The Bayesian Advisor Project

Alex Dekhytar and Judy Goldsmith
Computer Science Dept.
University of Kentucky
Lexington, KY 40506-0046
{dekhtyar,goldsmit}@cs.uky.edu
phone: 859-257-{1839, 4245}, fax: 859-323-1971

March 15, 2002

Abstract

Once a student chooses courses for an upcoming semester, a good advisor could give stochastic predictions for the student's upcoming grades based on their transcript to date, though sharing such information with the student would be ill-advised. Because of this stochastic predictability, however, academic advising provides a rich domain for investigating planning with constraints over stochastic systems. An academic program could be modeled with a Bayes net. For reasons of confidentiality and of scale, such modeling should be automated to the greatest degree possible.

The Bayesian Advisor Project will produce a variety of tools to automate and support building stochastic models, and will design and test planning methods.

1 INTRODUCTION

Academic advising is a required and often under-rewarded part of many academics' job descriptions. An advisor's job has two parts: knowing the course offerings and the requirements relevant to the students' academic careers, and interacting with individual students. Both are complicated, dynamic, and time-consuming.

Ideally, the advisor bases her advice on her accurate knowledge of the student's interests and abilities. With a student to faculty ratio of approximately 40:1,[1] the ideal of personal attention to each student's needs and desires is almost inevitably compromised. Too much of an advising session is spent on the advisor reading the student's transcript, which provides the advisor with the bulk of the information about student's performance. The advisor then looks for unmet requirements, advises the student to meet those requirements in a presumably optimal manner that enables the student to move toward her goal of completing the degree with reasonable GPA in reasonable time. The key to good advising is the advisor's ability to *accurately assess the student's abilities.*

Such assesement is bound to be uncertain. A student's true abilities can be considered *hidden variables* in the advising process. The only indication of student abilities that the most advisors get is a student transcript and supplementary materials such as SAT, ACT, GRE or TOEFL scores, all of which can be considered *observable variables.* The goal of advisor, thus, is twofold: (i) to predict the advisee's abilities (hidden variables) based on past performance (observable variables); (ii) based on these predictions and the student's priorities, to find a satisfactory recommendation.

The Bayesian Advisor Project models student success on a course-by-course basis, using an extended form of Bayes nets [19]. A Markov decision process models a controlled stochastic system. Because the advisor system is explicitly factored, Bayes nets seem the obvious representation. The project's goals are:

- to automate the decision-making and constraint checking components of academic advising, so human advisors may focus on personal interaction;
- to automate Bayes net-based model construction;
- to provide a testbed for studying planning with Bayes nets under constraints;
- to provide a testbed for studying the management and fusion of probabilistic information.

The project will produce general-purpose tools (described in later sections) for modeling and planning with Bayes nets. These will be made available to researchers for other applications.

The Bayesian Advisor Project has two main stages: the model-building stage and the planning stage.

[1] as is the case at the Computer Science Department at the University of Kentucky

Model-Building stage: Before advising occurs, a model of the academic program must be created and tested. While this process needs to be driven by human experts, there are good reasons to automate significant portions of it. We are building software tools to

- create the structure of the Bayes net for the academic program;
- obtain conditional probability tables (CPTs) from existing data (anonymous student transcripts) and experts;
- store, manage and query obtained probability distributions;
- combine probability distribution functions for the same random variables, obtained from different sources (fusion of probabilistic information).

Planning stage: The Bayesian Advisor will generate advice using a planning algorithm that takes as input (i) a Bayes net model of the academic program, (ii) the advisee's transcript, and (iii) the advisee's preferences and priorities for their academic experience. The proposed plan should (preferably) optimize the expected probability of advisee's success in the courses taken, while taking into account advisee's preferences and the requirements of the academic program.

Planning for stochastic systems is intractable in all but the simplest of situations [18, 12, 17]. The compactness of the Bayes net representation intensifies this problem. The addition of constraints to the problem may also add to the complexity [7], although some constraints may be used to narrow down choices and simplify the problem.

The options considered at this point for planning include straightforward planning with constraints, and an interleaving of planning and preference elicitation or constraint development. A software tool for eliciting advisee preferences and translating them into constraints of the planning framework is under construction.

The rest of the paper is organized as follows. We describe in more detail how we are modeling academic advising as a problem of planning with constraints on Bayes nets in Section 2. In Section 3 we describe in detail the architecure and different components of the Bayesian Advisor Project.

