Research Directions in Interactive AI

Marie desJardins
Dept. of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250
mariedj@cs.umbc.edu

Abstract

The MAPLE (Multi-Agent Planning and LEarning) research group at UMBC has several ongoing projects that emphasize user interaction with AI methods. This position paper briefly summarizes these projects: (1) representing and incorporating a user's qualitative domain knowledge into machine learning algorithms; (2) creating understandable visual representations of high-dimensional probabilistic domain models (Bayesian networks); (3) developing cost-sensitive constraint satisfaction methods that allow the system to reason about the cognitive costs of involving a user in checking constraints; and (4) integrating automated planning and constraint satisfaction methods with interactive plan editing techniques to permit mixed-initiative plan generation in complex, real-world domains.

The long-term research vision of the projects described here is to develop techniques and infrastructure for building interactive, life-long AI systems. *Interactive* systems would work closely with a domain expert or end user, leveraging the strengths of both participants. By contrast, most current AI methods act as "blackboxes," taking data as input and generating solutions as output. *Life-long* systems would persist over time, learning from experience and transferring knowledge from earlier tasks and environments to later ones. This is in contrast to most current AI systems, which take a "single-shot" approach, solving single problems in isolation from each other. The objective of this research is to work towards building "glass-box" AI systems that can be understood, controlled, and interacted with by a user, and that persist in time, adapting themselves to a changing environment and "learning to learn" through knowledge transfer across a series of tasks.

Using Qualitative Knowledge in Learning

We are developing interactive learning methods that allow domain experts for a particular learning problem to express their background knowledge in an expressive interaction language, semantically grounded in qualitative reasoning methods (Forbus 1996; Kuipers 1986). This domain knowledge would be incorporated into the learning system in the form of qualitative partial models and annotated training examples (i.e., explanations of why a particular example is an instance of the phenomenon of interest).

We plan to explore and compare the use of embedded methods (modifying the learning algorithms themselves) and wrapper methods (preprocessing the data or postprocessing the model) for incorporating this background knowledge, and to develop model assessment techniques that can analyze and compare the learned models with respect to the background knowledge and observed data. The latter techniques would be able to answer questions such as, "Is the background knowledge consistent with the data? If not, in what regions of attribute space is it inconsistent?"

These methods are being developed and applied in the context of web searching; we also plan to investigate other applications such as meeting scheduling, e-mail sorting, and game playing.

Visualizing Probabilistic Models

Using inductive machine learning techniques to construct classification models from large, high-dimensional data sets is a useful way to make predictions in complex domains. However, these models can be difficult for users to interpret. We have developed visualization methods that help users to understand and analyze the behavior of a learned model.

In (Rheingans & desJardins 2000), we explored three classes of methods for projecting the high-dimensional model space into a two- or three-dimensional display space: feature subset selection, principal components analysis, and self-organizing maps.

In our approach, the probability distribution associated with a learned model[1] is projected into the display space and visualized as a color-coded background map. Instances are displayed as colored glyphs (red for negative instances and yellow for positive instances), allowing the user to easily visualize misclassified instances: False positives are seen as yellow instances that appear in low-probability regions of the space; false negatives are seen as red instance that appear in high-probability regions. The original attribute space can

[1] In the work we have done to date, we have used Bayesian networks as the learned models, but the visualization methods could be applied to any methods that produces probabilistic class predictions.

be visualized as contour lines (isolevels) or surfaces in the display space.

These projection methods have been applied to visualize models formed for several classification domains, including automobile gas mileage, income data from the U.S. census, and predicted outcomes in a military simulation. We are currently performing theoretical and empirical analyses to measure the quality of these projections along various dimensions such as region preservation, attribute contour smoothness, and specificity (clustering) of the projection.

Cost-Sensitive Constraint Satisfaction

A constraint satisfaction problem consists of a set of variables and constraints, or restrictions, on the legal values of these variables. A solution to a CSP is a set of values for all of the variables such that all or most of these constraints are satisfied (Kumar 1992).

In real-world domains, there may be costs or penalties associated with testing the constraints among the variables. In interactive constraint satisfaction, humans may need to verify the legality or practicality of certain variable choices, resulting in time delays and possibly significant cognitive load on the human. Costs may also arise from other sources: for example, information gathering may be required to test the constraint; expensive or time-consuming simulations may be necessary; or lab experiments may need to be done. Although some of these costs are incurred only after a complete solution is found, many of them can arise when evaluating a partial solution.

