



UNIVERSIDADE FEDERAL DA BAHIA

TRABALHO DE GRADUAÇÃO

Utilização de Prolog para implementação intuitiva de regras de inferência fuzzy

Genicleito Carvalho Beltrão Gonçalves

Programa de Graduação em Ciência da Computação

Salvador
05 de fevereiro de 2018

GENICLEITO CARVALHO BELTRÃO GONÇALVES

**UTILIZAÇÃO DE PROLOG PARA IMPLEMENTAÇÃO INTUITIVA
DE REGRAS DE INFERÊNCIA FUZZY**

Este Trabalho de Graduação foi apresentado ao Programa de Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Profa. Dra. Tatiane Nogueira Rios

Salvador
05 de fevereiro de 2018

Ficha catalográfica.

Gonçalves, Genicleito C. B.

Utilização de Prolog para implementação intuitiva de regras de inferência fuzzy/ Genicleito Carvalho Beltrão Gonçalves– Salvador, 05 de fevereiro de 2018.

41p.: il.

Orientadora: Profa. Dra. Tatiane Nogueira Rios.
Monografia (Graduação)– UNIVERSIDADE FEDERAL DA BAHIA, INSTITUTO DE MATEMÁTICA, 05 de fevereiro de 2018.

“1. Prolog. 2. Fuzzy Prolog 3. Lógica de Predicados. 3. Lógica de Primeira Ordem. 4. Mineração de dados. 3. Lógica Fuzzy. 4. Aprendizado de máquina. 5. Inteligência Artificial.”.

I. Rios, Tatiane Nogueira. II. UNIVERSIDADE FEDERAL DA BAHIA. INSTITUTO DE MATEMÁTICA. III Título.

CDD 000.00000

TERMO DE APROVAÇÃO

GENICLEITO CARVALHO BELTRÃO GONÇALVES

UTILIZAÇÃO DE PROLOG PARA IMPLEMENTAÇÃO INTUITIVA DE REGRAS DE INFERÊNCIA FUZZY

Este Trabalho de Graduação foi julgado adequado à obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Programa de Graduação em Ciência da Computação da Universidade Federal da Bahia.

Salvador, 05 de fevereiro de 2018

Profa. Dra. Tatiane Nogueira Rios
Universidade Federal da Bahia

Prof. Dr. Danilo Barbosa Coimbra
Universidade Federal da Bahia

Prof. Dr. Ricardo Rios
Universidade Federal da Bahia

Dedico esse trabalho primeiramente a Deus e a meus amados pais, incansáveis na tarefa de criar e educar três filhos. Sem este empenho de ambos, certamente, este trabalho não teria sido desenvolvido.

AGRADECIMENTOS

É com bastante felicidade que chego neste momento tão importante na vida de um jovem estudante universitário, o famoso TCC. Este é um momento de celebrar pelos momentos árduos que foram passados e forneceram uma bagagem de experiências espetaculares. Esta é a materialização de mais uma etapa acadêmica, que começou desde o processo de alfabetização.

Primeiramente gostaria de agradecer a Deus pela sua graça imensurável em todos os momentos da minha vida.

Agradecer a minha amada mãe Gilza, que apesar de todas as dificuldades presentes em uma pequena cidadezinha do interior, foi incansável na tarefa de criar a mim e meus irmãos com muito amor, prezando sempre pelos nossos estudos e para que nos tornássemos pessoas de caráter.

A meu amado pai Carlos, lavrador, que derramou muito suor para que pudesse sustentar com dignidade uma família com cinco integrantes, mesmo com o ensino fundamental incompleto sempre priorizou pelos estudos dos filhos, é o homem mais sábio que eu conheço.

A minha orientadora, professora Tatiane, que orientou-me com excelência durante todo o período de desenvolvimento desse trabalho, sempre com muita paciência e cobrando com muita compreensão.

Aos meus professores, desde os responsáveis pela minha alfabetização até os professores do curso que me apresentaram visões esplendidas de novas formas de vê o mundo.

A meus amados irmãos, Genicleide e Carlos Júnior, que compartilharam comigo momentos de apreensão e de muitas felicidades.

A minha grande amiga Dalila, antiga professora do ensino médio, que acabou se tornando uma pessoa muito especial e sem sombra de dúvidas merece muito a minha gratidão. Se hoje tenho a oportunidade de estar finalizando esse curso que tanto amo em uma excelente Universidade é, também, graças a um pontapé dela lá no início.

A Miriam, uma segunda mãe, sempre com conselhos que levarei para a vida inteira, além de preocupar-se sempre com meus estudos e com o andamento do curso, mesmo com a distância.

A Associação dos Estudantes Baixagrandenses (ASSEB) que desde o final do ensino médio já dava apoio para que eu pudesse sair do interior e ter a oportunidade de fazer o curso que eu sempre sonhava e continua dando apoio até que eu possa finalmente finalizá-lo.

A minha madrinha, Sara, que me deu meu primeiro notebook assim que entrei no curso e não tinha condições de adquirir um.

A Pró-Reitoria de Ações Afirmativas e Assistência Estudantil (PROAE) pela assistência estudantil que permitiu que um menino pobre do interior pudesse se manter

em uma cidade desconhecida e sem familiares por perto.

Ao grupo de Gestores da Rede Acadêmica de Computação (GRACO) que me permitiu obter conhecimentos fantásticos sobre redes e sistemas Linux.

Enfim, agradeço a todas as pessoas que de alguma forma, direta ou indiretamente, contribuíram para que eu pudesse estar concluindo mais esta importante e feliz etapa da minha vida.

Não se deve ir atrás de objetivos fáceis, é preciso buscar o que só pode ser alcançado por meio dos maiores esforços.

—ALBERT EINSTEIN

RESUMO

O presente trabalho de conclusão de curso pretende realizar, com base nas pesquisas da área de programação em lógica, mineração de dados e campos de conhecimento relacionados, uma investigação da utilização de lógica Fuzzy em Lógica de Primeira Ordem, por meio do Prolog. Foi demonstrada, por meio de evidências experimentais e análises dos erros médios absolutos dos resultados dos experimentos, que o Fuzzy-Prolog obteve bons resultados quando comparado com a biblioteca JFuzzyLogic, implementada na linguagem Java. Ao longo desta monografia, portanto, são discutidos aspectos relevantes da etapa de pré-processamento das bases de dados utilizadas para classificação; da realização da predição das classes de instâncias de teste e dos critérios de validação da classificação obtida; da comparação dos resultados da abordagem com Fuzzy-Prolog e JFuzzyLogic. Adicionalmente, para contextualizar outras abordagens de programação lógica com Fuzzy-Prolog, são apresentadas e destacadas pesquisas em trabalhos relacionados. A pesquisa realizada para desenvolvimento deste trabalho tem como contribuição uma forma automática de geração de regras fuzzy sem a presença de especialistas mesclada à utilização do Fuzzy-Prolog em classificação. Os resultados da pesquisa estimulam novas formas de abordagens, cuja execução certamente irá transcorrer em trabalhos futuros.

Palavras-chave: mineração de dados, lógica fuzzy, classificação, prolog, lógica de primeira ordem, aprendizado de máquina, inteligência artificial

ABSTRACT

The present work of course completion intends to carry out an investigation of the use of Fuzzy logic in First Order Logic, through Prolog, based on research in the of the programming area in logic, data mining and related fields of knowledge. Throughout this monograph, therefore, important aspects of the preprocessing stage of the databases used for classification are discussed; the performance of the prediction of the classes of test instances and the criteria of validation of the classification obtained; of the results of the Fuzzy-Prolog approach with JFuzzyLogic. In order to contextualize other logical programming approaches with Fuzzy-Prolog, research is presented and highlighted in related works. The research carried out for the development of this work has as contribution an automatic form of fuzzy rules generation without the presence of specialists, already well known, but that can be mixed with the use of Fuzzy-Prolog in classification, for example. We show here, through experimental evidence and analyzes of the absolute mean errors of the results of the experiments, by comparing Fuzzy-Prolog and JFuzzyLogic, that Fuzzy-Prolog obtained good results. The results of the research stimulate new forms of approaches, whose execution will certainly occur in future works.

Keywords: data mining, fuzzy logic, classificaton, prolog, first-order logic, machine learning, artificial intelligence

SUMÁRIO

Capítulo 1—Introdução	1
Capítulo 2—Fundamentação Teórica	4
2.1 Considerações iniciais	4
2.2 Conjuntos Fuzzy	5
2.2.1 Funções de Pertinência	5
2.3 Sistemas Fuzzy baseado em regras	6
2.3.1 Métodos de Defuzzificação	8
2.4 Prolog	8
2.5 Considerações finais	9
Capítulo 3—Trabalhos Relacionados	10
3.1 Considerações Iniciais	10
3.2 Utilização do Prolog como mecanismo de inferência Fuzzy	10
3.3 Considerações finais	13
Capítulo 4—Abordagem proposta	15
4.1 Considerações iniciais	15
4.2 Geração de Regras de Wang&Mendel	15
4.2.1 Etapa 1	16
4.2.2 Etapa 2	16
4.2.3 Etapa 3	16
4.2.4 Etapa 4	17
4.2.5 Etapa 5	17
4.3 Abordagem Proposta	18
4.4 Experimentos	20
4.4.1 Resultados e discussão da base de dados Iris	21
4.4.2 Resultados e discussão da base de dados Wine Quality	23
4.4.3 Resultados e discussão da base de dados Contraceptive Method Choice	26
4.4.4 Resultados e discussão da base de dados Abalone	29
4.4.5 Resultados e discussão da base de dados Wireless Indoor Localization	32
4.4.6 Discussão dos Resultados	35
4.5 Considerações finais	36

SUMÁRIO	xi
Capítulo 5—Conclusão	38
5.1 Trabalhos futuros	39

LISTA DE FIGURAS

2.1	Funções de Pertinência mais comuns: (a) Triangular, (b) Trapezoidal e (c) Gaussiana.	6
2.2	Base de Conhecimento de um Sistema Fuzzy Baseado em Regras	7
4.1	Função de pertinência para o atributo [1] sepal length da base Iris. Com 3 regiões ($N = 1$)	17
4.2	Desenho do processo de experimentação que será abordado mais adiante nesta monografia	19
4.3	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Iris. Regiões de Saída fixas em 3.	22
4.4	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Iris. Regiões de Saída fixas em 5.	23
4.5	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-Red. Regiões de Saída fixas em 3.	26
4.6	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-Red. Regiões de Saída fixas em 11.	27
4.7	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-White. Regiões de Saída fixas em 3.	28
4.8	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-White. Regiões de Saída fixas em 11.	28
4.9	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-White. Regiões de Saída fixas em 3.	30
4.10	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-White. Regiões de Saída fixas em 11.	30
4.11	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Abalone. Regiões de Saída fixas em 3.	32
4.12	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Abalone. Regiões de Saída fixas em 29.	33
4.13	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wireless. Regiões de Saída fixas em 3.	34
4.14	Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wireless. Regiões de Saída fixas em 5.	35

LISTA DE TABELAS

4.1	Resumo das características das bases de dados	21
4.2	Características dos atributos do conjunto de treinamento da base de dados <i>Iris</i>	21
4.3	Erros obtidos na classificação da base de dados <i>Iris</i> . RA (Regiões de Atributos), RS (Regiões da Saída) e RG (Regras Geradas)	22
4.4	Características dos atributos do conjunto de treinamento da base de dados <i>Wine Quality - Red</i>	24
4.5	Características dos atributos do conjunto de treinamento da base de dados <i>Wine Quality - White</i>	24
4.6	Erros obtidos na classificação da base de dados <i>Wine-Red</i> . RA (Regiões de Atributos), RS (Regiões da Saída) e RG (Regras Geradas).	25
4.7	Erros obtidos na classificação da base de dados <i>Wine-White</i> . RA (Regiões de Atributos), RS (Regiões da Saída) e RG (Regras Geradas).	25
4.8	Características dos atributos do conjunto de treinamento da base de dados <i>Contraceptive Method Choice</i>	27
4.9	Erros obtidos na classificação da base de dados <i>Contraceptive</i> . RA (Regiões de Atributos), RS (Regiões da Saída) e RG (Regras Geradas).	29
4.10	Características dos atributos do conjunto de treinamento da base de dados <i>Abalone</i>	31
4.11	Resultados obtidos na classificação da base de dados <i>Abalone</i> . RA (Regiões de Atributos), RS (Regiões da Saída) e RG (Regras Geradas).	31
4.12	Características dos atributos do conjunto de treinamento da base de dados <i>Wireless Indoor Localization</i>	33
4.13	Resultados obtidos na classificação da base de dados <i>Wireless</i> . RA (Regiões de Atributos), RS (Regiões da Saída) e RG (Regras Geradas).	34
4.14	Resumo dos resultados obtidos em cada base de dados para as duas abordagens apresentadas nesta monografia.	36

Uma breve introdução sobre do que se trata esta monografia e a maneira como o texto está organizado.

INTRODUÇÃO

Uma grande quantidade de dados é gerada diariamente, quer seja pela internet ou por várias empresas ao redor do mundo¹. Para retirar informação útil desses dados é necessário processar essas informações em bases de dados muitas vezes mal estruturadas e de difícil compreensão.

Neste contexto, a área de Aprendizado de Máquina tem crescido em investimentos, estudos e ferramentas cada vez mais confiáveis desenvolvidas com o intuito de automatizar todo esse processo de análise, aquisição de informações úteis e classificação sobre essas bases de dados.¹

Considerando que muitos dados e informações que são coletadas retratam tomadas de decisões humanas e em alguns casos modelam o mundo real à forma humana de pensar, faz-se necessária uma abordagem que considere a imprecisão e incerteza inerente aos dados. Em [Klir e Yuan \(1995\)](#) há a seguinte afirmação sobre incerteza:

De acordo com a visão alternativa (ou moderna), a incerteza é considerada essencial para a ciência; não é apenas uma praga inevitável, mas, de fato, é uma ótima utilidade.

Considerar a incerteza uma ótima utilidade, como afirmado por [Klir e Yuan \(1995\)](#), favoreceu o uso da lógica Fuzzy, a qual é baseada na teoria de conjuntos fuzzy cujas sentenças que representam afirmações do mundo real variam em grau de verdade entre 0 e 1, podendo ser parcialmente verdadeira ou parcialmente falsa, garantindo graus de imprecisão e incerteza.

