Research Statement

Devika Subramanian
Rice University

August 2002

1 The enterprise

The essence of intelligence is performing actions appropriate to one's objectives given uncertain, incomplete information and limited computational resources. My research goal is to develop principles for designing and analyzing task-directed, adaptive systems embedded in the world. Such systems are natural progressions of controllers like thermostats, automatic braking systems, autopilots, and I believe, ultimately, humans.

I am interested in three general, inter-related questions in the design and analysis of adaptive, embedded systems. First, what aspects of the environment, task and its internal dynamics does a system need to model for autonomous decision-making? Second, how can a system build and maintain such models in changing environments? Third, how can a system with limited resources efficiently use these models to make decisions?

I believe that the path to answering these questions and uncovering the science of adaptive embedded systems runs through building useful systems in a variety of applications, and distilling core principles. Three technological confluences have made this approach especially timely: sensors which generate enormous amounts of low-level data, cheap computational cycles available in highly portable form to extract high-level information to base decisions on, and networks to effectively share extracted information widely. They provide new opportunities for building adaptive embedded systems on an unprecedented scale in every conceivable context.

My approach has led me to attack important problems in science and engineering using methods and models from adaptive systems and machine learning. The challenges posed by specific applications have enabled me to enrich and extended the algorithmic base for the design and analysis of adaptive systems. Along the way, I have also produced valuable and novel solutions to these problems.

Over the last decade, I have forged exciting inter-disciplinary collaborations in cognitive science and psychology, political science, compilers, operating systems and networks, mobile robotics, optomechanical design, biochemistry and macromolecular crystallography. In the next section, I describe each domain problem, the fundamental scientific or engineering question addressed by it, the results obtained, and their significance in the context of both the application domain and the overall science of adaptive embedded systems.

2 The projects

2.1 Tracking human learning

How do humans learn tasks with significant strategic and visualmotor components? Examples are submarine navigation and flight control. These tasks are difficult for humans to learn because they require the coordinated acquisition of a task strategy (e.g., an evasive maneuver) and the skills to implement that strategy (e.g., a visualmotor servo-loop). Current computerized training environments emphasize repeated performance of the task, and offer little help in tailoring learning to individual needs. Human teachers are also ineffective because subjects are rarely able to verbalize their strategies. In these training environments, over half the people fail to learn tasks of even minimal complexity.

Under the Hybrid Learning Program of the Office of Naval Research, my collaborators (Sandra Marshall of San Diego and Diana Gordon of the Naval Research Laboratory) and I worked on an instance of the above task class called the NRL Navigation Task, which involves learning to drive a simulated submarine[2, 3, 4, 5]. At each time step, subjects observe an instrument panel which can be in one of 10^{18} possible configurations and choose one of 153 action vectors. Success/failure feedback is provided at the end of a driving episode which can last between 50-200 time steps. The task is challenging enough that less than half the human subjects tested achieve the desired competence level. The task is even more challenging for machine learning algorithms. Out-of-the-box reinforcement learning achieves 3% success after 100,000 trials.[1]

I saw an opportunity to construct an embedded system that uses non-invasive recordings of visualmotor activity (coordinated eyetracker and joystick readings) to build high-level models of task strategies learned by humans. If the system could track the evolution of strategies in real-time, it would be useful in customizing training protocols.

The problem of acquiring and tracking models of strategies from low-level visualmotor data can be formulated in the framework of learning predictive stochastic models from non-stationary data. *Given a non-stationary Markovian data source generating a time series of input-output vector pairs of the form* $(\vec{i}(t), \vec{o}(t))$*, with inputs* $\vec{i}(t)$ *drawn from the set* $\vec{I}$ *and outputs* $\vec{o}(t)$ *from* $\vec{O}$*, find and track the output predictor* $\pi_t : \vec{I} \times \vec{I} \rightarrow \mathcal{P}(\vec{O})$. The predictor generates a probability distribution over the outputs $\vec{O}$ given the input configurations at the current and the previous time step. To solve this problem, we first characterize the nature of the non-stationarity of π_t. Does it change abruptly with time (regime-switching), or does it drift gradually? Next, we search for changepoints that segment the available data stream of input-output pairs into quasi-stationary pieces characterized by a single π. Finally, we evaluate the quality of the learned π_t both in terms of the accuracy of its action predictions, and its information content.

There are many challenges in learning the output predictor π_t from the given time series: the data set is massive, noisy and high-dimensional. There is no off-the-shelf technology for analyzing this data. Surprisingly, local instance-based models turn out to be the most effective representations for the π_t's. These models are sparse and consist of a set of representative input-output vectors. Prediction is done by a new variant of locally weighted regression on the stored vector pairs called *biased dimension elimination*, described in [1]. To determine changepoints, we compute derivatives of successive π's constructed from the time series. This strategy derivative is the average of the Kullback-Leibler (KL) divergences of the predicted output vector distributions over a small randomly sampled set of input vectors. It is very robust and can be computed in real-time.

Figure 1 shows strategy derivatives for two subjects from our study. It reveals that learning is characterized by periods of slow evolution of a strategy punctuated by conceptual shifts in which policies change radically. Successful subjects have a distinct derivative profile, while unsuccessful subjects change strategies too rapidly. Strategy derivatives provide early warning about failing subjects and identify those who are unable to learn the task due to the lack of a good, stable strategy. The strategy models π_t match the learning curves of subjects extremely well (see `http://www.cs.rice.edu/~devika/ONR/animations.html`

---

[1] After significant engineering of state space descriptions by observing human learning, as well as by providing informative intermediate reward functions, reinforcement learning achieves 99.99% accuracy on the task after 10,000 trials. The resulting policy is the gold-standard policy that we use for shaping human training.

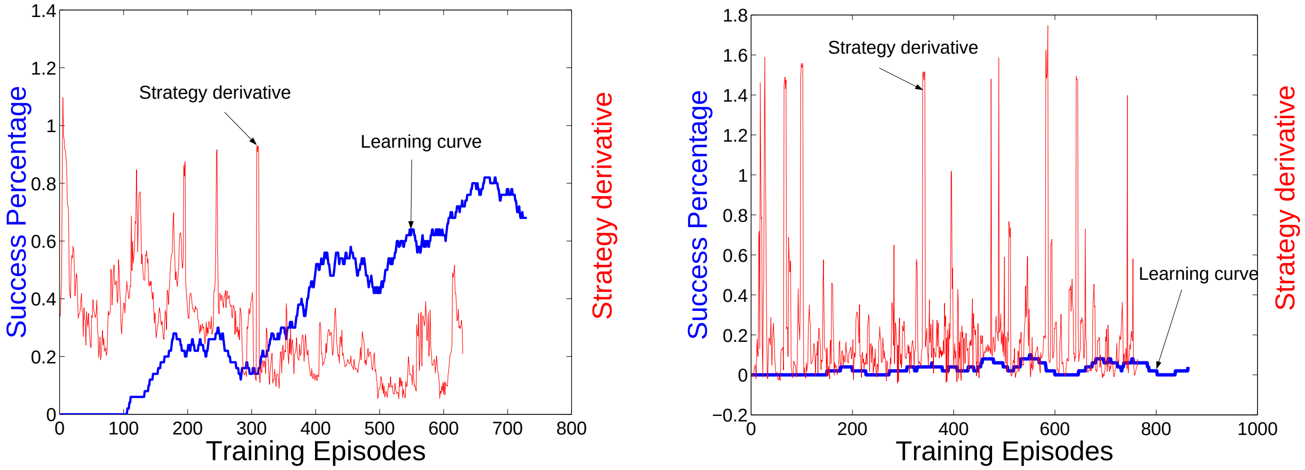


Figure 1: Strategy derivative for two subjects. The peaks in the strategy derivatives correspond to radical changes in the strategy used by the subject. While successful subjects have between 3-5 "aha" moments in the first half of the training period, in which they dramatically alter their action choice distributions; unsuccessful subjects continue to experiment with new strategies throughout the training protocol. The peaks in the strategy derivatives are correlated tightly with performance improvements for the subject on the left. The difference in the rate at which strategies are changed gives a clear early indication of success in learning. Note that for the subject on the left, success percentage on the task continues to improve even though the strategy has stabilized. We conjecture that the improvement is due to compilation of the strategy in the subject's visualmotor system. We are presently engaged in a functional brain imaging study to verify this conjecture.

