[9] A. L. Madsen and F. V. Jensen, “Lazy evaluation of symmetric Bayesian decision problems,” in *Proc. 15th Conf. Uncertainty in Artificial Intelligence*, 1999, pp. 382–390.

[10] R. D. Shachter, “An ordered examination of influence diagrams,” *Networks*, vol. 20, no. 5, pp. 535–563, 1990.

[11] P. P. Shenoy, “Valuation network representation and solution of asymmetric decision problems,” *Eur. J. Oper. Res.*, vol. 121, no. 3, pp. 574–608, 2000.

[12] T. D. Nielsen and F. V. Jensen, “Representing and solving asymmetric Bayesian decision problems,” in *Proc. 16th Conf. Uncertainty in Artificial Intelligence*, 2000, pp. 416–425.

A Method for Evaluating Elicitation Schemes for Probabilistic Models

Haiqin Wang, Denver Dash, and Marek J. Druzdzel

***Abstract*—We present an objective approach for evaluating probability and structure elicitation methods in probabilistic models. The main idea is to use the model derived from the experts’ experience rather than the true model as the standard to compare the elicited model. We describe a general procedure by which it is possible to capture the data corresponding to the expert’s beliefs, and we present a simple experiment in which we utilize this technique to compare three methods for eliciting discrete probabilities: 1) direct numerical assessment, 2) the probability wheel, and 3) the scaled probability bar. We show that for our domain, the scaled probability bar is the most effective tool for probability elicitation.**

***Index Terms*—Bayesian network, evaluation of elicitation methods, learning, probability elicitation.**

I. Introduction

As more and more decision-analytic models are being developed to solve real problems in complex domains, extracting knowledge from experts is arising as a major obstacle in model building [1]. Quite a few methods have been proposed to elicit subjective probabilities from domain experts. These techniques balance quality of elicitation with the time required to elicit the enormous number of parameters associated with many practical models. Structure elicitation is likewise a tedious problem and formal techniques for this task are even less mature. Systematic evaluation and comparison of different model elicitation methods are thus becoming of growing concern.

In Bayesian probabilistic models, encoded probabilities reflect the degree of personal beliefs of the experts. The sole purpose of probability elicitation is to extract an accurate description of the expert’s personal beliefs. In order to judge whether the elicitation procedure has produced an accurate model, therefore, the elicitor must know intimate details about the expert’s knowledge. Unfortunately, these details that the elicitor is seeking from the start are hidden from explicit expressions; so it has not been possible to evaluate elicitation schemes directly. Less direct methods are the only possibility.

In this paper, we present an objective approach for evaluation of elicitation methods that avoids the assumptions and pitfalls of existing approaches. Our technique is much closer to the ideal “direct” comparison between the elicited network and the expert’s beliefs. The main idea is to simulate the training/learning process of an expert by allowing the trainee to interact with a virtual domain. Underlying the domain is a Bayesian network that is used to stochastically update the state of the world in response to the subject’s interaction. Then, by recording every state of the world that is experienced by the trainee, we can effectively gain direct access to the trainee’s knowledge. It is quite an established fact that people are able to learn observed frequencies with amazing precision if exposed to them for a sufficient time [2]. Therefore, after training, the trainee obtains some level of knowledge of the virtual world and, consequently, becomes an expert at a certain proficiency level. This knowledge, in the form of a database of records $D_{\exp}$, can be converted to an “expected” model of the expert $\hat{M}_{\exp}$, by applying Bayesian learning algorithms to $D_{\exp}$. Finally, this expected expert model can be directly compared to the model elicited from the expert to judge the accuracy of elicitation.

Our approach captures a subject’s state of knowledge of the probabilistic events in the toy world. The subject’s experience with the toy world, rather than the actual model underlying the world, forms the basis of his or her knowledge. For this reason, the learned model should be the standard used to evaluate the elicitation schemes, rather than the original toy model. This technique allows us to avoid the expensive process of training subjects to fully-proficient expertise. For example, our expert’s experience may have led him to explore some states of the world very infrequently. In this case, even if our elicitation procedure is perfect, the elicited probabilities of these states may be significantly different from the underlying model. Using the expert’s experience rather than the original model gets around this problem completely because we know precisely how many times our expert has visited any given state of the world.

