Narrative Planning in Large Domains through State Abstraction and Option Discovery

Mira Fisher
Narrative Intelligence Lab, Department of Computer Science, University of Kentucky
329 Rose Street
Lexington, Kentucky 40506
mira.fisher@uky.edu

Abstract

Low-level game environments and other simulations present a difficulty of scale for an expensive AI technique like narrative planning, which is normally constrained to environments with small state spaces. Due to this limitation, the intentional and cooperative behavior of agents guided by this technology cannot be deployed for different systems without significant additional authoring effort. I propose a process for automatically creating models for larger-scale domains such that a narrative planner can be employed in these settings. By generating an abstract domain of an environment while retaining the information needed to produce behavior appropriate to the abstract actions, agents are able to reason in a lower-complexity space and act in the higher-complexity one. This abstraction is accomplished by the development of extended-duration actions and the identification of their preconditions and effects. Together these components may be combined to form a narrative planning domain, and plans from this domain can be executed within the low-level environment.

Introduction

Some expensive AI techniques are especially constrained with respect to the size of problems they can be used on. Many modern environments, such as those which appear in open world games or other simulations, cannot benefit from these technologies, and this limitation prevents some exciting developments in AI from reaching wider settings. Narrative planning is one such exciting technology, able to endow agents with typically-elusive abilities like the ability to plan to cooperate with other agents, but limited by the expense of computation. Without significant authoring effort, this type of technique cannot be deployed in settings that are not expressly designed for it.

I propose that this problem may be overcome by developing an abstraction of a domain, including capturing the information necessary for navigating the initial domain with abstract actions. A suitable abstraction should develop high-level *features* composed from key features in the lower-level domain, *actions* for affecting changes to those features, and *policies* for accomplishing the required effects of these actions in the lower-level environment. Taken as a whole this represents a high-level model of the domain, representative of an understanding of how the environment relates to the tasks at hand. This type of abstraction is normally a task for human authors, but by putting it in the hands of agents we save on effort and required expertise.

The research I propose is to construct a method or collection of methods which augment an environment, represented as a Markov Decision Process (MDP), with extended-duration actions, yielding a semi-Markov Decision Process (sMDP) (Sutton, Precup, and Singh 1999). These actions (variously referred to as options, sub-tasks, sub-goals, and skills) are developed such that when one is completed others are guaranteed to be available, and as a result the states the environment assumes between these actions can be extracted as a higher-level state space. The system may then identify preconditions and effects for these actions within this state space, developing a narrative planning domain such that a narrative planner may be applied to the abstract task domain. Finally, plans can be executed in the original environment by executing the actions provided by the sMDP.

Related Work

While much of the work in this project is not necessarily associated with the training mechanisms that define reinforcement learning, the development of complex "macro-operators" from the primitive operators in an MDP has been previously explored as hierarchical reinforcement learning (HRL). The *options* framework describes these complex operators as *policies* that may begin executing when an agent is in the operator's *initiation set* and stop executing according to a *termination condition* (Sutton, Precup, and Singh 1999). Other HRL approaches—Hierarchies of Abstract Machines (Parr and Russell 1997) and MAXQ value function decomposition (Dietterich 2000)—present related methods of abstraction, but I focus on options due to the wealth of research underpinning that framework. In particular, extensive work into *option discovery* provides a breadth of methods to draw inspiration from, such as: identifying strongly connected regions of states and defining options that escape them (Davoodabadi and Beigy 2011), linking distant parts of the state space (Jinnai et al. 2019), exploring in specific directions through the state space (Machado, Bellemare, and Bowling 2017), navigating to bottleneck states (Stolle 2004; Menache, Mannor, and Shimkin 2002), developing an agent's basic skills of movement and interaction

(Konidaris and Barto 2007; James, Rosman, and Konidaris 2018, 2020; Kulkarni et al. 2016), and deliberately chaining together options for effective problem solving (Bagaria, Senthil, and Konidaris 2021). Many methods have been considered for the development of options, but the focus on solving tasks rather than modeling the domain prevents these from being directly applied to solve the problem introduced here.

State abstraction in MDPs provides another key underpinning of this work, as the development of an abstract state space which an agent can operate within is vital for reducing complexity to a point that symbolic planning can be used. Existing abstraction methods identify symmetries within an MDP (Ravindran and Barto 2002), utilize relational operators (Croonenborghs, Driessens, and Bruynooghe 2007), or develop a symbolic model of the MDP based on the results of executing actions available to the agent (Andersen and Konidaris 2017). There is reason to believe that a symbolic model of the environment should be a direct consequence of the skills an agent possesses (Konidaris, Kaelbling, and Lozano-Perez 2018), but a reversed relationship where actions are "desired high-level observations" is also of significant interest (Bakker and Schmidhuber 2004). In an approach that leads with state-abstraction, it is key that the abstract MDP maintains the Markov property and does not become a partially-observable MDP (POMDP), or this may burden a downstream system with a memory requirement. Maintaining the Markov property can be accomplished by a principled approach to abstraction which focuses on preserving the property (Allen et al. 2021). This work in state abstraction provides key underpinnings for some methods I will use, but I am aware of no research which explicitly combines the abstraction process with the process of developing high-level actions.