Adicionalmente, sistemas que consideram a imprecisão humana requerem o uso de uma linguagem de programação em que seja possível expressar convenientemente regras para interpretar informações e extrair elementos-chaves para o processamento de conhecimento incerto útil, bem como uma tecnologia que possa executar essas regras de forma muito eficiente.

¹<http://www.bigdatabusiness.com.br/os-grandes-e-impresioantes-numeros-de-big-data/>

Sendo assim, o Prolog, uma linguagem do paradigma de Programação em Lógica Matemática, apresenta-se como uma linguagem ideal devido à sua simplicidade, capacidade expressiva e proximidade com linguagem natural. É uma linguagem para implementar, de maneira intuitiva, qualquer lógica ou raciocínio baseado em regras, inclusive para representar tomadas de decisões incertas, como na lógica Fuzzy.

Por outro lado, a linguagem Java, criada na década de 90, é uma linguagem orientada a objetos, estruturada, imperativa, fortemente tipada e baseada em classes. Normalmente, a linguagem é compilada para o conjunto de instruções *bytecode* e o formato binário definido na Especificação de Máquina Virtual Java². Além disso, Java é considerada segura e confiável³.

Para a realização de inferência fuzzy e, conseqüentemente, extração de conhecimento incerto, tais linguagens podem ser utilizadas por intermédio de ferramentas desenvolvidas especificamente para tal atividade. Sendo assim, utilizando a linguagem Java tem-se, entre outras, a ferramenta JFuzzyLogic; enquanto utilizando a linguagem Prolog tem-se a abordagem Fuzzy-Prolog; ambas utilizadas como mecanismos para investigação neste trabalho.

Diante deste contexto, tendo em vista que muitos dados apresentados para analistas de dados representam informações que ele pode não ter conhecimento especialista suficiente para realizar uma análise detalhada, faz-se necessário uma abordagem capaz de analisar estes dados com certa confiabilidade.

De forma a servir de guia para a análise apresentada neste trabalho, foi formulada a seguinte hipótese:

A abordagem Fuzzy-Prolog permite obter resultados expressivos provenientes da linguagem natural do Prolog, apresentando, em geral, um menor erro médio absoluto na tarefa de classificação usando regras fuzzy, em comparação com a ferramenta JFuzzyLogic.

Para demonstrar a validade desta hipótese formulada, com base na exploração das abordagens existentes na literatura sobre inferência fuzzy utilizando Prolog, definiu-se o seguinte objetivo:

Conduzir uma investigação em torno de inferência Fuzzy com Prolog e com Java para compreender o comportamento das linguagens para execução de conhecimento incerto.

A fim de atender a esse objetivo, foram realizadas as seguintes tarefas ao longo do desenvolvimento desta monografia: estudo dos fundamentos teóricos necessários para a implementação flexível de conhecimento, revisão das abordagens existentes que se dizem respeito à incorporação de lógica fuzzy no mecanismo de inferência do Prolog, condução de diversos experimentos para analisar o comportamento desta abordagem com Fuzzy-Prolog e compará-lo à uma abordagem Fuzzy.

²<https://docs.oracle.com/javase/7/docs/technotes/guides/language/index.html>

³https://www.java.com/pt_BR/download/faq/whatis_java.xml

Detalhes de cada tarefa realizada, bem como a comprovação da hipótese levantada, são apresentados ao longo deste trabalho como segue:

Capítulo 2: Neste capítulo, os fundamentos teóricos necessários para compreender o processo de inferência em Fuzzy e no Prolog.

Capítulo 3: Este capítulo aborda uma breve revisão das principais literaturas encontradas sobre o tema com diversas abordagens apresentadas pelos pesquisadores.

Capítulo 4: Neste capítulo, um estudo dos impactos da utilização do Fuzzy-Prolog é apresentado por meio da realização de análises experimentais em comparação com análises experimentais do JFuzzyLogic, que traz uma abordagem Fuzzy.

Capítulo 5: Por fim, este capítulo contempla as conclusões obtidas de todo o estudo realizado nesta monografia, assim como discussões a respeito dos resultados obtidos nos experimentos. Aqui também estão apontadas algumas possibilidades de extensões que podem derivar desta pesquisa.

Este capítulo tem como objetivo fundamentar as bases necessárias dos campos de estudos utilizados nesta monografia.

FUNDAMENTAÇÃO TEÓRICA

2.1 CONSIDERAÇÕES INICIAIS

A lógica fuzzy foi proposta por [Zadeh \(1965\)](#), onde o autor define conjuntos fuzzy, os quais constituem classes de objetos com valores contínuos de pertinência. Cada conjunto é caracterizado por uma função de pertinência, a qual atribui a cada elemento do conjunto um grau de pertinência que varia entre zero e um. As operações matemáticas da teoria dos conjuntos, como inclusão, união, intersecção, complemento, relação, etc., também são estendidas aos conjuntos fuzzy, assim como várias propriedades dessas notações são definidas.

Uma forte motivação para o uso da lógica fuzzy, veio da maneira como os seres humanos estão constantemente classificando e rotulando o mundo real. Por exemplo, ao amanhecer já classificam se o dia está quente ou frio. Assim como ao fazer uma corrida matinal podem considerar a distância percorrida como longa ou curta. Este conjunto de dias quentes ou frios e distâncias longas ou curtas não são abrangidas na sua totalidade na lógica clássica, afinal, estas formas imprecisas de descrever o mundo tem papel importante no processo cognitivo humano.

Diante dessa estreita relação entre lógica fuzzy e a forma humana de pensar surge então uma nova motivação para o uso do Prolog. O Prolog foi introduzido por [Colmerauer e Roussel \(1996\)](#), já definindo alguns conceitos, ainda preliminares na época, sobre predicados e fatos. A Programação em Lógica do Prolog torna-se bem atraente com as características da lógica fuzzy.

Neste sentido, esta seção tem como propósito contextualizar os principais aspectos da lógica fuzzy e do Prolog que os tornam importantes no contexto de aprendizagem de máquina. Ressalta-se, portanto, que definições mais aprofundadas sobre a lógica fuzzy e sobre o Prolog fogem do escopo desse texto.

2.2 CONJUNTOS FUZZY

Um conjunto S é definido, na lógica clássica, por meio da função característica

$$S : X \rightarrow \{0, 1\}$$

em que elementos do conjunto universo X pertencem totalmente ao conjunto S quando $S(x) = 1$ e não pertencem quando $S(x) = 0$.

A generalização da função característica para função de pertinência associará elementos de um dado conjunto universo X a números reais no intervalo $[0,1]$, definindo um conjunto fuzzy.

Logo, um conjunto fuzzy S no conjunto universo X é caracterizado pela função

$$S : X \rightarrow [0, 1]$$

e $S(x)$ definirá o grau que o elemento x pertence ao conjunto fuzzy S , segundo [Klir e Yuan \(1995\)](#). Uma das mais importantes aplicações está no tratamento de imprecisão e incerteza. O que nos permite modelar soluções mais adequadas para ambientes imprecisos e incertos.

2.2.1 Funções de Pertinência

A forma de um conjunto fuzzy depende da função de pertinência que o define, ([Nogueira, 2008](#)). Algumas tornaram-se mais populares por serem mais conhecidas e são frequentemente utilizadas pela sua simplicidade. Nesta subseção serão apresentados os três tipos de funções de pertinência mais conhecidos, cuja representação gráfica pode ser observada na Figura 2.1.

1. **Função Triangular:** é definida pelos parâmetros a , m e b , tal que $a \leq m \leq b$.

$$A(x) = \begin{cases} 0 & \text{se } x \leq a \\ \frac{x - a}{m - a} & \text{se } x \in (a, m) \\ 1 & \text{se } x = m \\ \frac{b - x}{b - m} & \text{se } x \in (m, b) \\ 0 & \text{se } x \geq b \end{cases}$$

2. **Função Trapezoidal:** definida pelos parâmetros a , m , n , b , tal que $a \leq m \leq n \leq b$.

$$A(x) = \begin{cases} 0 & \text{se } x \leq a \\ \frac{x-a}{m-a} & \text{se } x \in (a, m) \\ 1 & \text{se } x \in [m, m] \\ \frac{b-x}{b-n} & \text{se } x \in (n, b) \\ 0 & \text{se } x \geq b \end{cases}$$

3. **Função Gaussiana:** definida pelos parâmetros m e k , tal que $k > 0$

$$A(x) = \left\{ e^{-k(x-m)^2} \right.$$

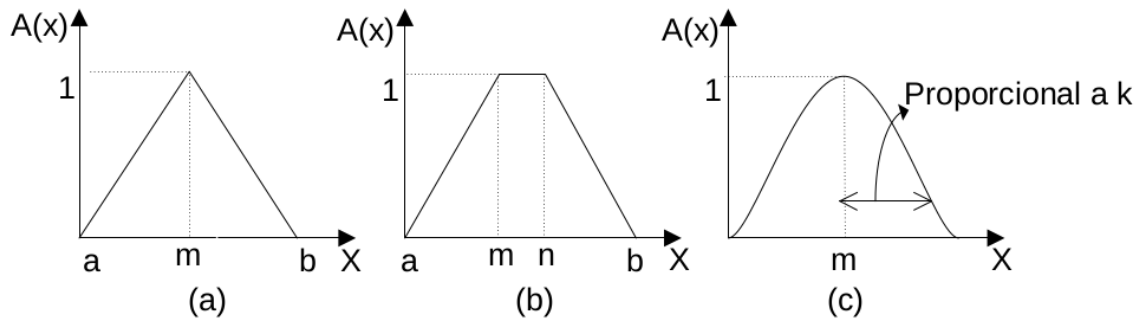


Figura 2.1: Funções de Pertinência mais comuns: (a) Triangular, (b) Trapezoidal e (c) Gaussiana.

2.3 SISTEMAS FUZZY BASEADO EM REGRAS

As regras são utilizadas como um mecanismo formal para representar o conhecimento de um domínio. Segundo [Nogueira \(2008\)](#), as regras fuzzy permitem a representação de conhecimento impreciso, o que muitas vezes torna a modelagem do problema mais fiel e adequada ao mundo real.

Para tanto, o conceito de variável linguística faz-se necessário no processo de criação de regras fuzzy, as quais são variáveis cujos valores são palavras ou sentenças em linguagem natural ao invés de números. Com este conceito formulado, problemas naturalmente imprecisos e complexos passam a ser manipuláveis por computadores. A interpretação de um determinado conhecimento expresso na forma linguística torna-se passível de uma representação matemática graças a essas variáveis.

Por meio destas variáveis linguísticas, as regras fuzzy possibilitam a expressão clara de diretivas e estratégias, captam o conhecimento experimental de especialistas humanos e possuem um formato linguístico de fácil compreensão.

Regras fuzzy são semelhantes às regras convencionais, por exemplo:

SE *antecedente* ENTÃO *consequente*

De acordo com [Nogueira \(2008\)](#), esse formato tem a finalidade de estabelecer relações entre as variáveis que aparecem no antecedente, também chamado de condição ou premissa, e as que aparecem no consequente, também chamado de conclusão ou ação. Por exemplo, a regra:

SE o *tempo* é *nublado* ENTÃO a *chance_de_jogar_tenis* é *muito_grande*

estabelece uma relação entre a variável *tempo* e *chance_de_jogar_tenis* usando os termos linguísticos (ou regiões) *nublado* e *muito_grande*, que são imprecisos.

Os Sistemas Fuzzy Baseados em Regras (SFBR) são compostos, principalmente, por dois componentes: a Base de Conhecimento e o Mecanismo de Inferência.

A Base de Conhecimento é composta pela Base de Dados, a qual contém as definições dos conjuntos fuzzy relacionados aos termos linguísticos utilizados nas regras fuzzy e pela Base de Regras, que armazenam um conjunto de regras que modelam um determinado problema. A Figura 2.2, apresentada por [Nogueira \(2008\)](#), ilustra a estrutura de uma Base de Conhecimento fuzzy.

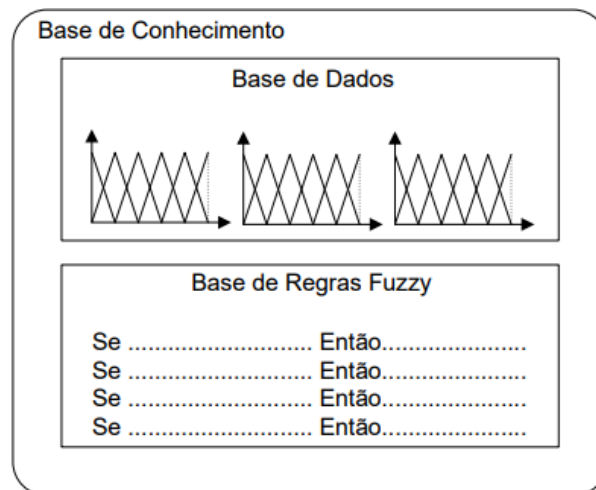


Figura 2.2: Base de Conhecimento de um Sistema Fuzzy Baseado em Regras

O Motor de Inferência fuzzy é responsável pelo processamento das regras, o qual é realizado por algum método de raciocínio, [Nogueira \(2008\)](#). A partir de regras e fatos conhecidos então, é possível aplicar um procedimento de inferência para derivar conclusões.

[Nogueira \(2008\)](#) apresenta o modo de operação de um Sistema Fuzzy Baseado em Regras com as seguintes etapas:

1. Fuzzificação: transformação de valores de entrada em conjuntos fuzzy. Para entradas numéricas, neste passo será realizado o cálculo do grau de pertinência de cada valor de entrada do conjunto fuzzy correspondente.

2. Cálculo do Grau de Disparo da regra: agregação dos antecedentes da regra para o cálculo de conjunção fuzzy.
3. Aplicação da inferência com a derivação dos resultados individuais para cada regra.
4. Combinação de possíveis saídas fuzzy.
5. Defuzzificação: transformação do resultado fuzzy em um resultado preciso. Este processo pode não ocorrer, dependendo da aplicação do Sistema Fuzzy.

