DYNAMIC BAYESIAN NETWORKS FOR STATE- AND ACTION-SPACE MODELLING IN REINFORCEMENT LEARNING

Lekhovitsky D., Khovrat A.
Scientific Supervisor – C. Ph-M Sc., assoc. prof. Sidorov M. V.
Kharkiv National University of Radioelectronics
(61166, Kharkiv, 14 Nauky ave., dept. of Applied Mathematics, +38 (057) 7021335),
e-mail: lekhovitsky@gmail.com, phone: +38 (095) 1760231

In recent years Reinforcement Learning has proven its efficiency in solving problems of sequential decision making, formalized with a concept called Markov Decision Process. Though, there is a lot of problems: high computational complexity for multivariate state- and action-space problems, needs to handle missing data and hidden variables, lack of both good model and a sufficient number of episodes for constructing an optimal policy. In this work we suggest Dynamic Bayesian networks (DBNs) as a solution. These models provide an elegant and compact representation of joint state-action space, efficient inference algorithms, which include Monte-Carlo methods and Belief Propagation, and can be used in Dyna-Q Algorithm for integrating real-world and simulated experience.

Markov Decision Process is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}, \gamma \rangle$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $\mathbb{P}$ is a transition model $\mathbb{P}[S', R \mid S, A]$ of next state S' and current reward R given the current state S and action A, and $\gamma \in [0; 1]$ is a discount factor. The goal is to find an optimal policy $\pi(A \mid S)$ which maximizes an overall expected reward $\mathbb{E}_{\pi}[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots \mid S_t]$ of following this policy. In Reinforcement Learning samples of previous experience are used to do this.

Bayesian network is a name for both a framework for representation of joint distribution of complex multivariate systems and for a core structure of this framework. Formally, Bayesian network is a directed acyclic graph $\mathcal{B} = \langle \mathcal{X}, \mathcal{E} \rangle$, where the set $\mathcal{X}$ is a set of random variables, and every node X of this graph (usually drawn as a circle) is augmented with a conditional probability distribution of corresponding variable given its parents in the graph $\mathcal{B}$: $\mathbb{P}[X \mid \mathrm{Par}_{\mathcal{B}}(X)]$.

The main property of Bayesian networks is factorization of joint distribution $\mathbb{P}[\mathcal{X}]$ in a form

$$\mathbb{P}[\mathcal{X}] = \prod_{X \in \mathcal{X}} \mathbb{P}[X \mid \mathrm{Par}_{\mathcal{B}}(X)].$$

Such factorization allows performing inference (e.g. computing marginal or conditional distributions given some evidence, expectation or the most probable assignment) for variables of our interest in a very efficient way. Inference algorithms, which include Belief Propagation and Monte-Carlo Markov Chain,

learning networks' parameters and structures, as well as other representational properties, are described in [1].

To incorporate Bayesian networks into the Reinforcement Learning problem, we need to provide a way of handling temporal structure. Let's consider a multivariate random process $\mathcal{X}^{(t)}$, $t \in \{0, 1, \ldots, T\}$, with joint distribution $\mathbb{P}$. We can write

$$\mathbb{P}[\mathcal{X}^{(0)}, \mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(T)}] = \mathbb{P}[\mathcal{X}^{(0)}] \prod_{t=1}^{T} \mathbb{P}[\mathcal{X}^{(t)} \mid \mathcal{X}^{(t-1)}],$$

assuming the Markov property holds: $\forall t \;\; \mathbb{P}[\mathcal{X}^{(t+1)} \mid \mathcal{X}^{(0:t)}] = \mathbb{P}[\mathcal{X}^{(t+1)} \mid \mathcal{X}^{(t)}]$. If we also assume that $\forall t \;\; \mathbb{P}[\mathcal{X}^{(t+1)} \mid \mathcal{X}^{(t)}] = \mathbb{P}[\mathcal{X}' \mid \mathcal{X}]$ (i.e. that our process is homogeneous), the only two things needed to represent a process are initial probability distribution $\mathbb{P}[\mathcal{X}^{(0)}]$ and a transition model $\mathbb{P}[\mathcal{X}' \mid \mathcal{X}]$, both of which can be represented as Bayesian networks. Such representation of a random process is called a Dynamic Bayesian network, and, in fact, it extends a notion of a Markov chain to a multivariate state-space case.

Moreover, Markov Decision Process itself can be viewed as DBN with $\mathcal{X}^{(t)}$ being all the variables S_t, A_t and R_t together, and Bayesian network used to represent inter-time-slice and within-time-slice dependencies (which are transition model $\mathbb{P}$ and policy π). Besides compactness usage and reduced inference computational cost, this approach has a lot of advantages which address some problems of modern Reinforcement Learning.

First, the inherent similarity with causal influence diagrams allows Bayesian networks to be used for knowledge representation. This property can be exploit by incorporating existing understanding of environment into the model by engineering priors on its structure and parameters, or by extracting new knowledge from already learnt model.

Second, there are simple and efficient modifications of classical ML and MAP estimation, EM algorithm for learning parameters of a network. Learning the structure is a lot more complex problem but some good algorithms exist.

Third, one can combine real-world and simulation experience by iterating between sampling from a real world, using this sample to update the policy, using this sample to update the model, sampling from the model and now using this simulated experience to update the policy. This is a so-called Dyna Architecture.

1. D. Koller, N. Friedman. Probabilistic Graphical Models: Principles and Techniques, – The MIT Press, 2009. – 1208 p.

2. R. Sutton, A. Barto. Reinforcement Learning: An Introduction, The MIT Press, 2018. – 444 p.