Doctoral Consortium Dissertation Abstract

Saket Joshi
Dept. of Computer Science
Tufts University
161 College Ave.
Medford, MA, USA
sjoshi01@cs.tufts.edu

Abstract

Broadly, my current research focuses on two areas. One area is Knowledge Representation. I co-invented First Order Decision Diagrams (FODD) and Generalized First Order Decision Diagrams (GFODD). (G)FODDs are a formalism for compactly representing real and Boolean valued functions over relational structures. We developed algorithms for composition and logical simplification (reduction) of these functions by theorem proving and model checking methods. We discovered properties and proved several theoretical results for (G)FODDs. (G)FODDs have many potential applications. We have already demonstrated their utility in representing and solving Relational Markov Decision Processes (RMDP). The other area focuses on building agents that can take actions in complex, dynamic and stochastic environments to achieve a predefined objective. Decision Theoretic Planning (DTP) has been one of the most successful frameworks for this task. The idea is to represent such problems as Markov Decision Processes (MDP), exploit structure in the problem definition and solve the MDP using Dynamic Programming methods. We extended the field of DTP by developing algorithms to represent and solve RMDPs using (G)FODDs. Based on this we also developed a planning system FODD-PLANNER that demonstrated performance comparable to top ranking systems from the international planning competition. Later we invented a new paradigm for planning by learning within the framework of the FODD-PLANNER and achieved drastic improvement in planning efficiency by leveraging model checking methods for FODD reduction.

Building agents (Russel and Norvig 2002) that adapt and thrive in their environment is a very important problem within Artificial Intelligence (AI). Usually such agents become able to take actions in the environment, towards achieving their objective, either by searching, by reasoning or by learning. In fact, in one view it is *the* problem of AI. This is because humans are agents that adapt and act in complex environments in order to achieve an objective - the maximization of total happiness. This broad research area can then be seen to encompass various other fields of AI like Problem Solving, Knowledge Representation, Planning and Acting, and Machine Learning. Research work in this field has always been of deep interest to me. Milestones along this way help to solve relevant practical problems of interest. To that extent I am interested in a variety of sub-fields of AI. My dissertation reflects this.

Current Research

I have concentrated on the following areas of research over the last few years.

Knowledge Representation:

I invented First Order Decision Diagrams (FODD) (Wang, Joshi, and Khardon 2008), a new knowledge representation formalism, along with my co-authors. A FODD compactly represents real or Boolean valued functions over relational structures. One can think of a FODD as a relational Algebraic Decision Diagram (ADD) (Bahar et al. 1993) or a relational Binary Decision Diagram (BDD) (Bryant 1986). When the leaf values are Boolean, a FODD represents a function free formula in First Order logic with existentially quantified variables. The semantics of FODDs follow those of (Groote and Tveretina 2003). Later we invented Generalized FODDs (GFODD) (Joshi, Kersting, and Khardon 2009) by extending the representation power of FODDs to arbitrary quantification. FODDs are a subclass of the set of function free formulas in First Order logic with existentially and universally quantified variables which itself forms a subclass of GFODDs. GFODDs are, therefore, very expressive. Further, since FODDs and GFODDs are functions, they can be composed to form more complex functions, and logically simplified or reduced to yield a compact representation. We developed several algorithms for the manipulation and logical simplification of FODDs and GFODDs, and proved several theoretical results about them. Algorithms for the reduction of FODDs and GFODDs can be categorized into as theorem proving and model checking based reductions. The model checking approach, in particular, (Joshi, Kersting, and Khardon 2009) has proved very efficient. FODDs and GFODDs have found utility in solving relational Markov Decision Processes (Joshi and Khardon 2008). Given, their expressive power and manipulability, however, we believe they can be utilized in a variety of applications.

