

Sistemas expertos y lógica difusa

Expert systems and fuzzy logic

Juan Felipe Bermúdez Andrade

Facultad de ingenierías, Universidad Tecnológica de Pereira, Pereira, Colombia

Correo-e: felipe98@utp.edu.co

Resumen— Debido a la gran demanda de almacenar y analizar una gran diversidad de información, los sistemas expertos se destacan porque han sido diseñados para facilitar tareas en múltiples campos de aplicación y proporcionar resultados muy profesionales emulando la capacidad humana de tomar decisiones.

En conjunto se trabajará la lógica difusa, que es pocas palabras es una extensión de la lógica tradicional que usa conceptos más parecidos a la manera de como piensa el humano. Desarrollada por Lofti Zadeth a través de la teoría de conjuntos difusos.

Palabras clave — Sistema experto, inteligencia artificial, decisiones, lógica humana, realidad, precisión, conjuntos.

Abstract— Due to the great demand to store and analyze a great diversity of information, expert systems stand out because they have been designed to facilitate tasks in multiple fields of application and provide very professional results emulating the human capacity to make decisions.

We will work with fuzzy logic, which is a few words, an extension of traditional logic that uses concepts that are more like the way humans think. Developed by Lofti Zadeth through the theory of fuzzy sets.

Key Word — Expert system, artificial intelligence, decisions, human logic, reality, precision, sets.

I. INTRODUCCIÓN

El término inteligencia artificial se refiere a la capacidad de emular las funciones inteligentes del cerebro humano, por lo tanto, los sistemas expertos se pueden considerar como un subconjunto de la IA. Un sistema experto es un sistema que emplea conocimiento humano capturado en un computador para resolver problemas que normalmente requieran de humanos expertos. Los sistemas bien diseñados imitan el proceso de razonamiento que los expertos usarían para resolver una problemática. Una ventaja de una buena implementación de un sistema es que puede ser usado por no-expertos para ayudarlo a mejorar sus habilidades o ser usado como asistente de un experto.

La lógica difusa tiene su origen en la década de los 60, en la universidad de Berkeley – California, la cual fue desarrollada por Lofti Zadeth a través de la teoría de conjuntos difusos. La lógica difusa surgió como una generalización de la lógica clásica, por el deseo de representar de manera mas fiel la realidad. Esta es usada para representar la información imprecisa, ambigua o vaga, para tomar decisiones razonables en un entorno de incertidumbre.

II. CONTENIDO

1. ¿Qué es un sistema experto?

Es un sistema computacional que imita la toma de decisiones de un humano experto en un tema específico. Están compuestos por una Base de Conocimientos (BC), una Base de Hechos (BH) y un Motor (o Máquina) de Inferencias (MI). Estos sistemas tienen ventajas técnicas respecto a un ser humano, como la rapidez, la permanencia, la fiabilidad, el bajo costo entre otros, pero presentan desventajas como la carencia de sentido común, la comunicación nula, la falta de flexibilidad, etc.

2. Concepto SE

Sistema que busca conclusiones a partir de hechos aplicando reglas.

Hechos:

- Llueve
- Futbol
- Cine

Reglas:

- R1. Llueve -> Cine
- R2. No llueve -> Futbol

Maquina:

Explora las reglas buscando concordancia de hechos

Llueve -> cine
Hecho -> Conclusión

3. Arquitectura SE

Los siguientes son los componentes básicos de un Sistema experto:

- Subsistema de adquisición de conocimiento: Es la acumulación, transferencia y transformación de la experiencia para resolver problemas de una fuente de conocimiento a un programa de computadora para construir o expandir la base de conocimiento.
- Base de conocimiento: Contiene el conocimiento necesario para comprender, formular y resolver problemas.

- c. Base de hechos: Es una memoria de trabajo que contiene los hechos sobre un problema, alberga los datos propios correspondientes a los problemas que se desean tratar.
- d. Motor de inferencia: Es el cerebro del sistema experto, también conocido como interpretador de reglas. Este componente provee metodologías para razonamiento de información en la base de conocimiento. Tiene un intérprete, programador, control de consistencia.
- e. Subsistema de justificación: Se encarga de explicar el comportamiento del sistema experto al encontrar una solución.