2.3.1 Métodos de Defuzzificação

Vários métodos de defuzzificação são utilizados para converter resultados linguísticos da inferência em um valor real que melhor o represente. Entre eles estão o *Centro de Área*, *Centro de Máximos*, *Média de Máximos*, *Centro de Máximos Ponderado*, *Valor mais plausível* e outros. Uma vez que o método utilizado nesta monografia é o Centro de Gravidade, este será o único método, destes descritos, que será detalhado nesta subseção.

O método do Centróide é um dos mais utilizados (para Centro de Gravidade ou Centro de Massa). A saída precisa y é dada pelo centro de gravidade do conjunto de consequente obtido pela composição das regras. Neste caso, K indica o número de regras fuzzy na base de regras, \bar{y}^i indica o valor central (centróide) da região O^i e $m_{O^i}^i$ o grau de pertinência da variável de saída. Por meio da média ponderada das regiões encontra-se o ponto de equilíbrio da região fuzzy, conforme visualizado na Equação 2.1.

$$y = \frac{\sum_{i=1}^K m_{O^i}^i \bar{y}^i}{\sum_{i=1}^K m_{O^i}^i} \quad (2.1)$$

A Equação 2.1 representa o método de defuzzificação do Centróide.

Desta forma, todas as instâncias testadas nesta monografia tiveram seus valores checados e verificados, aplicando a Fórmula 2.1 a todas as bases de dados utilizadas.

2.4 PROLOG

Segundo [Bratko \(1990\)](#): o Prolog é uma linguagem de programação centrada em torno de um pequeno conjunto de mecanismos básicos, incluindo correspondência de padrões, estruturação de dados baseados em árvores e *backtracking* automático. Este pequeno conjunto constitui uma estrutura de programação robusta e flexível. Prolog representa uma linguagem poderosa para Inteligência Artificial e programação simbólica e não-numérica em geral.

Enquanto as linguagens convencionais são orientadas para o procedimento, o Prolog apresenta a visão descritiva ou declarativa. Isso altera muito o modo de pensar sobre problemas e torna a aprendizagem programada no Prolog um desafio intelectual ([Bratko, 1990](#)).

A estrutura de um projeto em Prolog é formada, basicamente, por fatos e cláusulas. Prolog é especialmente adequado para resolver problemas que envolvem objetos e relações entre objetos. O exemplo abaixo demonstra uma relação de paternidade e mostra como um fato é escrito em Prolog:

```
pai(joao, felipe).
```

que pode ser lido naturalmente como “João é pai de Felipe”.

2.5 CONSIDERAÇÕES FINAIS

Neste capítulo foi apresentado e fundamentado a teoria necessária para se compreender os temas estudados nesta monografia. De maneira geral, foi apresentado que a proposição da lógica fuzzy, proporcionou a capacidade de tratamento de imprecisão e incerteza inerentes ao mundo real e ao conhecimento humano. Esta imprecisão e incerteza são utilizadas com muita frequência pelas pessoas para classificar objetos.

Foi visto que o Prolog, nativamente criado para Programação em Lógica, é uma ferramenta renomada nas pesquisas de Inteligência Artificial. Diferentemente das linguagens convencionais, Prolog é especialmente adequado para resolver problemas que envolvem relações entre objetivos.

Foram vistas algumas outras breves definições sobre a etapa de Pré-Processamento e Classificação em Aprendizado de Máquina. A Classificação será uma etapa importante no processo de experimentação desta monografia, com ela que serão obtidos os resultados que serão analisados no Capítulo 4.

Considerando o objetivo definido nessa monografia, será proposto então uma junção da Lógica Fuzzy com a Lógica de Primeira Ordem do Prolog com abordagens que serão vistas a partir do próximo capítulo.

No próximo capítulo, será apresentada uma breve revisão das pesquisas desenvolvidas sobre o tema abordado nesta monografia.

Trabalhos relacionados a utilizando logica fuzzy com Prolog.

TRABALHOS RELACIONADOS

3.1 CONSIDERAÇÕES INICIAIS

A proposta de utilização da linguagem natural Prolog para inferência em lógica fuzzy abrange vários campos de estudo. Isso a torna bastante densa e robusta. Para que a atividade de pesquisa e seleção do conhecimento científico fosse bem fundamentada, as produções científicas mais referenciadas são citadas a seguir.

Foram realizadas buscas por artigos que tratassem do tema nos repositórios IEEEExplore¹ e ACM Digital Library².

Tais artigos, apesar de serem mais antigos, são bastante citados em produções mais atuais, tendo alta relevância para esta produção científica e foram adicionados para completar os resultados obtidos da busca.

3.2 UTILIZAÇÃO DO PROLOG COMO MECANISMO DE INFERÊNCIA FUZZY

A lógica fuzzy foi proposta por Zadeh (1965) para fornecer mecanismos que pudessem lidar com a imprecisão e incerteza presentes em diversos problemas do mundo real. A partir desse trabalho desenvolvido por Zadeh (1965), várias pesquisas futuras se propuseram a explorar intensivamente os benefícios concedidos pela flexibilidade proporcionada pela lógica fuzzy. Dentre esses trabalhos, muitos pesquisadores preocuparam-se em apresentar uma alternativa ao paradigma tradicional, baseado apenas em *true* ou *false*, através do uso de conjuntos difusos e variáveis linguísticas para modelagem de expressões de linguagem natural que pudessem fazer a ponte entre o mundo simbólico impreciso da linguagem natural e o paradigma tradicional do computador, como em Kaminski (1992).

Neste sentido, sabe-se que o Prolog tem sido amplamente utilizado na pesquisa de inteligência artificial. Porém, uma das suas deficiências é a falta de um mecanismo natural para lidar com a incerteza, segundo relata Martin et al. (1987). Uma possível

¹<http://ieeexplore.ieee.org/>

²<https://dl.acm.org/>

solução para este problema, delineada em várias pesquisas científicas e nesta monografia, é basear Prolog em lógica difusa ao invés de lógica convencional de dois valores.

Segundo Bratko (1990) pode-se modelar a incerteza em Prolog ao atribuir às asserções alguma qualificação que não seja *true* ou *false*, algo que não é natural da linguagem. Essa qualificação pode ser expressa por termos linguísticos - por exemplo, verdade, altamente provável, provável, improvável, impossível. Alternativamente, o grau de crença pode ser expresso como um número real em algum período - por exemplo, entre 0 e 1. Esses números apresentam várias definições, como “fator de certeza”, “medida de crença”, “certeza subjetiva” ou “grau de pertinência”. Seguindo essa linha de representação da incerteza por meio de expressões de linguagem natural, Kaminski (1992) ressalta que abordagens difusas foram utilizadas em muitas áreas de aplicação, incluindo sistemas de controle, reconhecimento de padrões, sistemas especializados, análise de risco e tomada de decisão médica. E, quando usado com variáveis linguísticas para modelar expressões de linguagem natural, o paradigma fuzzy fornece um bom formato para explorar alguns dos problemas da conexão humano/computador e representa com precisão a semântica da linguagem cotidiana, modelando o esquema de representação do conhecimento humano, operacionalizando e quantificando conceitos imprecisos e não discretos.

Sendo assim, o Prolog é uma ferramenta muito conveniente para projetar regras próximas ao raciocínio humano, uma vez que é mais direcionada ao conhecimento do que em algoritmos, apresenta uma abordagem lógica e simbólica, consulta à base de dados, baseado em regras e modelagem do raciocínio humano. Por conseguinte, assim como o conhecimento humano lida com incertezas e imprecisões, o Prolog também precisa lidar com essas incertezas, confusões ou incompletude da informação.

Neste contexto, Martin et al. (1987) introduziu o sistema FProlog, um interpretador Fuzzy Prolog escrito em Franz Lisp. Ressalta fortemente a presença, no Prolog, da estratégia de backtracking de profundidade e primeira profundidade, usada para explorar a árvore de pesquisa. O processo de correspondência (unificação) garante, por substituições apropriadas, que o cabeçalho da cláusula e o sub-objetivo são idênticos. Algo que garante vantagens à modelagem do mecanismo de inferência da lógica fuzzy em Prolog.

Martin et al. (1987) considera Prolog uma linguagem versátil e conveniente para pesquisas de Inteligência Artificial, uma vez que em muitas de suas aplicações, informações difusas estão envolvidas e é necessário refletir isso no Prolog. Por exemplo, dado um conjunto de fatos sobre temperaturas médias, pode-se definir um predicado que irá determinar quão quente e ensolarada será uma área, por exemplo:

$$\begin{aligned} \text{area_quente_e_ensolarada}(X, CHI) :- & \text{temp_media}(X, T), \text{quente}(T, CHI1), \\ & \text{incidencia_raios_solares}(X, S), \text{ensolarado}(S, CHI2), \\ & \text{combinar}(CHI1, CHI2, CHI). \end{aligned} \quad (3.1)$$

Onde, *CHI* é o grau de pertinência da regra, obtido por alguma combinação (*min* ou *max*) dos graus de pertinência obtidos nos predicados anteriores (*quente*(*T*, *CHI1*) e *ensolarado*(*S*, *CHI2*), que determinam quão quente é uma determinada temperatura *T* e quão ensolarada é uma determinada área *S*, respectivamente). Martin et al. (1987)

considera que embora seja viável lidar com a incerteza dessa maneira, os programas Prolog resultantes podem ser desordenados e difíceis de interpretar, se não estruturados de uma forma coerente. Uma solução alternativa, proposta por ele, é levar a incerteza no interpretador Prolog, por meio do FProlog ao invés de tentar adicionar incerteza a um sistema existente. Esta é uma abordagem muito mais natural, produzindo um código mais legível, juntamente com um sistema mais geral que se reduz ao padrão Prolog quando não há incerteza.

Mais recentemente, Rubio-Manzano e Julian-Iranzo (2014) propuseram um modelo de raciocínio mais robusto, baseado em relações semânticas (palavras sinônimas, por exemplo) entre palavras para incorporá-la no mecanismo de inferência de uma linguagem de programação lógica. Este modelo está integrado em uma estrutura de programação de lógica fuzzy e é implementado no sistema Bousi~Prolog³ usando o WordNet⁴. Todo esse processo é transparente para o programador e o raciocínio com palavras é automático. A semântica lexical entre símbolos proporciona a capacidade de raciocinar com palavras e transforma a representação do conhecimento em um processo mais natural e menos complexo. Bousi~Prolog (BPL) é uma extensão da linguagem Prolog padrão.

Levando em conta que o raciocínio lógico é dedutivo e o raciocínio humano é principalmente indutivo, abduutivo e empírico, Rubio-Manzano e Julian-Iranzo (2014) consideram o uso de relações semânticas em uma estrutura de programação lógica com o objetivo de melhorar seus mecanismos de representação e inferência. Enquanto a lógica é uma representação sintética criando e manipulando símbolos em um mundo fechado, a lógica enriquecida com as relações semânticas da linguagem natural é uma representação empírica aberta, combinando conceitos que já foram definitivos e relacionados à linguagem humana. Esta abordagem tem o intuito de projetar um modelo de raciocínio baseado em relações semânticas entre palavras e incorporá-la no mecanismo de inferência de uma linguagem de programação de lógica fuzzy. No entanto, utilizando um dicionário de sinônimos geral para obter relações semânticas, não é possível considerar o contexto, pois no Bousi~Prolog não são usadas informações de contexto para gerar as equações de proximidade. Por exemplo: segundo Rubio-Manzano e Julian-Iranzo (2014), ao consultar no WordNet quais palavras estão mais próximas de “banco” obtém-se “jurar”, “confiança” e “depósito”. Claramente o contexto não foi levado em consideração.

Gisolfi (1990) ressalta que o Prolog não possui um mecanismo para lidar com informações fuzzy. Para lidar com esse problema, podem ser utilizados predicados que gerenciam a incerteza. Alternativamente, pode-se imergir a incerteza no próprio intérprete do Prolog, de modo que possa ser tratada automaticamente, alcançando códigos mais curtos e programas compreensíveis. Este intérprete pode ser adequadamente modificado para não só resolver um problema, mas também fornecer uma medida de incerteza da solução. O conhecimento humano inclui declarações parcialmente verdadeiras e pode ser comunicada através de uma linguagem contendo símbolos imprecisos, permitindo lidar com a incerteza dentro da programação lógica tradicional.

Munoz-Hernandez e Wiguna (2007) propõem a utilização da abordagem Fuzzy-Prolog

³<http://dectau.uclm.es/bousi/>

⁴<https://wordnet.princeton.edu/>

como camada cognitiva no RoboCupSoccer, utilizando uma combinação de Prolog (que é nítido) e Fuzzy-Prolog para implementar esta camada cognitiva, que tem a vantagem de estar incorporando os detalhes técnicos e a lógica fuzzy de forma encapsulada pela representação natural do Prolog.

Guadarrama et al. (2004) apresenta uma definição de uma linguagem de Prolog Fuzzy que modela a lógica Fuzzy. Esta extensão para Prolog é implementada interpretando o raciocínio difuso como um conjunto de restrições que se propagam através dos predicados por meio de operadores de agregação. O intérprete apresentado por Guadarrama et al. (2004) para essa linguagem de Prolog Fuzzy utiliza a *Constraint Logic Programming (Real)*($CLP(\mathbb{R})$).

Segundo Yasui et al. (1995), sistemas “Fuzzy Prolog” substituem o mecanismo de inferência do Prolog por uma variante fuzzy capaz de lidar com a verdade parcial. A maioria desses sistemas implementa a resolução fuzzy introduzida por Lee (1972), sendo exemplos o sistema Prolog-Elf (Ishizuka e Kanai (1985)), o sistema Fril Prolog (Baldwin et al. (1995)) e o F-Prolog (Li e Liu (1990)). Alguns desses sistemas só consideram os discursos dos predicados, enquanto outros sistemas consideram fatos fuzzy ou regras difusas, além de muitos deles utilizarem T-normas e T-conormas, *min* e *max*, (para modelar as operações de conjunção e disjunção). Dessa forma, o autor então adicionou lógica fuzzy a um compilador Prolog usando a programação de lógica de restrição sobre números reais ($CLP(\mathbb{R})$) em vez de implementar uma nova resolução fuzzy como outros sistemas Fuzzy Prolog. O objetivo da pesquisa desenvolvida em Yasui et al. (1995) é mostrar como a introdução de um raciocínio difuso em um sistema Prolog pode produzir uma ferramenta poderosa para resolver problemas fuzzy complexos e incertos e apresentar uma implementação do Prolog Fuzzy como uma aplicação natural do $CLP(\mathbb{R})$.

