Knowledge-Based Systems 000 (2017) 1–11

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A proactive decision support method based on deep reinforcement learning and state partition

Yongheng Wang*, Shaofeng Geng, Hui Gao

College of Information Science and Engineering, Hunan University, Changsha 410082, China

ARTICLE INFO

Article history:
Received 13 January 2017
Revised 1 November 2017
Accepted 2 November 2017
Available online xxx

Keywords:
Event stream
Proactive decision support
Deep reinforcement learning
State partition

ABSTRACT

Big streaming data is an important kind of big data which we need new technology to process. Getting knowledge from online streaming data and making decision online can help us get more value from Big data. A proactive decision support system can predict future states and mitigate or eliminate undesired future states by taking some actions proactively. But it is difficult to handle some issues like the data distribution change in streaming data, combination of prediction and decision making, and the huge state space in decision making. In this paper, we propose a proactive decision support method based on deep reinforcement learning and state partition. The predictive analytics part uses deep belief networks with two level incremental training method. The deep reinforcement learning part uses deep belief networks as function approximation which is learned by semi-gradient method. Off-policy is supported through important sampling. Two kinds of state partition and parallel execution methods are proposed to improve the performance. The experimental evaluation in traffic congestion control application shows this method works well in both accuracy and performance.

1. Introduction

Big streaming data is an important kind of big data. Usually stream data processing and online decision making can help us getting more value than batch processing of historical data. Recently some famous tools for stream processing have been developed, such as Spark streaming, Storm and Apache Flink. Having been used in many applications, these tools shows high scalability and reliability when processing big streaming data. However, the limitation of these tools is that they can only support basic processing of stream, such as filtering, transformation and counting. Currently we require new technology to get knowledge form online streaming data and to make decisions at run time. Traditional decision making is reactive which means agents are triggered by events and react to change states of the system. A new kind of decision making is called proactive decision making, which means system can mitigate or eliminate undesired future states, or to identify and take advantage of future opportunities, by applying prediction and automated decision-making technologies [1]. Proactive decision support is valuable for some applications in which the emerging of some states might be a disaster for the system. For example, a financial institution wishes to detect frauds or a financial regulator wishes to catch illegal trading patterns. Another example is to predict the traffic status in road networks and take some actions proactively to mitigate or eliminate traffic congestion.

Currently there are still some challenges to support proactive decision support for big streaming data. To predict future states, we need prediction methods for streaming data that has high accuracy and performance. An important challenge for streaming data prediction is concept drift. Concept drift reflects a situation in which the input and/or output concepts do not follow a fixed and predictable data distribution. To address this issue, we need models/algorithms that can be adjusted according to the change of data distribution. Sequential decision making for big streaming data is a more challenging technology. We need to handle the huge state space and action space, which usually make it difficult to use exact methods. If the state is composed of sub-states and action is composed of sub-actions, this problem becomes more serious. Another important challenge is the exploration- exploitation dilemma which means the agent must exploit what it already knows to obtain reward, but it also must explore to make better action selections in the future.

In this paper, we proposed a Proactive Decision Support method based on Reinforcement Learning (PDSRL). This method uses a Deep Belief Network (DBN) to predict future states from event streams and reinforcement learning algorithm to support proactive decision making. The main contribution includes the following:

- We propose a streaming data prediction method based on DBN. A two-layer incremental training method for DBN is proposed to address the concept drift issue of streaming data.

* Corresponding author.
E-mail addresses: wyh@hnu.edu.cn (Y. Wang), sfgeng@hnu.edu.cn (S. Geng), coffey@hnu.edu.cn (H. Gao).

https://doi.org/10.1016/j.knosys.2017.11.005
0950-7051/

- We propose a proactive decision-making method with finite Markov Decision Processes (MDP). Prediction and reinforcement learning are combined to support proactive decision making. A n-step off-policy Temporal-Difference (TD) method is used for reinforcement learning. DBN is used as model for approximate value function.
- Two state partition methods – vertical state partition and horizontal state partition, are proposed to improve the performance of reinforcement learning.

We evaluated the PDSRL method with a traffic simulation system to support proactive congestion control. The evaluation result shows it works well to support proactive decision making for complex event streams.

2. Related work

2.1. Predictive analytics for streaming data

Predictive analytics has been studied for many years and a series of models and algorithms are proposed. Among these models, Bayesian Network (BN) model and neural networks are widely used. Zheng et al. proposed a linear conditional Gaussian Bayesian network model to consider both spatial and temporal dimensions of traffic as well as speed information for short-term traffic flow prediction [2]. Their model allows both continuous and discrete variables, which enables the consideration of categorical variables in traffic flow prediction. Borchani et al. proposed a risk prediction method in credit operations stream based on dynamic Bayesian networks [3]. Their model is a special case of a more general framework that can accommodate more expressive models containing latent variables as well as more sophisticated feature selection schemes. The advantage of BN is that it has rich mathematical theories and it can explain to users how the system came to its conclusions.

