Nicholas K. Jong
nkj@cs.utexas.edu
(512) 417-5573

5107 Leralynn Street
Apartment 201
Austin, Texas 78751

Research Statement

Despite decades of research into artificial intelligence, today's computer systems still require extensive manual effort to program and to customize to our needs. These current limitations motivate my research, which aims to create autonomous agents that can adapt to the complexities and uncertainties of the real world. I hope to develop what I believe is the missing ingredient: knowledge representations and algorithms that are simultaneously grounded in experience yet abstract enough to permit effective reasoning.

Much of my work builds upon the foundation of reinforcement learning, a computational framework for learning behaviors from experience. Its emphasis on interaction data over prior knowledge leads to algorithms that are designed to handle arbitrary environments but that learn too inefficiently for many actual applications. My doctoral thesis grants learning agents inductive biases that fit general forms of real-world structure. Hierarchy plays a particularly important role in how I allow agents to generalize more effectively from finite data to infinite environments.

Apart from my contributions to fundamental learning algorithms, I also have hands-on experience with promising application domains. I worked on a practical hybrid algorithm for combining reinforcement learning with human expertise in an autonomic computing setting, and I designed a hierarchical behavior-execution framework for a team of soccer-playing AIBO robots. My experience with these projects has convinced me that even these complex environments are now within the reach of learning algorithms that map effectively between low-level data and high-level concepts. This theme of abstracting deep structure from surface experience pervades my research.

Reinforcement Learning

One of my goals is to realize the promise of reinforcement learning for creating autonomous agents. A strength of the standard approach is its elegant domain-independent representation of learned knowledge: a real-valued function that maps states and actions to the expected long-term value possible by performing that action in that state. This value function encodes knowledge in a form grounded entirely in states, actions, and rewards, which form the interface between agent and environment. This representation is invaluable for a complete agent, which must transform perceived states into executed actions, but it is too shallow to permit efficient reasoning and learning. I have championed "indirect" algorithms that estimate the value function by modeling the world and then planning with this model. Others have investigated the theoretical and practical benefits of such model-based algorithms in discrete environments where exact solutions are possible, but my thesis demonstrates that this deeper knowledge can play a critical role in allowing agents to generalize effectively in realistic environments with continuous state spaces and hierarchical structure.

Most current algorithms only apply generalization at the level of the value function, implementing the simplistic inductive bias that an action has similar *values* in similar states. I have shown that agents can learn more rapidly by making the reasonable additional assumption that an action has similar *effects* in similar states [3, 4]. A key insight is that the dynamics of a given action, which only concern one time step and one action, generalize more broadly than the action's value, which depends on other actions in future time steps. For example, a mobile robot should quickly learn that the effect of taking a step forward is similar in most situations: the action's effect generalizes to every state free of an obstacle. In contrast, the value of taking a step forward depends on the distance to the robot's goal. Endowing learning agents with such effective and broadly applicable inductive biases remains a central focus of my research.

Hierarchy is another form of real-world structure absent in the basic formalism that underlies reinforcement learning. Most current algorithms learn behaviors as direct mappings from each possible state to a primitive action. Like using assembly code for every programming task, this approach scales poorly. Although many researchers have incorporated subtasks into standard algorithms, I contributed deeper understandings of how precisely hierarchy speeds up learning [2]. By integrating hierarchical decomposition and model learning, I show that hierarchy can be a valuable means for enforcing hard biases in an agent's planning and exploration [5]. The practical benefits of this work include guidelines for the design of hierarchies that effectively communicate domain knowledge to learning agents.

Autonomic Computing

While at IBM's Watson Research Center, I investigated a practical methodology for using human expertise to bias reinforcement learning in the setting of autonomic computing, where the goal is to make computer systems self-configuring, self-healing, self-optimizing, and self-protecting. Previous work had learned effective resource allocation policies in a corporate data center using ad hoc learning constraints, but I developed a hybrid algorithm that learned superior policies by using neural networks to generalize values estimated from data from a human-engineered policy [8, 9, 10]. The engineered policy provided a good starting point, but it ignored certain real-world effects, such as the transient switching costs of changing a resource allocation, that a learning algorithm observes in the data.

Autonomic computing remains a promising platform for real-world applications of reinforcement learning algorithms, which can now achieve tangible and socially relevant performance improvements over other methods. The demands of this domain inspired my later design of sample-efficient model-based algorithms that can tackle the continuous state spaces of realistic problems. The growing prevalance of multi-core systems makes it particularly affordable to investigate interesting resource-management problems where reinforcement learning can make a difference.