We use these techniques along with a toy cat–mouse game to evaluate the accuracy of three methods for eliciting discrete probabilities from a fixed structure: 1) direct numerical elicitation, 2) the probability wheel [3], and 3) the scaled probability bar [4]. We use mean-squared errors (MSEs) between the learned and the elicited probabilities to evaluate the accuracy of each of the three methods. We show that, for our domain, the scaled probability bar is the most effective and least time-consuming.

We begin with a brief review of the existing evaluation techniques for probability elicitation methods. Then, we present the relevant learning equations that allow us to capture a subject’s beliefs in the form of learned network parameters. We describe the cat–mouse game that we used to train our subjects and collect data for learning. We present our experimental design and results followed by a discussion of our findings.

II. Evaluation Schemes of Probability Elicitation Methods

The difficulty in evaluating elicitation methods is that the true model is needed in order to be compared to the elicited model. Since the former is encapsulated in the expert’s mind, it is not readily available for comparison. Previous comparisons of elicitation schemes followed essentially three lines of reasoning: 1) expert’s preference, 2) benchmark model, and 3) performance measure.

The first approach, *expert’s preference*, is based on the assumption that when an elicitation method is preferred by the expert, it will

Manuscript received May 15, 2001; revised July 15, 2001, September 2, 2001, and October 15, 2001. This work was supported by the Air Force Office of Scientific Research under Grant F49620-00-1-0112 and the National Science Foundation under Faculty Early Career Development (CAREER) Grant IRI-9624629. This paper was recommended by Associate Editor E. Santos.

H. Wang and D. Dash are with the Decision Systems Laboratory, Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: whq@mail.sis.pitt.edu; ddash@mail.sis.pitt.edu).

M. Druzdzel is with the Decision Systems Laboratory, School of Information Sciences, and the Intelligence Systems Program, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: marek@mail.sis.pitt.edu).

Publisher Item Identifier S 1083-4419(02)00710-0.

yield better quality estimates. While this assumption is plausible, to our knowledge it has not been tested in practice. There are a variety of factors that can influence the preference for a method, such as its simplicity, intuitiveness, or familiarity, and these factors are not necessarily correlated with accuracy.

The second approach, *benchmark model*, compares the results of elicitation using various methods against an existing benchmark (gold standard) model $\hat{M}$ of a domain (or a correct answer that is assumed to be widely known). Accuracy is measured in terms of deviation of the elicited model from $\hat{M}$. For example, in a study of people's perception of frequencies of lethal events, there was a readily available collection of actuarial data on those events [5]. Similarly, in another study on effects of a relative-frequency elicitation question on likelihood judgment accuracy, general knowledge was used[6]. An important assumption underlying the benchmark model method is that the model $\hat{M}$ is shared by all experts. While in some domains this assumption sounds plausible, human experts notoriously disagree with each other [7], [8], and an experimenter is never sure whether the model elicited is derived from a gold standard model or some other model in the expert's mind. A debiasing training of experts with an established knowledge base may help to establish a benchmark model among them. For example, Hora *et al.* [9] trained their subjects in a formal probability elicitation process directed toward assessing the risks from nuclear power generating stations and compared two elicitation methods for continuous probability distributions. Their subjects were scientists and engineers who quite likely possessed extensive background knowledge about the risks. Effectively, it is hard in this approach to make an argument that the elicited model is close to the experts' actual knowledge, as the latter is simply unknown.

The third approach, *performance measure*, takes a pragmatic stand and compares the predictive performance of models derived using various methods. This reflects, in practice, how well calibrated the expert's knowledge is [10]. An example of this approach is the study performed by van der Gaag *et al.* [11], who used prediction accuracy to evaluate their probability elicitation method in the construction of a complex influence diagram for cancer treatment. While it is plausible that the quality of the resulting model is correlated with the accuracy of the elicitation method, this approach does not disambiguate the quality of the expert's knowledge from the quality of the elicitation scheme. A model that performs well can do so because it was based on superior expert knowledge, even if the elicitation scheme was poor. Conversely, a model that performs poorly can do so because the expert's knowledge is inferior, even if the elicitation scheme is perfect.

The next section introduces an evaluation method that we believe does not suffer from the problems identified in the existing evaluation schemes.