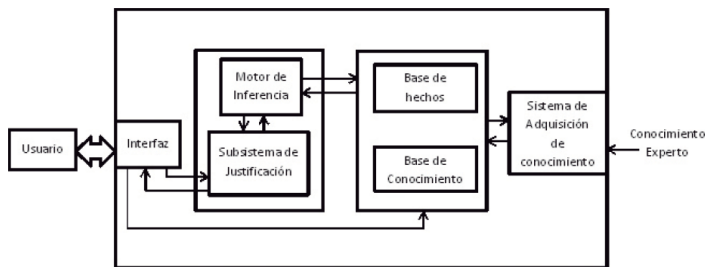


Figura 1. Arquitectura de un SE

4. CLIPS

A mediados de los años ochenta, la NASA requería el apoyo de Sistemas expertos para el desarrollo de proyectos. Por lo tanto, una serie de prototipos surgen, pero sus resultados no fueron lo suficientemente buenos para cumplir con los requerimientos internos. En consecuencia, se desarrolló un prototipo de un Sistema Experto, denominado CLIPS (C Language Integrated Production System) cuya principal característica era su capacidad para funcionar con otros sistemas existentes. Posteriores mejoras y ampliaciones han convertido CLIPS en un punto de referencia para el desarrollo de otros Sistemas Expertos (CLIPS, 1994).

Inicialmente, con el desarrollo de CLIPS se buscaba tener mayor conocimiento sobre la construcción de sistemas expertos y sentar las bases de un lenguaje para reemplazar las herramientas comerciales que estaban siendo usadas. La versión 1.0 de CLIPS demostró que eso era posible. Después de un desarrollo adicional, se vio que el costo de CLIPS sería significativamente menor al de otras herramientas y que sería ideal para entrenamiento. Otro año de desarrollo y de uso interno sirvió para mejorar portabilidad, desempeño, funcionalidad y documentación de soporte. A mediados de 1986, CLIPS v3.0 estuvo disponible para grupos fuera de la NASA.

Desarrollo y Expansión:

Otras funcionalidades agregadas transformaron CLIPS; ya no era una herramienta para entrenamiento sobre construcción de sistemas expertos, sino que ahora servía también para el desarrollo y ejecución de estos. Las versiones CLIPS v4.0 y v4.1 (1987) tenían significativas mejoras en cuanto a desempeño,

integración con otros lenguajes, y capacidad de ejecución. CLIPS v4.2 (1988) fue una completa reescritura del código fuente con el fin de hacerlo más modular. Esta versión también incluyó un manual detallado de la arquitectura de CLIPS y una aplicación de ayuda para la verificación y validación de programas basados en reglas. Nuevas funcionalidades vinieron con la CLIPS v4.3 (1989). Originalmente, CLIPS era un lenguaje de reglas basado en el Algoritmo Rete (Programación Lógica). CLIPS v5.0 (1991) introdujo dos nuevos paradigmas de programación:

Programación Imperativa y Programación Orientada a Objetos (POO). El lenguaje POO dentro de CLIPS es llamado COOL (CLIPS Object-Oriented Language, o Lenguaje Orientado a Objetos de CLIPS). CLIPS v5.1 (1991) ya soportaba las recientemente desarrolladas o mejoradas interfaces X Windows, MS-DOS y Macintosh. CLIPS v6.0 (1993) tenía nuevas funcionalidades relacionadas con el reconocimiento de patrones en objetos/reglas y soporte a Ingeniería de Software basada en reglas. CLIPS v6.1 (1998) soportaba compiladores C++, aunque ya no soportaba los viejos compiladores C no ANSI. También se agregaron comandos para llevar control del tiempo de desarrollo y para funciones definidas por el usuario. CLIPS v6.2 es soportada por diversos sistemas operativos, y tiene mejoras en su interfaz de desarrollo para Windows 95/98/NT y MacOS. Actualmente, CLIPS es mantenido fuera de la NASA como software de dominio público.

Desarrollador:

Gary Riley tiene más de treinta años de experiencia en el campo de los sistemas expertos. Mientras trabajaba en el Centro Espacial Johnson de la NASA, fue responsable del diseño y desarrollo de los componentes basados en reglas de CLIPS. Desde que dejó la NASA, ha continuado desarrollando y manteniendo de forma independiente una versión de dominio público de CLIPS. Es el coautor del libro de texto 'Sistemas expertos: Principios y programación', ahora en su cuarta edición. Además de crear herramientas de sistemas expertos, también ha sido un usuario experto de herramientas de sistemas, desarrollando y manteniendo tres sistemas expertos durante un período de diez años.

