From: AAAI Technical Report WS-02-13. Compilation copyright © 2002, AAAI (www.aaai.org). All rights reserved.

Eliciting Utilities by Refining Theories of Monotonicity and Risk

Angelo Restificar
Dept. of EE & CS
Univ. of Wisconsin-Milwaukee
Milwaukee, WI 53211

Peter Haddawy
Computer Science Program
Asian Institute of Technology
Pathumthani, Thailand 12120 and
Dept. of EE & CS
Univ. of Wisconsin-Milwaukee
Milwaukee, WI 53211

Vu Ha
Automated Reasoning Group
Honeywell Technology Center
3660 Technology Drive
Minneapolis, MN 55418

John Miyamoto
Dept. of Psychology
Univ. of Washington
Seattle, WA 98195

Abstract

Interest in such diverse problems as development of user-adaptive software and greater involvement of patients in medical treatment decisions has increased interest in development of automated preference elicitation tools. A design challenge of these tools is to elicit reliable information while not overly fatiguing the interviewee. We address this problem by using domain background knowledge in a flexible manner. In particular, we use knowledge-based artificial neural networks to encode assumptions about a decision maker's preferences. The network is then trained using answers to standard gamble type questions. We explore the use of a domain theory encoding simple monotonicity assumptions and another additionally encoding assumptions concerning attitude toward risk. We present empirical results using a data set of real patient preferences showing that learning speed and accuracy increase as more domain knowledge is included in the neural net.

Introduction

Applications ranging from development of user-adaptive software to patient-centered medical decision care require the ability to acquire and model people's preferences. A design challenge in developing automated tools for preference modeling is to elicit reliable information while not overly fatiguing the decision maker. We approach this problem by using assumptions about preferences to guide but not constrain the elicitation process. In contrast to approaches that directly exploit constraints to narrow down the search for a utility function consistent with a subject's preferences, we start the process by mapping generally known background information about the domain into a knowledge-based artificial neural network (KBANN) (Towell & Shavlik 1995). We then refine this domain theory by learning from answers to standard gamble questions. An advantage of using a neural network to elicit a subject's utility is that assumptions about utility independence and of the shape of the utility function can be relaxed. Our experimental results show that KBANN outperforms an unbiased artificial neural networks (ANN) in terms of accuracy. Moreover, the training time for KBANN is shorter than that of ANN.

Copyright © 2002, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

In previous work, Geisler, Ha, and Haddawy (Geisler, Ha, & Haddawy 2001) explored the use of KBANN to learn preferences under certainty. In this paper we explore the more difficult task of learning preferences under uncertainty, *i.e.,* utility functions. In particular, we formulate two domain theories: one expressing monotonicity of the utility function and another additionally expressing attitude toward risk. We show how to formulate the domain theories in terms of propositional Horn-clauses and to encode them in KBANN networks. The networks are constructed so that they can be trained with answers to standard gamble questions. A network encodes a utility function, which can then be easily extracted from the network by simulating elicitation, treating the network as the decision maker. We compare the performance of KBANN networks incorporating the two different domain theories with that of an ANN. We train the networks with data from patient answers to standard gamble questions and compare the accuracy and learning rates of the three networks.

Domain Theory

Miyamoto and Eraker (Miyamoto & Eraker 1988) described a psychology experiment with 27 inpatients at the Ann Arbor Veterans Administration Medical Center and the University of Michigan Hospital. The sample included patients aged between 20 and 50 with cancer, heart disease, diabetes, arthritis, and other serious ailments. The subjects were asked standard gamble questions that involved two attributes: duration of survival (Y) and health quality (Q). Given health quality Q, each subject was asked how many years in health quality Q would be equal in value to an even-chance gamble G between Y_1 and Y_2 years. The value of Q was held fixed for varying values of Y_1 and Y_2. We would like to construct a neural network that represents a subject's preferences and that can be trained using the answers to the even-chance questions (or certainty equivalents) provided by the subject. The certainty equivalents can be used to construct a subject's utility function.

