Agent abstraction for scalable theory-of-mind in many-agent systems

Anonymous Author(s)
Submission Id: 1105

ABSTRACT

In pursuit of goals, adaptive agents narrow their behaviour. The more agents there are in a system, the greater is the reduction in joint policy entropy and the more benefit is gained from exploiting this order to represent the system more efficiently and adapt to it more quickly. This can be especially true when the agents pursue goals by coordinating. Here, inspired by the notion of state abstraction, we formulate optimal agent and joint action space abstractions, provide illustrative examples, and present experiments where agent abstraction leads to improved sample efficiency during theory of mind prediction. Formulated for arbitrary joint policies, our approach provides a potentially more versatile alternative to mean-field approaches in making learning in many-agent systems tractable.

KEYWORDS

Abstraction, Many-agent systems, Efficient planning, Theory of Mind

ACM Reference Format:
Anonymous Author(s). 2023. Agent abstraction for scalable theory-of-mind in many-agent systems. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 4 pages.

1 INTRODUCTION

- learning in multi-agent settings hard because have to learn about other agents
- The additional difficulty in many-agent settings in particular: can't possibly model them all!
- highlighting the two perspectives of being in the system or just observing it (theory of mind)
- theory of mind: brief outline of this research (psych?, cogsci?, in AI), mention machine theory of mind [1].
- abstraction also natural for multi-agent systems for which collective effects dominant, e.g. flocking (or brains: Schneidman et al 2006.)
- abstraction as useful when system is compressible. What determines compressability? Structure. Independent structure vs. coordinated structure. The distinction between compressability, coordination, and cooperation.
- briefly mention of Other approaches: mean field (limitations in its own section?)
- Our contributions
  - Definitions of agent and action space abstraction
  - Formulation of an optimal abstraction
  - Abstraction-dependent bounds on TOMM learning problem
  - Experiments demonstrating improvements

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom. © 2023 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2 THEORY

2.1 Agent and action space abstraction

Here, we illustrate agent abstraction through formalism adopted from standard multi-agent reinforcement learning. Without loss of generality, consider an $N+1$-agent system from the perspective of agent $N+1$. With state space $\mathcal{S}$ and action space for each agent $\mathcal{A}_i$ for $i = 1, \dots, N+1$, let $T(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ be the system's (possibly stochastic) transition function with $s \in \mathcal{S}$ and $s' \in \mathcal{S}$ the current and next state, respectively, and $\boldsymbol{a} = (a_1, \dots, a_N, a_{N+1})$ the joint action of the $N+1$ agents, with $a_i \in \mathcal{A}_i$ sampled from the ith agent's policy $\pi_i(a_i|s)$. From the perspective of agent $N+1$, the effective system transition function, T_{N+1}, is

$$T_{N+1}(\boldsymbol{s}'|\boldsymbol{s}, a_{N+1}) = \sum_{a_1,\dots,a_N} T(\boldsymbol{s}'|\boldsymbol{s}, \underbrace{\boldsymbol{a}) \times \pi_1(a_1|\boldsymbol{s}) \times \cdots \times \pi_N(a_N|\boldsymbol{s})}_{\text{target for abstraction}} \,.$$

Even in decentralized and model-free settings, it is then necessary for agent $N+1$ to predict the actions of other agents to make optimal decisions and maximize rewards. For such an embedded agent, the utility of abstraction arises from more efficient (e.g. transferable) learning. However, even for an external agent tasked with building a *theory of multiple minds* (TOMM), *i.e.* a model of the N-agent system, the observations of $\boldsymbol{a}$ can serve as the signal with which an agent and action space abstraction can be learned and a more efficient representation of the system obtained. We leave further development of the embedded agent problem to future work, hereon omit mention of agent $N+1$, and focus on the TOMM problem of joint policy estimation of N agents.