to see animations of the model and the subject). The models can be compared with a known optimal policy to identify sensor configurations for which the subject does not have the right action choice; providing direct feedback to subjects on their decision making.

The adaptive embedded system functions as a *computational microscope* for understanding human learning by extracting probabilistic models from large scale, objective, low-level sources of performance data. It is a scalable solution that harnesses the power of new sensors and computing to fundamentally change practice in training. It has yielded efficient new algorithms for detecting changepoints and building predictive stochastic models for massive, noisy, non-stationary, vector time series data.

Existing methods for modeling human learning in psychology and cognitive science have been unable to analyze this class of tasks, in part because of difficulties with obtaining verbal protocol data. My work is one of the first in the community to directly use objective performance data from eyetrackers and joysticks to derive high-level strategy models using statistical machine learning techniques. It sheds light on how humans acquire tasks with strategic and visualmotor components by providing the first computational glimpse of strategy-level dynamics on a complex task. My work has received considerable attention in cognitive science. I gave an invited talk at the hybrid learning symposium at the Annual Cognitive Science Conference in 1999 along with key figures in the cognitive modeling community. In November 2001, I was invited to the Distinguished Speaker Series of the Office of Naval Research to present my work. I am currently engaged in a collaboration with a brain imaging scientist to verify strategy shifts at the neural level.

2.2 Forecasting international conflict

With the end of the Cold War, the threat of a global-scale nuclear conflagration has receded. Unfortunately, this has not meant that all forms of serious conflict have been eliminated. Since the fall of the Berlin Wall, the United States has engaged in three wars (the Gulf War, Kosovo, and Afghanistan), as well as having its people - and even its territory - subject to attack. Other areas of the world have either experienced even more conflict (the Middle East) or threaten to do so in the immediate future (South Asia, the Korean Peninsula). At the same time, there is a vast amount of information that exists (and is growing daily) on the World Wide Web. This information is in the form of websites maintained by media outlets throughout the world, such as Reuters. Many of these sites contain not only current news, but extensive historical archives. There are automated event extraction programs such as (`http://www.ukans.edu/keds`) developed by political scientists to automatically extract event data from news reports. Event data are coded time series; each element in the series contains information on when an event occurs, who participated in it and what its characteristics are.

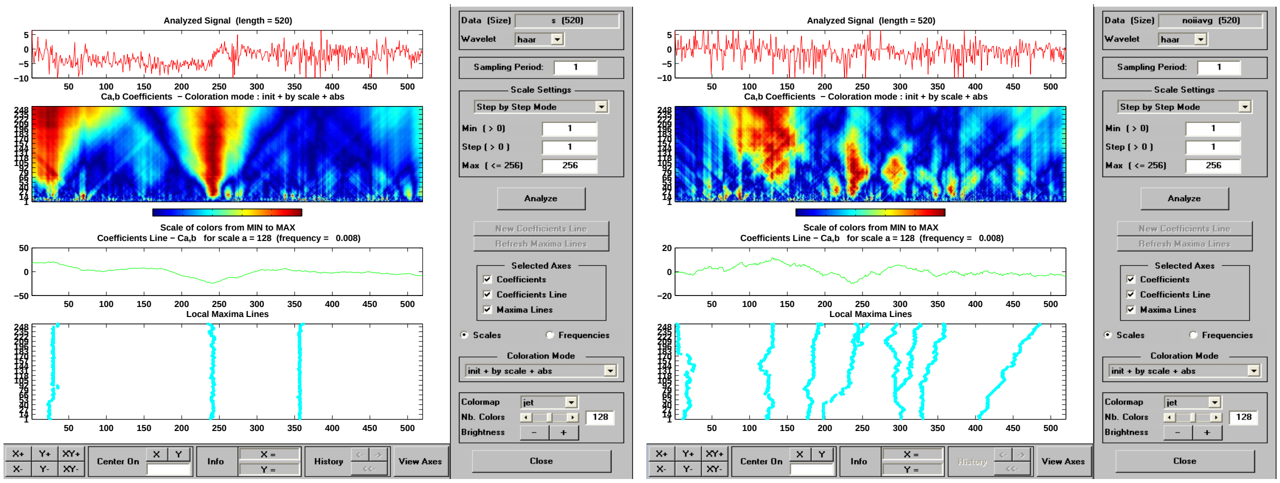
With my collaborator Richard Stoll of the Political Science Department at Rice, and with funding from the NSF's ITR program (as of Fall 2002), I am exploring the following question: is it possible to build predictive models of conflict based only on long-term event data extracted from news reports? Is a purely data-driven approach to modeling the genesis and evolution of conflict in various regions of the world even possible? This is an empirical question about the information content of automatically extracted event data, and requires the development of algorithms that can build and track stochastic models of conflict evolution based on low-level event data.

A critical step is to find a suitable representation for the event data stream; a hitherto unsolved problem. We have explored several choices, and the result of one of these is shown in Figure 2. In this experiment, we aggregated event data from the Middle East over the last twenty years into a one-dimensional time series, with each element representing an averaged score of events (ranging from +10 for cooperative events to -10 for hostile ones) for two week periods over all pairs of countries in the region. This time series turns out to be non-stationary. The algorithmic problem is: *given a noisy non-stationary time series, how do we determine its properties? What is the nature of the non-stationarity? Can we segment the time series into nearly stationary segments? How can we fit predictive models over these stationary segments? How can we detect the onset of a change in dynamics in such a time series?* Answers to each of these questions has real world significance. Singularities are a type of non-stationarity that correspond to abrupt changes in the time series. Being able to detect the onset of a singularity provides us with the basis for predicting a discontinuous change in the hostility levels.

Our results thus far are extremely exciting. We can detect singularities in the aggregated event score time series using continuous wavelets and Mallat's modulus maxima technique. They correspond very closely to militarized interstate disputes in this region over the last twenty years. Further, with high probability, we can predict singularities about eight weeks before they occur. I believe that our project has the potential to change the science of conflict prediction in a fundamental way, and to add significantly to the design of algorithms for analyzing non-stationary time series over long time scales.

