodiment of the present application, which further explained a case where the low-dimensional state-action pair is taken as the input of the value network.

[0075] As shown in FIG. 3, the corresponding value of the state-action value function is obtained by changing the input of value network in a current reinforcement learning method. In an embodiment, taking the original state-action pair $X(s,a)$ in the offline data set as the input of the encoding layer, the low-dimensional representation of the original state-action pair is obtained, and the value of the corresponding state-action value function can be obtained by inputting the low-dimensional representation of the original state-action pair $Z(X)$ to the value network.

[0076] It should be noted that in the process of optimizing and updating the value network, a relevant network parameter involved in the encoding layer in FIG. 3 do not participate in an update. The reinforcement learning algorithm provided in this embodiment can be applied to any existing algorithm including the state-action value function.

[0077] Based on the above embodiment, the offline reinforcement learning method further includes:

setting a constraint item on the policy based on an obtained value of the state-action value function, to obtain the behavior policy corresponding to the original state.

[0078] It should be noted that, in the related art, the set constraint item on the policy is the difference of the original action, which can be expressed by the following formula:

$$\pi = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{(s,a) \sim D} [\lambda(Q(s, a^\pi)) - (a^\pi - a)^2],$$

where $\underset{\pi}{\operatorname{argmax}}$ is the behavior policy corresponding to the maximum expectation value of returning to the original state-action, π is an obtained behavior policy, $\mathbb{E}_{(s,a) \sim D}$ represents an expectation based on the state-action pair in the data set, D represents the

offline data set, $Q(s, a^\pi)$ is the value of the state-action value function, λ is the hyper-parameter, and $(a^\pi - a)$ is the set constraint item which is the difference of the original action, namely the difference between the policy output action a^π and the original action a .

[0079] In this embodiment, based on the original state-action pair and the obtained value of the state-action value function, as well as the low-dimensional data obtained by processing of the encoder and the bidirectional dynamical model, different constraint items on the policy are set respectively.

[0080] In an embodiment, constraint item on the policy 1: the behavior policy corresponding to the original state in the original state-action pair is obtained by taking the value of the state-action value function and the difference value between a low-dimensional policy action and the low-dimensional action in the low-dimensional state-action pair as the constraint item, which can be expressed by the following formula:

$$\pi = \underset{\pi}{argmax} \mathbb{E}_{(s,a) \sim D} [\lambda(Q(s, a^\pi)) - (z_{a^\pi} - z_a)^2],$$

where z_{a^π} is the low-dimensional policy action, and z_a is the low-dimensional action in the low-dimensional state-action pair.

[0081] In an embodiment, constraint item on the policy 2: the behavior policy corresponding to the original state in the original state-action pair is obtained by taking the value of the state-action value function, and the difference value between the forward time inversion state variation obtained based on the low-dimensional state-action pair and the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair as the constraint item, which can be expressed by the following formula:

$$\pi = \underset{\pi}{argmax} \mathbb{E}_{(s,a) \sim D} [\lambda(Q(s, a^\pi)) - (\dot{z}_{s,a^\pi} - \dot{z}_{s,a})^2],$$

where $\dot{z}_{s,a^\pi}$ is the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair and $\dot{z}_{s,a}$ is the forward time inversion state variation obtained based on the low-dimensional state-action pair.

[0082] In an embodiment, constraint item on the policy 3: the behavior policy corresponding to the original state in the original state-action pair is obtained by taking the value of the state-action value function, the difference value between the forward time inversion state variation obtained based on the low-dimensional state-action pair and the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair, and the difference value between the low-dimensional policy action and the low-dimensional action in the low-dimensional state-action pair as the constraint item, which can be expressed by the following formula:

$$\pi = \underset{\pi}{argmax} \mathbb{E}_{(s,a) \sim D} [\lambda(Q(s, a^\pi)) - (\dot{z}_{s,a^\pi} - \dot{z}_{s,a})^2 - (z_{a^\pi} - z_a)^2].$$

[0083] In an embodiment, constraint item on the policy 4: the behavior policy corresponding to the original state in the original state-action pair is obtained by taking the value of the state-action value function, the difference value between the forward

time inversion state variation obtained based on the low-dimensional state-action pair and the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair, the difference value between the low-dimensional policy action and the low-dimensional action in the low-dimensional state-action pair, and the difference value between the low-dimensional policy action and the original action in the original state-action pair as the constraint item, which can be expressed by the following formula:

$$\pi = \underset{\pi}{argmax} \mathbb{E}_{(s,a) \sim D} [\lambda(Q(s, a^\pi)) - (a^\pi - a)^2 - (\dot{z}_{s,a^\pi} - \dot{z}_{s,a})^2 - (z_{a^\pi} - z_a)^2].$$

[0084] It should be noted that, in this embodiment, a relationship between the four different constraint items on policy is an or relationship, that is, when obtaining the behavior policy corresponding to the original state in the original state-action pair, any one of the above constraint item on the policy 1-4 can be selected.