Yasui et al. (1995) implementou este intérprete como uma extensão sintática de um sistema $CLP(\mathbb{R})$, que foi incorporado como biblioteca no sistema Ciao Prolog (sistema de programação lógica que, entre outros recursos, foi projetado com a compilação incremental modular em mente. O seu sistema de módulos suporta módulos clássicos e módulos fuzzy no mesmo programa e ele incorpora a lógica de restrições). Yasui et al. (1995) escreveu uma biblioteca/pacote, chamada fuzzy, que implementa o interpretador da linguagem Fuzzy Prolog descrita em sua pesquisa.

3.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas pesquisas relacionadas ao tema desta monografia, em que foi possível observar o Prolog como uma ferramenta muito conveniente para representar também incertezas usando regras próximas ao raciocínio humano. Por conseguinte, assim como o raciocínio humano lida com incertezas, o Prolog precisa lidar com essas incertezas ou incompletude da informação e os artigos apresentados neste capítulo destacaram que faz-se muito útil o uso de ferramentas que apresentam a abordagem Fuzzy-Prolog, com o intuito de realizar inferência fuzzy em Lógica de Primeira Ordem.

O objetivo de todas as pesquisas apresentadas neste capítulo é apresentar formas para que o Prolog não só resolva um problema, mas também fornece uma medida de incerteza da solução deste problema.

Desta forma, os estudos presentes nesta monografia são a fundamentação para que a abordagem com Fuzzy-Prolog pudesse ter uma base de regras geradas de forma automática por meio do Wang&Mendel e pudesse ser comparada com uma ferramenta de inferência fuzzy no estado da arte, com o intuito de validar o diferencial dessa abordagem. A ferramenta JFuzzyLogic encapsula todas as suas operações sobre os conjuntos fuzzy por meio do uso de várias classes Java e limita recursos, como o número de métodos de defuzzificação, por exemplo. A abordagem Fuzzy-Prolog, proposta nesta monografia, ressalta a utilização de regras, na forma da apresentada no Exemplo 3.1, de forma transparente e expressiva ao pesquisador.

ABORDAGEM PROPOSTA

4.1 CONSIDERAÇÕES INICIAS

Este capítulo tem o objetivo de detalhar as contribuições desta monografia para o uso do Prolog, com suas características próximas à linguagem humana, juntamente com o mecanismo de inferência fuzzy, para a realização de classificação por meio do uso de regras geradas de acordo com as características de cada base de dados sem uma real necessidade da presença de um especialista, conforme proposto por Wang e Mendel (1992) e doravante chamado Wang&Mendel.

4.2 GERAÇÃO DE REGRAS DE WANG&MENDEL

É necessário aplicar uma técnica apropriada de geração de regras fuzzy diante da ausência do conhecimento de especialista sobre uma determinada base de dados. Considerando que os conjuntos de dados são compostos por atributos de entradas e um único ou vários atributos de saída, as bases de dados exploradas nesse trabalho seguem o seguinte padrão:

$$(x_1^{(1)}, x_2^{(1)}, \dots, x_{n-1}^{(1)}, y_n^{(1)}), \dots, (x_1^{(i)}, x_2^{(i)}, \dots, x_{n-1}^{(i)}, y_n^{(i)}) \quad (4.1)$$

onde, n representa o número de atributos da base de dados; $x_1^{(1)}, \dots, x_{n-1}^{(1)}$ representam os valores das variáveis de entrada ou antecedentes da regra da primeira instância; $y_n^{(1)}$ representa o valor da variável de saída ou consequente da regra da primeira instância. $x_1^{(i)}, \dots, x_{n-1}^{(i)}$ representam os valores das variáveis de entrada da i -ésima instância da base de dados; $y_n^{(i)}$ representa o valor da variável de saída da i -ésima instância.

Wang e Mendel (1992) apresentou um método muito eficaz para gerar regras difusas a partir de dados numéricos, sem necessidade de conhecimento do especialista sobre a base de dados. Para isso, são apresentadas cinco etapas que foram seguidas para a geração de regras nesta monografia:

1. Dividir os espaços de entrada e saída dos dados numéricos em regiões (conjuntos) fuzzy.
2. Gerar regras fuzzy a partir desses dados fornecidos.
3. Atribuir um grau para cada regra gerada, com o intuito de resolver conflitos entre as regras
4. Criar uma base combinada de regras fuzzy e com base nas regras geradas e nas regras linguísticas de especialistas humanos (quando houver)
5. Defuzzificação: determinar um mapeamento do espaço de entrada para o espaço de saída, baseado na base de regras fuzzy combinadas utilizando o procedimento de defuzzificação *Centroide*.

4.2.1 Etapa 1

Conforme proposto por Wang e Mendel (1992), é necessário dividir os espaços de entrada e saída da base de dados em regiões fuzzy, assumindo que os intervalos de domínio das variáveis x_1 , x_2 (duas variáveis antecedentes apenas, como exemplo) e da variável de saída y são: $[x_1^-, x_1^+]$, $[x_2^-, x_2^+]$ e $[y^-, y^+]$. A partir de então, é necessário dividir esse intervalo de domínio em $2N + 1$ regiões (N é a variável que o pesquisador irá alterar para conseguir um determinado número de regiões), que podem representar termos como $\{\textit{pequeno}, \textit{médio}, \textit{grande}\}$. Com isso, são atribuídas então funções de pertinência a cada região.

Para a realização desta pesquisa foram utilizadas funções de pertinência do tipo triangular, detalhadas no Capítulo 2. Por exemplo, é possível visualizar na Figura 4.1 a função de pertinência utilizada para o primeiro atributo da base de dados Iris levando em conta 3 regiões ($2N + 1$, com $N = 1$) e considerando este atributo normalizado.

4.2.2 Etapa 2

É necessário determinar os graus das variáveis de entrada e da variável de saída em diferentes regiões. Por exemplo, $x_1^{(1)}$ pode ter grau 0.3 na região *pequeno*, 0.7 na região *medio* e 0.9 na região *grande*. $x_2^{(1)}$, por exemplo, pode ter grau 0.7 na região *pequeno*, 0.6 na região *medio* e 0.3 na região *grande*, pode-se supor, para exemplo, que o grau máximo de y foi 0.8 na região *medio*. Em seguida, serão escolhidas a regiões que contém o maior grau, neste exemplo, serão as regiões *grande* para $x_1^{(1)}$ e *pequeno* para $x_2^{(1)}$, gerando uma regra na forma:

$$\textit{IF } x_1 \textit{ is grande and } x_2 \textit{ is pequeno, THEN } y \textit{ is medio} \quad (4.2)$$

4.2.3 Etapa 3

Duas regras são conflitantes quando possuem os mesmos antecedentes e consequentes diferentes. A solução é atribuir um grau a cada regra e permanecer apenas a regra que obtiver o maior grau, reduzindo também o volume de regras. A estratégia utilizada por

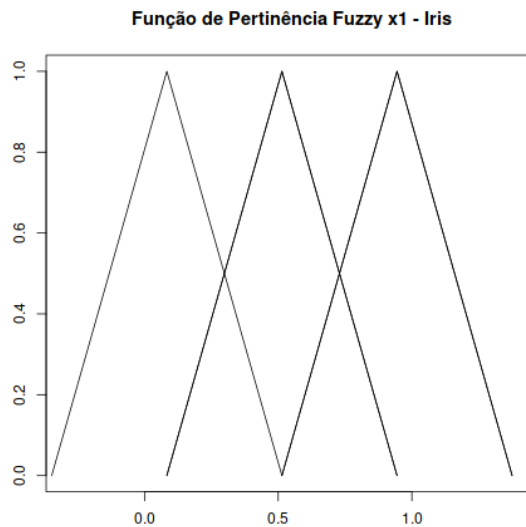


Figura 4.1: Função de pertinência para o atributo [1] sepal length da base Iris. Com 3 regiões ($N = 1$)

Wang e Mendel (1992) para atribuir graus às regras é por meio do produto dos graus dos antecedentes às regras. Para a regra utilizada como exemplo na etapa anterior: "IF x_1 is grande and x_2 is pequeno, THEN y is medio", o grau para esta regra, descrito por Wang e Mendel (1992) como $D(Rule)$, é definido como:

$$D(Rule) = m_{grande}(x_1)m_{pequeno}(x_2)m_{medio}(y). \quad (4.3)$$

onde, m representa o grau máximo que uma certa variável obteve.

Por mais que regras fuzzy sejam geradas automaticamente é interessante a opinião de um especialista humano. O modelo proposto por Wang e Mendel (1992) permite que seja atribuído a cada regra um grau de especialista quando este visualiza que um certo conjunto de dados é relevante. Para obtenção dos resultados presentes nesta monografia não houve a presença de um especialista humano, o que não afeta o processo de geração de regras pois é similar ao especialista atribuir grau 1 a todas as regras da base de regras.

4.2.4 Etapa 4

A quarta etapa é realizada com o intuito de preencher a base de regras difusas. Caso havendo mais de um regra no mesmo conjunto de regras, a que tiver maior grau é sempre a que permanece.

4.2.5 Etapa 5

A quinta etapa utiliza uma técnica de defuzzificação para determinar o valor da variável de saída y dadas as entradas (x_1, x_2) , por exemplo. Primeiro, os antecedentes da i -ésima

regra fuzzy são combinados por meio de um operador de agregação para determinar o grau de saída $m_{O_i}^i$, considerando as entradas (x_1, x_2) , ou seja:

$$m_{O_i}^i = m_{I_1^i} x_1 m_{I_2^i} x_2 \quad (4.4)$$

O_i equivale à região de saída da *Regra i*, e I_j^i equivale à região de entrada da *Regra i* para a j -ésima variável de entrada. Wang e Mendel (1992) utiliza o método de defuzzificação *Centroide* para determinar o valor de saída:

$$y = \frac{\sum_{i=1}^K m_{O_i}^i \bar{y}^i}{\sum_{i=1}^K m_{O_i}^i} \quad (4.5)$$

onde, \bar{y}^i representa o valor central da região O_i (menor valor cuja função de pertinência tem valor igual a um), e K equivale ao número de regras fuzzy da base de regras.

Nele, é realizado o processo de defuzzificação por meio da técnica de Centro de Gravidade, apresentada no Capítulo 2, com base na entrada de novas instâncias.

Estas 5 etapas para geração de regras fuzzy, proposto por Wang e Mendel (1992) foi implementado na linguagem Java¹, sendo responsabilidade dele gerar a base de regras que serve como entrada para o Fuzzy-Prolog e a JFuzzyLogic² (biblioteca Java que implementa a Linguagem de Controle Fuzzy), que terão seus resultados comparados na Seção 4.4.

4.3 ABORDAGEM PROPOSTA

A Figura 4.2 retrata como foram todas as etapas desta monografia, desde a coleta das bases de dados até a obtenção e comparação dos resultados entre a abordagem com Fuzzy-Prolog e a ferramenta JFuzzyLogic (que utiliza a *Fuzzy Control Language (FCL)*³).

A Figura 4.2 está dividida em 4 etapas:

1. A primeira etapa refere-se à obtenção das bases de dados do repositório online *UCI Machine Learning Repository*⁴, (Fisher, 1936).
2. A segunda etapa diz respeito ao pré-processamento realizado sobre essas bases de dados. Pré-processamento é o processo de limpeza e preparação dos dados para extração de padrões. É realizado com a aplicação de várias técnicas para captação, organização, tratamento e a preparação dos dados. Neste trabalho foram realizadas as tarefas de limpeza dos dados, conversão de valores nominais para numéricos e normalização dos atributos. Cada base de dados foi dividida em dois subconjuntos, um conjunto de treinamento, com $\frac{2}{3}$ das instâncias da base de dados original e um

¹<http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html>

²<http://jfuzzylogic.sourceforge.net/html/index.html>

³A FCL padroniza a programação de sistemas em lógica Fuzzy e reduz os tempos de programação. [http://jfuzzylogic.sourceforge.net/html/pdf/iec_1131_7.cd1.pdf]

⁴<https://archive.ics.uci.edu/ml/index.php>

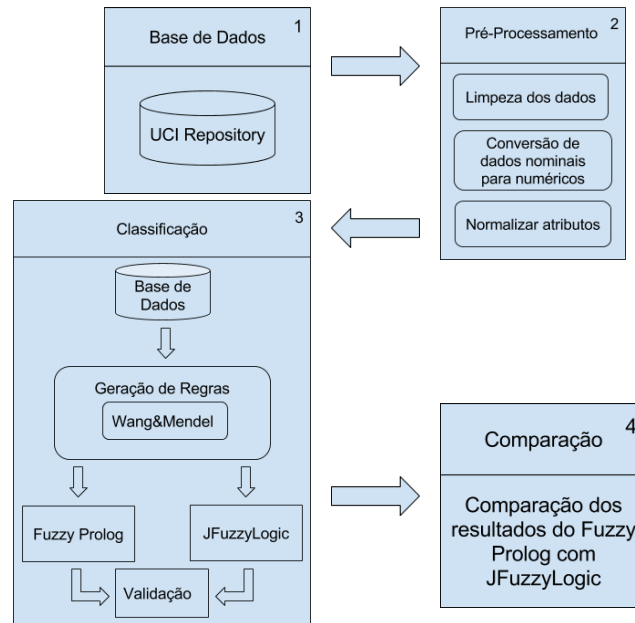


Figura 4.2: Desenho do processo de experimentação que será abordado mais adiante nesta monografia

conjunto de teste, com $\frac{1}{3}$ das instâncias da base de dados original. As instâncias de cada um desses conjuntos foram escolhidas aleatoriamente.