2.3 Traffic-sensitive routing algorithms for dynamic networks

Network routing algorithms used today calculate shortest paths between nodes. The cost of a path is the sum of the cost of all links on that path. The cost of a link can be static or dynamic (i.e., a function of traffic on the link). The benefit of routing using dynamic cost metrics is that paths can be recomputed based on prevailing traffic patterns to reduce congestion, packet delay, and packet drops. Despite their advantages, dynamic metrics are not widely used in today's Internet. The technical reasons are that the overheads of updating routes based on dynamic link costs are very high and increase with increased



Data (Size) s (520)
Wavelet haar
Sampling Period: 1
Step by Step Mode
Min (> 0) 1
Step (> 0) 1
Max (<= 256) 256
init + by scale + abs
Colormap jet
Nb. Colors 128

Data (Size) noiiavg (520)
Wavelet haar
Sampling Period: 1
Step by Step Mode
Min (> 0) 1
Step (> 0) 1
Max (<= 256) 256
init + by scale + abs
Colormap jet
Nb. Colors 128

Figure 2: The figure on the left shows the analysis of the time series of scaled event scores averaged over all pairwise interactions among countries in the Middle East over two week blocks from April 1979 to March 1999. The figure on the right shows the same data with Iran and Iraq eliminated from the set. The Iran-Iraq war dominates events in this dataset and the figure on the right considers interactions with the Iran-Iraq war "subtracted" out of the data. The panel on the top shows the raw time series and the third panel from the top shows a moving average over the signal (computed with a window of 128 weeks). The moving average plot highlights the non-stationarity in the data and is a summary of the ebb and flow of conflict in the region. The second panel from the top in both figures is a multi-resolution wavelet analysis with the Haar wavelet. The red regions in the panel correspond to significant singularities. In the bottom panel on the left, the three lines that reach from the lowest scale to the highest correspond in time to the middle of the Iran-Iraq war, the Gulf War, and a military engagement between Israel and Lebanon in 1994/1995. These singularities are computed using Mallat's modulus maximum technique. On the right, the bottom panel highlights conflicts in the region involving countries other than Iran and Iraq – these conflicts become prominent once the effect of Iran and Iraq are subtracted from the analysis. The analysis was performed with the Continuous Wavelet 1-D tool in Matlab.

network traffic, and such route-updates often cause oscillations and instabilities.

With Peter Druschel of Rice University and student Johnny Chen, and with support from Southwestern Bell, I developed a new family of traffic-sensitive routing algorithms for dynamic networks. The problem of finding optimal routes in a network can be modeled as a distributed reinforcement learning problem. In fact, one of the popular algorithms in current use, called the distance-vector algorithm, performs Bellman updates of a distributed distance function $V : node \rightarrow \Re$. V records at every node, the shortest distance to every other node in the network. Neighboring nodes exchange their own V's and local link costs, until convergence. The primary challenge in this environment is maintaining optimal routes in the face of two distinct types of non-stationarities: topology changes caused by link/node failure or recovery and link cost changes caused by network traffic.

The general question, of which the routing problem is an instance, is: *how do we design distributed reinforcement learning algorithms that maintain optimal policies in environments with non-stationarities that occur at different time scales? How do we demonstrate their effectiveness, both theoretically and experimentally?*

Traditional distributed reinforcement learning algorithms are *indirect path-sampling* algorithms. Neighbors exchange costs of edges linking them as well as their own estimates of the cost of the shortest path between themselves and the other nodes in the network. Indirect path-sampling algorithms

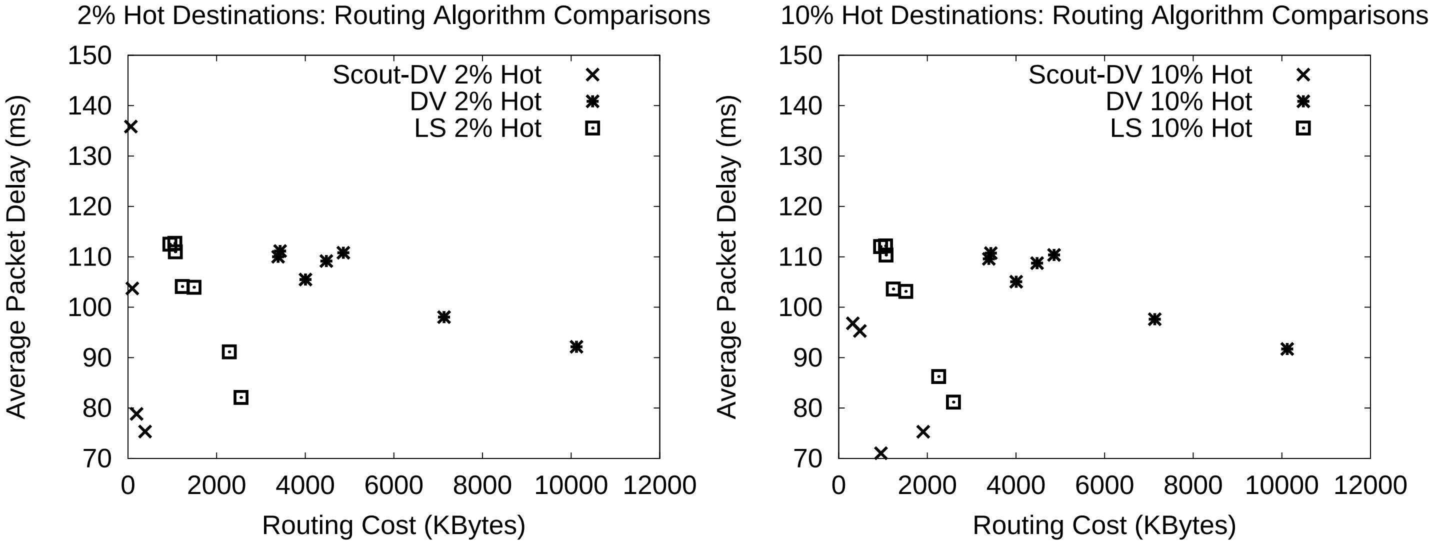


Figure 3: We compare the routing efficiency and end-to-end performance of our hybrid algorithm Scout-DV, documented in [8], against those of traditional algorithms(distance vector (DV) and Link State (LS)) when using dynamic link metrics. The performance comparison was done on a moderately sized Internet-like network. with a standard network simulator called `netsim`. The x-axis of the plot shows the overhead measured in the number of bytes of messages dedicated to routing, while the y-axis displays the quality of routes as measured by average end-to-end packet delay. The results of our experiments show that our algorithm can provide higher route quality at a significantly lower routing cost under traffic conditions that arise in real networks. The plot on the left shows the route cost/quality tradeoff when 2% of the nodes are hot, i.e., recompute routes based on traffic, while the plot on the right documents the same tradeoff when 5% of the nodes are hot. The effectiveness of the algorithm stems from independent, time-staggered recomputation of important paths using dynamic metrics, allowing for splits in congested traffic that cannot be made by the traditional routing algorithms.

update all paths in the network simultaneously, in response to the current system state. This has the unfortunate effect of causing route oscillation, as every node switches paths from a currently congested link to a currently uncongested one.

I developed a new family of distributed reinforcement learning algorithms that learn path costs in the network by direct sampling [12, 11]. Each node in the network sends probes (called ants) which update cost information along all visited nodes. Information about costs is propagated only through the direct probes. Direct sampling algorithms effectively decouple path recomputations among nodes, eliminating stability problems that arise with traditional algorithms. However, the route computation overheads increase because there is no pooling of information between neighbors. To keep these overheads low, we only allow a small fraction of the nodes in the network to perform direct path sampling updates.

Analysis of Internet packet traces shows that a high percentage of network traffic is destined for a small number of nodes. Only these *hot* nodes perform direct-path sampling to update routes in response to changes in traffic. All other nodes use the traditional algorithms to compute routes, and update them only when the network topology changes.

