

Un enfoque bayesiano de reconocimiento de planes para agentes de interfaz

Marcelo Armentano, Analía Amandi

Instituto de investigación ISISTAN, UNICEN
Campus Universitario (CP 7000),
Tandil, Bs. As., Argentina,
También CONICET, Consejo Nacional de Investigaciones Científicas y Técnicas
{marmenta, amandi}@exa.unicen.edu.ar

Resumen

Los agentes de interfaz son entidades de software que proveen asistencia personalizada a un usuario en el uso de aplicaciones de software. Comprendiendo las tareas que el usuario ejecuta en una aplicación de software, un agente de interfaz puede ser consciente del contexto que representa el foco de atención del usuario en cada momento. Con este propósito, el reconocimiento de planes apunta a identificar los planes y objetivos de un usuario a partir de las tareas que éste¹ ejecuta. Un prerrequisito para el reconocimiento de planes es tener conocimiento acerca de las posibles tareas que puede ejecutar el usuario y de la combinación de estas tareas que describen el comportamiento típico del usuario. El reconocimiento de planes le permitirá a un agente de interfaz razonar acerca de lo que intenta hacer el usuario de tal forma que pueda colaborar con él. En este trabajo proponemos un modelo probabilístico de las intenciones de un usuario que va a permitir a un agente de interfaz realizar reconocimiento de planes en las tareas que ejecuta. El modelo propuesto es capaz de tratar con incertidumbre, múltiples planes, múltiples objetivos intercalados, tareas sobrecargadas, tareas espurias, interrupciones y capaz de adaptarse a un usuario particular de la aplicación.

Palabras clave: Agentes de Interfaz, Reconocimiento de planes

1. Introducción

Los agentes de interfaz [Maes, 1994] son entidades de software diseñados para asistir a usuarios humanos en las tareas que realiza en una computadora. Este tipo de agentes es capaz de aprender los intereses, preferencias, prioridades, objetivos y necesidades de un usuario con el objetivo de proveerle asistencia proactiva y reactiva para incrementar la productividad del usuario en el uso de la aplicación de software en cuestión.

Con este objetivo, los agentes de interfaz no sólo deben aprender las preferencias y hábitos del usuario con respecto al uso de la aplicación en sí, sino que también deberían tener en cuenta la tarea que está realizando antes de iniciar una interacción con el

usuario. Considerar el estado de atención del usuario (la tarea que está ejecutando o el objetivo que está siguiendo) y la incertidumbre acerca de sus objetivos son factores críticos para una integración efectiva de servicios automatizados con interfaces de manipulación directa [Horvitz, 1999]. En consecuencia, debemos construir agentes capaces de reconocer o predecir momentos oportunos para ganar la atención del usuario, tal como cuando éste termina con una tarea.

Un agente de interfaz tiene dos formas de detectar la intención de un usuario: preguntarle o inferirla del contexto. Preguntarle al usuario cuál es su intención es una manera directa de acceder a la misma, pero corremos el riesgo de molestarlo. Inferir la intención del usuario a partir del contexto, por otro lado, es una

¹ Sin intención de ser sexistas, por simplicidad nos referimos al usuario como "él".

tarea difícil, porque la información obtenida a partir de las interacciones del usuario con la aplicación es de bajo nivel comparada con las intenciones que lo impulsan a llevar a cabo dicha interacción. Sin reconocimiento de planes, el usuario debe declarar cada objetivo antes de comenzar a ejecutar las tareas para lograrlo para que el agente sea capaz de entender sus intenciones, o el agente tiene que preguntarle al usuario acerca de las mismas.

Hay muchas ventajas de usar información de contexto en vez de consultar al usuario directamente acerca de dudas que surjan al agente o para inferir qué está intentando hacer. Consultar al usuario ralentiza la interacción, interrumpe la línea de pensamiento del usuario y incrementa la posibilidad de errores [Lieberman y Selker, 2000]. Sin embargo, el contexto debe ser tratado cuidadosamente.

Consideremos, por ejemplo, un agente de interfaz que cada vez que el usuario organiza una reunión le ofrece enviar un mail de invitación a los participantes de la misma, aunque el usuario generalmente rechaza este ofrecimiento. La primera vez es tolerable, pero esta situación se torna cada vez más molesta para el usuario a medida que éste sigue rechazando este tipo de sugerencias y el agente continúa realizándolas. El agente debería considerar este hecho y no realizar más este tipo de sugerencias. Sin embargo, la solución es probablemente tener en cuenta un contexto más específico, en lugar de no considerarlo. El agente debería detectar que la sugerencia es rechazada por el usuario en todos los casos, o debería considerar otros parámetros de la reunión, tales como los participantes, el lugar en el que se realiza o el tipo de reunión como clasificadores que le permitan decidir si sugerir enviar un mail de invitación o no.

