Hierarchical Representations of Behavior for Efficient Creative Search

Christopher M. Vigorito and **Andrew G. Barto**
Computer Science Department
University of Massachusetts Amherst
Amherst, Massachusetts 01003
{vigorito,barto}@cs.umass.edu

Abstract

We present a computational framework in which to explore the generation of creative behavior in artificial systems. In particular, we adopt an evolutionary perspective of human creative processes and outline the essential components of a creative system this view entails. These components are implemented in a hierarchical reinforcement learning framework and the creative potential of the system is demonstrated in a simple artificial domain. The results presented here lend support to our conviction that creative thought and behavior are generated through the interaction of a sufficiently sophisticated variation mechanism and a comparably sophisticated selection mechanism.

Introduction

Some have argued that creative thought and behavior are the products of an ongoing process of blind variation and selection, analogous to Darwinian natural selection, of both symbolic cognitive structures and overt action sequences (Campbell 1960; Simonton 1999). From this perspective creative products, whether scientific theories or musical symphonies, are produced by way of sequential manipulations of intermediate constructs in a trial-and-error fashion. The trials comprising these trajectories are chosen blindly (i.e., irrespective of any specific goal) and either executed explicitly in the world or simulated (e.g., in the creator's mind). Any resulting structures, whether physical or symbolic, are then evaluated according to some consistent metric and selectively retained for manipulation in subsequent trials.

Implicit in this approach to modeling creativity is the existence of a search space in which creative products and their intermediate configurations are embedded. We maintain that creative products then are the result of a sequential decision process consisting of trajectories through this space. It has been argued by many that the size of this search space in most realistic domains is astronomical and precludes any hope for success of blind trial-and-error processes in producing complex creative works. We aim to show, however, that just as this argument does not hold with respect to natural selection in biology, so is it untenable in our view of creativity.

Following the analogy of natural selection, one would be hard pressed to argue that something as complex as a compound eye, digestive system, or mammalian brain could have evolved from the precursors of single cells even by natural selection if the only variational mechanism available was point mutation of DNA. In such a scheme, where recombination of existing genetic substructures is conspicuously absent, the chances of stumbling upon adaptive genetic modules through blind trial and error is, as noted by the critics cited above, astronomically small. This is because point mutations produce variation only at the most primitive level, essentially limiting the steps taken in the search process to the smallest ones possible. Blind trial-and-error search in this scenario is correctly judged as doomed to failure, or at least severely crippled.

However, once recombination of previously discovered components is introduced (through sexual reproduction and crossing over), suddenly the evolution of complex artifacts like the vertebrate nervous system becomes possible, indeed inevitable. The difference results from the accumulation and combinatorial manipulation of useful substructures that effectively reduce the search space by affording larger, more meaningful steps through it. Because these substructures can be manipulated hierarchically, the variation between trials can occur at many scales, and thus over time appropriately sized steps through the search space can be made with ease toward structures of increasing fitness, even if the steps are chosen blindly.

In what follows we argue that just as complex biological systems are evolved by this mechanism in nature, so are complex creative products evolved by a similar process in human creative thought and action. The two processes are of course not identical. One of the key differences is that in human creative processes there exists no explicit population of trials undergoing simultaneous evaluation as there does in natural selection, but rather a set of potential trials, only some of which are realized and evaluated based on the current context at each step of the process. Gabora (2005) calls this "context-driven actualization of potential", and it is consistent with our view of creativity as a sequential decision process.

From a computational point of view, this distinction corresponds to the differences between techniques like genetic algorithms (Holland 1975) and genetic programming (Koza 1992) that maintain explicit populations for simultaneous evaluation, and others such as reinforcement learning (Sutton & Barto 1998) that maintain these populations only implicitly and evaluate them sequentially. Given the characteristics of creative processes mentioned above, we consider reinforcement learning a more appropriate framework for modeling certain aspects of creativity than the more genetically inspired methods cited, and so we choose this framework to outline a formal basis for artificial creative systems.

Other subtle distinctions abound, but it is sufficient to note here that creativity is a multi-faceted and complex cognitive process that resists strict reductionist approaches, and as such the work presented here reflects only what we believe to be necessary but likely insufficient criteria for an intelligent creative system. In particular, we believe that creative behavior can be generated in an artificial system through the interaction of a sufficiently sophisticated variation mechanism and a comparably sophisticated selection mechanism. In the following section we elaborate on the details of these criteria and how they facilitate efficient creative search. We then present a computational reinforcement learning framework that realizes these criteria and demonstrate its potential for creativity in a simple artificial domain.

