

FuzzyF – Fuzzy Logic Framework : Uma Solução Software Livre para o Desenvolvimento, Ensino e Pesquisa de Aplicações de Inteligência Artificial Multiplataforma

João Ricardo Bittencourt¹, Fernando Osório¹

¹Mestrado em Computação Aplicada – Centro de Ciências Exatas e Tecnológicas
Universidade do Vale do Rio dos Sinos (UNISINOS)

Caixa Postal 275 – 93.022-000 – São Leopoldo – RS – Brasil

jrbit@netu.unisinos.br, osorio@exatas.unisinos.br

Abstract. This work describes FuzzyF – Fuzzy Logic Framework, a framework developed in order to facilitate and to improve the design and implementation of portable Fuzzy Logic based applications. The implemented tools allow a rapid application development of Fuzzy Expert Systems. We also discuss the importance of free software development and distribution to the scientific community, where FuzzyF plays an important role, as an open and portable tool that can be very useful for teaching, researching and developing new applications based on concepts from this Artificial Intelligent research field.

Resumo. Este trabalho descreve a implementação de um framework (conjunto de classes) destinado a facilitar o desenvolvimento de aplicações multiplataforma baseadas na Lógica Nebulosa (Fuzzy Logic), denominado de FuzzyF – Fuzzy Logic Framework. Esta implementação permite um rápido desenvolvimento de aplicações que se utilizam de Sistemas Especialistas baseados na Lógica Nebulosa. Também é destacada a importância do desenvolvimento e da distribuição de softwares livres junto a comunidade científica, onde esperamos que este trabalho, pelas suas características de software aberto e multiplataforma, possa contribuir para um maior desenvolvimento do ensino e da pesquisa no campo da Inteligência Artificial.

1. Introdução

A Lógica Nebulosa (Fuzzy Logic), surgiu dos estudos desenvolvidos por L. Zadeh[1] sobre os conjuntos nebulosos (Fuzzy Sets), e tem merecido um lugar de destaque junto as pesquisas da área de Inteligência Artificial (I.A.). A lógica nebulosa possibilita que seja abordado de forma mais adequada um problema muito importante referente à representação e manipulação de conhecimentos em I.A.: a representação da imprecisão e da incerteza em informações (vide Fig 1).

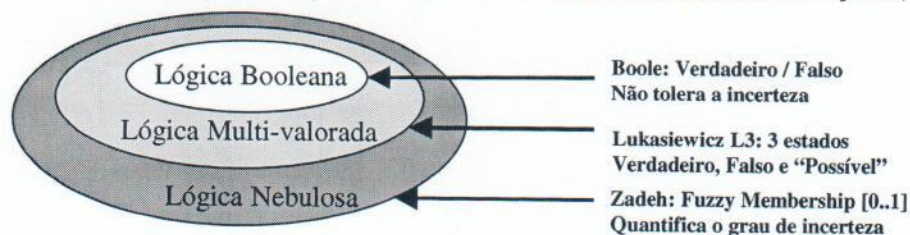


Figura 1 - Lógica de Boole, Lukasiewicz e a Lógica Nebulosa de Zadeh

A lógica nebulosa tem sido aplicada com sucesso em diversos trabalhos de pesquisa nas Universidades e também junto à indústria [10], onde atualmente podemos encontrar produtos como câmeras fotográficas, aparelhos de ar condicionado e lavadoras de roupa que utilizam a tecnologia fuzzy. O uso crescente de aplicações baseadas na lógica nebulosa tem

levado a um aumento do interesse nesta área, seja junto à comunidade acadêmica, ou junto às empresas privadas. Constatamos assim a necessidade de disponibilizar a comunidade científico-acadêmica uma ferramenta que permita o estudo e desenvolvimento de pesquisas em aplicações baseadas na lógica nebulosa.

