Fast Uniform [Distributional] State Aggregation in Reinforcement Learning

Authors [*] [1]

Abstract

Reinforcement learning (RL) problems often incur large scale state spaces needing sizable amount of computation to process. State aggregation as an analytically-transparent mean approximates and yields lower-dimension state spaces. We present a state aggregation algorithm method in Markov decision processes (MDP) inspired by histograms. Our approach explores a *splitting policy* in direct relationship with inter-quartile range of collected samples. We then accompanied an experiment by which we show the practical promises of the approach to achieve superior results.

1. Introduction

Real-world problems embody high-dimensional state spaces where in computational issues arise. Various approximation methods are proposed in the literature to reduce the resulted value function dimensionality (Sutton & Barto, 1998; Buoniu et al., 2010; François-Lavet et al., 2018). State aggregation is one of the simplest and most transparent state representation approximation method.

State aggregation, at the core level, is to discretize together similar states to ultimately yield a lower-dimension MDP. The reduced MDP requires considerably less amount of computational iterations or samples to solve, which improves convergence to the .

Although state aggregation may not generalize as well as neural network-based methods, the techniques has both analytical and practical advantages (Lagoudakis & Parr, 2003). In comparison, state aggregation takes less computational power to calculate representative features. Moreover, analysis and troubleshooting in a state aggregation process is more obvious than a black-box neural network. Approximation is a corner stone in all successful RL methods to hold a ”good” mapping between a large state space to a much smaller one where in value function is computable or at least feasible. These ”good” approximators can end up with ”good” features to represent the projected state space.

The cross-fertilization in using state aggregation and deep neural networks in solving immense state space problems in RL is also profound. While the newly emerged methods are growing at a staggering pace, a parallel area of research has been formed wherein a variety of techniques have been utilized to make these methods contravene their promised functionality.

Our contribution is a bifurcation. We first introduce an algorithm for aggregation in MDPs, then we move on to show advantages of taking such aggregation method to work. Hereby, we explain the paper structure in brief. We first lay down the core concepts of the MDP and the aggregation framework in Sections 3 and 4 then continue to describe the methodology in Section 5.

2. Related Work

Aggregation was extensively appeared throughout the operations and computations research literature (Chatelin & Miranker, 1982; Rogers et al., 1991; Douglas & Douglas, 1993). It then introduced to the optimization by an iterative approach in linear programming (Mendelssohn, 1982). On the end of this spectrum, dynamic programming (DP) leveraged state aggregation to reduce computational size of the problem (Bean et al., 1987), as well as using an extension to state aggregation, *soft state aggregation*, in an approximate value iteration method (Singh et al., 1995), and general approximate DP frameworks (Gordon, 1995; Tsitsiklis & Van Roy, 1996).

In the recent body of work, state aggregation has applications in conjunction to other concepts such as temporal aggregation, so called *options* (Ciosek & Silver, 2015), *bottleneck simulator* (Serban et al., 2018), and also in continuous space optimal control (Zhong & Todorov, 2011).

Error analyses and regret bounds incured by state aggregation methods are comprehensively discussed by Van Roy

[*] Equal contribution [1] Department of Computer Science, University of New Hampshire. Correspondence to: Authors <c.vvvvv@googol.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020. Copyright 2020 by the author(s).

(2006) and Petrik & Subramanian (2014). Near optimality criterion in aggregation is also investigated in Bernstein & Shimkin (2008).

It is noteworthy to point out that, state aggregation exists in the literature under different co-hyponyms.

3. Markov Decision Processes

Reinforcement learning problems can be formulated a Markov Decision Processes (MDP) framework (Puterman, 1994; Sutton & Barto, 1998). MDPs facilitate handling sequential decision-making under uncertainty. Discrete discounted reward criterion is the center of focus for ease of exposition in this paper. For an environment and a decision maker, consider the tuple <> state space S and action space A are defined, respectively. The transition probabilities P describe how likely is the next state to incur by taking a specific action a by the agent in a certain state s which will later receive reward r. For a taken action $a_i \in A$ at the state $s_i \in S$, the decision maker will be ended up in state $s_{i+1} \in S$ with a likelihood expressed by the transition probabilities $P : S \times A \times S \rightarrow [0, 1]$ and wil receive reward r_{i+1} from the rewards $R : S \times A \rightarrow \mathbb{R}$.

4. State Aggregation

Value function approximation alleviates the computational hurdle of a large state-space problem. We first introduce an approximate value iteration method to solve an aggregate problem. State aggregation is an approximation approach to make a problem more tractable by a reduction in dimensionality in state space [1]. State aggregation, in particular, is a parametric feature-based approximation where the features are membership functions of 0-1 form. Aggregation consolidates similar states together, so called representative states, which allows the *aggregate* problem to be solved by an exact method (Bertsekas, 2019) The computed value function from the exact solution of the aggregate problem can be then used in the original problem.