The performance of our hybrid algorithm is shown in Figure 3. It is one of the first routing techniques that is stable under dynamic cost metrics, allowing for splits in congested traffic that cannot be made by traditional routing algorithms. It has lower overheads and better route quality than current routing methods. We published this algorithm at the top networking conference [8]. We introduced the concept of direct path sampling reinforcement learners [12]; our IJCAI 1997 paper is cited by subsequent work on network routing using reinforcement learning. Our hybrid blend of direct-path and indirect-path sampling is one of the few reinforcement learning algorithms that has been successfully fielded in the context of a large-scale real-world problem.

2.4 Sensor fusion for autonomous outdoor mobile robots

My goal is to build an autonomous mobile robot that gives tours of the Rice University campus. While there are many successful mobile robots that autonomously navigate and explore *indoor* environments like museums and offices over extended periods of time, there are very few examples of outdoor robots. This is because outdoor environments are highly variable. The challenge of solving the mapping and localization problem for outdoor robots is still open. *I believe that pursuing the engineering goal of building an autonomous outdoor tour guide for an urban campus will advance the science of robust navigation in unstructured outdoor environments.*

With several undergraduate students in computer science and electrical engineering and with support of the School of Engineering at Rice University, I have developed a low-cost, robust localization method for an outdoor tour guide robot. The robot is equipped with on-board odometry and a cheap differential marine GPS system. It fuses odometry and GPS data using extended Kalman filtering. The biggest challenge is handling two types of *non-stationarities* in GPS data quality. There are abrupt changes in GPS position readings caused by sudden obstructions to line-of-sight access to satellites when the robot is near large concrete structures. More gradual changes in GPS data quality are caused by disparities in atmospheric conditions. We build and track conditional probability models of these variations, by actively and repeatedly sampling GPS data along different routes. These distributions are updated continuously by exponential forgetting to handle drift in GPS data quality.

We experimentally demonstrate [13] that our localization technique which explicitly models and learns variations in GPS data performs better than those that do not adapt to such variations. With a GPS receiver that delivers 1 meter accuracy in clear spaces and 10-15 meter accuracy otherwise, we have been able to localize consistently to within 40 cm through a challenging tour of the Engineering Quadrangle of Rice University. This is one of the first demonstrations of accurate localization in an environment as challenging as an urban campus.

2.5 Adaptive compilers

In the past decade, we have seen an explosion in the number of computers in use and in their diversity. As computers have found new applications, developers and users have new priorities for their programs. Traditionally, users have wanted programs to run fast. Now, issues such as code size, page faults, and power consumption have become important. Since most computers run compiled programs, a large part of the responsibility for addressing these issues falls to the compiler. While we understand how to build high-quality compilers that produce efficient code for most modern processors, actually building such a compiler is an expensive, labor-intensive effort that requires experts— viz., compiler writers who are in short supply.

A major component of a compiler writer's expertise is the ability to select code transformations and determine the order in which to apply them, for any program presented to the compiler. Generally, a small number (two or three) of different transformation sequences are bundled as levels of optimization. The choice of a transformation sequence has a direct impact on the quality of code that

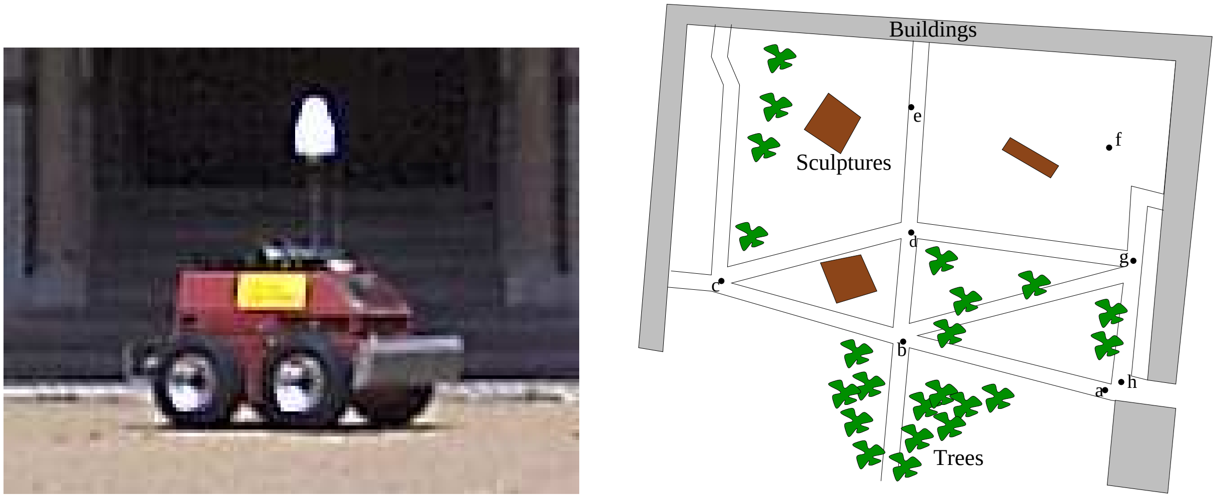


Figure 4: Virgil, the Rice campus tour guide shown on the left, is an ATRV Jr. from RWI Inc. It is named after the guide in Dante's Inferno. It comes equipped with sonars and odometry. We have added a differential marine GPS receiver and touch-sensitive bumpers for obstacle detection and avoidance. The way points in the tour of the Engineering Quadrangle are shown on the right. This is a challenging route that runs close to several tall buildings, large granite sculptures and several rows of trees.

the compiler generates. Making good choices is difficult for several reasons. The sheer number of code transformations to choose from is daunting. The relationship between features of the input program and improvements (or degradations) produced by a single code transformation is not formally characterized. The interaction between the code transformations themselves is poorly understood.

From a typical user's point of view, a compiler is a black box that only allows a few discrete optimization levels to be set. There is no way to communicate a new objective function (e.g., optimize size of compiled code) to it. Using human compiler writers to design transformation sequences for every conceivable program class, target machine architecture and objective function is not a scalable solution.

I am exploring the following question with collaborators Keith Cooper and Linda Torczon of Rice University, with support from NSF's ITR program: is it possible to build an adaptive compiler that autonomously selects code transformations and their application order, based on the input program, the performance characteristics of the target machine, and an explicit objective function?

This is an instance of the following general problem with the design of adaptive embedded systems: *what aspects of the environment, task and objective function does a system need to model in order to allow autonomous decision-making? Can these models be built from actively gathered observations?* For adaptive compilers, the environment is the target machine architecture and the task is compilation of the given input program. The decision that needs to be made is the selection of an transformation sequence that optimizes the given objective function. Observations correspond to evaluation of the objective function after compilation with a specific transformation sequence.

We have built a preliminary version of such an adaptive compiler[14, 16]. This compiler learns models of the impact (with respect to a variety of objective functions) of its transformation sequences on different program classes and target architectures. It does so by biased random sampling of the space of transformation sequences. After a training phase, the compiler uses the models to tailor a transformation sequence for a given program, target machine and objective function.