Criteria for Efficient Creative Search

As mentioned above, the search space implicit in the view of creativity espoused here is the space of potential creative products within a given domain, and the creative process a sequential decision process defined over it. In general, the features of this space and the operators for moving through it can be defined at many different levels of abstraction, with some definitions being more conducive to trial-and-error search than others. Without prior domain knowledge or environmental experience, however, creative search must necessarily begin with decision-making at the most primitive level of abstraction afforded by the innate behavioral and perceptual repertoire of a given creative agent. In this case each decision corresponds to a single primitive behavior, and as such variation at this level is generally impractical for efficient creative search in complex domains given the miniscule steps through the enormous search space each decision engenders.

It is for this reason that a creative agent must possess the ability to expand its skill set to include new behaviors at increasing levels of abstraction. These behaviors can be thought of as procedures for reliably fixing one or more features of a creative product to a certain set of values. During the decision-making process they can be treated as single decisions with predictable outcomes, even though they may entail the execution of long sequences of more primitive behaviors. This has the effect of redefining the search space from the perspective of the agent, though the underlying intrinsic space remains unchanged. By allowing for larger steps through the space, the space effectively shrinks from the agent's point of view, and blind trial-and-error search becomes more efficient and meaningful. Another consequence is the increased probability of reaching areas of the space previously unreachable through blind variation, and in turn potential discovery of new features and thus new skills to manipulate them.

As an illustrative example, consider a composer in the process of writing a symphony. If his only source of variation is to add or delete a single note of specific pitch and duration to the score at each decision point, then clearly producing any finished product that even sounds decent, let alone a unique and well-recieved work, will take an unacceptable amount of time. However, if he adds to his variational repertoire chords and phrases known to work well in many contexts based on prior experience, then recombination of these substructures affords steps towards a complex, coherent work of art not attainable otherwise. Good composers can easily manipulate and generate high-level variations of musical substructures like chords, phrases and themes, skills which novice composers have not yet mastered.

Even with a sufficiently deep hierarchy of skills for maneuvering through the large spaces characteristic of real-world domains, execution of the numerous trials necessary for the discovery of a highly-valued creative product is generally infeasible since, unlike in natural selection, these trials must be executed sequentially. The temporal and energetic costs associated with performing trials necessitate a surrogate for the true selection metric that can evaluate proposed decisions along a search trajectory in the absence of their explicit execution. This surrogate of course must be learned over time from explicit evaluations by the real metric, but in general the number of these evaluations will be small and thus the surrogate must generalize well from sparse feedback signals to accurately predict the effect of individual decisions on the value of the final product. During a search trajectory, each proposed variation must be evaluated by the surrogate and only the most highly valued chosen for consideration at the next decision point or for explicit execution in the environment.

To give another example, the production of a full-length motion picture is a very costly and time-consuming process with evaluative feedback occurring only at the end of the decision process, coming in the form of critical reviews, box-office success, etc. Successful movie producers/directors must be able to predict accurately the success or failure of individual decisions made during the production process, since separate trials representing variations at each of these decision points cannot be executed to completion. These predictions are of course based on prior experience and evaluations of previously completed projects, but on a relatively tiny number of them when compared to the number of decisions made. Good producers/directors have very accurate surrogates for the evaluation metrics by which their work is judged.

The criteria discussed above are essential to efficient creative search in any reasonably complex domain. Without the necessary hierarchy of skills, variation at non-primitive levels of abstraction is impossible and blind variation infeasible as a generator of potentially useful alternatives during the creative process. In the overwhelming majority of domains in which trials have high temporal and energetic costs, the absence of an accurate surrogate for the true evaluation metric also precludes efficient creative search, since it is impossible to explicitly evaluate every trial proposed during the creative process in a reasonable amount of time. The following section discusses some potential machine learning techniques for satisfying these criteria in an artificial system.

A Formal Basis for Efficient Creative Search

When considering computational frameworks that satisfy the aforementioned criteria, evolutionary methods based directly on Darwinian evolution such as genetic algorithms (Holland 1975) or genetic programming (Koza 1992) naturally come to mind. However, as mentioned earlier, creative processes in humans differ from biological evolution in that they do not maintain an explicit population of simultaneously actualized alternatives to select among. Rather, creative products evolve through an iterative, context-sensitive decision process. For this reason, we choose computational reinforcement learning (Sutton & Barto 1998) as our formalism, which is a context-sensistive behavioral variation-and-selection paradigm for optimal decision-making with a well-developed mathematical foundation. We outline the relevant details of a possible reinforcement learning framework for efficient creative search in the following section.