We define an *agent abstraction* as a partition $\mathcal{K} = \{\boldsymbol{k}_k\}_{k=1}^{K}$ of N agents into K abstract agents ($1 \le K \le N$), such that the kth partition element abstracts $|\boldsymbol{k}_k|$ agents into a single abstract agent. Note $\sum_{k=1}^{K} |\boldsymbol{k}_k| = N$. Given an agent abstraction, extending the notion of state abstraction to *action space abstraction* defines the abstraction map from the joint action space of N other agents to the joint action space of K abstract agents, $\tilde{\boldsymbol{a}} = \phi_{\mathcal{K}}(\boldsymbol{a})$,

$$\phi_{\mathcal{K}} : \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \to \tilde{\mathcal{A}}_1 \times \cdots \times \tilde{\mathcal{A}}_K, \qquad (1)$$

where $\tilde{\mathcal{A}}_k$ is the action space of the kth abstract agent of size $m_k := |\tilde{\mathcal{A}}_k| \le \prod_{i \in \boldsymbol{k}_k} |\mathcal{A}_i|$. The abstract joint action is denoted as $\tilde{\boldsymbol{a}} = (\tilde{a}_1, \dots, \tilde{a}_K)$ with $\tilde{a}_k \in \tilde{\mathcal{A}}_k$. Action policies for abstract agents, denoted $\tilde{\pi}_k$, similarly define the joint policy with abstraction, $\tilde{\boldsymbol{\pi}} = (\tilde{\pi}_1, \dots, \tilde{\pi}_K)$.

Table 1: The 4 distinct ways of abstracting 3 agents where the first 2 agents are identical (single-agent action space size, A).

Label	Partition $\mathcal{K}$ of $N = 3$ agents	Element size, $\lvert \boldsymbol{k}_k \rvert$	Action space size, m_k	Sum	Product
correct	$\{1, 2\}, \{3\}$	2, 1	$A,\ A$	$2A$	A^2
super	$\{1, 2, 3\}$	3	A^2	A^2	A^2
granular	$\{1\}, \{2\}, \{3\}$	1, 1, 1	$A,\ A,\ A$	$3A$	A^3
rest	$\{1, 3\}, \{3\}$ and $\{1\}, \{2, 3\}$	2, 1 and 1, 2	$A^2,\ A$ and $A,\ A^2$	$A + A^2$	A^3

2.2 Illustrative Agent Abstraction Example

Take the case of abstracting $N = 3$ agents. To illustrate the different agent abstractions, let's have it that agent 1 and 2 are actually a team acting fully correlated with each other (e.g. $a_1 = a_2$)[1], while agent 3 acts independently. In this case, some compression, i.e., reduced size of the joint action space representation ($\boldsymbol{a} = (a_1, a_2, a_3)$), should be possible in the (a_1, a_2)-subspace without sacrificing performance. The 5 possible agent abstractions of the $N = 3$ agents are shown in table 1, where the abstraction simply leaves out joint actions that are never visited. Here, we list the sizes of the action space of each abstraction element and have computed their sum and product. The sum is the number of parameters of the abstracted agent's policies and describes the difficulty of learning a model that can be used for planning to find an optimal policy. The product is the number of parameters of the abstracted value function and describes the difficulty of learning a policy that maximizes rewards. The *correct* agent abstraction, i.e. the one that reflects the actual (here perfect) correlations in the joint action vector (between a_1 and a_2), has the minimum size and thus maximum compression for both the sum and product objectives without further knowledge of the actual policies being used. For the sum objective the optimal agent abstraction is uniquely the *correct* one.

2.3 Agent abstraction as optimal compression

Action space abstraction exploits structure (*i.e.* non-uniformity) in the probability distribution of the joint action subspace for each abstract agent. In the illustrative example (section 2.2), this was about exploiting the perfect correlation, which concentrated the distribution of the pair of actions onto only two of the four possible states. More generally, useful abstraction is possible whenever the joint action probability distribution is non-uniform, which will typically be the case even in the absence of agent-agent action correlation [2]. This can be stated formally by rephrasing Shannon's source coding theorem: the optimal lossless compression of the joint policy of the $\lvert \boldsymbol{k}_k \rvert$ agents grouped into the single kth abstract agent gives an abstract action space with 2^H equally probable abstracted actions, where H is the conditional entropy (in bits) of the joint policy distribution conditioned on the state (note $H \leq \sum_{i \in \boldsymbol{k}} \log_2 \lvert \mathcal{A}_i \rvert$, i.e. the entropy of the uniform distribution).

Lossy compression allows for arbitrary smaller representations at the expense of some loss in an objective (e.g. reconstruction). For example, another way to exploit non-uniformity is by absorbing specific joint actions having small expected contribution to the objective into some other (typically higher probability) ones. In the worst case, all the expected reward from these absorbed joint actions is lost, but this contribution is bounded by construction.