El objetivo del reconocimiento de planes es identificar el objetivo (o intención¹) de un sujeto basándose en las tareas que realiza. En este contexto, una "tarea" corresponde a una acción que el usuario puede ejecutar en la aplicación de software y un "objetivo" es una intención de más alto nivel del usuario (que será cumplido por medio de la ejecución de un conjunto de tareas). Un objetivo usualmente tiene uno o más planes asociados que permiten predecir el comportamiento subsiguiente del usuario. Además el historial de interacción del usuario puede utilizarse para refinar el contexto que implícitamente está estableciendo, considerando no sólo las tareas que probablemente ejecutará, sino también los atributos de dichas tareas.

En este trabajo proponemos un enfoque probabilístico que permita a un agente de interfaz

detectar las intenciones del usuario. En la Sección 2 presentamos el problema de reconocimiento de planes y diferentes enfoques que han sido propuestos. En las secciones 3 y 4 presentamos nuestro enfoque. Finalmente en la Sección 5 exponemos nuestras conclusiones.

2. Antecedentes en reconocimiento de planes

Los sistemas de reconocimiento de planes observan las tareas que ejecuta un usuario y tratan de encontrar todos los posibles planes que pueden ser explicados a partir de dichas tareas. Estas posibles explicaciones, o planes candidatos, son luego reducidos a medida que el usuario continúa ejecutando tareas. El reconocimiento de planes ha sido aplicado a diferentes tareas informáticas tales como análisis de discurso [Grosz y Sidner, 1990], planeamiento colaborativo [Huber y Durfee, 1993], comprensión de historias [Charniak y Goldman, 1993], tutores inteligente [Koehn y Greer, 1995], sistemas de ayuda [Horvitz et al., 1998], agentes de interfaz y colaborativos [Lesh, 1999; Brown, 1998].

Si bien la idea básica tras el reconocimiento de planes parece ser directa, cuando es aplicada a un ejemplo práctico debemos tener en cuenta varios puntos:

- La incertidumbre inherente al momento en el que el usuario comienza un nuevo plan y a los objetivos del usuario en sí.
- Algunas tareas pueden ser parte de más de un plan (tareas sobrecargadas)
- El usuario puede interrumpir o suspender el plan que está ejecutando.
- El usuario puede ejecutar tareas que no forman parte de su objetivo principal (tareas espurias)
- Frecuentemente hay más de una forma de alcanzar un objetivo (múltiples planes)
- El usuario puede intentar alcanzar varios objetivos a la vez (múltiples objetivos intercalados)

Otro aspecto clave a tener en cuenta cuando consideramos la tarea de reconocimiento de planes en el contexto de agentes de interfaz es la adaptación del proceso a las peculiaridades de cada usuario particular de la aplicación.

Todos los sistemas de reconocimiento de planes necesitan una base de conocimiento que codifique de alguna forma las creencias del agente en cuanto a

¹ Los términos "objetivo" e "intención" son usados indistintamente en este trabajo

cómo el usuario puede alcanzar cada objetivo particular en el dominio de la aplicación. Varias representaciones de los planes del usuario han sido propuestas para tratar con este problema, y varios métodos han sido aplicados para inferir la intención del usuario. Estos métodos pueden agruparse en dos categorías principales: *enfoques de consistencia* y *enfoques probabilísticos*.

Los enfoques de consistencia hacen frente al problema de reconocimiento de planes determinando cuál o cuales de un conjunto de objetivos de entrada son consistentes con las tareas observadas. Se dice que un objetivo g es consistente con una secuencia de tareas observadas A si A puede ser ejecutada en servicio de g . Kautz [Kautz y Allen, 1986] provee la primer teoría formal de reconocimiento de planes en la cual los planes y los objetivos se representan en una jerarquía de eventos que describe todo el comportamiento que puede exhibir un usuario en un dominio particular. Toda tarea observada es parte de uno o más planes de alto nivel, y el proceso de reconocimiento de planes consiste en minimizar el conjunto de planes de alto nivel suficientes para explicar las tareas observadas. El sistema de reconocimiento de planes presentado en [Lesh y Etzioni, 1995] utiliza un grafo que representa las relaciones entre las tareas y posibles objetivos en el dominio. El sistema iterativamente aplica reglas de poda para eliminar de este grafo los objetivos que no son consistentes con ningún plan, dadas las tareas observadas. En un trabajo posterior [Lesh, 1998] Lesh introduce los conceptos de sesgos en planes y objetivos, que son suposiciones acerca de qué tipo de planes y objetivos tienen los usuarios para construir automáticamente la librería de planes a partir de tareas primitivas y predicados de objetivos. Finalmente, COLLAGEN [Rich et al., 2001] es un sistema de interfaz a usuario inteligente que modela la interacción humano-computadora basándose en los conceptos de SharedPlan [Grosz y Kraus, 1999]. COLLAGEN utiliza un conjunto de tareas, de las que distingue tareas primitivas y tareas de alto nivel. Luego, utiliza *recetas* para descomponer las tareas no primitivas en sub-objetivos. Las recetas se representan como funciones que mapean una tarea a un plan para ejecutar esa tarea.