Considerando-se esta necessidade da comunidade científico-acadêmica, levantou-se a seguinte lista de requisitos para o desenvolvimento de um software aplicado ao ensino e pesquisa na área de lógica nebulosa:

- (a) Criar um **Código Livre e Aberto**, baseado na licença GNU GPL [6]: softwares acadêmicos e de pesquisa devem ser gratuitos facilitando o seu emprego e disseminação junto aos alunos; além disto devem ser abertos, pois permitem que se possa conhecer (ensino) e alterar (pesquisa) o seu código, bem como aumentar o intercâmbio e a troca de conhecimentos entre os usuários do sistema;
- (b) Criar um **Código Multi-Plataforma**: as instituições de pesquisa e ensino devem poder usar o software independente do tipo de plataformas instaladas, assim como os alunos também devem ter a liberdade de escolha do tipo de Sistema Operacional que desejam utilizar para o desenvolvimento de seus projetos;
- (c) Criar um **Código Extensível**: o software deve permitir que sejam criadas novas funções, ou que as existentes possam ser personalizadas pelo usuário, de acordo com as suas necessidades;

2. Lógica Nebulosa

Os sistemas nebulosos (*fuzzy logic systems*) foram criados por Lotfi A. Zadeh nos anos sessenta [1,7,8], fundamentando-se basicamente na representação e manipulação de informações incertas e imprecisas tão comuns no cotidiano humano. Expressões tais como “quase”, “muito” e “pouco” representam esta imprecisão, que usualmente não pode ser tratada pelos sistemas da lógica clássica.

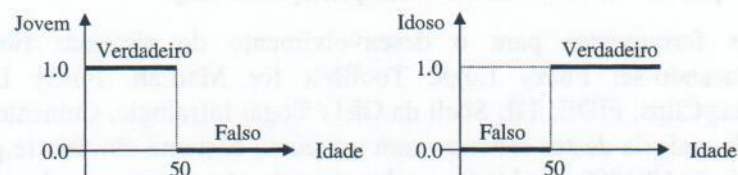


Figura 2 - Lógica Booleana – Exemplo de definição de uma pessoa idosa

A lógica fuzzy é um super conjunto da lógica booleana tradicional (Fig. 1) que estende esta lógica incluindo o conceito de parcialmente verdadeiro, ou seja, valores entre o completamente verdadeiro (1) e o completamente falso (0). Esta variação é representada por um conjunto nebuloso, através de uma função que gera um número dentro de um intervalo de 0 à 1. Por exemplo, vamos determinar que um indivíduo com mais de 50 anos é idoso (vide Fig. 2). Isto significa que, pela lógica booleana, uma pessoa com 51 anos e outra com 89 anos são ambas idosas. Entretanto isto significa também que uma pessoa com 49.8 anos não seria considerada idosa, e conseqüentemente esta pessoa seria considerada uma pessoa jovem. A questão é: como definir de modo mais adequado se uma pessoa é jovem ou idosa, ou até mesmo uma “mistura de ambos”? Isto pode ser obtido através dos conjuntos nebulosos, onde inicialmente definimos as variáveis linguísticas, que no caso deste exemplo poderia ser a variável *idade*, e depois definimos os conjuntos nebulosos associados a esta variável (*fuzzy sets*), tais como: jovem, adulto e idoso (vide a Fig. 3). Uma variável nebulosa não resultaria em uma única resposta determinística, mas em um conjunto de respostas, sendo que cada posição deste conjunto é o valor para cada termo nebuloso da variável. Por exemplo, uma pessoa com 33 anos seria { 0.07 ; 1.0 ; 0.0 }, ou seja, ainda é pouco jovem (0.07), totalmente mediana (1.0) e não é idosa (0.0).