2 ADVISING AS A BAYES NET

A Bayes net [19] is a graphical model of the relationships between different random variables in a system. The network itself is a directed acyclic graph where nodes represent known random variables and edges represent conditional dependencies between different random variables. Associated with each node of the network is a conditional probability table (CPT) describing the probability distribution of the values at the node for each possible combination of values of its parents.

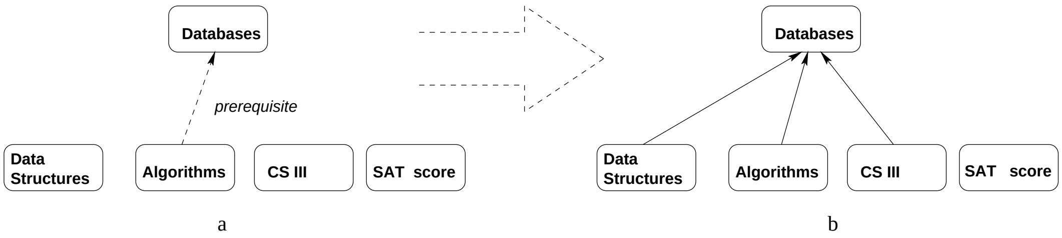

Figure 1: From stand-alone random variables and course prerequisite information (left) to building Advisor Bayes Net (right).

When building a Bayes net model of an academic program, we must identify our nodes (random variables), edges (conditional dependencies) and build the appropriate CPTs. In order to do this in the context of academic advising, we make two key assumptions about it:

- student performance in future classes *chiefly depends* on particular knowledge and abilities possessed by a student, and
- *observable variables* available to us, such as student transcript and test score information allow fairly accurate prediction of these abilities (*hidden variables*).

Random Variables

An academic program consists of a collection of courses offered at a single university and rules governing student advancement to a degree. Academic programs considered here can range in scope from small (single department or single major within a department) to medium (college) to large (university-wide curriculum). Student progress in the program is a process of taking courses and receiving grades in them. This progress represents the observable part of our model.

With each course in the curriculum we associate a *unique random variable* representing student performance in this course. Overall predictors of student successs such as current grade point average (GPA), High School GPA, as well as exam scores (SAT, ACT, GRE, TOEFL, etc.) are supplemental information. Figure 1.a shows some possible random variables for a model of our Computer Science major.

When a correspondence between some *observable* and *hidden* variables in a model is assumed to be well established, these hidden variables may be added to the network explicitly. Thus, in some cases, the academic program model may have nodes for certain skills/abilities, such as mathematical maturity, analytic ability, etc.

Random variable domains (state spaces). Random variables representing student performance in university courses have values that are specified by the

University grading policies. E.g., at the University of Kentucky, a grade in the course is one the following: A, B, C, D, E (failure), W (withdrawal), I (incomplete). In addition, for specifying conditional probabilities we add the value NT (not taken), to the list.

Other random variables, such as test scores, may have large domains, but those can be partitioned into domains of manageable size. For instance, test score domains could be: top 5%, top 10%, top 25%, top 50%, and bottom 50%.

Structure of the model

The structure of the Bayes net model of an academic program represents the dependecies between different variables present in the model. The rules for adding edges to the Bayes net model with only observable variables are:

- If it is believed that the knowledge of the material in class X and/or student's performance in class X shall affect student's performance in course Y, then the edge from node X to node Y is included in the network;
- If it is believed that student's parformance at a particular activity X represented as *supplemental information* shall affect student's performance in course Y, the edge from node X to node Y in included in the network.

Building The Model

The procedure for determining the structure of the Bayes net model of an academic program depends on the scope of the particular model desired. As mentioned above, within the university hiererchy a number of different scopes can be identified: a single departmental degree program (e.g., M.S. in Computer Science); all programs offered at a single department; all programs offered by a university college; the full university curriculum.

We expect that the difference in the size of the Advisor Bayes Net between the smallest and the largest scope within a typical public university is around two orders of magnitude[2].

For an Advisor model of a single academic program or a single department, a rough estimate of the structure of the Bayes net can be read off of the prerequisite structure for the courses (see Figure 1.a). However, this does not reflect the full information about a student's performance in a given course. Prerequisites, usually decided by departmental committees, tend to reflect the necessity of allowing students to graduate in some small number of semesters (8–12, perhaps). Thus, courses that teach skills or approaches useful for some later courses are often not explicit prerequisites.