Otra forma natural de ver el reconocimiento de planes es como un razonamiento probabilístico. El plan que el sistema selecciona como intención candidata es aquel con la mayor probabilidad a la luz de la evidencia en cada momento. El punto más fuerte del enfoque probabilístico es que permite capturar el hecho de que a priori algunos planes son más probables que otros. Los enfoques probabilísticos representan explícitamente la

incertidumbre asociada a los planes del usuario y permiten generar un ranking de las intenciones del usuario.

Los enfoques probabilísticos al reconocimiento de planes hacen principalmente uso de redes de Bayes [Charniak y Goldman, 1993; Brown, 1998; Horvitz et al., 1998; Huber y Simpson, 2004] y modelos de Markov [Davison y Hirsh, 1998; Gorniak, 2000].

La estructura de los modelos de Markov es un grafo de transición de estados. Esta simple estructura es debida a la suposición de Markov para representar secuencias de eventos, que enuncia que la ocurrencia del siguiente evento depende sólo de una cantidad fija de eventos previos. Dado ciertos eventos previamente observados, el siguiente evento se predice a partir de la distribución de probabilidad de los eventos que siguieron a dichos eventos observados en el pasado.

Las redes de Bayes [Pearl, 1988], por otro lado, son una representación popular para razonar bajo incertidumbre porque combinan una representación gráfica de relaciones causales con una buena base probabilística. La estructura de grafo dirigido acíclico de las redes de Bayes contiene representaciones tanto de las dependencias como de las independencias condicionales entre los elementos del dominio del problema. El conocimiento se representa por medio de nodos llamados variables aleatorias y de arcos que representan relaciones causales entre las variables. La fuerza de estas relaciones se describen utilizando parámetros codificados en tablas de probabilidad condicional (TPCs). Cada entrada en estas tablas indica la probabilidad de que la variable en cuestión esté en un determinado estado dada cada configuración de estados de las variables padre.

Otros enfoques, tales como el presentado en [Geib, 2004] siguen un enfoque mixto en el que primero encuentran las posibles explicaciones para las tareas observadas y luego establecen una distribución de probabilidad sobre ellas.

En resumen, ambos tipos de enfoques son independientes del dominio y pueden producir predicciones precisas siempre que las librerías de planes son completas en el caso de los enfoques de consistencia y que las probabilidades son correctas en el caso de los enfoques probabilísticos. Sin embargo, si las observaciones hasta el momento permiten explicar más de una posible intención, los enfoques probabilísticos son capaces de encontrar la intención más probable mientras que los enfoques de consistencia deben esperar por una única explicación consistente ya que no consideran las probabilidades a priori de los diferentes planes. No obstante, ningún

enfoque previo es capaz de tratar con todos los puntos expuestos previamente en esta sección.

En este trabajo proponemos un modelo de intenciones del usuario capaz de tratar con incertidumbre, múltiples planes, múltiples objetivos intercalados, tareas sobrecargadas, tareas espurias, interrupciones y capaz de adaptarse a un usuario particular.

3. Modelo de las intenciones del usuario

Proponemos la definición de un modelo de intenciones que captura tanto la información específica de la aplicación (descrita en esta sección) como las preferencias y hábitos del usuario en el uso de la aplicación (explicado en la sección 4). Este modelo de intenciones puede ser utilizado por un agente de interfaz para mejorar y personalizar su interacción con los usuarios, y para adaptar su comportamiento a los requerimientos específicos de cada usuario en particular.

En general, los usuarios ejecutan tareas para alcanzar un objetivo de más alto nivel. Por ejemplo, para organizar una reunión con sus amigos, el usuario debe mirar el calendario para buscar una fecha conveniente, debe agregar un nuevo evento a su agenda y escribir un mail de invitación a cada participante de la reunión. Por lo tanto, cuando un usuario ejecuta una tarea, el agente debería considerar el conjunto de las posibles intenciones asociadas a la misma, con el objetivo de encontrar una forma de colaborar con él. La predicción de la intención de un usuario es una tarea que posee una incertidumbre inherente. Seguimos un enfoque centrado en redes de Bayes, ya que son una representación de conocimiento capaz de capturar y modelar dinámicamente la incertidumbre existente en las interacciones usuario-aplicación.