3. A terceira etapa refere-se ao processo de classificação. Segundo [Webb \(2010\)](#), em Aprendizado de Máquina, o termo classificação é comumente associado a um tipo particular de aprendizado, onde os exemplos de uma ou mais classes, rotuladas com o nome da classe, são atribuídos ao algoritmo de aprendizagem. O algoritmo produz um classificador que mapeia as propriedades desses exemplos, normalmente expressos como pares atributo-valor, para os rótulos das classes. Inicialmente, no processo de classificação realizado neste trabalho, as bases de dados pré-processadas foram utilizadas como entrada para o algoritmo Wang&Mendel que gerou como saída uma base de regras para cada base de dados e foi utilizada, tanto pelo Fuzzy Prolog, quanto pela JFuzzyLogic, para prever a classe a qual cada instância da base de dados de treinamento pertencia.
4. Por fim, é realizada a comparação dos resultados obtidos pelo Fuzzy Prolog e pela JFuzzyLogic através da análise dos Erros Médios Absolutos (EMA). Faz-necessário uma forma de avaliar a precisão das previsões geradas tanto pelo Fuzzy-Prolog quanto pela JFuzzyLogic. Abordado em [Webb \(2010\)](#), o Erro Médio Absoluto-será uma importante forma de comparação dos resultados de previsão dessas duas abordagens apresentadas nessa monografia.

Nas seções a seguir são apresentadas informações das bases de dados utilizadas, como as suas características, origem e composição destas bases; um estudo sobre o comportamento da classificação quando são ajustados alguns parâmetros da etapa de geração das regras fuzzy; as propostas sugeridas por essa monografia; e, por fim, os resultados obtidos com os experimentos realizados.

4.4 EXPERIMENTOS

As bases de dados utilizadas neste trabalho são: *Iris*, *Wine Quality*, *Contraceptive Method Choice*, *Abalone* e *Wireless*. Todas são bem conhecidas na literatura e são livremente apresentadas a seguir.

A base de dados *Iris*⁵ é bastante conhecida na literatura de reconhecimento de padrões, é um conjunto de dados totalmente balanceado: contém 150 instâncias, das quais 50 pertencem à cada uma das 3 classes de flores (Setosa, Versicolour, Virginica). Os 4 atributos numéricos preditos, dados em cm, referem-se às características de comprimento e largura de sépala e pétala. É uma base de características multivariadas, com atributos reais e voltada para classificação.

A base de dados *Wine Quality*⁶ apresenta dois modelos, um do vinho vermelho e outro do vinho branco português. Apenas as variáveis físico-químicas (entradas) e sensoriais (saída) estão disponíveis. Este conjunto de dados além de ser usado para classificação, pode ser útil também para regressão. O modelo para vinho vermelho contém 1599 instâncias, já o modelo para vinho branco contém 4898 instâncias, ambos possuem 11 atributos e a classe que define a qualidade do vinho por uma pontuação entre 0 e 10.

*Contraceptive Method Choice*⁷ contém amostras de mulheres casadas que não estavam grávidas ou não sabiam se estavam no momento da entrevista. O problema é prever a escolha atual do método contraceptivo (sem uso, métodos de longo prazo ou métodos de curto prazo) de uma mulher com base em suas características demográficas e socioeconômicas. Este conjunto de dados apresenta 9 atributos e 1473 instâncias.

A base de dados *Abalone*⁸ apresenta 8 atributos, com valores categóricos, inteiros e reais. Trata-se de características de um gênero de moluscos, apresentadas por meio de 4177 instâncias. O objetivo é prever a idade do abalone por meio do número de anéis (valores entre 0 e 29).

A base de dados mais nova das escolhidas nesta monografia é a *Wireless Indoor Localization*⁹, doada ao *UCI Machine Learning Repository* em dezembro de 2017. Apresenta 7 atributos reais que representam a força do sinal wi-fi observado no smartphone. As 2000 instâncias foram recolhidas para realizar experimentos sobre como os pontos fortes do sinal de wi-fi podem ser usados para determinar uma das localizações internas. As classes que representam essas localizações variam entre 1 e 4.

Outro aspecto não menos importante, são as características particulares das bases de

⁵<http://archive.ics.uci.edu/ml/datasets/Iris>

⁶<http://archive.ics.uci.edu/ml/datasets/Wine+Quality>

⁷<https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice>

⁸<https://archive.ics.uci.edu/ml/datasets/abalone>

⁹<https://archive.ics.uci.edu/ml/datasets/Wireless+Indoor+Localization>

dados. Pois ressalta-se que, para uma análise mais apurada dos resultados, é pertinente considerar as particularidades de cada modelo, com a finalidade de encontrar possíveis justificativas para os resultados apresentados, realizando-se indagações comparativas às peculiaridades sabidamente conhecidas dos métodos analisados. A Tabela 4.1 apresenta um resumo das características de cada base utilizada.

Base	#atributos	#instâncias	#classes
Iris	4	150	3
Wine-Red	11	1599	11
Wine-White	11	4898	11
Contraceptive	9	1472	3
Abalone	8	4177	29
Wireless	7	2000	4

Tabela 4.1: Resumo das características das bases de dados

4.4.1 Resultados e discussão da base de dados Iris

Uma das mais famosas bases de dados em vários campos de Inteligência Artificial, o conjunto de dados Iris também foi utilizado na obtenção de resultados nesta monografia. As informações referentes ao conjunto de dados de treinamento para geração da base de regras são apresentadas na Tabela 4.2.

Atributo	Mínimo	Máximo	Média	Mediana	Instâncias (treino): 100
Sepal length	0.0000	1.0000	0.4508	0.4583	
Sepal width	0.0000	1.0000	0.4271	0.4167	
Petal length	0.0170	0.9661	0.5027	0.5932	
Petal width	0.0000	1.0000	0.4958	0.5417	
Species (class)	0.0000	0.5500	0.5000	0.5000	

Tabela 4.2: Características dos atributos do conjunto de treinamento da base de dados *Iris*

Na Tabela 4.3 estão os erros obtidos quando a base de regras é utilizada pelo Fuzzy-Prolog e pela JFuzzyLogic. Como pode ser observado, a abordagem Fuzzy-Prolog obteve os menores erros médios absolutos com 3 e 5 Regiões de Saída (RS) e 58 Regras Geradas (RG).

As Figuras 4.3 e 4.4 mostram a comparação dos Erros Médios Absolutos (EMA) da abordagem Fuzzy-Prolog com a ferramenta JFuzzyLogic.

Como pode ser observado no gráfico da Figura 4.3, a coluna 7-3, que representa as regras com 7 RA e 3 RS, apresenta o menor erro médio absoluto, obtido com a abordagem Fuzzy-Prolog (coluna em azul). A coluna em vermelho representa os EMAs da ferramenta JFuzzyLogic, que obteve o menor EMA com 7 RA e 3 RS.

Abordagem	Erro Abs.	Erro Médio Abs.	RA	RS	RG
Fuzzy-Prolog	5.2198	0.1044	3	3	14
Fuzzy-Prolog	1.4338	0.0287	5	3	38
Fuzzy-Prolog	0.5222	0.0104	7	3	58
Fuzzy-Prolog	5.2198	0.1044	3	5	14
Fuzzy-Prolog	1.4338	0.0287	5	5	38
Fuzzy-Prolog	0.5222	0.0104	7	5	58
JFuzzyLogic	8.4419	0.1688	3	3	14
JFuzzyLogic	2.4670	0.0493	5	3	38
JFuzzyLogic	1.1694	0.0234	7	3	58
JFuzzyLogic	9.1993	0.1840	3	5	14
JFuzzyLogic	2.6320	0.0526	5	5	38
JFuzzyLogic	1.2422	0.0248	7	5	58

Tabela 4.3: Erros obtidos na classificação da base de dados Iris. **RA** (Regiões de Atributos), **RS** (Regiões da Saída) e **RG** (Regras Geradas)

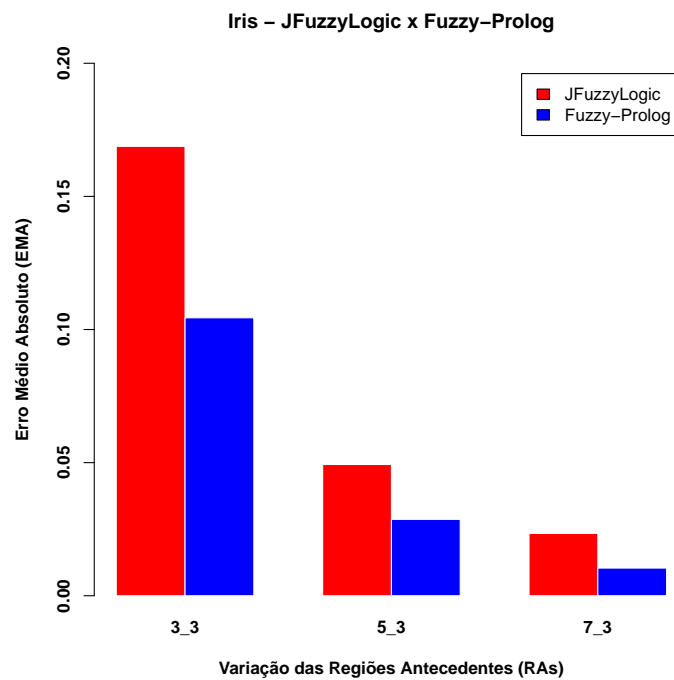


Figura 4.3: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Iris. Regiões de Saída fixas em 3.

Na Figura 4.4, a coluna com 7 RA e 3 RS (7_3) apresenta o menor Erro Médio Absoluto das duas abordagens. É notório que não houveram grandes mudanças utilizando 3 (Figura 4.3) ou 5 Regiões de Saída.

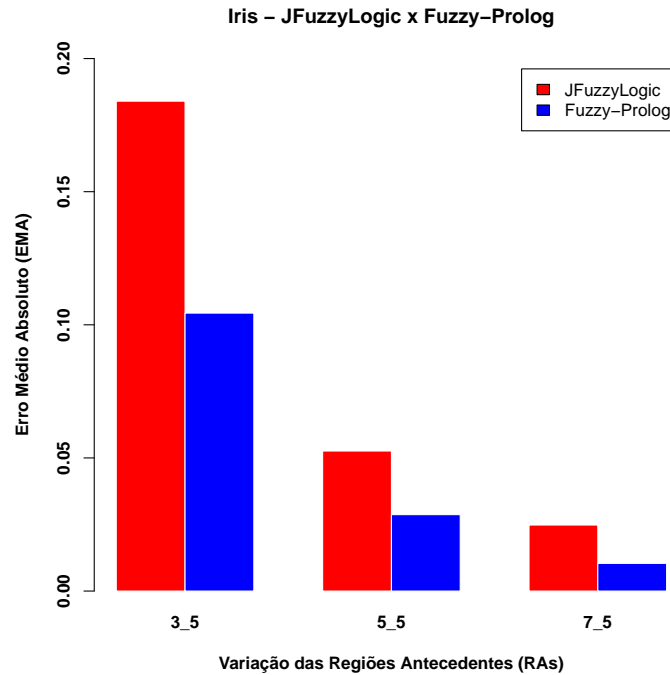


Figura 4.4: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Iris. Regiões de Saída fixas em 5.

4.4.2 Resultados e discussão da base de dados Wine Quality

Este conjunto de dados apresenta duas bases de dados, a que se refere à *Wine-White* e *Wine-Red*, portanto nesta subseção serão apresentadas tabelas para ambos conjuntos de dados referentes à base de dados *Wine Quality*. As Tabelas 4.4 e 4.5 apresentam algumas informações e características dos dados do conjunto de treinamento de *Wine-Red* e *Wine-White*, respectivamente.

A Tabela 4.6 apresenta os Erros Médios Absolutos (EMAs) obtidos após utilizar a base de regras e o conjunto de teste da base de dados Wine-Red como entrada para a abordagem Fuzzy-Prolog e para a ferramenta JFuzzyLogic. Pode-se notar que o melhor resultado, com o menor EMA, foi obtido pela abordagem Fuzzy-Prolog, com 5 RA, 11 RS e 644 regras geradas.

A Tabela 4.7 apresenta os Erros Médios Absolutos (EMAs) obtidos após utilizar a base de regras e o conjunto de teste da base de dados Wine-White como entrada para a abordagem Fuzzy-Prolog e para a ferramenta JFuzzyLogic. Pode-se notar que o menor Erro Médio Absoluto (EMA), foi obtido pela abordagem Fuzzy-Prolog, com 5 RA, 11 RS e 1233 regras geradas.

Atributo	Mínimo	Máximo	Média	Mediana	Instâncias (treino): 1066
Fixed acidity	0.0000	1.0000	0.3290	0.2920	
Volatile acidity	0.0000	1.0000	0.2798	0.2740	
Citric acid	0.0000	0.7900	0.2685	0.2500	
Residual sugar	0.0206	1.0000	0.1108	0.0890	
Chlorides	0.0000	0.7596	0.1242	0.1119	
Free sulfur dioxide	0.0000	0.9437	0.2073	0.1831	
Total sulfur dioxide	0.0000	0.9611	0.1425	0.1095	
Density	0.0419	1.0000	0.4923	0.4905	
pH	0.1024	1.0000	0.4528	0.4488	
Sulphates	0.0000	0.9880	0.1958	0.1737	
Alcohol	0.0000	1.0000	0.3088	0.2615	
Quality (class)	0.0000	1.0000	0.5231	0.6000	

Tabela 4.4: Características dos atributos do conjunto de treinamento da base de dados *Wine Quality - Red*

Atributo	Mínimo	Máximo	Média	Mediana	Instâncias (treino): 3265
Fixed acidity	0.0000	0.7692	0.2942	0.2885	
Volatile acidity	0.0000	1.0000	0.1941	0.1765	
Citric acid	0.0000	0.7410	0.2008	0.1867	
Residual sugar	0.0015	0.4755	0.0895	0.0706	
Chlorides	0.0000	1.0000	0.1092	0.1009	
Free sulfur dioxide	0.0035	1.0000	0.1162	0.1115	
Total sulfur dioxide	0.0000	1.0000	0.3004	0.2923	
Density	0.0004	0.4471	0.1339	0.1290	
pH	0.0182	0.9818	0.4243	0.4182	
Sulphates	0.0116	1.0000	0.3129	0.2907	
Alcohol	0.0000	1.0000	0.4029	0.3710	
Quality (class)	0.0000	1.0000	0.4772	0.5000	

Tabela 4.5: Características dos atributos do conjunto de treinamento da base de dados *Wine Quality - White*

Abordagem	Erro Abs.	Erro Médio Abs.	RA	RS	RG
Fuzzy-Prolog	63.3504	0.1189	3	3	238
Fuzzy-Prolog	66.4594	0.1247	5	3	644
Fuzzy-Prolog	84.4871	0.1585	7	3	822
Fuzzy-Prolog	61.5972	0.1156	3	11	238
Fuzzy-Prolog	60.3541	0.1132	5	11	644
Fuzzy-Prolog	69.2235	0.1299	7	11	822
JFuzzyLogic	64.0775	0.1202	3	3	238
JFuzzyLogic	72.8287	0.1366	5	3	644
JFuzzyLogic	85.6092	0.1606	7	3	822
JFuzzyLogic	64.6061	0.1212	3	11	238
JFuzzyLogic	66.3535	0.1245	5	11	644
JFuzzyLogic	73.3307	0.1376	7	11	822

Tabela 4.6: Erros obtidos na classificação da base de dados Wine-Red. **RA** (Regiões de Atributos), **RS** (Regiões da Saída) e **RG** (Regras Geradas).