Relational Markov Decision Processes:

Markov Decision Processes (MDP) have been a successful formalism for solving probabilistic planning problems. The fact that solutions to MDPs generate policies rather than action sequences is particularly attractive for probabilistic planning. This approach came to be known as Decision Theoretic Planning (Boutilier, Dean, and Hanks 1999). Classical solution techniques for MDPs, like value iteration (VI) (Bellman 1957) and policy iteration (PI) (Howard 1960), were based on dynamic programming. These early solutions, however, required enumeration of the state space. Owing to the curse of dimensionality (Bellman 1957), even for reasonably small problems, the state space can be very large. This can be seen easily for propositionally factored domains when the state is defined by N binary variables and the number of possible states is 2^N.

Several approaches were developed to handle propositionally factored domains (Boutilier, Dearden, and Goldszmidt 1999; Kearns and Koller 1999; Guestrin et al. 2003; Hoey et al. 1999). One of the most successful of these, SPUDD (Hoey et al. 1999) demonstrated that if the MDP can be represented using algebraic decision diagrams (ADDs) (Bahar et al. 1993), then VI can be performed entirely using the ADD representation thereby avoiding the need to enumerate the state space. Propositionally factored representations show an impressive speedup by taking advantage of the propositional domain structure. However, they do not benefit from the structure that exists with objects and relations. (Boutilier, Reiter, and Price 2001) developed the first approach using relational structure and provided the Symbolic Dynamic Programming (SDP) algorithm in the context of situation calculus. This algorithm provided a framework for dynamic programming solutions to Relational MDPs that was later employed in several formalisms and systems (Kersting, van Otterlo, and De Raedt 2004; Hölldobler, Karabaev, and Skvortsova 2006; Sanner and Boutilier 2009; Wang, Joshi, and Khardon 2008).

A Relational Markov Decision Process (RMDP) describes a mathematical model of interaction between an agent and a stochastic, dynamic environment, when the world is described by objects and relations among them. The advantage of the relational representation is abstraction. An abstract solution to the relational problem is also a solution to a concrete problem with a very large number of state variables. One can plan at the abstract level without grounding the domain, potentially leading to more efficient algorithms. In addition, the solution at the abstract level is optimal for every instantiation of the domain and can be reused for multiple problems. However, this approach raises some difficult computational issues because one must use theorem proving to reason at the abstract level, and because for some problems optimal solutions at the abstract level can be infinite in size. Following (Boutilier, Reiter, and Price 2001) several abstract versions of the value iteration (VI) algorithm have been developed using different representation schemes. For example, approximate solutions based on linear function approximations have been developed and successfully applied in several problems (Sanner and Boutilier 2009).

Motivated by the success of algebraic decision diagrams in solving propositional MDPs (Hoey et al. 1999; St-Aubin, Hoey, and Boutilier 2000), we introduced a novel approach to solving RMDPs (Wang, Joshi, and Khardon 2008). This approach is based on SDP, the dynamic programming solutions for RMDPs. Specifically, we represented the RMDP using FODDs and then solved the dynamic programming problem by manipulation and simplification of FODDs. Later we made the algorithm practical by introducing approximation techniques and built a prolog-based system, FODD-PLANNER based on these ideas (Joshi and Khardon 2008). FODD-PLANNER has been successful in solving problems from the international planning competition (IPC) with results comparable to top ranking systems from IPC (Joshi and Khardon 2008; Joshi, Kersting, and Khardon 2010). Finally we designed the algorithm VI-GFODD (Joshi, Kersting, and Khardon 2009), wherein a RMDP is represented using the more expressive GFODDs and solved by manipulating them. We proved that the same dynamic programming approach as with FODDs works for a very useful subclass of GFODDs. This subclass includes function free First Order formulas with existentially and universally quantified variables.