For the TOMM problem, the loss arises from estimation error in the joint policy. In the general stochastic setting we consider, this loss is typically a negative log-likelihood or probability of misestimation [3]. An optimal abstraction is found by solving the following constrained optimization problem:

$$\min_{\phi} \ f(\phi) \tag{2}$$

$$\text{subject to} \ \ \mathrm{Loss}(\phi) \leq \varepsilon, \tag{3}$$

for the abstraction objective f and loss tolerance ε. As discussed in the previous section and table 1, we consider both the sum objective, $f_{\text{sum}}(\phi) = \sum_{k=1}^{K} m_k$, and product objective $f_{\text{product}}(\phi) = \prod_{k=1}^{K} m_k$, where m_k is the size of the action space of the kth abstract agent (*c.f.* columns 4, 5 and 6 of table 1). [conclusing sentance setting up next section]

2.4 Agent abstraction affects learning efficiency

A TOMM problem focuses on estimating the joint policy function $\boldsymbol{\pi}(\boldsymbol{a}|s)$, a state-conditioned distribution over the space of joint actions, $\boldsymbol{a} \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$. The ground truth is that each agent acts independently given the state so that $\boldsymbol{\pi}(\boldsymbol{a}|s) = \prod_{i=1}^{N} \pi_i(a_i|s)$, i.e. a policy for each agent. There are then $\sum_{i=1}^{N}(\lvert \mathcal{A}_i \rvert - 1)\lvert \mathcal{S} \rvert$ probabilities in this case. However, there can be an advantage in terms of learning efficiency to instead group agents and learn the parameters of these grouped, or abstract policies over joint action subspaces. In the extreme case that all agents' models and their policies are in fact copies of one single policy with action space $\mathcal{A}$, this reduces the number of parameters that need to be learned by a factor of N from $\sum_{i=1}^{N}(\lvert \mathcal{A}_i \rvert - 1)\lvert \mathcal{S} \rvert = N\lvert \mathcal{S} \rvert(\lvert \mathcal{A} \rvert - 1)$ to $\lvert \mathcal{S} \rvert(\lvert \mathcal{A} \rvert - 1)$. However, in the case that all joint action probabilities are non-negligible (e.g. of the same order as they are in the maximum entropy distribution) so that no action space abstraction is possible, the number of abstract actions needed to represent the space of observed joint actions grows exponentially with the number of agents grouped. In the extreme case, grouping all agents into a single abstract agent demands that $\lvert \mathcal{S} \rvert(\lvert \mathcal{A} \rvert^N - 1)$ parameters be learned, which is on the order $\lvert \mathcal{A} \rvert^{N-1}/N$ times larger than in the ground truth case and so blows up when there are many agents ($N \gg 1$)! An example of this fact for 2 agents is given in fig. 1, where it is clear that abstraction

[1]Perfect anti-synchrony would also work the same in this example.

[2]In particular, even when there is zero mutual information between its components, the joint action distribution can be highly structured. For example, let $\boldsymbol{x}$ denote the N-dimensional random variable with independent components $x_i \sim \text{Bernoulli}(p)$. The joint distribution $P(\boldsymbol{x})$ has an entropy that is a factor $0 \leq H(p) \leq 1$ smaller than the maximum over all $P(\boldsymbol{x})$ and $H(p)$ goes to 0 as p goes to 0 or 1.

[3]For the embedded agent perspective, loss could be defined as the step-wise regret incurred by agent N+1 for using the abstraction, $\mathrm{Loss}(\phi) := \rho^{\pi^*} - \rho^{\bar{\pi}^*}(\phi)$, where the optimal average reward of the ground MDP, ρ^{π^*}, upper bounds the optimal average reward for the abstraction ϕ, $\rho^{\bar{\pi}^*}(\phi)$.