III. Data Mining Expert Beliefs

To evaluate the accuracy of an elicitation method is to make a judgment about how good the elicited model reflects the expert's real degree of personal belief. The closer the elicited model reflects the expert's real beliefs, the more accurate we say the method of elicitation is. But how can we measure an expert's real degree of personal belief? What can be used as a standard to evaluate the accuracy of a subjective probability? What we need is a method to capture the knowledge/beliefs that are held by our expert, then we need a method to construct a model entailed by that knowledge.

On the other hand, if we have a set of records in the form of a database, there are many machine-learning algorithms that are available to learn various types of models from that database. In this section, we will present the theory needed to learn probabilistic network models from data.

A. Capturing the Expert's Knowledge

Complicating this effort is the fact that a person becomes an expert from a novice in a process of acquiring knowledge from a wide array of sources. Sources of knowledge range from reading books, talking to other experts, and most importantly for us, to observing a series of instances in the real world. In the method that we are proposing, we *create* an expert in a particular toy domain. In the process, we confine the source of knowledge available to that expert to be strictly of the latter type; namely, a series of observations of the real world. Being assured that our expert accumulates only this knowledge allows a particularly simple analysis of what our expert's beliefs about the domain should be. Throughout the paper, we will refer to this type of knowledge as *observational knowledge*.

If we assume that we have an expert whose entire knowledge of a domain is observational, then the expert's knowledge can be viewed as originating entirely from a database, $D_{\exp}$, of records filled with instances of the domain our expert has committed to memory. If we further assume that we have recorded all relevant instances of the domain that our expert has actually observed into a database D, then our database D will be identical to $D_{\exp}$ under the assumption that the subject has paid attention to the occurrence of each event during his or her observation process. Thus, in any experiment designed to measure $D_{\exp}$, it will be important to incentivate the subject in some way to pay attention to all events in the world.

B. Learning Bayesian Networks From Data

Assuming that we can assess $D_{\exp}$ correctly, we must now construct a probabilistic model that is most consistent with that data. Much work has been done on this problem in recent years [12], [13], [14], [20]. We will present just the key results of some of this work here. A good review of the literature can be found in [15].

Bayesian methods [14] for learning a probabilistic model over a set of variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, assume that the learner begins with a set of prior beliefs governing the domain. In the case of an unrestricted multinomial distribution, each variable X_i is discrete, having r_i possible values $x_i^1, \ldots, x_i^{r_i}$, where $i = 1, \ldots, n$. In this case, it is assumed for convenience that the priors take the form of a Dirichlet distribution [16], having parameters α_{ijk}. One common sense interpretation of α_{ijk} in a Bayesian network capturing this domain is that it is the number of times an expert has observed variable $X_i = x_i^k$ when the parents of X_i achieved the jth configuration: $Pa_i = pa_i^j$. As a bit of notation, we define θ_{ijk} to be the true probability that $X_i = x_i^k$ given that $Pa_i = pa_i^j$. In other words, it is the conditional probability parameter corresponding to the α_{ijk}. We use $\boldsymbol{\theta_{ij}} = \{\theta_{ijk} | 1 <= k <= r_i\}$ to denote the conditional probability distribution of X_i under the jth parent configuration. We assume *parameter independence*, which states that $\boldsymbol{\theta_{ij}}$ is independent of $\boldsymbol{\theta_{ij'}}$ for all $j \neq j'$.

In the Bayesian approach, the data set D is considered fixed. To find a good network structure which encodes the physical joint probability distributions for $\mathbf{X}$, we need to select a network structure that has highest posterior probability $p(S|D)$. Assuming all possible structures are equally likely, $p(S|D)$ is proportional to the marginal likelihood of the data given structure $p(D|S)$. Under the assumption of complete data set D, Dirichlet prior parameters α_{ijk}, and parameter independence, the most likely structure can be selected using the following scoring metric:

$$p(D|S) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \tag{1}$$

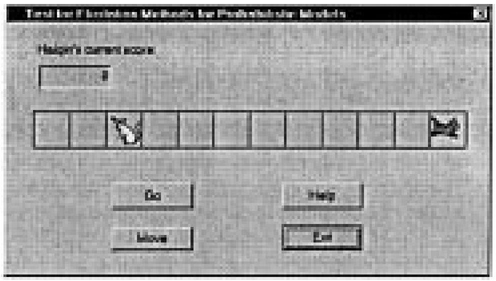

Fig. 1. Screen snapshot of the cat–mouse game.

