The Workshops of the Thirty-Second
AAAI Conference on Artificial Intelligence

Concept-Aware Feature Extraction for Knowledge Transfer in Reinforcement Learning

John Winder, Marie desJardins
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250
jwinder1@umbc.edu, mariedj@umbc.edu

Abstract

We introduce a novel mechanism for knowledge transfer via concept formation to augment reinforcement learning agents operating in complex, uncertain domains. Based on their observations, agents form concepts and associate them with actions to generalize their decisions at higher levels of abstraction. Concepts serve as simple, portable, efficient packets of hierarchical information that can be learned in parallel. The use of conceptual knowledge simultaneously provides an interpretable, semantic explanation of an agent's decisions, making the techniques promising for human-interaction domains such as games, where human observers wish to inspect an agent's rationale. This technique extends previous work on probabilistic learning with Markov decision processes (MDPs) by introducing rich hierarchical feature structures that can be learned from experience, enabling more effective learning transfer to new, related tasks.

Introduction

Knowledge transfer in reinforcement learning aims to have agents record and persist skills associated with features in their environment to better solve new challenges. We describe our novel approach to transfer, *concept-aware feature extraction* (CAFE). CAFE creates multi-layered abstractions of a domain's state-action space, re-representing it in terms of derived features called concepts. An agent automatically extracts concepts from observations as high-level descriptors, constructs a hierarchy of these experiences, and records learned behaviors over this structure. Specifically, CAFE uses formal concept analysis to produce concept lattices that cluster extracted features in a partial ordering of increasing abstraction. Concept formation permits knowledge learned from one task to be applied to a new problem by identifying the appropriate level of generalization, transferring behaviors between related tasks in unseen environments. CAFE is especially useful for generalizing across domains with related objects, enabling problem solving using a scalable, incremental learning strategy, extensible with respect to totally novel concepts, which often capture semantically meaningful aspects of state space. We demonstrate preliminary results in two simulated environments, and sketch a plan for its use in a more complex domain.

Background

Reinforcement Learning (RL) is a paradigm for solving decision-making problems where an agent makes an observation, interacts with the environment, and receives feedback in the form of a reward or punishment. The Markov decision process (MDP) is a common formalization for such problems defined as a set of states, actions, transition probabilities, and rewards received upon making a transition from one state to another by some action. The object-oriented Markov decision process (OO-MDP) is a factored state space variant of the MDP where states are composed of objects whose behavior is defined by the instantiated values of their attributes (Diuk, Cohen, and Littman 2008). In any given OO-MDP state, each object has a specific assignment of values to each of its attributes. The main benefit of the OO-MDP formalization is that it allows more general, expressive, and easily extensible definitions for decision-making problems: it grants a natural way of describing environments as a collection of objects and their properties. As objects or attributes can easily be added or removed, OO-MDPs are ideal for state abstractions. OO-MDPs have served as the basis for recent research in abstractions in RL, including portable option discovery (Topin et al. 2015) and planning over hierarchies of abstract Markov decision processes (Gopalan et al. 2017).

A decision problem is typically solved by finding the optimal policy, a mapping from states to actions that specifies the best action to take for each state, taking the action that maximizes the expected discounted future rewards. RL algorithms find a policy by computing either the value function or the action-value function, which represents a state's utility, based on the expected value of discounted future rewards that would be received by taking the actions specified by the policy from any given state. Q-learning and SARSA(λ) are commonly used algorithms for computing the optimal action-value function, which induces an optimal policy. Since the value and action-value functions are specific to a given MDP, they are not directly transferable and do not generalize to new tasks in most cases.