In our early experiments, reported in [16], we optimized a set of well-known benchmark programs for code space using a simple biased random sampling algorithm based on genetic algorithms. The resulting

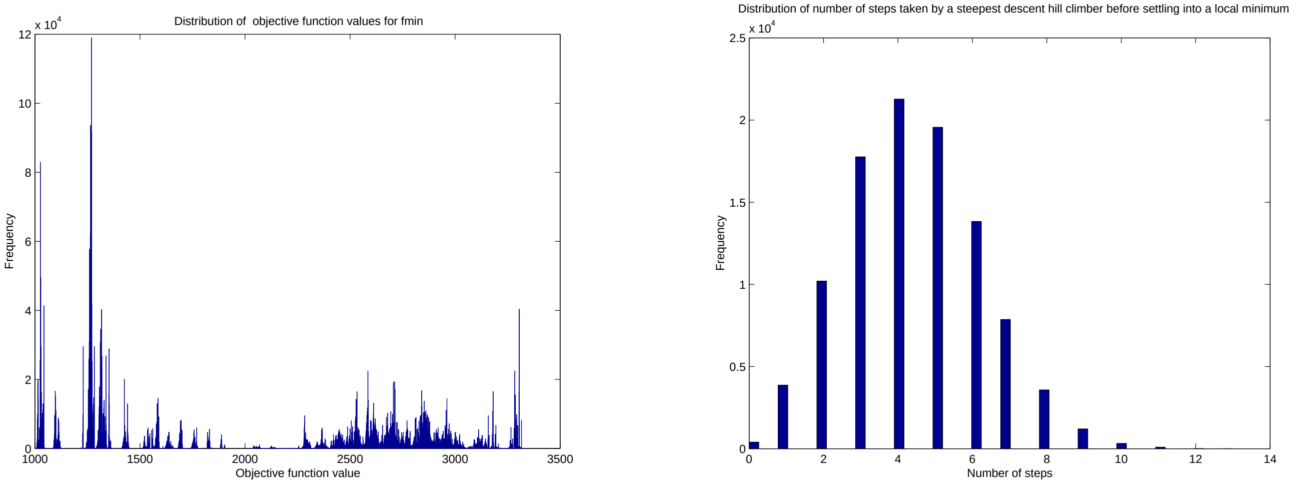


Figure 5: On the left is the distribution of objective function values for the `fmin` benchmark for all 5^{10} transformation sequences of length ten drawn from a pool of five optimizations known to be important for it. The x-axis denotes the range of objective function values from a minimum of 1003 to a maximum of 3300. The y-axis is the frequency of sequences which generate a given objective function value. About 10% of the solutions are within 10% of the global optimum. The graph on the right shows the distribution of the number of steps needed by a hill climbing algorithm with random restarts to settle into a local minimum which is within 10% of the global minimum. On average half the transformations in a randomly generated sequence need to be modified by a stochastic gradient descent algorithm to find a good local minimum. From other experiments we have calculated that the probability that the hill climber with random restarts finds a good local minimum is 94.6%

sequences yielded an average of 13% reduction in code size over a standard optimizing compiler. The compressed code was on average 25% faster than the original optimized code. The best conventional techniques for code compression in the compiler community get a 5% reduction on space, but slow down code by 2% on the same benchmarks.

To scale the compiler to richer objective functions, we need to devise new models for capturing the effect of transformation sequences on programs running on specific target architectures. In the experiments detailed in [14], our goal was to understand the impact of each sequence in the space, to help us determine the kinds of conditional probability distributions over transformation subsequences that need to be built. We chose a benchmark program called `fmin`, a Fortran program with 44 basic blocks and a complex control-flow graph which displays interesting behavior from an optimization standpoint. It is small enough for us compile the program with all 5^{10} sequences of length ten drawn from a suite of the five transformations known to be useful for this program. Generating the results of 5^{10} compiles was a massive computational experiment which took approximately 5 CPU months. The distribution of objective function values for these sequences is shown in Figure 5. 10% of the solutions are within 10% of the value of the single global optimum in this space of sequences. To understand the distribution of these "good" solutions, we devised a hill climber with randomized restarts to iteratively improve a randomly initialized sequence of transformations of length ten. We observed that 94.97% of the solutions found by this algorithm were within 10% of the global minimum. It demonstrated that with an appropriate number of restarts, the probability of finding a good solution is very high despite the preponderance of local minima and the relative sparsity of good solutions. Figure 5 shows the distribution of the number of steps taken (swaps of elements in the sequence) by the randomized hill climber before it settles into a local minimum. On average, half of the elements in a randomly initialized sequence have to be modified to obtain a good solution. We have also built sequential models characterizing the sequence space. For this, we trained Hidden Markov Models (HMMs) to distinguish good sequences from poor ones using several hundred randomly drawn sequences from the entire space. These trained HMMs are able to classify the remaining sequences accurately, and the structure of the HMMs reveal long range dependencies in good transformation sequences.

Adaptive compilers that learn explicit probabilistic models of good transformation sequences can generate optimizations tailored to particular programs, architectures and objective functions. More extensive experimental work from which I hope to abstract general active sampling schemes for learning models of interactions between modules in a large system, is underway. The methods will be used to generate probabilistic models of the interaction between code transformations available in a compiler. Our new NSF ITR award to pursue this vision over the next five years gives us the opportunity to build and field adaptive compilers. We hope to change the economics of the process of building, tuning, and using optimizing compilers over the next five to ten years.

2.6 Active experimentation in protein crystallography

While there are over 10^9 protein sequences known, less than 5000 of them have their structures determined. Crystallization of a biological macromolecule is the essential first step in structure determination by X-ray crystallography. Crystallization of a new macromolecule is still very much a trial and error process requiring the search of a large parameter space of experimental settings. It takes anywhere between a few weeks to several years, to find the relatively few, idiosyncratic conditions that lead to diffraction quality crystals. Over the years, the crystallography community has accumulated an electronic repository of all published crystallization conditions in the Biological Macromolecular Crystallization Database (BMCD).

With support from the NIH, and with crystallographer and biophysicist John Rosenberg and computer scientist Bruce Buchanan of the University of Pittsburgh, I worked on the following question: is it possible use the accumulated data on successful crystallizations to bias the search for conditions under which new macromolecules will crystallize? Is it possible to learn probabilistic models that capture the relationships between macromolecular class and the sets of conditions under which diffraction quality crystals can be obtained? Another way of stating this is to recall Nobelist Max Perutz's famous quote that "crystallization is a little like hunting, requiring knowledge of your prey and a certain low cunning". Using Perutz's analogy, we ask, are there good "hunting grounds" in the space of the experimental conditions wherein it is profitable to concentrate one's efforts? Do different classes of macromolecules have different "favorite haunts" that can be exploited by a cunning crystallographer? Can these haunts be modeled as conditional probability distributions and learned directly from the BMCD? Can the models be used to design sets of experimental conditions that maximize probability of success, given the BMCD data?