In constructing the network, we consider two domain theories: D_1 and D_2. In the rules that comprise D_1 and D_2, let $Y_i, i = 1, 2$ be the duration of survival in number of years and Q be the health quality. Let Y_1 and Y_2 be the outcomes

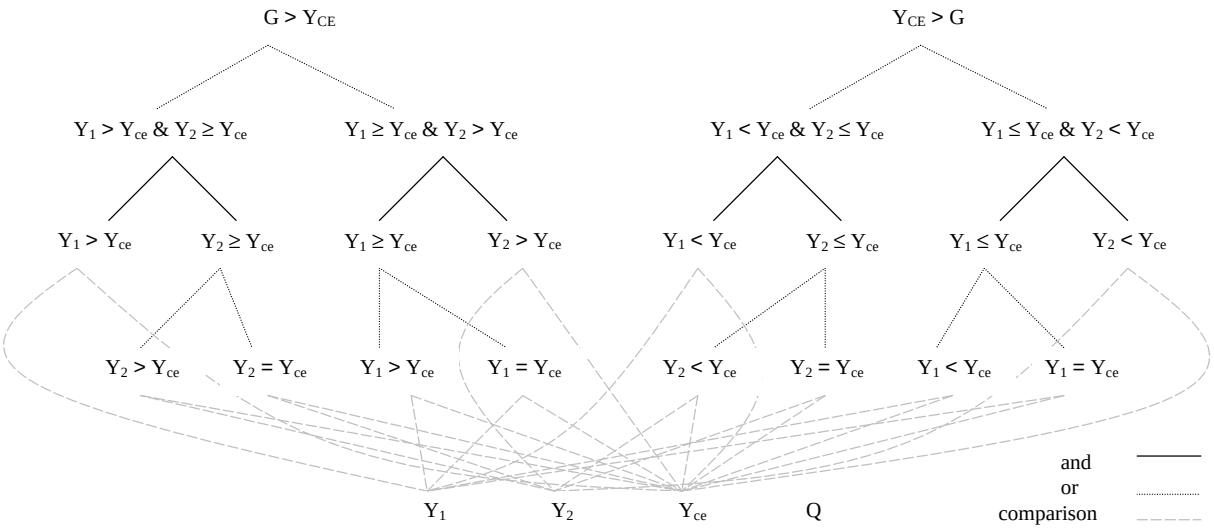

Figure 1: KBANN with D_1

in an even-chance gamble G[1], $Y_1 > Y_2$, Y_{ceq} be the certainty equivalent of the gamble, and Y_{ce} be any arbitrary outcome for certain. The first domain theory, D_1, consists of rules that implicitly specify the independence of Y and Q, as well as the fact that the utility is monotonically increasing in Y. Since Q is held fixed in each question, it need not appear in the rules. D_1 consists of the following rules:

$$(Y_1 > Y_{ce}) \wedge (Y_2 \geq Y_{ce}) \longrightarrow (G \succ Y_{ce}) \quad (1)$$
$$(Y_1 \geq Y_{ce}) \wedge (Y_2 > Y_{ce}) \longrightarrow (G \succ Y_{ce}) \quad (2)$$
$$(Y_{ce} > Y_1) \wedge (Y_{ce} \geq Y_2) \longrightarrow (Y_{ce} \succ G) \quad (3)$$
$$(Y_{ce} \geq Y_1) \wedge (Y_{ce} > Y_2) \longrightarrow (Y_{ce} \succ G) \quad (4)$$

Rules 1 and 2 say that whenever the outcomes of the even-chance gamble dominate Y_{ce}, the decision-maker prefers the gamble. Rules 3 and 4 say that in cases where Y_{ce} dominates the outcomes of the gamble, the decision maker prefers Y_{ce}.

Before we present the rules that comprise the second domain theory, we must first define proportional match. The concept of proportional match allows us to compare certainty equivalents for different pairs of values of Y_1 and Y_2 in an even-chance gamble.

Definition 1 *Let Y_{ceq} be the certainty equivalent of an even-chance gamble G. Let Y_1 and Y_2, where $Y_1 > Y_2$, be the outcomes of G, each of which has a 50% chance of occurring. A proportional match PM corresponding to Y_{ceq} is defined as follows:*

$$PM = \frac{Y_{ceq} - Y_2}{Y_1 - Y_2}$$

The motivation behind the use of the proportional match is to allow us to encode information about a decision-maker's risk attitude into the domain theory. The rules that comprise D_2 are as follows:

$$(Y_1 > Y_{ce}) \wedge (Y_2 \geq Y_{ce}) \longrightarrow (G \succ Y_{ce}) \quad (5)$$
$$(Y_1 \geq Y_{ce}) \wedge (Y_2 > Y_{ce}) \longrightarrow (G \succ Y_{ce}) \quad (6)$$
$$Y_{ce} \leq ((Y_1 - Y_2)PM_{lo} + Y_2) \longrightarrow (G \succ Y_{ce}) \quad (7)$$
$$Y_{ce} > ((Y_1 - Y_2)PM_{hi} + Y_2) \longrightarrow (Y_{ce} \succ G) \quad (8)$$