RL has been employed notably in learning to play the games Backgammon (Tesauro 1995) and Go (Silver et al. 2016), and in achieving human-level performance on Atari video games (Mnih et al. 2015). In each of those cases, the technique of *value function approximation* (VFA) was key

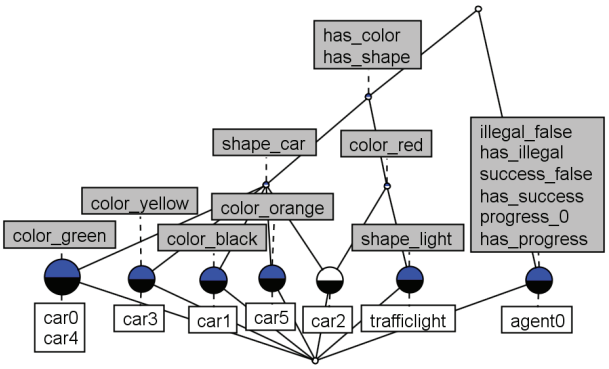

Figure 1: The concept lattice from an initial state of the TL domain. Each node is a concept, used as features in VFA.

to addressing the issue of generalizing experience over a large state space. When the number of states far exceeds what can feasibly be stored in memory, it is common to approximate the value or action-value function as a linear combination: $\hat{V}_\theta(s) = \theta^T\phi(s)$ and $\hat{Q}_\theta(s,a) = \theta^T\phi(s,a)$, for a vector of weight parameters θ and vector of basis functions ϕ (Geist and Pietquin 2013). The basis functions serve as the set of features representing a given state, so that rather than computing and storing the exact value of states, algorithms need only maintain and update the weight vector. Common forms of featurization for linear VFA include tile-coding (also known as cerebellar model articulator controller, CMAC) (Albus 1971), sparse distributed memories (Kanerva 1993), or Fourier basis functions (Konidaris, Osentoski, and Thomas 2011). Neural networks are also used for VFA in deep RL approaches (Mnih et al. 2015). In terms of transfer in RL, we follow recent theory in analyzing transfer across tasks, training on source and evaluating in related (but more complex) target tasks (Taylor and Stone 2009; 2011).

For knowledge extraction, formal concept analysis (FCA) is a technique based on order theory that has been growing in popularity in recent years (Poelmans et al. 2013). FCA provides a means of extracting knowledge structures from data in the form of formal concepts (paired sets of objects and attributes). Taken together, the concepts extracted from a set of data yield a partial ordering called a *concept lattice* that embodies a hierarchical relation of concepts from the most abstract to the most specific. FCA defines a data set as a *formal context*, which contains the set of objects, the set of attributes, and an incidence relation that expresses if an object possesses an attribute. Within a context, a *formal concept* is a pair of object and attribute sets $(\mathcal{A}, \mathcal{B})$ such that all objects in $\mathcal{A}$ have all attributes in $\mathcal{B}$, and all attributes in $\mathcal{B}$ are found in all objects of $\mathcal{A}$. A way of viewing formal concepts is as biclusters that are maximally inclusive on both the object and attribute sets (Veroneze, Banerjee, and Von Zuben 2017). Several algorithms exist for mining formal concepts from a context, such as FASTCLOSEBYONE and IN CLOSE2 (Andrews 2011).

All the concepts obtained from a context inherently yield a partial ordering (the concept lattice). In particular, concepts are ordered from the unit element (the top $\top$, a paired set of all objects and any attributes found in all objects) to the zero element (the bottom $\bot$, all attributes and any objects that possess all attributes). Hence, if concept $(\mathcal{A}, \mathcal{B}) \leq (\mathcal{C}, \mathcal{D})$, then $(\mathcal{A}, \mathcal{B})$ is a more specific sub-concept of the more general super-concept $(\mathcal{C}, \mathcal{D})$. A lattice can be visualized graphically where each node is a formal concept and the arcs express the natural sub- and super-concept relationships, such as the ones shown in Figures 1 and 2. Attribute labels (in gray) are attached to the highest concept for which their respective attribute is a member, and object labels (white) to the lowest. FCA is most commonly used for mining static data sets such as text corpora for semantic relations. This work introduces a novel approach that employs FCA interactively in an agent-based decision-making context. The motivation for conveying agent knowledge through formal concepts is that they are descriptive yet small and hierarchical, arising simply from a data set itself. Moreover, a lattice provides a type of natural unsupervised clustering of objects in an agent's world, forming a kind of ontology, where its concepts are informative groupings of perceptions and components of the world that can be used to reason and learn.