Although there has been significant work by the constraint programming research community on minimizing total *solution* cost (e.g., in weighted CSPs), and many researchers use the number of constraint checks as a metric for search cost, previous research had not addressed the problem of minimizing *validation* cost in the case where the cost of validating constraints can vary. If we can develop effective general heuristics for minimizing constraint-checking costs, while mazimizing the quality of the final solution, then we can apply these heuristics in interactive CSPs to ensure that the constraints presented to the user for for checking are those constraints that are the most relevant and critical to the success of the overall task.

We have developed a formal framework for modeling constraint checking costs, and have applied this framework to develop and test heuristics for two domains (N-queens and sports scheduling) (Sansare 2002).

An obvious heuristic for minimizing total search cost to order the tasks so that the less costly tasks are completed (i.e., the lower-cost constraints are checked) before the more costly ones in order to reduce the overall cost of the solution process. In the case of interactive constraint satisfaction, this heuristic means that constraints that can be checked by the system should be tested first, only then asking the user to validate the remaining constraints.

Depending on the structure of the constraint network and the distribution of costs, however, this obvious solution may not always be the best heuristic. Consider, for example, a travel planning domain. The user may need to confirm the dates selected. The “cheap-constraints-first” heuristic would recommend selecting the dates last, only after the other steps are completed. However, this could potentially result in enormous amounts of backtracking, if the system selects plane flights and hotels that are not available on the user’s preferred dates. Therefore, if many lower-cost decisions depend on the outcome of a particular constraint test, or the probability of satisfying a particular constraint is low (i.e., in the travel case, if only a few dates are acceptable to the user), it may make sense to perform the expensive test first, before committing (even tentatively) to any of the subordinate decisions. As shown in the experimental results of our early work, the structure of the solution space can sometimes lead this counterintuitive heuristic to perform better [Sansare 2002].

Mixed-Initiative Planning

PASSAT (Plan-Authoring System based on Sketches, Advice, and Templates) is currently being developed and applied by SRI International[2] to support rapid distributed planning for military operations (Myers *et al.* 2002). PASSAT uses AI planning and constraint representations to encode a library of templates that describe known planning strategies. Users can apply any these templates to develop a plan, can create their own plans through editing and “plan sketching” facilities, and can specify advice or guidelines for the planning system. PASSAT then applies generative planning and constraint reasoning methods to expand the plan, identify flaws in the plans sketched by the user, and track the status of the planning process.

The cost-sensitive constraint satisfaction methods were originally developed to support the PASSAT planning system. The costs in this case are “cognitive penalties” for requesting information from the user, or information gathering costs for collecting relevant data for evaluating the plan.

In future work on PASSAT, UMBC and SRI plan to jointly develop methods that would make PASSAT more adaptive and responsive to a particular user’s requirements. Learning methods would be applied to learn user preferences (e.g., which tasks are most important, how the user prefers to view the plan and planning status, and which templates the user typically applies in a given class of situations). Case-based learning and reinforcment learning methods would be applied to acquire procedural knowledge from previous planning instances.

Conclusion

There are many ways in which intelligent agents can be personalized. One key research problem is to develop interactive version of previously automated AI methods. These interactive methods would allow users to control the behavior of intelligent agents in flexible, yet powerful ways. We have described four ongoing research projects at UMBC that are exploring a range of methods to increase the interactivity of AI learning, constraint satisfaction, and planning techniques and the agents that use them.

[2]The author of this paper was a co-PI on the PASSAT project before leaving SRI to join UMBC.

References

Forbus, K. D. 1996. Qualitative reasoning. In *CRC Handbook of Computer Science*. CRC Press.

Kuipers, B. 1986. Qualitative simulation. *Artificial Intelligence* 29:289–338.

Kumar, V. 1992. Algorithms for constraint-satisfaction problems: A survey. *AI Magazine* 13(1):32–44.

Myers, K. L.; Wolverton, M. J.; Tyson, W. M.; Jarvis, P. A.; Lee, T. J.; and desJardins, M. 2002. PASSAT: User-centric planning technology. Submitted to 3rd International NASA Workshop on Planning and Scheduling for Space.

Rheingans, P., and desJardins, M. 2000. Visualizing high-dimensional predictive model quality. In *Proceedings of IEEE Visualization 2000*.

Sansare, S. 2002. Incorporating constraint checking costs in constraint satisfaction problems. M.S. thesis.