and the expected value of the network parameters given a structure can be expressed as

$$\hat{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}. \tag{2}$$

In (1) and (2), Γ is the gamma function, N_{ijk} are the number of times in D that the variable X_i took on value x_i^k when the parents of X_i took on configuration pa_i^j, $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Equation (2) computes the probability parameters by using *maximum a posteriori* probability. The α parameters represent the experts' domain knowledge and result in a different set of probability parameter distributions from maximum likelihood parameters.

For a domain where the expert has little or no previous experience, we assume that all α_{ijk} are equal and small. Under this assumption, when no data are present for a particular (i, j) configuration of the world (i.e., $N_{ij} = 0$), then the N_{ijk} terms drop out of (2) and the small equal priors produce a uniform distribution. However, even if a small amount of data is involved, the priors have little influence on the parameters learned.

For example, assume we are estimating the probability that a given coin will come up heads on an arbitrary toss, and assume that for our subject $\alpha_{\text{heads}} = \alpha_{\text{tails}} = 0.001$. Such a low prior indicates that our subject has had very little experience with coins, but still assumes initially that the coin is equally likely to be weighted toward heads or tails. After one flip of the coin (say a "heads" outcome), our subject's estimate of $P(\text{heads}) = (1 + 0.001)/(1 + 0.002) \approx 1$, so our subject's initial belief in uniformity has quickly been affected by the data. On the other hand, if our subject's initial beliefs were $\alpha_{\text{heads}} = \alpha_{\text{tails}} = 10$, then after one flip, his or her new assessment would be $P(\text{heads}) = 11/21 \approx 0.5$, much closer to his initial estimate. Therefore, the larger the α parameters, the more weight our subject's expertise will play into his estimate of parameters.

IV. Evaluating Elicitation Schemes With a toy Virtual World

We designed a game in which a subject can move a cat to capture a mouse. We recorded the state changes of the cat–mouse game during the game playing process. What each subject experiences is unique and depends on the subject's actions. The recorded data allows for the learning of the probabilistic model of the toy world as seen by the subject. This learned model, in turn, gives us a standard by which to measure the accuracy of the model elicited from the subject.

A. The Cat and Mouse Game: A toy Virtual World

Our toy world includes three characters: a cat and two mice. The objective of the game is for the cat to capture a mouse. There are 12 possible positions indicated by the grid cells in a horizontal line (see Fig. 1). The cat can move one cell at a time between the current cell and an adjacent cell. One and only one mouse is present at any given time, and it can only bounce back and forth between two positions on each side of the screen. The two special positions for the mice are called *left-pos* and *right-pos*, respectively. When the cat enters the cell/position where the mouse is located, it catches the mouse and the game is over.

TABLE I
YELLOW MOUSE AND GREY MOUSE

yellow	*grey*

TABLE II
FOUR STATES OF THE CAT

normal	*angry*	*frustrated*	*alert*

The two mice are characterized by a color: *yellow* or *grey*. The cat can be in one of four states: *normal*, *angry*, *frustrated*, and *alert*. Four icons are used to represent the states of the cat. Tables I and II illustrate the icons we used in the game.[1]

Two buttons, labeled *move* and *go*, respectively, are provided for the subject to manipulate the position of the cat. After the subject clicks a button, the cat moves to either the left or the right. Its moving direction is uncertain and depends on the current state of the world (i.e., which mouse is present, the position of the mouse, the state of the cat, and which button the subject has clicked). There is a short delay (half a second in our experiment) between button clicks during which the buttons are disabled. This prevents the subject from clicking the buttons too frequently and paying little attention to probabilistic relationships among the variables. It allows the subject to have enough time to observe how the moving direction of the cat is influenced by the state of the world and the subject's own actions.[2]

After this delay, the toy world is updated to a new state. One mouse may disappear and another may show up instead. The mouse may appear in a different position. The cat may change its state. The two buttons for the subject's action become enabled.