Approach

We introduce concept-aware feature extraction (CAFE) as a technique that encompasses both abstraction and featurization of state space. The first step in CAFE is the process of mapping a symbolic representation of an agent's current state to a set of concepts. For an OO-MDP state, the set of objects serves as a formal context. At each state we can extract a concept lattice from its context using an algorithm such as IN-CLOSE2 (Andrews 2011). CAFE thus re-represents states in terms of the concepts that were extracted from them. In particular, it applies state abstraction by projecting the ground state into concept space, spanned by the basis functions corresponding to the concepts of the extracted lattice. We define a unique abstraction function ψ that produces *concept states*. For each concept c_i formed at time t, $\psi(s_t, c_i) \rightarrow z_i$ by removing all objects and attributes from the ground state s_t that are not present in a concept c_i, yielding a concept state $z_i \in \mathcal{Z}_t$. For example, suppose a state contains one red chair, two blue chairs, and one red backpack. If this state is abstracted by the concept "red," the resulting concept state would consist of two red objects (subtracting all shapes, other colors, and objects not matching any attribute in the concept). Similarly, if that same state is abstracted by the concept "chair," it would produce a concept state of solely three chair objects. In some sense, z_i is a sub-state of the OO-MDP, describing how the ground state appears when filtering through the lens of its particular concept c_i. The featurization of a state using concept-based abstraction would produce a feature vector $\phi(s_t, \mathcal{Z}_t)$ such that each element follows: $\phi_i(s_t, \mathcal{Z}_t) = 1$ if concept state $\psi(s_t, c_i) \in \mathcal{Z}_t$ (0, otherwise), indicating the presence or absence of the concept in the ground state. A *concept-aware* agent can then use ϕ directly to learn the parameter weight vector θ and approximate the value or action-value

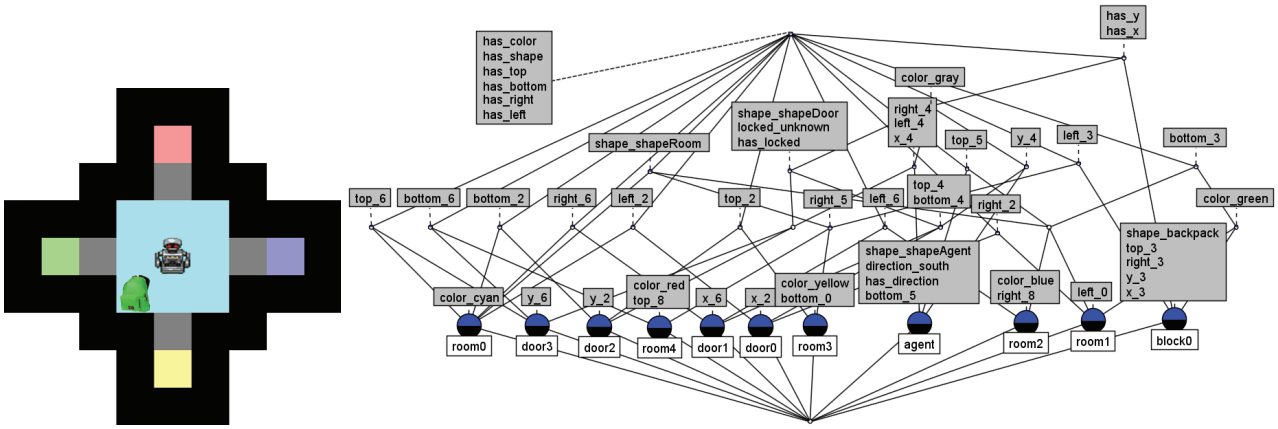

Figure 2: An example starting Cleanup state for the Cardinal Closets task (CC-1), with the concept lattice derived from it.

