



**FEPEG**

FÓRUM DE ENSINO,  
PESQUISA, EXTENSÃO  
E GESTÃO

TRABALHOS CIENTÍFICOS APRESENTAÇÕES ARTÍSTICAS E CULTURAIS DEBATES MINICURSOS E PALESTRAS

23 A 26 SETEMBRO DE 2015  
Campus Universitário Professor Darcy Ribeiro

ISSN 1806-549X

A HUMANIZAÇÃO NA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

REALIZAÇÃO



APÓIO



## DESENVOLVIMENTO E COMPARAÇÃO DE SISTEMAS DE INFERÊNCIA FUZZY PARA AVALIAÇÃO DE TRABALHOS DE CONCLUSÃO DE CURSO

*Luanna Ferreira Neves, Amanda Gabriely Santos Alves, Joana Gabriela Ribeiro de Souza, João Paulo Pereira Nery, Lucas Araújo Borges, Rodrigo Mota Lacerda*

### Introdução

O presente trabalho visa apresentar o desenvolvimento e estudo comparativo entre alguns métodos de *defuzzificação*, utilizando como base o sistema de inferência *Fuzzy* de Mamdani. Para a realização do trabalho foi utilizado a toolbox do Matlab para lógica *Fuzzy* e foram testadas e comparadas diversas funções tanto nas variáveis de entrada e saída e também vários métodos de defuzzificação.

O presente trabalho foi motivado devida a forma como as notas para a avaliação dos trabalhos de conclusão de curso são feitas, no curso de Bacharelado em Sistemas de Informação Unimontes. Atualmente, a banca composta por três professores avalia o trabalho do acadêmico, de acordo com certos requisitos base, e após cada um atribuir sua nota ao trabalho são atribuídos pesos para as notas de cada professor da banca, e então é feito um cálculo sobre as notas para se obter a nota final, que define a aprovação ou não do trabalho. A proposta então do trabalho foi criar um sistema que avaliasse trabalho de projeto de conclusão de curso utilizando lógica *Fuzzy*, e um sistema de inferência Mamdani.

O sistema de inferência fuzzy proposto é capaz de calcular a nota que um trabalho de conclusão de curso, obterá, a partir da atribuição de notas para algumas variáveis linguísticas pré-estabelecidas, definindo o respectivo grau de pertinência de cada uma em relação ao conjunto.

São apresentados alguns conceitos que tomamos como base para o desenvolvimento do sistema e também descrevemos como foi o desenvolvimento e as conclusões após todos os testes, além das referências utilizadas.

### Material e métodos

#### A. Lógica Fuzzy

Na busca por um modelo não-binário e que lidasse com problemas nebulosos, ou seja, incertos, a primeira publicação sobre lógica “fuzzy” foi feita em 1965, quando recebeu este nome. O autor foi Lotfi Asker Zadeh (ZAH-da), professor em Berkeley, Universidade da Califórnia. Zadeh criou a lógica “fuzzy” combinando os conceitos da lógica clássica e os conjuntos de Lukasiewicz, definindo graus de pertinência. Ele observou que recursos tecnológicos, baseados na lógica booleana, não eram suficientes para automatizar atividades relacionadas a problemas de natureza industrial, biológica ou química.

A lógica fuzzy é baseada na teoria de conjuntos não-exatos, chamados de conjuntos fuzzy, e difere dos sistemas tradicionais em várias características e detalhes. A principal característica dessa, é que o raciocínio exato corresponde ao limite do raciocínio aproximado, e é interpretado como uma composição de relações não-exatas ou nebulosas. Ao contrário dos sistemas binários, onde a verdade de uma proposição pode assumir dois valores: verdadeiro (1) ou falso (0), nos sistemas lógicos multi-valores, o valor verdade de uma proposição pode ser um elemento de um conjunto finito ou uma álgebra booleana.

Na lógica nebulosa, os valores verdade de uma proposição são expressos através de termos linguísticos, onde cada um é interpretado como um subconjunto fuzzy do intervalo de 0 a 1. Outra característica da lógica fuzzy que a difere dos sistemas lógicos binários é que, nesses últimos, os predicados são exatos (ex.: par, ímpar, maior que), enquanto na lógica fuzzy os predicados são nebulosos/não-exatos (ex.: alto, baixo, pouco). Também, em contraste com a lógica modal clássica, o conceito de possibilidade é interpretado utilizando-se subconjuntos fuzzy no universo dos reais (Zadeh, 1988). Isso faz com que a lógica fuzzy seja amplamente aplicada em diversos contextos e diversos problemas.