Planning by Learning:

A number of recent papers have employed ideas from the field of Machine Learning to solve planning problems (Fern, Yoon, and Givan 2003; Gretton and Thiebaux 2004). The thread connecting these approaches is the utilization of a training set of example states to generate a policy or a utility function (which indirectly defines a policy) and then use it as a heuristic to choose actions. One of the paradigms in this area is the "learning to act" model (Khardon 1999; Martin and Geffner 2000; Yoon, Fern, and Givan 2002) where a policy is learned from solved examples of small problems and is used to solve larger problems. Within this paradigm, we invented RETELL, a new learning algorithm that induces relational decision trees with leaves that are themselves relations among objects. Given data in the form of *state-action* pairs, RETELL induces a decision tree with actions in the leaves. This classifier can then be used as a policy to solve planning problems. We achieved state-of-the-art performance with RETELL. Later we introduced a new paradigm for planning by learning (Joshi, Kersting, and Khardon 2010), where given a model of the world the planner generates a set of states of interest, and uses this additional information to help focus the dynamic programming on regions of the state space that are of interest and lead to improved performance. We demonstrated the value of this idea by introducing novel model checking reduction operations for FODDs in FODD-PLANNER. Once again we achieved state-of-the-art performance and a drastic improvement in planning efficiency.

Sequential Supervised Learning (SSL):

SSL is a paradigm for learning classifiers for data when examples come in sequences. Text-to-speech mapping, part-of-speech tagging and protein sequence prediction are some applications of SSL. Along with my co-author, I developed and experimented with recurrent sliding window based learning tools. We discovered some interesting properties of the learning algorithms and achieved state-of-the-art performance on benchmark datasets.

Future Research Ideas

Following ideas that stem from my current work, I am interested in investigating the use of FODDs and GFODDs in other applications. The success of BDDs and ADDs at representing propositional logic formulas in a number of applications is good news for FODDs and GFODDs. For instance FODDs and GFODDs can find application in automated reasoning and formal verification. Another potential application is lifted inference (Poole 2003; Braz, Amir, and Roth 2005; Jaimovich, Meshi, and Friedman 2007; Milch et al. 2008; Singla and Domingos 2008; Sen, Deshpande, and Getoor 2008; 2009; Kersting, Ahmadi, and Natarajan 2009; Kisynski and Poole 2009). There exist approaches where ADDs represent the conditional probability tables in Bayesian Networks. Such a representation can then be used to compile the Bayesian Network for faster inference. It would be interesting to explore the possibility of using FODDs and GFODDs to provide the same leverage for Statistical Relational Models (Getoor and Tasker 2007). Another interesting direction for further research is towards developing efficient model checking reductions for GFODDs. Success in this area would immediately lead to a practical version of the VI-GFODD algorithm. This, in turn, would mean a dramatically more expressive, scalable and efficient approximate dynamic programming approach to solving RMDPs. Yet another research direction for RMDPs is through Reinforcement Learning (RL) (Sutton and Barto 1998). Many of the ideas for RMDPs discussed above can be extended and employed in a RL approach to solving RMDPs. For instance, a learning algorithm for FODDs and/or GFODDs would lead to a model-based RL method.

In the long run I would like to work on building adaptive agents for real world applications by leveraging research work in the areas of Decision Theoretic Planning, Reinforcement Learning, Statistical Relational Learning and lifted inference and information extraction from the web. Finally, since all of these areas can benefit from research in Machine Learning, I am interested in Machine Learning and its applications as well.

References

Bahar, R.; Frohm, E.; Gaona, C.; Hachtel, G.; Macii, E.; Pardo, A.; and Somenzi, F. 1993. Algebraic decision diagrams and their applications. In *IEEE /ACM ICCAD*, 188–191. Santa Clara, California: IEEE Computer Society Press.

Bellman, R. 1957. *Dynamic Programming*. Princeton University Press, Princeton, NJ.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 11:1–94.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 1999. Stochastic dynamic programming with factored representations. *Artificial Intelligence* 121:2000.

Boutilier, C.; Reiter, R.; and Price, B. 2001. Symbolic dynamic programming for first-order mdps. In *Proceedings of IJCAI*, 690–700.

Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In *In Proceedings of IJCAI-05, 19th International Joint Conference on Artificial Intelligence*, 1319–1325. Morgan Kaufmann.

Bryant, R. 1986. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers* C-35(8):677–691.

Fern, A.; Yoon, S.; and Givan, R. 2003. Approximate policy iteration with a policy language bias. In *Proceedings of NIPS*.

