Decision Theoretic Perspective on Optimizing Intelligent Help

Chulwoo Kim and Mark R. Lehto

School of Industrial Engineering, Purdue University, West Lafayette, IN 47907, USA
{kim218,lehto}@purdue.edu

Abstract. With the increasing complexity of systems and information overload, agent technology has become widely used to provide personalized advice (help message) to users with their computer-based tasks. The purpose of this study is to investigate the way to optimize advice provided by the intelligent agent from a decision theoretic perspective. The study utilizes the time associated with processing a help message as the trade-off criterion of whether to present a help message or not. The proposed approach is expected to provide guidance as to where, when and why help messages are likely to be effective or ineffective by providing quantitative predictions of value of help messages in time.

Keywords: intelligent agent, intelligent help, decision theoretic perspective, help optimization.

1 Introduction

The increasing complexity of systems and information overload makes an assistance system an essential component of a software system. Computer interfaces are having more and more functionality, and the limitations of computer development will be how to enable the computers to know what the users want the computers to do [1]. As indicated in the production paradox [2], novice users try to perform tasks with a computer system almost immediately without knowing how to do them, resulting in frustrations with the system. As a way of overcoming these problems, agent technology has become widely used to provide personalized assistance to users with their computer-based tasks. The role of intelligent agents as an advice provider has been advocated [1, 3, 4].

The difficulty of providing advice (help messages) is that there are several possible answers to the problems based on limited inputs or no appropriate cues. Given that it is impossible to develop a perfect intelligent agent which always provides an appropriate help message, it is important to develop a framework to trade off among various variables related to selecting the right sort of help message. The traditional approach to providing advice is to provide the most likely or suitable advice suggested by the implemented algorithm. The problem with this approach is that it does not properly consider the cost and benefit of the advice. As the scheme of

J. Jacko (Ed.): Human-Computer Interaction, Part III, HCII 2007, LNCS 4552, pp. 358–365, 2007.

trading off the cost and benefit of the advice, expected utility has been proposed [3] and used in some intelligent help systems using subjective cost and benefit functions. Although the decision theoretic optimization based on the expected utility (cost) provides a powerful and flexible tradeoff mechanism, users or domain experts have to be directly asked about their preferences to develop a cost (utility) function. Domain-specific learning techniques have been used occasionally, but most practitioners parameterize the cost function and then engage in a laborious and unreliable process of hand-tuning [5].

This study investigates the way of optimizing the provision of a help message with the assumption that the cost and value of a help message can be measured in terms of time.

2 Literature Review

2.1 Intelligent Agent

The intelligent agent is generally defined as a hardware or software-based computer system with autonomy, social ability, reactivity, and pro-activeness [6, 7]:

- Autonomy: agents act without the direct interruption of human users, and have some control over their behaviors;
- Social ability: agents interact with other agents or human users using some type of communication language;
- Reactivity: agents perceive their environment and respond to changes;
- Pro-activeness: agents show goal-directed behavior by taking the initiative

The basic function of intelligent agents is to provide personalized assistance to users with their computer-based tasks.

Although the intelligent agent technology provides a promising future, it still has problems for its fullest adoption to the interface design. The debate on the advantages and disadvantages of intelligent agents and direct manipulation [8] has highlighted differing views on the most promising opportunities for user interface innovation [3]. One group has expressed optimistic opinions for refining intelligent-interface agents with suggestions that research should focus on developing more powerful tools for understanding a user's intention and taking automated actions. Another group has shown concerns that the efforts should be directed toward tools and metaphors that improve users' ability to understand and directly manipulate systems and information.

[9] summarizes problems associated with the user-initiated approach and the system-initiated approach, and argue the adequacy of the mixed approach. The problems of the user-initiated approach can be summarized in three aspects: 1) the user-initiated approach is often ineffective; 2) the user-initiated approach is often inefficient in that it makes no use of information about the user and the task progress; and 3) users handle their tasks without necessarily optimizing the solution due to the cost of learning. The challenges faced by the system-initiated approach can be

summarized in three aspects: 1) it is often inappropriate since the correct identification of a user's goal is not always possible; 2) the major challenge is the timing of advice; and 3) users tend to have predictability and the control of the system.

