Context-Based Concurrent Experience Sharing in Multiagent Systems

(Extended Abstract)

Dan Garant
University of Massachusetts
Amherst, MA 01002
dgarant@cs.umass.edu

Bruno C. da Silva
Federal University of Rio Grande do Sul. Porto Alegre, Brazil 91540-000
bsilva@inf.ufrgs.br

Victor Lesser
University of Massachusetts
Amherst, MA 01002
lesser@cs.umass.edu

Chongjie Zhang
Tsinghua University
Beijing, China
chongjie@tsinghua.edu.cn

ABSTRACT

One of the key challenges for multi-agent learning is scalability. We introduce a technique for speeding up multi-agent learning by exploiting concurrent and incremental experience sharing. This solution adaptively identifies opportunities to transfer experiences between agents and allows for the rapid acquisition of appropriate policies in large-scale, stochastic, multi-agent systems. We introduce an online, supervisor-directed transfer technique for constructing high-level characterizations of an agent's dynamic learning environment—called contexts—which are used to identify groups of agents operating under approximately similar dynamics within a short temporal window. Supervisory agents compute contextual information for groups of subordinate agents, thereby identifying candidates for experience sharing. We show that our approach results in significant performance gains, that it is robust to noise-corrupted or suboptimal context features, and that communication costs scale linearly with the supervisor-to-subordinate ratio.

Keywords

Transfer Learning; Multi-agent Systems; Reinforcement Learning

1. INTRODUCTION

In large-scale multi-agent systems consisting of hundreds to thousands of reinforcement-learning agents, convergence to a near-optimal joint policy, when possible, may require a large number of samples. These systems, however, may contain groups of agents working on nearly identical local tasks or under approximately similar environmental dynamics. Identifying such groups may prove useful in cooperative domains, due to the opportunity of exploiting shared information. Information sharing has been extensively studied in single-agent settings with the goal of transferring knowledge from a source task to novel tasks [11, 6, 2]. Applying this idea to the multi-agent setting (MAS), it is apparent that experiences may be transferred not only across similar tasks, but also between concurrently-learning agents in a shared environment. This paper focuses on the problem of online transfer of experiences between such agents—with an emphasis on the adaptive discovery of groups of agents where experience sharing is possible and beneficial.

In multi-agent settings, agents interact and learn concurrently. It is difficult to identify when experiences may be usefully exchanged and reused by other agents, since they might be operating under different local environments and may be interacting with different types of neighbors. To address this issue we propose modeling *contexts* as dynamic local characterizations of the environment under which agents learn. They are defined over short timescales during which policies and models are approximately static. We introduce a context-similarity measure grounded in the comparison of abstract representations of environment dynamics, and advocate the use of supervisory agents as a way of identifying contextually-compatible agent groups where experiences may be shared.

In the single-agent setting, many metrics for comparing learning environments (and determining transfer opportunities) exist. Most are based on comparing policies, Q-values, or reward functions [1, 4, 10]. These are often negatively impacted by the existence of multiple optimal policies and by estimates constructed with different numbers of experiences [9, 7]. Transfer methods also exist to address multi-agent settings, but most assume that agents do not interact or that mappings from source to target tasks are available [5, 1, 10, 8]. To our knowledge, our algorithm[1] is the first to allow experience sharing in concurrent and interacting MAS with $\sim$1000 agents with low communication and computational overhead. We show that *1)* its complexity scales with the number of agents in each supervisory group, not the number of agents in the system; *2)* its communication costs scale linearly with the supervisor-to-subordinate ratio; and *3)* it is robust to suboptimal context features.

2. CONTEXT-BASED LEARNING

Context features are compact abstractions of the local learning environment under which an agent operates. We wish to capture a measure of *context compatibility*: if agents are working under a same local transition and reward model, they face a same learning problem and experiences may be transferred. We rely on context features to form broad-scope summaries, or *abstractions*, of transition and reward models as experienced by individual agents. Our method determines sharing opportunities by grouping agents based

[1]For a more comprehensive discussion of our method and experiments see `https://arxiv.org/abs/1703.01931` [3].