In the beginning, the yellow mouse is put in the *left-pos* position. The cat is put in the farthest position away from the mouse. After the cat has caught a mouse, the game ends and a new round of the game begins. A new game always begins with the same initial positions for both the mouse and the cat, but the states of the rest of the world are uncertain.

Scoring rules are adopted to encourage the subject's involvement in the game. Whenever the cat captures a mouse, the subject's score increases as an incentive. Also, the game emits a celebratory sound as a reward for the subject.

[1] Our experimental subjects only saw the figures as the representation of the cat's states and mouse color. The verbal expressions are used to encode the cat's states and mouse color in the Bayesian network for the cat–mouse world due to the restraint of the modeling environment. These labels, "normal," "angry," etc., were not provided to the subjects during game play but were used, together with the pictures, to identify the states of the cat during the elicitation process.

[2] The delay length of the disabled state of the buttons was selected based on our experiments with pilot subjects. We first tried 1 s and 2 s as the delay, but our pilot subjects soon complained the delay was too long and made the game boring. Therefore, we selected the maximum delay (half a second) with which the subjects still felt comfortable.

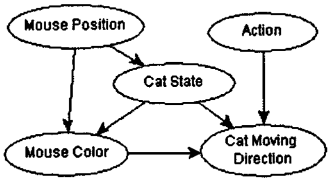

Fig. 2. Bayesian network of the cat–mouse world.

B. The Bayesian Network for the Cat–Mouse World

The cat–mouse world is based on a simple Bayesian network (see Fig. 2) consisting of five variables: *Action*, *Mouse Color*, *Mouse Position*, *Cat State*, and *Cat Moving Direction*.

Variable *Action* with two outcomes, *move* and *go*, models the observed subject's action. *Mouse Color*, which could be *yellow* and *grey*, defines which of the two mice is present. *Mouse Position* indicates the current position of the present mouse: *left-pos* and *right-pos*. *Cat State* represents four possible states of the cat: *normal*, *angry*, *frustrated*, and *alert*. The last variable, *Cat Moving Direction*, reflects the moving direction of the cat in the current step. Two directions are defined: *left* and *right*.

The five variables influence each other probabilistically. The states of the variables change at each step according to the probabilities encoded in the network. Their probability distributions, either prior or conditional, were assigned randomly when the network was built to avoid biases to a particular probability distribution. One exception is the probability distribution of the *Action* node. The value of the *Action* node is always instantiated to the state that corresponds to the subject's action, and hence, the prior probability distribution becomes irrelevant. We chose the two nearly identical action words, *move* and *go*, to avoid any semantic difference which could have a potential influence on the subjects' preference.

C. The State Change of the World by Sampling

After the subject has clicked a button to take an action, the state of the world and the cat's moving direction are updated. The new states are selected by generating a stochastic sample on the cat–mouse network following the partial parent order of the graph. We use probabilistic logic sampling [17] to generate node states on the basis of their prior probabilities of occurrence. By choosing more likely states more often, we simulate the state changes of the toy world. The subjects are exposed to changes in the world that are an effect of their actions and the underlying joint probability distribution.

D. Collecting Data for Expert's Knowledge

Every time the state of the toy world changes, it is recorded automatically. In our data set, a case consists of the outcomes of all five variables encoded in the cat–mouse Bayesian network. The database of a subject's experience contains all states of the world that the subject has seen and it is the subject's observational knowledge about the toy virtual world. This knowledge comes completely from the subject's game-playing experience. Therefore, the records constitute a perfect data set for learning the subject's knowledge about the cat–mouse domain.

We used (1) and (2) in the learning algorithm to learn the desired network from data. The assumptions required were satisfied. First, the probability distributions were unrestricted multinomial. Second, our data set was complete. Third, we assigned the probability parameters randomly in the cat–mouse Bayesian network and made them satisfy parameter independence.

V. Experimental Design

We demonstrated our method in an experimental study that investigated the effectiveness of three elicitation methods: 1) asking for numerical parameters directly, 2) translating graphical proportions by using the probability wheel, and 3) using the scaled probability bar. We used the graphical modeling system *GeNIe* [18] and built a module of cat–mouse game in *GeNIe* as well.

A. Subjects

The subjects were 28 graduate students enrolled in an introductory decision analysis course at the University of Pittsburgh. They received partial course credit for their participation.