Therefore, the Bayes net structure used in a Bayesian advisor model will be a modification of the prerequisite structure, based on information *elicited from experts* (see Figure 1.b for an example of such modification). The experts could

[2]For example, University of Kentucky Department of Computer Science has roughly 60 courses on the books, while the University catalog contains about 3000.

be academic advisors, in the case of a reasonably small program, or may need to be professors who are familiar with each individual course.

Conditional probabilities

There are two potential sources of the conditional probabilties:

- statistical information extracted from a(n anonymous) student transcript database;
- subjective probabilty distributions supplied by experts (faculty).

There are two issues for knowledge discovery: confidentiality, and fluidity of courses. The confidentiality concern arises in accessing data from student records. Theoretically, only statistical information is extracted from the transcript database. In fact, information extraction can be made only using *aggregate database queries* to the transcript database. In practice, however, certain steps need to be taken in order to ensure that personal student information is not misused[3].

Although it is difficult to officially revise a curriculum, it is the case in many departments that courses evolve[4] over time, and from professor to professor. The latter can be addressed by eliciting probabilities from each individual professor, and by distinguishing in the knowledge discovery phase which professor taught each section of each course considered. Storage of context-dependent probabilistic data is addressed in Section 4.

The problem of courses, curricula, and requirements evolving over time is addressed by considering only relatively recent data, and by periodic updates of the dependencies, probabilities, constraints, and goal states of the system. The problem of student preferences changing over time must be addressed by regular revisions of each student's utility function.

3 THE BAYESIAN ADVISING TOOL

3.1 OVERVIEW OF THE ARCHITECTURE OF THE PROJECT

Figure 2 presents an overview of the architecture of the Bayesian Advisor Project. We begin with an overview of the project, and then focus on particular aspects and tools in subsequent sections.

The architecture of the project reflects two main stages: model-building and planning.

The focal point of the project, its main clearinghouse, is the Advisor Tool, its components supporting the advising application and the flow of information

[3] In order to respect student privacy, we have arranged with the University of Kentucky registrar's office to obtain labeled student data without student IDs or demographic data (gender, race, age, etc.)

[4] Kentucky law no longer requires that we say, "change with time" instead of "evolve".

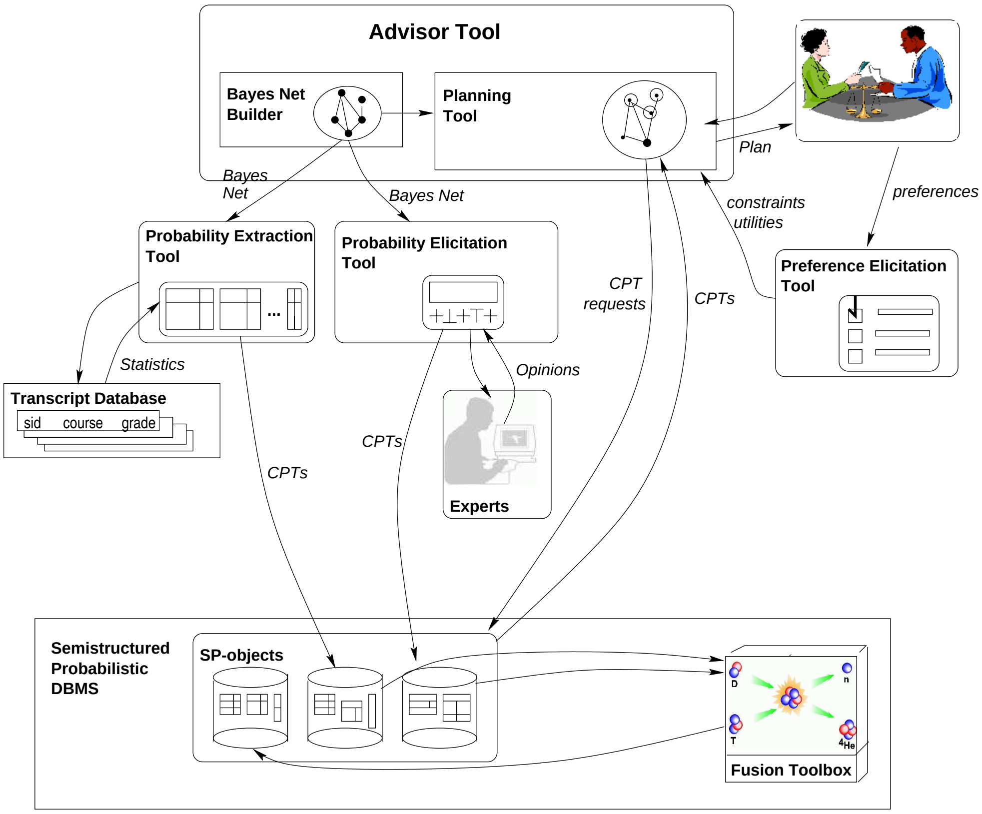

Figure 2: Architecture of the Bayesian Advisor Project

within the architecture. To build a curriculum Bayes net, the Bayes Net Builder component constructs the Bayes net structure; the actual advice comes from its Planning Tool component.