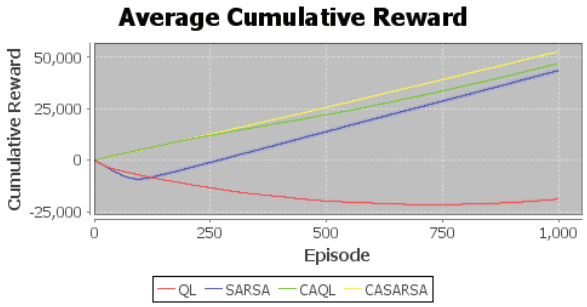

Figure 3: Evaluation on the TL-20 task with 12 cars, transferring from source task of TL-10 with 6 cars.

function ($\hat{V}_\theta(s_t) = \theta^T \phi(s_t)$ and $\hat{Q}_\theta(s_t, a) = \theta^T \phi(s_t, a)$). Thus, CAFE projects an observed state into a concept lattice and uses the resulting abstract concept states for linear VFA, with $\phi(s_t)$ expressing the concept states present in state s_t. This FCA-derived featurization is advantageous because it gives an automated process of finding portable, relevant groupings of objects and their attributes (concepts). The nature of VFA inherently facilitates generalization: it forces an agent to learn behaviors by features that may be seen in new combinations in states for novel tasks or domains. For VFA with CAFE specifically, using the extracted concepts grants an added benefit: extensibility. That is, the concept space can expand gracefully and indefinitely to accommodate any new concepts that may be generated, while still making full use of previously learned concepts as the reoccur. It also permits clustering and measuring various types of similarity, leading to greater interpretability of agents' decisions.

Preliminary Results

Preliminary results indicate that CAFE is effective at task-based transfer learning. We evaluate concept-aware algorithms in a simple domain, Traffic Light (TL), and the more challenging Cleanup. In TL, an agent must go when a traffic light turns green, and otherwise wait while the light is red for it to change color, all while ignoring the surrounding objects (cars of various random colors). TL is a simple decision problem that can be solved by learning to associate the actions "go" with the concept of a green traffic light and "wait" with a red light, while ignoring confounding concepts such as green or red objects generally, and cars of any specific color. TL tasks are differentiated as TL-n, where n is their horizon (the number of times the agent must correctly choose to go forward). The optimal policy is always to wait once for the traffic light to turn green and then go n times until the progress threshold is reached. The larger Cleanup domain presents an agent with a simplified model of a home or office that must be cleaned up by putting objects in their proper place. Cleanup in essence is an extended "grid-world" where an agent may move in the cardinal direction or "pull" a block (swapping places with it), and it was designed to emulate the game Sokoban (MacGlashan et al. 2015). In particular, we look at the Cardinal Closets (CC) task, where the agent is in a central room with four branching closets and a block of a random color, with the goal of maneuvering the block into the closet of the matching color. CC is challenging because the optimal policy for agent navigation changes completely based on the color of the goal block (i.e., if the block is red then the agent must manipulate the block towards the north, or south if the block is yellow), in essence folding four tasks into one. We consider transfer from one block domains (CC-1) to 2 blocks (CC-2). The size of these domains is considerably different, as each added block grows the state space combinatorially . Therefore, knowledge transfer of concepts should be especially beneficial in identifying features that are valuable from one cleanup task to the next.