Two approaches have been studied to overcome the limitations of the user-initiated and system-initiated approaches: the mixed initiative approach and the advice approach. The mixed initiative approach indicates a creative integration of direct manipulation and automation [3]. [3] argues that the mixed initiative approach could provide a different kind of user experience characterized by more natural collaborations between users and computers, rather than advocating one approach over the other.

The advice approach is to consider computer systems as an advisor to provide people with suggestions, help and assistance, and users decide whether to use them or not [1]. A key idea in achieving the transition from a *command* structure to a more flexible and collaborative one will be the development of computer interfaces based on the idea of *advice*. This study focuses on the advice giving aspect of the intelligent agent.

2.2 Decision Theoretic Perspective

The decision theoretic perspective provides a more fundamental explanation of how and why more or less information might be transmitted, in particular situations, that takes into consideration both uncertainty and the outcome of the consequences occurred [10]. The basic idea is that the amount of information transmitted can be determined by taking into consideration user knowledge, uncertainty, and the outcome of the consequences incurred. It indicates that the amount of information transmitted is not necessarily the index of a better intelligent agent. In most cases, it can be assumed that the more information transmitted the better. However, this is not always true because the amount of information transmitted doesn't take into consideration the cost and benefits of the help messages provided by the intelligent agents. Especially in the information overloaded environment such as the computer systems of these days, the problem is when and how much information or how many messages the agent needs to present to the user.

There has been a recent trend to use decision theoretic optimization, of which the objective is to minimize the expected cost and maximize expected utility for designing a user interaction interface [5, 11, 12]. The basic idea of decision theoretic optimization for designing intelligent agents is that a proposed action will be taken only when the agents believe that the action has greater expected value than inaction [11]. The approach used in developing these systems involves manual preference elicitation methods by directly or indirectly asking for users' subjective ratings on the preference or by domain experts. Although decision-theoretic optimization provides a powerful and flexible approach for these systems, the accuracy of the underlying utility function determines the success of these systems [5].

3 Expected Time-Based Optimization Framework

3.1 Classification of User Needs

The way that the intelligent help agents respond to the user needs represents the proactiveness of the intelligent help agents. The intelligent help agents are required to provide proactive help, not just respond passively to the user's requests. In the traditional signal detection theory, a signal indicates an event or object that needs to be identified. With this definition in mind, the user's need for help can be considered as a signal in the context of using the intelligent help agents.

There are two cases where the user needs help. The first case is the system-driven need where the system considers certain information may be helpful to the user. As indicated in [9], users often do not look for needed help probably because users do not have experience or skills to find the needed information. If the system recognizes that certain information may assist the user to complete the user's task, the system should provide help messages. Help messages that may shorten task completion time or help messages that indicate problematic situations are the examples representing the system-driven need.

The second case is the user-driven need that the user recognizes certain information is needed. The agent is supposed to provide help messages to the user when the user looks for help because the user does not know what to do or how to do the user's task. In traditional systems, the user uses a help function which provides a list of help messages related to operations of the systems. The intelligent agent approach is to provide help messages autonomously based on the user's context. If the user is doing the user's task without any problem, the user does not need this type of help so that the agent should not provide any help messages.

The agent's actions of providing a help message can be categorized depending on the user need as summarized in Table 1. Correct identification indicates that the agent provides a help message when the user needs it. False alarm indicates that the agent does not provide a help message when the user needs it. Miss indicates that the agent provides a help message when the user does not need it. Correct rejection indicates that the agent does not provide a help message when the user does not need it.