Model-Building Stage.

The model-building stage uses a variety of tools and sources to construct an academic program (curriculum) Bayes net, suitable for use on the planning stage of the project. The steps in construction are briefly described below:

- Building the Bayes Net Structure. The Bayes Net Builder component of the Advisor Tool is used to allow users to describe the graph structure of the network.

- Obtaining Conditional Probability Tables. There are two sources of information about the CPTs for the curriculum Bayes net: experts (faculty) and anonymized student transcripts. These two tasks are handled independently and in parallel by the Probability Extraction Tool and the Probability Elicitation Tool. Both tools recieve from the Bayes Net Builder the description of the structure of the curriculum Bayes net, and use that information to, respectively, obtain the statistical information about student performance in classes and solicit opinions about it from the faculty experts.

- Storing and Maintaining CPTs. CPTs created by both the Probability Elicitation and Probability Extraction Tool are passed to and stored in a Semistructured Probabilistic Database Management System (SP-DBMS), which provides the mechanisms for storing and querying diverse probability distributions.

- Combining CPTs. CPTs for the same nodes elicited from experts and extracted from the transcript data may be quite different. However, to finalize the curriculum Bayes net, a single CPT for each node needs to be presented to the Advisor Tool. This can be achieved by combining the CPTs using the methods from the Fusion Toolbox, which is under development as a part of SP-DBMS.

Planning Stage.
Once the curriculum Bayes net is prepared, the actual advising can take place. As mentioned above, advising is viewed in our model as planning on Bayes nets in the presence of constraints. These constraints represent the requirements of the advisee's degree program and logistical scheduling constraints, and may include personal constraints based on a student's preferences. The advising process will proceed as follows.

- Preference Elicitation. Advisee's preferences and priorities are elicited by the Preference Elicitation Tool and translated into a utility function suitable for use in the decision-theoretic planning stage.

- Planning. The actual planning is done by the Planning Tool component of the Advisor Tool. While a variety of planning algorithms can be implemented and tested within the project, they all have the same task. They will take as input (i) information about the curriculum Bayes net structure and CPTs, (ii) the advisee's utility function, (iii) constraints representing degree requirements and logistics, and (iv) the advisee's current transcript, and produce as output a suggested plan of action. The output can be either a set of courses for the next semester or an extended program of courses leading to graduation. It is expected, however, that the student re-plan after each semester, to take into account progress so far and any changes in preferences and priorities.

In the sections below we describe the problems involved in implementing specific parts of the Bayesian Advisor Project in more detail.

3.2 CONSTRUCTING BAYES NETS

The process of designing the structure of the Bayes net is facilitated by the Bayes Net Builder component of the Advisor Tool. This component allows the administrator of the Advisor project to enter the graphical model. When completed, the model is stored in an XML format that had been derived from the popular XMLBIF format [6] for representing information about Bayes nets. The modification was necessary in order to store dependency information separately from the CPTs.

3.3 Obtaining Probability Distributions

Once the structure of the curriculum Bayes net is finalized, we must obtain the CPTs for it. In doing so we use two sources of information: statistical information from (anonymous) student transcript database and expert opinion.

The Extraction Tool parses the structure of the curriculum Bayes net passed to it by the Bayes Net Builder and constructs a sequence of queries to the transcript database to collect the necessary statistical information. For example, for a Databases node from Figure 1.b, the following will be among the queries sent to the transcript database: "What is the distribution of grades in Databases for students who received A in Data Structures, B in Algrithms and B in CC III ?" Similar queries will be sent for each combination of grades for the parent classes.

While naive implementations of the extraction mechanism satisfy the goals of the Advisor Project (as far as delievering the statistical information is concerned), the number quries sent to the transcript database is going to be large enough to warrant the study of the possible query optimization. Research in this area is currently underway.

Eliciting probabilities from experts has always been part of the stochastic modeling repetoire. While there are a variety of techniques for doing so by hand or via online tools (see, for instance, [21, 22, 16, 24]), our application demands more than any of these methods delivers.