#### B. Fuzzy Logic Toolbox - MATLAB™

O Matlab é um software interativo de alta performance que possui alta performance, integrando análise numérica, cálculo com matrizes, processamento de sinais e construção de gráficos em ambientes interativos e de fácil usos. Nesse



ambiente, os problemas e suas soluções são expressos somente como são descritos matematicamente, ao contrário de programações tradicionais. Seu elemento básico de informação é uma matriz que não requer dimensionamento, permitindo a resolução de muitos problemas numéricos em apenas uma fração de tempo. Esta particularidade do software deve-se ao fato de que o MATLAB é construído na linguagem MATLAB, às vezes chamada M-código ou simplesmente M, que expressa problemas de forma matemática e requer menos tempo para que possa ser escrito, comparando à escrita de programas em C ou linguagem Fortran.

O Fuzzy Logic Toolbox™ é uma extensão do MatLab que fornece funções, aplicativos e blocos para a análise, concepção e simulação de sistemas baseados em lógica difusa. Ele o orienta durante todas as etapas de concepção do sistema de inferência fuzzy, fornecendo funções para muitos métodos comuns, incluindo agrupamento difuso e aprendizagem *neurofuzzy* adaptativa. Desta forma a toolbox permite modelar comportamentos complexos do sistema, usando regras lógicas simples ou complexas e, em seguida, implementando essas regras em um sistema de inferência fuzzy, permitindo seu uso como um motor autônomo de inferências *fuzzy*. Outra alternativa é usar blocos de inferência fuzzy e simular os sistemas dentro de um modelo que abrange todo o sistema dinâmico.

Na *toolbox* é possível definir as variáveis, tipo de variáveis, alcance das variáveis, função de pertinência da variável, funções, regras e etc., definindo de forma precisa o seu sistema de inferência fuzzy através de uma interface interativa e de fácil uso. A figura 1 mostra a apresentação inicial da toolbox para o usuário.

A inferência difusa, é o processo desde a formulação de entradas, até a geração de saída/resposta utilizando a lógica fuzzy. Este método interpreta um vetor de entrada, aplica sobre ele um conjunto de regras definidas anteriormente, atribui valores ao vetor de saída. A formulação desse tipo de sistema envolve a escolha das funções de pertinência, dos operadores lógicos difusos, o desenho das regras difusas, mecanismo de implicação e agregação das regras, a escolha do método de defuzzificação para se obter a saída do sistema. (Jang and Sun, 1996).

Os modelos baseados em regras difusas, aplicam abordagens diferentes para o mecanismo da inferência, estes modelos diferem quanto à forma de representação dos termos da premissa, à representação das ações de controle e aos operadores utilizados na saída.

O modelo difuso Mamdani foi proposto pelo professor Ebrahim H. Mamdani, sendo ele o primeiro pesquisador que fez uma prática da lógica *fuzzy*. Segue o modelo a seguir:

A inferência faz o cálculo “SE” das regras avaliando o grau de pertinência de cada regra para os valores das variáveis de entrada, isso é representado por:  $\mu(n)$  onde  $n$  é a regra avaliada. Nesta fase, geralmente, são utilizados os operadores *min* e produto para avaliar o conector das proposições do lado esquerdo da regra “E”, no cálculo da parte “ENTÃO” da regra partindo do grau de ativação é determinado a conclusão da regra, nesta parte o mecanismo de inferência atribui para cada uma das variáveis de saída, o conjunto difuso, além de isso, modifica o valor de acordo com a função de associação avaliada, a nova função de pertinência é representada como:  $\mu(n)(Y)$  onde  $n$  é a regra avaliada e  $y$  a variável de saída.

Na implicação nebulosa, Mamdani propôs operar os conjuntos utilizando o operador lógico *min*. Dessa forma, caso hajam dois conjuntos  $A$  e  $B$  dos universos de discurso  $U$  e  $V$ , o modelo Mamdani propõe que  $A \rightarrow B$  é uma relação nebulosa em  $U \times B$  se determina como  $\mu(A) \rightarrow (x, y) = \min\{\mu A(x), \mu B(y)\}$ .

