

Dynamic Generalization Kanerva Coding in Reinforcement Learning for TCP Congestion Control Design

(Extended Abstract)

Wei Li*, Fan Zhou†, Waleed Meleis‡ and Kaushik Chowdhury[¶]

Department of Electrical and Computer Engineering
Northeastern University
Boston, USA

{*li.wei, †zhou.fan1}@husky.neu.edu, {‡meleis, ¶krc}@ece.neu.edu

ABSTRACT

Traditional reinforcement learning (RL) techniques often encounter limitations when solving large or continuous state-action spaces. Training times needed to explore the very large space are impractically long, and it can be difficult to generalize learned knowledge. A compact representation of the state space is usually generated to solve both problems. However, simple state abstraction often cannot achieve the desired learning quality, while expert state representations usually involve costly hand-crafted strategies.

We propose a new technique, generalization-based Kanerva coding, that automatically generates and optimizes state abstractions for learning. When applied to adapting the congestion window of the highly complex TCP congestion control protocol, a standard Internet protocol, this technique outperforms the current standard-TCP New Reno by 59.5% in throughput and 6.5% in delay. Our technique also achieves a 35.2% improvement in throughput over the best previously proposed Kanerva coding technique when applied in the same context.

Keywords

state abstraction; TCP congestion control; dynamic generalization; Kanerva coding

1. INTRODUCTION

RL has been effectively applied to a variety of problem domains, such as robotics [1], control [2] and computer game playing [3]. In this paper, we explore how RL techniques can be effective when applied to hard networking control problems. Applying RL to TCP congestion control is challenging due to the problem's continuous and high-dimensional state space in a dynamically-changing network environment. For continuous state systems, it is impossible to directly represent all system states. This motivates the need for approximating representations to store the value functions.

Many approximating approaches have been developed to abstract or compress full state spaces and one popular approach is function approximation [4]. Many function approx-

imation techniques have been applied, including tile coding [5] (also known as CMAC) and its variants [6], [7], and tree-based state partitions [8]. However, the use of manual partitions or inflexible abstractions of the state space limit the applicability of these techniques when solving real-world problems.

Kanerva coding [9], also known as Sparse Distributed Memories, provides another approach for dealing with a complex, high-dimensional state space. This technique uses a subset of the state space to represent the whole state space. If the subset of states is chosen wisely, this approach works well [10]. Its effectiveness in complex learning domains has been evaluated by [11], [12], [13], [14] and [15]. However, its ability to abstract successfully is very sensitive to the selection of the subset [16].

In this paper, we propose a novel generalization-based Kanerva approach that explores the near-optimal structure of the set of prototypes (a subset of the state space), and dynamically modifies and fully utilizes the set of prototypes over time. Our algorithm can provide a user-specified level of abstraction when exploring the state space. Specifically, we maintain an additional set of states, the candidate prototype set, that allows each state to learn its value and record its level of generalization. When useless prototypes are removed from the standard set of prototypes, states with certain levels of generalization and well-trained values in the candidate prototype set can be immediately migrated to and used by the standard set of prototypes.

We apply our proposed generalization-based Kanerva coding algorithm to the complex network environment in which we reformulate the congestion control algorithm in TCP. We observe considerably improved performance through a comprehensive simulation study.

2. METHODS

Kanerva-based function approximation approaches are usually initialized with a number of randomly generated states as prototypes, and value updates are applied to those prototypes. Inappropriate allocations of prototypes would result in poor value approximation. However, simply generating new prototypes and/or deleting prototypes usually provides limited improvements to the prototype set. We propose a generalization-based Kanerva coding technique that provides a general methodology to dynamically adjust the potential state space abstractions and manage the levels of generalization on prototypes in an easy and flexible manner.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.
Copyright © 2017, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

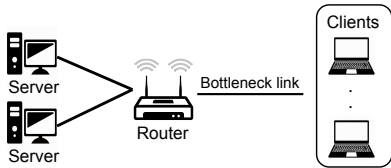


Figure 1: Network topology with a bottleneck link that can be set to a fixed or varying bandwidth.