Reinforcement Learning

Reinforcement learning (RL) (Sutton & Barto 1998) is a computational paradigm for learning optimal goal-directed behavior in sequential decision problems. In this framework, an agent takes actions in an environment from which it receives sensory signals as well as a scalar-valued reinforcement signal, the long-term sum of which it tries to maximize over time. The agent does this through an iterative process of exploration and modification of its behavior. In this sense, RL is at its heart a variation-and-selection process, generating variations in behavior that are selected for or against based on the accumulation of rewards they effect over time.

To maximize its cumulative reward, an RL agent tries to learn a policy which maps states—specific representations of sensory signals—to actions that maximize long-term reward. This policy can be learned in a variety of ways, one of the most common being indirectly through the approximation of a value function which maps states to real values signifying the expected sum of future rewards an agent will receive given its current state. A policy is then derived from the value function by selecting actions greedily with respect to the function's expected value over successor states.

It should be clear that such a system which selects among a set of potential actions according to the evaluations given by its value function is a model instantiation of the type of system described in the previous section. What is essential to note is that the behavioral variation mechanism, often called the "actor", can be hierarchical in nature, selecting among not only primitive but also temporally abstract actions. In addition, the value function, or "critic", is a prime example of a sophisticated surrogate for the true selection metric in that it can provide accurate evaluations of proposed behaviors without the need for their explicit execution. The following sections outline the relevant mathematical formalisms commonly used in the RL literature, which often assume that a given environment can be modeled as a Markov decision process.

Markov Decision Processes

A finite Markov decision process (MDP) is a tuple $\langle S, A, P, R \rangle$ in which S is a set of states, A is a set of actions, P is a one-step transition model that specifies the distribution over successor states given a current state and action, and R is a one-step expected reward model that determines the real-valued reward an agent receives for taking a given action in a given state. An MDP is assumed to satisfy the Markov property, which guarantees that the one-step models R and P are sufficient for defining the reward and transition dynamics of the environment.

When the environment of an RL agent is formulated as a finite MDP, the task of the agent is to learn a policy $\pi : S \to A$, which maps states to actions that maximize its expected sum of future rewards, also called expected return. It is often assumed that the transition and reward models are unavailable to the agent. When this is the case, a policy can be learned through estimation of an action-value function $Q^\pi : S \times A \to \Re$, which maps state-action pairs $(s, a) \in S \times A$ to real values representing the expected return for executing action a in state s and from then on following policy π. If $Q^\pi = Q^*$, where Q^* represents the optimal value function for the MDP, then the agent can act optimally by greedily selecting actions in each state that maximize Q^π. The Q-learning algorithm (Watkins 1989) is one method for estimating this function online from experience.

When the transition dynamics of the environment are known or estimated from experience, model-based reinforcement learning (Sutton 1991) can be employed to expedite value function learning in the sense of requiring less experience for Q^π to converge to Q^*. Model-based RL methods constitute one way of adding more sophistication to the agent's selection mechanism, as they permit the surrogate for the true selection metric to be more accurate given relatively little data. This is because the transition model is used to simulate actual experience offline without the agent having to explicitly execute every action, thereby increasing the accuracy of the value function through hypothetical trials.

Given the criteria laid out in the previous section, one should prefer a selection mechanism to be as sophisticated as possible, and so model-based approaches seem the most promising for artificial creative systems. We subsequently describe model-based methods which assume that the transition and reward models of the environment are available to the agent, but other techniques exist which relax this assumption (Sutton 1991; Degris, Sigaud, & Wuillemin 2006). Indeed, we are currently engaged in work experimenting with motivational systems for active learning of these models online from minimal experience.