A agregação funciona da seguinte forma: dada uma regra, primeiro avalia-se a parte esquerda e depois a direita de acordo com o que foi dito anteriormente, daí são obtidos  $N$  conjuntos difusos de acordo com as  $N$  regras difusas, no final, o conjunto nebuloso de saída é fornecido como uma agrupação dos conjuntos, nesse modelo o operador *max* é utilizado.

A fase de *defuzzificação* é a última etapa do modelo Mamdani, é necessário que a saída seja clara, sem ambiguidades. A determinação do modelo de *defuzzificação* é importante, à medida que define um funcionamento melhor e ou mais concisa para a saída que pode ser usada como entrada de um controlador por exemplo. Estes são alguns dos métodos mais utilizados: Método do Primeiro Máximo (SOM), Método da Média dos Máximos (MOM), Método do Centro da Área. Sendo que o modelo proposto por Mamdani aplica o Método de Centro das Áreas. No presente trabalho, foram utilizados vários métodos para avaliar qual, melhor atende ao problema proposto.

## Desenvolvimento

Para desenvolver a aplicação de avaliação de projeto de TCC utilizou-se o toolbox do Matlab para aplicações fuzzy e foram usadas variáveis de avaliação que foram repassadas pela coordenadoria de monografia do curso de sistemas de informação.



Durante o trabalho foram feitos vários testes com diferentes funções para as variáveis de entrada e a medida em que foram implantadas foi-se acompanhando o resultado. Nas variáveis de entrada decidiu-se usar então funções gaussianas, por terem um melhor desempenho nos testes realizados. Para as funções de saídas foram testadas funções, gaussianas, triangulares e trapezoidais, foi visto que as funções trapezoidais na maioria dos testes apresentaram melhor resultado, seguidas pelas funções gaussianas e a trapezoidal nas funções de saída não apresentaram bons resultados.

Como função de *defuzzificação* foram testadas todas disponíveis na *toolbox* do Matlab e os resultados são citados abaixo:

- A função *centroide* não funcionam bem quando as notas estão nos picos de alta ou baixa, por sempre dar o resultado baseado no centro da função de saída.
- A função *som* não apresentou bons resultados por basear no primeiro valor dos máximos, ou seja, quando se tem uma nota muito alta na primeira variável, mesmo tendo uma nota bem baixa nas demais variáveis, o resultado é uma nota alta.
- A função *lom* apresenta o inverso do problema da anterior, por fazer a inferência a partir do último valor dos máximos, ela não apresenta bom resultado quando se tem um valor muito alto na última variável, pois mesmo se tendo valores baixos nas demais, o valor da nota é baixo.
- A função *mom* foi a que apresentou um dos melhores resultados, visto que ela se baseia em uma média dos valores máximos gerados na saída do sistema. Este método apesar de funcionar bem, não apresenta bom resultado quando as notas das variáveis giram em torno de valores médios, visto que certas vezes a nota estaciona em valores próximos ao valor médio.

## Conclusões

Usando funções gaussianas e/ou trapezoidais na saída do sistema de inferência, este funcionou bem para casos onde alguma das notas eram bem acima da média ou abaixo. Porém não apresentou bons resultados quando os valores de todas as variáveis eram próximos do valor médio, pois após a defuzzificação o valor final estabilizava em um único valor médio, não apresentando variação como era esperado.

Foi identificado que a função de *defuzzificação mom* foi a que apresentou os melhores resultados, quanto ao objetivo proposto, por apesar de apresentar o problema citado anterior a respeito dos valores médios, foi a função que retornou um resultado aceitável na maioria dos casos.

No conjunto de saída do sistema a função que foi usada para definir nota baixa, média e alta foi a gaussiana, por também apresentar os melhores resultados. A função trapezoidal também apresentou bons resultados em alguns casos.

## Referências

- [1] Jang, J.-S. R. and Sun, C.-T. (1996). Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence, Prentice-Hall.
- [2] MathWorks. MATLAB: Fuzzy Logic Toolbox. < <http://www.mathworks.com/products/fuzzy-logic>>. Acesso em 25 de junho de 2015.