Getoor, L., and Tasker, B. 2007. *An Introduction to Statistical Relational Learning*. MIT Press.

Gretton, C., and Thiebaux, S. 2004. Exploiting first-order regression in inductive policy selection. In *Proceedings of UAI*.

Groote, J., and Tveretina, O. 2003. Binary decision diagrams for first order predicate logic. *Journal of Logic and Algebraic Programming* 57:1–22.

Guestrin, C.; Koller, D.; Parr, R.; and Venkataraman, S. 2003. Efficient solution algorithms for factored mdps. *JAIR* 19:399–468.

Hoey, J.; St-Aubin, R.; Hu, A.; and Boutilier, C. 1999. SPUDD: stochastic planning using decision diagrams. In *Proceedings of UAI*, 279–288.

Hölldobler, S.; Karabaev, E.; and Skvortsova, O. 2006. FluCaP: a heuristic search planner for first-order MDPs. *JAIR* 27:419–439.

Howard, R. 1960. *Dynamic Programming and Markov Processes*. MIT Press.

Jaimovich, A.; Meshi, O.; and Friedman, N. 2007. Template-based inference in symmetric relational markov random fields. In *Proceedings of UAI*.

Joshi, S., and Khardon, R. 2008. Stochastic planning with first order decision diagrams. In *Proceedings of ICAPS*.

Joshi, S.; Kersting, K.; and Khardon, R. 2009. Generalized first order decision diagrams for first order markov decision processes. In *Proceedings of IJCAI*.

Joshi, S.; Kersting, K.; and Khardon, R. 2010. Self-taught decision theoretic planning with first order decision diagrams. In *Proceedings of ICAPS*, to appear.

Kearns, M., and Koller, D. 1999. Efficient reinforcement learning in factored mdps. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, 740–747.

Kersting, K.; Ahmadi, B.; and Natarajan, S. 2009. Counting belief propagation. In A. Ng, J. B., ed., *Proceedings of UAI*.

Kersting, K.; van Otterlo, M.; and De Raedt, L. 2004. Bellman goes relational. In *Proceedings of ICML*.

Khardon, R. 1999. Learning to take actions. *Machine Learning* 35:57–90.

Kisynski, J., and Poole, D. 2009. Lifted aggregation in directed first-order probabilistic models. In *Proceedings of IJCAI*, 1922–1929.

Martin, M., and Geffner, H. 2000. Learning generalized policies in planning using concept languages. In *Proceedings of the International Conference on Knowledge Representation and Reasoning*. Morgan Kaufmann.

Milch, B.; Zettlemoyer, L.; Kersting, K.; Haimes, M.; and Kaelbling, L. 2008. Lifted probabilistic inference with counting formulas. In *Proceedings of AAAI*, 1062–1068.

Poole, D. 2003. First-order probabilistic inference. In *Proceedings of IJCAI*.

Russel, S., and Norvig, P. 2002. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence.

Sanner, S., and Boutilier, C. 2009. Practical solution techniques for first order mdps. *AIJ* 173:748–788.

Sen, P.; Deshpande, A.; and Getoor, L. 2008. Exploiting shared correlations in probabilistic databases. In *VLDB*.

Sen, P.; Deshpande, A.; and Getoor, L. 2009. Bisimulation-based approximate lifted inference. In *Proceedings of UAI*.

Singla, P., and Domingos, P. 2008. Lifted first-order belief propagation. In *Proceedings of AAAI*.

St-Aubin, R.; Hoey, J.; and Boutilier, C. 2000. APRICODD: approximate policy construction using decision diagrams. In *Proceedings of NIPS*, 1089–1095.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning*. MIT Press.

Wang, C.; Joshi, S.; and Khardon, R. 2008. First order decision diagrams for relational mdps. *JAIR* 31:431–472.

Yoon, S.; Fern, A.; and Givan, R. 2002. Inductive policy selection for first-order mdps. In *Proceedings of UAI*.