Representamos las posibles intenciones en el dominio en un Grafo de Intenciones. Un Grafo de Intenciones representa un contexto de ejecución de tareas y está materializado por una red de Bayes. El contexto es visto como el conjunto de tareas que el usuario ejecutó recientemente, e influirá en la confianza que el agente tiene en cada posible intención que el usuario pueda tener para ejecutar dichas tareas.

En un Grafo de Intenciones, los nodos de la red representan tareas que el usuario puede ejecutar en una aplicación de software, y posibles intenciones del usuario. Las relaciones dirigidas entre nodos representando tareas denotan el hecho de que la ejecución de una tarea tiene influencia en la ejecución de la otra. Por otro lado, un arco dirigido

entre un nodo de tipo intención y un nodo de tipo tarea indica que el usuario probablemente ejecutará dicha tarea cuando esté siguiendo el objetivo en cuestión. Veamos, por ejemplo el simple Grafo de Intenciones presentado en la Figura 1, en el cual los nodos etiquetados con fondo negro y tipografía blanca representan nodos de tipo intención y los nodos etiquetados con fondo blanco y tipografía negra representan nodos de tipo tarea. Existen arcos desde las tareas "SeleccionaContacto" a las tareas "NuevaReunion" y "EscribirMailAContacto". De esta forma, el agente podrá ser capaz de calcular la probabilidad de que el usuario ejecute alguna de estas dos tareas dado que anteriormente ejecutó la tarea predecesora ("SeleccionaContacto"). También existen arcos desde el nodo de intención "AgregarContactoAReunion" hacia los nodos de tarea "NuevaReunion" y "SeleccionaContacto", indicando que cuando la intención del usuario es agregar un contacto a una reunión, probablemente ejecutará las tareas "SeleccionaContacto" y "NuevaReunion". De forma similar, cuando el usuario tiene la intención de enviar un mail (representada por el nodo de intención "EnviarMail"), probablemente ejecutará las tareas correspondientes a los nodos "SeleccionaContacto" y "EscribirMailAContacto".

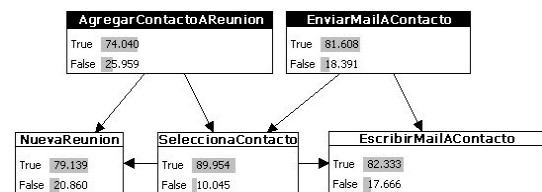


Figure 1. Ejemplo de Grafo de Intenciones en el dominio de una agenda.

Llamamos *nivel de confianza* a la probabilidad de un nodo (de tarea o de intención) de estar en el estado "True", dadas las tareas ejecutadas por el usuario. En otras palabras, cada nodo dentro del Grafo de Intenciones tendrá asociado un nivel de confianza calculado utilizando los mecanismos de actualización de creencias de las redes de Bayes, dadas las tareas evidenciadas. En términos de redes de Bayes, *evidencia* es una colección de hallazgos acerca de los valores que tomaron algunas variables. En nuestro caso, tendremos evidencia que el usuario ejecutó una tarea particular. Los niveles de confianza permitirán al agente ordenar probabilísticamente la posible intención que tiene el usuario y las posibles siguientes tareas

3.1 Tareas sobrecargadas, múltiples objetivos intercalados y múltiples planes

Una tarea que el usuario ejecuta en la aplicación puede formar parte de más de un objetivo posible. En el Grafo de Intenciones presentado en la Figura 1, “SeleccionaContacto” es una tarea sobrecargada, ya que puede servir para alcanzar las dos intenciones representadas en el mismo.

Las redes de Bayes nos permiten modelar este tipo de tareas directamente, ya que la evidencia se agrega a nodos individuales y hace que algunos nodos se vuelvan más probables que otros, sin descartar nodos “inconsistentes”. Las tareas y las intenciones que no tienen soporte dado la evidencia simplemente tendrán sus probabilidades marginales. Además podemos ordenar probabilísticamente las intenciones cuando el usuario ejecuta una tarea sobrecargada. Luego, este orden de intenciones será actualizado de acorde cuando haya disponible nueva evidencia acerca de las tareas ejecutadas por el usuario.