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.*

on their local learning environments (*contexts*). Each agent collects observations from its environment in the form of state, action, reward, and next state tuples. Every K time steps, agents report such observations to their supervisors. Supervisors aggregate information from all subordinates to compute *context summary vectors*, one per agent. These vectors are dynamic local characterizations of the environment under which agents operate, and are used to identify sharing opportunities. Supervisors measure the similarity between the context summary of each subordinate with respect to a covariance-appropriate and scale-independent metric; similar agents are organized into sharing groups, and supervisors relay experiences between members of a sharing group. The system periodically regroups agents according to updated contextual information.

A supervisor overseeing n agents computes contextual information for each subordinate by using a function f mapping experiences to an n-tuple of context summary vectors. Each such vector is a sample from the (latent) underlying context distribution characterizing the agent's local environment. Our method identifies sharing opportunities via a stochastic sampling process that probabilistically partitions agents into sharing groups, given their contextual similarity. In particular, agents $h, j \in A$ are marked as *context-compatible* with probability $P_{h,j}$ based on the similarity of their context summary vectors, $\phi(V_h, V_j)$, where ϕ is a suitable kernel:

$$P_{h,j} = \frac{\exp\left(\phi(V_h, V_j)\right)}{\sum_{a,b \in A, a \neq b} \exp\left(\phi(V_a, V_b)\right)} \quad (1)$$

Agents operating under similar local dynamics are near in context space and have a higher probability of undergoing sharing. Once sharing groups have been determined, supervisors relay experiences within each group; agents incorporate them into their policies using any off-policy learning algorithm. The complexity of our method is $O(dn^3)$, where d is the dimension of a context summary vector and n is the number of agents in a supervisory group. Our method scales independently of the number of supervisors.

3. EXPERIMENTS

We evaluated our algorithm on large network-distributed task allocation problems, where agents can choose to work on tasks or to forward them to a neighbor. Tasks are generated according to patterns that are *unknown* to the agents, which makes the problem non-stationary. Agents learn policies using an extension of Q-Learning to the multi-agent case with stochastic policies [12]. Context features are defined as an agent's task load relative to its neighbors, and the rate at which each neighbor receives tasks from the environment and from other agents. Networks are lattices of up to 729 agents and were constructed by varying the task creation frequency and the region where tasks may originate. Four supervisory structures were considered: two baseline configurations (one with no supervision/no information sharing, and one with a single supervisor, where *all* agents may share information), as well as intermediate configurations with 4 and 9 supervisors. We measure performance as the area under the curve of service time as a function of time: when the system converges quickly, this area is small. Figure 1 shows that the single-supervisor configuration far outperforms the baseline approach with no transfer, with information-sharing agents accumulating nearly *half* the area under the curve compared to agents that do not share experiences. Adding more supervisors diminishes this benefit, since there are fewer sharing opportunities. Even with a high supervisor-subordinate ratio of 1:10, however, sharing still results in a *25% reduction* in the area under the curve.

Given these results, one could expect that information sharing becomes more beneficial as the system grows: larger systems have a more diverse pool of agents that may benefit from sharing. To

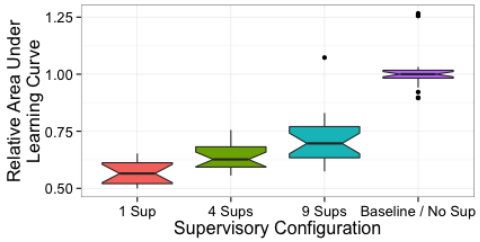

Figure 1: Performance of different supervisory configurations in a 100-agent network (smaller values=faster learning).

