Refining Diagnostic POMDPs with User Feedback

Omar Zia Khan and Pascal Poupart
Cheriton School of Computer Science, University of Waterloo
{ozkhan, ppoupart}@cs.uwaterloo.ca

John Mark Agosta
Intel Research, Intel Corporation
john.m.agosta@intel.com

Abstract

Bayesian networks have been widely used for diagnostics. These models can be extended to POMDPs to select the best action. This allows modeling partial observability due to causes and the utility of executing various tests. We describe the problem of refining diagnostic POMDPs when user feedback is available. We propose utilizing user feedback to pose constraints on the model, *i.e.*, the transition, observation and reward functions. These constraints can then be used to efficiently learn the POMDP model and incorporate expert knowledge about the problem.

Introduction

Probabilistic models have been widely proposed as diagnostic methods for trouble-shooting and repair (Breese and Heckerman 1996; Agosta and Gardos 2004). In such models, Bayesian networks are described in terms of causes and evidence. The objective is to infer the cause given the evidence received by executing diagnostic tests. The process starts by entering observations of symptoms, then applying probabilistic inference to infer the most likely causes given the current symptoms, and then ranking tests on expected value of information or expected gain in entropy. A major issue in deploying such models is to design an accurate Bayesian network, *i.e.*, specifying accurate conditional probability tables. Current approaches include learning the model from historical data, however, sufficient data is not always available so the resulting model is often inaccurate. An alternative is to adopt a 'learning while executing' approach (Agosta, Gardos, and Druzdel 2008) so that experts can specify new components while using the existing model for diagnosis.

In this paper, we propose an extension to this approach to incorporate user feedback. Feedback is available when an expert does not change the model but acts according to or against the recommendations of the model. We plan to use POMDPs that naturally provide a recommendation of the diagnostic action (Littman et al. 2004; Joshi et al. 2005). Further, the POMDP considers the costs of various actions, where the cost represents the time taken to execute a test, and can vary according to the test. In this paper, we propose utilizing user feedback to learn and refine POMDPs.

ICAPS 2010 POMDP Practitioners Workshop, May 12, 2010, Toronto, Canada.

Diagnostic POMDP Model

The diagnostic POMDP model can be described as follows. The hidden states of the model comprise the set of root causes. The observable states are the symptoms and the evidence from tests. The actions are running tests and repair jobs. The objective is to maximize the running time of the machine, which can be computed through Time Between Repairs (TBR) and Time To Repair (TTR). TBR is the duration for which the machine remains online before the next repair occurs. TTR is the time required in corrective repair to diagnose and repair the component. So the objective function is a trade-off between the TBR and TTR. Decreasing TTR such that it results in lower TBR is undesirable, and vice versa. The transition and observation functions can be learnt from data using logging tools that record historical information about results of different tests given various symptoms. Similarly, TTR can be learnt from data using tools that record the time taken to run various tests whereas TBR can be learnt from predictive maintenance techniques that determine when a tool may be close to breaking down and can be pre-emptively repaired. Information about rare cases may not be available from data, so expert knowledge may also be needed to specify the transition, observation and reward functions. The diagnostic problem is an indefinite horizon problem since all tools are subject to predictive maintenance, which returns each machine to a good default state at regular intervals.

Problem Formulation

Developing the model for a POMDP involves identifying the states and actions, and then specifying the transition, reward and observation functions. We propose extending the approach of building the model while using it (Agosta, Gardos, and Druzdel 2008) from Bayesian networks to POMDPs. At any given point, the model can present the most likely cause and the best action to pursue, given its current belief. If a recommendation from the model is followed, we consider this positive feedback. If the user chooses a different action we consider this negative feedback. Thus, the feedback is available implicitly, without requiring separate interaction with the user.

The positive feedback indicates that the expert agreed with the output of the model. Such feedback should be

stored so that the model does not act differently in the future. The negative feedback indicates an inaccurate or incomplete model. In either case, the desirable behavior is for the model to conform with the feedback in future. For an inaccurate model, its parameters need to be updated, *i.e.*, the values of the transition, reward or observation function need to be adapted. An inaccurate transition function can lead to an incorrect ranking of actions, an inaccurate observation function can lead to an incorrect belief about the current state which influenced the ranking of the actions, and an inaccurate reward function could mislead the system about the possible costs (rewards) that influenced the ranking of actions. It is possible that all these distributions need to be updated simultaneously. However, a more complicated case is that the model is incomplete, *i.e.*, the structure of the model needs to be updated. This requires adding new relationships between existing concepts, *i.e.*, adding new links in the transition, observation or reward functions to create new dependencies. If additional feedback can be elicited from the user, it can can help determine whether the parameters or the structure need to be updated. However, in most cases, such feedback may not be available, so the system should be able to consider both types of updates.

The feedback provides preference of one action over another, so it can be represented as ordering constraints. For positive feedback this means the value of the recommended action is constrained to be higher than the value of all other actions. For negative feedback this means the value of the recommended action is constrained to be lower than the value of the chosen action. Using these constraints, the first option should be to update the parameters of the model. The update to the parameters needs to be evaluated through a metric, such as KL-divergence. If the updated model is significantly different from the existing model then modifications to the structure also need to be considered, using a metric such as Bayesian Information Criterion (BIC). We assume the feedback is provided by an expert is correct., but it can be conflicting, so in such a case the minimum number of constraints should be violated. If the conflicting constraints can be identified, the expert can then review them to correct mistakes. Such a refinement procedure would speed up the process of designing POMDPs.