[0085] The obtained behavior policy based on the constraint item on the policy 3 has the best effect.

[0086] In this embodiment, based on the original state-action pair and the obtained value of the state-action value function, as well as the low-dimensional data obtained by processing of the encoder and the bidirectional dynamical model, different constraint items on the policy are set respectively, as such the behavior policy corresponding to the original state is obtained, and the optimization of the behavior policy is realized more comprehensively and efficiently.

[0087] In addition, FIG. 4 is a schematic diagram an overall process of a dynamical symmetry modeling based offline reinforcement learning method according to an embodiment of the present application.

[0088] As shown in FIG. 4, there are mainly two stages. Firstly, an original dimensional state-action pair is processed by the bidirectional dynamical model, to obtain the low-dimensional data with extremely high generalizability and interpretability, and then the low-dimensional data is used to set a corresponding constraint item, the state-action value function (i.e. the above-mentioned value network)

is learned, and the value of the state-action value function $Q(Z(X))$ is obtained.

[0089] Secondly, on the basis of obtaining the value of the state-action value function, combined with the obtained low-dimensional data, the constraint item on the policy is set, and the policy network is optimized and updated, so as to obtain the behavior policy π that maximizes the reward value.

[0090] The overall process of the dynamical symmetry modeling based offline reinforcement learning method provided by this embodiment can be generalize as the process of “processing the original state-action pair, obtaining the state-action value function and obtaining behavior policy”.

[0091] FIG. 5 is a schematic structural diagram of a dynamical symmetry modeling based offline reinforcement learning apparatus according to an embodiment of the present application. As shown in FIG. 5, an apparatus for offline reinforcement learning includes: a data obtaining module 501, used for obtaining the offline data set; a data processing module 502, used for processing the original state-action pair in the offline

data set by the encoder and the bidirectional dynamical model, and obtaining the low-dimensional state-action pair and the current time inversion state variation; a value obtaining module 503, used for inputting the preset input data to the value network, and obtaining the value of the state-action value function corresponding to the original state-action pair; where the current time inversion state variation includes the forward time inversion state variation and the reverse time inversion state variation

[0092] The dynamical symmetry modeling based offline reinforcement learning apparatus provided by an embodiment of the present application and the above-mentioned dynamical symmetry modeling based offline reinforcement learning method can be referred to each other, which is not repeated herein.

[0093] In this embodiment, processing the original state-action pair in the offline data set by an encoder and a bidirectional dynamical model, and obtaining the low-dimensional representation of the original state-action pair, the forward time inversion state variation and the reverse time inversion state variation, and then inputting the low-dimensional representation of the original state-action pair in the value network, the value of the state-action value function corresponding to the original state-action pair can be obtained. The apparatus employs the bidirectional dynamical model to process the offline data, and extracts effective information from the offline data, namely the forward time inversion state variation and reverse time inversion state variation, where the effective information has extremely high generalizability and interpretability. As such the value of the state-action value function obtained by taking the effective information as an input of the value network can effectively assist a reinforcement learning to obtain a policy that maximizes a long-term reward and realize an optimization of the behavior policy.

[0094] FIG. 6 is a schematic structural diagram of an electronic device according to an embodiment of the present application. As shown in FIG. 6, the electronic device may include: a processor 610, a communication interface 620, a memory 630 and a communication bus 640, where the processor 610, the communication interface 620 and the memory 630 communicate with each other through the communication bus 640. The processor 610 can call a logic instruction in the memory 630 to execute the dynamical symmetry modeling based offline reinforcement learning method, and the method includes: obtaining an offline data set; processing the original state-action pair in the offline data set by an encoder and a bidirectional dynamical model, and obtaining a low-dimensional state-action pair and a current time inversion state variation; inputting the low-dimensional state-action pair to a value network, and obtaining the value of the state-action value function corresponding to the original state-action pair; where the current time inversion state variation includes the forward time inversion state variation and the reverse time inversion state variation.