To assess the concept-aware algorithms, we follow a training-testing pattern for evaluating transfer, with an agent first learning to solve a source task on a set of training domains (TL-10, CC-1). The agent is then evaluated on another set of domains to solve a target task (TL-20, CC-2), transferring over the set of known concept states and their learned weights, θ. We evaluate Q-learning (QL) and SARSA(λ) as baselines (no transfer from source to target),

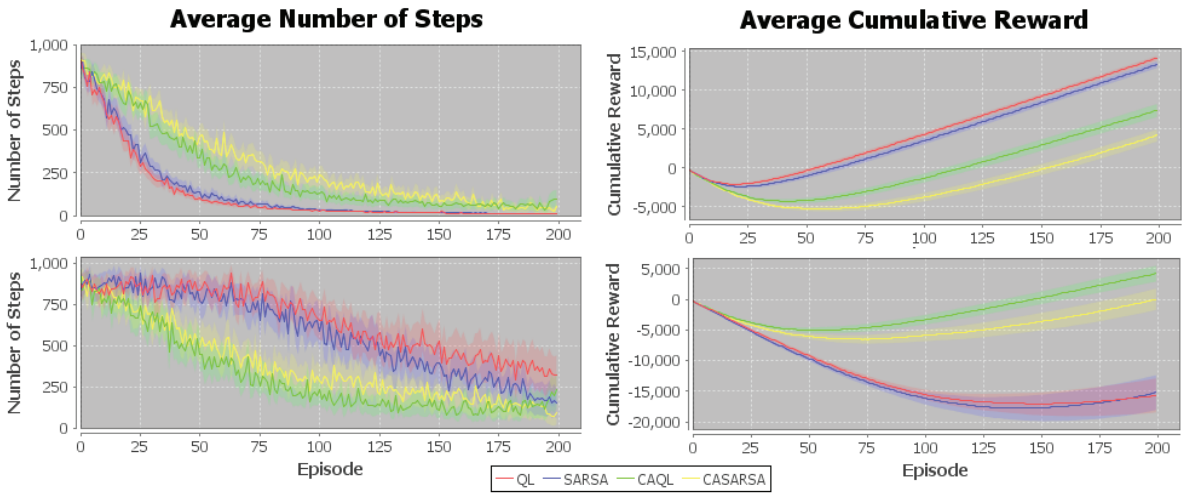

Figure 4: Training on CC-1 over 100 trials (top row) and evaluation on CC-2 for 50 trials (bottom row). CAQL and CASARSA need more upfront training to tie concepts with actions, but they can reuse learned behaviors to solve the harder task.

with concept-aware Q-learning (CAQL) and concept-aware SARSA(λ) (CASARSA). The latter two transfer experience in the form of the concept feature weight parameters (θ). All algorithms follow an ϵ-greedy policy with a decay schedule from $\epsilon = 1.0$ proportional to episode number (i.e., $\epsilon = 0.0$ at episode 200), to enforce a degree of initial exploration.

The results of transfer are shown in Figure 3, with evaluation on TL-20 with 12 cars for 50 trials, after training on TL-10 with 6 cars for 100 trials. Both CAQL and CASARSA rapidly find the optimal policy in the target task. After just a few episodes of training they discover concepts associated with good and bad behaviors. The feature-action pair with the largest weight is for the action "go" and the concept state corresponding to [shape:light, color:green]. Symmetrically, the pair with the smallest weight is [shape:light, color:red] and "go" (as this is only observed when the task terminates in a failure and receives negative reward).

Results of training on CC-1 and assessing transfer to CC-2 are visualized in Figure 4. Initially in training on the CC-1 domain, the concept-aware algorithms are worse in performance, taking approximately twice as many steps and episodes to reach a good policy. They require more exploration before homing in on a solution, and they suffer from larger variance in the number of steps for later episodes. Yet they are able to persist and transfer the knowledge they acquire through this extended training. In evaluation it is evident that CAQL and CASARSA more quickly adapt to the greatly expanded state space of CC-2, reusing the transferred concepts related to maneuvering the blocks to the doors immediately adjacent to the goal closets. Specifically, the highest valued feature-action weights are for concepts associated with a block being in the door adjacent to the block's respective goal room, paired with the navigational action that transitions to a goal state (e.g., "north" if a red block is inside the door to the red closet). Other highly valued weights include those for concept-action pairings that align the block with the door and goal room, or otherwise manipulate it out of a corner. Each state in CC-2 has on average 40 concept states, but in total, the concept-aware algorithms find 466 distinct concept states, and thus 2330 feature-action pairs (with 5 actions, the final $|\theta| = 2330$) across all states from all domains. The space of concept-actions, therefore, is considerably smaller than the state-action space of CC-2, explaining why VFA with CAFE grants both the benefits of jump start transfer and increased asymptotic performance.