In D_2, Rules 5 and 6 are the same as in D_1. Rules 7 and 8 encode attitude toward risk. Note that $[((Y_1 - Y_2)PM_{lo} + Y_2), ((Y_1 - Y_2)PM_{hi} + Y_2)]$ is another way of writing Y_{ceq} in interval form. This representation allows flexibility when approximating Y_{ceq}. Rule 7 say that a decision-maker prefers to gamble if $Y_{ce} \leq Y_{ceq}$. On the other hand, Rule 8 say that Y_{ce} is preferred whenever $Y_{ce} > Y_{ceq}$.

Using KBANN to Model Utilities

KBANN (Towell & Shavlik 1995) is a hybrid learning system that allows rules in propositional logic to be mapped into neural networks. The mapping of rules into the network correspond to modelling preferences without an extensive set of examples. The preference information is then refined by learning from answers to standard gamble questions. Unlike an Artificial Neural Network (ANN), in KBANN the weights of the links that correspond to the domain theory are given higher values, in effect creating biased links. In our experiments, the weight, ω, is set to 4. All other links are randomly initialized to a value near 0.

The network has three hidden layers, an input layer, and an output layer . This is true for both domain theories, D_1 and D_2. For domain theory D_1 (see Fig. 1), the input layer consists of 4 nodes: Y_1, Y_2, Y_{ce}, and Q. The health state is held at Q throughout for both D_1 and D_2. For D_2 (see Fig. 2), the input layer consists of 6 nodes: Y_1, Y_2, Y_{ce}, Q, $((Y_1 - Y_2)PM_{lo} + Y_2)$ and $((Y_1 - Y_2)PM_{hi} + Y_2)$. The

[1] An even-chance gamble means that both Y_1 and Y_2 each has a 50% chance of occurring.

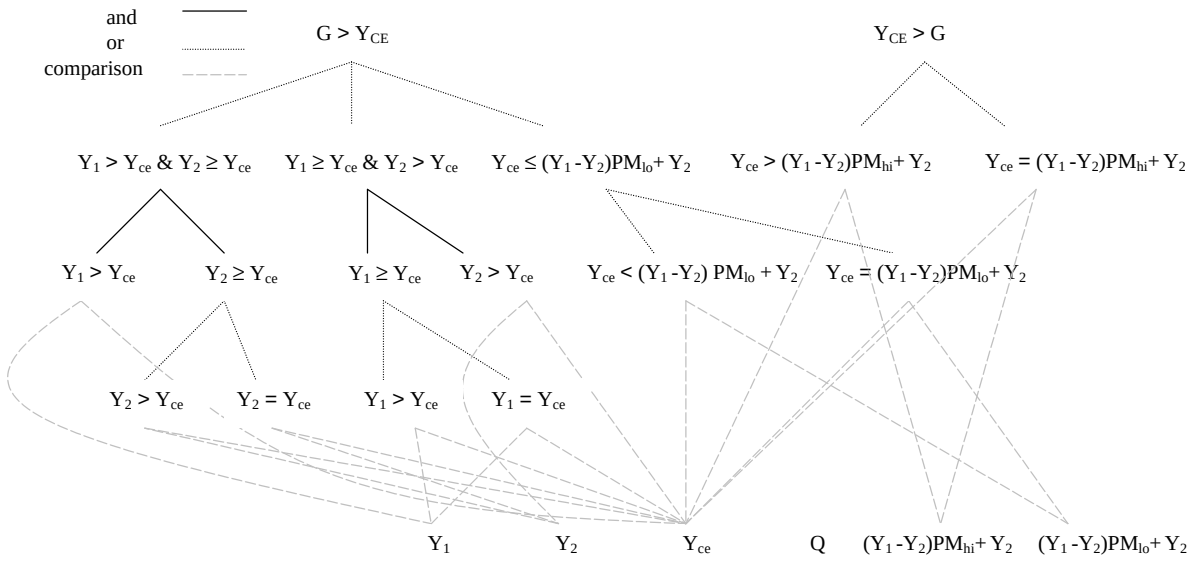

Figure 2: KBANN with D_2

output layers for both D_1 and D_2 consist of 2 nodes: The left output node is for $G \succ Y_{CE}$. This node gives an output near 1 if either Rule 1 or Rule 2 for domain theory D_1 is true or if either Rule 5, Rule 6, or Rule 7 for domain theory D_2 is true. Otherwise it gives an output near 0. The right output node is for $Y_{CE} \succ G$. It outputs a value near 1 if either Rule 3 or Rule 4 for domain theory D_1 is true, or if Rule 8 for domain theory D_2 is true. Otherwise, an output near 0 is produced.