[0095] In addition, the logic instruction in the memory 630 can be realized in a form of a software functional unit and sold or used as an independent product, and the logic instruction can be stored in a computer readable storage medium. Based on such understanding, an essence part or a part beneficial to the related art of the technical solution of the present application can be embodied in the form of a software product, and the software product is stored in a storage medium and includes several instructions

for enabling a computer device (which can be a personal computer, a server, or a network device, etc.) to perform all or part of the steps of the method according to various embodiments of the present application. The storage medium includes various media which can store program codes, such as a USB flash disk, a mobile hard disk, a read-only memory (ROM), a random access memory (RAM), a magnetic disk or an optical disc.

[0096] The present application also provides a computer program product, the computer program product includes a computer program, the computer program can be stored on a non-transient computer readable storage medium, and when the computer program is executed by the processor, the computer can execute the dynamical symmetry modeling based offline reinforcement learning method provided by the above embodiments, and the method includes: obtaining an offline data set; processing the original state-action pair in the offline data set by an encoder and a bidirectional dynamical model, and obtaining a low-dimensional state-action pair and a current time inversion state variation; inputting the low-dimensional state-action pair to a value network, and obtaining the value of the state-action value function corresponding to the original state-action pair; where the current time inversion state variation includes the forward time inversion state variation and the reverse time inversion state variation.

[0097] The present application further provides a non-transient computer readable storage medium, a computer program is stored on the non-transitory computer readable storage medium, and when the computer program is executed by the processor, the dynamical symmetry modeling based offline reinforcement learning method provided by the above embodiments can be realized, and the method includes: obtaining an offline data set; processing the original state-action pair in the offline data set by an encoder and a bidirectional dynamical model, and obtaining a low-dimensional state-action pair and a current time inversion state variation; inputting the low-dimensional state-action pair to a value network, and obtaining the value of the state-action value function corresponding to the original state-action pair; where the current time inversion state variation includes the forward time inversion state variation and the reverse time inversion state variation.

[0098] The above-mentioned apparatuses are merely schematic, where the units described as separate components may or may not be physically separated, and components displayed as units may or may not be physical units, may be located in one place, or may also be distributed on multiple network units. Some or all of the modules can be selected according to actual needs to achieve the purpose of the scheme of the embodiment. It should be noted and implemented by a person of ordinary skill in the art without involving any inventive effort.

[0099] Through the description of the above embodiments, a person skilled in the art can clearly understand that the embodiments can be implemented by means of software plus a necessary general hardware platform, and certainly can also be implemented by means of hardware. Based on this understanding, an essence part or a part beneficial to the related art of the technical solution of the present application can be embodied in the form of a software product, and the computer software product can be stored in a computer-readable storage medium, such as a ROM/RAM, a magnetic disk, an optical

disc, etc., and includes several instructions for enabling a computer device (which may be a personal computer, a server, or a network device, etc.) to perform the method described in various embodiments or some parts of the embodiments.

WHAT IS CLAIMED IS:

1. A dynamical symmetry modeling based offline reinforcement learning method, applied to a policy optimization, comprising:

obtaining an offline data set;

processing an original state-action pair in the offline data set by an encoder and a bidirectional dynamical model to obtain a low-dimensional state-action pair and a current time inversion state variation; and

inputting the low-dimensional state-action pair to a value network, and obtaining a value of a state-action value function corresponding to the original state-action pair,

wherein the current time inversion state variation comprises a forward time inversion state variation and a reverse time inversion state variation.

2. The method of claim 1, wherein the processing the original state-action pair in the offline data set by the encoder and the bidirectional dynamical model to obtain the low-dimensional state-action pair and the current time inversion state variation comprises:

processing the original state-action pair by the encoder to obtain the low dimensional state-action pair; and

processing a low-dimensional state-action pair at a current moment and a low-dimensional state-action pair at a next moment respectively by the bidirectional dynamical model, and obtaining the forward time inversion state variation and the reverse time inversion state variation through a preset loss function.