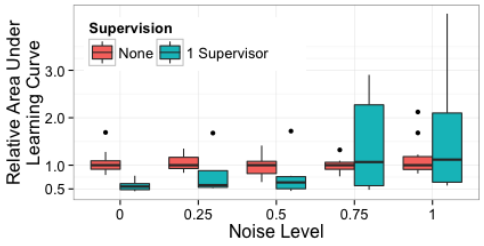

Figure 2: Robustness under sub-optimal context features.

test this hypothesis, we evaluated our system on networks with different numbers of agents (100, 324, 729) and different task-generation patterns and frequencies. We observed that performance in a 100-agent network was roughly *30% higher* than the baseline. As network size increased to 729 agents, performance median *improved by 40%* compared to the baseline. We also observed that the communication volume incurred by our method was invariant with respect to K: on average 43 bytes per step per subordinate. This suggests that these costs scale *linearly* with the supervisor-to-subordinate ratio, and that even when accounting for communication overhead, more distributed configurations tend to perform better. All information-sharing configurations we evaluated surpassed the baselines while incurring low communication overhead. Finally, we analyzed our method's robustness to corrupted or sub-optimal context features. Poorly-constructed features that do not properly abstract the underlying learning environment make it difficult to identify sharing opportunities. We added different levels of normally-distributed noise to context features. Figure 2 shows that when noise dominates (approaches 1), performance becomes increasingly volatile. As features become less meaningful, our mechanism is equally likely to achieve a 50% reduction in the learning curve area as it is to increase it by 100%. In other words, when the information-sharing process is guided by ill-specified features, there is no consistent positive or negative impact on performance; the most prominent impact is on performance *variability*.

4. DISCUSSION

We introduced a method that adaptively identifies sharing opportunities between context-compatible agents, where contexts provide abstract characterizations of local learning environments. It scales with the number of agents in each supervisory *group*, not in the entire system. Experiments suggest significant performance improvements over baseline settings with no experience sharing, and quantitative analyses demonstrate that sharing becomes increasingly advantageous as the system grows. We also show that our method is robust to suboptimal context features and that communication costs scale linearly with the supervisor-to-subordinate ratio.

REFERENCES

[1] G. Boutsioukis, I. Partalas, and I. Vlahavas. Transfer learning in multi-agent reinforcement learning domains. In *Recent Advances in Reinforcement Learning*, pages 249–260. Springer, 2012.

[2] J. L. Carroll and K. Seppi. Task similarity measures for transfer in reinforcement learning task libraries. In *Proceedings of the International Joint Conference on Neural Networks*, pages 803–808. IEEE, 2005.

[3] D. Garant, B. C. da Silva, C. Zhang, and V. Lesser. Context-based concurrent experience sharing in multiagent systems. *ArXiv e-prints*, March 2017.

[4] Y. Hu, Y. Gao, and B. An. Learning in multi-agent systems with sparse interactions by knowledge transfer and game abstraction. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 753–761. IFAAMAS, 2015.

[5] R. M. Kretchmar. Parallel reinforcement learning. In *Proceedings of the 6th World Conference on Systemics, Cybernetics, and Informatics*, 2002.

[6] A. Lazaric, M. Restelli, and A. Bonarini. Transfer of samples in batch reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pages 544–551. ACM, 2008.

[7] M. L. Littman. Value-function reinforcement learning in markov games. *Cognitive Systems Research*, 2(1):55–66, 2001.

[8] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *CoRR*, abs/1602.01783, 2016.

[9] B. Price and C. Boutilier. Accelerating reinforcement learning through implicit imitation. *Journal of Artificial Intelligence Research*, 19:569–629, 2003.

[10] A. Taylor, I. Duparic, E. Galván-López, S. Clarke, and V. Cahill. Transfer learning in multi-agent systems through parallel transfer. In *Theoretically Grounded Transfer Learning at the International Conference on Machine Learning*, 2013.

[11] M. E. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *The Journal of Machine Learning Research*, 10:1633–1685, 2009.

[12] C. Zhang and V. Lesser. Multi-Agent Learning with Policy Prediction. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 927–934, Atlanta, 2010.