The first difficulty in building an advisor network is the sheer volume of probabilities needed. There are three to seven—or more—values for each variable (from Pass/Fail/NotTaken to A/B/C/D/E/NT/W, plus possible information about time elapsed since the course). Each course might depend on zero to (optimistically only) three others. Furthermore, a department might offer 30 courses, a university almost 3000.

The second difficulty is the wide variety of experts to be consulted. In order to model an entire university curriculum, potentially all the full-time faculty must be consulted. This immediately rules out the possibility of individual personal elicitations.

Thus, we need a web-based tool that allows a user to input many probabilities quickly; to stop and restart an elicitation session without penalty; to notice and bring to a user's attention any inconsistencies arising from tiredness or interruptions. It must also be as user-friendly as possible, since we will be asking for input from individuals with relatively little vested interest in this project.

Our Probability Elicitation Tool utilizes a variety of techniques to achieve the goals stated above. These techniques include simple ones such as input of patterns, copying distributions from one set of priors to another, and graphic

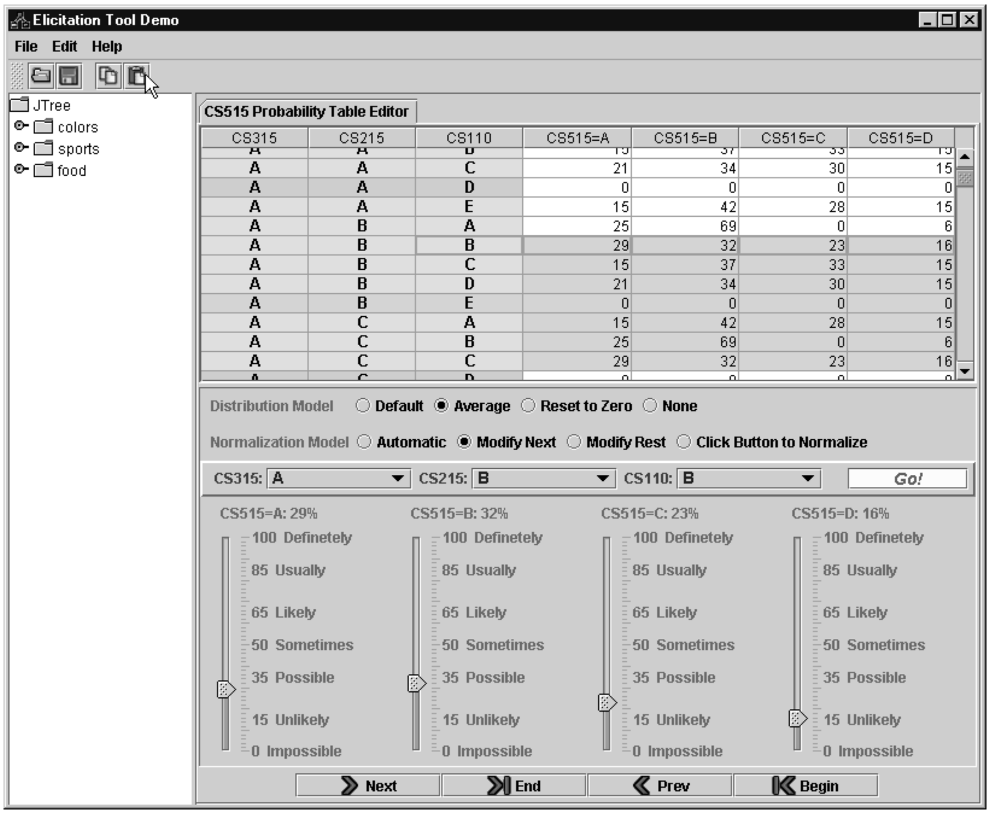

Figure 3: Screen shot of Probability Elicitation Tool

and verbal cues. We are in the process of testing the tool and hope to report on the results soon. A sample screenshot, illustrating the user interface of the tool is shown in Figure 3.

4 Semistructured Probabilistic Databases

Typically, in Bayes net applications, CPTs are stored together with the network structure in a specially defined format such as XMLBIF [6] determined by the inference software to be used.