Even when model-based methods are used to improve data efficiency for value function learning, tabular representations of value functions and policies (i.e., those with one entry per state or state-action pair) become infeasible to learn efficiently in large MDPs. For this reason much work has focused on approximation techniques that allow for both generalization of value between similar states and compact representations of value functions (Sutton & Barto 1998). One class of these methods is appropriate when the MDP is highly structured and can be represented in factored form,

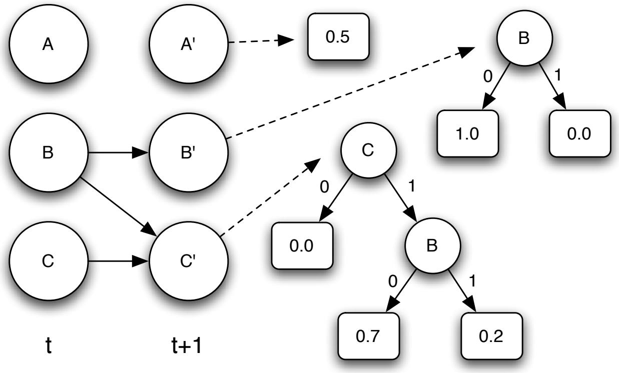

Figure 1: A simple DBN for a given action with corresponding conditional probability trees.

affording the potential for certain dimensions of the MDP to be irrelevant to predicting the effects of actions on other dimensions. In these cases, this structure can be exploited to learn compact representations of value functions and policies efficiently (Boutilier, Dearden, & Goldszmidt 1995; 2000). Many domains in which creative behavior is possible are highly structured, and we focus on methods from this approach to function approximation.

Factored MDPs

A factored MDP (FMDP) is an MDP in which the state space is defined as the Cartesian product of the domains of a finite set of random variables $\{S_1, \ldots, S_n\} = \mathbf{S}$. While the variables in an FMDP can be either discrete or continuous, we restrict our attention to the discrete case so that each $S_i \in \mathbf{S}$ can take on one of finitely many values in $\mathcal{D}(S_i)$, the domain of S_i. States in factored MDPs are thus represented as vectors of assignments of specific values to the variables in $\mathbf{S}$. As the number of variables in an FMDP increases linearly, the number of states increases exponentially. This has been referred to as the curse of dimensionality (Bellman 1957). However, if the FMDP contains relatively sparse inter-variable dependencies, we can exploit this structure to reduce the effect this exponential growth has on computing optimal policies.

FMDPs are often represented as a set of Dynamic Bayesian Networks (DBN) (Dean & Kanazawa 1989), one for each action. A DBN is a two-layer directed acyclic graph with nodes in layers one and two representing variables of the FMDP at times t and $t + 1$, respectively (see Figure 1). Edges represent dependencies between variables given an action. We make the common assumption that there are no synchronic arcs in the DBN, meaning that variables within the same layer do not influence each other. The transition model for a given DBN can often be represented compactly as a set of decision trees, one for each variable S_i, each of which contains internal nodes corresponding to the parents of S_i and leaves containing a probability distribution over $\mathcal{D}(S_i)$ at time $t+1$. Figure 1 shows a simple DBN (for some action a) consisting of three binary variables and their corresponding decision trees, with the probability that $S_i = 1$ displayed at the leaves.

When the transition model of an FMDP is known, there are algorithms for efficiently computing compact value functions and policies that exploit domain structure (Boutilier, Dearden, & Goldszmidt 2000). While it is possible to use these methods to design a sophisticated selection mechanism, the requirement of a sophisticated variation mechanism is not satisfied by these methods alone. Fulfilling this criterion requires a hierarchical action representation for these methods to exploit. Fortunately there exist formal techniques for both learning and planning with temporally abstract actions in reinforcement learning, which we outline next.

Hierarchical Reinforcement Learning

The options framework (Sutton, Precup, & Singh 1999) is a formalism for temporal abstraction in RL that details how to learn closed-loop control policies for temporally extended actions in MDPs. An option is defined as a tuple $\langle I, \pi, \beta \rangle$, where $I \subseteq S$ is a set of states over which the option is defined (the initiation set), π is the policy of the option, defined over I, and β is a termination condition that gives the probability of the option terminating in a given state. Options can also be understood as sub-MDPs embedded within a (possibly) larger MDP, and so all of the machinery associated with learning in MDPs applies to learning options as well, with some subtle differences. Since options can call other options in their policies, this framework allows for construction of the complex hierarchies of behavior essential to our view of creativity.

Because options are essentially MDPs in themselves, models for the transition and reward dynamics of an option can be learned as well. Algorithms for learning the policy, reward model, and transition model of an option from experience are given in Sutton, Precup, and Singh (1999). The advantage of having access to the transition and reward models of an option is that the option can be treated as an atomic action in planning or model-based RL methods. In this way, the outcomes of complex trials proposed by an actor with a repertoire of options can be easily predicted and the resulting state evaluated by a critic in a model-based RL system. The properties of such a system afford the essential conditions for creative behavior we have outlined. The following section presents techniques for generating compact option policies and models in large structured domains modeled as FMDPs.