In both networks, the hidden layer contains a mixture of $>$, $\geq$, $<$, and $\leq$ comparison nodes. The $\geq$ nodes are implemented as OR nodes, which output a value near 1 if either $x > y$ or $x = y$ is true. The weights of the incoming links for an $x > y$ node are ω for x and $-\omega$ for y. The activation function used to implement the $>$ node is:

$$\frac{1}{1 + e^{-(10 NetInput)}} \tag{9}$$

The activation function in Equation (9) gives an output close to 1 if $x > y$, an output close to 0.5 if $x = y$, and an output close to 0 if $x < y$. $NetInput$ is the usual sum of the product: weight of the link $\times$ activation output. Like the $>$ operator, the weights of the incoming links for an $x = y$ node are also $-\omega$ for x and ω for y. The activation function for this operator is:

$$\frac{1}{1 + e^{-(-30|NetInput|+5)}} \tag{10}$$

Equation (10) gives an output near 1 if x and y have values that are close to each other. However, as their difference widens the activation function produces values toward 0. The AND and OR nodes are implemented as:

$$\frac{1}{1 + e^{-(NetInput_i - Bias_i)}} \tag{11}$$

where $Bias = (P - 0.5)\omega$ for AND node and $Bias = 0.5\omega$ for an OR node. P is the number of positive antecedents on a rule, *e.g.*, Rule (5) has a value of $P = 2$. The initial weights of unbiased links, *i.e.*, weights that are not initialized to either ω or $-\omega$, are randomly set to a value in the interval $[-0.01, 0.01]$.

Empirical Analysis

We evaluated our approach using the KBANN representations of domain theories D_1 and D_2, and an artificial neural network (ANN) with 4 input nodes, 3 nodes in the hidden layer, and two nodes in the output layer. We used the set of data from (Miyamoto & Eraker 1988) which contains the answers of 27 subjects who were asked even-chance gamble questions concerning years of survival and the quality of life. Given health quality Q, each subject was asked how many years in health quality Q would be equal in value to an even-chance gamble G between Y_1 and Y_2 years. In this data set, Q was held fixed for varying values of Y_1 and Y_2. There are 24 such questions with varying values in years of survival and health quality for each of the subjects. The answers to the even-chance gamble questions are values of Y_{ceq}. Since we know what the gamble is worth to the subject, answers for values that are greater than Y_{ceq} and values that are less than Y_{ceq} are implicit: the subject prefers the gamble to values below Y_{ceq} and prefers values above Y_{ceq} to the gamble. Thus to provide the networks with data to specify preferences for values above and below Y_{ceq} we can simply supplement the answers to the standard gamble questions by including answers that are implicit. We do this by including in the training set for both KBANN and ANN randomly generated data points that are above and below the subject's Y_{ceq}. We have generated 50 such points for each of above and below Y_{ceq} regions. The networks include an input node for a value of Y for certain labeled Y_{ce}. We di-

