Using a Bayesian Knowledge Base for Hint Selection on Domain Specific Problems

John C. Stamper[1], Tiffany Barnes[2], and Marvin Croy[3]
[1] john@stamper.org, [2] tiffany.barnes@gmail.com, [3] mjcroy@uncc.edu
[1]Human Computer Interaction Institute, Carnegie Mellon University
[2]Department of Computer Science, University of North Carolina at Charlotte
[3]Department of Philosophy, University of North Carolina at Charlotte

Abstract. A Bayesian Knowledge Base is a generalization of traditional Bayesian Networks where nodes or groups of nodes have independence. In this paper we describe a method of generating a Bayesian Knowledge Base from a corpus of student problem attempt data in order to automatically generate hints for new students. We further show that using problem attempt data from systems used to teach propositional logic we could successfully use the created Bayesian Knowledge Base to solve other problems. Finally, we compare this method to our previous work using Markov Decision Processes to generate hints.

1 Introduction and Method

One goal of our research has been to provide help and feedback to students with as little human intervention as possible. Our hint generation methods using MDPs have been successful in fulfilling this goal [1], but are limited to individual problems with previously collected data. We believe that extracting information from multiple problem MDPs and create a corpus of knowledge that can be used to solve new problems would inherently incorporate student preferences and bias hints toward the rules that students actually use. To accomplish this solution we extracted individual model components (MCs) from each of the MDPs to create a large Bayesian Knowledge Base (BKB) which can then be applied to any problem in the domain. A MC represents a piece of knowledge in the BKB, and in our case this will contain all or part of a student step. A BKB is a generalization of a BN where each node has independence from the others [3]. In our implementation each node in the BKB represents a group of individual model components where each component consists of a state, actions, and new states. We know that our data contains certain patterns that we can exploit. We would expect that some level of background knowledge will be needed in order to generalize overall knowledge in a domain. The logic domain is a good candidate for study since there are actual rules that we can discover. In fact, logic proof solvers already exist [2], but it is difficult for a solver to use the work that a student has already done as a basis for the remainder of a proof. Our goal is to determine hints that would lead students from where they are to a valid proof solution, and also to show the potential uses of such hints in other domains.

Model components (MCs) in the BKB can be at various levels of granularity. We have identified three levels from which the data from the MDP can be transformed. Level 3 MCs are simply all the state, actions, new state pairs extracted from the MDP. Level 2 MCs contain only one action, but can contain multiple new states. Level 1 MCs contain the smallest knowledge component that can be extracted and consist of the specific part of a state used by an action, the action, and the resulting part of the new state. In practice in the logic domain, a level 1 MC represents a logic rule application. Level 3 MCs contain the most context specific information, but will be the hardest to match. For our experiments we use only the level 1 MCs. The level 1 MCs have only have the original state information needed for the specific action, and only the new state information that is derived from this action. Again, this information is normalized such that any combination of letters in a problem can be matched to the state features.

2 Experiment and Conclusion

We create a BKB from 12 problems in our logic data sets and verify that the level 1 MCs correspond to valid rules in the logic domain. These MCs can be combined to make a general rule set that can be used to solve a new proof problem. We show this by deriving a BKB from a set of problems and testing the MCs on a new problem to see if we could select rules in such a way that the problem can be solved. This is done with a leave-one-out cross validation where all but one problem is used to make the BKB and then the remaining problem is the test case for the BKB. To provide hints with the BKB method, the current state in the new problem will be selected against the BKB and all matching model components will be returned. Based on the specificity of the states and the overall value of the returned items a hint can be given using the model component that is selected as the best for that step. To test if the BKB method is able to give hints we used all the MCs from other problems to solve the current problem. For each of the problems the BKB could solve the new problem with the exception of one problem. This is the only problem in the set that requires the “Equivalence” rule in order to reach a solution. This shows that we would be able to give hints using the BKB with most new problems unless they contained rules that had not previously occurred in a problem data set.

The primary findings of this research suggest that we can generalize the MDP method into a Bayesian Knowledge Base, which contain MCs that can be used to solve new problems. The structure of the MCs can be stated in an “if-then” format that is very similar to the production rules used in the cognitive tutors. This is extremely encouraging since the ability to automate some or all production rules for a cognitive tutor would save a tremendous amount of time in tutor development.

References

[1] Barnes, T., Stamper, J., Croy, M., Lehman, L. (2008). A Pilot Study on Logic Proof Tutoring Using Hints Generated from Historical Student Data. *In Baker, Barnes, Beck (Eds.) Procs of the 1st Intl Conf on Educational Data Mining* (EDM 2008), pp. 197-201.

[2] McCune, W. and Shumsky O., "Ivy: A Preprocessor and Proof Checker for First-order Logic". Chapter 16 in Computer-Aided Reasoning: ACL2 Case Studies (ed. M. Kaufmann, P. Manolios, and J Moore), Kluwer Academic Publishers, 2000.

[3] Santos, E., Jr., & Santos, E. (1996) Bayesian Knowledge-bases. *Technical report AFIT/EN/TR96-05*, Department of Electrical and Computer Engineering, Air Force Institute of Technology.