Hierarchical Decomposition of Factored MDPs

Jonsson and Barto (2006) present a framework for option discovery and learning in FMDPs. The VISA algorithm discovers options by analyzing the causal graph of a domain, which is constructed from the dependencies exhibited in the DBNs that define the FMDP. It then decomposes the FMDP into sub-tasks solved by these options. The algorithm identifies in the causal graph action-context pairs, called exits, that cause one or more variables to change value when the given action is executed in the corresponding context. By searching through the conditional probability distributions that define the DBN, exit options are then constructed to reliably reach this context and execute the appropriate action. VISA takes advantage of structure in the domain to efficiently learn compact policies for options by ignoring irrelevant variables. The framework also provides a method for computing compact option models from a given DBN model. This allows the use of options in planning as single atomic units as mentioned above.

Sophisticated Variation and Selection in Structured Environments

Given the techniques outlined in the previous sections, one can begin to see how sophisticated variation in an artificial creative agent might be implemented. Assuming that a given creative domain is highly structured, as are many real-world environments, the VISA algorithm coupled with compact option models endows a reinforcement learning agent with a hierarchical behavioral repertoire that can easily be used to propose complex variations of existing creative structures. By possessing skills to reliably alter any property of a creative product, the agent can take steps through the creative search space at varying levels of abstraction, essentially transforming the search space to an appropriate size for blind variation to be effective.

For sophisticated selection, an accurate surrogate for the true selection metric as discussed above must be present. The value functions of the RL framework provide a mathematically sound formalization of this type of surrogate. These critics allow for the evaluation of a large set of potential trials proposed by a variation mechanism without the need for their explicit execution. Of course, value functions must be learned from data, but the model-based RL methods and compact option model computation techniques we have cited can help make that learning process very data efficient.

One important question remains, however. What are the true selection metrics that value functions might be used to predict? Obviously these metrics will be domain dependent, and it is primarily for this reason that we do not emphasize the selection component in our results. Criteria for judging creative works vary immensely between disciplines, with some drawing more from cultural norms (e.g., sculpture and music) and others being founded upon more objective evaluations (e.g., science and engineering). Whatever their source, extrinsic ratings will in general be difficult or expensive to come by. It is for this reason that value functions are invaluable to an RL system for creative search as outlined here.

While there are certainly many ways in which to implement a sophisticated variation-and-selection framework to satisfy the criteria set forth earlier, the sound mathematical foundations and well-tested performance of the RL techniques we have outlined make such a framework a promising candidate for the development of intelligent creative systems. There are indeed many ways to further increase the sophistication of some of the components we have mentioned and many avenues of research for relaxing some of the assumptions made in the above formalisms. We are currently exploring some of these options ourselves and outline a few of these directions in the discussion section. First we present some preliminary results exhibiting the advantages of hierarchical representations of behavior for sophisticated variation in a simple artificial domain.

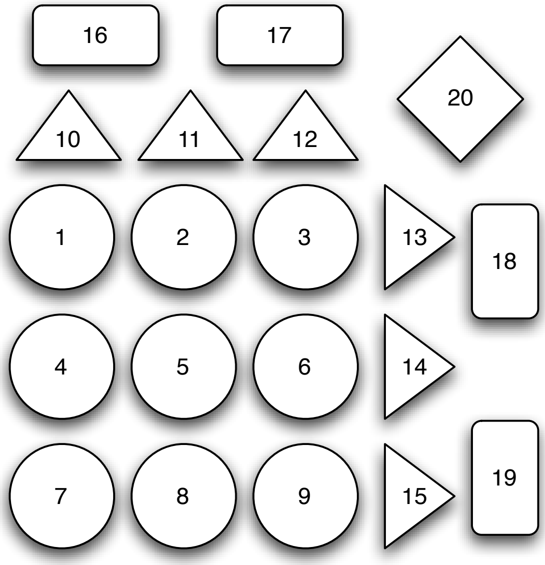

Figure 2: A visual rendering of the Light Box domain.