Robotics

Another promising domain for reinforcement learning is robotics, where machine learning techniques are just beginning to find compelling applications. My own experience with robotics stem from two separate projects. The first project aimed to help mobile robots form topological decompositions of their environments, aiding the process of simultaneous localization and mapping by identifying recognizable places [1]. One of my long-term goals is to develop general algorithms that might subsume such specific examples of abstracting noisy data into structural knowledge.

The other project on which I've worked is a RoboCup robot soccer team at the University of Austin, using the four-legged SONY AIBO platform. My specific contribution was the development of a hierarchical architecture for executing behaviors such as finding the ball and kicking it towards the opponents' goal [6, 7]. This project reinforced the importance of "closing the loop" by developing complete agents that combine sensing the world and acting in the world. The limited capabilities of current robots illustrates the hazards of studying individual components of intelligence, such as perception and planning, independently.

Future Research

No one could have predicted the full impact of the microprocessor and networks such as the Internet, and in just a couple of decades, robust artificial intelligence technology will have no less of a benefit for society. Cars and other machines that now require human intervention will operate themselves, circumventing human error and freeing people for less mundane activities. Computer and other systems, which are growing in complexity beyond our ability to program effectively, will learn to manage themselves. In short, machines will adapt to human needs, instead of forcing humans to adapt to the inflexible logic of computers, making possible devices beyond current imagining.

I have taken steps towards this future by investigating opportunities for learning agents in autonomic computing and robotics, and I have designed algorithms that employ realistic inductive biases to generalize efficiently from limited data. My next steps will include closing the gap between my existing work on domain-independent learning agents in artificial environments and the real applications for which I've engineered customized algorithms. I will design scientific agents that don't just learn the values of human-specified parameters but that instead test hypotheses to discover structure such as hierarchy. To this end, agents must reason efficiently about their beliefs over possible worlds. Designing reasonable representations and approximations for this process will be a primary research challenge.

Practical representations will also allow autonomous agents to benefit from manually encoded domain knowledge and special purpose algorithms, when available. Model-based and hierarchical approaches create opportunities for combining learned models in some portions of a hierarchy with existing models in other portions. Another specific possibility involves using the hierarchical abstraction of low-level, stochastic action models into high-level, deterministic subtasks, permitting the application of classical planning algorithms. In general, machine learning has the potential to connect challenging applications to decades of artificial intelligence research by defining grounded representations compatible with existing algorithms. In the years to come, I will continue to dedicate my research to developing these connections, with the ultimate goal of bringing artificial intelligence into the real world.

References

[1] Patrick Beeson, Nicholas K. Jong, and Benjamin Kuipers. Towards autonomous place detection using the extended voronoi graph. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[2] Nicholas K. Jong, Todd Hester, and Peter Stone. The utility of temporal abstraction in reinforcement learning. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems*, 2008.

[3] Nicholas K. Jong and Peter Stone. Model-based exploration in continuous state spaces. In *Proceedings of the Seventh Symposium on Abstraction, Reformulation and Approximation*, 2007.

[4] Nicholas K. Jong and Peter Stone. Model-based function approximation in reinforcement learning. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007.

[5] Nicholas K. Jong and Peter Stone. Hierarchical model-based reinforcement learning: R-MAX + MAXQ. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, 2008.

[6] Peter Stone, Kurt Dresner, Selim T. Erdogan, Peggy Fidelman, Nicholas K. Jong, Nate Kohl, Gregory Kuhlmann, Ellie Lin, Mohan Sridharan, Daniel Stronger, and Gurushyam Hariharan. Ut austin villa 2003: A new robocup four-legged team. *RoboCup-2003: Robot Soccer World Cup VII*, 2004.

[7] Peter Stone, Mohan Sridharan, Daniel Stronger, Gregory Kuhlmann, Nate Kohl, Peggy Fidelman, and Nicholas K. Jong. From pixels to multi-robot decision-making: A study in uncertainty. *Robotics and Autonomous Systems*, 54(11):933–943, November 2006. Special issue on Planning Under Uncertainty in Robotics.

[8] Gerald Tesauro, Nicholas K. Jong, Rajarshi Das, and Mohamed N. Bennani. A hybrid reinforcement learning approach to autonomic resource allocation. In *Proceedings of the IEEE International Conference on Autonomic Computing*, 2006.

[9] Gerald Tesauro, Nicholas K. Jong, Rajarshi Das, and Mohamed N. Bennani. Improvement of systems management policies using hybrid reinforcement learning. In *Proceedings of the Seventeenth European Conference on Machine Learning*, 2006.

[10] Gerald Tesauro, Nicholas K. Jong, Rajarshi Das, and Mohamed N. Bennani. On the use of hybrid reinforcement learning for autonomic resource allocation. *Cluster Computing*, 10(3), September 2007.