La Figura 2 muestra un ejemplo de esta situación con evidencia sobre una tarea sobrecargada en la Figura 2.a, y evidencia subsiguiente en otra tarea en la figura 2.b. Comparando estas figuras con la Figura 1 podemos ver un incremento en los niveles de confianza de ambas intenciones. Sin embargo, el objetivo “EnviarMailAContacto” tiene un nivel de confianza más alto hasta que el usuario ejecuta “NuevaReunion”. Si el usuario luego ejecuta “EscribirMailAContacto” como se muestra en la Figura 2.c, probablemente intenta alcanzar ambos objetivos, dado que los niveles de confianza de ambos nodos de intención son altos.

Como ejemplo de múltiples planes para alcanzar un objetivo, consideremos que los nodos de intención en la Figura 1 tienen un nodo de intención padre llamado “TrabajarConUnContacto”, como se muestra en la Figura 3. Este objetivo puede lograrse tanto agregando un contacto a una reunión, o enviando un mail a un contacto. Podemos modelar las relaciones probabilísticas entre estos dos objetivos con las TPCs correspondientes en la red de Bayes. Si conocemos por medio de feedback del usuario que intenta trabajar con un contacto seleccionado, el sub-objetivo más probable para lograr su intención puede obtenerse desde el Grafo de Intenciones.

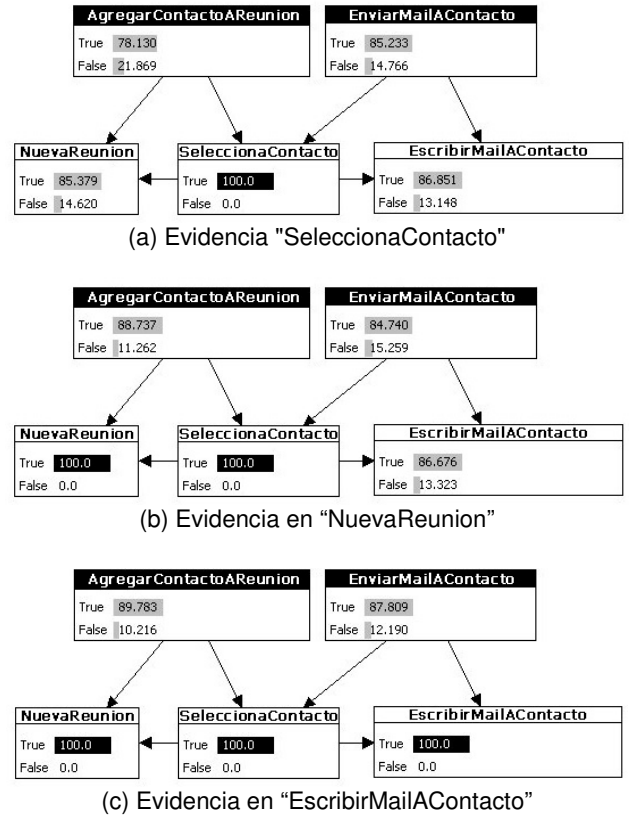


Figura 2. Actualización de creencias cuando se observa una tarea sobrecargada

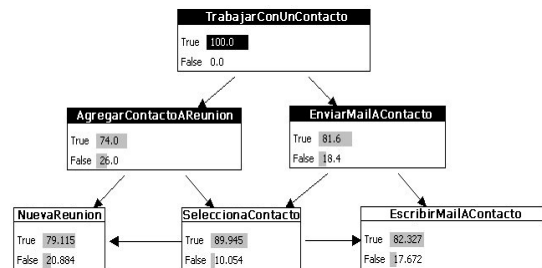


Figura 3. Ejemplo de múltiples planes

3.2 Tareas espurias y cambios los planes del usuario

El soporte para tareas sobrecargadas y cambios en los planes del usuario se logra utilizando un enfoque de desvanecimiento. Cuando el agente observa la ejecución de una tarea, introduce una evidencia dura (hard evidence) en el grafo de intención. El concepto de evidencia dura corresponde al ejemplo de evidencia introducido anteriormente, es decir, que sabemos con 100% de certeza que una variable tomó un determinado valor. Sin embargo, a medida que el usuario ejecuta tareas subsecuentes, nos olvidamos gradualmente de las observaciones pasadas, reduciendo la probabilidad de la evidencia actual de acuerdo a una función de desvanecimiento. El uso de probabilidades para establecer evidencia se conoce como evidencia blanda (soft evidence). Dejamos de olvidar la evidencia cuando ésta alcanza el nivel de confianza que tendría si la evidencia no estuviera presente en la red.