Now we formulate the aggregation by introducing the dynamics of the aggregate problem. The transition probabilities between two representative states:

$$\hat{P}_{ss'}(a) = \sum_{j=1}^{n} P_{sj}(a)\phi_{jy}, \tag{1}$$

where p_{sj} is the transition probabilities from a representative state, s, to an original state, j and $\phi_{js'}$ is the aggregation probabilities from an original state, j, to a representative state, s'. The rewards are defined as:

$$\hat{r}(s, a) = \sum_{j=1}^{n} P_{sj}(a) r(s, a, j), \tag{2}$$

where $\hat{r}(s, a)$ is the received reward at state s by taking action a.

Aggregation then helps us to derive the value function. Approximated optimal value function is a weighted sum of optimal rewards and can be computed by Equation 3 where $r^*_{s'}$ is optimal reward at represented state s.

$$\tilde{V}(j) = \sum_{s' \in \mathcal{U}} \phi_{js'} r^*_{s'}, \tag{3}$$

5. Method

Splitting the observations into spans of similar observations is one way to interpret an aggregation problem. Histograms as classical and simple density estimators were around in the literature for decades (Scott, 1979; 2015). Inspired by histograms and the statistics behind finding the proper number of bins for a given distribution, we build our proposed aggregation method on top of such bases. Proper *bin width*, according to the definition, is to find a trade-off to capture similarities in data features and diminish variations caused by random sampling (Knuth, 2019).

Finding the bin width for a given distribution could be calculated by Freedman-Diaconis rule (Freedman & Diaconis, 1981). As the rule states:

$$|S_{agg}| = 2\frac{\mathrm{IQR}(x)}{\sqrt[3]{n}} \tag{4}$$

where n is number of observations and $IQR(x)$ is the interquartile range of data. This segmentation rule exhibits robustness to sample distribution due to low sensitivity of IQR to outliers and heteroscedasticity in the given distribution compared to conventional dispersion measures such as variance or standard deviation. State values, in this paper, are considered as the similarity metric [measure] in state aggregation.

By having the inter-quartile range of data for each feature, a discretization policy can be calculated.

For the sake of clarity, we elaborate on some preliminary definitions here to have a consistent set of notations to the end of this paper. We use model-free LSPI and LSTDQ (Lagoudakis & Parr, 2003) to update and evaluate the policy of actions, respectively.

We developed an aggregation scheme to lower the state space dimensionality by assigning same action to the state

[1] aggregation in action space although is possible, has less significance comparatively.

in a neighbourhood, so called nearest neighbour/ piecewise linear aggregation. We then solve directly the lower dimension problem which relatively takes less computation.

The aggregate problem is stochastic even if the original problem is deterministic. Once we lay out the aggregate problem framework we can solve the problem by any exact methods either value- or policy-space solutions.

The objective is finding an optimal aggregate model which falls in the span of spectrum from a fine-grid model to a rigid aggregate model. Our presented method calculates a discretization policy for which we calculate a feasible value function. Return of a policy shows the quality of the gauged policy. Hence, by calculating the accumulative return (eq.5) over run of the *true* model and in a comparison with the original return we can measure the improvement.

$$R = \sum_{k=0}^{\infty} \gamma^k r_{k+1}. \tag{5}$$

5.1. Aggregation

The original states with the closest *calculated* values forgather to build aggregate states. For this purpose, state values are clustered by K-means analysis to calculate the relevant aggregate states. State aggregation merge similar transitions based on a triple $< s, s', a >$ where s indicates the starting state, s' is the ending state, and the take action in that transition represented by a. In the process of stacking similar transitions, we here presume all inbound transitions as outbound transition of the ending state. This assumption render all transitions as outbound ones, therefore we assign the average probability as the transition probability in the aggregate model.

6. Experiment

We apply the approach to two MDPs proposed by (Strehl & Littman, 2004). *RiverSwim* an MDP consists of six states and *SixArms* one with seven states. In *RiverSwim* flow of the river is to left and the swimmer picks probabilistically equal either state 1 or 2 as the start point. All states but two terminal states have zero reward to land at while the right most one rewards substantially higher than the other one. The agent in SixArms selects between six distinct actions which pulls different arms of a multi-armed bandit. By pulling an arm the agent traverses to a new state. Although the agent does not obtain reward by pulling the arms but going to the connected state to that arm is highly rewarding in which the transition probability is in inverse relationship with the reward value.

As both problems favor keeping the agent in states with lower reward, exploration is paramount to maximize the accumulated reward. This is also confirmed by the authors in (Strehl & Littman, 2004) that the agent will fail to learn when a rudimentary ϵ-greedy algorithm.

We use a randomized policy to simulate trajectories and collect samples to attenuate bias in action selection process. As it has shown the literature, ϵ-greedy has a bias in solving exploration-exploitation dilemma, we explore the environment by following a randomized policy with a uniform distribution.