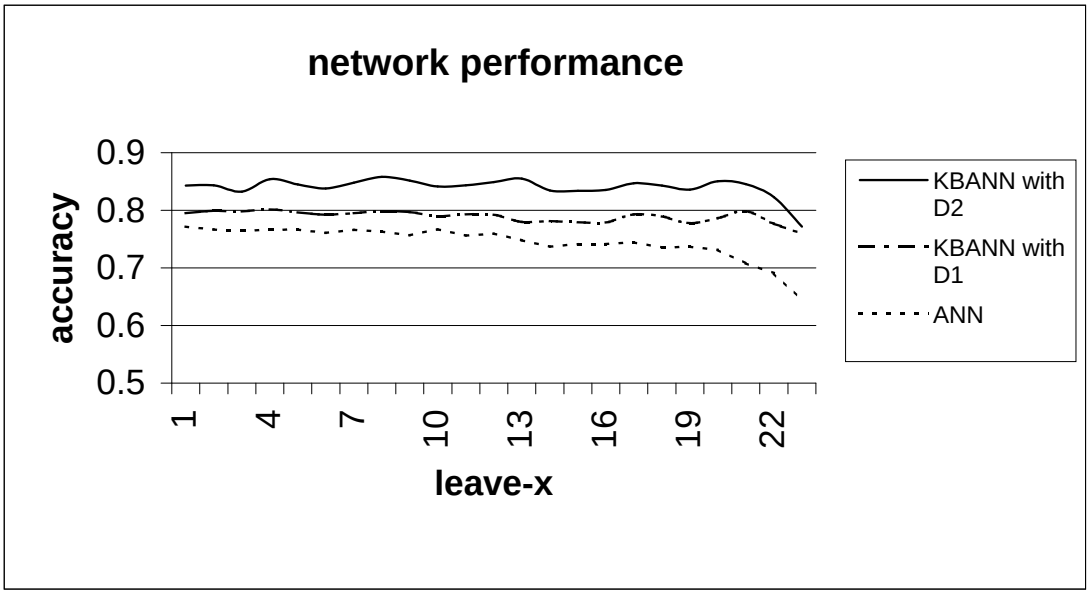

Figure 3: Accuracy comparison between networks

vided the data set into a training set, a tuning set, and a test set. For each $x = 1, \ldots, 23$ of the 24 answers, we leave out x answers for testing purposes and train the network with the remaining $24 - x$ answers. We used early stopping with a look-ahead of 100 epochs to avoid overfitting and used 10% of the training examples as the tuning set. To obtain the accuracy we simply took the percentage of the test points that are correctly classified. There are 24 experiment runs for each leave-x for a particular subject. The values obtained are averaged to get the prediction accuracy for each leave-x for a particular subject. The results of the experiment for each of the 27 subjects are averaged over all leave-x, $x = 1 \ldots 23$.

Figure 3 shows the accuracy of KBANN evaluated for risk prone subjects using domain theory D_2 (top curve), KBANN for all subjects using domain theory D_1 (middle curve), and that of ANN for all subjects (bottom curve). For the risk-prone experiment, only subjects whose average PM of the corresponding Y_{ceq} greater than 0.55 and with a standard deviation $\sigma_{\overline{PM}} \leq 0.1$ were evaluated. The performance of KBANN using domain theory D_2 is better than the performance of KBANN using domain theory D_1. Also KBANN with D_1 performs better than ANN. A right-tailed z test on the difference between the accuracy of KBANN with D_1 and ANN shows significance in 22 out of 23 tests at 90% confidence level. As the size of the training set decreases, the dominance of KBANN over ANN becomes more pronounced. This difference from leave-17 to leave-23 is statistically significant at 95% confidence level. Moreover, KBANN with D_2 requires less training time compared to KBANN with D_1 and ANN. Figure 4 shows the average number of epochs required to train the networks as the number of training examples decreases, *i.e.,* leave-x increases.

Related Work and Summary

Preference elicitation has become an important tool in medical-decision making. For example, to assess the cost-effectiveness of a treatment under study against other commonly accepted treatments, it is often necessary to measure the patients' preferences. Unlike human assessors who may potentially influence the results of the utility elicitation, automated methods of elicitation allow for a uniform process of obtaining patients' preferences. More importantly, responses maybe more frank in computer interviews than when responding on paper or to a human being(Locke *et al.* 1992). In a separate study reported in (Sanders *et al.* 1994), majority of the patients preferred to use the computer to disclose sensitive information regarding risk behaviors. U-titer (Sumner, Nease, & Littenberg 1992) is an automated, modular, utility assessment tool that allows the following methods of assessments: rating scale, category scaling, standard gamble, and time trade-off. The development of U-titer grew out of the need to find an assessment tool that minimizes systematic biases that may be present in interviews conducted by human assessors. Until the development of U-titer, computer-based assessment tools were considered to be too expensive.

IMPACT (Lenert *et al.* 1995) or *I*nteractive *M*ultimedia *P*reference *A*ssessment *C*onstruction *T*ool is designed to measure health preferences and provide support for the rapid construction of multimedia computer interviews. IMPACT has two parts: an editor and a player. The editor part is a multimedia shell program that allows researchers to interactively construct patient interviewing instruments without the need for programming. It supports text, graphics, synthesized speech, digital sound and Quicktime movies. Validation studies show that preference assessments using IM-

Figure 4: Comparison of training times between networks

PACT have high test-retest reliability and that most subjects understand the elicitation methods and believe that explanations were provided clearly. Recently, IMPACT was developed further into iMPACT3 to support the construction of instruments via the internet (Lenert 2000). Our work complements tools like U-titer and IMPACT. The number of questions the interviewee has to answer using U-titer and IMPACT may be reduced by exploiting information that has already been learned via the networks. If during the first few questions the network predicts the interviewee's answers with good accuracy then the network represents a good approximation of the interviewee's utility function. We need ask only additional number of questions to retrain the network if its predictive accuracy falls below a certain threshold.