La función de desvanecimiento más simple que puede utilizarse es reducir la probabilidad de la misma por un factor constante cada vez que el usuario ejecuta una tarea. Este enfoque nos permitirá ignorar, después de una cierta cantidad de

tareas ejecutadas por el usuario, una tarea espuria previamente ejecutada o un objetivo que dejó de seguirse. Un potencial problema con esta función, sin embargo, es que puede hacer olvidar rápidamente la ejecución de una tarea que sí contribuye al objetivo actual del usuario. Podemos superar este problema con una función más compleja que mantenga casi intacta la evidencia de un cierto número de tareas y luego reduzca rápidamente el nivel de confianza hasta el valor original (es decir, sin evidencia).

La Figura 4 muestra un ejemplo del efecto de una función en la cual reducimos la evidencia por un factor constante de 0.1. En la Figura 4.a tenemos evidencia en "SeleccionaFecha", luego el usuario ejecuta otra tarea (no mostrada en la figura) y la evidencia es reducida a un valor de 0.9. Luego el usuario ejecuta "SeleccionaContacto" (Figura 4.b); la evidencia en "SeleccionaFecha" toma ahora un valor de 0.8. Finalmente, el usuario ejecuta otra tarea; podemos ver en la figura 4.c que la evidencia en "SeleccionaContacto" se reduce a 0.9 y la evidencia en "SeleccionaFecha" se quita de la red, ya que se vuelve menor que el valor de confianza original de este nodo.

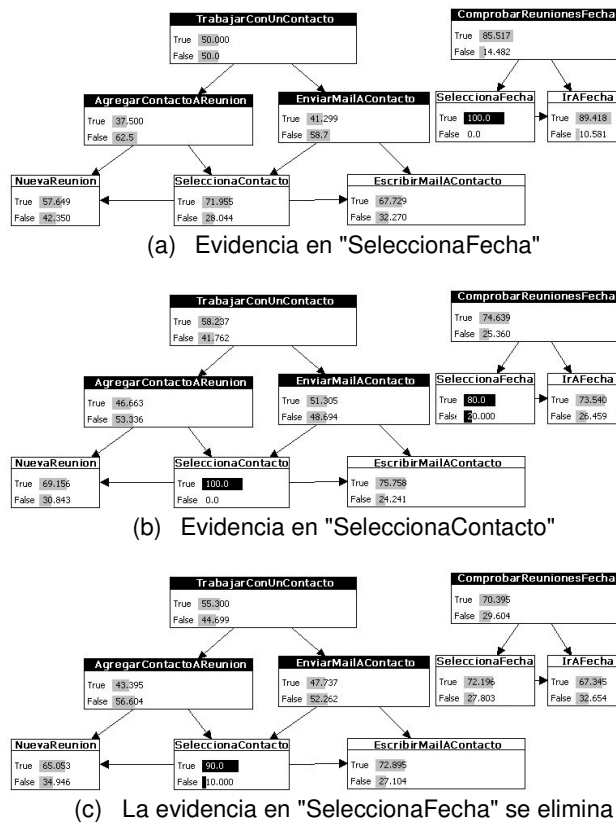


Figure 4. Ejemplo del efecto de una función de desvanecimiento

4. Incorporación de las preferencias del usuario

El modelo de intenciones presentado hasta el momento es construido manualmente durante el diseño del agente. Sin embargo, cuando queremos adaptar el proceso de detección de planes a un usuario en particular, es muy importante tener un modelo apropiado de sus preferencias, ya que éstas son la razón por la que cada usuario actúa de una manera específica cuando se enfrenta a una situación en particular. Por situación entendemos un estado particular de la aplicación, dado por las tareas que el usuario ejecutó y los valores tomados por los atributos relevantes de cada tarea. En el dominio de la agenda, por ejemplo, la situación relacionada a la adición de un nuevo evento puede ser descrita por el día, la hora, los participantes, el lugar y el tema del mismo.

La adaptación del Grafo de Intenciones a un usuario en particular puede hacerse tanto adaptando las probabilidades de las TPCs dado el feedback del usuario, y por medio de aprendizaje de nuevas relaciones que pueden surgir entre los atributos de las tareas ejecutadas por el usuario y los nodos de intención en el Grafo de Intenciones

4.1 Ajuste del Grafo de Intenciones: Aprendizaje de nuevas relaciones

El modo de actuar de un usuario particular puede aprenderse por medio de un enfoque de aprendizaje de máquina inductivo capaz de encontrar regularidades en la decisión del usuario de actuar de un modo particular, y de describirlo en términos de los atributos de la situación, siendo el más importante de ellos la acción tomada. El uso de técnicas de aprendizaje de máquina permitirá al agente adaptarse automáticamente a las particularidades de cada usuario.