B. Design and Procedure

The subjects were first asked to read the instructions from a help window that introduced the game characters and the game rules. They were asked to pay attention to the probabilistic influences from the state of the toy world and their action choice to the direction of the cat's movement. The subjects were told that knowledge of these probabilistic relationships would help to improve their performance. To motivate the subjects to perform well, extra credit was offered for higher scores in the cat–mouse game and lower errors of estimates of the probabilities in elicitation.

Each trial included two stages. The subjects first played the cat–mouse game for 30 min. The data about their experienced states of the toy virtual world were automatically recorded. The data sets in our experiment typically contained between 400 and 800 records.

The second stage involved probability elicitation by each of the three elicitation methods. The subjects were shown the Bayesian network structure in Fig. 2 and were asked to estimate the conditional probability table (CPT) for the node *Cat Moving Direction* by:

1) typing the numerical parameters directly in CPTs;
2) giving graphical proportions in the probability wheel; and
3) giving graphical proportions in the scaled probability bar.

We applied here a within-subject design in which each subject used the three elicitation methods. To offset the possible carry-over effects, we counterbalanced the order of method usage across our subjects.

The CPT elements θ_{ijk} elicited were compared to $\hat{\theta}_{ijk}$, the CPT elements learned by applying (2) to the subjects' acquired data. The MSE of the parameters was calculated as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\theta_{ijk} - \hat{\theta}_{ijk})^2.$$

In order to evaluate the speed of the elicitation methods, we also recorded the time taken for each elicitation procedure.

Since our domain experts had very little knowledge about the probability distributions of the cat–mouse domain prior to playing the game, we assigned small uniform values to the α parameters for the learning algorithm. For all possible values of i, j, k, we used the assignment $\alpha_{ijk} = 5$ and $\alpha_{ijk} = 10$, respectively. In order to test purely data-based learning, we also used $\alpha_{ijk} = 10^{-4}$.

C. Results

Table III shows the means and standard deviations of the MSEs of the three elicitation methods when compared to the probabilities learned with different α parameters. A time comparison is also shown as the last two lines in the table. Fig. 3 plots the elicitation time and MSE ($\alpha = 5$) for each of the three elicitation methods.

For each pair of elicitation methods, we conducted one-tailed, paired sample t test for comparison of accuracy corresponding to the learned results with different αs. Three similar t tests were also done for time

TABLE III
MEANS ($\overline{x}$) AND STANDARD DEVIATIONS (s) FOR MSEs AND TIME FOR EACH OF THE THREE ELICITATION METHODS

		wheel	bar	direct
$\alpha = 5$	$\overline{x}$	0.0786	0.0758	0.0850
	s	0.0384	0.0383	0.0448
$\alpha = 10$	$\overline{x}$	0.0685	0.0663	0.0744
	s	0.0376	0.0371	0.0431
$\alpha = 10^{-4}$	$\overline{x}$	0.1217	0.1182	0.1283
	s	0.0462	0.0468	0.0520
time	$\overline{x}$	6.6	4.9	6.9
(minutes)	s	4.0663	2.1141	2.3242

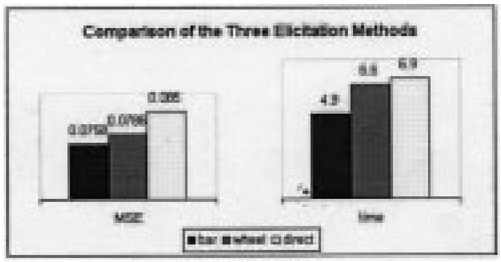

Fig. 3. MSE ($\alpha = 5$) and elicitation time for each of the three methods tested.

TABLE IV
p VALUES OF ONE-TAILED t TESTS FOR EACH PAIR OF THE ELICITATION METHODS

	bar *vs.* wheel	bar *vs.* direct	wheel *vs.* direct
$\alpha = 5$	0.19	0.03	0.07
$\alpha = 10$	0.22	0.03	0.07
$\alpha = 10^{-4}$	0.21	0.05	0.12
time	0.007	0.0005	0.37

comparison. The p values which resulted from the t tests are shown in Table IV.