Related Work

Refining a model based on user feedback can be considered similar to parameter or structure learning with constraints, where the constraints are derived from user feedback. Parameter learning is usually performed through maximum likelihood using algorithms such as EM. An alternative is to adopt a Bayesian approach in which a prior is maintained over all possible values of the parameters and then every observation is used to compute a posterior. After every observation, the posterior is updated and provides a revised distribution over the values of the parameters. Feelders and van der Gaag (2006) present a technique to incorporate inequality constraints in parameter learning. Niculescu *et al.* (2006), compute maximum likelihood estimates in closed form with different types of linear constraints represented on priors. Tong and Ji (2008) extend this work to allow more types of constraints on the parameters. Mao and Lebanon (2009) incorporate feedback as soft constraints on priors, thus allowing parameter learning even when the feedback is inaccurate.

In structure learning, the objective is to determine the dependencies between various nodes of the Bayesian network. Search algorithms, such as greedy search and best-first search, can be used to learn the best update. Campos *et al*., (2009) present an algorithm based on branch-and-bound search that learns the structure of a Bayesian network given a set of parameter and structure constraints. The structural constraints enforce the presence or absence of an arc and allow specifying the exact or maximum indegree of a node.

The constraints based on feedback in a MDP or POMDP are not on the parameters of a single distribution (like in Bayesian networks) but jointly on those of transition, observation or reward functions. For an MDP, the feedback can be interpreted as learning the policy for a state, so the model does not need to be updated (as the policy for this state is now known). For POMDPs the current state is unknown, so we cannot assume that the policy is learnt and the feedback has to be posed as a constraint on the model to update its parameters. Parameter learning for POMDPs requires learning the transition, reward and observation functions simultaneously. There has been work in learning reward function by observing the policy (Ng and Russell 2000) or learning the transition function using an oracle in active learning (Jaulmes, Pineau, and Precup 2005). Both these approaches assume the rest of the parameters are correctly specified. Bayesian reinforcement learning (Poupart et al. 2006) naturally optimizes the exploration-exploitation trade-off by using a POMDP to guide the process. Pavlov and Poupart (2008) extend this work to assume the presence of constraints received from prior knowledge. We plan to develop an approach to refining POMDP models by posing constraints obtained from implicit feedback available in the diagnostic process.

Discussion

We have presented the problem of refining diagnostic POMDPs. In such POMDPs, implicit feedback is available at every step. We plan to extend previous approaches on refining Bayesian networks to refine POMDPs by incorporating constraints received from feedback. Refining such POMDPs by utilizing this feedback can provide several advantages. First, it can lead to the development of more accurate models since it provides an opportunity to learn from domain experts. Second, it can lead to faster development of models as the domain expert does not have to explicitly specify all the changes to the model. Third, it allows efficient use of the domain expert's time by learning in the same amount of interaction. Most importantly, if the system can update the parameters and structure of the model automatically, the domain expert can concentrate on specifying domain knowledge without worrying about the details of decision-theoretic approaches.

References

Agosta, J. M., and Gardos, T. 2004. Bayes network "Smart Diagnostics". *Intel Technology Journal "Toward The Proactive Enterprise"* 8(4):361–372.

Agosta, J. M.; Gardos, T.; and Druzdel, M. J. 2008. Query-based diagnostics. In *The Fourth European Workshop on Probabilistic Graphical Models*.

Breese, J. S., and Heckerman, D. 1996. Decision-theoretic troubleshooting: A framework for repair and experiment. In *Proceedings of Twelfth Conference on Uncertainty in Artificial Intelligence*, 124–132.

de Campos, C. P.; Zeng, Z.; and Ji, Q. 2009. Structure learning of Bayesian networks using constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 113–120.

Feelders, A., and van der Gaag, L. C. 2006. Learning Bayesian network parameters under order constraints. *International Journal of Approximate Reasoning* 42:37–53.

Jaulmes, R.; Pineau, J.; and Precup, D. 2005. Active learning in partially observable Markov decision processes. In *16th European Conference on Machine Learning*.

Joshi, K. R.; Sanders, W. H.; Hiltunen, M. A.; and Schlichting, R. D. 2005. Automatic model-driven recovery in distributed systems. In *Proceedings of the 24th IEEE Symposium on Reliable Distributed Systems*, 25–38. Washington, DC, USA: IEEE Computer Society.

Littman, M. L.; Ravi, N.; Fenson, E.; and Howard, R. 2004. An instance-based state representation for network repair. In *Proceedings of the 19th National Conference on Artifical Intelligence*, 287–292. AAAI Press / The MIT Press.

Mao, Y., and Lebanon., G. 2009. Domain knowledge uncertainty and probabilistic parameter constraints. In *25th Conference on Uncertainty in Artificial Intelligence (UAI)*.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-00)*, 663–670. Stanford, CA, USA: Morgan Kaufmann Publishers Inc.

Niculescu, R. S.; Mitchell, T. M.; and Rao, R. B. 2006. Bayesian network learning with parameter constraints. *Journal of Machine Learning Research* 7:1357–1383.

Pavlov, M., and Poupart, P. 2008. Towards global reinforcement learning. In *NIPS Workshop on Model Uncertainty and Risk in Reinforcement Learning*.

Poupart, P.; Vlassis, N.; Hoey, J.; and Regan, K. 2006. An analytic solution to discrete Bayesian reinforcement learning. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, 697–704. New York, NY, USA: ACM.

Tong, Y., and Ji, Q. 2008. Learning Bayesian networks with qualitative constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.