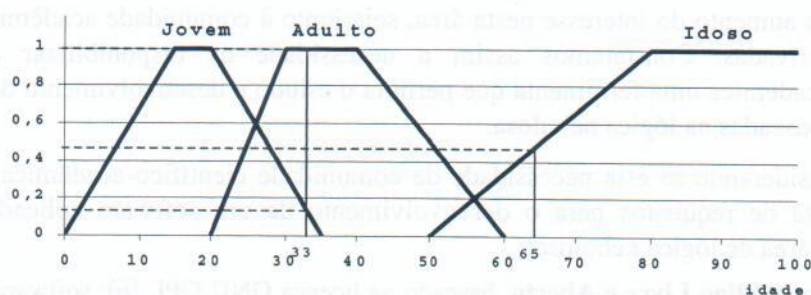


Figura 3-Variável nebulosa (idade) formada por 3 termos nebulosos (jovem, adulto e idoso)

Os sistemas especialistas fuzzy utilizam um conjunto de regras do tipo “If-Then” [2,4], baseadas em variáveis nebulosas. Primeiramente as variáveis de entrada sofrem um processo de *fuzzificação*, ou seja, os conjuntos nebulosos das variáveis lingüísticas de entrada são ativados. Terminado este processo efetua-se a inferência sobre o conjunto de regras nebulosas obtendo os valores dos termos das variáveis de saída. Finalmente, as variáveis de saída sofrem um processo de *defuzzificação*. Este processo consiste em converter os dados nebulosos para valores numéricos precisos. Para isto são utilizadas várias técnicas, tais como valor máximo, média dos máximos, média local dos máximos, centro de gravidade, ponto central da área e o centro da média [3,7,8]. Estes métodos citados anteriormente referem-se ao modelo nebuloso denominado Mamdani [8]. Também existe um outro método chamado Takagi-Sugeno [3,8] que se diferencia do primeiro pelo processo de defuzzificação empregado. Portanto, um sistema fuzzy deve oferecer ao usuário a capacidade de: (i) definir as variáveis e conjuntos nebulosos; (ii) definir regras If-Then e operadores lógicos (*And*, *Or*) que manipulem as variáveis nebulosas; (iii) realizar inferências usando as regras; (iv) oferecer diferentes métodos de defuzzificação das variáveis de saída.

2.1. Ferramentas para o desenvolvimento de aplicações Fuzzy

Existem diversas ferramentas para o desenvolvimento de sistemas fuzzy disponíveis atualmente, destacando-se: Fuzzy Logic ToolBox for MatLab, Fuzzy Logic Pack for Mathematica, FuzzyClips, FIDE, TIL Shell da OEI / Togai Infralogic, Cubicalc da HyperLogic [8,9,10]. A grande maioria destes sistemas tem um custo bastante elevado (e.g. Fuzzy Tlbox + MatLab custa mais de US\$1800 a unidade), onde estes não são em sua grande maioria softwares abertos, além de imporem restrições quanto ao seu uso e distribuição. Consta-se uma grande carência de ferramentas para o desenvolvimento do ensino, pesquisa e implementação de aplicações baseadas em lógica nebulosa, que satisfaçam os requisitos (a), (b) e (c) indicados na introdução deste trabalho. Não temos conhecimento à presente da disponibilidade de alguma ferramenta que satisfaça a todos estes três requisitos.

3. FuzzyF

A partir desta constatação da necessidade de criação de uma ferramenta de *software aberto*, *multi-plataforma* e *extensível* voltada ao desenvolvimento de sistemas fuzzy, foi criado assim o **FuzzyF** – *Fuzzy Logic Framework*. O FuzzyF é um conjunto de classes usadas para representar domínios nebulosos e efetuar inferências sobre este domínio, desenvolvido em Java sob a licença GNU GPL, satisfazendo aos itens (a), (b) e (c) citados acima. Além disto, o FuzzyF oferece as funcionalidades (i), (ii), (iii) e (iv) descritas no item 2 deste trabalho.

As principais funcionalidades do FuzzyF são: possui uma série de funções membros implementadas, tais como gaussiana, triangular, trapezoidal, constante, trapezoidal esquerda e trapezoidal direita; os principais métodos de defuzzificação – centro de gravidade, ponto central da área e a média dos máximos; uso de operadores (*hedge*); gramática para definição de domínios que inclui declaração de variáveis de entrada, intermediárias e de saída com seus

respectivos termos nebulosos e definição de regras do tipo “If-Then” com valores de confiança associados e o motor de inferência para ser utilizado em tempo de execução. Na versão atual deste framework está implementado somente o modelo de defuzzificação Mamdani, mas podemos facilmente estender para o modelo Takagi-Sugeno.