Abordagem	Erro Abs.	Erro Médio Abs.	RA	RS	RG
Fuzzy-Prolog	180.8290	0.1108	3	3	382
Fuzzy-Prolog	176.5438	0.1082	5	3	1233
Fuzzy-Prolog	186.0380	0.1140	7	3	2242
Fuzzy-Prolog	179.3203	0.1099	3	11	382
Fuzzy-Prolog	161.7942	0.0991	5	11	1233
Fuzzy-Prolog	161.8038	0.0992	7	11	2242
JFuzzyLogic	187.2474	0.1147	3	3	382
JFuzzyLogic	189.1907	0.1159	5	3	1233
JFuzzyLogic	202.3055	0.1240	7	3	2242
JFuzzyLogic	185.9473	0.1139	3	11	382
JFuzzyLogic	173.4904	0.1063	5	11	1233
JFuzzyLogic	172.5269	0.1057	7	11	2242

Tabela 4.7: Erros obtidos na classificação da base de dados Wine-White. **RA** (Regiões de Atributos), **RS** (Regiões da Saída) e **RG** (Regras Geradas).

Na Figura 4.5, a coluna 3_3 (3 RA e 3 RS) apresenta o menor Erro Médio Absoluto, tanto da abordagem Fuzzy-Prolog quanto da ferramenta JFuzzyLogic.

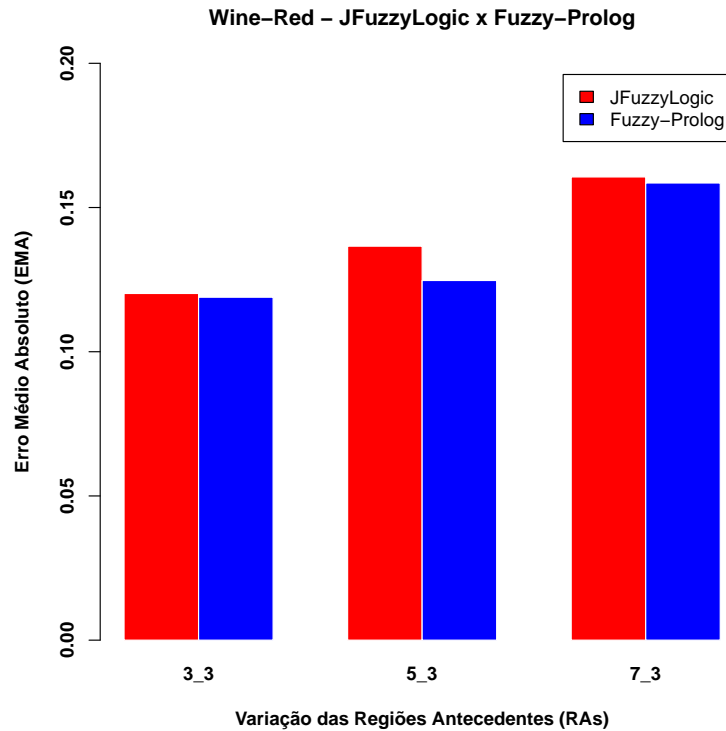


Figura 4.5: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-Red. Regiões de Saída fixas em 3.

Na Figura 4.6 é possível visualizar a coluna que apresenta o menor Erro Médio Absoluto obtido, quando foram utilizadas 5 RA e 11 RS na abordagem Fuzzy-Prolog.

A coluna 3_3 do gráfico da Figura 4.7 mostra o menor EMA da ferramenta JFuzzyLogic na base de dados Wine-White. A coluna 5_3 contém o menor EMA da abordagem Fuzzy-Prolog considerando as Regiões de Saída fixas em 3.

As diferenças entre as colunas 5_11 e 7_11 presentes no gráfico da Figura 4.8 são quase imperceptíveis, no entanto, com auxílio da Tabela 4.7 é possível identificar que o menor EMA foi obtido pela abordagem Fuzzy-Prolog, com 5 RA e 11 RS.

4.4.3 Resultados e discussão da base de dados Contraceptive Method Choice

São apresentadas na Tabela 4.8 informações referentes ao conjunto de treinamento da base de dados “Contraceptive Method Choice”, que representa $\frac{2}{3}$ dos dados originais. Essas são importantes informações presentes em todas as bases de dados que irão definir a eficiência das regras, os intervalos das funções de pertinência fuzzy e o número de regras da base de regras. Esse conjunto de dados, em especial, apresenta a maior parte de seus atributos do tipo categórico.

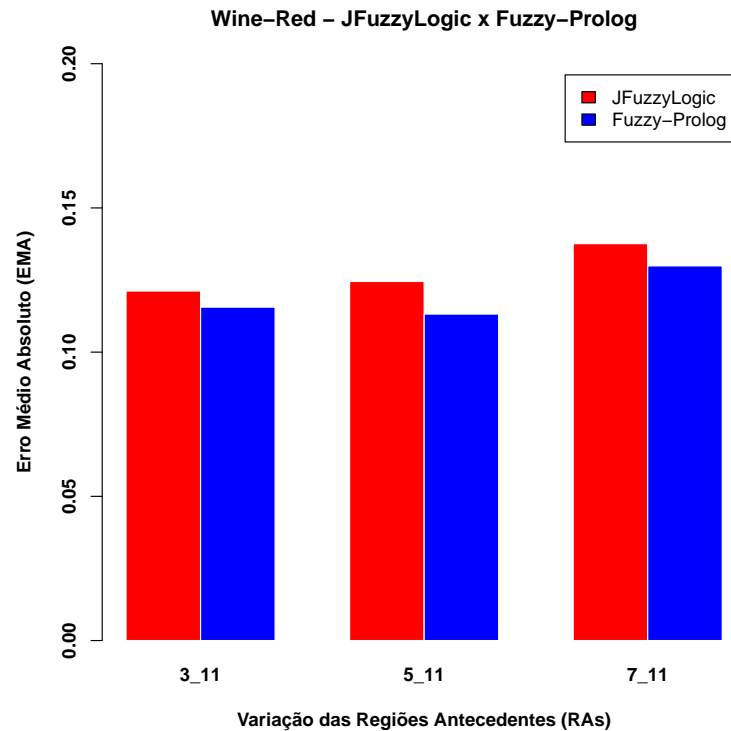


Figura 4.6: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-Red. Regiões de Saída fixas em 11.

Atributo	Mínimo	Máximo	Média	Mediana	Instâncias (treino): 982
Idade da esposa	0.0000	1.0000	0.4947	0.4545	
Educação da esposa	0.0000	1.0000	0.6504	0.6667	
Educação do marido	0.0000	1.0000	0.8079	1.0000	
Número de filhos	0.0000	1.0000	0.1991	0.1875	
Religião da esposa	0.0000	1.0000	0.8432	1.0000	
A esposa trabalha?	0.0000	1.0000	0.7373	1.0000	
Ocupação do marido	0.0000	1.0000	0.3805	0.3333	
Padrão de vida	0.0000	1.0000	0.7159	0.6667	
Exposição na mídia	0.0000	1.0000	0.0713	0.0000	
Escolha do método (class)	0.0000	1.0000	0.4501	0.5000	

Tabela 4.8: Características dos atributos do conjunto de treinamento da base de dados *Contraceptive Method Choice*

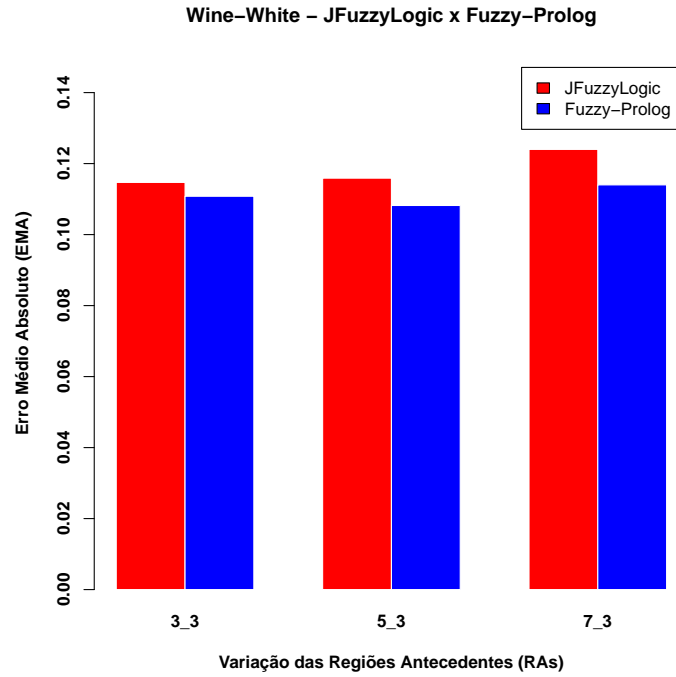


Figura 4.7: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-White. Regiões de Saída fixas em 3.

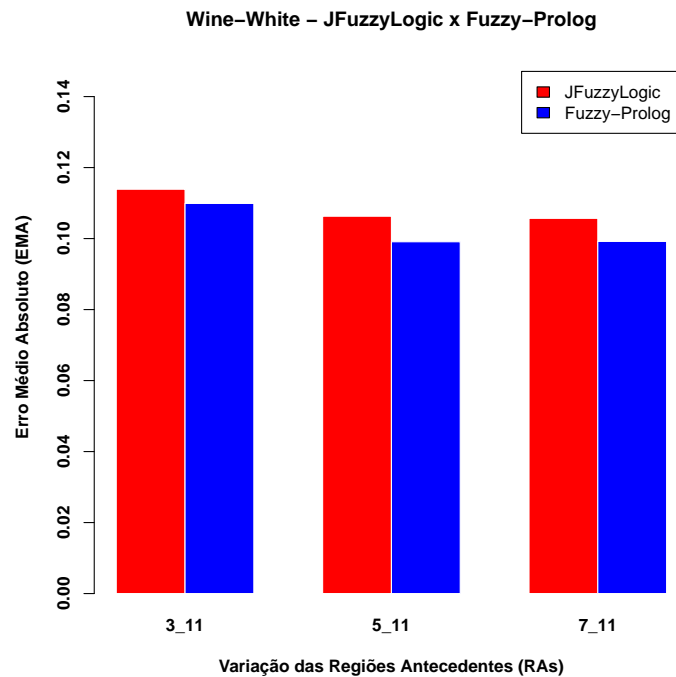


Figura 4.8: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-White. Regiões de Saída fixas em 11.

Tomando os dados de treinamento com essas características como entrada para o gerador de regras, a Tabela 4.9 pôde ser construída levando em conta as saídas geradas pelo Fuzzy-Prolog e JFuzzyLogic, cuja base de regras foi fundamentada com os dados representados na Tabela 4.8.

É possível notar na Tabela 4.9 que os erros médios absolutos obtidos pela abordagem Fuzzy-Prolog e pela ferramenta JFuzzyLogic foram bem próximos, no entanto, a ferramenta JFuzzyLogic obteve o menor erro médio absoluto para essa base de dados, com 3 RA, 11 RS e 287 regras geradas.

Abordagem	Erro Abs.	Erro Médio Abs.	RA	RS	RG
Fuzzy-Prolog	189.1379	0.3853	3	3	287
Fuzzy-Prolog	191.2044	0.3894	5	3	615
Fuzzy-Prolog	198.7118	0.4047	7	3	721
Fuzzy-Prolog	189.1379	0.3853	3	11	287
Fuzzy-Prolog	191.2044	0.3894	5	11	615
Fuzzy-Prolog	198.7118	0.4047	7	11	721
JFuzzyLogic	189.6079	0.3861	3	3	287
JFuzzyLogic	190.5215	0.3880	5	3	615
JFuzzyLogic	198.0117	0.4032	7	3	721
JFuzzyLogic	189.1320	0.3852	3	11	287
JFuzzyLogic	190.4205	0.3878	5	11	615
JFuzzyLogic	197.9021	0.4031	7	11	721

Tabela 4.9: Erros obtidos na classificação da base de dados Contraceptive. **RA** (Regiões de Atributos), **RS** (Regiões da Saída) e **RG** (Regras Geradas).

Os EMAs da base de dados *Contraceptive* visualizados nas Figuras 4.9 e 4.10 foram bem próximos, com diferenças quase imperceptíveis com a análise apenas gráfica, tanto na abordagem Fuzzy-Prolog quanto na ferramenta JFuzzyLogic. No entanto, conforme visto na Tabela 4.9, o JFuzzyLogic apresentou o menor EMA para essa base de dados, utilizando 3 RA e 11 RS.

4.4.4 Resultados e discussão da base de dados Abalone

A Tabela 4.10 apresenta algumas informações relevantes de cada atributo do conjunto de treinamento da base de dados *Abalone*. Nesta base de dados, o atributo *Sex*, originalmente nominal (M, F, e I), foi convertido em numérico (0.0, 0.5, 1.0).

Fez-se necessário uma análise dos erros gerados pela abordagem Fuzzy-Prolog e JFuzzyLogic durante o processo de classificação do subconjunto de teste da base de dados *Abalone*.

Na Tabela 4.11 é possível visualizar o desempenho dessas duas abordagens. Embora a ferramenta JFuzzyLogic tenha obtido uma média de erros baixos, a abordagem Fuzzy-Prolog obteve um erro médio absoluto muito baixo quando utilizadas 29 RS, gerando 379 regras.