This learning and active experimentation problem is an instance of a more general question that arises in the context of adaptive embedded systems: *How do we design an embedded system that learns models of its interaction with an environment, and uses them to make directed probes of that environment to refine them further?*

Using inductive learning[19, 20] (decision rules) and statistical learning (naive Bayes classifiers), we have built useful predictive models that classify experimental conditions as being likely to generate diffraction quality crystals or not. Some of the decision rules have extracted associations between experimental parameters that were hitherto unknown to protein crystallographers[19]. In [18], we used the experimental data to learn the conditional probability distributions specifying a Bayesian network describing the interactions between the experimental parameters. The learned Bayesian network infers the probability of obtaining diffraction quality crystals given a set of experimental conditions. These empirically derived probabilities guide an incomplete factorial design algorithm that generates sets of

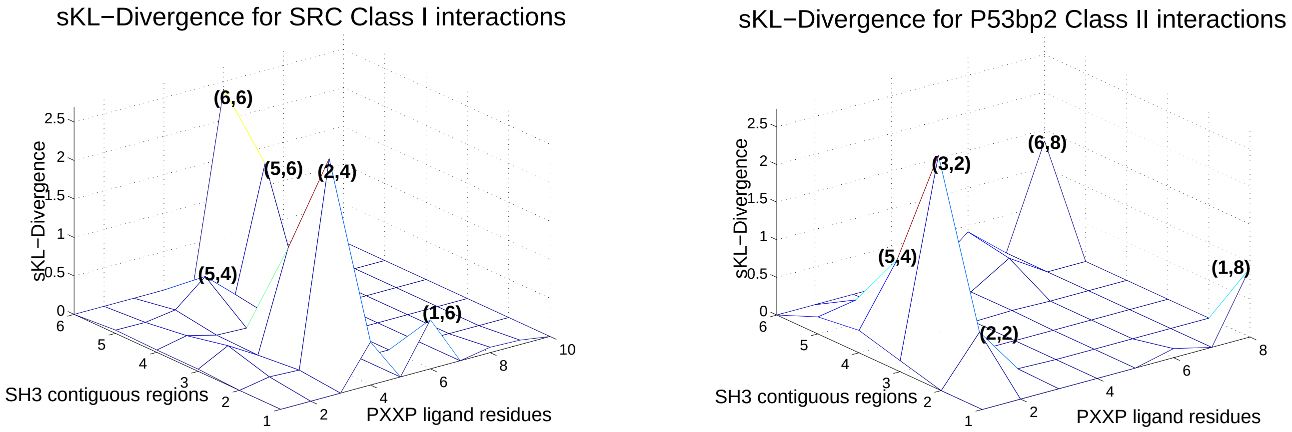


Figure 6: The sKL-divergences between distributions for positive and negative ligands for the Src SH3 domain and the p53bp2 domain. The peaks identify the SH3 region-PxxP residue pairs where there is the most difference in charge and hydropathy distributions between positive and negative ligands. They are therefore the most likely interaction sites.

experimental conditions most likely to succeed given the data in the BMCD. Our experiment generation program is currently in use in the crystallography community. It promises to improve the probability of success of future crystallization trials on new proteins.

2.7 Learning models of protein-protein interaction

Protein-protein interactions underlie almost all cellular processes including DNA replication, transcription, signalling and cycle cell control. Knowledge of protein-protein interactions is critical for a variety of applications, including pharmaceutical design which involves the search for small ligands that bind with larger macromolecules. Less than 0.1% of the expected number of protein-protein interactions for the human genome have been detected and documented via wet laboratory experiments. This is because even the high-throughput wet laboratory techniques for detecting protein-protein interactions take a long time to perform. Further, some of the methods (e.g., yeast-two-hybrid) have high false positive rates. Computational prediction of protein-protein interactions based on available experimental data on interactions appears to be essential to map out protein-protein interaction networks at the genomic level.

The problem of learning predictive models of protein-protein interactions from available interaction data is an instance of the general problem of learning models of complex systems from observational data. What makes this instance particularly challenging is that there is no standard approach to conceptualizing and representing protein-protein interactions. There is great diversity in such interactions, ranging from ones in which there are few direct contacts between the amino acids on the proteins (PxxP ligands binding to SH3 domains is an example), to ones (like the leucine zipper) involving a large number of contact points. It is unlikely therefore that a single model or representation will fit all interactions. Another challenge in this problem is the issue of validating computational predictions; the gold standard is still the result of a wet laboratory experiment.

With my collaborator Kathy Matthews of Rice University, her postdoctoral fellow Xinxing Tan, and my student Raj Bandyopadhyay, I have developed an approach for building models of protein-protein interactions from data. We have identified an interesting and useful class of protein-protein interactions for which (1) there is enough data to learn robust models, and (2) for which there are partial theories accounting for the interaction in the literature. These human-posited models of interaction offer a different basis for verification of the results of our computational predictors.

We have studied the interactions between proteins containing SH3 domains and proteins containing the PxxP sequence motif [21]. SH3 domains have now been identified in more than 1100 different proteins in organisms ranging from yeast to humans. They act in diverse processes including signal transduction, cell cycle regulation, and actin organization. Individual SH3 domains display distinct binding specificities. However, many questions remain in the study of SH3 domains and their interaction with PxxP containing proteins. What are the factors that determine the ability of different SH3 domains to recognize particular PxxP containing proteins? Given a set of known SH3-PxxP interactions and non-interactions, can we learn a classifier to distinguish between them based solely on the properties of the residues in the pair? How well can a learned classifier reproduce the known binding specificities of SH3 domains? Can we analyze the learned classifier to find the primary determiners of the recognition specificity between SH3 domains and their PxxP containing partners? How well can a learned classifier predict binding of SH3 domains to unstudied PxxP containing proteins in a protein databank such as Swiss-Prot?

We used laboratory data gathered from combinatorial peptide library screening of 8 diverse SH3 domains to construct a body of positive and negative examples. We learned naive Bayes models of binding specificity of these SH3 domains and test them using a cross-validation approach. The models have prediction accuracies of 90% and higher with low false positive and negative rates. In addition, using the information-theoretic measure of symmetric Kullback-Liebler divergence, we visualize our classification model, as shown in Figure 6 to reveal sites on both proteins that contribute to the interaction. We used our classifiers to screen PxxP-containing proteins from Swiss-Prot for the eight SH3 domains. Between 80% to 99% of them are eliminated for five of the eight SH3 domains considered.

This is a significant result, because we have been able to derive an empirical theory of SH3-PxxP interactions from a fairly small data set. The theory is powerful enough to prune a significant number of potential interactions from consideration for wet experiments. Our empirical theory is consistent with the partial theory of interactions developed by biochemists, and refines it by highlighting individual differences in interaction patterns in SH3 domains. We are now verifying specific predictions made by our theory in the laboratory. Our hope is that our predictions will be vindicated which will allow us to publish our findings in *Science* later this year.

2.8 Theoretical foundations of embedded systems

In most of my work, I have addressed questions in the design and analysis of adaptive embedded systems by generalizing concrete solutions to problems that arise in a variety of challenging applications. I alo believe that there are some core theoretical questions that are best investigated by first formulating abstract solutions and then instantiating them in particular problem contexts. In the subsections below I describe four such fundamental questions and my contributions to their solution.

2.8.1 Bounded optimality

What does it mean to say that an embedded system is optimal? Since its inception, artificial intelligence has relied upon *perfect rationality* as the optimality criterion for embedded systems. A system is perfectly rational if its action choices maximize its objective function, with this maximization occurring under two assumptions: the system can compute all the consequences of what it knows, and this computation happens instantaneously. Such a criterion is inappropriate because it imposes fundamentally unsatisfiable requirements. Bounded systems cannot calculate all consequences of the knowledge they have, and they compute at less than infinite speeds. To evaluate bounded agents, we need a criterion that not only examines *what* actions are chosen, but also *when* they are chosen, since actions have time value in the world. With Stuart Russell of U.C. Berkeley, I proposed a new optimality criterion called *bounded optimality.* An embedded system is bounded optimal if its action selection procedure is a solution to the constrained optimization problem presented by its architecture (sensory and computing limitations) and the task environment. In [25] we show how to construct bounded optimal systems for a simple class of machine architectures in a broad class of real-time environments. We illustrate these results in the context of scheduling zipcode recognition algorithms for an automated mail sorting facility. We also develop a weaker property called *asymptotic bounded optimality* that we believe forms the basis of a new complexity theory for bounded embedded systems. This work has had a significant impact in resource-limited planning, anytime algorithms and design of bounded systems. I believe this is because of two reasons. First, we offer a realistic normative standard against which bounded systems can be evaluated. And second, we pioneer a model as well as an algorithmic approach for rationally trading off quality of solutions against computation time.