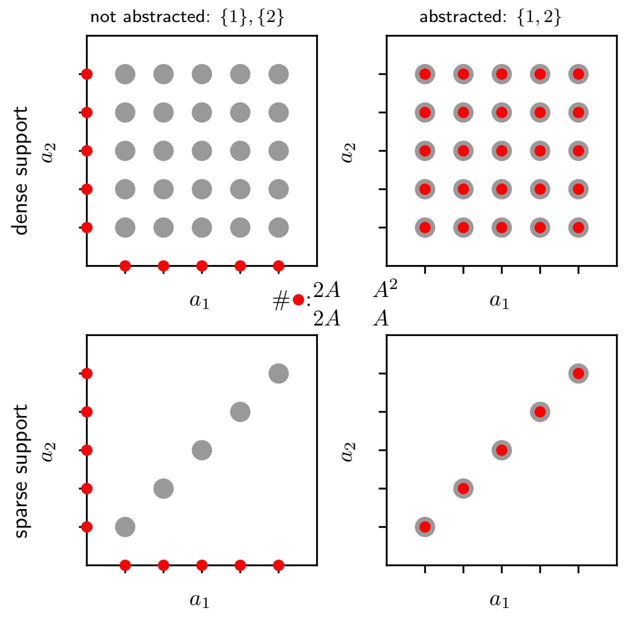

Figure 1: Sparsity favors abstraction. For the single agent action space size, $A = 5$, the support (gray) over the joint action space for 2 agents can be dense (top row) or sparse (bottom row). Joint action space is naturally parametrized using a product encoding of each agent's action (left column). The space can be abstracted by instead encoding the action pairs (right column). Parameters (red) count for each of the 4 combinations shown at the figure's center.

is beneficial when the support of the joint policy is sparse and detrimental when it is dense. Abstraction is thus only beneficial when the state-joint action probabilities are sufficiently distinct that an abstraction that groups them into fewer actions with minimal loss is possible.

The number of parameters is only a proxy for learning inefficiency. What is more relevant is *sample complexity*: how can agent and action abstraction reduce the number of samples needed to estimate the joint policy *to a given precision*. We can make these arguments more formal using the PAC framework from statistical learning theory as follows.

For a given agent and action abstraction, we consider the online learning problem that estimates the probabilities of the corresponding abstract policy from the observation of a T-length sequence of state-joint action tuples, $\mathcal{D}_T = \{(s_t, \boldsymbol{a}_t)\}_{t=1}^T$. We assume each step provides an *iid* sample of $(s, \boldsymbol{a})$ from an unknown distribution $p(s,\boldsymbol{a}) = \boldsymbol{\pi}(\boldsymbol{a}|s)p(s)$. All sample-based quantities depend on the realization of $\mathcal{D}_T$ even though we will not denote this dependence explicitly. Let $m_T(s,\boldsymbol{a})$ denote the number of times the state-joint action tuple $(s,\boldsymbol{a})$ appears in $\mathcal{D}_T$. These state-joint action counts map through the given abstraction to counts of state-abstract action tuples for each abstract agent. Under the *iid* assumption and without prior information, the count distribution is multinomial for which the maximum likelihood estimate (MLE) of the probability given the observed sequence is simply the empirical frequency. We aim to obtain the convergence rates of this estimator onto the true probability to see how they are affected by agent and action abstraction.

We first present the baseline case of no abstraction and for only a single agent and its corresponding set of $|\mathcal{S}| \times (|\mathcal{A}| - 1)$ count variables, $m_T(s, a_i)$, for agent $i \in \{1, \dots, N\}$. The MLE for $p(s, a_i)$ is $\hat{p}_T(s, a_i) = m_T(s, a_i) / \sum_{(s,a_i)} m_T(s, a_i)$. With an estimate of all $p(s, a_i)$ probabilities, we can form an estimate of agent i's policy from the ratio estimator

$$\hat{p}_T(a_i|s) := \hat{p}_T(s,a_i) \Big/ \sum_{a_i \in \mathcal{A}} \hat{p}_T(s,a_i), \tag{4}$$

whose distribution propagates the uncertainty in $\hat{p}_T(s,a_i)$.