6.1. Machine Replacement

This domain as a *non-episodic* (continuous) task maintains a set of machines. The goal is to find a policy to incur the least repair needed. incure

7. Future Works

We developed the presented approach with having reproducibility in mind. It could be served as a launching pad for the future contributors who want to surf this newly emerged field deeper. Future ideas to pursue is a generative model to be able to generate adversaries from a network.

References

Bean, J. C., Birge, J. R., and Smith, R. L. Aggregation in Dynamic Programming. *Operations Research*, 35(2): 215–220, jun 1987. ISSN 0030364X, 15265463.

Bernstein, A. and Shimkin, N. Adaptive Aggregation for Reinforcement Learning with Efficient Exploration: Deterministic Domains. Technical report, 2008.

Bertsekas, D. P. *Reinforcement Learning and Optimal Control*. 2019. ISBN 9781886529397.

Buoniu, L., Babuška, R., De Schutter, B., and Ernst, D. *Reinforcement learning and dynamic programming using function approximators*. 2010. ISBN 9781439821091. doi: 10.1201/9781439821091.

Chatelin, F. and Miranker, W. L. Acceleration by aggregation of successive approximation methods. *Linear Algebra and Its Applications*, 43(C):17–47, mar 1982. ISSN 00243795. doi: 10.1016/0024-3795(82)90242-7.

Ciosek, K. and Silver, D. Value Iteration with Options and State Aggregation. jan 2015.

Douglas, C. C. and Douglas, J. Unified convergence theory for abstract multigrid or multilevel algorithms, serial and parallel. *SIAM Journal on Numerical Analysis*, 30(1):136–158, jul 1993. ISSN 00361429. doi: 10.1137/0730007.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., and Pineau, J. An Introduction to Deep Reinforcement Learning. 2018. doi: 10.1561/2200000071.

Freedman, D. and Diaconis, P. On the Histogram as a Density Estimator: L 2 Theory. 1981.

Gordon, G. J. Stable Function Approximation in Dynamic Programming. In *Machine Learning Proceedings 1995*, pp. 261–268. Elsevier, jan 1995. doi: 10.1016/b978-1-55860-377-6.50040-2.

Knuth, K. H. Optimal data-based binning for histograms and histogram-based probability density models. *Digital Signal Processing: A Review Journal*, 95:102581, dec 2019. ISSN 10512004. doi: 10.1016/j.dsp.2019.102581.

Lagoudakis, M. G. and Parr, R. Least-Squares Policy Iteration. Technical report, 2003.

Mendelssohn, R. An Iterative Aggregation Procedure for Markov Decision Processes. *Operations Research*, 30 (1):62–73, 1982. ISSN 0030364X. doi: 10.1287/opre.30.1.62. URL https://www.jstor.org/stable/170309.

Petrik, M. and Subramanian, D. RAAM: The Benefits of Robustness in Approximating Aggregated MDPs in Reinforcement Learning. Technical report, 2014.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994. ISBN 0471727822.

Rogers, D. F., Plante, R. D., Wong, R. T., and Evans, J. R. Aggregation and Disaggregation Techniques and Methodology in Optimization. *Operations Research*, 39(4):553–582, aug 1991. ISSN 0030-364X. doi: 10.1287/opre.39.4.553.

Scott, D. *Multivariate density estimation: theory, practice, and visualization*. 2015.

Scott, D. W. On Optimal and Data-Based Histograms. *Biometrika*, 66(3):605, dec 1979. ISSN 00063444. doi: 10.2307/2335182.

Serban, I. V., Sankar, C., Pieper, M., Pineau, J., and Bengio, Y. The Bottleneck Simulator: A Model-based Deep Reinforcement Learning Approach. jul 2018.

Singh, S. P., Jaakkola, T., and Jordan, M. J. Reinforcement Learning with Soft State Aggregation. *Advances in Neural Information Processing Systems 7*, pp. 361–368, 1995. ISSN 1049-5258.

Strehl, A. L. and Littman, M. L. An empirical evaluation of interval estimation for Markov decision processes. In *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2004. ISBN 076952236X. doi: 10.1109/ICTAI.2004.28.

Sutton, R. S. and Barto, A. G. Sutton & Barto Book: Reinforcement Learning: An Introduction. Technical report, 1998.

Tsitsiklis, J. N. and Van Roy, B. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22(1-3):59–94, 1996. ISSN 08856125. doi: 10.1007/BF00114724.

Van Roy, B. Performance Loss Bounds for Approximate Value Iteration with State Aggregation. *Mathematics of Operations Research*, 31(2):234, 2006. doi: 10.1287/moor.1060.0188.

Zhong, M. and Todorov, E. Aggregation methods for lineary-solvable Markov decision process. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, volume 44, pp. 11220–11225. IFAC Secretariat, jan 2011. ISBN 9783902661937. doi: 10.3182/20110828-6-IT-1002.03729.