O FuzzyF pode ser utilizado dentro de qualquer aplicação, basta definir um arquivo textual descrevendo o domínio do problema (variáveis de entrada, intermediárias e de saída e as regras – conforme Fig. 4) e ativar o motor de inferência com um vetor contendo os valores de entrada. O motor de inferência fuzzy irá retornar um outro vetor com os valores de saída.

#Variável Idade	#Regras
INPUT_VAR = Idade	BLOCK_RULES
TERM = Jovem TriangularFunction 0 18 35	RL = IF Idade IS Idoso AND Pressao IS Alta
TERM = Adulto TrapezoidalFunction 20 30 45 60	THEN Cuidados_Medicos IS Rigoroso
TERM = Idoso TrapezoidalFunction 50 80 100 150	RL = IF Idade IS Adulto AND Pressao IS Muito_Alta
(...)	THEN Cuidados_Medicos IS Rigoroso (...)
END_VAR	END_RULES

Figura 4 – Exemplo de um domínio de problema (variáveis e regras)

4. Considerações Finais

Conclui-se o presente trabalho destacando a importância do desenvolvimento de uma aplicação de livre distribuição dentro de um cenário com poucas ferramentas desta natureza e com intensa pesquisa neste campo da Inteligência Artificial (I.A.). Atualmente o FuzzyF encontra-se implementado, em fase final de testes e de documentação (consulte o site do FuzzyF em: <http://www.inf.unisinos.br/~jrbitt/fuzzyf/>). O FuzzyF vem complementar o desenvolvimento de uma família de pacotes de ferramentas de I.A. (Free AID-K), que vem sendo implementada junto a Unisinos, onde já contamos com uma outra importante ferramenta implementada, o ANNeF (Artificial Neural Network Framework) [5], criado nos mesmos moldes do FuzzyF.

Referências Bibliográficas

- [1] Zadeh, Lotfi A. “Fuzzy Sets”. Fuzzy Sets, Information and Control, 8:338-353, 1965.
- [2] Zadeh, L. The Calculus of Fuzzy If-Then Rules, AI Expert, V.7(3), 23-27, March 1992.
- [3] Eberhart, Russ; Simpson, Pat; Dobbins, Roy. “Computational Intelligence PC Tools”. AP Professional, Boston, 1996. 464p.
- [4] Nikolopoulos, Chris. “Expert Systems: Introduction to first and second generation and hybrid knowledge-based systems”. New York: Marcel Dekker Press, 1997. 327p
- [5] Bittencourt, João; Osório, Fernando. “ANNeF – Artificial Neural Networks Framework: Uma Solução Software Livre para o Desenvolvimento, Ensino e Pesquisa de Aplicações de inteligência Artificial Multiplataforma”. II WSL – Workshop de Software Livre 2001. Porto Alegre, RS. Maio 2001.
- [6] GNU General Public License, <http://www.fsf.org/licenses/gpl.html> (acessado em Nov./2001).
- [7] NUS–National Univ. of Singapore, PRIS – Program for Research into Intelligent Systems. “SoftComputing GuideBook”. <http://www.comp.nus.edu.sg/~pris/> (acessado em Fev./2002).
- [8] Fuzzy Logic FAQ (Frequently Asked Questions) – Comp.ai.fuzzy newsgroup, <http://www.faqs.org/faqs/fuzzy-logic/> (acessado em Março/2002).
- [9] Ortech OEI. “Fuzzy Logic Reservoir”. <http://www.ortech-engr.com/fuzzy/reservoir.html> (acessado em Março/2002).
- [10] FuzzyNet On-Line. <http://www.aptronix.com/fuzzynet/> (acessado em Março/2002).