Conclusion and Future Work

Concept-aware agents can transfer knowledge across tasks, reusing policies and extrapolating new behaviors upon recognizing familiar concepts, while re-representing anomalous objects in novel tasks using super- and sub-concepts that are already known. CAFE with VFA achieves a condensed featurization of state space at multiple levels of abstraction, even when objects' attributes can take any number of categorical values (such as color and shape). Concepts capture semantically meaningful clusters, interpretable by the set of objects and attributes present in them. Preliminary results indicate the promise of CAFE to achieve more explicable knowledge extraction and transfer in RL, and we intend to produce a more thorough investigation that assesses additional measures of transfer in comparison with more sophisticated VFA techniques.

We plan to assess CAFE-based task transfer more extensively in an immensely rich domain: NetHack. NetHack is a single-player video game in which the player must collect items and defeat enemies to escape from a multi-level dungeon. The vast variety of object types and possible actions makes transferring knowledge and adapting to novelty necessary. An agent often encounters new objects functionally similar to learned classes of objects (but differing by some specific attributes). Transferring concepts should facilitate an agent's ability to respond to unseen types of items, fur-

niture, and enemies. We expect concept-aware agents will more readily comprehend these anomalies, in terms of concepts already observed, and transfer behavior accordingly.

References

Albus, J. S. 1971. A theory of cerebellar function. *Mathematical Biosciences* 10(1-2):25–61.

Andrews, S. 2011. IN-CLOSE2, a high performance formal concept miner. In *International Conference on Conceptual Structures*, 50–62. Springer.

Diuk, C.; Cohen, A.; and Littman, M. L. 2008. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, 240–247. ACM.

Geist, M., and Pietquin, O. 2013. Algorithmic survey of parametric value function approximation. *IEEE Transactions on Neural Networks and Learning Systems* 24(6):845–867.

Gopalan, N.; desJardins, M.; Littman, M. L.; MacGlashan, J.; Squire, S.; Tellex, S.; Winder, J.; and Wong, L. L. S. 2017. Planning with abstract Markov decision processes. In *27th International Conference on Automated Planning and Scheduling*.

Kanerva, P. 1993. Sparse distributed memory and related models. In *Associative neural memories*, 50–76. Oxford University Press, Inc.

Konidaris, G.; Osentoski, S.; and Thomas, P. 2011. Value function approximation in reinforcement learning using the Fourier basis. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 380–385. AAAI Press.

MacGlashan, J.; Babeş-Vroman, M.; desJardins, M.; Littman, M. L.; Muresan, S.; Squire, S.; Tellex, S.; Arumugam, D.; and Yang, L. 2015. Grounding English commands to reward functions. In *Robotics: Science and Systems*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Poelmans, J.; Kuznetsov, S. O.; Ignatov, D. I.; and Dedene, G. 2013. Formal concept analysis in knowledge processing: A survey on models and techniques. *Expert systems with applications* 40(16):6601–6623.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Taylor, M. E., and Stone, P. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10(Jul):1633–1685.

Taylor, M. E., and Stone, P. 2011. An introduction to intertask transfer for reinforcement learning. *Ai Magazine* 32(1):15.

Tesauro, G. 1995. Temporal difference learning and td-gammon.

Topin, N.; Haltmeyer, N.; Squire, S.; Winder, J.; desJardins, M.; and MacGlashan, J. 2015. Portable option discovery for automated learning transfer in object-oriented Markov decision processes. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 3856–3864. AAAI Press.

Veroneze, R.; Banerjee, A.; and Von Zuben, F. J. 2017. Enumerating all maximal biclusters in real-valued datasets. *Information Sciences* 379:288–309.