We have chosen, instead, to store CPTs in a new kind of database designed for such data. Our decision is based on the following:

- the large number of CPTs to be stored;
- the need for *fusion* of collected CPTs suggests the need for intermediate (prior to association with the Bayes net) storage of CPTs;

- the different sources of CPTs and need for a single format;

Another important issue became apparent to us as we were investigating advising. In a variety of situations, the probability distribution of student performance in a course is affected by some non-stochastic variables related to the student (or to the course), such as student's major, course instructor, student status (graduate/undergraduate) etc. E.g., Computer Science majors are expected to do somewhat better in undergraduate Databases class than other College of Engineering majors. Standard Bayes nets do not provide immediate means for modeling non-stochastic variables as all nodes in a network represent random variables. We purposefully refuse associate probability distributions with student major or other similar parameters. In our model, we call such parameters context; we store probability distributions for each context that needs to be considered. This leads to even further growth of the number of CPTs to be stored.

Database models for probabilistic informations have been a subject of research in the past 15 years [4, 1, 10, 15], but with the notable exception of the ProbView system [15] no probabilistic DBMS has been implemented. In [8] we have proposed a new, flexible framework for storing and querying information about diverse probability distributions called Semistructured Probabilistic Databases. A unit of information in this framework is a Semistructured Probabilistic Object (SPO), which stores information about one probability distribution of one or more random variables together with applicable context and conditional information. A sample SPO (together with the description of the probability distribution it represents) is illustrated in Figure 4. A Semistructured Probabilistic query algebra (SP-Algrebra) that allows to query the diverse content of SPOs had also been proposed in [8]. A prototype SP-DBMS has been developed and is currently undergoing testing.

The semistructured nature of the SP-Database model allows us to conveniently represent SPOs in XML format [13]. The Probability Extraction and Elicitation Tools convert the probability tables obtained from data or experts into SPOs in this format and pass it to the SP-DBMS server which stores the information.

The structure of a sample semistructured probabilistic object (SPO) and the corresponding English description of the probability distribution stored in it are shown in Figure 4.

4.1 Fusion of Conditional Probabilities

Because courses change, both from professor to professor and over time, data on historic student performance in a course may not accurately reflect future performance. As described in Section 3.3, we will be collecting conditional probabilities directly from professors as well as from anonymized transcript data.

It is expected that the probability tables for the same nodes in the curriculum Bayes net obtained from different sources will differ, sometimes significantly. Some of the difference may arise from human imprecision, some – from

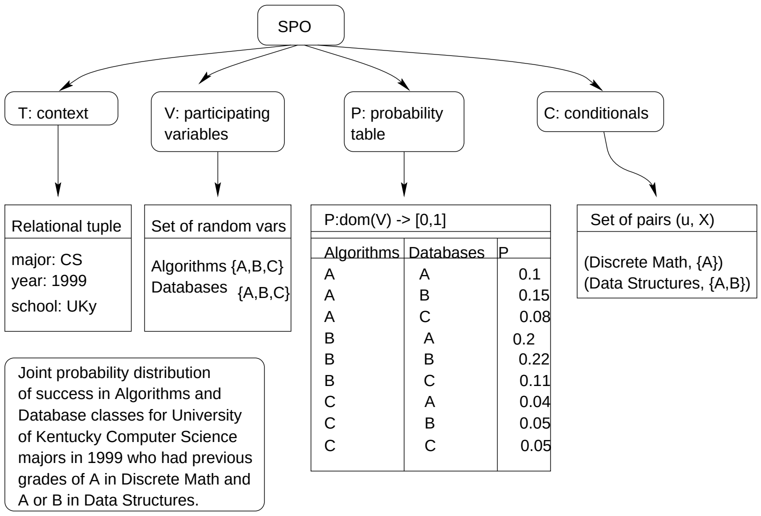

Figure 4: Structure of a Semistructured Probabilistic Object according to [8].

differences in the underlying models of the courses and grading schemes under consideration. Yet these differing probability tables must be combined to form one useable CPT for the Bayes net.

There are two questions that a probability fusion toolbox must address:

- When is fusion necessary?
- How should necessary fusion be done?

Some of the decision-making about fusion is taken care of in the extended Bayes net model by context: if a student is considering a course by Prof. Goldsmith, and the probabilities elicited from Prof. Goldsmith are stored with the context variable Professor=Goldsmith, then other probability sources might be ignored.

It is also possible that fusion can be avoided because the differences between the probability tables under consideration will not affect the outcome of the planning algorithm. Unfortunately, the problem of recognizing this case is at least NP-hard [9].

There has been significant work on online adjustments of probability fusions in the vision community. Unfortunately, that work does not help us; the Bayes net must be stable in order to apply any of the current planning methods. This is, however, a possible area of investigation: planning with Bayes nets with variability.

Most probability fusion techniques boil down to taking weighted averages of the probabilities from different sources. The choice of weights might boil down to the size of the sample spaces the experts draw from [20], or to the reliability of the experts. It is not clear how these factors will or should be computed in this setting.