Table 1. Classification of help messages

	s (User need)	n (No user need)
S (Help message)	Correct Identification	False Alarm
N (No help message)	Miss	Correct Rejection

3.2 Cost of Help

The above classification only involves the identification of user needs without the relevance of a help message being considered. The Correct Identification only indicates that the agent identifies the user need at the right time. Although the system correctly identifies the user need, it would not be helpful if the system provides a message which is irrelevant or costs more to process than without the message. Providing an irrelevant message causes extra time to process the help message without assisting the user. Presenting a message at the wrong time will not be helpful since a message at the wrong time is considered an irrelevant one.

Although there are several considerations for help message presentation, the usefulness of a help message can be evaluated from the utilitarian perspective. Assuming that the use of a computer system depends on the utilitarian aspect of the system, time can be a good measure of the system performance. Although relevance can be an important standard of determining accuracy of the help system, a relevant help message which the user already knows is not considered helpful since the message only increases the user's task completion time with the extra time to process the help message. The introduction of time is beneficial in that it can explain why even a correct, relevant message can be useless and undermines user performance. The basic concept of this approach is supported by the GOMS model [13]. [13] consider that tasks can be described as a serial sequence of cognitive operations and the time associated with operations can be approximated. By analyzing time associated with processing a help message, we can decide whether to provide a help message or not.

To complete a task, the user is supposed to perform several subtasks. There is not only one way of performing a task so that at each decision point, the user decides which subtask to perform. The amount of time devoted to each routine varies, so it is desirable to provide a help message in the way that the help message reduces the amount of time required to complete the user's task. The type of help message that guides the user to a shorter route will be beneficial in that the user can wisely select the subsequent subtask. Some help messages can even help the user to skip some subtasks. The help messages when there is only one route may not be helpful, only causing extra time to read the messages.

3.3 Optimization of Help

Given that cost can be assigned to the outcome, the optimal help threshold can be assessed using the expected utility of each action [11]. The help should be given when the probability that the user needs help exceeds the optimal help threshold. With time being the utility of each action, the expected utility of presenting a help message or presenting no message can be expressed in the time associated with each action. Since it is difficult to calculate the time associated with processing a help message without considering the task performance time, the expected time of presenting a help message and presenting no help message is estimated using the total task performance time.

The expected time of presenting a message and presenting no message can be calculated as shown below:

$$\begin{aligned} ET(message) &= P_s * T_{HIT} + (1-P_s) * T_{FA} \\ &= P_s * (T_{Task} + T_{Message\_HTT} - T_{Saved}) + (1-P_s)(T_{Task} + T_{Message\_FA}) \\ ET(no\_message) &= P_s * T_{MISS} + (1-P_s) * T_{CR} \\ &= P_s * T_{Task} + (1-P_s) * T_{Task} \\ &= T_{Task} \end{aligned} \quad (1)$$

Where:

P_s = the probability of a signal (help)

T_{CR} = cost of a correct rejection (the system does not provide a help message when help is not needed)
$= T_{Task}$

T_{FA} = cost of a false alarm (the system provides a help message when help is not needed)
$= T_{Task} + T_{Message\_FA}$

T_{HIT} = cost of a correct identification (the system provides a help message when help is needed)
$= T_{Task} + T_{Message\_HTT} - T_{Saved}$

T_{MISS} = cost of a missed signal (the system does not provide a message when help is needed)
$= T_{Task}$

T_{Task} = Time needed to complete a task without a help message

T_{Saved} = Time saved by a help message

$T_{Message\_HIT}$ = Time to process a help message when the system provides a help message when help is needed

$T_{Message\_FA}$ = Time to process a help message when the system provides a message when help is not needed

The expected task performance time of presenting a help message is the summation of the times associated with processing a help message and performing a task with a help message. The time to process a help message does not only include the time to read the message but also the time associated with performing the suggestion in the message and the remedy action if the suggestion in the message is wrong. The basic assumption of the equation is that the time to process a help message varies according to whether it is a correct identification or a false alarm. The user will be more likely to follow the suggestion in the help message if the message is provided when the user needs help. The expected time of presenting no help message is exactly same as the task performance time since there is no change.