Konidaris, Kaelbling, and Lozano-Perez (2018) use options to introduce the ability to plan within MDP environments (Konidaris, Kaelbling, and Lozano-Perez 2018). This research demonstrates the possibility of solving tasks in lower-level environments with symbolic planning, including environments with continuous features and stochastic transitions. Another key lesson from this work is that options must lead consistently to other options in order to be suitable for planning, and in general they should funnel an agent from many possible initiating states to a smaller set of final ones to satisfy this requirement. This work has been expanded with parameterized options that allow for greater flexibility in their usage (Ames, Thackston, and Konidaris 2018). Methods for ensuring the condition on options have been developed, including deliberately learning options that reach other options with skill chaining (Bagaria and Konidaris 2019), extending this to a full graph structure (Bagaria, Senthil, and Konidaris 2021), and using a method of optimistic and pessimistic classification to stabilize the initiation sets of options as they change (Bagaria et al. 2021). Another potentially key component of this goal is the rapid combining of existing options (Barreto et al. 2019). This body of work is closely related to the research proposed here, but focuses on modeling solutions to a given task rather than the domain as a whole.

Current Work

To date, the work I have done for this research has focused on defining the problem, establishing an experimental low-level domain with an explicitly defined target high-level domain, and investigating the relationship existing options research has with this work.

Formal Definition

To understand the required structure of the high-level abstraction, the problem can be defined as a graph transformation. Given a deterministic MDP (S, A, T, R) (where S is a set of states, A is a set of actions, $T(s, a, s')$ yields 1 if taking the action a in state s leads to s' and 0 otherwise, and $R(s)$ is the reward recieved for entering state s), a directed graph can be constructed such that $G = (V, E) = (S, \{(s_1, s_2) : s_1 \in S, s_2 \in S, \exists A \in a \quad s.t. \quad T(s_1, a, s_2) = 1\})$. The task is to find an *abstraction* $G_A = (V_A, E_A)$ of G such that V_A consists of strongly connected components (SCCs) in V, and the edges in E_A commute the reachability of components from V to the associated vertices in V_A.

V_A is 'nearly' a *partition* of V, but is not quite one—the abstraction may remove vertices in V that only occur on paths between the components that are represented in V_A. These removals occur because the edges in E_A represent paths in G, and the intermediate vertices are not reachable in V_A. This abstraction is related to, but distinct from, a *condensation* of the graph into maximally sized SCCs. A contribution of this work is to identify meaningful actions and state changes that may occur within a SCC, and as such it may be divided into smaller SCCs—the abstraction should be valuable even if every action in a problem is reversible.

In this formulation, G_A represents the abstract domain. The vertices in V_A are abstract states, and an agent can identify where it is in in this model by which partition its current state belongs to. The edges in E_A are high-level actions. An agent can execute them by following a path in E that an edge from E_A represents. It is assumed that navigation within an abstract state in V_A is essentially free.

This definition only describes the *structure* of the solution, and does not define what makes one abstraction better or worse than another. Investigating what traits may suggest a state should be considered meaningful to a problem, and what actions should be considered significant, is a major aspect of the research I am proposing.

Experimental Domain

Since there is no preexisting definition of what might make for a "good" abstraction of a domain, I establish an initial possibility explicitly. A low-level domain must exist for work done for this research, so I have selected a preexisting high-level domain to develop a low-level instance of. For this, I use an adaptation of the Grandma narrative planning domain (Ware et al. 2019). It is important that the target is a narrative planning domain rather than simply a solution to the domain, as a full domain is required to enable intentional behavior on the part of agents. My initial work intentionally avoids attempting to abstract a truly multi-agent system, and so key reasons for choosing such a domain are not immediately apparent, but it will be more valuable when this point of the research is reached.

For these first stages, the Grandma domain is rendered into a single-agent problem where Tom must navigate the environment to heal his Grandma with a potion. Tom can solve this problem by venturing to the Market to buy the potion, venturing to the Market to buy a sword and then rob the Merchant for the potion, venturing to the Bandit's Camp to acquire an additional coin with which to buy the potion, etc. The low-level adaptation of this problem breaks each action into multiple expected steps that will be required to execute them in full—for example, the attack action might be accomplished by holding a weapon, then striking multiple times. It also renders high-level features into conditions on one or more lower-level features, such as location being understood as an (x, y) position.