Características Principales:

Representación del Conocimiento: CLIPS permite manejar una amplia variedad de conocimiento, soportando tres paradigmas de programación: el declarativo, el imperativo, y el orientado a objetos. La programación lógica basada en reglas permite que el conocimiento sea representado como reglas heurísticas que especifican las acciones a ser ejecutadas dada una situación. La POO permite modelar sistemas complejos como componentes

modulares. La programación imperativa permite ejecutar algoritmos de la misma manera que en C, Java, LISP y otros lenguajes.

Portabilidad: CLIPS fue escrito en C con el fin de hacerlo más portable y rápido, y ha sido instalado en diversos sistemas operativos (Windows 95/98/NT, MacOS X, Unix) sin ser necesario modificar su código fuente. CLIPS puede ser ejecutado en cualquier sistema con un compilador ANSI de C, o un compilador de C++. El código fuente de CLIPS puede ser modificado en caso de que el usuario lo considere necesario, con el fin de agregar o quitar funcionalidades.

Integralidad: CLIPS puede ser embebido en código imperativo, invocado como una subrutina, e integrado con lenguajes como C, Java, FORTRAN y otros. CLIPS incorpora un completo lenguaje orientado a objetos (COOL) para la elaboración de sistemas expertos. Aunque está escrito en C, su interfaz más próxima se parece a LISP. Pueden escribirse extensiones a CLIPS sobre C, y, al contrario, CLIPS puede ser llamado desde C. CLIPS puede ser extendido por el usuario mediante el uso de protocolos definidos.

Desarrollo Interactivo: La versión estándar de CLIPS provee un ambiente de desarrollo interactivo y basado en texto; este incluye herramientas para la depuración, ayuda en línea, y un editor integrado. Las interfaces de este ambiente tienen menús, editores y ventanas que han sido desarrollados para MacOS, Windows 95/98/NT, X Windows, entre otros.

Verificación/Validación: CLIPS contiene funcionalidades que permiten verificar las reglas incluidas en el sistema experto que está siendo desarrollado, incluyendo diseño modular y parcialmente de la base de conocimientos del sistema, chequeo de restricciones estático y dinámico para funciones y algunos tipos de datos, y análisis semántico de reglas para prevenir posibles inconsistencias.

Documentación: En la página web oficial de CLIPS se encuentra una extensa documentación que incluye un Manual de Referencia y una Guía del Usuario.

Bajo Costo: CLIPS es un software de dominio público

5. ¿Qué es la lógica difusa?

La lógica difusa se basa en lo relativo de lo observado como posición diferencial. Este tipo de lógica toma dos valores aleatorios, pero contextualizados y referidos entre sí. Se basa en reglas heurísticas de la forma SI (Antecedente) ENTONCES (Consecuente), siendo el antecedente y el consecuente conjuntos difusos. Este tipo de lógica presenta valores intermedios, no solo 0 y 1.

6. Modelo LD



III. CONCLUSIONES

Los Sistemas Expertos han sido diseñados para facilitar las tareas en múltiples campos de aplicación y proporcionar equivalentes resultados a los de un experto humano. Se sabe que no son aptos para resolver problemas generales pero su eficiencia aumenta mientras más acotado o específico sea el dominio de aplicación. Actualmente los Sistemas Expertos son empleados en áreas diversas y existe una importante variedad de herramientas y tecnologías para implementarlos.

Como todo sistema, poseen fortalezas y debilidades, no obstante, y en términos generales, por su flexibilidad, confiabilidad y escalabilidad se los puede considerar como una tecnología de probada efectividad y lo suficientemente madura para confiar decisiones de considerable criticidad.

La lógica difusa se utiliza cuando la complejidad del proceso en cuestión es muy alta y no existen modelos matemáticos precisos, para procesos altamente no lineales y cuando se envuelven definiciones y conocimiento no estrictamente definido.

En cambio, no es una buena idea usarla cuando algún modelo matemático ya soluciona eficientemente el problema, cuando los problemas son lineales o cuando no tienen solución.

REFERENCIAS

1. Sistemas Expertos: Fundamentos, Metodologías y aplicaciones. Sebastián Badaró, Leonardo Javier Ibañez y Martín Jorge Agüero.
2. <http://profesores.elo.utfsm.cl/~tarredondo/info/soft-comp/Introduccion%20a%20la%20Logica%20Difu%20sa.pdf>
3. https://www.academia.edu/31151686/L%C3%B3gica_difusa_Fuzzy_logic_Inteligencia_Artificial_Integrante