Por ejemplo, en el Grafo de Intenciones presentado en la Figura 3, si el usuario selecciona un contacto, el contacto seleccionado puede ser relevante en la intención que tenga el mismo (invitarlo a una reunión o enviarle un mail). Así, mientras el usuario utiliza la aplicación en una primera fase de entrenamiento, el agente almacena casos en la forma

```
<Grupo, Ciudad, Pais,
  AgregarContactoAREunion,
  EnviarMailAContacto>
```

Donde “Grupo”, “Ciudad” y “Pais” son los atributos relevantes del contacto y “AgregarContactoAREunion” y

“EnviarMailAContacto” son las posibles intenciones que el usuario puede estar siguiendo. Disponiendo de esta base de experiencias, utilizamos un algoritmo de aprendizaje de redes de Bayes a partir de datos, tal como K2 [Cooper y Herskovits, 1992] para encontrar relaciones causales entre los parámetros y las posibles intenciones del usuario. Luego, aplicamos un algoritmo de aprendizaje paramétrico, tal como EM [Lauritzen, 1995], para aprender las probabilidades correspondientes a las TPCs a partir de la misma base de ejemplos.

La Figura 5 muestra un ejemplo de relaciones causales aprendidas a partir de un conjunto de ejemplos de entrenamiento basados en los valores tomados por los atributos de la tarea “SeleccionaContacto”. Los nodos etiquetados con fondo gris corresponden a atributos de una tarea.

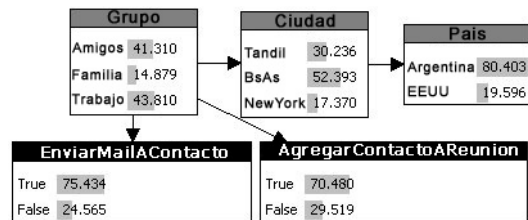


Figura 5. Red de Bayes aprendida a partir de ejemplos de entrenamiento

Una vez aprendido este modelo del usuario, tiene que ser combinado con el Grafo de Intenciones para que sea considerado cuando el agente quiera detectar la intención del usuario. Las TPCs de los nodos que están presentes en ambas redes tienen que ser actualizadas. En general, supongamos que X es un nodo que está presente en ambas redes. Z_1, Z_2, \dots, Z_m son los padres X (Pa(X)) en una red y Y_1, Y_2, \dots, Y_n son los Pa(X) en la otra red. X tiene K_x estados, cada Z_i tiene K_{z_i} estados y cada Y_i tiene K_{y_i} estados. Para una configuración dada de X y Pa(X), X_j ($1 \leq j \leq K_x$), $Z_{jZ_i}^1$ ($1 \leq jZ_i \leq K_{z_i}$), $Y_{jY_i}^1$ ($1 \leq jY_i \leq K_{y_i}$), la entrada correspondiente en la TPC de X será actualizada de acuerdo a la siguiente ecuación:

$$P(X_j | Y_{jY_1}^1, \dots, Y_{jY_n}^1, Z_{jZ_1}^1, \dots, Z_{jZ_m}^1) = P(X_j | Y_{jY_1}^1, \dots, Y_{jY_n}^1) * (1 - \pi) + P(X_j | Z_{jZ_1}^1, \dots, Z_{jZ_m}^1) * (\pi)$$

donde π es un factor que sopesa la importancia que damos a la red aprendida con respecto al Grafo de Intenciones.

4.2 Feedback del usuario: Adaptación de las probabilidades de los nodos

Cuando el Grafo de Intenciones es construido, las probabilidades condicionales establecidas por el diseñador pueden no ser precisas para todos los usuarios de la aplicación. Por otra parte, cuando el usuario utiliza la aplicación, se observan nuevas situaciones y es deseable que el agente pueda aprender de ellas.

Para adaptar las probabilidades condicionales de los nodos en el Grafo de Intenciones, seguimos el enfoque estadístico presentado en [Jensen, 2001] llamado *Fractional Updating*. Para ello, necesitamos del feedback del usuario acerca de qué intención tiene en un momento dado. De esta forma, podemos ingresar la evidencia correspondiente en el nodo asociado a su intención e invocar al algoritmo de adaptación para que aprenda acerca de la nueva situación que corresponde a la evidencia presente en el Grafo de Intenciones en ese momento.

5. Conclusiones

El problema de reconocimiento de planes, es decir el problema de deducir el plan de un usuario a partir de la observación de las tareas que ejecuta, es principalmente un problema de inferencia bajo condiciones de incertidumbre. Si bien existen enfoques no probabilísticos al problema, tienen el problema de no poder decidir hasta qué grado la evidencia soporta cada hipótesis de plan particular. Este es un punto importante a considerar de tal forma que el agente pueda ser capaz de ordenar las posibles explicaciones soportadas por las tareas ejecutadas por el usuario.