2.9 Automating design of optomechanical systems

Given a description of the desired behavior of a system and given knowledge of basic structural units from which the system can be composed, can we rapidly enumerate structures which implement that behavior? I studied this question in the context of conceptual design of opto-electromechanical systems. In [24, 22], with my student Edward Wang, I developed two new algorithms for systematic real-time enumeration of mechanism topologies from qualitative specifications of desired motions. The power of the methods stemmed from the use of qualitative representations of configuration spaces, the design of two new composition primitives, and the use of constraint programming. From first principles, we recreated many known designs found in design compendia and several novel designs for every day mechanisms ranging from clocks and eggbeaters to windshield wipers. In my current work, funded by the NSF, and in collaboration with Ron Goldman of Rice University, I have extended these techniques to cover opto-mechanical systems like imaging systems of copiers. There are two key ideas: (1) modeling the structure and behavior of optical components using affine transformations, and (2) using a systematic search algorithm integrated with a powerful algebraic constraint solver to compute three-dimensional layouts of optical components in the design.

By embedding the design tool in a constraint programming environment, we are able to automate the conceptual design of optical imaging systems — a step performed manually by human designers until now. All known copier imaging system designs have been reproduced very quickly by the program, and novel methods for folding optical paths have been devised, increasing the range of conceptual options for designing imaging systems of copiers.

2.10 Complexity of random 3-SAT

Many important problems in logical reasoning, scheduling and planning can be formulated in terms of finding models of propositional formulae. The problem of determining whether a propositional formula has a model is known to NP-complete. The last decade has seen an intense focus on the complexity of finding models for randomly generated propositional formulae in the 3-SAT normal form. This interest was stimulated by the discovery of a fascinating connection between the *density* of 3-SAT instances and their computational complexity.[2]

However, many questions regarding the average-case complexity of random 3-SAT formulae are still open. How does the average-case complexity of finding a model for a random 3-SAT formula vary as a function of the order (number of variables) and density? Is there a phase transition in which

---

[2] The density of a 3-SAT problem is the ratio of the number of clauses to the number of propositional variables in the instance.

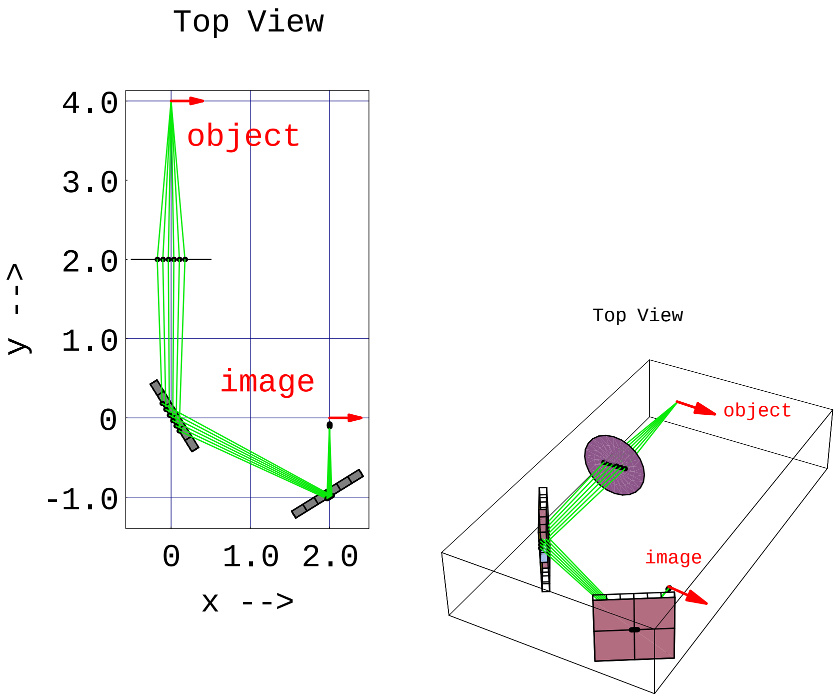


Figure 7: An imaging system designed by our optomechanical synthesis program. The top view and 3-D arrangement of components are shown. The system transforms an object at (0,4,0) oriented along the x axis into a real image of the same size at (2,0,0) also oriented along the x axis. Our constraint solving approach can automatically derive the rectilinear speeds of the optical components to keep the image of an object moving along a line at constant speed, projected on a stationary drum.

the complexity shifts from polynomial in order to exponential in order for a given density? Is the transition dependent or independent of the solver? While phase transitions have been documented for the probability of finding a model, there is no such evidence for average case complexity. Identifying complexity theoretic phase transitions for 3-SAT has significant implications not only for the design of model finders for propositional formulae, but also for our understanding of average-case complexity of solving hard combinatorial problems.

We studied [27, 28]three different complete SAT solvers, embodying very different underlying algorithms: GRASP, CPLEX, and CUDD. We observed new phase transitions for all three solvers, where the median running time shifts from polynomial in the order to exponential. The location of the phase transition is solver-dependent. While GRASP and CUDD shift from polynomial to exponential complexity at a density of about 3.8 (below the probability crossover density of 4.26), CUDD exhibits this transition between densities of 0.1 and 0.5. Our results are the first documentation of solver-dependent complexity phase transitions in the field. Some of our empirically results have been theoretically confirmed for DLL solvers like GRASP.

2.10.1 The origin of new representations

Given an embedded system performing a task, can we reformulate its representations of the task to make it perform the same task more efficiently? Wisdom is knowing what to ignore, and a computationally bounded system cannot afford to make distinctions irrelevant to its task. My thesis research[32] presented a theory of constructing new task-specific representations of problems based on the principle that irrelevant distinctions result in wasted computation. With support from the NSF, I developed algorithms for finding and collapsing unneeded distinctions by meta-theoretic analysis of computations performed by a system. Explicit minimization of unneeded computation uncovers new distinctions that trade off accuracy for computational effort. An extension to my theory was developed by my first PhD student, Adam Webber[31]. Starting from a specification of unneeded computation in a functional language, he developed algorithms for automatically deriving compiler optimizations for that language (both well-known optimizations like loop unrolling, loop fusion and common sub-expression elimination, as well as novel optimizations that do not have common names). In 1997, I was invited to edit a special issue of *Artificial Intelligence* on relevance with Russ Greiner and Judea Pearl. This special issue[29] has consolidated some of the best current work (some of which are follow-ups to my own thesis work) on the automatic determination of relevant distinctions for a variety of computational systems.

3 Lessons learned

My experience with a variety of rich applications has revealed the importance of three core principles for adaptive system design. First, adaptive systems must build probabilistic models of themselves, their tasks and their interactions with their environments. Second, to ensure robustness, it is imperative they detect and learn reliable models of environmental non-stationarity. Third, they need to actively probe the environment to continuously refine learned models, and improve task performance.