Initial Investigation

Using the test domain, deploying existing option discovery methods provides an impression of where these methods are with respect to the goal of the proposed research. A number of these prove to be unsuitable for the task set out here, either due to requirements on the input or the results that they produce. For example, the option discovery algorithm Q-Cut is designed to augment blind exploration, and produces results of limited use when too much information is available (Menache, Mannor, and Shimkin 2002). On the other end of difficulties with information, Bagaria and Konidaris's skill chaining method requires agents to reach a goal before developing any options, and so is not suitable for the large domains with sparse reward that we anticipate.

A more straightforward approach of graph condensation can quickly reduce a graph to strongly-connected components, which are important to the structure of our problem. However, as discussed in the Formal Definition section, these components may hide actions that are interesting for use in the high-level domain, despite being reversible. As such it may be a suitable starting point for some domains, but the resulting components then need to be segmented, and it is not immediately apparent how that should be done.

Future Work

The next step of this work is to develop a method of option discovery which has the necessary properties to support planning in a deterministic MDP environment. Then, a process can be developed which extracts the abstract state space created by these options and generates a planning domain. An alternative approach may identify an ideal abstract state space first and construct options to navigate it. With a system in place, it can be adapted to work within a multi-agent MDP. With either the single- or multi-agent systems in place, evaluations with human subjects may be valuable.

Option Discovery

Options for the purpose of planning must be arranged such that when an agent ceases executing an option they are always able to execute a new option (Konidaris, Kaelbling, and Lozano-Perez 2018), or are otherwise able to recover from a failed action or plan. Additional desirable properties are that options seem important to an agent's goals and are generally impactful but short: time-consuming options are more likely to fail in multi-agent simulations.

Constructive Approach By identifying novel facts that can be reached within the simulation, either by using any achievable fact or employing a definition of novelty such as width (Geffner and Geffner 2015), an expansive set of relatively low-level options (aiming only to achieve these facts) are be produced. Study of these options may reveal information about preconditions which are necessary *for* their successful execution, or side-effects which are unavoidable *if* they successfully execute. These facts can then indicate which options can be composed to form higher-level ones.

Decompositional Approach This approach begins with an initial, small set of options which target predetermined goal states. These options are studied for points where they might be broken into smaller ones, using concepts such as bottleneck states that successful trajectories frequently visit (McGovern and Barto 2001; Menache, Mannor, and Shimkin 2002), or landmark facts (or actions) that *must* be true (or executed) at some point during the option (Hoffmann, Porteous, and Sebastia 2004; Zhu and Givan 2003). These identify points at which the initial options can be segmented, producing component options that preserve the necessary condition of connectivity.

Extracting Abstract Representation

With suitable options in place, the abstract representation can be determined by identifying preconditions and effects of options, and composing features representative of these facts. Alternatively, state abstraction may be the first step, focused on identifying key groupings of facts in the domain. In this case, option discovery proceeds easily from the constructive approach, using the abstract features as the target effects.

Multi-agent MDPs (MMDPs)

A multi-agent environment presents difficulty both by increasing the size of the MDP's state space, as well as the uncertainty inherent in acting around other agents who may move or act in ways that disrupt the expected state of the simulation. Expanding the initial work to nondeterministic MDPs is likely to be key to solving this problem, as it provides a way to model that disruption.

Evaluation

Some results of this work may be evaluated against other methods for performance (the generated options may prove to be more effective than other option discovery methods would produce), but superior performance is not the goal of this system. Instead, the goal is to create high-level actions that appear to be a reasonable abstraction of activities that are significant in the domain. The success of this aspect of the work will be evaluated with human subject studies, using feedback on the suitability of the generated actions relative to the expectations of non-experts.

References

Allen, C.; Parikh, N.; Gottesman, O.; and Konidaris, G. 2021. Learning Markov State Abstractions for Deep Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34.

Ames, B.; Thackston, A.; and Konidaris, G. 2018. Learning Symbolic Representations for Planning with Parameterized skills. In *Proceedings of the 2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 526–533. IEEE.

Andersen, G.; and Konidaris, G. 2017. Active Exploration for Learning Symbolic Representations. *Advances in Neural Information Processing Systems*, 30.

Bagaria, A.; and Konidaris, G. 2019. Option Discovery Using Deep Skill Chaining. In *Proceedings of the Eight Int. Conf. on Learning Representations*.

Bagaria, A.; Senthil, J.; Slivinski, M.; and Konidaris, G. 2021. Robustly Learning Composable Options in Deep Reinforcement Learning. In *Proceedings of the 30th Int. Joint Conf. on Artificial Intelligence*.

Bagaria, A.; Senthil, J. K.; and Konidaris, G. 2021. Skill Discovery for Exploration and Planning Using Deep Skill Graphs. In *Proceedings of the 38th Int. Conf. on Machine Learning*, 521–531.