The t tests showed that scaled probability bar performed significantly better than direct numerical elicitation $p \leq 0.05$ for all three values of the learning parameter α. Probability wheel was marginally better than direct numerical elicitation $p < 0.1$ for $\alpha = 5$ and $\alpha = 10$. The p value (0.12) was a little higher when $\alpha = 10^{-4}$. However, probability wheel was almost as accurate as scaled probability bar. Even though the latter had a slightly lower MSE, the difference was not statistically significant ($p \approx 0.20$ under all values of α).

From the t tests conducted for the comparison of elicitation time, we can see that generally using scaled probability bar took the shortest time ($p \leq 0.007$). However, using probability wheel did not improve the time compared with direct numerical assessment ($p = 0.37$).

D. Discussion

The experimental results showed that the learning approach to evaluate elicitation methods for probabilities is quite robust. From the results of the paired sample t tests, we can draw a conclusion about the accuracy of the three elicitation methods. Both the scaled probability bar and the probability wheel performed better than direct numerical elicitation, though the latter difference was not statistically significant. Scaled probability bar may be more accurate than probability wheel. However, the difference was again not quite statistically significant at $p = 0.05$ level. Considering time taken in elicitation processes, we can order the three methods according to their speed: probability bar, probability wheel, and direct numerical elicitation.

An interesting effect is evident in Table IV. When the value of the prior parameters was 5 or 10, the MSE for all techniques is lower than when the αs are set to 10^{-4}. In fact, when the α parameters were set to very small values, it was observed that the probabilities elicited from the experts were closer to the *true* model than they were to the *expected* models calculated with the α parameters. We believe that the reason for this discrepancy is the following. The subjects will naturally have a small but substantial prior belief of uniformity in the parameters, which may act like an anchor in the elicitation. For example, if a subject were given a loaded coin with the instruction to estimate the probability of the coin coming up heads, he or she is likely to require at least a few (5 or 10) flips of the coin before concluding that the coin is weighted one way or the other.

When the α parameters are set to 5 and 10, the elicited models are closer to the expected models than they are to the original model. Furthermore, our results were observed to be statistically significant; whereas with $\alpha = 10^{-4}$ our results were not significant. Another way of looking at this result is that if the user explored one configuration of a node's parents only a few times, then the small-α parameter model would produce very extreme, nonsmooth probability distributions under certain parent configuration. For example, if the user explored one configuration just one time, then the low-α model would produce a probability distribution with the one visited state having probability $\approx$1; whereas, a sensible user would not predict such an extreme distribution, but would rather assume that the probability was still roughly uniformly distributed.

This observation may be related to the well-known finding that people tend to overestimate very low probabilities [5]. In fact, what may be happening in the case of low-probability events is that the person's assumption of weak prior uniformity is smoothing the distribution, producing "erroneous" estimates. This fact may suggest a means of correcting for low-probability event estimates by first subtracting out the small uniform distribution from the assessed distribution.

One objection that could be raised to our technique is that a 30-min training session is not sufficient for the subjects to achieve expert status. This would be a key objection if we were comparing the elicited models to the *original* model underlying the toy-world; however, the main point in using the trainees' actual acquired knowledge is to deflect this criticism: we are comparing the elicited model precisely to the knowledge that we know our trainee has observed. In principle, this technique should work regardless of the expertise of the trainee. Nonetheless, we acknowledge that there may be some transition during the process of achieving true expertise that alters the trainees' elicitation behavior. We assume that these effects will affect the elicitation techniques in a uniform way, so that the relative assessment of elicitation techniques is not affected.

It may be that the effectiveness of different elicitation techniques varies from expert to expert. In that case, our evaluation technique can provide a relatively quick and effective way to judge which elicitation procedure is most effective for a given expert. The expert can be quickly trained on a toy model, and then our experimental procedure can be used to decide which elicitation technique is most effective for that particular expert.

VI. Conclusion

We proposed a method that allows for an objective evaluation of the elicitation methods for probability distributions and the structure of probabilistic models. Our method is based on machine learning the expert's beliefs when data of the expert's learning knowledge is available. We illustrated the evaluation approach with a toy virtual world and evaluated three elicitation methods for probabilities: 1) direct numerical elicitation, 2) the probability wheel, and 3) the scaled probability bar. Based on the results of our experiment, we concluded that the probability wheel and the scaled probability bar both performed better than direct numerical elicitation, which supports the proposition that graphical tools are useful in eliciting experts' beliefs.