My work in tracking visualmotor learning, predicting international conflict, and constructing robust sensor fusion algorithms for localization in complex outdoor environments exemplify the first and second principles. They all involve tracking high-level models (represented as conditional probability distributions) from non-stationary, low-level sensor data. In human learning, the distribution of action choices $P(\vec{o}|\vec{i_t}, \vec{i_{t-1}})$ underwent sudden shifts as revealed by a KL-divergence analysis. In the prediction of conflict, the distribution of aggregated event scores exhibited singularities when viewed through the lens of multi-resolution wavelets. In the robot navigation example, the GPS sensor distribution $P(reading|location)$ changed characteristics abruptly as a function of the number of visible satellites. All these problems have pushed the science of representation and rapid estimation of probability distributions on high-dimensional data.

My work on predicting protein-protein interactions, crystallography, adaptive compilers, and network routing algorithms underscore the importance of the first and third principles. While the two examples from bioinformatics pertain to one-shot prediction tasks, the network routing and adaptive compiler problem are sequential in nature. In the crystallography and the protein-protein interaction problems, the adaptive system builds conditional probability distributions of the form $P(outcome|inputs, action)$ from a starter set of examples to calculate actions that maximize probability of the desired outcomes. These actions also serve as probes of the environment that in turn refine the learned probability distributions. This data-driven approach to construction and refinement of models may well be the way for the all sciences to abstract theories and to guide data gathering using new high-throughput devices. In the compiler and network routing problems, the adaptive system samples sequences of transformations and nodes to learn good paths in the combinatorial landscape of all possible sequences. The information is summarized in the form of local cost and probability gradients. In the networks application, the need to keep cost models current in a dynamic environment leads to the development of a different sampling approach called direct-path sampling. This sampling technique is useful in any environment where it is important to decouple updates to different pieces of a distributed model.

All of the problems I have worked on have a common representational and computational core. By abstracting away from individual applications and revealing the relationships between task, environment and system design, I have made progress in the science of adaptive systems. By elucidating design principles for adaptive embedded systems and by building a comprehensive toolbox of techniques for constructing and tracking high level probability models from sensory data, I hope to make the construction of complex embedded systems as routine as the design of automatic braking systems and cruise controllers.

References

[1] Tracking the evolution of learning in a visualmotor task, Technical report TR02-401, Department of Computer Science, Rice University, August 2002.

[2] Inducing hybrid models of learning from visualmotor data, in Proceedings of 22nd Annual Conference of the Cognitive Science Society, August 2000.

[3] Modeling individual differences in the ONR Navigation task, in Proceedings of 20th Annual Conference of the Cognitive Science Society, July 1998 (with D. Gordon).

[4] A cognitive model of learning to navigate, in Proceedings of 19th Annual Conference of the Cognitive Science Society, Stanford, CA, July 1997. (with D. Gordon).

[5] Cognitive modeling of action selection learning, in Proc. 18th Annual Conference of the Cognitive Science Society, San Diego, July 1996. (with D. Gordon)

[6] Comparison of action selection learning methods, in Proc. 3rd International Workshop on Multistrategy Learning, Harpers Ferry, May 1996, pp 95-102. (with D. Gordon)

[7] Events, patterns and analysis: forecasting international conflict in the twenty-first century, funded NSF ITR proposal (at `http://www.cs.rice.edu/~devika/projects/`) (with R. Stoll), February 2002.

[8] A new approach to routing using dynamic metrics, Proceedings of INFOCOMM, 1999 (with P. Druschel and J. Chen).

[9] An efficient multi-path forwarding method, in Proceedings of INFOCOMM-98, (with P. Druschel and J. Chen).

[10] A simple, practical, distributed multipath routing algorithm, TR98-320, July 1998, Rice University (with J. Chen and P. Druschel).

[11] Ants and reinforcement learning: a case study in routing in dynamic networks, TR96-259, Rice University (updated July 1998) (with J. Chen and P. Druschel).

[12] Ants and reinforcement learning: a case study in routing in dynamic networks, in Proceedings of IJCAI-97, (with P. Druschel and J. Chen).

[13] Robust localization methods for an autonomous campus tour guide, International Conference on Robotics and Automation, 2001, (with R. Thrapp and C. Westbrook).

[14] Compilation order matters, Technical Report, January 2002, Rice University (with K. Cooper, L. Torczon and T. Harvey).

[15] Adaptive optimizing compilers for the 21st century, *Journal of Supercomputing*, 23:(1), 7-22, 2002 (with K. Cooper and L. Torczon).

[16] Optimizing for reduced code space using genetic algorithms, *ACM SIGPLAN Workshop on Languages, Compilers, and Tools for Embedded Systems (LCTES)*, Atlanta, GA, 1999, (Proceedings will appear as an issue of SIGPLAN Notices) (with P. Schielke and K. D. Cooper).

[17] An experimental evaluation of list scheduling, TR98-, September 1998, Rice University (with K. Cooper and P. Schielke).

[18] Statistical methods for the objective design of screening procedures for macromolecular crystallization, *Acta Crystallographica D* 56:817-827, 2000 (with J. Rosenberg, B. G. Buchanan, D. Hennessey, P. Wilkocz).

[19] Induction of Rules for Biological Macromolecule Crystallization, Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology, Stanford University, AAAI Press, pp 179-187, August 1994 (with J. Rosenberg, B. Buchanan, V. Gopalakrishnan, D. Hennessy).

[20] The Crystallographer's Assistant, Proceedings of AAAI-94, AAAI/MIT Press, August 1994 (with B. Buchanan, V. Gopalakrishnan and D. Hennessey).

[21] Predicting protein-ligand interactions from primary structure, Technical Report TR02-387, Rice University, February 2002 (with R. Bandyopadhyay, X.X. Tan and K. Matthews).

[22] Kinematic Synthesis with Configuration Spaces, in *Research in Engineering Design*, Vol 7, pp 193-213, 1995 (with E. Wang).

[23] Artificial Intelligence and Conceptual Design, Proceedings of IJCAI-93 (invited paper), Morgan Kaufmann, pp 800-809, August 1993.

[24] Constraint-based Kinematic Synthesis, Proceedings of the 7th International Workshop on Qualitative Reasoning about Physical Systems, pp 228-239, May 1993 (with E. Wang).

[25] Provably bounded optimal agents, in *Journal of Artificial Intelligence Research*, pp 575-609, Vol 2, May 1995 (with S.J. Russell).

[26] Provably Bounded Optimal Agents, Proceedings of IJCAI-93, Morgan Kaufmann, pp 338-344, August 1993 (with S. Russell and R. Parr).

[27] Random 3-SAT: the plot thickens, in *Constraints*, 2002 (to appear) (with M. Vardi, D. Demopolous, A. San Miguel Aguirre, C. Coarfa).

[28] Random 3-SAT: the plot thickens, in *Proceedings of the International Conference on Constraint Programming*, 2000 (with M. Vardi, D. Demopolous, A. San Miguel Aguirre, C. Coarfa)

[29] The relevance of relevance, Artificial Intelligence, Vol 97(1-2), pp 1-5, 1997 (with R. Greiner and J. Pearl).

[30] Shift of vocabulary bias in speedup learning, in *Machine Learning*, Vol 20, pp 155-191, 1995.

[31] Adam Webber, Optimization of functional programs by grammar thinning, ACM Transactions on Programming Languages, Vol 17, No. 2, pp. 293-330, 1995.

[32] A theory of justified reformulations, PhD thesis, Stanford University, March 1989.