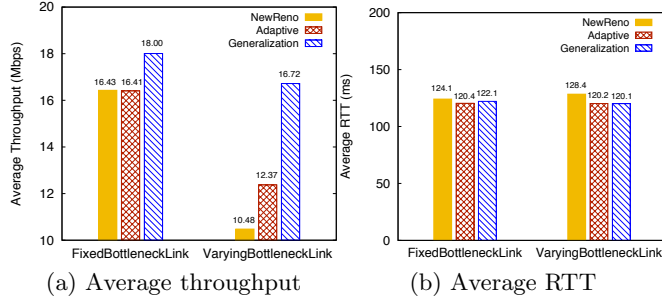


Figure 2: Average throughput and RTT comparisons.

Our generalization-based prototype optimization algorithm, while cooperating with Q-learning to do value updates, follows the following steps to construct successful sets of prototypes. First, we initialize a prototype set S by randomly selecting a number of prototypes from the state space. Second, after every k time steps of learning, we check the level of generalization, that is, the number of adjacent states encountered, for each prototype and dynamically adjust the level of generalization of the whole prototype set S to a desired value v . To do that, we first record the level of generalization of each prototype and then remove prototypes whose levels of generalization are much bigger or smaller than v . Finally, a number of states that have approximately v levels of generalization from the candidate prototype set are selected and introduced to the prototype set S .

One contribution of our technique is that it dynamically identifies prototypes with inappropriate levels of generalization and replaces those prototypes with ones having desired levels of generalization, giving agents the ability to adjust the levels of generalization on prototypes between coarse-grain and fine-grain over time.

To better serve the dynamic generalization on the prototype set, we also propose a prototype migration mechanism that utilizes two sets of states referred to as the hot zone and the cold zone. The hot zone is the regular prototype set used to represent the state abstraction. We apply generalization-based prototype removal and introduction on the hot zone. The cold zone consists of a large number of prototypes that include candidate prototypes and prototypes removed from the hot zone. Prototypes in the cold zone are continuously trained over time. The cold zone is used to provide qualified prototypes that have desired levels of generalization and sufficiently leaned values. With this mechanism, when we need new prototypes to supply to the hot zone, qualified prototypes that have certain generalizing ability can be easily found in the cold zone and when they are migrated to the hot zone, their values are already preset. In addition, since inappropriate prototypes in the hot zone are migrated to the cold zone instead of being deleted, we reduce the risks of permanently losing previously learned values and deleting prototypes with as-yet undiscovered potentials.

3. RESULTS

We embed our learning algorithm in the ns-3 network simulator as a new congestion control protocol and design a typical network topology that can be encountered in real-life network environment. This network topology has continuous, high-dimensional state spaces.

We evaluate the performance of New Reno, which is a standard TCP congestion control protocol, adaptive Kanerva coding, which is the best previously proposed Kanerva-based technique, and our new generalization-based Kanerva coding approach, by applying all to the fixed-bottleneck link network (the bottleneck link has a fixed bandwidth) and the varying-bottleneck link network (the bottleneck link has two bandwidths that alternate) shown in Figure 1. Since each server acts as a learning agent, competition for available bandwidths among different leaning agents exists in this network, making the network environment complex. And the bottleneck link, that has a fixed or varying bandwidth, introduces more dynamics to the network environment.

We repeat our simulation five times using each algorithm in both network settings, and report the average throughput and delay in Figure 2(a) and Figure 2(b). Figure 2(a) shows that in the fixed-bottleneck link network, the average throughput of generalization-based Kanerva coding is 18.0 Mbps which outperforms both adaptive Kanerva coding and New Reno by 9.6%. In the more complex varying-bottleneck link network, the average throughput achieved by our algorithm is 16.7 Mbps which outperforms adaptive Kanerva coding by 35.2% and outperforms New Reno by 59.5%. As shown in Figure 2(b), we observe that both Kanerva-based algorithms achieve lower delay than New Reno, and the generalization-based Kanerva coding has comparable average RTT with adaptive Kanerva coding in both network settings.