Theory refinement via KBANN has been shown to effectively elicit and model user preferences in the certainty case(Geisler, Ha, & Haddawy 2001). This also provides a robust method of representing preferences in the case where independence assumptions have been violated to some degree. Empirical evaluation in a flight domain shows that the approach can significantly reduce the amount of information to be asked from the user. User preferences in the form of instructions are used to construct intelligent agents with the KBANN network as the core machine learning component in the *Wisconsin Adaptive Web Assistant* (WAWA) (Eliassi-Rad & Shavlik 2001). The networks are initially constructed from users' advices and refined via system-generated and user-supplied examples. Empirical results show that the theory refinement approach is highly effective in the task of retrieving and extracting information from the web.

In more recent work, (Chajewska, Koller, & Ormoneit 2001) propose an approach that estimates a probability distribution over possible courses of action of a user by observing the user's past decisions. The idea is to view the past decisions as constraints to be used to condition a prior probability distribution in order to obtain an estimate of the posterior probability distribution. Since this proposed method refine prior knowledge of utility functions, they can also be viewed as a form of theory refinement. We see the method as complementary to ours. However, our approach allows us to capture and represent utilities without the need to know the explicit utility equation.

In this paper, we have outlined an approach to utility elicitation by refining a domain theory. Our experiments show that encoding domain rules into a KBANN and refining it results in a reduction in training time and a significant improvement in accuracy over ANN. Moreover, our empirical results suggest that the inclusion of additional relevant information into the domain theory also improves prediction accuracy. Software tools like IMPACT or U-titer may be able to reduce the number of questions that will be asked from the interviewee by exploiting the information already represented in the network.

References

Chajewska, U.; Koller, D.; and Ormoneit, D. 2001. Learning an agent's utility function by observing behavior. In *Proceedings of the International Conference on Machine Learning*.

Eliassi-Rad, T., and Shavlik, J. 2001. A system for building intelligent agents that learn to retrieve and extract information. *International Journal on User Modeling and*

User-Adapted Interaction, Special Issue on User Modeling and Intelligent Agents.

Geisler, B.; Ha, V.; and Haddawy, P. 2001. Modeling user preferences via theory refinement. In *Proceedings of the Conference on Intelligent User Interfaces (IUI)*.

Lenert, L. A.; Michelson, D.; Flowers, C.; and Bergen, M. R. 1995. IMPACT: An object-oriented graphical environment for construction of multimedia patient interviewing software. In *Proceedings of the Annual Symposium on Comp Application to Medical Care*, 319–323.

Lenert, L. A. 2000. iMPACT3: Online tools for development of web sites for the study of patients' preferences and utilities. In *Proceedings of the AMIA Annual Fall Symposium*, 1172.

Locke, S. E.; Kowaloff, H. B.; Hoff, R. G.; Safran, C.; Popovsky, M. A.; Cotton, D. J.; Finkelstein, D. M.; Page, P. L.; and Slack, W. V. 1992. Computer-based interview for screening blood donors for risk of HIV transmission. *JAMA* 268(10):1301–1305.

Miyamoto, J. M., and Eraker, S. A. 1988. A multiplicative model of the utility of survival duration and health quality. *Journal of Experimental Psychology: General* 117(1):3–20.

Sanders, G. D.; Owens, D. K.; Padian, N.; Cardinalli, A. B.; Sullivan, A. N.; and Nease, R. F. 1994. A computer-based interview to identify HIV risk behaviors and to assess patient preferences for HIV-related health states. In *Proceedings of the Annual Symposium on Comp Application to Medical Care*, 20–24.

Sumner, W.; Nease, R.; and Littenberg, B. 1992. U-titer: A utility assessment tool. In *Proceedings of the Annual Symposium on Comp Application to Medical Care*, 701–705.

Towell, G. G., and Shavlik, J. W. 1995. Knowledge-based artificial neural networks. *Artificial Intelligence* 70.