The Fusion Toolbox developed in this project will be a part of the underlying SP-DBMS. We expect to implement a variety of fusion techniques and allow the system, in cooperation with the user decide which, if any, fusion techniques should be applied to data at hand.

5 Creating Bayesian Advisor Plans

The Bayes net model of an academic program is one of the necessary inputs to the Planning Tool, a collection of planning algorithms for generating academic advice. Other important inputs for the planning algorithms is the collection of constraints based on student's priorities and preferences and student's transcript. A Bayes net must be assembled that is appropriate for the particular advisee, using the transcript and preferences for context and conditionalization of probabilities. Then some planning mechanism must be appled.

5.1 Preference Elicitation

The subject of preference elicitation has long been of interest to people in marketing [14] and economics [23]. It has been coming to the attention of the AI community in recent years, as evidenced by the AAAI workshop on the topic this year.

There have been several approaches to this in recent AI papers. Some work on preferences has focussed on online deduction of preferences. This may not be suitable for Advisor Project; by the time a student has fully developed preferences about courses and academic priorities, she's usually ready to graduate.

Other work has suggested interleaving planning and preference elicitation. (See [2, 5], for instance). This is certainly a possibility in this setting, although it might rule out using off-the-shelf Bayes net planning systems. The major constraint here is the advisee's attention span. The underlying planning problem, solving a compactly represented Markov decision process with constraints, is NP-hard. Will a student be willing to spend sufficient time interacting with an online tool to make such interleaving possible? If the planning algorithm can use the constraints to *simplify* the problem by narrowing down its searches, this could prove to be practical.

There are three major questions to be answered in the design of a preference elicitation tool.

- The underlying representation of preferences
- The needs of the planning tools
- The elicitation procedure and GUI design.

Given the expected user community of 18–22 year olds, it is important that this tool have a significant and entertaining GUI. We conjecture that presenting them with scenarios to choose between will be more effective than asking them about explicit trade-offs.

5.2 Assembling a Bayes Net

As discussed in Section 3.2, the Bayes Net Builder will be able to specialize a Bayes net for a user. The inputs to the Builder will be assembled from the student's context (major, student status, or other non-stochastic attributes), the student's preferences, the courses available for the semester(s) in question and information about those course, and conditionalization (the student's grades so far). For instance, if a student refuses to take courses before 9 AM, any such courses can be discarded. Conditional probability tables can be based on the probabilities elicited from the professors scheduled to teach the individual courses.

It will be the job of the Assembly Tool to choose the relevant aspects of the student's and program's descriptions in order to find the most appropriate Bayes net for this student. This problem can best be modeled as an expert system which interacts with the input mechanism to examine the transcript, and with the Preference Elicitation Tool, and feeds its output to the Bayes Net Builder.

In Figure 2, the Assembly Tool has been folded into the Planning Tool.

5.3 Planning

The goal of a good advisor in not only to generate a set of courses for the student's next semester, but to have in mind a longer-term plan that will allow the student to satisfy all of the relevant university and departmental requirements. Furthermore, within that plan, the student should reach her goals with high utility with as high a probability as possible. This requires more than a simple recommender system; it requires full-blown decision-theoretic planning.

There are many constraints inherent in an academic setting. Students may only sign up for one course per time slot; they have prerequisites (sometimes enforced) for certain courses. Some sets of courses cannot be jointly taken for credit (e.g., Physics I and Physics for Poets). Such constraints complicate the planning process.

There is considerable work on decision-theoretic planning with Bayes nets (see [3] for a survey of work up to 1999), and some initial work on combining such planning with constraints ([7], for instance). As mentioned above, there are clear trade-offs between separating the preference elicitation phase from the planning phase and combining the two phases. Planning/elicitation algorithms must be developed and tested, and the complexity of the combined problem considered.

6 Conclusions

The academic advising domain is a novel application. There have been other applications of AI to academic processes (see [11]), but not to this particular aspect of academia.

Parts of this work have been described in [8, 13, 9]. The following tools exist in prototype form: Probability Elicitation Tool, Probability Extraction Tool, and Semistructured Probabilistic Database Management System. Reports on these tools are in preparation. The Preference Elicitation Tool and Fusion Toolbox are currently under development. Planning tools and the Bayes Net Builder are in the planning stages.