Recently, some research work using deep neural network for predictive analytics is proposed. In the work of Lv et al., a stacked autoencoder model is used to learn generic event stream features, and it is trained in a greedy layer wise fashion [4]. Huang et al. proposed a predictive analytics method based on deep belief networks [5]. Their method uses deep belief networks in the bottom level and a regression model is used on the top level to support the final prediction. Multitask learning is supported in the top regression model to improve performance. Singh et al. proposed a deep learning method for stock prediction which uses 2-directional 2-dimensional Principal Component Analysis (PCA) and deep Radial Basis Function Neural Network (RBFNN) [6]. The advantage of deep neural network model is that it supports unsupervised low-level features learning which can help to get high accuracy when processing complex event streams.

For streaming data analysis, the concept drift issue is still a difficult problem at present. The key problem is how to design model that can reflect the concept drift and algorithms that can train the model incrementally. For abrupt drifts, a common idea is to learn different models from historical data and switch among models when data distribution is changed. Incremental drifts are more difficult to be handled, because it cannot be detected by global drift detection approaches in the structural learning scenario and cannot be addressed by the parameter learning scenario either. Recently some incremental learning methods and evolving models are studied. Acharya et al. proposed an incremental causal network construction algorithm over event streams [7]. Their algorithm infers causality by learning the temporal precedence relationships using incremental temporal network construction algorithm and the dependency by adopting an incremental Bayesian network construction algorithm called the Incremental Hill-Climbing Monte Carlo. For neural network, a reasonable technology is self-organizing incremental neural network (SOINN) [8,9]. SOINN is a two-layer neural network based on competitive learning, which can adjust the weights of neurons and the connections between neurons according to input data without any prior knowledge. However, currently SOINN is still lack of rigorous theoretical proof, and how to expand it to multilayer neural network is also a difficult problem. Another research direction is evolving neural network which supports changing connection weights, learning rules and network architecture with the change of data distribution. Yevgeniy et al. proposed an evolving neural network method for streaming data which is based on group method of data handling and least square support vector machine [10]. Hall et al. proposed an evolving spike neural network method for streaming data, which supports features autoregression based on multivariate Hawkes point process model [11]. Compared to these methods, we propose a DBN model with a two-layer incremental training method.

Recently a rich set of technology and tools are developed for big data environment. In this work, Spark is used as basic computing framework and Spark Streaming is used for basic processing of streaming data. Apache Kafka is used as event bus. These tools are used to support big data processing, but this work does not try to improve them.

2.2. Proactive decision making with reinforcement learning

Proactive decision making has been used in many applications. Bala et al. proposed a proactive decision-making method for dynamic assignment and routing of unmanned aerial systems [12]. Their method extends the open vehicle routing problem and tries to avoid the possible load unbalancing through adjusting the path of unmanned aerial systems proactively. In the work of Kimon, a proactive decision support system for business process execution is developed [13]. Their system forecasts events, provides the best corresponding action and generates appropriate business rules that can be used by the process engine to make optimal decisions during process runtime. A new research area related to proactive decision making is proactive complex event processing. Lajos et al. proposed a conceptual framework of proactive complex event processing which combines complex event processing and predictive analytics to support proactive decision making [14]. Fabiana et al. proposed an event-based proactive decision-making architecture which includes complex event processing, predictive analytics, decision support and visualization components [15]. They also provide a prototype to demonstrate their system.

Since we usually have not complete knowledge of the application system, reinforcement learning is a reasonable technology for proactive decision making. Reinforcement learning has been studied for many years and many methods are proposed. An important research is how to use approximate methods to handle the huge problem space. Zhao et al. proposed a probably approximately correct method which can be used to find optimal policy in limited time range [16]. An important work recently is the deterministic policy gradient proposed by Silver et al. [17]. Instead of using traditional value function, policy gradient finds the optimal policy directly by methods like gradient ascendant. Another important work is deep reinforcement learning which combines deep learning and reinforcement learning. The basic idea of deep reinforcement learning is to use deep neural network to approximate the value function. Mnih et al. proposed deep Q-network which uses deep neural network to approximate Q-function, and applied their method in video game successfully [18]. Many other works on deep reinforcement learning are based on deep Q-network. Currently the commonly used deep model is convolutional neural network and recursive neural network. Van et al. proposed a double deep Q-network model which uses two deep Q-networks to avoid over