Por otro lado, nuestro modelo combina información acerca de las intenciones generales que el usuario puede seguir en la aplicación con información específica de un usuario en particular respecto a sus intenciones. Esta es una cuestión importante a considerar cuando se diseña un sistema de reconocimiento de planes que será utilizado por un agente de interfaz y que frecuentemente no es tenida en cuenta por enfoques generales al problema.

Referencias

- [Azarewicz et al., 1989] Azarewicz, J., Fala, G., and Heithecker, C. (1989). Template-based multi-agent plan recognition for tactical situation assessment. In Fifth IEEE Conference on Artificial Intelligence Applications. [Brown, 1998] Brown, S. (1998). A Decision Theoretic Approach for Interface Agent Development. PhD thesis, Faculty of the Graduate School of Engineering of the Air Force Institute of Technology Air University.
- [Charniak and Goldman, 1993] Charniak, E. and Goldman, R. P. (1993). A Bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53-79.
- [Cooper and Herskovits, 1992] Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Mach. Learn.*, 9(4):309-347.
- [Davison and Hirsh, 1998] Davison, B. D. and Hirsh, H. (1998). Predicting sequences of user actions. In *Predicting the Future: AI Approaches to Time-Series Problems*, pages 5-12, Madison, WI. AAAI Press. Proceedings of AAAI-98/ICML-98 Workshop, published as Technical Report WS-98-07.
- [Geib, 2004] Geib, C. W. (2004). Assessing the complexity of plan recognition. In McGuinness, D. L. and Ferguson, G., editors, AAAI, pages 507-512. AAAI Press / The MIT Press.
- [Gorniak, 2000] Gorniak, P. (2000). Keyhole state space recognition with applications to user modeling. Master's thesis, University of British Columbia.
- [Grosz and Kraus, 1999] Grosz, B. J. and Kraus, S. (1999). The evolution of shared-plans. In Rao, A. and Wooldridge, M., editors, *Foundations and Theories of Rational Agencies*, pages 227-262.
- [Grosz and Sidner, 1990] Grosz, B. J. and Sidner, C. (1990). Plans for discourse. *Intentions and Communications*, pages 417-444.
- [Horvitz, 1999] Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 159-166, New York, NY, USA. ACM Press.
- [Horvitz et al., 1998] Horvitz, E., J., B., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The lumière project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 256-265, Madison, WI.
- [Huber and Durfee, 1993] Huber, M. and Durfee, E. (1993). Observational uncertainty in plan recognition among interacting robots. *International Joint Conference on AI*

- (IJCAI).Workshop of Dynamically Interacting Robots, pages 68-75.
- [Huber and Simpson, 2004] Huber, M. and Simpson, R. (2004). Recognizing the plans of screen reader users. In Modeling Other Agents from Observations (MOO 2004). Workshop W3 at the International Joint Conference on Autonomous Agents and Multi-Agent Systems, Columbia University, NY, USA.
- [Jensen, 2001] Jensen, F. V. (2001). Bayesian Networks and Decision Graphs. Springer Verlag, New York.
- [Kautz and Allen, 1986] Kautz, H. and Allen, J. (1986). Generalized plan recognition. In AAAI, pages 32-37.
- [Koehn and Greer, 1995] Koehn, G. M. and Greer, J. E. (1995). The peculiarities of plan recognition for intelligent tutoring systems. M. Bauer, editor, Working notes from the IJCAI workshop 'The next generation of plan recognition systems', pages 54-59.
- [Lauritzen, 1995] Lauritzen, S. L. (1995). The EM algorithm for graphical association models with missing data. *Comput. Stat. Data Anal.*, 19(2):191-201.
- [Lesh, 1999] Lesh, N.; Rich, C. S. C. (1999). Using plan recognition in human-computer collaboration. In Seventh International Conference on User Modeling, Ban Canada.
- [Lesh, 1998] Lesh, N. (1998). Scalable and Adaptive Goal Recognition. PhD thesis, University of Washington.
- [Lesh and Etzioni, 1995] Lesh, N. and Etzioni, O. (1995). A sound and fast goal recognizer. In Fourteenth International Joint Conference on Artificial Intelligence, pages 1704-1710, San Francisco, CA. Morgan Kaufmann.
- [Lieberman and Selker, 2000] Lieberman, H. and Selker, T. (2000). Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal*, 39(3 & 4):617-632.
- [Maes, 1994] Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30-40.
- [Pearl, 1988] Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann Publishers, San Mateo, California.
- [Rich et al., 2001] Rich, C., Sidner, C. L., and Lesh, N. (2001). Collagen: Applying collaborative discourse theory to human-computer interaction. *Artificial Intelligence Magazine*, 22(4):15-25.