ACKNOWLEDGEMENTS

We want to thank Rina Dechter for suggesting we pursue this; members of the University of Kentucky Computer Science Department who agreed to participate in the control group for our experiment, and the rest of the Bayesian Advisor group, for teaching us some of the skills we need for this project, and learning the rest with us. Finally, we thank the faculty members from other schools who have encouraged us and asked to be involved in this project, including Eastern Kentucky University; Berea College; University of Dayton; the US Naval Academy, and UC Irvine.

References

[1] D. Barbara, H. Garcia-Molina, and D. Porter. The management of probabilistic data. *IEEE Trans. on Knowledge and Data Engineering*, 4:487–502, 1992.

[2] Craig Boutilier, Ronen Brafman, Christopher Geib, and David Poole. A constraint-based approach to preference elicitation and decision making. In Jon Doyle and Richmond H. Thomason, editors, *Working Papers of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, pages 19–28, Menlo Park, California, 1997. American Association for Artificial Intelligence, American Association for Artificial Intelligence.

[3] Craig Boutilier, Thomas L. Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of AI Research*, 11:1–94, 1999.

[4] R. Cavallo and M. Pittarelli. The theory of probabilistic databases. In *Proc. VLDB'87*, pages 71–81, 1987.

[5] Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00) and of the 12th Conference on Innovative Applications of Artificial Intelligence (IAAI-00)*, pages 363–369, Menlo Park, California, July 30– 3 2000. AAAI Press.

[6] Fabio Cozman. The interchange format for Bayesian networks. `http://www-2.cs.cmu.edu/afs/cs/user/fgcozman/www/Research/InterchangeFormat`, 1998.

[7] Rina Dechter and David Larkin. Hybrid processing of beliefs and constraints. In *Proc. Uncertainty in Artificial Intelligence*, 2001.

[8] Alex Dekhtyar, Judy Goldsmith, and Sean Hawkes. Semistructured probabilistic databases. In *Proc. Statistical and Scientific Database Management Systems*, 2001.

[9] Alex Dekhtyar, Judy Goldsmith, and Jan Pearce. When probabilies distinguish Bayes nets. In *Proc. Workshop on Uncertainty in AI, KI '01*, 2001.

[10] D. Dey and S. Sarkar. A probabilistic relational model and algebra. *ACM Transactions on Database Systems*, 21(3):339–369, 1996.

[11] Kenneth D. Forbus and Paul J. Feltovich, editors. *Smart Machines in Education*. AAAI Press/ The MIT Press, 2001.

[12] Judy Goldsmith, Michael L. Littman, and Martin Mundhenk. The complexity of plan existence and evaluation in probabilistic domains. In *Proceedings of the 13th Conference on Uncertainty in Artifical Intelligence*. Morgan Kaufmann Publishers, 1997.

[13] S. Hawkes and A. Dekhtyar. Designing markup languages for probabilistic information. Technical Report 319-01, University of Kentucky Dept. of Computer Science, 2001.

[14] R. L. Keeney. *Value-Focused Thinking: A Path to Creative Decisionmaking*. Harvard University Press, Cambridge, MA, 1992.

[15] V.S. Lakshmanan, N. Leone, R. Ross, and V.S. Subrahmanian. Probview: A flexible probabilistic database system. *ACM Transactions on Database Systems*, 22(3):419–469, 1997.

[16] Kathryn Blackmond Laskey and Suzanne M. Mahoney. Network engeinnering for agile belief networks. *IEEE Transactions on Knowledge and Data Engineering*, 12(4):487–497, 2000.

[17] Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity results for finite-horizon Markov decision process problems. *Journal of the ACM*, 47(4):681–720, 2000.

[18] Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[19] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, California, 1988.

[20] Pedrito Maynard Reid II and Ursula Chajewska. Aggregating learned probabilistic beliefs. In *Proc. Uncertainty in Artificial Intelligence (UAI '01)*, number 17, pages 354–361, 2001.

[21] Silja Renooij and Cilia Witteman. Talking probabilities: communicating probabilistic information with words and numbers. *International Journal of Approximate Reasoning*, 22(3):195–215, 1999.

[22] Linda C. van der Gaag, Silja Renooij, Cilia Witteman, B.M.P Aleman, and B.G. Taal. How to elicit many probabilities. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 647–654, Stockholm, Sweden, 1999.

[23] John von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, Princeton, New Jersey, first edition, 1944.

[24] Haiquin Wang and Marek J. Druzdzel. User interface tools for navigation in conditional probability tables and elicitation of probabilities in Bayesian networks. In *Proc. 16th Conference on Uncertainty in Artificial Intelligence*, 2000.