The use of increasingly specific user models in the design of mixed-initiative systems

Michael Fleming

Faculty of Computer Science
University of New Brunswick
Fredericton, NB
mwf@unb.ca

Introduction

- Designing mixed-initiative systems
  - intelligent system and human user working together on a problem
  - both parties able to contribute to decision-making

- Decisions about whether to interact with potentially helpful users

- One essential element of this design: modeling the probability that a particular user will have the knowledge required to answer a particular question

- Using increasingly specific user models as the system gains familiarity with each user

User modeling

- User model: system's internal representation of a user's knowledge, abilities, preferences, goals, etc.
- Allows the system to adapt its behaviour in a way that is appropriate for each user
- The user model is a crucial component of my model for reasoning about interaction with users.
- Primarily interested in modeling
  - users' knowledge about the domain
  - users' willingness to interact with the system

Modeling user knowledge

- Taking into account information about
  - the *specific* user's knowledge
  - the knowledge expected of users in the same *stereotype*
  - the general knowledge expected of *all* users

  and the probability of getting an answer to
  - the *specific* question being considered
  - a particular *class* of questions
  - *any* question in the domain

- Combining all of the above to determine a single value (P_{UK}) representing the probability of the user being able to answer a particular question

- Weighted according to the amount of experience the system has with particular users and topics

User vectors

- For each user, construct a $1 \times (m + n + 1)$ vector $\mathbf{u}$.
  - first m entries identify the user as one of the m users about whom the system has a profile
  - next n entries classify the user into one or more of the n stereotypes of which the system is aware

- Example below:
  - The user is the second of four known users.
  - There are three stereotypes; the user is primarily identified as belonging to the first stereotype, but has some properties of the third as well.
  - The final entry of this vector is always 1 and indicates that the user is a member of the class of "all users" of the system.

$$\left(\begin{array}{cccc|ccc|c} 0 & 1 & 0 & 0 & 0.9 & 0 & 0.1 & 1 \end{array} \right)$$

User vectors

- Also, for each user, construct a second $1 \times (m + n + 1)$ vector **uw**.
- Stores weights to indicate how much impact each *type* of user-knowledge information should have in computing P_{UK}.
- Initially: low weight on user-specific information, high weight on stereotypical information and knowledge expected of *any* user

$$\left(\begin{array}{cccc|ccc|c} 0 & 0 & 0 & 0 & 0.8 & 0.8 & 0.8 & 0.2 \end{array} \right)$$

- Weights in this vector updated as system observes knowledge of specific users
- Weight of contribution from user-specific information increases gradually until it eventually overwhelms other weights

Topic vectors

- For each possible topic or question, construct two $(i + j + 1) \times 1$ vectors $\mathbf{t}$ and $\mathbf{tw}$. These are analogous to the user vectors described earlier.

- In the first vector $\mathbf{t}$:
  - first i entries uniquely identify the question
  - next j entries classify it into one or more classes of questions
  - final entry always 1

- Example below: topic vector for the third of five questions, classified as belonging to the second of two classes of questions or topic areas.

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & | & 0 & 1 & | & 1 \end{pmatrix}^T$$

- Second vector $\mathbf{tw}$ indicates desired weights on specific questions, general topic areas, and domain as a whole

PK matrix

- Large $(m + n + 1) \times (i + j + 1)$ matrix **PK** storing probability values
  - Rows: users and user stereotypes
  - Columns: questions and topic areas.

- Example below: bold entries indicate that the first user has a probability of 0.8 of being able to answer the first question, and that users belonging to the third stereotype have a probability of 0.7 of answering questions in the second topic area.

$$\left(\begin{array}{ccccc|cc|c}
\mathbf{0.8} & 0.6 & 0.5 & 0.2 & 0.0 & 0.7 & 0.3 & 0.7 \\
0.7 & 0.6 & 0.7 & 0.2 & 0.0 & 0.7 & 0.3 & 0.6 \\
0.7 & 0.6 & 0.7 & 0.2 & 0.0 & 0.6 & 0.3 & 0.5 \\
0.7 & 0.6 & 0.7 & 0.2 & 0.0 & 0.6 & 0.3 & 0.6 \\
\hline
0.7 & 0.6 & 0.7 & 0.2 & 0.0 & 0.8 & 0.5 & 0.6 \\
0.8 & 0.7 & 0.8 & 0.2 & 0.0 & 0.6 & 0.6 & 0.7 \\
0.9 & 0.8 & 0.9 & 0.2 & 0.0 & 0.4 & \mathbf{0.7} & 0.6 \\
\hline
0.8 & 0.7 & 0.8 & 0.2 & 0.0 & 0.6 & 0.6 & 0.6
\end{array}\right)$$

Calculating P_{UK}

- Matrix multiplication to determine a single P_{UK} value, based on a weighted combination of the above information.

- Multiply **u** by **uw** element-wise, to yield a new vector $\mathbf{u_{wtd}}$

- Multiply **t** by **tw** element-wise, to yield a new vector $\mathbf{t_{wtd}}$

- Matrix multiplication to obtain $P_{UK} = \mathbf{u_{wtd}}\ \mathbf{PK}\ \mathbf{t_{wtd}}$.

- In practice, when the vectors and matrices are very large, it makes sense to perform these multiplications not with the entire matrices, but by selecting only the rows and columns that are relevant to the current user and the current question.

Probabilities and weights

- Actual probabilities of different users being able to answer different questions and weights on the various components of the model also maintained through the use of a set of matrices

- $\mathbf{N_{ask}}$: tracks how many times each question (or type of question) has been asked of a user

- $\mathbf{N_{ans}}$, tracks how many times the user has actually provided an answer to each question.

- The values in these matrices are used to compute the individual values stored in the matrix **PK**.

Final comments

- Allowing systems to adapt gradually from very general models about expected user knowledge to user-specific models
- Speed at which this adjustment occurs is under the control of the system designer
- Similar model could be used to weigh the contributions of many different users/agents in multiple-participant decision-making situations
- Addresses two general concerns in user modeling:
  - how to initialize values in user models
  - how to adjust these models over time