Experiments

The Light Box Domain

To illustrate the potential for creative behavior in a system with the components described above we ran preliminary experiments in a simple artificial environment called the Light Box (Figure 2). The domain consists of a set of twenty "lights", each of which is a binary variable with a corresponding action that toggles the light on or off. Thus there are twenty actions and $2^{20} \approx 1$ million states. The nine circular lights are simple toggle lights that can be turned on or off by executing their corresponding action. The triangular lights are toggled similarly, but only if certain configurations of circular lights are active, with each triangular light having a different set of dependencies. Similarly, the rectangular lights depend on certain configurations of triangular lights being active, and the diamond-shaped light depends on configurations of the rectangular lights. In this sense, there is a strict hierarchy of dependencies in the structure of this domain. The domain is also stochastic in that any primitive action fails with probability 0.1.

Uncovering the hierarchical structure of the domain and learning options to toggle each light are essential to producing sophisticated variation of behavior in this environment. It should be noted, however, that the agent does not perceive any structure directly as may be evident in the visual rendering of the domain. Rather the agent perceives only a string of twenty bits at any given time. The structure must be discovered from the transition model of the domain. Exploita-

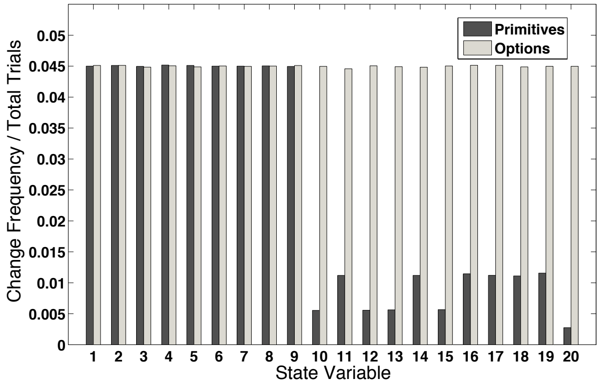

Figure 3: Ratios of changes in state variable values in the Light Box domain to total number of trials (uniformly random action selections) for an agent with primitive actions only and for one with options.

tion of this structure is essential for efficient computation of option policies and models in a large domain such as this.

Results

To evaluate the utility of hierarchical representations of behavior in generating sophisticated variation we conducted an experiment in the Light Box domain to compare the variational capabilities of an agent with only primitive actions to those of one with a hierarchy of options. We let each agent generate blind trials by choosing each trial uniformly randomly from its available action set. Each agent was run for 100,000 time steps and the data were averaged over 50 runs. Results are presented in Figure 3. The graph shows the ratio of changes in the value of each state variable to the total number of trials generated. Each state variable number corresponds to the labels of the lights in Figure 2. Note that the number of trials corresponds to the number of decisions made, not the number primitive actions executed. Thus for the agent using options each trial may take longer than one time step to execute, but is the result of only one decision.

One can see from Figure 3 that the agent with only primitive actions is able to successfully change the states of the circular lights easily, as they have no dependencies. However, using only primitive actions to alter the states of the triangular, rectangular, and diamond-shaped lights meets with little success since these variables have dependencies that are satisfied only by chance in previous trials. The agent with options on the other hand exhibits equal change frequencies for each state variable by using its options to set the appropriate configuration of dependencies for each variable it decides to change. Note that the distribution of values for the agent with options still does not sum to one because of the inherent stochasticity of the domain.

These results show the advantage of having a behavioral repertoire at an appropriate level of abstraction when generating blind variations for creative search in large structured domains. From the point of view of the agent with options, the search space of possible configurations is relatively small in that few trials need to be generated to chance upon any specific configuration of lights. This is not the case for the agent with only primitives, since it must often generate many trials before successfully altering the states of the non-circular lights.

Discussion

We have presented computational techniques for modeling some critical aspects of creative processes in artificial systems based on a perspective which views creativity as a blind variation-and-selection process. This process takes place in a space defined by the set of potential creative products in a given domain and can be modeled as a sequential decision process consisting of trial-and-error search through this space for highly-valued products according to some evaluation metric. Although this space is in general too large to search through at primitive levels of abstraction, we have outlined a set of criteria for creative systems that allow for transformation of this space into abstract spaces in which efficient search for complex creative products becomes feasible.

The first of these is a hierarchical set of skills for altering existing creative products along different dimensions at various levels of abstraction. A blind variation mechanism must possess the ability to take steps through the creative landscape at many different granularities to make efficient search possible. The second is a method for predicting both the outcomes and corresponding expected values of complex manipulations of existing structures proposed by the variation mechanism. Without the ability to make these predictions, every proposed trial must be executed and evaluated explicitly according to the selection metric, which makes efficient search an impossibility in complex environments. We have presented formal methods from the RL literature which satisfy these criteria and have shown some preliminary results illustrating their capacity for creative behavior.