Bakker, B.; and Schmidhuber, J. 2004. Hierarchical Reinforcement Learning with Subpolicies Specializing for Learned Subgoals. In *Proceedings of the 2nd IASTED Int. Conf. on Neural Networks and Computational Intelligence*, 125–130.

Barreto, A.; Borsa, D.; Hou, S.; Comanici, G.; Aygün, E.; Hamel, P.; Toyama, D.; hunt, J.; Mourad, S.; Silver, D.; and Precup, D. 2019. The Option Keyboard: Combining Skills in Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Croonenborghs, T.; Driessens, K.; and Bruynooghe, M. 2007. Learning Relational Options for Inductive Transfer in Relational Reinforcement Learning. In *Proceedings of the 17th Int. Conf. on Inductive Logic Programming*, 88–97.

Davoodabadi, M.; and Beigy, H. 2011. A New Method for Discovering Subgoals and Constructing Options in Reinforcement Learning. In *Proceedings of the 5th Indian Internation Conf. on Artificial Intelligence*, 441–450.

Dietterich, T. G. 2000. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *Journal of Artificial Intelligence Research*, 13: 227–303.

Geffner, T.; and Geffner, H. 2015. Width-based Planning for General Video-game Playing. In *Proceedings of the 11th AAAI Int. Conf. on Artificial Intelligence and Interactive Digital Entertainment*, volume 11, 23–29.

Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered Landmarks in Planning. *Journal of Artificial Intelligence Research*, 22: 215–278.

James, S.; Rosman, B.; and Konidaris, G. 2018. Learning to Plan with Portable Symbols. In *ICML/IJCAI/AAMAS 2018 Workshop on Planning and Learning*.

James, S.; Rosman, B.; and Konidaris, G. 2020. Learning Portable Representations for High-level Planning. In *Proceedings of the 37th Int. Conf. on Machine Learning*, 4682–4691.

Jinnai, Y.; Abel, D.; Park, J. W.; Hershkowitz, D. E.; Littman, M. L.; and Konidaris, G. 2019. Skill Discovery with Well-Defined Objectives. In *Proceedings of the 7th Int. Conf. on Learning Representations Workshop on Structure and Priors in Reinforcement Learning*.

Konidaris, G.; Kaelbling, L. P.; and Lozano-Perez, T. 2018. From Skills to Symbols: Learning Symbolic Representations for Abstract High-Level Planning. *Journal of Artificial Intelligence Research*, 61: 215–289.

Konidaris, G. D.; and Barto, A. G. 2007. Building Portable Options: Skill Transfer in Reinforcement Learning. In *Proceedings of the 20th Int. Joint Conf. on Artificial Intelligence*, 895–900.

Kulkarni, T. D.; Narasimhan, K.; Saeedi, A.; and Tenenbaum, J. 2016. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. *Advances in Neural Information Processing Systems*, 29.

Machado, M. C.; Bellemare, M. G.; and Bowling, M. 2017. A Laplacian Framework for Option Discovery in Reinforcement Learning. In *Proceedings of the 34th Int. Conf. on Machine Learning*, 2295–2304. PMLR.

McGovern, A.; and Barto, A. G. 2001. Automatic Discovery of Subgoals in Reinforcement Learning Using Diverse Density. In *Proceedings of the 18th Int. Conf. on Machine Learning*, ICML '01, 361–368.

Menache, I.; Mannor, S.; and Shimkin, N. 2002. Q-Cut - Dynamic Discovery of Sub-Goals in Reinforcement Learning. In *Proceedings of the 13th European Conf. on Machine Learning*, ECML '02, 295–306.

Parr, R.; and Russell, S. 1997. Reinforcement Learning with Hierarchies of Machines. *Advances in Neural Information Processing Systems*, 10.

Ravindran, B.; and Barto, A. G. 2002. Model Minimization in Hierarchical Reinforcement Learning. In *Proceedings of the 5th Int. Symp. on Abstraction, Reformulation, and Approximation*, 196–211.

Stolle, M. 2004. *Automated Discovery of Options in Reinforcement Learning*. Master's thesis, McGill University.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, 112(1–2): 181–211.

Ware, S. G.; Garcia, E. T.; Shirvani, A.; and Farrell, R. 2019. Multi-agent Narrative Experience Management as Story Graph Pruning. In *Proceedings of the 15th AAAI Int. Conf. on Artificial Intelligence and Interactive Digital Entertainment*, 87–93.

Zhu, L.; and Givan, R. 2003. Landmark Extraction via Planning Graph Propagation. In *Proceedings of the 13th Int. Conf. on Automated Planning and Scheduling Doctoral Consortium*, 156–160. Citeseer.