3. The method of claim 2, wherein the preset loss function comprises a dynamical loss function, a low-dimensional dynamical loss function and a reconstruction loss function,

wherein formulas of the dynamical loss function, the low-dimensional dynamical loss function and the reconstruction loss function are as follows:

$$L_{dl} = \|\Delta X(s, s') - D_\varphi(\dot{Z}_\theta(X))\|_2^2;$$

$$L_{ldl} = \|E_\varphi(\Delta X(s, s')) - \dot{Z}_\theta(X)\|_2^2;$$

$$L_{rl} = \|X(s, a) - D_\varphi(Z(X))\|_2^2,$$

wherein, s is a state in the offline data set, a is an action in the state of s in the offline data set, $\Delta X(s, s')$ is a state variation at the next moment, s' is a state at the next moment in the offline data set, θ and φ are hyper-parameters, $\dot{Z}_\theta(X)$ is the forward time inversion state variation obtained by processing of the bidirectional dynamic model, $D_\varphi(\dot{Z}_\theta(X))$ is a state-action variation at the next moment obtained by

decoding the forward time inversion state variation, $E_\varphi(\Delta X(s, s'))$ is the forward time inversion state variation obtained by directly encoding of the encoder, $X(s, a)$ is the original state-action pair in the offline data set and $D_\varphi(Z(X))$ is the original state-action pair obtained by decoding the low-dimensional state-action pair.

4. The method of claim 1, wherein the inputting the low-dimensional state-action pair to the value network, and obtaining the value of the state-action value function corresponding to the original state-action pair comprises:

calculating a low-dimensional policy state variation at the next moment based on a low-dimensional state and a low-dimensional policy output action in the low-dimensional state-action pair;

calculating the forward time inversion state variation based on the low-dimensional state and a low-dimensional representation of an original state at the next moment in the low-dimensional state-action pair;

obtaining the value of the state-action value function by taking a difference value between the low-dimensional policy state variation at the next moment and the forward time inversion state variation as a preset constraint item.

5. The method of claim 4, further comprising:

obtaining a behavior policy corresponding to an original state in the original state-action pair by taking the value of the state-action value function and the difference value between a low-dimensional policy action and the low-dimensional action in the low-dimensional state-action pair as a constraint item; or,

obtaining a behavior policy corresponding to an original state in the original state-action pair by taking the value of the state-action value function, and the difference value between the forward time inversion state variation obtained based on the low-dimensional state-action pair and the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair as a constraint item; or,

obtaining a behavior policy corresponding to an original state in the original state-action pair by taking the value of the state-action value function, the difference value between the forward time inversion state variation obtained based on the low-dimensional state-action pair and the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair, and the difference value between the low-dimensional policy action and the low-dimensional action in the low-dimensional state-action pair as a constraint item; or,

obtaining a behavior policy corresponding to an original state in the original state-action pair by taking the value of the state-action value function, the difference value between the forward time inversion state variation obtained based on the low-dimensional state-action pair and the forward time inversion state variation obtained based on the low-dimensional state and the low-dimensional policy action in the low-dimensional state-action pair, the difference value between the low-dimensional policy action and the low-dimensional action in the low-dimensional state-action pair, and the difference value between the low-dimensional policy action and an original action in the original state-action pair as a constraint item.

6. The method of claim 1, wherein the bidirectional dynamical model is obtained by training a time inversion loss function.

7. A dynamical symmetry modeling based offline reinforcement learning apparatus, applied to a policy optimization, comprising:

a data obtaining module, used for obtaining an offline data set;
a data processing module, used for processing an original state-action pair in the offline data set by an encoder and a bidirectional dynamical model to obtain a low-dimensional state-action pair and a current time inversion state variation; and
a value obtaining module, used for inputting the low-dimensional state-action pair to a value network, and obtaining a value of a state-action value function corresponding to the original state-action pair,
wherein the current time inversion state variation comprises a forward time inversion state variation and a reverse time inversion state variation.

8. An electronic device, comprising:

a processor;
a memory; and
a computer program stored on the memory and executable on the processor,
wherein the computer program, when executed by the processor, causes the electronic device to perform steps of the method of any one of claims 1 to 6.

9. A non-transitory computer-readable storage medium storing computer programs, when executed by a processor, cause the processor to perform steps of the method of any one of claims 1 to 6.

10. A computer program product comprising computer programs, when executed by a processor, cause the processor to perform steps of the method of any one of claims 1 to 6.