Some critics of the viewpoint of creativity as a search process have argued that creative processes in humans seem to be more than just search in the good-old-fashioned-AI (GOFAI) sense. One argument is that often either one does not have access to the full specification of the search space or the space itself cannot be represented explicitly, and as such one cannot apply generic search algorithms. Although we agree that the search space may indeed be too large either to represent or to search though efficiently using standard techniques, we believe that creative processes are in fact searching through reduced or mapped spaces that represent various levels of abstraction of the intrinsic space. The necessity of these abstract representations and hierarchical skill sets to manipulate products represented within them motivates our criteria for efficient creative search.

Another criticism of the blind variation-and-selection view is that creative variations of existing products often do not seem to be blind as suggested here, but rather more directed. We maintain a rejection of this criticism similar to Campbell (1960), who points out that although many variations may not appear to be chosen blindly, these variations themselves are the result of previous blindly-driven trial-and-error processes. The abstractions accumulated through learning make possible blind variations that seem directed because of the granularity at which they are generated. The large jumps in the search space they afford are indeed composed of directed behaviors which were selectively retained from previous, blindly generated trials, but the choice of such a jump itself is made blindly. Thus, what appear to be directed, goal-driven paths through the search space are in fact blind variations and selective retentions at differing levels of abstraction. The value functions of an RL framework provide a well-developed formal mechanism for selecting variations at the right level of abstraction to generate highly-valued creative products given a consistent selection metric.

Although we have presented formal methods that exhibit some of the characteristics of creative processes, there are many dimensions of creativity that our work does not address, and several assumptions in these approaches that may not be realistic in many real world domains. The first and most obvious is the prior knowledge of the primitive transition dynamics assumed in our demonstration. In general it is most desirable for a creative agent to discover this structure on its own and use the estimated model to construct useful abstract skills. We are currently working on methods for autonomous structure discovery in factored domains that builds upon the work of Degris, Sigaud, and Wuillemin (2006).

Other directions for future research include the incorporation of perceptual abstraction over the features of a given domain. The work presented here constructs hierarchical representations of behaviors for efficient creative search, but leaves flat the representation of creative products whose construction these skills afford. We are also looking into incorporating perceptual abstraction into such a framework to further increase its sophistication. The ability to abstract at both the behavioral and perceptual levels will likely confer additional advantages to those outlined above for an artificial creative system.

Acknowledgements

We are ever grateful to Anders Jonsson for his helpful insights and especially for making his code available to us. Additionally, we would like to thank George Konidaris, Scott Kuindersma, Özgür Şimşek, and Pippin Wolfe for useful discussions and comments. The work presented here was supported by the National Science Foundation under Grant No. IIS-0733581. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Bellman, R. E. 1957. *Dynamic Programming*. Princeton, New Jersey: Princeton University Press.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 1995. Exploiting structure in policy construction. In *The 14th Annual International Joint Conference on Artificial Intelligence (IJCAI)*.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 2000. Stochastic dynamic programming with factored representations. *Artificial Intelligence* 121(1):49–107.

Campbell, D. T. 1960. Blind variation and selective retention in creative thought as in other knowledge processes. *Psychological Review* 67:380–400.

Dean, T., and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational Intelligence* 5:142–150.

Degris, T.; Sigaud, O.; and Wuillemin, P.-H. 2006. Learning the structure of factored markov decision processes in reinforcement learning problems. In *The 23rd Annual International Conference on Machine Learning*.

Gabora, L. M. 2005. Creative thought as a non-darwinian evolutionary process. *Journal of Creative Behavior* 39(4):65–87.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, Michigan: University of Michigan Press.

Jonsson, A., and Barto, A. G. 2006. Causal graph based decomposition of factored mdps. *Journal of Machine Learning Research* 7:2259–2351.

Koza, J. R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Simonton, D. K. 1999. *Origins of Genius: Darwinian Perspectives on Creativity*. New York: Oxford University Press.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: MIT Press.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112:181–211.

Sutton, R. S. 1991. Integrated modeling and control based on reinforcement learning and dynamic programming. In *Advances in Neural Information Processing Systems 3*.

Watkins, C. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation, King's College, Cambridge.