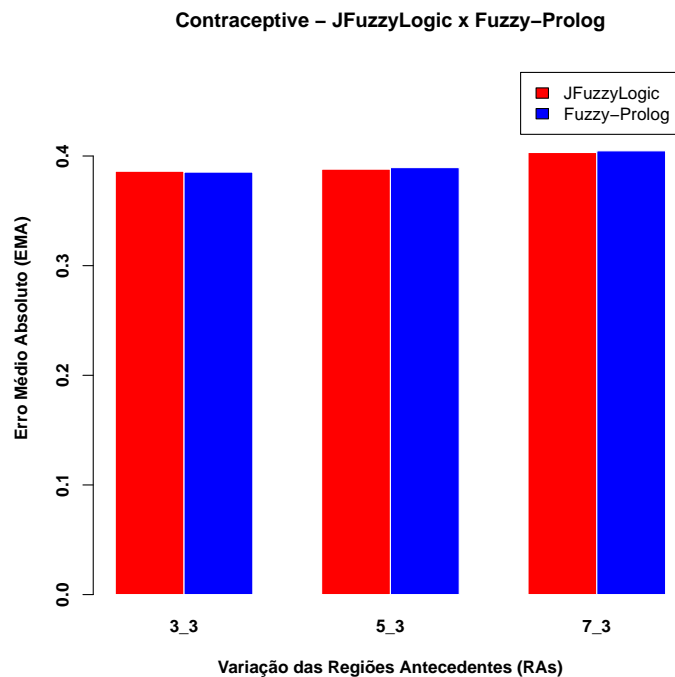


Figura 4.9: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-White. Regiões de Saída fixas em 3.

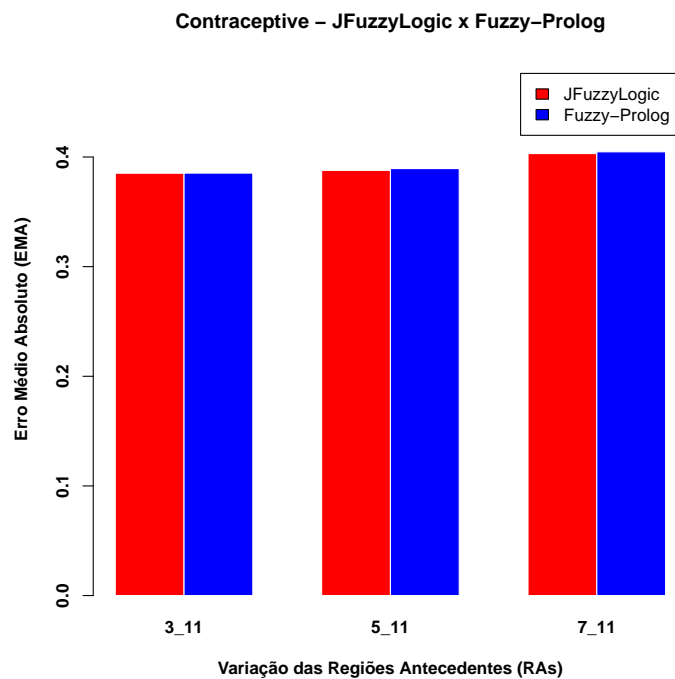


Figura 4.10: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wine-White. Regiões de Saída fixas em 11.

Atributo	Mínimo	Máximo	Média	Mediana	Instâncias (treino): 2784
Sex	0.0000	1.0000	0.4932	0.5000	
Length	0.0000	0.9527	0.6066	0.6351	
Diam	0.0000	0.9664	0.5928	0.6218	
Height	0.0089	0.4558	0.1232	0.1239	
Whole	0.0000	1.0000	0.2938	0.2822	
Shucked	0.0000	1.0000	0.2423	0.2249	
Viscera	0.0000	1.0000	0.2378	0.2245	
Shell	0.0000	1.0000	0.2370	0.2317	
Rings (class)	0.0000	1.0000	0.3176	0.2857	

Tabela 4.10: Características dos atributos do conjunto de treinamento da base de dados *Abalone*.

Abordagem	Erro Abs.	Erro Médio Abs.	RA	RS	RG
Fuzzy-Prolog	139.3530	0.1001	3	3	115
Fuzzy-Prolog	180.1130	0.1294	5	3	203
Fuzzy-Prolog	158.1717	0.1136	7	3	379
Fuzzy-Prolog	88.1630	0.0633	3	29	115
Fuzzy-Prolog	93.0523	0.0668	5	29	203
Fuzzy-Prolog	86.3424	0.0620	7	29	379
JFuzzyLogic	121.0692	0.0870	3	3	115
JFuzzyLogic	157.3133	0.1130	5	3	203
JFuzzyLogic	122.9457	0.0883	7	3	379
JFuzzyLogic	140.2617	0.1008	3	29	115
JFuzzyLogic	125.9018	0.0904	5	29	203
JFuzzyLogic	112.9934	0.0812	7	29	379

Tabela 4.11: Resultados obtidos na classificação da base de dados *Abalone*. **RA** (Regiões de Atributos), **RS** (Regiões da Saída) e **RG** (Regras Geradas).

É possível verificar no gráfico presente na Figura 4.11 que a ferramenta JFuzzyLogic obteve EMAs inferiores aos obtidos pela abordagem Fuzzy-Prolog, utilizando 3 RS fixas.

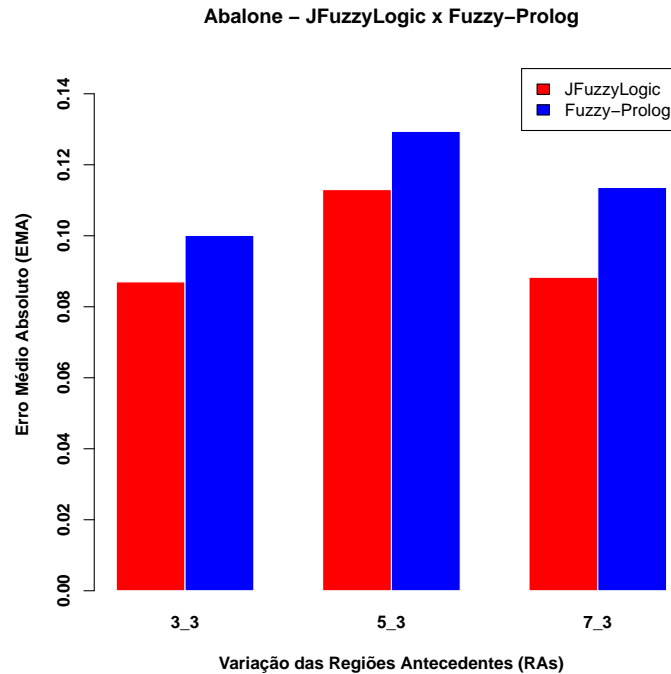


Figura 4.11: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Abalone. Regiões de Saída fixas em 3.

O gráfico da Figura 4.12 permite a identificação visual do menor EMA obtido pela abordagem Fuzzy-Prolog, presente na Tabela 4.11. É possível verificar que a coluna 7_29 do gráfico apresenta o menor EMA, obtido pela abordagem Fuzzy-Prolog.

4.4.5 Resultados e discussão da base de dados Wireless Indoor Localization

Este é o conjunto de dados mais novo dos utilizados nesta monografia. A Tabela 4.12 apresenta um sumário das características das instâncias do conjunto de dados de treinamento. Cada atributo representa a força do sinal wifi no smartphone em um certo local interno. O objetivo é determinar este local interno. Na Tabela 4.13 são apresentados os erros gerados após rodar o conjunto de teste.

É possível verificar na Tabela 4.13, em negrito, que a abordagem Fuzzy-Prolog obteve o menor erro médio absoluto no conjunto de dados de teste da base de dados *Wireless*, com 852 regras geradas, 7 RA e 5 RS.

Na Figura 4.13 os menores EMAs, tanto da abordagem Fuzzy-Prolog quanto da ferramenta JFuzzyLogic, foram obtidos na coluna 7_3.

O menor EMA obtido na base de dados *Wireless* foi obtido pela abordagem Fuzzy-Prolog e está apresentado na coluna 7_5 do gráfico presente na Figura 4.14.

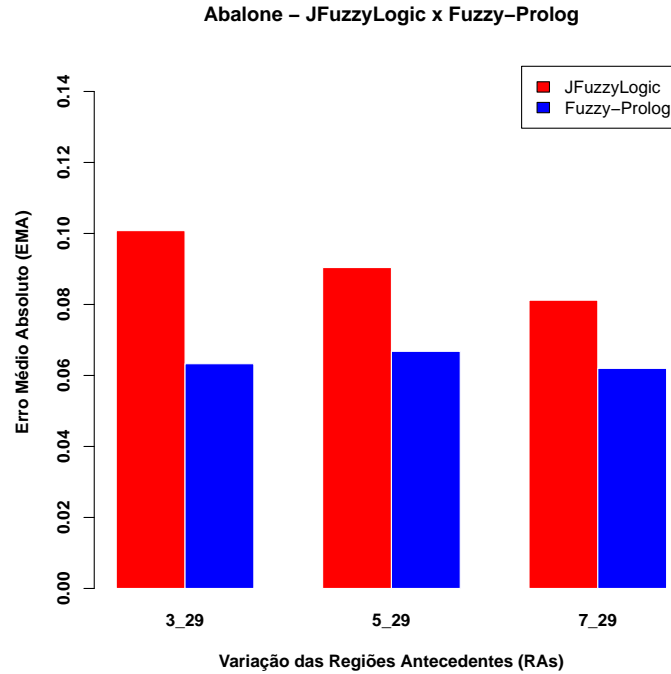


Figura 4.12: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Abalone. Regiões de Saída fixas em 29.

Atributo	Mínimo	Máximo	Média	Mediana	Instâncias (treino): 1333
V1	0.0000	1.0000	0.3414	0.2969	
V2	0.1379	1.0000	0.6340	0.6207	
V3	0.0909	1.0000	0.5431	0.5455	
V4	0.0000	1.0000	0.3569	0.3182	
V5	0.0000	1.0000	0.4951	0.4717	
V6	0.0278	1.0000	0.4467	0.4444	
V7	0.0000	0.9714	0.4670	0.4286	
V8 (class)	0.0000	1.0000	0.4936	0.3333	

Tabela 4.12: Características dos atributos do conjunto de treinamento da base de dados *Wireless Indoor Localization*

Abordagem	Erro Abs.	Erro Médio Abs.	RA	RS	RG
Fuzzy-Prolog	119.2429	0.1790	3	3	127
Fuzzy-Prolog	80.8568	0.1214	5	3	402
Fuzzy-Prolog	65.9622	0.0990	7	3	852
Fuzzy-Prolog	108.0881	0.1623	3	5	127
Fuzzy-Prolog	42.5674	0.0639	5	5	402
Fuzzy-Prolog	38.6210	0.0580	7	5	852
JFuzzyLogic	194.2227	0.2916	3	3	127
JFuzzyLogic	108.1247	0.1623	5	3	402
JFuzzyLogic	91.0750	0.1367	7	3	852
JFuzzyLogic	187.4722	0.2815	3	5	127
JFuzzyLogic	90.3384	0.1356	5	5	402
JFuzzyLogic	67.2647	0.1010	7	5	852

Tabela 4.13: Resultados obtidos na classificação da base de dados *Wireless*. **RA** (Regiões de Atributos), **RS** (Regiões da Saída) e **RG** (Regras Geradas).

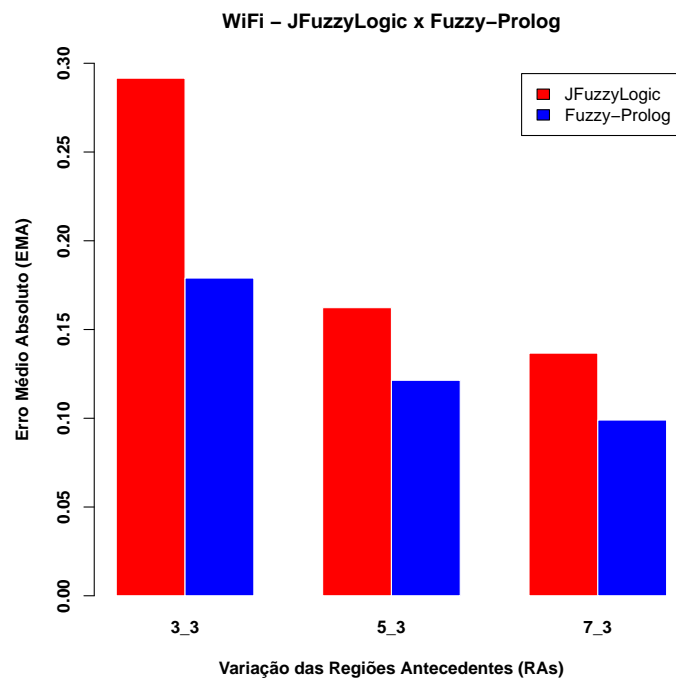


Figura 4.13: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados *Wireless*. Regiões de Saída fixas em 3.

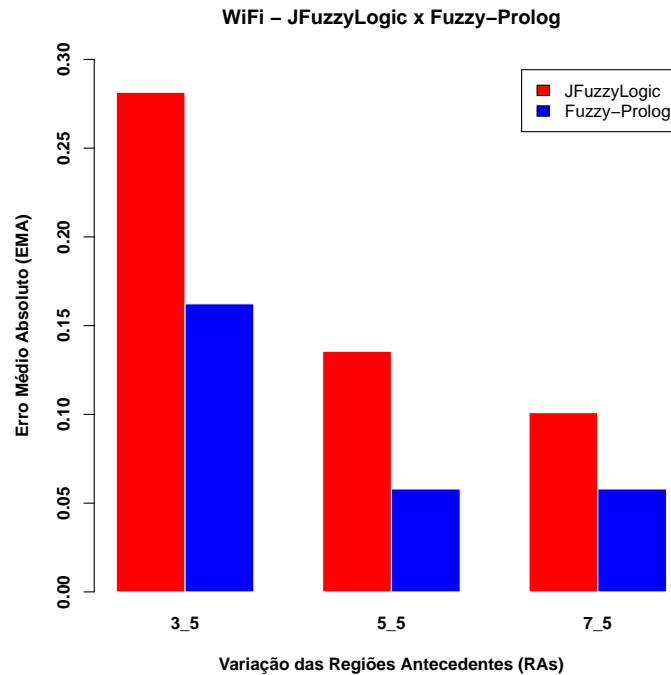


Figura 4.14: Gráfico com os Erros Médios Absolutos obtidos na classificação da base de dados Wireless. Regiões de Saída fixas em 5.