The optimal help threshold probability can be calculated as shown below:

$$\begin{aligned} & ET(message) = ET(no\_message) \\ & P_s(T_{Task} + T_{Message\_HTT} - T_{Saved}) + (1-P_s)(T_{Task} + T_{Message\_FA}) = T_{Task} \\ & P_s * = \frac{T_{Message\_FA}}{T_{Saved} + T_{Message\_FA} - T_{Message\_HTT}} \end{aligned} \quad (2)$$

The help message should be provided when the probability that the user needs help exceeds the optimal threshold probability (P_s^*).

4 Conclusion

This study proposed the decision theoretic framework of optimizing the provision of a help message considering time as its cost and value. Although there are multiple purposes for providing advice such as increasing accuracy, increasing user confidence, etc., the purpose of reducing time required to complete a task is more appropriate for many tasks which are not safety-related or not critical. Many tasks that people do every day using their computers would belong to this category. By assigning the time associated with processing a help message provided by an intelligent agent as cost, The proposed approach would provide guidance as to where, when and why help messages are likely to be effective or ineffective by utilizing quantitative predictions of value and cost of intelligent help messages in time.

Future studies can be directed toward implementing the proposed approach in the real system and test if the proposed approach can save time and improve user satisfaction.

References

1. Lieberman, H.: Interfaces that give and take advice. In: Carroll, J.M. (ed.) Human-Computer Interaction in the New Millennium, pp. 475–485. ACM Press/ Addison-Wesley, N.Y (2001)
2. Carroll, J., Rosson, M.B.: Paradox of the active user. In: Carroll, J.M. (ed.) Interfacing thought: cognitive aspects of human-computer interaction, pp. 80–111. MIT Press, Cambridge, MA (1987)
3. Horvitz, E.: Uncertainty, Action, and Interaction: In Pursuit of Mixed-Initiative Computing. Intelligent Systems, IEEE Computer Society, pp. 17-20 (September / October 1999)
4. Virvou, M., Kabassi, K.: Evaluating an intelligent graphical user interface by comparison with human experts. Knowledge-Based Systems 17, 31–37 (2004)
5. Gajos, K., Weld, D.S.: Preference elicitation for interface optimization. In: Proceedings of the 18th annual ACM symposium on User interface software and technology, Seattle, WA, USA, pp. 173 –182 (2005)
6. Wooldridge, M.: Intelligent agents. In: Weiss, G. (ed.) Multiagent systems, a modern approach to distributed artificial intelligence, pp. 27–78. MIT Press, Cambridge, MA (1999)
7. Klusch, M.: Information agent technology for the Internet: A survey. Data. and Knowledge Engineering 36, 337–372 (2001)
8. Shneiderman, B., Maes, P.: Direct manipulation vs. interface agents. Interactions 4(6), 42–61 (1997)
9. Mao, J., Leung, Y.W.: Exploring the potential of unobtrusive proactive task support. Interacting with Computers 15, 265–288 (2003)

10. Lehto, M.R.: Optimal warnings: an information and decision theoretic perspective. In: Wogalter, M.S. (ed.) The handbook of warnings, pp. 89–108. Erlbaum, Mahwah, NJ (2006)
11. Horvitz, E.: Principles of mixed-initiative user interfaces. In: Proceedings of the ACM SIGCHI, Pittsburgh, PA, pp. 159–166. ACM Press, New York (1999)
12. Zhou, M.X., Wen, Z., Aggarwal, V.: A graph-matching approach to dynamic media allocation in intelligent multimedia interfaces. In: Proceedings of the 10th international conference on Intelligent user interfaces, San Diego, California, pp. 114-121 (2005)
13. Card, S.K., Moran, T.P., Newell, A.: The Psychology of Human-Computer Interaction. Lawrence Erlbaum, Hillsdale, N.J (1983)