4. CONCLUSIONS

Our work describes an effective RL agent that has the ability to handle highly complex domains, making the RL approach widely applicable. The learning agent applies a novel generalization-based Kanerva coding approach to better utilize available bandwidth in network traffic by adjusting the congestion window size to adapt to real-time network congestion. In our experiments, the policy converges quickly to a stable set of actions. Our technique reformulates the RL process with dynamically changed state abstractions, making it possible to abstract useful information from the rich details of the environment, while using a very small set of prototypes to approximate the value functions as closely as possible.

In the experiments, our learning agent achieved better throughput and delay performance than both the best-known Kanerva-based algorithm and New Reno in a complex network topology. We conclude that generalization-based Kanerva coding can be used to manage congestion in complex networks, and that the technique enables effective state space abstraction in reinforcement learners.

5. ACKNOWLEDGMENTS

This work was supported in part by the U.S. Office of Naval Research under grant number N00014-16-1-2651.

6. REFERENCES

- [1] Peggy Fidelman and Peter Stone. Learning ball acquisition on a physical robot. In *2004 International Symposium on Robotics and Automation (ISRA)*, page 6, 2004.
- [2] Andrew Y Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Autonomous inverted helicopter flight via reinforcement learning. In *Experimental Robotics IX*, pages 363–372. Springer, 2006.
- [3] Martin Allen and Phil Fritzsche. Reinforcement learning with adaptive kanerva coding for xpiilot game ai. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1521–1528. IEEE, 2011.
- [4] Lutz Frommberger. *Qualitative Spatial Abstraction in Reinforcement Learning*. Springer Science & Business Media, 2010.
- [5] J. Albus. *Brains, Behaviour, and Robotics*. McGraw-Hill, 1981.
- [6] Shimon Whiteson, Matthew E Taylor, Peter Stone, et al. *Adaptive tile coding for value function approximation*. Computer Science Department, University of Texas at Austin, 2007.
- [7] Stephen Lin and Robert Wright. Evolutionary tile coding: An automated state abstraction algorithm for reinforcement learning. In *Abstraction, Reformulation, and Approximation*, 2010.
- [8] Sonia Chernova and Manuela Veloso. Tree-based policy learning in continuous domains through teaching by demonstration. In *Proceedings of Workshop on Modeling Others from Observations (MOO 2006)*, 2006.
- [9] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [10] Cheng Wu and Waleed Meleis. Function approximation using tile and kanerva coding for multi-agent systems. In *Proc. Of Adaptive Learning Agents Workshop (ALA) in AAMAS*, 2009.
- [11] B. Ratitch and D. Precup. Sparse distributed memories for on-line value-based reinforcement learning. In *Proc. of the European Conf. on Machine Learning*, 2004.
- [12] Cheng Wu and Waleed M Meleis. Adaptive kanerva-based function approximation for multi-agent systems. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pages 1361–1364. International Foundation for Autonomous Agents and Multiagent Systems, 2008.
- [13] Cheng Wu and Waleed Meleis. Adaptive fuzzy function approximation for multi-agent reinforcement learning. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, pages 169–176. IEEE Computer Society, 2009.
- [14] Cheng Wu and Waleed Meleis. Fuzzy kanerva-based function approximation for reinforcement learning. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 1257–1258. International Foundation for Autonomous Agents and Multiagent Systems, 2009.
- [15] Cheng Wu, Waleed Meleis, et al. Rough sets-based prototype optimization in kanerva-based function approximation. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 2, pages 283–291. IEEE, 2015.
- [16] P. W. Keller, S. Mannor, and D. Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proc. of Intl. Conf. on Machine Learning*, 2006.