4.4.6 Discussão dos Resultados

Com o intuito de mensurar as medições observadas nas subseções desta seção, os experimentos realizados com as bases de dados utilizadas nesta monografia passaram por uma análise de seus resultados levando em consideração uma técnica de precisão da previsão que adota o Erro Médio Absoluto (EMA) como métrica (Equação 4.6). Conforme visto nesta seção, os experimentos visaram principalmente variar o número de regiões das variáveis antecedentes à regra, com o intuito de que fosse encontrado a menor diferença possível entre o valor esperado e o valor obtido, obtendo assim, portando, um Erro Médio Absoluto menor. Isso pôde ser visualizado por meio das tabelas e dos vários gráficos presentes neste capítulo.

$$EMA = \frac{\sum_{i=1}^n |e_i|}{n}; e_i = \text{erro da } i\text{-ésima instância} \quad (4.6)$$

equação do Erro Médio Absoluto, utilizado como critério de comparação dos resultados dos experimentos desta monografia.

Uma vez que as propostas apresentadas aqui tem como objetivo trazer o sistema de inferência e Lógica de Primeira Ordem do Prolog como uma opção para a lógica fuzzy por meio de sua linguagem natural, expressiva e bem próxima da linguagem humana, foram realizados os experimentos apresentados nas subseções anteriores com o intuito de obter resultados que também justifiquem a condição do Prolog ser uma opção, por suas

características de linguagem de programação lógica e por obter resultados satisfatórios.

Trabalhar com o parâmetro N , da equação $2N + 1$, vista na Seção 4.2 para determinação do número de regiões das variáveis antecedentes à regra e da variável de saída, proposto por Wang e Mendel (1992), obteve alguns resultados interessantes que foram visualizados e apresentados por meio de gráficos e tabelas variadas nas subseções desta seção.

Na próxima seção será apresentado uma síntese dos experimentos conduzidos nesse capítulo, pontuando as principais contribuições e descobertas encontradas ao longo desse processo de investigação.

4.5 CONSIDERAÇÕES FINAIS

Neste capítulo foi detalhado todo o processo de investigação realizado nessa monografia, assim como a metodologia utilizada para a realização dos experimentos, informações das bases de dados utilizadas e as devidas análises dos resultados. A abordagem Fuzzy-Prolog apresentou resultados satisfatórios quando comparado com a JFuzzyLogic. Esses resultados justificam seu uso para inferência fuzzy utilizando lógica de predicados. Por ser uma biblioteca, JFuzzyLogic acaba encapsulando algumas informações que podem ser transparentes e de fácil compreensão no Prolog.

A Tabela 4.14 a seguir traz um resumo dos resultados encontrados nos experimentos apresentados neste capítulo em cada base de dados. Nela é possível visualizar que na maior parte dos resultados, a abordagem com Fuzzy-Prolog obteve Erros Médios Absolutos menores do que JFuzzyLogic, mesmo sendo pequena esta diferença. Na base de dados *Contraceptive*, o JFuzzyLogic obteve um Erro Médio Absoluto menor do que a abordagem com Fuzzy-Prolog, sendo este o único caso em que isto aconteceu dentre todas as bases de dados analisadas nesta monografia.

	Fuzzy-Prolog		JFuzzyLogic	
	Menor EMA	Qtd. Regras	Menor EMA	Qtd. Regras
Iris	0.0104	58	0.0234	58
Wine-Red	0.1132	644	0.1202	238
Wine-White	0.0991	1233	0.1057	2242
Contraceptive	0.3853	287	0.3852	287
Abalone	0.0620	379	0.0812	379
Wireless	0.0580	852	0.1010	852

Tabela 4.14: Resumo dos resultados obtidos em cada base de dados para as duas abordagens apresentadas nesta monografia.

A Tabela 4.14 apresenta os resultados encontrados nos experimentos desta pesquisa e serve de embasamento para a utilização da programação em lógica do Prolog para realizar inferência em lógica fuzzy. Em bases de dados com classes desbalanceadas, como a Wine-Quality, por exemplo, foi necessário um número maior de regras para que resultados satisfatórios fossem obtidos. É necessário uma atenção especial à base de dados Wine-Red, embora a abordagem com Fuzzy-Prolog obteve uma menor taxa de erro médio

absoluto, foram necessárias mais que o dobro de regras utilizadas pela ferramenta JFuzzyLogic, que obteve um erro médio absoluto quase semelhante.

Em geral, a abordagem Fuzzy-Prolog obteve erros médios absolutos menores do que a ferramenta JFuzzyLogic, tendo o mesmo número ou inferior de regras geradas. Era de se esperar que ambas as abordagens tivessem o mesmo resultado, por estarem utilizando as mesmas bases de regras, as mesmas bases de dados e o mesmo método de defuzzificação. No entanto, JFuzzyLogic encapsula muitos detalhes de sua implementação, o que torna difícil esse processo de identificação das causas dessas diferenças dos erros médios absolutos. No entanto, vale ressaltar que a programação com a abordagem Fuzzy-Prolog é transparente e dar-se de forma natural, desde objetivos didáticos, como proposto por [Kaminski \(1992\)](#), até tarefas de classificação como as realizadas no decorrer deste trabalho. Comparada com uma ferramenta no estado da arte, bastante utilizada, como a JFuzzyLogic, obteve resultados tão bons quanto esta pode oferecer.

No próximo capítulo, serão abordadas as conclusões a respeito de toda a pesquisa apresentada nessa monografia.

Síntese da investigação e dos experimentos realizados nesta monografia

CONCLUSÃO

A pesquisa desenvolvida nesta monografia investigou um campo, de certa forma, pouco explorado ainda. Prolog é bem conhecido em Inteligência Artificial, assim como lógica Fuzzy já bastante renomada, conhecida e utilizada. No entanto, trabalhos que abordem a utilização desses dois grandes conceitos juntos ainda são um pouco escassos.

Um dos grandes desafios da abordagem Fuzzy-Prolog é o de incorporar lógica Fuzzy na lógica de primeira ordem do Prolog, conforme visto no Capítulo 3, no entanto, uma vez feito isso, tem-se uma linguagem bem robusta e poderosa para realizar inferência em lógica Fuzzy utilizando as várias possibilidades que o Prolog adiciona, como o *backtracking*. Uma outra abordagem que também foi apresentada durante esta monografia foi a utilização de variáveis que retornariam os graus de pertinência dos predicados, conforme pôde ser visto no exemplo 3.1, que representa uma forma mais simples e didática de introduzir lógica fuzzy em Prolog, mesma forma abordada por Kaminski (1992).

Neste sentido, essas duas lógicas juntas, Fuzzy-Prolog, foram utilizadas nesta monografia para classificação de dados.

A área de classificação de dados é bastante extensa em diversos ramos de Aprendizado de Máquina, fundamenta-se em aplicar técnicas de reconhecimento de padrões e algumas características dos atributos com o intuito de predizer a qual classe determinada instância pertence. Esta foi a forma de experimentação e obtenção de resultados abordada nesta monografia.

Nesse contexto, diversos experimentos foram realizados, permitindo comprovar a hipótese de que:

A abordagem Fuzzy-Prolog permite obter resultados expressivos provenientes da linguagem natural do Prolog, apresentando, em geral, um menor erro médio absoluto na tarefa de classificação usando regras fuzzy, em comparação com a ferramenta JFuzzyLogic.

Com base na exploração das abordagens existentes na literatura sobre inferência Fuzzy utilizando lógica de primeira ordem do Prolog e da comprovação da hipótese formulada, foi alcançado o objetivo de:

Conduzir uma investigação em torno de inferência Fuzzy com Prolog e com Java para compreender o comportamento das linguagens para execução de conhecimento incerto.

Visando alcançar esse objetivo, foi realizado nesta monografia um estudo dos fundamentos da lógica Fuzzy e da lógica de primeira ordem do Prolog e uma revisão das estratégias utilizadas por pesquisadores da área utilizando diferentes abordagens de implementação de lógica fuzzy no Prolog. E, por fim, como principal resultado, o estudo do impacto da utilização dessa abordagem em tarefas de classificação.

Uma contribuição desta monografia dar-se ao fato de terem sido apresentadas variadas pesquisas sobre o tema. As pesquisas trazem abordagens diferentes sobre o uso do Fuzzy-Prolog, o que enriquece o processo de pesquisa.

Uma outra contribuição importante é dada aos desenvolvedores de sistemas especialistas, que precisam lidar com a necessidade de criação de uma base inicial de regras que o sistema irá consultar. Classificar é uma atividade muito semelhante à gerar um sistema especialista e assim como foi utilizado aqui, o Wang&Mendel se torna útil na geração das regras também em sistemas especialistas que podem, além disso, utilizar Fuzzy-Prolog em sua construção.

5.1 TRABALHOS FUTUROS

Ao considerar todas as análises realizadas no Capítulo 4, existem algumas possibilidades que podem derivar deste trabalho. Em especial, é necessária uma análise mais apurada acerca das variações do N de Wang&Mendel (que foi analisado no Capítulo 4) e das implicações no erro médio absoluto e no número total de regras com essas variações.

Essa variação do N do Wang&Mendel pode ser feita, inclusive de forma dinâmica, escolhendo o melhor N para cada variável de acordo com as características das instâncias de cada variável. É algo que pode ser analisado em trabalhos futuros.

Uma outra pretensão, um pouco maior, é utilizar esse Fuzzy-Prolog em decisões em tempo real em robótica. O comportamento do robô abarca muitas decisões que são imprecisas e recebe muitos dados de sensores que necessitam de uma análise fuzzy, dessa forma, esta seria uma abordagem muito interessante para ser explorada em um futuro próximo.

REFERÊNCIAS BIBLIOGRÁFICAS

- BALDWIN, J. F.; MARTIN, T. P.; PILSWORTH, B. W. *FriI- Fuzzy and Evidential Reasoning in Artificial Intelligence*. New York, NY, USA: John Wiley & Sons, Inc., 1995. ISBN 047195523X.
- BRATKO, I. *PROLOG Programming for Artificial Intelligence*. 2nd. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1990. ISBN 0201416069.
- COLMERAUER, A.; ROUSSEL, P. History of programming languages—ii. In: BERGIN JR., T. J.; GIBSON JR., R. G. (Ed.). New York, NY, USA: ACM, 1996. cap. The Birth of Prolog, p. 331–367. ISBN 0-201-89502-1. Disponível em: <http://doi.acm.org/10.1145/234286.1057820>.
- FISHER, R. *UCI Machine Learning Repository*. 1936. Disponível em: <http://archive.ics.uci.edu/ml>.
- GISOLFI, A. A prolog implementation of a fuzzy algebraic structure for the uncertainty modelling. In: *[1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis*. [S.l.: s.n.], 1990. p. 1–6.
- GUADARRAMA, S.; MUÑOZ, S.; VAUCHERET, C. Fuzzy prolog: a new approach using soft constraints propagation. *Fuzzy Sets and Systems*, v. 144, n. 1, p. 127 – 150, 2004. ISSN 0165-0114. Possibilistic Logic and Related Issues. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0165011403004494>.
- ISHIZUKA, M.; KANAI, N. Prolog-elf incorporating fuzzy logic. *New Generation Computing*, v. 3, n. 4, p. 479, Dec 1985. ISSN 1882-7055. Disponível em: <https://doi.org/10.1007/BF03037082>.
- KAMINSKI, D. M. Introducing the fuzzy paradigm using prolog. *SIGCSE Bull.*, ACM, New York, NY, USA, v. 24, n. 1, p. 202–206, mar. 1992. ISSN 0097-8418. Disponível em: <http://doi.acm.org/10.1145/135250.134551>.
- KLIR, G. J.; YUAN, B. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1995. ISBN 0-13-101171-5.
- LEE, R. C. T. Fuzzy logic and the resolution principle. *J. ACM*, ACM, New York, NY, USA, v. 19, n. 1, p. 109–119, jan. 1972. ISSN 0004-5411. Disponível em: <http://doi.acm.org/10.1145/321679.321688>.
- LI, D.; LIU, D. *A Fuzzy Prolog Database System*. New York, NY, USA: John Wiley & Sons, Inc., 1990. ISBN 0-471-92762-7.

MARTIN, T. P.; BALDWIN, J. F.; PILSWORTH, B. W. The implementation of fprolog—a fuzzy prolog interpreter. *Fuzzy Sets Syst.*, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, The Netherlands, v. 23, n. 1, p. 119–129, jul. 1987. ISSN 0165-0114. Disponível em: [http://dx.doi.org/10.1016/0165-0114\(87\)90104-7](http://dx.doi.org/10.1016/0165-0114(87)90104-7).

MUNOZ-HERNANDEZ, S.; WIGUNA, W. S. Fuzzy prolog as cognitive layer in robocupsoccer. In: *2007 IEEE Symposium on Computational Intelligence and Games*. [S.l.: s.n.], 2007. p. 340–345. ISSN 2325-4270.

NOGUEIRA, T. M. *Modelagem Fuzzy usando Agrupamento Condicional*. 2008.

RUBIO-MANZANO, C.; JULIAN-IRANZO, P. Reasoning with words: A first approximation. In: *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.: s.n.], 2014. p. 569–574. ISSN 1098-7584.

WANG, L. X.; MENDEL, J. M. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 22, n. 6, p. 1414–1427, Nov 1992. ISSN 0018-9472.

WEBB, G. I. Model evaluation. In: _____. *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. p. 683–683. ISBN 978-0-387-30164-8. Disponível em: https://doi.org/10.1007/978-0-387-30164-8_550.

YASUI, H.; HAMADA, Y.; MUKAIDONO, M. Fuzzy prolog based on lukasiewicz implication and bounded product. In: *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*. [S.l.: s.n.], 1995. v. 2, p. 949–954 vol.2.

ZADEH, L. Fuzzy sets. *Information and Control*, v. 8, n. 3, p. 338 – 353, 1965. ISSN 0019-9958. Disponível em: <http://www.sciencedirect.com/science/article/pii/S001999586590241X>.