With the estimator defined, we can apply PAC theory to obtain its convergence. As an example, take the general case of N agents each having a distinct action space $\mathcal{A}_i$, but in an environment with only a single state, s, for simplicity. The probability of the joint event that estimated probability for each and every agent falls within ε factorizes into the product of probabilities of each event individually,

$$\begin{aligned}
&\mathbb{P}\left(\bigwedge_{i=1}^{N} \left\{ ||\hat{p}_T(s,a_i) - p(s,a_i)|| < \varepsilon \right\} \middle| s \right) && (5)\\
&= \prod_{i=1}^{N} \mathbb{P}\left(||\hat{p}_T(s,a_i) - p(s,a_i)|| < \varepsilon \middle| s \right) && (6)\\
&= \prod_{i=1}^{N} \left[1 - \mathbb{P}\left(||\hat{p}_T(s,a_i) - p(s,a_i)|| \geq \varepsilon \middle| s \right) \right] && (7)\\
&\geq \prod_{i=1}^{N} \left[1 - \delta_i \right], && (8)
\end{aligned}$$

where in the last line we have used an upperbound, δ_i on the probability that the estimate for agent i deviates away from the actual value by at least ε. Using standard bound derivation techniques (see the appendix), we can use $\delta_i := \exp\left(-\frac{\varepsilon^2}{16|\mathcal{A}_i|} \cdot \frac{T^2}{T-1} + \frac{1}{4} \right)$. We can thus define the lower-bound $1 - \delta$ on the joint-event probability using the deviation $\delta := 1 - \left[1 - \delta_{\max}\right]^N$, where $\delta_{\max}$ is the deviation for the agent with the largest $|\mathcal{A}_i|$, denoted $A_{\max}$. Then we have the desired estimation precision $\varepsilon > 0$, with probability at least $1 - \delta$ if T satisfies

$$\frac{A_{\max}}{\varepsilon^2} \left(4 - 16 \log \left(1 - (1-\delta)^{\frac{1}{N}} \right) \right) \leq T, \tag{9}$$

for $T \gg 1$. This lower bound grows linearly with the largest single-agent action space and as logarithmically with N (when N is large) as more samples are needed to achieve the desired precision, ε.

When joining agents together in an agent abstraction, the N is lowered, but $A_{\max}$ can increase, exponentially in the worst case. It is thus the tradeoff between the two that determines the optimal amount of abstraction in terms of the lowest lowerbound on the number of samples needed.

Take the case of $|\mathcal{A}_i| = 2$ ($a_i \in \{0,1\}$) in which case the policy of agent i is a Bernoulli distribution for each state. In this simple case, there is a known PAC bound for $\hat{p}_T(s, a_i = 1)$ assuming s is fixed. It states that for any $\varepsilon > 0$,

$$\mathbb{P}\left(|\hat{p}_T(s, a_i = 1) - p(s, a_i = 1)| \geq \varepsilon\right) \leq \delta_i := 2e^{-2T\epsilon^2} \; . \qquad (10)$$

Hence, the precision is achieved for all N agents with probability at least $1 - \delta$ if T satisfies

$$\frac{2}{\varepsilon^2}\left(\frac{1}{4}\log 2 - \frac{1}{4}\log\left(1 - (1-\delta)^{\frac{1}{N}}\right)\right) \leq T \; . \qquad (11)$$

Our more general bound eq. (9) reduces to this established PAC bound when evaluated on $A_{\max} = 2$ (at least in scaling behaviour; the coefficients and constant term of the known result give a tighter bound). Our goal here is not to achieve a better bound, but to illustrate the generality of the form of scaling dependence.

3 EXPERIMENTS

Our experiments are motivated by demonstrating the benefits of agent abstraction in solving a TOMM problem in a contemporary multi-agent reinforcement learning setting.

3.1 Petting Zoo's Simple Tag

Here, agents aim to ...[TASK DESCRIPTION]...

3.2 MAagent's Battle

Here, agents aim to ...[TASK DESCRIPTION]...

3.3 Supervised learning of the joint policy

We trained the agents in each respective environment. We then rolled out the trained policies into a dataset of sampled episodes, which served as the targets to train abstractions via supervised learning.

4 RELATED WORK

Theory of Machine Mind [1], State abstraction, ...

5 DISCUSSION & FUTURE WORK

- full joint action observation/timestep means that observations aren't limited and so PAC bounds only depend on the largest action space. If observer had to learn policies from observing interactions among limited number of agents (worst case one by one), then all action space sizes would make an appearance in the PAC bound?
- ...

Figure 2: Performance for different agent abstractions

REFERENCES

[1] Neil C Rabinowitz, Frank Perbet, H Francis Song, Chiyuan Zhang, and Matthew Botvinick. 2018. Machine Theory of mind. In *35th International Conference on Machine Learning, ICML 2018*, Vol. 10. 6723–6738. arXiv:1802.07740