Acknowledgment

The authors would like to thank Dr. M. Lewis, Dr. S. Iyengar, and members of the Decision Systems Laboratory for useful suggestions.

References

[1] M. J. Druzdzel and L. C. van der Gaag, "Building probabilistic networks: 'Where do the numbers come from?' guest editors' introduction," *IEEE Trans Knowl. Data Eng.*, vol. 12, pp. 481–486, July–Aug. 2000.

[2] W. Estes, "The cognitive side of probability learning," *Psychol. Rev.*, vol. 83, pp. 37–64, Jan. 1976.

[3] C. Spetzler and C.-A. Staël von Hostein, "Probability encoding in decision analysis," *Manage. Sci.*, vol. 22, pp. 340–358, 1975.

[4] H. Wang and M. J. Druzdzel, "User interface tools for navigation in conditional probability tables and graphical elicitation of probabilities in Bayesian networks," in *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*. San Mateo, CA: Morgan Kaufmann, 2000, pp. 617–625.

[5] S. Lichtenstein, P. Slovic, B. Fischhoff, M. Layman, and B. Combs, "Judged frequency of lethal events," *J. Experimental Psychology: Human Learning and Memory*, vol. 4, pp. 551–578, Nov. 1978.

[6] P. C. Price, "Effects of a relative-frequency elicitation question on likelihood judgment accuracy: The case of external correspondence," *Org. Beh. Human Decision Processes*, vol. 76, no. 3, pp. 277–297, 1998.

[7] M. G. Morgan and M. Henrion, *Uncertainty: A Guide to Dealing With Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[8] R. Cooke, *Experts in Uncertainty: Opinion and Subjective Probability in Science*. London, U.K.: Oxford Univ. Press, 1991.

[9] S. C. Hora, J. A. Hora, and N. G. Dodd, "Assessment of probability distributions for continuous random variables: A comparison of the bisection and fixed value methods," *Org. Beh. Human Decision Processes*, vol. 51, pp. 133–155, 1992.

[10] S. Lichtenstein, B. Fischhoff, and L. Philips, "Calibration of probabilities: The state of the art to 1980," in *Judgement Under Uncertainty: Heuristics and Biases*. Cambridge, U.K.: Cambridge Univ. Press, 1982.

[11] L. van der Gaag, S. Renooij, C. Witteman, B. Aleman, and B. Taal, "How to elicit many probabilities," in *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*. San Mateo, CA: Morgan Kaufmann, 1999, pp. 647–654.

[12] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. New York: Springer-Verlag, 1993.

[13] J. Pearl and T. S. Verma, "A theory of inferred causation," in *KR-91, Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, J. Allen, R. Fikes, and E. Sandewall, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 441–452.

[14] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Mach. Learn.*, vol. 9, no. 4, pp. 309–347, 1992.

[15] D. Heckerman, "Bayesian networks for data mining," *Data Mining Knowl. Disc.*, vol. 1, no. 1, pp. 79–119, 1998.

[16] D. Geiger and D. Heckerman, "A characterization of the Dirichlet distribution with application to learning Bayesian networks," in *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*. San Mateo, CA: Morgan Kaufmann, 1995, pp. 196–207.

[17] M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling," in *Uncertainty in Artificial Intelligence 2*, L. Kanal, T. Levitt, and J. Lemmer, Eds. Amsterdam, The Netherlands: Elsevier, 1988, pp. 149–163.

[18] GeNIe. (1999) GeNIe: A development environment for graphical decision-theoretic models. [Online]. Available: http://www2.sis.pitt.edu/~genie

[19] A. Oniśko, M. J. Druzdzel, and H. Wasyluk, "Extension of the Hepar II model to multiple-disorder diagnosis," in *Intelligent Information Systems, Advances in Soft Computing Series*, S. W. M. K̃lopotek and M. M̃ichalewicz, Eds. Heidelberg, Germany: Physica-Verlag, 2000, pp. 303–313.

[20] J. Pearl and T. S. Verma, "A theory of inferred causation," in *KR-91, Principles of Knowledge Representation and Reasoning: Proc. 2nd Int. Conf.*, Cambridge, MA.